



UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

SEM0538 - Sistemas de Controle II

Prova 1

Data: 10/05/2024

João Vítor de Oliveira - 12611734

- 1) (1,0 ponto) O arquivo *dados.txt* contém os dados de um experimento realizado no qual uma entrada degrau de $V_m = 9,3$ V foi aplicada durante 1 s e mediu-se a resposta do sistema, y , em cm.
- Identifique a função transferência contínua, $G(s)$, entre a tensão no motor, V_m e a posição da esfera, y . Considere uma função transferência de segunda ordem.

```
%Exercício 1
dados = load('dados.txt');
time = dados(:,1);
tensao = dados(:,2);
y = dados(:,3);
%systemIdentification - Rodado uma vez pra pegar a FT

num = [16.3];
den = [1,0.6513,5.085];

sys1 = tf(num,den)
```

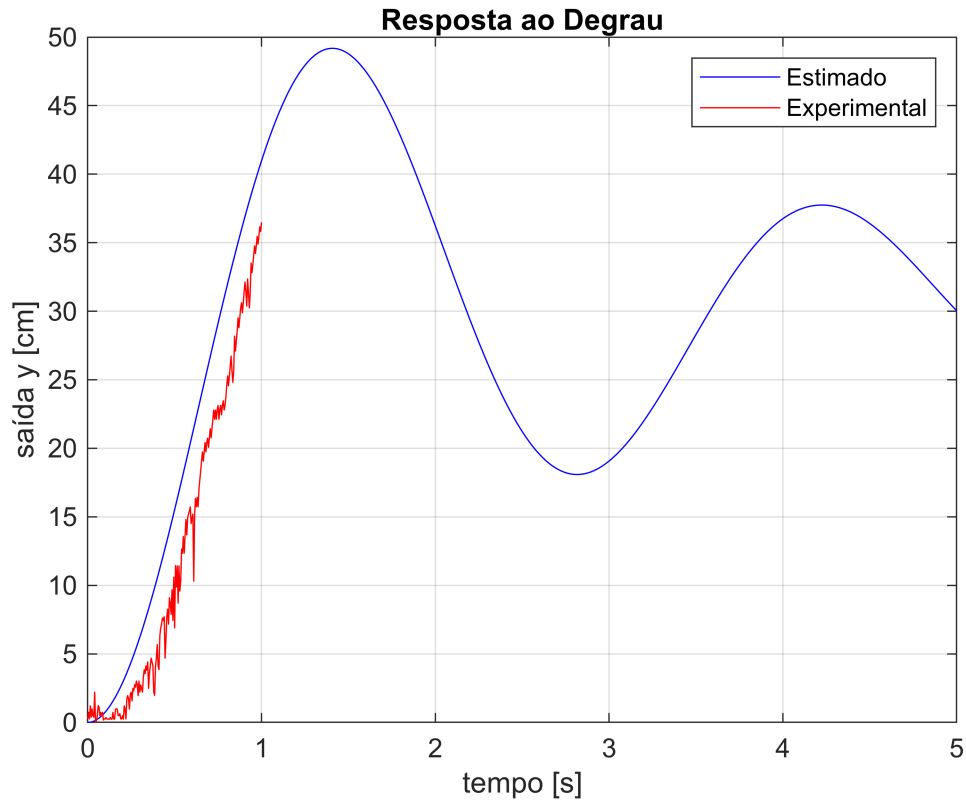
sys1 =

$$\frac{16.3}{s^2 + 0.6513 s + 5.085}$$

Continuous-time transfer function.

- Plote a resposta da função transferência encontrada juntamente com os dados experimentais para um tempo de 5 s.

```
time2 = [0:0.005:5];  
y_step = 9.4*step(sys1,time2);  
figure()  
plot(time2,y_step,'b')  
hold on  
grid  
plot(time,y,'r')  
title('Resposta ao Degrau');  
legend('Estimado','Experimental');  
xlabel('tempo [s]');  
ylabel('saída y [cm]');
```



- Analise as respostas teórica e experimental.

As respostas estão similares com uma pequena diferença de atraso.

- 2) (1,0 ponto) Encontre a representação no Espaço de Estados do sistema, considerando como estado:

$$\mathbf{x} = \begin{bmatrix} \int y \\ y \\ \dot{y} \end{bmatrix}$$

e como entrada: $u = V_m$.

```
%%Exercício 2
syms s
Y2 = 16.3*9.4/(s*(s^2 + 0.6513*s + 5.085));

%y''(t) + y'(t)*0.6513 + 5.085*y(t) = 153.22
%y''(t) = - y'(t)*0.6513 - 5.085*y(t) + 153.22
```

$$x_1(t) = \int y(t)dt, x_2(t) = y(t) \text{ e } x_3(t) = \frac{dy(t)}{dt}.$$

$$\frac{dx_1(t)}{dt} = \int y(t)dt = x_2(t)$$

$$\frac{dx_2(t)}{dt} = y(t) = x_3(t)$$

$$\frac{dx_3(t)}{dt} = \frac{dy(t)}{dt} = y'(t) = -0.6513x_3(t) - 5.085x_2(t) + 153.22$$

$$\frac{d}{dt} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -5.085 & -0.6513 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 153.22 \end{bmatrix}$$

- 3) (2,0 pontos) Calcule o ganho do controlador, $K = [K_i \ K_p \ K_d]$, tal que o sistema em malha fechada tenha polos em $s = -1$, $s = -5 + 5j$ e $s = -5 - 5j$.

- Para os polos complexos, identifique: fator de amortecimento, ζ ; frequência natural, ω_n , sobressinal esperado, M_p , tempo de subida esperado, t_r .

```
%%Exercício 3
A = [0 1 0; 0 0 1; 0 -5.085 -0.6513];
B = [0; 0; 16.3];

P = [-1, -5+5i, -5-5i];
K = place(A, B, P)
```

```
K = 1x3
    3.0675    3.3690    0.6349
```

```
polin = poly(P(2:3));
omega_n = sqrt(polin(3))
```

```
omega_n = 7.0711
```

```
zeta = polin(2)/(2*omega_n)
```

```
zeta = 0.7071
```

```
Mp = exp(-zeta*pi/sqrt(1-zeta^2))*100
```

```
Mp = 4.3214
```

```
tr = 1.8/omega_n
```

```
tr = 0.2546
```

```
G = sys1
```

```
G =
```

```
      16.3  
-----  
s^2 + 0.6513 s + 5.085
```

Continuous-time transfer function.

```
Kp = K(2)
```

```
Kp = 3.3690
```

```
Ki = K(1)
```

```
Ki = 3.0675
```

```
Kd = K(3)
```

```
Kd = 0.6349
```

```
C = tf([Kd Kp Ki],[1 0]) %[kps + ki + kds^2]/[s+0]
```

```
C =
```

```
0.6349 s^2 + 3.369 s + 3.067  
-----  
s
```

Continuous-time transfer function.

- Plote a resposta do sistema em malha fechada para uma entrada degrau de 20 cm e analise se o sistema apresenta as condições do item anterior. Caso haja diferença, explique o motivo.

```
%%b)
```

```
MFs = feedback(G*C,1)
```

```
MFs =
```

```
10.35 s^2 + 54.91 s + 50  
-----  
s^3 + 11 s^2 + 60 s + 50
```

Continuous-time transfer function.

```
figure()
step(20*MFs,time2);
hold on
```

Isso ocorre já que desconsideramos um polo q traria instabilidade ao sistema.

4) (2,0 pontos) Considere a implementação deste controlador em um sistema digital com tempo de amostragem $T_0 = 0.1$ s.

- Calcule a função de transferência discreta equivalente, $G(z)$, utilizando o Método do Segurador de Ordem Zero.

```
%Exercício 4
T0 = 0.1
```

```
T0 = 0.1000
```

```
Gdiscreta = c2d(G,T0, 'zoh'); %%Via MatLab
```

- Calcule o controlador PID discreto dado pela seguinte parametrização:

$$C_{PID}(z) = \frac{q_0 z^2 + q_1 z + q_2}{z^2 - z} = \frac{q_0(z - z_1)(z - z_2)}{z^2 - z}, \quad \frac{q_1}{q_0} = -(z_1 + z_2), \quad \frac{q_2}{q_0} = z_1 z_2.$$

$$\text{sendo } q_0 = K_p + \frac{K_d}{T_0} + K_i T_0, \quad q_1 = -K_p - \frac{2K_d}{T_0} \text{ e } q_2 = \frac{K_d}{T_0}.$$

```
q0 = Kp + Kd/T0 + Ki*T0
```

```
q0 = 10.0247
```

```
q1 = -Kp - 2*Kd/T0
```

```
q1 = -16.0668
```

```
q2 = Kd/T0
```

```
q2 = 6.3489
```

```
C_PID_discreto = tf([q0 q1 q2],[1 -1 0], T0);
```

- Calcule a função transferência em Malha Fechada, $T(z)$.

```
MFz = feedback(Gdiscreta*C_PID_discreto,1)
```

```
MFz =
```

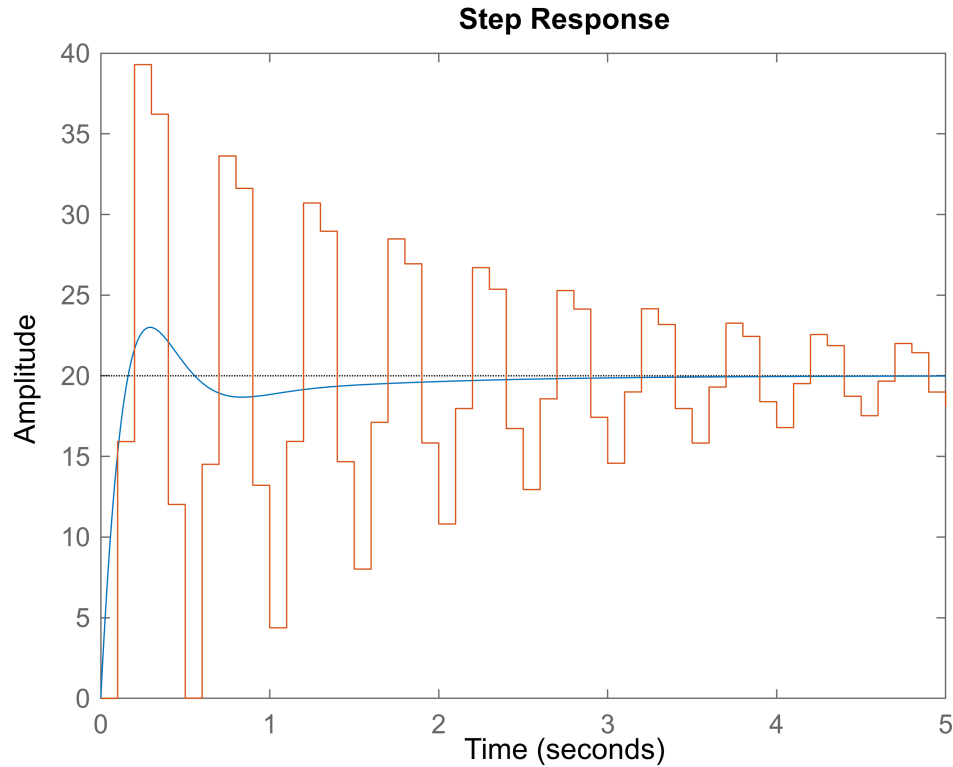
$$\frac{0.7962 z^3 - 0.497 z^2 - 0.7444 z + 0.4934}{z^4 - 2.092 z^3 + 2.328 z^2 - 1.681 z + 0.4934}$$

```
Sample time: 0.1 seconds
```

```
Discrete-time transfer function.
```

- Plote a resposta do sistema em malha fechada para uma entrada degrau de 20 cm.

```
step(20*MFz)
```



- Mostre *analiticamente* que o sistema **NÃO** satisfaz as especificações de desempenho do item (3) para este valor de T_0 .

```
%Polos contínuos para discretos  
polos_c2d = pole(c2d(MFs, T0))
```

```
polos_c2d = 3×1 complex  
    0.9048 + 0.0000i  
    0.5323 + 0.2908i  
    0.5323 - 0.2908i
```

```
%Polos discretos  
polos_discretos = pole(MFz)
```

```
polos_discretos = 4×1 complex  
    0.2909 + 0.9046i  
    0.2909 - 0.9046i  
    0.9084 + 0.0000i  
    0.6015 + 0.0000i
```

```
%[p,z]=pzmap(MFz)  
%sisotool(MFz) %Só para explorar um pouco o sisotool
```

Os polos discretos e contínuos passados para discretos são diferentes, justificando o fato do sistema não satisfazer as especificações.

5) (2,0 pontos) Encontre novos ganhos K_p , K_d e K_i do controlador PID discreto que satisfaçam as especificações utilizando o Método do Lugar das Raízes.

- Indique na figura abaixo a posição desejada para os polos de Malha Fechada.

```
%Exercício 5

%%PLOTANDO EM Z
%Polos c2d
x = [0.9048 0.5323 0.5323];
y = [0 0.2908 -0.2908];
z = x + 1i*y;
figure();
zgrid;
hold on;
plot(z, 'x');
```

- Calcule o ganho do controlador utilizando a formulação do Método do Lugar das Raízes.

$$C(z) = \frac{10,0247z^2 - 16,0668z + 6,3489}{z^2 + 3}$$

$$C(z) = \frac{10,0247(z^2 - 1,6027z + 0,6333)}{z^2 + 3}$$

$$= \frac{10,0247(z - 0,8949)(z - 0,7078)}{z^2 + 3}$$

montando $z_1 = 0,8949$

$$* C(z) = \frac{10,0247(z - 0,8949)(z - z_2)}{(z)(z - 1)}$$

$$G(z) = \frac{0,07943z + 0,0777}{z^2 - 1,888z + 0,9369}$$

$$G(z) = \frac{(z + 0,9785) \cdot 0,07943}{(z - (0,944 + 0,214j)) \cdot (z - (0,944 - 0,214j))}$$

daí: $s_0 = -s + s_j \rightarrow z_0 = e^{s_0 T_0} = e^{(-s + s_j)0,1} = 0,53 + 0,29j$

- queremos que z_0 pertence ao lugar dos polos: $|G(z_0)K(z_0)| = 160^\circ$

$$z_0 = 0,53 + 0,29j$$

$$\frac{|z_0 - 0,8949| \cdot |z_0 - z_2| \cdot |z_0 + 0,9785| \cdot |z_0 - 1|}{|z_0 - (0,944 + 0,214j)| \cdot |z_0 - (0,944 - 0,214j)|} = 180^\circ$$

: Calculadora

$$|z_0 - z_2| = 503,46 \therefore 143,46^\circ + 360^\circ$$

- sendo z_2 real: $z_0 - z_2 = 0,53 - z_2 + 0,29j$

$$\frac{0,29}{0,53 - z_2} = \tan(143,46^\circ) = -0,741$$

$$\hookrightarrow z_2 = 0,9213$$

- Ganho

$$K = \frac{|z_0| \cdot |z_0 - 1| \cdot |z_0 - (0,944 + 0,214j)| \cdot |z_0 - (0,944 - 0,214j)|}{K_0 \cdot |z_0 - 0,8949| \cdot |z_0 - 0,9213| \cdot |z_0 + 0,9785|}$$

\hookrightarrow com $K_0 \approx 0,07942$ e $z_0 = 0,53 + 0,29j$

$$\therefore K = 3,3059$$

$$C_z = (q_0 * z^2 + q_1 * z + q_2) / (z^2 - z)$$

$$C_z =$$

$$-\frac{\frac{81701}{8150} z^2 - \frac{5643071657904299}{1125899906842624}}{z - z^2}$$

$$C_z = q_0 * (z - 0.8949) * (z - 0.7078) / (z^2 - z)$$

$$C_z =$$

$$-\frac{\left(\frac{81701}{8150} z - \frac{731142249}{81500000}\right) \left(z - \frac{3539}{5000}\right)}{z - z^2}$$

$$z_0 = 0.53 + 0.29j;$$

$$z_2 = 0.9213;$$

$$k_0 = 0.07942;$$

$$K_{\text{new}} = (\text{abs}(z_0) * \text{abs}(z_0 - 1) * \text{abs}(z_0 - (0.944 + 0.214j)) * \text{abs}(z_0 - (0.944 - 0.214j))) / \text{abs}(k_0 * \text{abs}(z_0 - 0.8949))$$

$$K_{\text{new}} = 3.3059$$

$$q_0_{\text{new}} = K_{\text{new}};$$

$$a = \text{expand}(q_0_{\text{new}} * (z - 0.8949) * (z - z_2))$$

$$a =$$

$$\frac{7444317818528925}{2251799813685248} z^2 - \frac{2704074004402446717}{450359962737049600} z + \frac{24550507642231816717509}{9007199254740992000000}$$

$$q_1_{\text{new}} = -2704074004402446717 / 450359962737049600$$

$$q_1_{\text{new}} = -6.0043$$

$$q_2_{\text{new}} = 24550507642231816717509 / 9007199254740992000000$$

$$q_2_{\text{new}} = 2.7257$$

$$K_{d\text{new}} = q_2_{\text{new}} * T_0$$

$$K_{d\text{new}} = 0.2726$$

$$K_{p\text{new}} = -q_1_{\text{new}} - 2 * (K_{d\text{new}} / T_0)$$

$$K_{p\text{new}} = 0.5529$$

$$K_{i\text{new}} = (q_0_{\text{new}} - K_{p\text{new}} - K_{d\text{new}} / T_0) / T_0$$

$$K_{i\text{new}} = 0.2734$$

%Novo C discreto

$$\text{CPID}_z_{\text{new}} = \text{tf}([q_0_{\text{new}} \ q_1_{\text{new}} \ q_2_{\text{new}}], [1 \ -1 \ 0], T_0)$$

$$\text{CPID}_z_{\text{new}} =$$

$$\frac{3.306 \ z^2 - 6.004 \ z + 2.726}{\text{-----}}$$

$$z^2 - z$$

Sample time: 0.1 seconds
Discrete-time transfer function.

%Nova MF discreta

```
MFz_new = feedback(Gdiscreta*CPID_z_new, 1)
```

MFz_new =

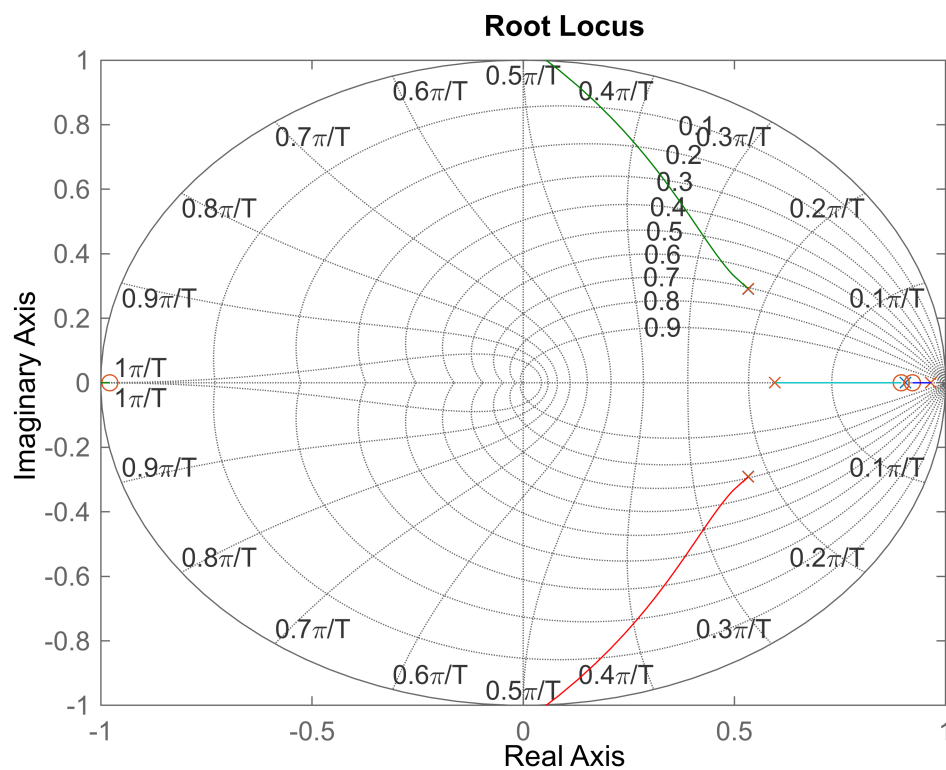
$$\frac{0.2626 z^3 - 0.22 z^2 - 0.2501 z + 0.2118}{z^4 - 2.625 z^3 + 2.605 z^2 - 1.187 z + 0.2118}$$

Sample time: 0.1 seconds
Discrete-time transfer function.

- Esboce o Lugar das Raízes na figura abaixo.

```
rlocus(MFz_new)
grid on;
```

```
rlocus(MFz_new)
ylim([-1,1]);
xlim([-1,1]);
grid on;
```

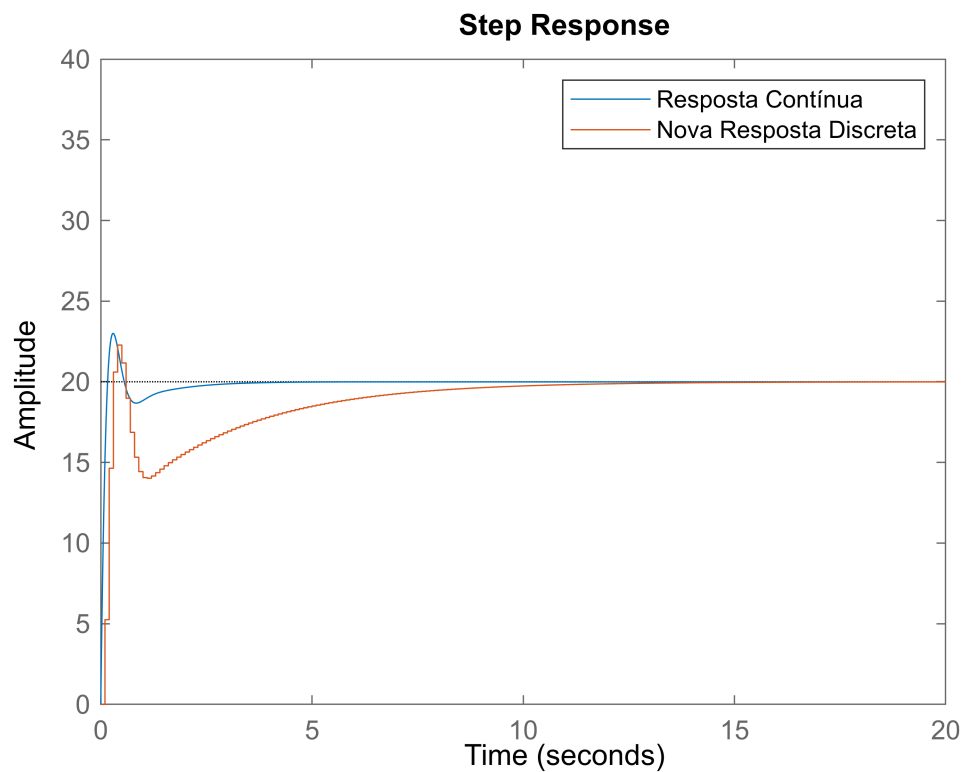


```
pole(MFz_new)
```

```
ans = 4×1 complex
    0.9650 + 0.0000i
    0.5322 + 0.2917i
    0.5322 - 0.2917i
    0.5959 + 0.0000i
```

- Plote a resposta do sistema em Malha Fechada para uma entrada degrau de 20 cm e compare com os gráficos anteriores.

```
%Comparação com sist. cont.
figure();
step(20*MFs)
hold on
step(20*MFz_new)
ylim([0,40]);
xlim([0,20]);
hold off
legend('Resposta Contínua', 'Nova Resposta Discreta')
```



6) (2,0 pontos) Implementação do controlador PID.

- Escreva, para ser implementado em Arduino, um programa para avaliar o desempenho do controlador.

```
#include <PID_v1_bc.h> // Inclua a biblioteca PID
#include <Servo.h> // Inclua a biblioteca Servo para controle de servo-motor
```

```

// Defina os pinos para o sensor e o atuador (saída de controle)
const int sensorPin = A0; // Pino analógico para leitura do sensor
const int actuatorPin = 9; // Pino PWM para controle do atuador (servo-motor)

// Defina os parâmetros do controlador PID
double Setpoint = 100; // Defina o valor desejado (setpoint)
double Input, Output; // Variáveis para leitura do sensor e saída do controlador
double Kp = 0.5529, Ki = 0.2734, Kd = 0.2726; // Ganho do controlador PID, coloquei o do discreto, mas depende

// Crie uma instância do objeto PID
PID myPID(&Input, &Output, &Setpoint, Kp, Ki, Kd, DIRECT);

// Crie uma instância do objeto Servo para controlar o servo-motor
Servo myServo;

void setup() {
    // Inicialize o sensor e o atuador
    pinMode(sensorPin, INPUT);
    myServo.attach(actuatorPin);

    // Inicialize o controlador PID
    myPID.SetMode(AUTOMATIC);
    myPID.SetSampleTime(100); // Defina o tempo de amostragem em milissegundos (100 ms)
    myPID.SetOutputLimits(0, 180); // Defina os limites de saída do controlador (0 a 180 graus para o servo)

    // Inicialize a comunicação serial
    Serial.begin(9600);
}

void loop() {
    // Leia o valor do sensor
    Input = analogRead(sensorPin);

    // Calcule o controle PID
    myPID.Compute();

    // Ajuste a saída do controlador para o atuador (servo-motor)
    myServo.write(Output);

    // Imprima os valores
    Serial.print("Sensor: ");
    Serial.print(Input);
    Serial.print(" | Kp: ");
    Serial.print(Kp);
    Serial.print(" | Ki: ");
    Serial.print(Ki);
    Serial.print(" | Kd: ");
    Serial.println(Kd);
    Serial.print(" | Setpoint: ");
    Serial.print(Setpoint);
    Serial.print(" | PWM: ");
    Serial.println(Output); // No caso de servo-motores, o valor de saída é diretamente o ângulo

    // Espere um tempo antes da próxima iteração
    delay(100); // Intervalo de amostragem de 100 ms
}

```

