Resolução da Lista de Exercícios

Francisco Davi Belo Rodrigues

27 de outubro de 2025

1 Exercício 1

Enunciado e dados

Consideram-se dois tanques cilíndricos interligados em série. O tanque 1 recebe uma alimentação constante e descarrega no tanque 2, que por sua vez escoa para o ambiente. As vazões de saída de cada tanque dependem do nível interno segundo a relação empírica $Q_i = k_i\sqrt{h_i}$. Os parâmetros fornecidos são resumidos na Tabela 1.

Parâmetro	**Valor**
Vazão de alimentação Q_0	$20\ \mathrm{m^3\,h^{-1}}$
Diâmetro do tanque 1 D_1	4 m
Diâmetro do tanque 2 D_2	3 m
Constante da válvula 1 k_1	$14\ \mathrm{m^{2.5}\,h^{-1}}$
Constante da válvula 2 k_2	$12\ \mathrm{m^{2.5}\,h^{-1}}$
Nível inicial no tanque 1 $h_1(0)$	3 m
Nível inicial no tanque 2 $h_2(0)$	2 m

Tabela 1: Dados operacionais da Questão 1.

Formulação do modelo

O modelo dinâmico é obtido a partir dos balanços de volume nos tanques e da relação empírica das válvulas. Considerando t em horas e mantendo as unidades fornecidas na Tabela 1, têm-se:

$$A_i = \frac{\pi D_i^2}{4}, \quad i = 1, 2, \tag{1}$$

$$Q_1 = k_1\sqrt{h_1}, \tag{2}$$

$$Q_2 = k_2\sqrt{h_2}, \tag{3}$$

$$A_1\frac{dh_1}{dt} = Q_0 - Q_1, \tag{4}$$

$$A_2\frac{dh_2}{dt} = Q_1 - Q_2, \tag{5}$$

com condições iniciais $h_1(0) = 3$ m e $h_2(0) = 2$ m. Este conjunto de equações está pronto para utilização em ambientes de simulação como o EMSO, onde os parâmetros podem ser definidos separadamente sem substituição numérica antecipada.

Resolução numérica

O sistema diferencial foi integrado em $0 \le t \le 20$ h empregando o método Runge–Kutta de quarta/quinta ordem adaptativo (`solve_ivp` do SciPy) com passo máximo equivalente a 10 s após conversão interna de unidades no script de apoio. A implementação registra também as trajetórias discretizadas (t, h_1, h_2) em arquivo auxiliar para rastreabilidade.

```python
import numpy as np
from math import pi
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

Q0 = 20.0 # m^3/h
D1 = 4.0 # m
D2 = 3.0 # m
k1 = 14.0 # m^{2.5}/h
k2 = 12.0 # m^{2.5}/h
h1_0 = 3.0 # m
h2_0 = 2.0 # m

A1 = pi * (D1 ** 2) / 4.0
A2 = pi * (D2 ** 2) / 4.0

Q0 /= 3600.0
k1 /= 3600.0
k2 /= 3600.0

T_sim = 20.0
Te = T_sim * 3600.0

def model(t, y):
    h1, h2 = y
    q1 = k1 * np.sqrt(max(h1, 0.0))
    q2 = k2 * np.sqrt(max(h2, 0.0))
    dh1dt = (Q0 - q1) / A1
    dh2dt = (q1 - q2) / A2
    return [dh1dt, dh2dt]

sol = solve_ivp(model, (0.0, Te), [h1_0, h2_0], max_step=10.0, dense_output=True)

t_hours = np.linspace(0.0, T_sim, 1000)
h1 = sol.sol(t_hours * 3600.0)[0]
h2 = sol.sol(t_hours * 3600.0)[1]

plt.figure(figsize=(6, 4))
plt.plot(t_hours, h1, label="h1 (m)")
plt.plot(t_hours, h2, label="h2 (m)")
plt.xlabel("Tempo (h)")
plt.ylabel("Nível (m)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig("figuras/questao1_niveis.png", dpi=300)

with open("figuras/questao1_niveis.dat", "w", encoding="utf-8") as f:
    f.write("tempo_h h1_m h2_m\n")
    for t, hv1, hv2 in zip(t_hours, h1, h2):
        f.write(f"{t:.6f} {hv1:.6f} {hv2:.6f}\n")

print("h1 final:", h1[-1])
print("h2 final:", h2[-1])
```

Listing 1: Script Python utilizado para a integracao numerica da Questao 1.

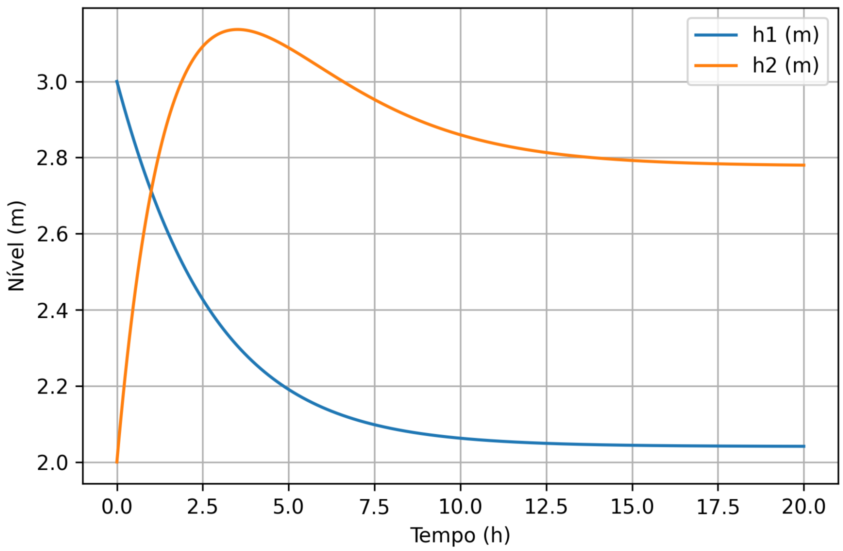

Figura 1: Perfis temporais simulados dos níveis h_1 e h_2 durante 20 horas.

Referências

[1] Autor, *Título do Livro ou Artigo*, Editora, Ano.