

# ALOCAÇÃO DE POLOS EM REGIÕES DO PLANO COMPLEXO PARA SISTEMAS DISCRETOS VIA LMIs

Alexandre Nascimento, Jr.

25 de outubro de 2022

## Sumário

<b>Sumário</b>	<b>1</b>
<b>1 INTRODUÇÃO</b>	<b>3</b>
<b>2 REGIÃO DE DESEMPENHO GARANTIDO</b>	<b>5</b>
2.1 Regiões de polos para sistemas contínuos	5
2.2 Regiões de polos para sistemas discretos	6
2.3 Aproximação das regiões do plano $z$ via LMIs	8
2.4 Condição de Liapunov	9
<b>3 ALGORITMO</b>	<b>11</b>
3.1 Aproximação cônica	11
3.2 Aproximação elíptica	13
3.3 Aproximação poligonal	13
<b>4 TESTES E SIMULAÇÕES</b>	<b>17</b>
<b>5 CONCLUSÃO</b>	<b>19</b>
<b>REFERÊNCIAS</b>	<b>21</b>



# 1 Introdução



## 2 Região de Desempenho Garantido

A alocação de polos é uma das principais ferramentas da teoria de controle, pois a partir desta, é possível projetar um sistema que seja estável e que tenha um bom desempenho Rosinová e Holíč (2014). A operação de alocar polos de um sistema linear dentro de uma região específica é chamada  $\mathcal{D}$ -estabilidade Wisniewski et al. (2017).

Entende-se por estável o sistema que, em termos de resposta a estímulos, possui uma convergência ao zero da resposta natural, restando apenas a resposta forçada (NISE, 2011). Assim, para um intervalo de tempo determinado, espera-se que o sistema apenas tenha dinâmica referente à entrada aplicada. Neste contexto, a estabilidade é o ponto de partida para projetos de compensadores.

### 2.1 Regiões de polos para sistemas contínuos

Na  $\mathcal{D}$ -estabilidade, a região referente à estabilidade em sistema contínuos e invariantes no tempo é o semi-plano esquerdo do plano complexo. Dado um ponto genérico no plano  $s$ , representado por:

$$s = x + jy \quad (2.1)$$

este estará na região estável somente se a parte real de tal ponto estiver à esquerda do eixo imaginário, ou em números:

$$\Re(s) < 0 \implies x < 0 \quad (2.2)$$

Assim, um sistema com  $n$  polos é dito estável se todos os seus polos estão localizados à esquerda do eixo imaginário. A partir deste conceito, é possível definir estabilidade relativa. Se um sistema é estável para um valor  $\sigma < 0$ , então aquele é dito estável relativo (ao valor de  $\sigma$ ).

A figura 1a mostra um esboço da região comentada. À medida que o valor de  $\sigma$  aumenta em valor absoluto, mais à esquerda a reta limitante se encontra e menor o plano estável relativo se torna. Além disso, as equações de tais retas podem ser generalizadas via:

$$x = -|\sigma| \quad (2.3)$$

Outros parâmetros de desempenho importantes para projetos de compensadores são o fator de amortecimento  $\zeta$  e a frequência natural não-amortecida  $\omega_n$ . São caracterizados pela resposta de sistemas de segunda ordem à função degrau (NISE, 2011)(OGATA, 2011) e representam as oscilações não-amortecidas do modelo físico. Dado um par de

polos conjugados via (2.1), é possível reescrevê-los em termos daqueles parâmetros:

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \quad (2.4)$$

com  $\sigma = -\zeta\omega_n$ . As regiões de  $\mathcal{D}$ -estabilidade referente a tais parâmetros são obtidos fixando um deles em (2.4) e variando o outro em um certo intervalo. Por esse motivo, (2.4) pode ser entendido como uma função de duas variáveis, dado como:

$$s(\zeta, \omega_n) = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \quad (2.5)$$

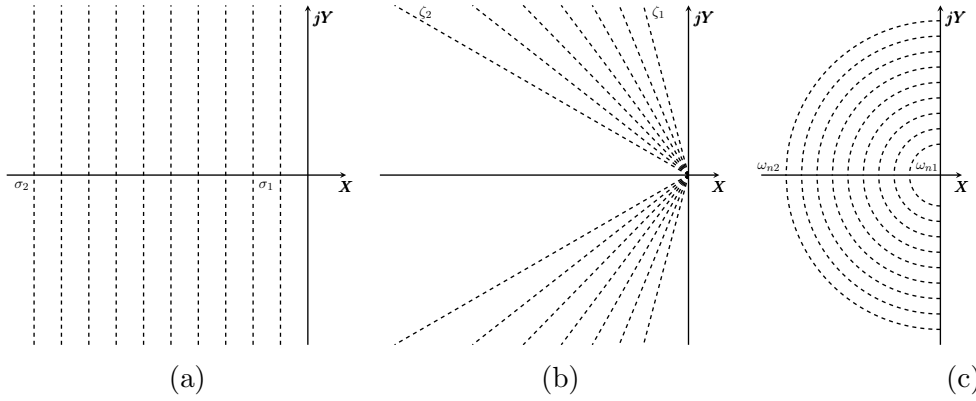


Figura 1 – Regiões de  $\mathcal{D}$ -estabilidade do plano  $s$ . Em (a) encontram-se retas verticais em vários valores de  $\sigma$ , sendo  $|\sigma_2| > |\sigma_1|$ . Em (b) encontram-se retas para vários valores de  $\zeta$ , sendo  $\zeta_2 > \zeta_1$ . Em (c) encontram-se circunferências de raios  $r = \omega_n$ , sendo  $\omega_{n2} > \omega_{n1}$ .

Com isso, é possível analisar as regiões geradas a partir de tais parâmetros. A região  $\zeta$ -constante é obtida fixando-se um valor para  $\zeta$  e variando-se o valor de  $\omega_n$ . As características obtidas são representadas na figura 1b. Os ângulos formados entre as retas e o eixo imaginário têm valores absolutos  $\beta = -\cot(\arccos(\zeta))$  e diminuem à medida que o valor de  $\zeta$  aumenta, tornando a região estreita.

Já as regiões  $\omega_n$ -constante possuem as características esboçadas na figura 1c. Os raios das semicircunferências formadas possuem valores iguais à  $\omega_n$  e aumentam ou diminuem à medida que se varia tal parâmetro. Conforme abordado em (CHILALI; GAHINET, 1996), a intersecção das regiões comentadas formam a Região  $\Omega$  de Desempenho garantido. Todos os polos dentro de tal região possuem um mínimo valor de  $\sigma$ ,  $\zeta$  e  $\omega_n$ .

## 2.2 Regiões de polos para sistemas discretos

As regiões de  $\mathcal{D}$ -estabilidade para sistemas discretos são obtidas seguindo os mesmos métodos abordados nos contínuos, com a diferença de serem descritos no plano  $z$ . A transformada de um ponto do plano  $s$  para  $z$  é dada por:

$$z = \exp(sT_s) \quad (2.6)$$

onde  $T_s$  é o período de amostragem, parâmetro importante para sistemas discretos (KUO, 1980). Substituindo (2.4) em (2.6), chega-se a seguinte relação:

$$z = \exp \left( -\zeta \omega_n T_s \pm j \omega_n T_s \sqrt{1 - \zeta^2} \right) \quad (2.7)$$

A  $\sigma$ -estabilidade nos contínuos foi encontrada verificando a parte real dos polos. Utilizando-se da mesma ideia, ao analisar apenas a parte real de (2.7) (igualando a parte imaginária igual a zero), chega-se na seguinte relação:

$$z = \exp (-|\sigma| T_s) \quad (2.8)$$

onde  $\sigma = -\zeta \omega_n$ . Tal função descreve uma circunferência com raio  $r = |\sigma|$  no plano complexo. Recordando (2.2) e (2.3), quando  $\sigma = 0$  em (2.8), a circunferência gerada possui raio unitário. Dessa maneira, o região estável nos sistemas discretos é o interior de uma circunferência unitária. A figura 2a mostra esboços para vários valores de  $\sigma$ .

Como realizado nos sistemas contínuos, (2.7) pode ser enxergada em função de  $\zeta$  e  $\omega_n$ :

$$z(\zeta, \omega_n) = \exp \left( -\zeta \omega_n T_s \pm j \omega_n T_s \sqrt{1 - \zeta^2} \right) \quad (2.9)$$

e a partir desta, é possível descrever as regiões geradas a partir de tais parâmetros. A região  $\zeta$ -constante possui o formato apresentado na figura 2a. Devido ao exponencial, as curvas geradas assemelham-se a cardioides, mas não o são, pois denominam-se espirais logarítmicas.

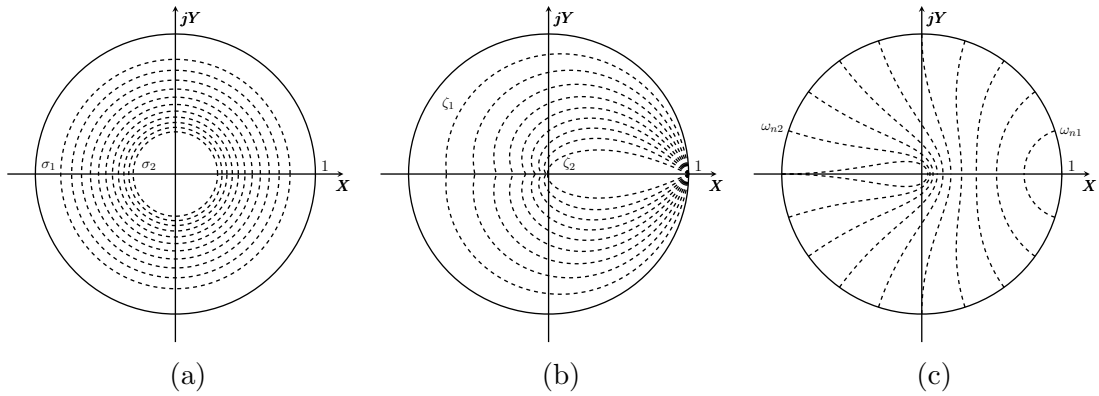


Figura 2 – Regiões de  $\mathcal{D}$ -estabilidade do plano  $z$ . Em (a) encontram-se circunferências com valores de raios crescentes, sendo  $|\sigma_2| > |\sigma_1|$ . Em (b) encontram regiões  $\zeta$ -constantes com áreas decrescentes em relação à  $\zeta$ , sendo  $\zeta_2 > \zeta_1$ . Em (c) encontram-se regiões semelhantes à cardioides que possuem áreas crescentes em relação à  $\omega_n$ , sendo  $\omega_2 > \omega_1$ .

Ambos os ramos começam a ser desenhadas a partir de  $(1, 0)$  (quando  $\omega_n = 0 = \omega_{nmin}$ ), e se deslocam da direita para a esquerda até cruzarem o eixo real primeira vez. À medida que o valor de  $\omega_n$  aumenta, mais voltas o contorno dá. E a cada  $n\pi$  voltas, o contorno cruza o eixo real pela  $n$ -ésima vez, conforme esboçado na figura 3. Como a espiral tende

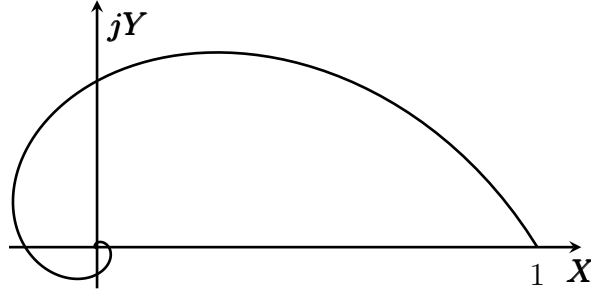


Figura 3 – Espiral logarítmica com 3 voltas gerada a partir de (2.9) com  $\zeta = 0.5$  constante.

para dentro da região limitada pela primeira volta, somente a primeira volta é considerada no plano  $z$ .

O valor de  $\omega_n$  no qual os ramos cruzam o eixo imaginário a cada  $n\pi$  voltas é encontrado quando o argumento de (2.9) é igual à  $\pi$  (meia volta da espiral, em radianos), isto é:

$$\begin{aligned} \arg z(\zeta, \omega_{nmax}) = \pi &\implies \omega_{nmax} T_s \sqrt{1 - \zeta^2} = \pi \implies \\ \omega_{nmax} &= \frac{\pi}{T_s \sqrt{1 - \zeta^2}} \end{aligned} \quad (2.10)$$

Assim, ambos os ramos cruzam o eixo imaginário pela primeira vez quando  $\omega_n = \omega_{nmax}$ , para o respectivo valor fixado de  $\zeta$ . Ainda, outra característica que pode ser citada é a influência do  $\zeta$  na região: quanto maior seu valor, menor a área da região  $\zeta$ -constante equivalente, assim como ocorre com seu dual nos contínuos.

Em relação às regiões  $\omega_n$ -constante, como o estudo de alocação de polos se restringe a sistemas de segunda ordem subamortecidos (NISE, 2011) (OGATA, 2011), os valores possíveis para a taxa de amortecimento estão no intervalo  $0 < \zeta < 1$ . Dito isso, os ramos esboçados na figura 2c começam a ser desenhadas a partir da circunferência unitária (quando  $\zeta = 0$ ) e vão em direção ao ponto  $(z(1, \omega_n), 0)$ . Tais pontos extremos possuem as seguintes relações:

$$z(\zeta_{min} = 0, \omega_n) = \exp(\pm \omega_n T_s) \quad (2.11a)$$

$$z(\zeta_{max} = 1, \omega_n) = \exp(-\omega_n T_s \pm j\omega_n) \quad (2.11b)$$

Com tais pontos extremos, é possível aproximar as regiões do plano  $z$  utilizando Desigualdades Matriciais Lineares (LMIs, em inglês). Esse estudo será abordado na subseção a seguir.

## 2.3 Aproximação das regiões do plano $z$ via LMIs

Em estudos anteriores, foram abordadas técnicas utilizando LMIs para mapear as regiões de  $\mathcal{D}$ -estabilidade no plano  $s$ . Tal feito foi realizado devido à convexidade de tais regiões, requisito para o uso de LMIs. Conforme visto na subseção 2.2, as regiões



$\zeta$ -constante e  $\omega_n$ -constante no plano  $z$  podem possuir características não-convexas, o que impossibilita o mapeamento exato via LMIs.

Estudos foram desenvolvidos para contornar a não-convexidade de algumas regiões do plano  $z$ , aproximando-os em regiões convexas. Em 2014, no artigo (ROSINOVÁ; HOLÍČ, 2014), a autora mapeou a região  $\zeta$ -constante utilizando a maior elipse ou circunferência inscrita possível. Mas foi em (ROSINOVÁ; HYPIUSOVÁ, 2019) que foi desenvolvido um algoritmo que traz várias aproximações utilizando elipses, para aproveitar da melhor forma a área daquela região.

Já em (WISNIEWSKI et al., 2017) foi abordada uma aproximação cônica, utilizando-se apenas de quatro pontos e, conseqüentemente, dois setores cônicos. Apesar de simples, a ideia poderia facilmente ser estendida para  $n$  pontos, o que foi feito em (WISNIEWSKI; MADDALENA; GODOY, 2019). Ao aumentar a área a cada iteração, o algoritmo verifica a solução proposta.

E finalmente, utilizando-se das ideias anteriores, (CHIQUETO, 2021) trouxe aproximações cônica, elíptica e poligonal da região  $\omega_n$ -constante.

## 2.4 Condição de Liapunov

$$\begin{bmatrix} -rP & * \\ PA + Z'B & -rP \end{bmatrix} \prec 0 \quad (2.12)$$

com  $r = \exp(-|\sigma|T_s)$ .

$$\begin{bmatrix} \sin(\theta)(AP + BZ + PA' + Z'B - 2\alpha P) & * \\ \cos(\theta)(PA' + Z'B' - AP - BZ) & \sin(\theta)(AP + BZ + PA' + Z'B - 2\alpha P) \end{bmatrix} \prec 0 \quad (2.13)$$

$$\begin{bmatrix} \sin(\theta)(2\alpha P - AP - BZ - PA' - Z'B') & * \\ \cos(\theta)(PA' + Z'B' - AP - BZ) & \sin(\theta)(2\alpha P - AP - BZ - PA' - Z'B') \end{bmatrix} \prec 0 \quad (2.14)$$

$$AP + BZ + Z'B' + PA' - 2\alpha P \succ 0 \quad (2.15)$$

$$N_y = \frac{\omega_s}{\omega_n} = \frac{2\pi}{\omega_n T_s} \quad (2.16)$$

$$a = \left(1 - \exp\left(\frac{-2\pi}{N_y}\right)\right) \quad (2.17)$$

$$b = a \sin\left(\frac{2\pi}{N_y}\right) \quad (2.18)$$

$$\begin{bmatrix} -P & * \\ -\frac{1}{a}P + \frac{1}{2}\left(\frac{1}{a} + \frac{1}{b}\right)AP + \frac{1}{2}\left(\frac{1}{a} - \frac{1}{b}\right)PA' & -P \end{bmatrix} \prec 0 \quad (2.19)$$

### 3 Algoritmo

O algoritmo apresentado neste trabalho é um compilado de algoritmos desenvolvidos anteriormente, utilizando-se das aproximações cônica, elíptica e poligonal das regiões de  $\mathcal{D}$ -estabilidade do plano  $z$ . O objetivo deste trabalho é desenvolver em *software* tais algoritmos e, ao informar parâmetros de projeto, determinar se é possível implementar um compensador que respeite os requisitos.

Para tal, o algoritmo pode ser dividido em três partes, uma para cada aproximação, sendo a aproximação desejada escolhida via chamada da função. O *software* utilizado foi o MATLAB, juntamente com o interpretador de LMIs YALMIP em conjunto com o solucionador numérico MOSEK.

Notação:  $loc(v_1, v_2)$  determina o ponto que a reta que passa por  $v_1$  e  $v_2$  cruza o eixo real.  $ang(v_1, v_2)$  refere-se ao ângulo entre aquela reta e o eixo real.

#### 3.1 Aproximação cônica

Para o mapeamento cônico das curvas  $\zeta$ -constante e  $\omega_n$ -constante, são utilizados os setores cônicos determinados via (2.14) e (2.13), e retas verticais como apresentado em 2.15.

Para a primeira curva, a ideia consiste em utilizar os pontos extremos calculados na seção 2.2, onde serão os centros dos setores cônicos. Os ângulos, medidos no sentido anti-horário, são determinados a partir de um terceiro ponto, conforme a figura 4a. A escolha do ponto  $Z$  é feita de maneira que a área do triângulo  $\widehat{Z_o Z Z_i}$  seja a maior possível. Um algoritmo linear foi usado para encontrar este ponto.

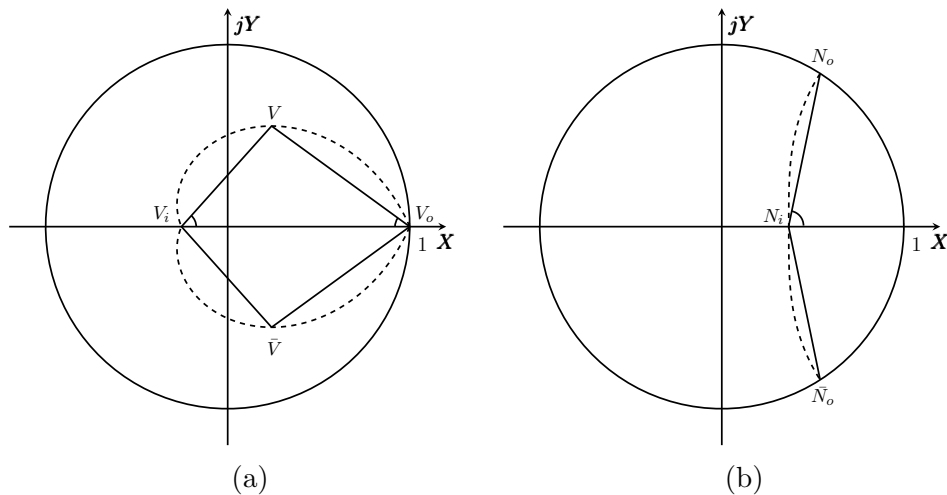


Figura 4

---

**Algoritmo 1** Aproximação cônica da taxa de amortecimento

---

**Entrada:**  $\sigma, \zeta, T_s$ **Saída:**  $K$ 

- 1:  $Z_o \leftarrow z(\zeta, \omega_{nmin})$
  - 2:  $Z_i \leftarrow z(\zeta, \omega_{nmax})$
  - 3:  $Z \leftarrow z(\zeta, \omega_n)$ , onde a área do triângulo formado é a maior possível
  - 4:  $F \leftarrow P \succ 0$
  - 5:  $F \leftarrow F \cap (2.12)$  com  $r = \exp(-|\sigma|T_s)$  ▷ Taxa de amortecimento
  - 6:  $F \leftarrow F \cap (2.13)$  com  $\alpha = Z_o$  e  $\theta = \text{ang}(Z, Z_i)$  ▷ Setor cônico esquerdo
  - 7:  $F \leftarrow F \cap (2.14)$  com  $\alpha = Z_i$  e  $\theta = \text{ang}(Z, Z_o)$  ▷ Setor cônico direito
  - 8:  $F \leftarrow F \cap (2.15)$  com  $\alpha = Z_i$  ▷ Reta vertical
  - 9: Verificar se o problema é factível
  - 10:  $K \leftarrow ZP^{-1}$
- 

Com tais informações determinadas, é possível aplicar o algoritmo 1. Um setor cônico voltado para a direita, com centro em  $Z_i$  e ângulo  $\theta_1$ , e outro voltado para a esquerda, com centro em  $Z_o$  e ângulo  $\theta_2$ , são aplicados. Além disso, para limitar a simetria do setor cônico com centro em  $Z_i$ , uma reta que passa por este ponto é aplicada.

A região de  $\mathcal{D}$ -estabilidade resultante é a intersecção de tais regiões. Ao ser unida com a restrição da taxa de decaimento, o setor cônico com centro em  $Z_o$  é limitado por esta região. Após resolvido o problema, o algoritmo determina a factibilidade da solução encontrada e retorna a matrix  $K$  de estabiliza o sistema com os parâmetros de projeto informados.

Em relação à largura de banda, a mesma ideia é aplicada (CHIQUETO, 2021). Contudo, neste caso, somente um setor cônico com centro em  $N_i$  limitado pela direita por uma reta que passa neste ponto são usados. Os pontos  $N_o$  e  $N_i$  são determinados via (2.11a) e (2.11b), respectivamente. Além disso, o ângulo  $\theta$  é determinado através de  $\text{ang}(N_o, N_i)$ .

---

**Algoritmo 2** Aproximação cônica da curva  $N_y$ 

---

**Entrada:**  $\sigma, \omega_n$ **Saída:**  $K$ 

- 1:  $N_o \leftarrow z(\zeta_{min}, \omega_n)$
  - 2:  $N_i \leftarrow z(\zeta_{max}, \omega_n)$
  - 3:  $F \leftarrow P \succ 0$
  - 4:  $F \leftarrow F \cap (2.12)$  com  $r = \exp(-|\sigma|T_s)$  ▷ Taxa de amortecimento
  - 5:  $F \leftarrow F \cap (2.14)$  com  $\alpha = N_i$  e  $\theta = \text{ang}(N_i, N_o)$  ▷ Setor cônico direito
  - 6:  $F \leftarrow F \cap (2.15)$  com  $\alpha = N_i$  ▷ Reta vertical
  - 7: Verificar se o problema é factível
  - 8:  $K \leftarrow ZP^{-1}$
- 

Com tais informações determinadas, é possível utilizar o algoritmo 2. Um detalhe que é facilmente observado é a rápida perda de convexidade da curva  $N_y$ . Logo, caso a constante  $N_y$  seja menor que 4.86 (CHIQUETO, 2021), o algoritmo retorna um alerta devido a falta de convexidade. Assim, para fins práticos, a pouca e a falta convexidade de tais curvas não foram tratadas.

### 3.2 Aproximação elíptica

Para a aproximação elíptica, apenas a região  $\omega_n$ -constante foi aproximada. A ideia consiste em encontrar a maior elipse inscrita, a fim de aproveitar melhor área. Para tal, é preciso verificar se o valor escolhido para  $\omega_n$  e  $T_s$  resultem em uma área convexa (CHIQUETO, 2021). Caso os parâmetros informados atendam às restrições, o algoritmo

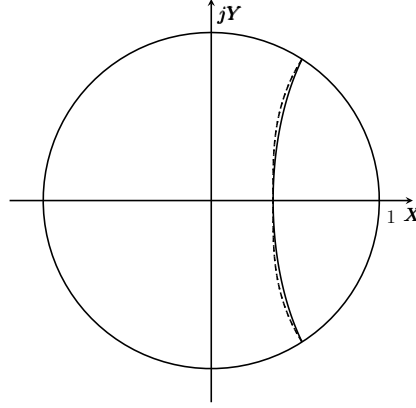


Figura 5

---

#### Algoritmo 3 Aproximação elíptica da curva $N_y$

---

**Entrada:**  $\sigma, T_s, N_y$

**Saída:**  $K$

- 1:  $F \leftarrow P \succ 0$
  - 2:  $F \leftarrow F \cap (2.12)$ , com  $r = \exp(-|\sigma|T_s)$  ▷ Taxa de amortecimento
  - 3:  $F \leftarrow (2.19)$ , com  $a = (2.17)$  e  $b = (2.18)$
  - 4: Verificar se o problema é factível
  - 5:  $K \leftarrow ZP^{-1}$
- 

### 3.3 Aproximação poligonal

A aproximação poligonal consiste na ideia de aproximar as regiões de interesse em um polígono com o maior número de lados possíveis. Para isto, o algoritmo irá partir de uma aproximação cônica simples. A partir daí, entre os dois pontos usados para definir o setor, um ponto intermediário é calculado e dois novos setores cônicos são definidos. Sob a ótica do número de lados, a cada iteração, um novo lado é acrescentado e, conseqüentemente, a área é incrementada. Em um número grande de iterações, a região aproximada tende a área total.

Para a região  $\zeta$ -constante, um setor cônico voltado para esquerda e centro em  $Z_o$  é usado como aproximação inicial (WISNIEWSKI; MADDALENA; GODOY, 2019). Contudo, devido à cúspide daquela, uma reta em  $Z_i$  é usada para eliminar tal convexidade. Dito isso, surge a necessidade de calcular o ponto  $\bar{z}_i$ , localizado entre os pontos máximo e mínimo, onde possui a mesma parte real que  $Z_i$ , conforme a figura:

---

**Algoritmo 4** Aproximação poligonal da região  $\zeta$ -constante

---

**Entrada:**  $\sigma, \zeta, T_s$ **Saída:**  $K$ 

```
1:  $l \leftarrow 0$ 
2:  $V_o \leftarrow z(\zeta, \omega_{nmin}, T_s)$ 
3:  $V_i \leftarrow z(\zeta, \omega_{nmax}, T_s)$ 
4:  $V \leftarrow z(\zeta, \omega_n, T_s)$ , tal que  $\Re(V) = V_i$ 
5:  $pontos1 \leftarrow [0 \ \omega_{ne}]$ 
6:  $pontos2 \leftarrow pontos1$ 
7:  $vec1 \leftarrow [V_o \ V]$ 
8:  $vec2 \leftarrow vec1$ 
9:  $F \leftarrow P \succ 0$ 
10:  $F \leftarrow (2.12)$  com  $r = \exp(-|\sigma|T_s)$  ▷ Taxa de amortecimento
11:  $F \leftarrow F \cap (2.13)$ , com  $\alpha = V_o$  e  $\theta = \text{ang}(V_o, V)$  ▷ Voltado para a esquerda
12:  $F \leftarrow F \cap (2.15)$ , com  $\alpha = V_i$  ▷ Reta vertical
13: Verificar se o problema é factível
14: enquanto Problema for infactível faça
15:   se  $l < \text{número de elementos em } vec1 - 1$  então
16:      $l \leftarrow l + 1$ 
17:   senão
18:      $l \leftarrow 1$ 
19:      $vec1 \leftarrow vec2$ 
20:      $pontos1 \leftarrow pontos2$ 
21:   fim se
22:    $F \leftarrow \emptyset$  ▷ Descarta as restrições anteriores
23:    $F \leftarrow P \succ 0$ 
24:    $F \leftarrow (2.12)$  com  $r = \exp(-|\sigma|T_s)$  ▷ Taxa de amortecimento
25:    $ponto_{new1} \leftarrow (pontos1(l) + pontos1(l+1))/2$ 
26:    $V_{new1} \leftarrow z(\zeta, ponto_{new1}, T_s)$ 
27:    $pontos2 \leftarrow [pontos2 \ ponto_{new1}]$ 
28:   Ordena de forma decrescente  $pontos2$ 
29:    $vec2 \leftarrow [vec2 \ V_{new1}]$ 
30:   Ordena de forma decrescente  $vec2$ 
31:    $F \leftarrow F \cap (2.15)$ , com  $\alpha = V_i$  ▷ Reta vertical
32:   para  $m = 1$  até número de elementos de  $vec1 - 1$  faça
33:      $u \leftarrow \text{loc}(vec2(m), vec2(m+1))$ 
34:     se  $u < 0$  então
35:        $F \leftarrow F \cap (2.14)$ , com  $\alpha = u$  e  $\theta = \text{ang}(vec2(m+1), u)$  ▷ Voltado para a direita
36:     senão
37:        $F \leftarrow F \cap (2.13)$ , com  $\alpha = u$  e  $\theta = \text{ang}(vec2(m+1), u)$  ▷ Voltado para a esquerda
38:     fim se
39:   fim para
40:   Verificar se o problema é factível
41: fim enquanto
42:  $K \leftarrow ZP^{-1}$ 
```

---

Para tal, utiliza-se o cálculo numérico para encontrar uma solução aproximada. Com tal ponto calculado e a não convexidade da cúspide tratada, é possível utilizar o algoritmo 4. Os vértices iniciais  $z_o$  e  $z_i$  do ramo são guardados em um vetor. Em paralelo a isso, os valores de  $\omega_n$  que geram tais valores em (2.9) também são armazenados. A ideia do algoritmo é, a cada iteração, calcular o ponto médio entre dois pontos consecutivos.

Contudo, antes do algoritmo voltar para o início dos vetores mencionados, é preciso que ele calcule o ponto médio entre todos os pontos do vetor atual. Para isso, cópias dos vetores de vértices e de pontos foram inicializados, a fim de controlarem tal fluxo. Assim,

---

**Algoritmo 5** Aproximação poligonal da região  $\omega_n$ -constante
 

---

**Entrada:**  $\sigma, \omega_n, T_s$ 
**Saída:**  $K$ 

```

1:  $l \leftarrow 0$ 
2:  $N_o \leftarrow z(\zeta_{min}, \omega_n, T_s)$ 
3:  $N_i \leftarrow z(\zeta_{max}, \omega_n, T_s)$ 
4:  $pontos3 \leftarrow [0 \ 1]$ 
5:  $pontos4 \leftarrow pontos3$ 
6:  $vec3 \leftarrow [N_i \ N_o]$ 
7:  $vec4 \leftarrow vec3$ 
8:  $F \leftarrow P \succ 0$ 
9:  $F \leftarrow (2.12)$  com  $r = \exp(-|\sigma|T_s)$  ▷ Taxa de amortecimento
10:  $F \leftarrow F \cap (2.14)$ , com  $\alpha = N_i$  e  $\theta = \text{ang}(N_o, N_i)$  ▷ Voltado para a direita
11:  $F \leftarrow F \cap (2.15)$ , com  $\alpha = N_i$  ▷ Reta vertical
12: Verificar se o problema é factível
13: enquanto Problema for infactível faça
14:   se  $l < \text{número de elementos em } vec3 - 1$  então
15:      $l \leftarrow l + 1$ 
16:   senão
17:      $l \leftarrow 1$ 
18:      $vec3 \leftarrow vec4$ 
19:      $pontos1 \leftarrow pontos2$ 
20:   fim se
21:    $F \leftarrow \emptyset$  ▷ Descarta as restrições anteriores
22:    $F \leftarrow P \succ 0$ 
23:    $F \leftarrow (2.12)$  com  $r = \exp(-|\sigma|T_s)$  ▷ Taxa de amortecimento
24:    $ponto_{new} \leftarrow (pontos3(l) + pontos3(l+1))/2$ 
25:    $V_{new2} \leftarrow z(ponto_{new2}, \omega_n, T_s)$ 
26:    $pontos4 \leftarrow [pontos4 \ ponto_{new2}]$ 
27:   Ordena de forma decrescente  $pontos4$ 
28:    $vec4 \leftarrow [vec4 \ V_{new2}]$ 
29:   Ordena de forma decrescente  $vec4$ 
30:    $F \leftarrow F \cap (2.15)$ , com  $\alpha = V_i$  ▷ Reta vertical
31:   para  $m = 1$  até número de elementos de  $vec3 - 1$  faça
32:      $u_2 \leftarrow \text{loc}(vec4(m), vec4(m+1))$ 
33:      $F \leftarrow F \cap (2.14)$ , com  $\alpha = u_2$  e  $\theta = \text{ang}(vec4(m), u_2)$ 
34:   fim para
35:   Verificar se o problema é factível
36: fim enquanto
37:  $K \leftarrow ZP^{-1}$ 

```

---

quando o algoritmo terminar de percorrer o "vetor anterior", tal conjunto é atualizado com os novos pontos e vértices calculados ao final deste processo.

Em relação ao cálculos dos pontos intermediários, a inclinação destes juntamente com o ponto anterior podem ser positiva ou negativa. O uso da função *loc* se torna enssencial, pois caso o ponto resultante for menor que zero, a reta que passa pelos pontos possui inclinação positiva e negativa, caso contrário (WISNIEWSKI; MADDALENA; GODOY, 2019). Assim, os setores cônicos gerados seguem a orientação de tal reta, com centro naquele ponto calculado.

Para a região  $\omega_n$ -constante, a ideia é similar. A aproxmição inicial emprega-se de apenas um setor cônico voltado para a direita com centro em  $n_i = z(\zeta_{max}, \omega_n, T_s)$ , com  $\zeta_{max} = 1$ . O ângulo  $\theta$  em (2.14) é definido a partir do ângulo entre a reta  $\overline{n_o n_i}$  e o eixo

real, onde  $n_o = z(\zeta_{min}, \omega_n, T_s)$ , com  $\zeta_{min} = 0$ .

Caso esta região não for factível, basta calcular o ponto intermediário entre  $n_o$  e  $n_i$  e definir dois novos setores cônicos, a fim de aumentar a área. Novamente, para o novo ponto calculado e o anteriormente, usa-se da função *loc* para determinar ponto que a reta que passa por aqueles pontos cruza o eixo real, onde será o centro do novo setor cônico. Já o ângulo seguirá o mesmo critério, sendo o valor entre aquela reta e o eixo real.

Assim, a cada iteração, o algoritmo adiciona dois novos setores cônicos e os intersecta com as regiões previamente definidos. Ao final de uma iteração, o algoritmo descarta tais regiões e começa a aumentar a área de  $\mathcal{D}$ -estabilidade a partir de  $n_i$  até  $n_o$ .

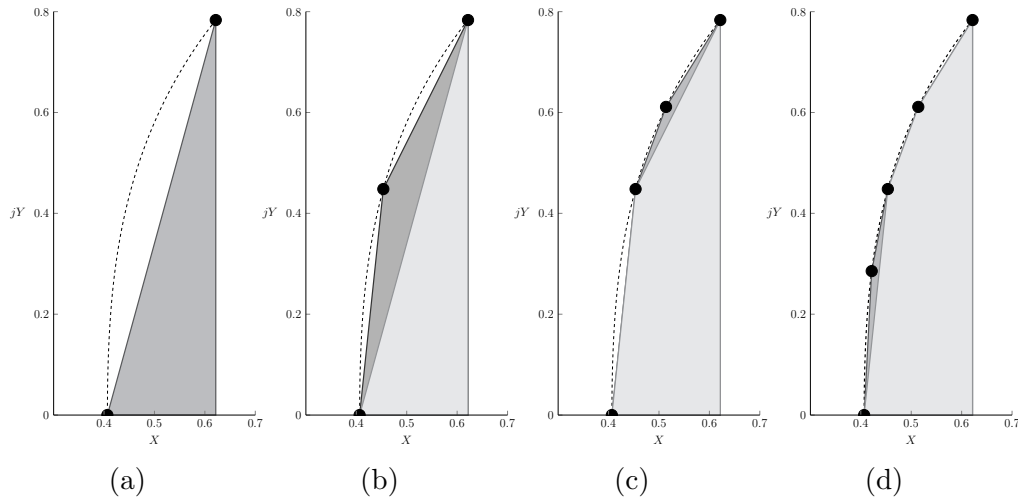


Figura 6



## 4 Testes e Simulações

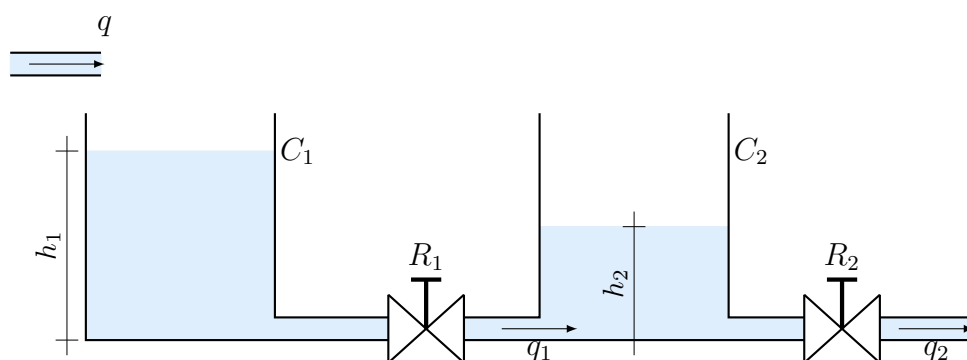


Figura 7 – Tanques comunicantes.



## 5 Conclusão



# Referências

- CHILALI, M.; GAHINET, P.  $H_\infty$  design with pole placement constraints: an lmi approach. *IEEE Transactions on Automatic Control*, v. 41, n. 3, p. 358–367, 1996.
- CHIQUETO, G. da S. Aproximações convexas via desigualdades matriciais lineares para o problema da largura de banda em sistemas em tempo discreto. 2021.
- KUO, B. *Digital Control Systems*. Holt, Rinehart and Winston, 1980. (HRW series in electrical and computer engineering). ISBN 9780030575686. Disponível em: <https://books.google.com.br/books?id=oNpSAAAAMAAJ>.
- NISE, N. *Control Systems Engineering, Sixth*. John Wiley & Sons, Incorporated, 2011. ISBN 9781118138168. Disponível em: <https://books.google.com.br/books?id=34zmCQAAQBAJ>.
- OGATA, K. *Engenharia de controle moderno*. Pearson Prentice Hall, 2011. ISBN 9788576058106. Disponível em: <https://books.google.com.br/books?id=iL3FYgEACAAJ>.
- ROSINOVÁ, D.; HOLÍČ, I. Lmi approximation of pole-region for discrete-time linear dynamic systems. In: *Proceedings of the 2014 15th International Carpathian Control Conference (ICCC)*. [S.l.: s.n.], 2014. p. 497–502.
- ROSINOVÁ, D.; HYPIUSOVÁ, M. Lmi pole regions for a robust discrete-time pole placement controller design. *Algorithms*, v. 12, n. 8, 2019. ISSN 1999-4893. Disponível em: <https://www.mdpi.com/1999-4893/12/8/167>.
- WISNIEWSKI, V.; MADDALENA, E.; GODOY, R. Discrete-time regional pole-placement using convex approximations: Theory and application to a boost converter. *Control Engineering Practice*, v. 91, p. 104102, 2019. ISSN 0967-0661. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0967066119301182>.
- WISNIEWSKI, V. L. et al. Regional pole placement for discrete-time systems using convex approximations. In: *2017 25th Mediterranean Conference on Control and Automation (MED)*. [S.l.: s.n.], 2017. p. 655–659.