

## Relatório – ST9: Find the transfer function

### 1. Enunciado

**Today's Short Test (ST9):** Find the transfer function  $H[z]$  whose poles are at  $z = \frac{1}{3}$  and  $z = -\frac{1}{6}$ , in addition to one zero at  $z = \frac{1}{2}$ . Write down the corresponding difference equation. Is the transfer function stable and causal? Why?

64/109

### 2. Achando a fórmula

### 3. Código Python

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import TransferFunction, zpk2tf, tf2zpk

poles = np.array([1/3, -1/6])
zeros = np.array([1/2])
gain = 1 # Assumindo ganho K=1

print(f"Polos: {poles}")
print(f"Zeros: {zeros}")
print(f"Ganho K: {gain}")

num_poly_z, den_poly_z = zpk2tf(zeros, poles, gain)

print(f"\nCoeficientes do numerador (polinômio em z): {num_poly_z}")
print(f"Coeficientes do denominador (polinômio em z): {den_poly_z}")

if len(num_poly_z) < len(den_poly_z):
    num_poly_z_padded = np.concatenate(([0]*(len(den_poly_z) -
len(num_poly_z) - 1), num_poly_z)) # Ajuste para que b0 seja para z^0
else:
    num_poly_z_padded = num_poly_z

b_coeffs_z_inv = np.array([0, 1, -0.5]) # Coefs de x[n], x[n-1], x[n-2]
a_coeffs_z_inv = np.array([1, -1/6, -1/18]) # Coefs de y[n], y[n-1],
y[n-2]

print(f"\nCoeficientes b (numerador para z^-1): {b_coeffs_z_inv} (para
x[n], x[n-1], x[n-2])")
```

```

print(f"Coeficientes a (denominador para z^-1): {a_coeffs_z_inv} (para y[n], y[n-1], y[n-2])")

# y[n] = (1/6)y[n-1] + (1/18)y[n-2] + x[n-1] - (1/2)x[n-2]
print("\n--- Equação de Diferenças ---")
eq_str = f"y[n] = ({a_coeffs_z_inv[1]:.4f})*y[n-1] + \
({a_coeffs_z_inv[2]:.4f})*y[n-2] " \
f"+ ({b_coeffs_z_inv[1]:.4f})*x[n-1] + \
({b_coeffs_z_inv[2]:.4f})*x[n-2]"
# Corrigindo os sinais para a forma y[n] = ...
# a[0]y[n] + a[1]y[n-1] + a[2]y[n-2] = b[0]x[n] + b[1]x[n-1] + b[2]x[n-2]
# y[n] = (-a[1]/a[0])y[n-1] + (-a[2]/a[0])y[n-2] + (b[0]/a[0])x[n] + \
(b[1]/a[0])x[n-1] + (b[2]/a[0])x[n-2]
term_y1 = -a_coeffs_z_inv[1]/a_coeffs_z_inv[0]
term_y2 = -a_coeffs_z_inv[2]/a_coeffs_z_inv[0]
term_x0 = b_coeffs_z_inv[0]/a_coeffs_z_inv[0] # b0 é 0
term_x1 = b_coeffs_z_inv[1]/a_coeffs_z_inv[0]
term_x2 = b_coeffs_z_inv[2]/a_coeffs_z_inv[0]

final_eq_str = f"y[n] = {term_y1:.4f}*y[n-1] + {term_y2:.4f}*y[n-2] + \
{term_x1:.4f}*x[n-1] + {term_x2:.4f}*x[n-2]"
print(final_eq_str.replace("+ -", "- "))

print("\n--- Estabilidade e Causalidade ---")
magnitudes_poles = np.abs(poles)
print(f"Magnitudes dos polos: {magnitudes_poles}")

is_stable = np.all(magnitudes_poles < 1)
is_causal_if_stable = is_stable

if is_stable:
    print("O sistema é ESTÁVEL, pois todos os polos estão dentro do círculo unitário.")
    if is_causal_if_stable: # Assumindo que queremos um sistema causal
        print("Assumindo uma Região de Convergência (ROC) causal (para fora do polo mais externo), o sistema também é CAUSAL.")
        print("Portanto, a função de transferência é ESTÁVEL e CAUSAL.")
    else:
        print("O sistema NÃO é estável, pois um ou mais polos estão no ou fora do círculo unitário.")

print("\n--- Plotando Polos e Zeros no Plano Z ---")
fig, ax = plt.subplots(figsize=(7, 7))

unit_circle = plt.Circle((0, 0), 1, color='grey', fill=False, linestyle='--', linewidth=1)
ax.add_artist(unit_circle)

if zeros.size > 0:
    ax.plot(np.real(zeros), np.imag(zeros), 'o', markersize=10, markerfacecolor='none', markeredgecolor='blue', label='Zeros')

```

```

if poles.size > 0:
    ax.plot(np.real(poles), np.imag(poles), 'x', markersize=10,
            markeredgcolor='red', label='Polos')

ax.set_xlabel("Parte Real ( $\mathbb{R}$ )")
ax.set_ylabel("Parte Imaginária ( $\mathbb{I}$ )")
ax.set_title("Plano Z: Polos e Zeros de  $H(z)$  para ST9")
ax.grid(True, linestyle=':', linewidth=0.5)
ax.axhline(0, color='black', lw=0.5)
ax.axvline(0, color='black', lw=0.5)
ax.axis('equal')

all_coords = np.concatenate((np.real(zeros), np.real(poles),
                             np.imag(zeros), np.imag(poles), [-1.1, 1.1]))
max_abs_val = np.max(np.abs(all_coords)) if all_coords.size > 0 else 1.1
plot_limit = np.ceil(max_abs_val * 1.5) # Adiciona uma margem
if plot_limit < 1.2:
    plot_limit = 1.2
ax.set_xlim([-plot_limit, plot_limit])
ax.set_ylim([-plot_limit, plot_limit])

ax.legend()
plt.show()

```

### 3. Resultado

```

Polos: [ 0.33333333 -0.16666667]
Zeros: [0.5]
Ganho K: 1

Coeficientes do numerador (polinômio em z): [ 1. -0.5]
Coeficientes do denominador (polinômio em z): [ 1.          -0.16666667 -0.05555556]

Coeficientes b (numerador para  $z^{-1}$ ): [ 0.   1. -0.5] (para  $x[n]$ ,  $x[n-1]$ ,  $x[n-2]$ )
Coeficientes a (denominador para  $z^{-1}$ ): [ 1.          -0.16666667 -0.05555556] (para  $y[n]$ ,  $y[n-1]$ ,  $y[n-2]$ )

--- Equação de Diferenças ---
 $y[n] = 0.1667y[n-1] + 0.0556y[n-2] + 1.0000x[n-1] - 0.5000x[n-2]$ 

--- Estabilidade e Causalidade ---
Magnitudes dos polos: [0.33333333 0.16666667]
O sistema é ESTÁVEL, pois todos os polos estão dentro do círculo unitário.
Assumindo uma Região de Convergência (ROC) causal (para fora do polo mais externo), o sistema também é CAUSAL.
Portanto, a função de transferência é ESTÁVEL e CAUSAL.

```

Plano Z: Polos e Zeros de  $H(z)$  para ST9

