

Atividade Orientada a Ensino: Estudo da estabilidade de sistemas via LMI's

Pedro Henrique Acioly Amaral¹, Prof. Dr. Victor Leonardo Yoshimura²

¹Discente do Curso de Engenharia da Computação, UFMS
pedro-h-a-amaral@ufms.br ²Faculdade de Computação, Universidade Federal de Mato Grosso
do Sul (UFMS) Campo Grande – MS – Brasil
victor.yoshimura@ufms.br

Abstract. *This report addresses the design of controllers for linear time-invariant continuous (LTI-C) systems using the state-feedback methodology. The stability of the closed-loop system is analyzed from the perspective of Lyapunov's Theorem, which leads to a stability condition in the form of a matrix inequality. The inherent non-linearity of this inequality is overcome through a change of variables, transforming the controller design problem into a convex optimization problem involving Linear Matrix Inequalities (LMIs). This approach is then applied to a practical system of two communicating tanks, where a state-feedback controller is synthesized to regulate the tank level and to track a reference signal. Simulations of the controlled system, carried out in the Xcos/Scilab environment, validate the effectiveness of the designed controller.*

Resumo. *Este relatório aborda o projeto de controladores para sistemas lineares invariantes no tempo contínuos (SLIT-C) utilizando a metodologia de realimentação de estados. A estabilidade do sistema em malha fechada é analisada sob a ótica do Teorema de Lyapunov, que conduz a uma condição de estabilidade na forma de uma desigualdade matricial. A não linearidade intrínseca desta desigualdade é contornada através de uma mudança de variáveis, transformando o problema de projeto do controlador em um problema de otimização convexa envolvendo Desigualdades Matriciais Lineares (LMI). Esta abordagem é então aplicada a um sistema prático de dois tanques comunicantes, onde um controlador de realimentação de estados é sintetizado para regular o nível do tanque e para seguir um sinal de referência. As simulações do sistema controlado, realizadas no ambiente Xcos/Scilab, validam a eficácia do controlador projetado.*

Controle por Realimentação de Estados, Estabilidade de Lyapunov, Desigualdades Matriciais Lineares (LMI), Sistemas de Tanques, Controle de Processos.

1. Introdução

Considere um sistema linear, invariante no tempo e contínuo (SLIT-C) com uma única entrada e uma única saída (SISO), representado no espaço de estados por [Ogata 2010]:

$$\dot{x} = Ax + Bu \quad (1)$$

$$y = Cx + Du \quad (2)$$

onde $x \in R^n$ é o vetor de estados, $u \in R$ é a entrada de controle, $y \in R$ é a saída do sistema, e A, B, C, D são matrizes de dimensões apropriadas. O problema fundamental

do controle é escolher a lei de controle u de forma a garantir a estabilidade do sistema e, possivelmente, atender a outros critérios de desempenho.

2. Projeto do Controlador por Realimentação de Estados

Utilizaremos a técnica de realimentação de estados, que pressupõe que todos os estados $x_1, x_2, \dots, x_n$ são mensuráveis. O objetivo é projetar um vetor de ganhos $K = [k_1, k_2, \dots, k_n]$ para compor o sinal de controle da seguinte forma:

$$u = -Kx = -(k_1x_1 + k_2x_2 + \dots + k_nx_n)$$

A substituição desta lei de controle na equação de estados do sistema resulta no sistema em malha fechada:

$$\begin{aligned}\dot{x} &= Ax + B(-Kx) \\ \dot{x} &= (A - BK)x\end{aligned}$$

Definindo a matriz de malha fechada como $A_f = (A - BK)$, o sistema se torna $\dot{x} = A_fx$. A estabilidade deste sistema autônomo depende dos autovalores da matriz A_f . Um procedimento para determinar K é a Fórmula de Ackerman, que permite alocar os polos de malha fechada em posições desejadas. Outra abordagem, mais robusta e explorada neste trabalho, baseia-se no Teorema de Lyapunov.

2.1. Estabilidade de Lyapunov e LMIs

O Teorema de Lyapunov estabelece que o sistema em malha fechada $\dot{x} = A_fx$ é assintoticamente estável se, e somente se, existe uma matriz P simétrica e definida positiva ($P = P^T > 0$) tal que $A_f^T P + PA_f$ seja definida negativa ($A_f^T P + PA_f < 0$).

Substituindo $A_f = A - BK$ na desigualdade, obtemos:

$$\begin{aligned}(A - BK)^T P + P(A - BK) &< 0 \\ (A^T - K^T B^T)P + PA - PBK &< 0 \\ A^T P + PA - K^T B^T P - PBK &< 0\end{aligned}$$

Esta desigualdade é uma Desigualdade Matricial Bilinear (BMI), pois envolve produtos das variáveis de decisão P e K , o que a torna um problema não convexo e de difícil solução no entanto conversão desta condição em uma forma tratável numericamente pode ser feita via Desigualdades Matriciais Lineares (LMIs) [Scherer and Weiland 2000].

Para linearizar o problema, realizamos uma transformação de congruência. Pré e pós-multiplicando a desigualdade por P^{-1} (que existe e é definida positiva, pois $P > 0$), a desigualdade não se altera. Definimos uma nova variável $Q = P^{-1}$ e uma variável auxiliar $W = KQ$. Isso nos leva a:

$$QA^T + AQ - W^T B^T - BW < 0$$

O problema de projeto do controlador se resume a encontrar matrizes Q e W que solucionem o seguinte conjunto de Desigualdades Matriciais Lineares (LMIs) :

1. $Q > 0$
2. $AQ + QA^T - BW - W^T B^T < 0$

Este é um problema de otimização convexa que pode ser eficientemente resolvido por solvers numéricos. Uma vez encontradas as soluções para Q e W , o ganho do controlador K é recuperado através da relação $K = WQ^{-1}$.

3. Aplicação ao Sistema de Tanques Comunicantes

O método LMI será aplicado para projetar um controlador para o sistema de tanques comunicantes ilustrado na Figura 1.

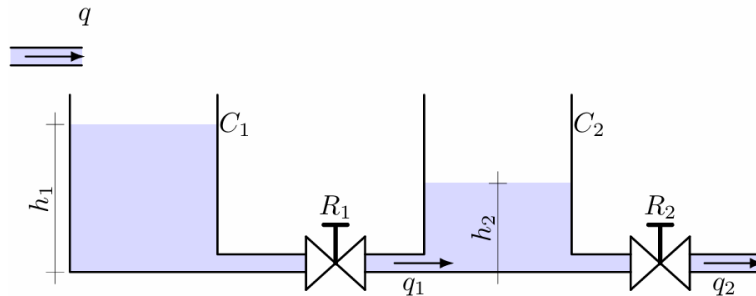


Figura 1. Sistema de tanques comunicantes.

Os parâmetros do sistema são: $R_1 = 1 \text{ s/m}^2$, $R_2 = 0.4 \text{ s/m}^2$, $C_1 = 2 \text{ m}^2$ e $C_2 = 5 \text{ m}^2$. As matrizes do sistema em espaço de estados são:

$$A = \begin{bmatrix} -0.5 & 0.5 \\ 0.2 & -0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

A saída de interesse é a altura do segundo tanque, portanto:

$$C = [0 \quad 1], \quad D = [0]$$

3.1. Cálculo do Ganho K via LMI

Um script em Python, utilizando CVXPY com o solver MOSEK, foi implementado para resolver as LMIs (Listagem 30).

```
1 import numpy as np
2 import cvxpy as cp
3
4 def lyapunov_solver():
5     # Valores fixos fornecidos
6     r1 = 1.0
7     r2 = 0.4
8     c1 = 2.0
9     c2 = 5.0
10    # Matriz A fixa
11    A = np.array([
12        [-1 / (r1 * c1), 1 / (r1 * c1)],
13        [1 / (r1 * c2), -(r1 + r2) / (r1 * r2 * c2)]
14    ])
15    # Matriz B fixa
16    B = np.array([[1 / c1], [0]])
17    # Matriz C fixa
18    C = np.array([0, 1])
19    # Matriz D fixa
20    D = np.array([0])
21    # Variáveis matriciais
22    P = cp.Variable((2, 2), symmetric=True)
23    Q = cp.Variable((2, 2), symmetric=True)
24    W = cp.Variable((1, 2)) # 1x2
25    # Variável auxiliar Z para a expressão da inequação de Lyapunov
26    Z = cp.Variable((2, 2), symmetric=True)
27    # Expressão da inequação de Lyapunov modificada
28    lyap_expr = Q @ A.T + A @ Q + W.T @ B.T + B @ W + Z
```

```

29 # Problema de otimiza o
30 objective = cp.Minimize(0)
31 problem = cp.Problem(objective, [
32     P >> 0,
33     Q >> 0,
34     Z >> 0,
35     lyap_expr << 0
36 ])
37 # Resolver com MOSEK
38 problem.solve(solver=cp.MOSEK, verbose=True)
39 # Calcular K = W @ Q^{-1}
40 Q_val = Q.value
41 W_val = W.value
42 K = W_val @ np.linalg.inv(Q_val)
43 # Retornar os resultados
44 return {
45     "K": K,
46 }
47 # Executar e exibir resultados
48 result = lyapunov_solver()
49 print("K =\n", result["K"])

```

Listing 1. Código Python para solução das LMIs e cálculo de K .

O valor do ganho K calculado foi:

$$K = [-1.01271145 \quad -1.41709803]$$

4. Simulação e Resultados

A fim de validar o desempenho do controlador por realimentação de estados obtido via LMIs, foram realizadas simulações no ambiente Xcos/Scilab, considerando dois cenários distintos: um problema de regulação e um problema de rastreamento de referência. As simulações visam verificar a estabilidade do sistema e o comportamento dinâmico das variáveis de estado e do sinal de controle frente aos objetivos propostos.

4.1. Caso 1: Problema de Regulação

Neste primeiro caso, o objetivo é levar o sistema a um ponto de equilíbrio nulo ($r = 0$) partindo de condições iniciais $x_0 = [0.1, 0.2]^T$, usando a lei de controle $u = -Kx$. O diagrama de simulação implementado no Xcos é ilustrado na Figura 2. A resposta da saída de interesse $y = x_2(t)$, que decai exponencialmente para zero, confirmando a estabilização do sistema como previsto pela teoria [Ogata 2010], está representada na Figura 3.

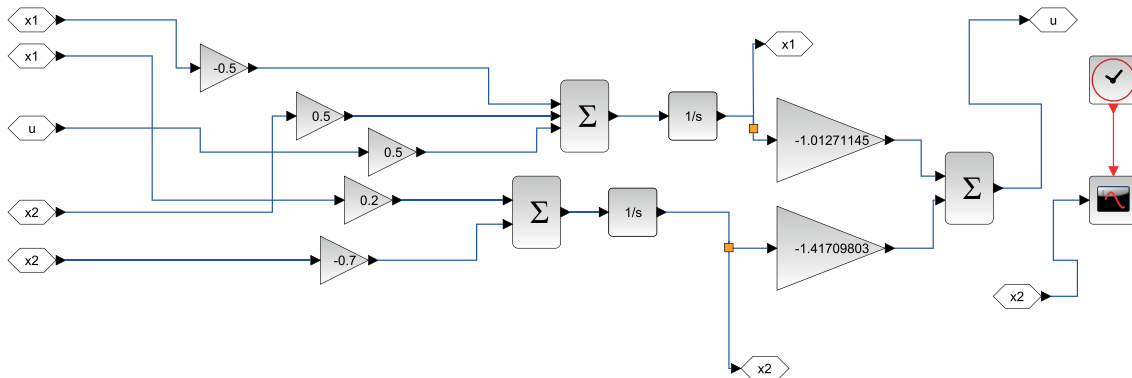


Figura 2. Diagrama Xcos para o problema de regulação ($u = -Kx$).

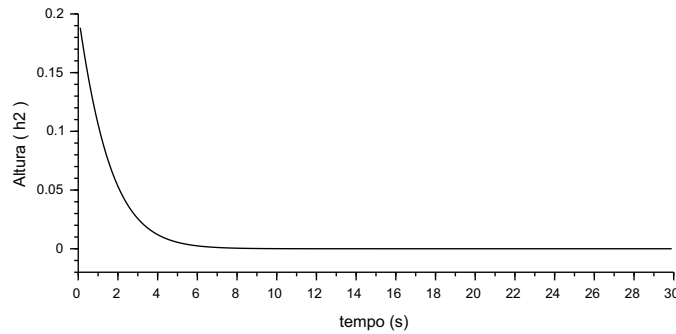


Figura 3. Resposta da saída $y = x_2(t)$ na regulação.

4.2. Caso 2: Problema de Rastreamento de Referência

Neste cenário, a saída $y = x_2$ deve seguir uma referência constante $r = 1$, partindo de $x_0 = [0, 0]^T$. A lei de controle é modificada para incluir um ganho de pré-compensação M : $u = -Kx + Mr$. O cálculo de M foi implementado em Python, conforme demonstrado na Listagem 30). O valor encontrado garante erro nulo em regime permanente para uma referência constante.

$$M = \frac{1}{[D - (C + DK)(A + BK)^{-1}B] \cdot \|K'\|^2} K'$$

```

1 # Passo 1: Calcular C_DK = C + DK
2   C_DK = C + D @ K
3
4
5 # Passo 2: Calcular A_BK = A + BK
6   A_BK = A + B @ K
7
8
9 # Passo 3: Calcular (A_BK)^-1
10  # Isso requer que A_BK seja quadrada e invertível
11  A_BK_inv = np.linalg.inv(A_BK)
12
13
14 # Passo 4: Calcular Termo_Intermediario = (C + DK)(A + BK)^-1 B
15  Termo_Intermediario = C_DK @ A_BK_inv @ B
16  # Extrai o valor escalar se for uma matriz 1x1
17  val_D = D.item() if hasattr(D, 'item') and D.size == 1 else D
18  Denom_Termol = D - Termo_Intermediario
19  # Passo 6: Calcular o segundo termo do denominador: ||K'||^2
20  # K' a transposta de K. A norma ao quadrado cp.sum_squares.
21  Norm_K_T_sq = cp.sum_squares(K.T)
22  M = (1 / (Denom_Termol * Norm_K_T_sq)) * K.T
23 return {
24     "K": K,
25     "M": M.value
26 }
27 # Executar e exibir resultados
28 print("M =\n", result["M"])

```

Listing 2. Código Python para cálculo de M .

4.3. Implementação no Xcos e Análise das Respostas

Para simular o cenário de rastreamento de referência com a nova lei de controle $u = -Kx + Mr$, foi necessário atualizar o diagrama de blocos no ambiente Xcos. As modificações envolveram a inclusão de um bloco somador para incorporar o termo Mr , além

do ajuste do sinal de referência e da conexão adequada com os blocos de ganho e planta. A Figura 4 ilustra o diagrama atualizado.

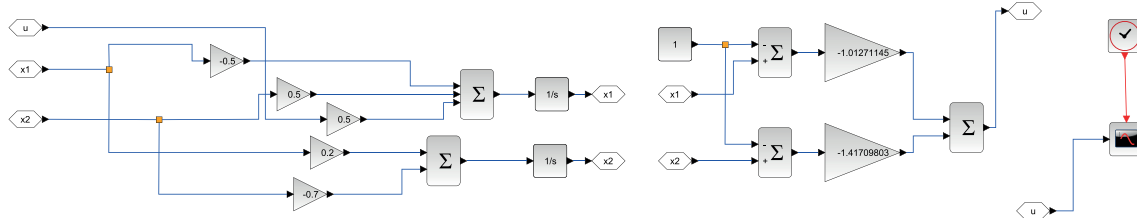


Figura 4. Diagrama Xcos para o rastreamento de referência.

As respostas obtidas com essa configuração são apresentadas nas Figuras 5, 6 e 7. que mostram, respectivamente, a altura do segundo tanque $x_2(t)$, a evolução do estado $x_1(t)$, e o sinal de controle $u(t)$ aplicado ao sistema.

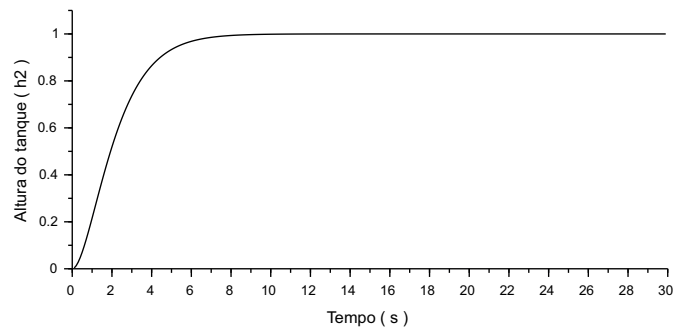


Figura 5. Resposta da saída $y = x_2(t)$ para referência constante 1.

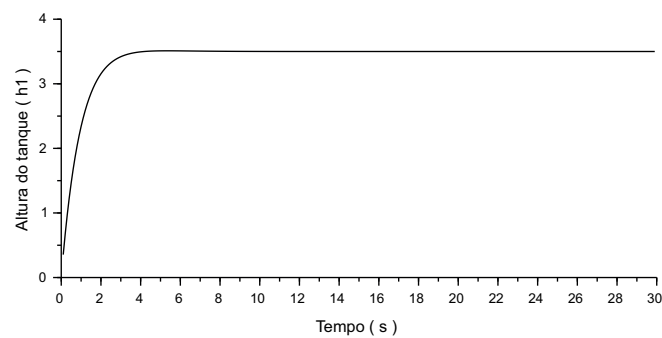


Figura 6. Resposta do estado $x_1(t)$ para referência constante 1.

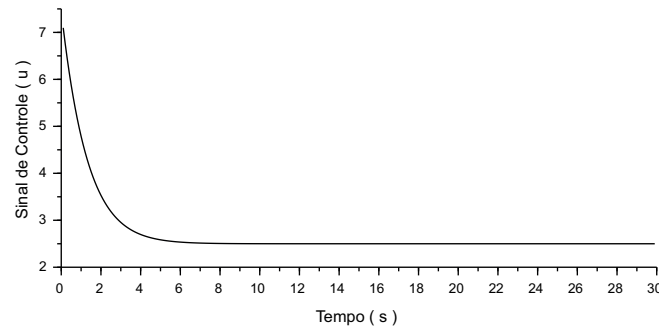


Figura 7. Sinal de controle $u(t)$ para referência constante 1.

As simulações confirmam que a estratégia de controle proposta atinge com êxito os objetivos de rastreamento. A saída $x_2(t)$ converge para o valor de referência com erro nulo em regime permanente, enquanto o estado $x_1(t)$ e o controle $u(t)$ também se estabilizam de forma suave, demonstrando a robustez do controlador mesmo diante da introdução de uma referência não nula.

5. Conclusão

Este trabalho demonstrou com sucesso o projeto de um controlador por realimentação de estados utilizando uma abordagem baseada em Desigualdades Matriciais Lineares. A metodologia fundamentada em teorias bem estabelecidas [Scherer and Weiland 2000, Ogata 2010], foi aplicada a um sistema de tanques comunicantes, e os resultados de simulação confirmaram a eficácia do controlador tanto para tarefas de regulação quanto para rastreamento de referência. As simulações realizadas no ambiente Xcos/Scilab comprovaram a eficácia da metodologia, apresentando respostas estáveis e com ótimo desempenho dinâmico. Além disso, o uso de ferramentas computacionais como Python, CVXPY e o solver MOSEK demonstrou ser uma estratégia eficiente para resolver o problema de projeto de forma automatizada, precisa e reproduzível. A técnica LMI prova ser uma ferramenta poderosa e sistemática para o projeto de controladores para sistemas lineares, convertendo um problema de estabilização em um problema de otimização convexa numericamente tratável.

Referências

- [Ogata 2010] Ogata, K. (2010). *Modern Control Engineering*. Prentice Hall, 5th edition. Um dos livros-texto mais utilizados em engenharia de controle.
- [Scherer and Weiland 2000] Scherer, C. and Weiland, S. (2000). *Linear Matrix Inequalities in Control*. SIAM (Society for Industrial and Applied Mathematics), Philadelphia, PA. Referência fundamental para o estudo de LMIs aplicadas a controle, tema central do seu trabalho.