

Pedro Henrique de Queiroz Ramos

Projeto de controladores Avanço/Atraso para sistemas lineares contínuos com abordagem via LMI

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação (FACOM) da Universidade Federal de Mato Grosso do Sul (UFMS) como requisito parcial para obtenção de grau bacharel em Engenharia de Computação.

Universidade Federal de Mato Grosso do Sul - UFMS

Faculdade de Computação - FACOM

Bacharelado

Orientador: Prof. Dr. Victor Leonardo Yoshimura

Brasil

2023

Dedico este trabalho a todos aqueles que usam o conhecimento para guiar o avanço da humanidade, desde os professores até os profissionais do dia a dia, à todos aqueles que conseguem olhar para si e enxergar as mudanças necessárias; à todos aqueles que combatem a miséria externa e interna com a esperança que habita em si.

Agradecimentos

Agradeço à minha família e amigos por todo o apoio, pai¹, mãe², irmãs³ e minha parceira⁴. Todos foram essenciais para toda e qualquer dificuldade encontrada durante a caminhada, ajudaram-me a suportar o medo e a tristeza, e reergueram-me colocando a esperança e o brilho no olhar para encontrar um futuro melhor para um mundo melhor.

Em especial, também agradeço à Universidade Federal de Mato Grosso do Sul (UFMS) pela oportunidade de participar e aprender durante estes cinco anos a vasta área da computação, assim fazendo jus à uma das melhores universidades da América Latina, segue rumo à formação de excelentes profissionais e pesquisadores a cada ano. Por fim, agradeço ao Professor Doutor Victor Leonardo Yoshimura, pela paciência e sabedoria transmitida ao longo de muitos semestres, além da orientação deste trabalho, com ensinamentos não só lógicos, mas a experiência transmitida e o olhar compassivo de quem quer ver o sucesso de seus alunos.

¹ Carlos Alberto Santos Ramos

² Vanderlene Pereira de Queiroz Ramos

³ Jéssica de Queiroz Ramos Polidoro
Kaline de Queiroz Ramos Silva

⁴ Bruna Parron Marodin

Resumo

Sistemas Lineares Invariantes no Tempo Contínuo (SLIT-C) aparecem com certa frequência no mundo real. Para a formalização deste, deve-se verificar um quadro de pré-requisitos, como a estabilidade, máximo sobressinal (*overshoot*), tempo de acomodação, etc, para então determinar sua operação.

Definiremos aqui algumas ferramentas para análise destes parâmetros através de métodos da teoria de controle moderno, como a estabilidade de Lyapunov, análise de espaço de estados, entre outras, que, para a classe de compensadores Avanço/Atraso, atenderá suficientemente aos critérios exigidos para um sistema em malha fechada com realimentação unitária.

A abordagem algorítmica usada será através de *Desigualdades Matriciais Lineares* (LMI - *Linear Matrix Inequality*), onde se manipula algumas notações algébricas a fim de obter uma desigualdade linear entre matrizes do espaço de estado, factíveis pelo solver Mosek[©] através do software Matlab[©].

Por fim, estimado o sistema (planta, sinal de entrada, referência, etc), o solver irá fornecer as possíveis matrizes de estado para o compensador, determinando se é do tipo Avanço ou Atraso, dado todas as restrições de projeto, onde é possível extrair a informação do espaço de estados do compensador necessário.

Palavras chave: LMI, estabilidade de Lyapunov, controle moderno, *lead-lag*.

Acrônimos

ARE *Algebraic Riccati Equation* (Equação Algébrica de Riccati).

BMI *Bilinear Matrix Inequality* (Desigualdade Matricial Bi-linear).

EDO Equação Diferencial Ordinária.

FT Função Transferência.

FTMF Função Transferência de Malha Fechada.

LMI *Linear Matrix Inequality* (Desigualdade Matricial Linear).

SCMF Sistema de Controle em Malha Fechada.

SPE Semi-plano Esquerdo.

SLIT-C Sistema Linear Invariante no tempo Contínuo.

Sumário

	Sumário	8
1	APRESENTAÇÃO DE CONCEITOS	3
1.1	Descrição de um sistema de controle ideal	3
1.2	O Espaço de Estados	3
1.3	Compensadores Avanço/Atraso	5
1.4	Modelagem em malha fechada	6
1.5	Crítério de Estabilidade de Lyapunov	8
1.6	Convexidade	8
1.6.1	Conjuntos Convexos	8
1.6.2	Funções Quase-Convexas	9
1.7	Linear Matrix Inequality(LMI)	9
1.8	Erro em regime permanente	11
1.8.1	Erro para entrada do tipo degrau	11
1.9	Definindo restrições do problema	11
2	PROJETO DO COMPENSADOR VIA LMI	13
2.1	Abordagem via Estabilidade de Lyapunov	13
2.2	Abordagem via Alocação de Polos de Chilali e Gahinet	14
2.3	Modelagem das LMI's de Chilali e Gahinet	15
2.3.1	Modelagem em γ	15
2.3.2	Modelagem em r	16
2.3.3	Modelagem em θ	16
2.4	Modelagem do sinal de entrada para tipo degrau	17
3	RESULTADOS PRÁTICOS	19
3.1	Softwares e Simulações	19
3.1.1	MATLAB	19
3.1.2	Mosek	19
3.1.3	YALMIP	20
3.2	Funções principais	20
3.2.1	<i>optimize</i>	20
3.2.2	<i>sdpvar</i>	20
3.2.3	<i>sdpsettings</i>	21
3.3	Código	21
3.4	Resultados práticos	22

3.4.1	Caso 1	22
3.4.2	Caso 2	23
3.5	Conclusões e Observações	25
	REFERÊNCIAS	27

1 Apresentação de conceitos

1.1 Descrição de um sistema de controle ideal

A estrutura de um sistema de controle é imposta com a seguinte configuração inicial descrito na [figura 1](#). Os blocos $G(s)$ e $C(s)$ determinam, respectivamente, a planta(ou processo) e o compensador(ou controlador) do sistema de controle. Há um sinal r para o modelo, que indica a referência a ser seguida pelo sistema. Chamamos também o sinal y como o sinal de saída do sistema.

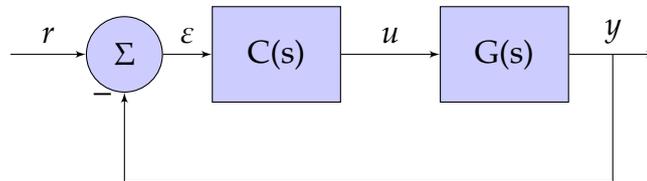


Figura 1 – Sistema de controle com realimentação unitária

Através do ponto de soma, podemos obter um novo sinal dado pela diferença entre a referência e a saída do sistema, expresso por

$$\varepsilon = r - y \quad (1.1)$$

indicando o sinal de erro, o que significa a diferença atual entre o que o sistema controlado está entregando em sua saída com o sinal que ele deveria em todo o instante de tempo. O sinal u é, ao mesmo tempo, a saída do compensador e a entrada para a planta.

Este esquema representa um simples sistema de controle ideal, pois indicamos uma leitura perfeita do sinal de saída através da realimentação unitária feita para o ponto de soma. Na prática, não existe realimentação unitária, isto é, não há um sensor perfeito que consiga ler a saída de um sistema para comparar com um sinal de referência e assim determinar o erro atual.

1.2 O Espaço de Estados

Cada bloco descrito na sessão anterior, pode representar um subsistema de controle, isto é, pode apresentar os mesmos sinais de entrada para os mesmos de saída. Isso também indica que cada bloco pode ser uma função transferência, no domínio s de *Laplace*.

Segundo Ogata (2010), o espaço de estados descreve o comportamento do sistema no domínio do tempo, relacionando variáveis de estado do sistema com suas derivadas e sinais de entrada/saída. Neste trabalho, descreveremos o espaço de estados para SLIT-C descrito por

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \quad (1.2a)$$

$$y = C\mathbf{x} + D\mathbf{u} \quad (1.2b)$$

onde a representação de cada variável é dada por:

$\mathbf{x}, \dot{\mathbf{x}} \in \mathbb{R}^{n \times 1} \rightarrow$ matriz coluna de variáveis de estado;

$\mathbf{u}, \mathbf{y} \in \mathbb{R} \rightarrow$ entrada e saída do sistema, respectivamente;

$A \in \mathbb{R}^{n \times n} \rightarrow$ matriz de estado;

$B \in \mathbb{R}^{n \times 1} \rightarrow$ matriz de entrada;

$C \in \mathbb{R}^{1 \times n} \rightarrow$ matriz de saída;

$D \in \mathbb{R} \rightarrow$ transmissão direta,

e n representa a ordem do sistema, isto é, a quantidade de variáveis de estado presente.

É possível realizar a conversão de espaços de estados para funções transferência e vice-versa. Considerando o espaço de estados 1.2, ao aplicar a transformada de Laplace com condições iniciais nulas, temos:

$$s\hat{\mathbf{x}}(s) = A\hat{\mathbf{x}}(s) + B\hat{\mathbf{u}}(s) \quad (1.3a)$$

$$\hat{y}(s) = C\hat{\mathbf{x}}(s) + D\hat{u}(s). \quad (1.3b)$$

Então, isolando $\hat{\mathbf{x}}(s)$ de 1.3a,

$$s\hat{\mathbf{x}}(s) - A\hat{\mathbf{x}}(s) = B\hat{\mathbf{u}}(s)$$

$$(sI - A)\hat{\mathbf{x}}(s) = B\hat{\mathbf{u}}(s)$$

$$\hat{\mathbf{x}}(s) = (sI - A)^{-1}B\hat{\mathbf{u}}(s)$$

e substituindo em 1.3b, temos:

$$\frac{\hat{y}(s)}{\hat{u}(s)} = C(sI - A)^{-1}B + D. \quad (1.4)$$

Nota-se que é possível descrever esta representação para funções transferência se, e somente se, $(sI - A)$ for invertível, ou seja, se esta for uma matriz não singular.

No contexto deste trabalho, é interessante descrever os sistemas da planta e compensador em espaço de estados separados, ou seja, cada um terá seus próprios sinais de entrada e saída. Então, é possível descrever os dois casos, respectivamente, por

$$\dot{\mathbf{x}}_g = A_g\mathbf{x}_g + B_g\mathbf{u} \quad (1.5a)$$

$$y = C_g\mathbf{x}_g + D_g\mathbf{u} \quad (1.5b)$$

$$\dot{\mathbf{x}}_c = A_c \mathbf{x}_c + B_c \varepsilon \quad (1.6a)$$

$$u = C_c \mathbf{x}_c + D_c \varepsilon \quad (1.6b)$$

pois a entrada do compensador é o sinal de erro e a saída do compensador é a entrada para a planta.

1.3 Compensadores Avanço/Atraso

Existem inúmeras classes de compensadores para sistemas de controle, como por exemplo do tipo Proporcional(P), Proporcional mais Integral(PI), Proporcional mais Derivativo(PD), Proporcional mais Integral mais Derivativo(PID), entre outros descritos em [Ogata \(2010\)](#) e [Geromel e Korogui \(2019\)](#). Neste trabalho, trataremos especificamente de projeto da classe de compensadores do tipo Avanço/Atraso.

A classe Avanço/Atraso é uma classe simples de compensadores, que pode ser modelada fisicamente de forma trivial para quaisquer tipos de sistemas (elétricos, mecânicos, térmicos, entre outros descritos por [Ogata \(2010\)](#)), em relação à outras classes de compensadores. Cada configuração de classe de compensadores apresenta as seguintes características.

Compensadores do tipo

Avanço São normalmente usados quando é necessário deixar o sistema mais rápido em relação à tempo de acomodação, tempo de pico, etc. Seus efeitos positivos tornam-se mais visíveis em regime transiente. Em contra partida, sua implementação traz uma vulnerabilidade à ruído no sistema devido ao efeito de polos mais à esquerda no [SPE](#), e, além disso, pode tornar o erro (ε) maior em regime permanente.

Atraso Seu efeito é inverso ao compensador do tipo Avanço. Sua implementação é necessária quando deseja-se impor o erro do sistema à zero, ou seja, fazer com que a saída do [SCMF](#) seja igual à referência. Além disso, como ponto negativo, este deixa o sistema mais lento, afetando negativamente o regime transiente e beneficiando o regime permanente.

Uma estrutura genérica pode ser imposta para o compensador Avanço/Atraso com a seguinte [FT](#):

$$C(s) = \frac{u(s)}{\varepsilon(s)} = \frac{\beta_1 s + \beta_0}{s + \alpha} \quad (1.7)$$

Se o polo for menor que o zero, será avanço, e, caso contrário, atraso. Isto é,

$$\alpha < \frac{\beta_0}{\beta_1} \rightarrow \text{Avanço} \quad (1.8a)$$

$$\alpha > \frac{\beta_0}{\beta_1} \rightarrow \text{Atraso} \quad (1.8b)$$

Ainda, é possível expressar esta classe de compensadores na forma de espaço de estados, similarmente ao que descreve a equação 1.6. Logo, é necessário voltar a FT de $C(s)$ para a forma de EDO.

$$C(s) = \frac{\beta_1 s + \beta_0}{s + \alpha} \Rightarrow \dot{u} + \alpha u = \beta_1 \dot{\varepsilon} + \beta_0 \varepsilon$$

$$\begin{aligned} \dot{u} - \beta_1 \dot{\varepsilon} &= -\alpha u + \beta_0 \varepsilon \\ \dot{x}_c &= -\alpha u + \beta_0 \varepsilon \end{aligned} \quad (1.9)$$

e, ainda,

$$\begin{aligned} \dot{x}_c &= \dot{u} - \beta_1 \dot{\varepsilon} \\ \int (\dot{x}_c) &= \int (\dot{u} - \beta_1 \dot{\varepsilon}) \\ x_c &= u - \beta_1 \varepsilon \\ u &= x_c + \beta_1 \varepsilon \end{aligned} \quad (1.10)$$

e então, substituindo 1.10 em 1.9, temos

$$\begin{aligned} \dot{x}_c &= -\alpha(x_c + \beta_1 \varepsilon) + \beta_0 \varepsilon \\ \dot{x}_c &= -\alpha x_c + (\beta_0 - \alpha \beta_1) \varepsilon \end{aligned} \quad (1.11)$$

Portanto, o espaço de estados para um compensador Avanço/Atraso será

$$\dot{x}_c = [-\alpha] x_c + [\beta_0 - \alpha \beta_1] \varepsilon \quad (1.12a)$$

$$u = [1] x_c + [\beta_1] \varepsilon \quad (1.12b)$$

onde a ordem do compensador é $n_c = 1$.

1.4 Modelagem em malha fechada

Imposta a estrutura do compensador a ser utilizado, iremos representar o SCMF como um todo. A motivação é para que se possa expressar um único espaço de estados, e assim determinar meios de encontrar um compensador ideal para uma

planta genérica. Recordando a relação do erro do SCMF dado em 1.1.

$$\begin{aligned}
\varepsilon &= r - y \\
&= r - (C_g \mathbf{x}_g + D_g u) \\
&= r - C_g \mathbf{x}_g - D_g (\mathbf{x}_c + \beta_1 \varepsilon) \\
&= r - C_g \mathbf{x}_g - D_g \mathbf{x}_c - D_g \beta_1 \varepsilon \\
\varepsilon + D_g \beta_1 \varepsilon &= r - C_g \mathbf{x}_g - D_g \mathbf{x}_c \\
(1 + D_g \beta_1) \varepsilon &= r - C_g \mathbf{x}_g - D_g \mathbf{x}_c \\
\varepsilon &= -(1 + D_g \beta_1)^{-1} C_g \mathbf{x}_g - (1 + D_g \beta_1)^{-1} D_g \mathbf{x}_c + (1 + D_g \beta_1)^{-1} r
\end{aligned} \tag{1.13}$$

onde $D_g \beta_1 \neq -1$. Substituindo 1.13 em 1.12a temos

$$\begin{aligned}
\dot{\mathbf{x}}_c &= -\alpha \mathbf{x}_c + (\beta_0 - \alpha \beta_1) (-(1 + D_g \beta_1)^{-1} C_g \mathbf{x}_g - (1 + D_g \beta_1)^{-1} D_g \mathbf{x}_c + (1 + D_g \beta_1)^{-1} r) \\
\dot{\mathbf{x}}_c &= (\alpha \beta_1 - \beta_0) (1 + D_g \beta_1)^{-1} C_g \mathbf{x}_g + (-\alpha - (\beta_0 - \alpha \beta_1) (1 + D_g \beta_1)^{-1} D_g) \mathbf{x}_c \\
&\quad + (\beta_0 - \alpha \beta_1) (1 + D_g \beta_1)^{-1} r
\end{aligned} \tag{1.14}$$

Para encontrar o espaço de estados da planta em relação ao compensador e a entrada como a referência r do sistema, substitui-se 1.12b em 1.5a e depois usa-se 1.13.

$$\begin{aligned}
\dot{\mathbf{x}}_g &= A_g \mathbf{x}_g + B_g (\mathbf{x}_c + \beta_1 \varepsilon) \\
&= A_g \mathbf{x}_g + B_g \mathbf{x}_c + B_g \beta_1 (-(1 + D_g \beta_1)^{-1} C_g \mathbf{x}_g - (1 + D_g \beta_1)^{-1} D_g \mathbf{x}_c + (1 + D_g \beta_1)^{-1} r) \\
\dot{\mathbf{x}}_g &= (A_g - B_g \beta_1 (1 + D_g \beta_1)^{-1} C_g) \mathbf{x}_g + (B_g - B_g \beta_1 (1 + D_g \beta_1)^{-1} D_g) \mathbf{x}_c \\
&\quad + B_g \beta_1 (1 + D_g \beta_1)^{-1} r
\end{aligned} \tag{1.15}$$

E, por fim, para determinar a saída do SCMF, usa-se 1.5b e posteriormente aplica-se 1.6b e 1.13.

$$\begin{aligned}
y &= C_g \mathbf{x}_g + D_g (\mathbf{x}_c + \beta_1 \varepsilon) \\
&= C_g \mathbf{x}_g + D_g \mathbf{x}_c + D_g \beta_1 (-(1 + D_g \beta_1)^{-1} C_g \mathbf{x}_g - (1 + D_g \beta_1)^{-1} D_g \mathbf{x}_c + (1 + D_g \beta_1)^{-1} r) \\
y &= (C_g - D_g \beta_1 (1 + D_g \beta_1)^{-1} C_g) \mathbf{x}_g + (D_g - D_g \beta_1 (1 + D_g \beta_1)^{-1} D_g) \mathbf{x}_c \\
&\quad + D_g \beta_1 (1 + D_g \beta_1)^{-1} r
\end{aligned} \tag{1.16}$$

Portanto, unindo 1.14, 1.15 e 1.16, o espaço de estados do SCMF é determinado por:

$$\dot{\mathbf{x}}_f = \begin{bmatrix} \dot{\mathbf{x}}_g \\ \dot{\mathbf{x}}_c \end{bmatrix} = \begin{bmatrix} A_g - B_g \beta_1 (1 + D_g \beta_1)^{-1} C_g & B_g - B_g \beta_1 (1 + D_g \beta_1)^{-1} D_g \\ (\alpha \beta_1 - \beta_0) (1 + D_g \beta_1)^{-1} C_g & -\alpha - (\beta_0 - \alpha \beta_1) (1 + D_g \beta_1)^{-1} D_g \end{bmatrix} \begin{bmatrix} \mathbf{x}_g \\ \mathbf{x}_c \end{bmatrix} + \begin{bmatrix} B_g \beta_1 (1 + D_g \beta_1)^{-1} \\ (\beta_0 - \alpha \beta_1) (1 + D_g \beta_1)^{-1} r \end{bmatrix} \tag{1.17a}$$

$$\begin{aligned}
y &= \begin{bmatrix} C_g - D_g \beta_1 (1 + D_g \beta_1)^{-1} C_g & D_g - D_g \beta_1 (1 + D_g \beta_1)^{-1} D_g \end{bmatrix} \begin{bmatrix} \mathbf{x}_g \\ \mathbf{x}_c \end{bmatrix} \\
&\quad + \begin{bmatrix} D_g \beta_1 (1 + D_g \beta_1)^{-1} \end{bmatrix} r
\end{aligned} \tag{1.17b}$$

1.5 Critério de Estabilidade de Lyapunov

Segundo [Geromel e Korogui \(2019\)](#), em 1892, Alexander Mikhailovich Lyapunov publicou uma tese a respeito da *Teoria da Estabilidade Global de Sistemas Dinâmicos*, que se tratava da definição de como os sistemas de controle podem ser estáveis/instáveis, sendo de natureza linear/não-linear, contínuo/discreto, variante/invariante no tempo. A ideia geral é de que exista o chamado *ponto de equilíbrio* do sistema dinâmico, dado por $x_e \in \mathbb{R}^n$ onde, a condição inicial do sistema é

$$x(0) = x_e \Rightarrow x(t) = x_e \quad \forall t \geq 0, \quad (1.18)$$

o que implica que para qualquer instante de tempo, o sistema tenderá ao ponto de equilíbrio. Definimos uma função $v(x)$ chamada *função de distância* que mede a distância de qualquer ponto $x \in \mathbb{R}^n$ no espaço de estado até o ponto de equilíbrio x_e , se para toda condição inicial $x(0) = x_0$ a função $v(x(t))$ diminuir ao longo do tempo, e consequentemente tender a zero para $t \rightarrow \infty$. Portanto, é necessário definir $v(x)$ como uma função decrescente para $t \rightarrow \infty$.

Portanto, o modelo da estabilidade de Lyapunov abordado para [SLIT-C's](#) é do tipo

$$\dot{x} = Ax \quad (1.19)$$

exatamente igual equação [1.2a](#). Impondo a função distância $v(x) = xPx'$, sendo $P \in \mathbb{R}^{n \times n} \mid P' = P > 0$, isto é, P é uma matriz quadrada, simétrica e definida positiva. Desta forma, obtém-se

$$\begin{aligned} \dot{v} &= \dot{x}'Px + x'P\dot{x} \\ &= x'(A'P + PA)x \end{aligned} \quad (1.20)$$

Do *cálculo* tem-se que, para uma função crescente, sua derivada é positiva, mas se uma função é decrescente, então sua derivada é negativa. Logo, como $v(x)$ implica em $P > 0$, então para \dot{v} temos que

$$A'P + PA < 0 \quad (1.21)$$

Este problema é tratado como uma [LMI](#), e, para que [1.21](#) seja negativa-definida, todos os seus autovalores devem ser negativos para satisfazer a condição.

1.6 Convexidade

1.6.1 Conjuntos Convexos

De acordo com [Boyd e Vandenberghe \(2004\)](#), dado um conjunto \mathcal{C} no espaço euclidiano, este é dito **convexo** se, para todo ponto x_1 e x_2 pertencente ao conjunto,

a norma euclidiana entre os dois pontos está inteiramente contida no conjunto. Isto significa que

$$\mathcal{C} \subset \mathbb{R}^n \mid \forall x_1, x_2 \in \mathcal{C} \text{ e } x_1 \neq x_2, \exists v \in [0, 1],$$

então

$$vx_1 + (1 - v)x_2 \in \mathcal{C}.$$

Além disso, se $v \in [0, 1]$ (intervalo fechado), diz-se que é um conjunto convexo não-estrito, mas se $v \in (0, 1)$ (intervalo aberto) então tem-se um conjunto estritamente convexo.

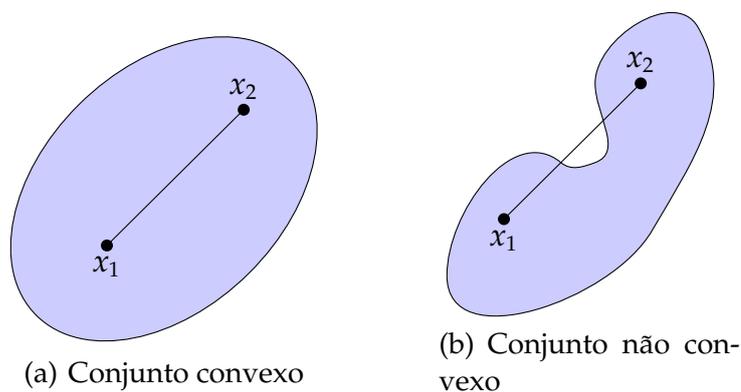


Figura 2 – Representação gráfica de conjuntos convexas.

1.6.2 Funções Quase-Convexas

Dizemos que uma função é **quase-convexa** se, e somente se, dado dois pontos $x_1, x_2 \in f(x) \mid x_1 \neq x_2$, se há um ponto intermediário que existe entre o máximo e o mínimo de x_1 e x_2 (ver figura 3). Se $\exists v \in (0, 1)$, a função estritamente quase-convexa é dada por:

$$\min\{f(x_1), f(x_2)\} < f(vx_1 + (1 - v)x_2) < \max\{f(x_1), f(x_2)\}$$

1.7 Linear Matrix Inequality(LMI)

LMI's ou *Desigualdades Matriciais Lineares*, de acordo com [Boyd et al. \(1994\)](#), são problemas convexas de otimização dentro do campo da matemática, que também é usado amplamente na teoria de controle. A forma mais comum de uma **LMI** é dada por

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0, \quad (1.22)$$

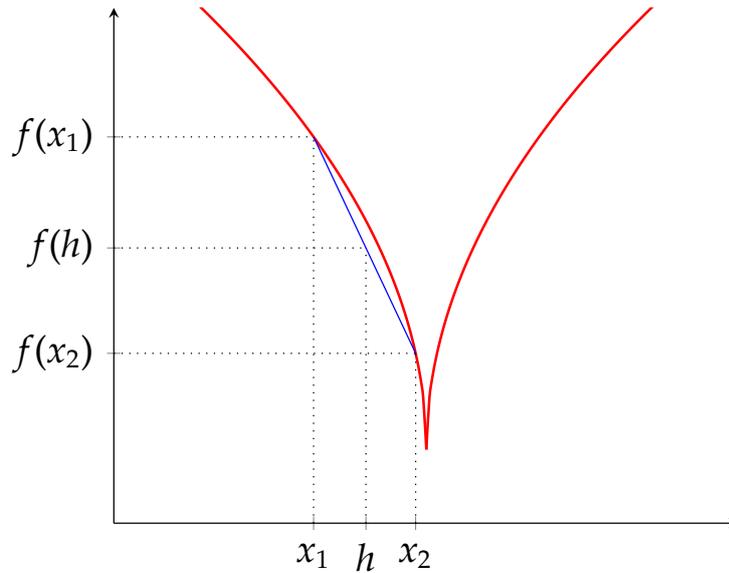


Figura 3 – Representação de um função quase-convexa, onde $h = vx_1 + (1 - v)x_2$.

no qual $F_i = F'_i \in \mathbb{R}^{n \times n}$ é fornecido, $x_i \in \mathbb{R}^{1 \times n}$ é a variável, e $i \in \mathbb{N}^*$. A desigualdade estrita imposta em 1.22 indica que $F(x)$ é definida positiva e que é equivalente a um conjunto de n desigualdades polinomiais em x , isto é, todos os autovalores e o cômputo dos menores principais líderes de $F(x)$ são positivos. O mesmo vale para a definida negativa.

A LMI em 1.22 é uma restrição convexa em x , onde x pertence a um conjunto convexo. Em particular, alguns tipos de problemas como, desigualdades lineares, desigualdades quadráticas, e algumas outras restrições que advém da área da teoria de controle, como a desigualdade de Lyapunov ou a Equação Algébrica de Riccati (ARE), podem ser convertidas em uma LMI usando outras ferramentas da álgebra (Complemento de Schur, por exemplo).

Além disso, múltiplas LMI's do tipo 1.22 também podem ser convertidas usando uma única representação. Supondo que exista $F^{(1)}(x) > 0, \dots, F^{(p)}(x) > 0$, onde $p > 1$, então uma única LMI pode ser expressa por

$$\begin{bmatrix} F^{(1)} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & F^{(2)} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & F^{(p)} \end{bmatrix} > 0 \quad (1.23a)$$

$$\text{diag}(F^{(1)}(x), \dots, F^{(p)}(x)) > 0 \quad (1.23b)$$

1.8 Erro em regime permanente

O erro de um certo **SLIT-C** pode ser definido como a diferença entre o sinal de saída e o sinal de referência do sistema, isto é a própria equação modelada em 1.1. Existem vários tipos de sinais de entrada para sistemas de controle que podem ser abordados, como entrada tipo rampa (relacionada à velocidade) e entrada do tipo aceleração (relacionado à um sinal de ordem quadrática)(Ogata (2010)). Para todos os tipos de sinais de entrada, deve-se abordar como será calculado o erro em regime permanente através do *Teorema do Valor Final*, pois como cada entrada tem uma característica peculiar, a resposta para $t \rightarrow \infty$ será diferente. Abordaremos neste trabalho somente entrada para tipo degrau.

1.8.1 Erro para entrada do tipo degrau

Considerando então uma referência do tipo degrau, isto é,

$$r(t) = \tilde{1}(t) = \begin{cases} 1 & \text{para } t \geq 0 \\ 0 & \text{para } t < 0 \end{cases} \quad (1.24)$$

e sua *Transformada de Laplace* é

$$\hat{r}(s) = \frac{1}{s} \quad (1.25)$$

podemos calcular o **erro estático de posição** utilizando do *Teorema do Valor Final* para o **SLIT-C** em malha fechada. Portanto, tomamos a função transferência do erro sobre a referência do sistema em malha fechada e sem compensador da figura 1 ($C(s) = 1$):

$$\frac{\hat{\varepsilon}(s)}{\hat{r}(s)} = \frac{1}{1 + G(s)}$$

$$\varepsilon(\infty) = \lim_{s \rightarrow 0} s \frac{\hat{\varepsilon}(s)}{\hat{r}(s)} .$$

Como $\hat{r}(s) = \frac{1}{s}$, o erro estático de posição pode ser calculado por

$$\varepsilon(\infty) = \lim_{s \rightarrow 0} \frac{1}{1 + G(s)} = \frac{1}{1 + G(0)} . \quad (1.26)$$

1.9 Definindo restrições do problema

Vamos iniciar uma nova modelagem à respeito do **SLIT-C** obtido em 1.17. Antes de tudo, será necessário definir alguns parâmetros do modelo e restrições para o estudo, a fim de tornar o problema factível pelo software a ser usado.

Vamos impor a restrição de que a planta do **SCMF não terá transmissão direta**, ou seja, a parcela de entrada do sistema não afetará a parcela de saída. Isso implica na equação 1.5b que

$$D_g = 0$$

Além disso, fixamos a variável β_1 do compensador Avanço/Atraso para evitar o produto de múltiplas variáveis e o problema se expandir para uma possível BMI.

$$\beta_1 = 1$$

Portanto, a equação 1.17 pode ser simplificada em

$$\dot{\mathbf{x}}_f = \begin{bmatrix} A_g - B_g C_g & B_g \\ (\alpha - \beta_0) C_g & -\alpha \end{bmatrix} \begin{bmatrix} \mathbf{x}_g \\ \mathbf{x}_c \end{bmatrix} + \begin{bmatrix} B_g \\ \beta_0 - \alpha \end{bmatrix} r \quad (1.27a)$$

$$y_f = \begin{bmatrix} C_g & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_g \\ \mathbf{x}_c \end{bmatrix} \quad (1.27b)$$

e torna-se notável a simplificação do espaço de estados para o SCMF.

2 Projeto do compensador via LMI

2.1 Abordagem via Estabilidade de Lyapunov

Apresentado o modelo da *Estabilidade de Lyapunov* descrito na sessão 1.5, desejamos projetar compensadores Avanço/Atraso para uma dada planta, com a restrição de que não possuirá transmissão direta ($D_g = 0$) e fixamos o parâmetro $\beta_1 = 1$ para o compensador. Então, tomamos o [SCMF 1.27](#) e substituímos na desigualdade de Lyapunov 1.21.

$$A'_f P + P A_f < 0$$

onde

$$P = \begin{bmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & p \end{bmatrix} > 0 \quad | \quad P_1 = P'_1 \in \mathbb{R}^{n_g \times n_g}, p \in \mathbb{R} \text{ e } P \in \mathbb{R}^{(n_g+1) \times (n_g+1)} \quad , \quad (2.1)$$

então

$$\begin{aligned} & \begin{bmatrix} A'_g - C'_g B'_g & (\alpha - \beta_0) C'_g \\ B'_g & -\alpha \end{bmatrix} \begin{bmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & p \end{bmatrix} + \begin{bmatrix} P_1 & \mathbf{0} \\ \mathbf{0} & p \end{bmatrix} \begin{bmatrix} A_g - B_g C_g & B_g \\ (\alpha - \beta_0) C_g & -\alpha \end{bmatrix} < 0 \\ & \begin{bmatrix} A'_g P_1 - C'_g B'_g P_1 & (\alpha p - \beta_0 p) C'_g \\ B'_g P_1 & -\alpha p \end{bmatrix} + \begin{bmatrix} P_1 A_g - P_1 B_g C_g & P_1 B_g \\ (\alpha p - \beta_0 p) C_g & -\alpha p \end{bmatrix} < 0 \\ & \begin{bmatrix} \mathbf{Herm}\{P_1 A_g - P_1 B_g C_g\} & (\alpha p - \beta_0 p) C'_g + P_1 B_g \\ \star & -2\alpha p \end{bmatrix} < 0 \end{aligned} \quad (2.2)$$

onde *Herm* é o operador Hermitiano ($\mathbf{Herm}(x) = x + x'$) e \star indica a parte superior direita transposta da matriz. Ainda se faz necessário eliminar o produto de incógnitas escalares, a fim de tornar o problema convexo. Portanto, fazemos

$$\bar{\alpha} = \alpha p \quad (2.3a)$$

$$\bar{\beta} = \beta_0 p \quad (2.3b)$$

$$\delta = \bar{\alpha} - \bar{\beta} \quad (2.3c)$$

e por fim, substituindo 2.3 em 2.2, a LMI modelada para resolver o problema da estabilidade de Lyapunov fica

$$\begin{bmatrix} \mathbf{Herm}\{P_1 A_g - P_1 B_g C_g\} & \delta C'_g + P_1 B_g \\ \star & -2\bar{\alpha} \end{bmatrix} < 0 \quad (2.4)$$

onde o objetivo torna-se maximizar o traço de P , e, se existir uma solução para o problema, então encontra-se um compensador Avanço/Atraso que é capaz de tornar o [SCMF](#) globalmente estável.

2.2 Abordagem via Alocação de Polos de Chilali e Gahinet

De acordo com o artigo de Chilali e Gahinet (1996), o tempo de acomodação, máximo sobressinal (*overshoot*) e largura de banda de um SCMF pode ser determinado através de LMI's baseadas na Teoria de Lyapunov, onde analisamos o comportamento dos polos dominantes do SCMF, isto é, em qual região se encontram os polos dominantes de malha fechada, e assim, determinamos um compensador para ajustar a melhor posição para as características de resposta transiente desejadas.

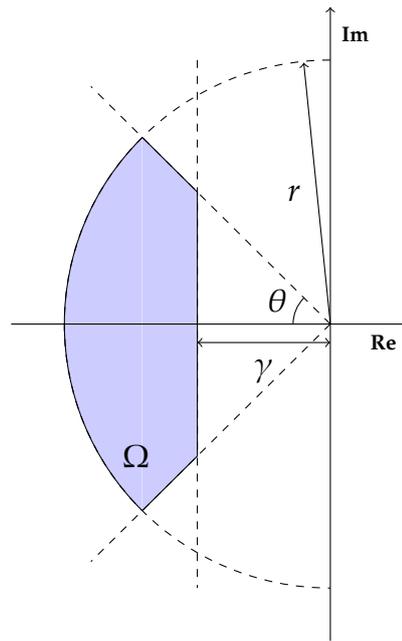


Figura 4 – Região Ω de desempenho

A região $\Omega(\gamma, r, \theta)$ do SPE é a região que oferece desempenho garantido para o SCMF, isto é, se os polos de malha fechada estiverem dentro da região, então o sistema atende aos requisitos de tempo de acomodação, máximo sobressinal, largura de banda, etc. Esta região é seccionada por γ , r e θ , e um número complexo arbitrário $c = a \pm jb \mid c \in \Omega(\gamma, r, \theta)$ se e somente se

$$a < -\gamma < 0, |a \pm jb| < r, \tan(\theta) a < -|b| \quad .$$

O parâmetro γ determina o tempo de acomodação do sistema, indicado pelas seguintes equações.

$$t_s = \frac{3}{\gamma} \quad (\text{crit. 5\%}) \quad (2.5a)$$

$$t_s = \frac{4}{\gamma} \quad (\text{crit. 2\%}) \quad (2.5b)$$

A equação 2.5a indica o tempo de acomodação com um critério de 5%, isto é, o sistema se acomoda no tempo t_s e permanece dentro de uma faixa de erro de até 5% da referência até o infinito. O mesmo é válido para 2.5b, onde a faixa de erro tolerável é de 2%.

O parâmetro θ indica o ângulo limitante da região Ω de desempenho. Ele está relacionado ao máximo sobressinal (*overshoot*) do [SCMF](#)

$$\theta = \arctan\left(\frac{\omega_d}{\gamma}\right) = \arctan\left(\frac{\omega_n \sqrt{1 - \zeta^2}}{\gamma}\right) \quad (2.6)$$

no qual ζ é o fator de amortecimento e ω_n é a frequência natural não-amortecida. Também podemos expressar o *overshoot* diretamente através de

$$M_p = e^{-\frac{\gamma\pi}{\omega_d}} \quad (2.7)$$

A largura de banda r delimita a região Ω à esquerda. Isso demonstra que há um limite de região os polos dominados devem se situar no [SPE](#), pois quanto mais à esquerda, mais rápido o sistema se comporta, porém, o sistema é mais suscetível ao efeito de ruído.

2.3 Modelagem das LMI's de Chilali e Gahinet

Definida a região $\Omega(\gamma, r, \theta)$ de desempenho garantido, através de [Chilali e Gahinet \(1996\)](#) podemos inferir as seguintes [LMI's](#) que relacionam os três parâmetros da região Ω , em suas respectivas formas primal e dual:

$$AP + PA' + 2\gamma P < 0 \equiv A'P + PA + 2\gamma P < 0 \quad (2.8a)$$

$$\begin{bmatrix} -rP & AP \\ PA' & -rP \end{bmatrix} < 0 \equiv \begin{bmatrix} -rP & A'P \\ PA & -rP \end{bmatrix} < 0 \quad (2.8b)$$

$$\begin{aligned} & \begin{bmatrix} \text{sen}(\theta)(AP + PA') & \text{cos}(\theta)(AP - PA') \\ \text{cos}(\theta)(PA' - AP) & \text{sen}(\theta)(AP + PA') \end{bmatrix} < 0 \equiv \\ & \begin{bmatrix} \text{sen}(\theta)(A'P + PA) & \text{cos}(\theta)(A'P - PA) \\ \text{cos}(\theta)(PA - A'P) & \text{sen}(\theta)(A'P + PA) \end{bmatrix} < 0 \end{aligned} \quad (2.8c)$$

2.3.1 Modelagem em γ

Tomando [2.8a](#), fazemos a modelagem da [LMI](#) dual com a estrutura de planta e compensador propostos. Portanto, tomando o [SLIT-C 1.27](#) usando a definição de [2.1](#) e a parte já desenvolvida em [2.4](#), temos

$$\begin{aligned} & A'_f P + PA_f + 2\gamma P < 0 \\ & \begin{bmatrix} \text{Herm}\{A'_g P_1 - C'_g B'_g P_1\} & \delta C'_g + P_1 B_g \\ \star & -2\bar{\alpha} \end{bmatrix} + \begin{bmatrix} 2\gamma P_1 & \mathbf{0} \\ \mathbf{0} & 2\gamma p \end{bmatrix} < 0 \\ & \begin{bmatrix} \text{Herm}\{A'_g P_1 - C'_g B'_g P_1\} + 2\gamma P_1 & \delta C'_g + P_1 B_g \\ \star & 2(\gamma p - \bar{\alpha}) \end{bmatrix} < 0 \end{aligned} \quad (2.9)$$

Como γ e p são escalares e P_1 é uma matriz, observemos que o problema torna-se **quase-convexo** devido ao produto de γp e γP_1 . Isso significa que, dado dois pontos do conjunto imagem de 2.9, a norma euclidiana entre esses dois pontos (como descrito na sessão 1.6.2) está contida inteiramente no conjunto, e há um mínimo ou máximo local que é também o global.

2.3.2 Modelagem em r

Tomando a LMI 2.8b, façamos uma reestruturação também para o problema ser restringido ao nosso estudo de caso para compensadores avanço-atraso. Logo, tomamos a LMI dual

$$\begin{bmatrix} -rP & A'_f P \\ P A_f & -rP \end{bmatrix} < 0$$

$$\left[\begin{array}{cc|cc} -rP_1 & \mathbf{0} & & \star \\ \mathbf{0} & -rp & & \\ \hline P_1 A_g - P_1 B_g C_g & P_1 B_g & -rP_1 & \mathbf{0} \\ p(\alpha - \beta_0) C_g & -\alpha p & \mathbf{0} & -rp \end{array} \right] < 0$$

e então, fazendo $\bar{r} = rp$ e usando 2.3, chegamos à

$$\left[\begin{array}{cc|cc} -rP_1 & \mathbf{0} & & \star \\ \mathbf{0} & -\bar{r} & & \\ \hline P_1 A_g - P_1 B_g C_g & P_1 B_g & -rP_1 & \mathbf{0} \\ \delta C_g & -\bar{\alpha} & \mathbf{0} & -\bar{r} \end{array} \right] < 0 \quad (2.10)$$

2.3.3 Modelagem em θ

Por fim, tomamos a LMI dual 2.8c e aplicamos o mesmo tipo de estrutura das outras já definidas.

$$\begin{bmatrix} \sin(\theta)(A'_f P + P A_f) & \cos(\theta)(A'_f P - P A_f) \\ \cos(\theta)(P A_f - A'_f P) & \sin(\theta)(A'_f P + P A_f) \end{bmatrix} < 0$$

Então, a fim de eliminar a restrição de seno e cosseno da variável θ , fazamos $\theta_s = \sin(\theta)$ e $\theta_c = \cos(\theta)$,

$$\begin{aligned} & \begin{bmatrix} \theta_s(A'_f P + PA_f) & \theta_c(A'_f P - PA_f) \\ \theta_c(PA_f - A'_f P) & \theta_s(A'_f P + PA_f) \end{bmatrix} < 0 \\ & \left[\begin{array}{cc|cc} \theta_s \mathbf{Herm}\{A'_g P_1 - C'_g B'_g P_1\} & \theta_s \delta C'_g + \theta_s P_1 B_g & \theta_c \overline{\mathbf{Herm}}\{A'_g P_1 - C'_g B'_g P_1\} & \theta_c \delta C'_g - \theta_c P_1 B_g \\ \star & -2\bar{\alpha}\theta_s & -\star & \mathbf{0} \\ \hline & \star & \theta_s \mathbf{Herm}\{A'_g P_1 - C'_g B'_g P_1\} & \theta_s \delta C'_g + \theta_s P_1 B_g \\ & & \star & -2\bar{\alpha}\theta_s \end{array} \right] < 0 \end{aligned} \quad (2.11)$$

onde $\overline{\mathbf{Herm}}\{x\} = x - x'$ é o operador Anti-Hermitiano.

2.4 Modelagem do sinal de entrada para tipo degrau

Para descrevermos a saída do **SLIT-C** com a inserção do compensador, é necessário fechar a malha do sistema e obter sua função transferência da saída y em relação à referência r . Porém, ainda que seja possível encontrar um compensador capaz de atender aos requisitos de overshoot e tempo de acomodação, não se sabe se o **SLIT-C** tem erro estático em regime permanente nulo, isto porque a planta é genérica e não há garantia de polos na origem.

Como é desejável que o erro do sistema seja zero, discutiremos então uma forma alternativa de eliminar o erro em regime permanente de forma a alterar o sinal de entrada. Portanto, se 1.1 é válida, e desejamos que $\varepsilon(\infty) = 0 \Rightarrow r = y = \frac{1}{s}$. Podemos considerar o sistema já compensado e realimentado em malha fechada, ou seja, podemos incluir $C(s)$ na malha fechada. Então, obtemos a **FTMF** e isolamos o sinal de referência:

$$\begin{aligned} \frac{\hat{y}(s)}{\hat{r}(s)} &= \frac{C(s)G(s)}{1 + C(s)G(s)} \\ \hat{r}(s) &= \left(\frac{C(s)G(s)}{1 + C(s)G(s)} \right)^{-1} \hat{y}(s) \end{aligned}$$

e então aplicamos o *Teorema do Valor Final* para a referência:

$$\begin{aligned} \lim_{t \rightarrow \infty} r(t) &= \lim_{s \rightarrow 0} s(C(sI - A)^{-1}B + D)^{-1} \frac{1}{s} \\ &= \lim_{s \rightarrow 0} (C(sI - A)^{-1}B + D)^{-1} \\ &= -\frac{1}{CA^{-1}B + D} \end{aligned} \quad (2.12)$$

Logo, para que a saída do sistema em $t \rightarrow \infty$ seja unitária, fazemos:

$$r(t) = -\frac{1}{CA^{-1}B + D} \tilde{\Gamma}(t) \quad (2.13)$$

3 Resultados Práticos

3.1 Softwares e Simulações

Concluído a parte de modelagem do problema proposto, aqui exibiremos os resultados práticos e simulações de problemas abordados via [LMI](#). Para isto, foram necessárias algumas ferramentas de simulação como [©MATLAB](#), e também o solver de [LMI](#) [©Mosek](#), ambos com licença acadêmica disponibilizada.

3.1.1 MATLAB

[©MATLAB](#) é um software de cálculo numérico de alto desempenho que permite o usuário simular diversos problemas ou cálculos experimentais teóricos e práticos. O software é mais voltado à área de engenharia, mas em suas últimas atualizações, também trouxe suporte à área de inteligência artificial e telecomunicações. Usaremos o [©MATLAB](#) como ferramenta para descrever espaços de estado e/ou funções transferências de sistemas de controle, e também para realizar as simulações no tempo contínuo e plotagem de gráficos.



Figura 5 – Logo MATLAB

3.1.2 Mosek

[©Mosek](#) é uma biblioteca de funções de cálculo numérico avançado que é capaz de solucionar desde problemas lineares até problemas exponenciais, isto é, problemas assintoticamente lineares até NP-Hard através de heurísticas. As funções são disponibilizadas em vários tipos de linguagens para fazer interface como por exemplo C/C++, R,

Python, .NET, Java e MATLAB. As funções disponibilizadas podem ser utilizadas para LP's (Linear Programming), QP's (Quadratic Programming), SOCP's (Second Order Cone Programming), SDP's (Semi-Definite Programming) e MIP's (Mixed Integer Programming).



Figura 6 – Logo Mosek

3.1.3 YALMIP

©YALMIP é *parser* capaz de traduzir a entrada de um programa em um certo código para outros solvers numéricos solucionarem. No nosso contexto, o ©YALMIP faz integração dos problemas escritos em ©MATLAB para o solver ©Mosek, isto é, o solver ©Mosek é invocado pelo ©YALMIP de acordo com cada problema específico definido pela modelagem na linguagem MATLAB. ©YALMIP tem direitos de licença detidos por Johan Löfberg, mas é distribuído gratuitamente.

3.2 Funções principais

3.2.1 *optimize*

A função *optimize* disponibilizada pelo ©YALMIP tem a seguinte sintaxe:

```
[resultados] = optimize([restrições],[objetivo], [opções])
```

O vetor de **restrições** são todas as imposições, limites numéricos ou desigualdades que podem ser impostas para o problema. **objetivo** é a variável a ser otimizada, e, por definição do ©YALMIP, esta variável é *minimizada*. O vetor de **opções** define as opções de solvers a serem utilizados pelo ©YALMIP. Todo o resultado de processamento e/ou informações de execução são armazenadas em **resultados**.

3.2.2 *sdpvar*

A função *sdpvar* cria variáveis de decisão simbólicas através do ©YALMIP, ou seja, essas são variáveis do problema, que podem ser escalares ou matrizes.

```
x = sdpvar(n) % Cria uma matriz de variáveis simétrica n x n
x = sdpvar(n,m) % Cria uma matriz de variáveis simétrica n x m
```

```
x = sdpvar(n,m,'full') % Cria uma matriz completa de variáveis n x n
sdpvar x y % Cria variáveis escalares x e y
```

3.2.3 sdpsettings

A função *sdpsettings* define um vetor de parâmetros a serem utilizados pelo ©YALMIP e definem o solver a ser utilizado.

```
[opções] = sdpsettings(['campo'],[valor],['campo'],[valor],...)
```

3.3 Código

O código implementado usando as ferramentas citadas anteriormente pode ser descrito através do seguinte pseudo-código.

Algoritmo 1: LMI de minimização em γ

Entrada: A_g, B_g, C_g, r, θ
Saída: α, β_0
Restrições: $\det(A_g) \neq 0, \lambda(A_g) < 0, r > 0, 0 < \theta < \frac{\pi}{2}$

- 1 $n_g \leftarrow \text{length}(A_g);$ // n_g é a ordem da planta
- 2 **sdpvar** $\bar{\alpha}, \bar{\beta}, \bar{r}, p, \delta, \gamma;$ // Declaração de variáveis escalares
- 3 $P1 \leftarrow \text{sdpvar}(n_g, n_g, 'symmetric');$ // Declaração de matrizes de variáveis
- 4 $P \leftarrow \begin{bmatrix} P1 & \mathbf{0} \\ \mathbf{0} & p \end{bmatrix};$
- 5 $LMI_{constraints} \leftarrow [LMI_\gamma \leq 0, LMI_r \leq 0, LMI_\theta \leq 0];$
- 6 $options \leftarrow \text{sdpsettings}('solver', 'mosek');$
- 7 $results \leftarrow \text{optimize}([P \geq 0, LMI_{constraints}, \gamma \geq 0], gamma, options);$
- 8 **if** $results.problem = 0$ **then**
- 9 $\alpha \leftarrow \frac{\bar{\alpha}}{p};$
- 10 $\beta_0 \leftarrow \frac{\bar{\beta}}{p};$
- 11 **return** α, β_0
- 12 **else**
- 13 **erro**
- 14 **end**

Notamos que na linha 5 está a descrição das três LMI's modeladas anteriormente, 2.9, 2.10 e 2.11, e ainda que a minimização ocorre em γ , isto é, o solver tentará solucionar o problema para o menor valor possível de γ , o que poderá ocasionar em um tempo de acomodação maior.

3.4 Resultados práticos

Nesta sessão, temos como objetivo mostrar os resultados práticos das simulações feitas através do algoritmo modelado. Tomamos então diferentes entradas para **SLIT-Cs** em malha aberta genéricos.

3.4.1 Caso 1

Considere a planta a seguir.

$$\begin{aligned}\dot{x}_g &= \begin{bmatrix} -3 & -6 & 0 \\ 0 & -2 & 2 \\ 0 & 0 & -1 \end{bmatrix} x_g + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} -1 & 0 & 0 \end{bmatrix} x_g\end{aligned}$$

Limitamos os valores de $r = 5$ e $\theta = 1.1868$ (68°). Através do método de minimização em γ visto pelo algoritmo 1, o solver retorna o compensador tipo atraso:

$$C(s) = \frac{s + 0.5563}{s + 1.3586} \Rightarrow \alpha = 1.3586 \text{ e } \beta_0 = 0.5563$$

$$P = \begin{bmatrix} 11.3987 & -2.3416 & 10.7616 & 0 \\ -2.3416 & 71.0155 & -3.9327 & 0 \\ 10.7616 & -3.9327 & 68.3002 & 0 \\ 0 & 0 & 0 & 182.9032 \end{bmatrix}$$

$$\lambda(P) = [9.3899, 66.3559, 74.9687, 182.9032].$$

O espaço de estado em malha fechada com compensador é

$$\begin{aligned}\dot{x}_f &= \begin{bmatrix} -3 & -6 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ 1 & 0 & -1 & 1 \\ -0.8022 & 0 & 0 & -1.3586 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -0.8022 \end{bmatrix} r \\ y &= \begin{bmatrix} -1 & 0 & 0 & 0 \end{bmatrix} x\end{aligned}$$

e os autovalores da matriz A_f de malha fechada são

$$\lambda(A_f) = [-0.6295, -1.0647 \pm j1.9969, -4.5997].$$

Pode-se observar que estes autovalores estão todos com a parte real limitada por $r = 5$ e a parte imaginária limitada por $\theta = 68^\circ$ conforme a região Ω de desempenho garantido estabelece. Simularemos a saída do sistema para a resposta à entrada degrau. Como o compensador não estabelece correção para o erro estático de posição, então

é necessário fazer com que o sinal de referência seja uma entrada degrau modificada através de 2.13. Logo,

$$r(t) = -\frac{1}{C_f A_f^{-1} B_f} \tilde{1}(t) = 2.221 \cdot \tilde{1}(t)$$

Observando então a figura 7, é notável que o sistema sem compensação não apresenta

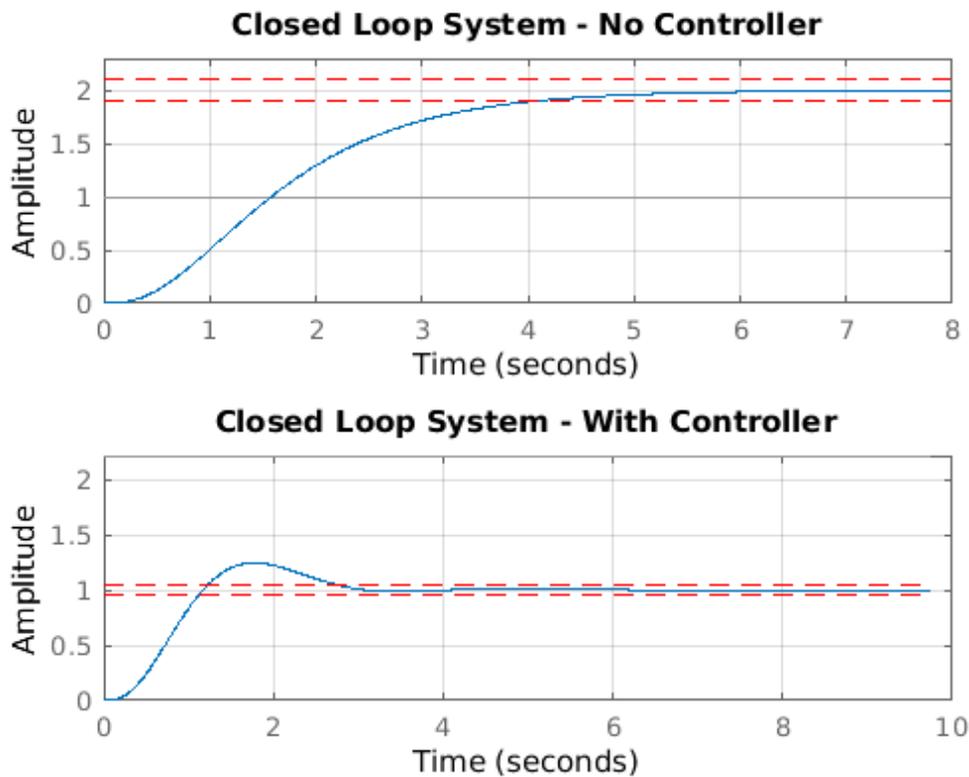


Figura 7 – Simulação do sistema 1 em malha fechada com e sem compensação.

overshoot e seu tempo de acomodação é na faixa de $t_s = 4$ segundos (crit. 5%). Após determinar o compensador, modelando o novo sistema em malha fechada, encontra-se um tempo de acomodação menor $t_s = 2.77$ segundos em relação ao sistema sem compensador, porém, apresenta-se um overshoot em torno de $M_p = 25\%$. O erro estático de posição está eliminado devido à entrada ser o inverso do valor de $y(\infty)$.

3.4.2 Caso 2

Considere a planta a seguir.

$$\dot{x}_g = \begin{bmatrix} -106 & -612 & -1208 & -800 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x_g + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = [0 \ 1 \ 11 \ 10] x_g$$

Para este sistema, restringimos os limitantes da região Ω de desempenho para $r = 100$ e $\theta = 0.2618$ (15°), e aplicamos as entradas ao algoritmo 1. O solver retorna a região factível com as variáveis otimizadas para um compensador tipo atraso.

$$C(s) = \frac{s + 0.6575}{s + 1.521} \Rightarrow \alpha = 1.521 \text{ e } \beta_0 = 0.6575$$

$$P = \begin{bmatrix} 181.365 & 1202 & 2618.6 & 1868 & 0 \\ 1202 & 19201 & 59747 & 52522 & 0 \\ 2618.6 & 59747 & 203910 & 189240 & 0 \\ 1868 & 52522 & 189240 & 183220 & 0 \\ 0 & 0 & 0 & 0 & 384010 \end{bmatrix}$$

$$\lambda(P) = [18.7278, 412.2714, 6355.6, 399740, 384010].$$

O espaço de estado em malha fechada com compensador é

$$\dot{x}_f = \begin{bmatrix} -106 & -613 & -1219 & -810 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0.8635 & 9.4983 & 8.6348 & -1.521 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -0.8635 \end{bmatrix} r$$

$$y = [0 \ 1 \ 11 \ 10 \ 0] x$$

e os autovalores da matriz A_f de malha fechada são

$$\lambda(A_f) = [-1.477 \pm j0.2087, -2.2883 \pm j0.5123, -99.9904].$$

Podemos notar que dois pares de polos conjugados dominam, e estes estão mais perto da origem, restando o polo real dominado bem próximo ao limite definido por r .

Para a correção do erro estático de posição, tomamos 2.13 como entrada e obtemos:

$$r(t) = -\frac{1}{C_f A_f^{-1} B_f} \tilde{1}(t) = 186.0657 \cdot \tilde{1}(t).$$

É perceptível que a amplitude da entrada para este caso é mais elevada que o primeiro caso. O overshoot apresentado é em torno de $M_p = 50\%$, que é bem significativo, e já o tempo de acomodação torna-se maior também se considerarmos o fato de que o overshoot do sistema sem compensador atinge aproximadamente a taxa de critério de erro do sistema (5%). Para este sistema, o maior benefício trazido pelo compensador é a correção do erro estático de posição, que anteriormente atingia cerca de $\varepsilon(\infty) = 98.75\%$.

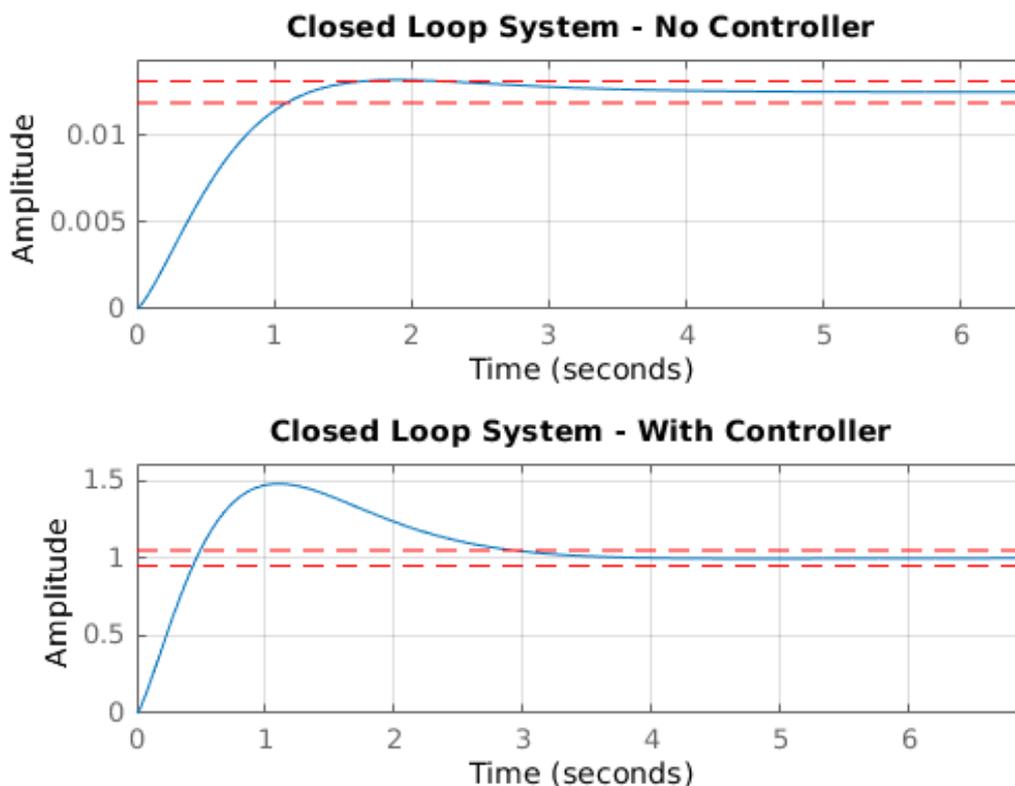


Figura 8 – Simulação do sistema 2 em malha fechada com e sem compensação.

3.5 Conclusões e Observações

Foram determinados métodos de compensação de sistemas de controle dinâmicos através de estratégias matemáticas usando [LMI's](#) para a classe de compensadores Avanço/Atraso que propiciam ajuste da saída de um sistema em relação, principalmente, ao overshoot e tempo de acomodação. É claro que existem outros métodos para aplicabilidade desses tipos de compensadores, como o método por projeto via Lugar das Raízes e pelo método via diagramas de Bode, que são métodos clássicos amplamente utilizados dentro do campo de controle de sistemas lineares.

A formulação deste novo método de projeto propõe em melhorar algumas características em relação ao tempo empregado no desenvolvimento do compensador, que por sua vez em grande parte é projetado analiticamente, e também implica em explorar novas variáveis de otimização de um sistema onde não é possível abordar de forma trivial nos métodos clássicos citados. Diferentemente dos métodos clássicos, esta abordagem via [LMI's](#) de [Chilali e Gahinet \(1996\)](#) propõe uma nova restrição de largura de banda do sistema, o que implica diretamente na atenuação de ruídos no sistema de controle. Restringindo-se o raio r de uma semi-circunferência de polos de malha fechada, reduzimos a largura de banda, e o sistema de controle atua como um filtro.

Além disso, a adoção da modelagem via **LMI** implica em algoritmos de tempo de execução polinomial (P), o que poderia ser diferente se fosse utilizado outras ferramentas próximas, como **BMI's**, que são problemas da classe NP e solucionáveis apenas por heurísticas. É notável que a construção de algoritmos de projeto de compensadores para sistemas genéricos admite grande facilidade de manutenção e aplicabilidade para ampla faixa de sistemas de alta ordem, isto é, com grande quantidade de variáveis de estado.

Referências

- BOYD, S. et al. *Linear matrix inequalities in system and control theory*. [S.l.]: SIAM, 1994.
- BOYD, S.; VANDENBERGHE, L. *Convex Optimization*. [S.l.]: Cambridge University Press, 2004.
- CHILALI, M.; GAHINET, P. H_{∞} design with pole placement constraints: an lmi approach. *IEEE Transactions on Automatic Control*, v. 41, n. 3, p. 358–367, 1996.
- GEROMEL, J. C.; KOROGUI, R. H. *Controle linear de sistemas dinâmicos 2ª edição: teoria, ensaios práticos e exercícios*. [S.l.]: Editora Blucher, 2019.
- OGATA, K. *Modern control engineering fifth edition*. [S.l.: s.n.], 2010.