



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FEMEC 42060

CONTROLE DE SISTEMAS LINEARES

Relatório de Laboratório 4 - Sistemas instáveis em MA

Prof. Pedro Augusto

20 de junho de 2022

1 Objetivos

Neste laboratório utilizaremos a teoria de controle para estabilizar um sistema instável em malha aberta. Verificaremos o efeito do atraso no controle desse tipo de sistemas.

2 Introdução

2.1 Estabilidade de sistemas dinâmicos

A estabilidade é a propriedade mais importante em sistemas de controle. Informalmente pode-se dizer um sistema é estável se a resposta a qualquer excitação razoável não divergir. Existem diversos critérios formais para avaliar a estabilidade de um dado processo: Estabilidade no sentido de Lyapunov, Estabilidade Interna, Estabilidade BIBO. Essa última definição é comumente utilizada em sistemas lineares e invariantes no tempo.

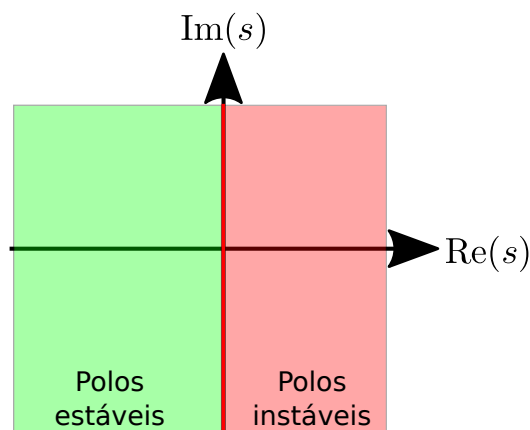
Um sistema é dito ser BIBO (*Bounded-Input Bounded-Output*) estável se, para uma entrada limitada, a saída correspondente também é limitada. É possível escrever essa definição na seguinte forma matemática:

$$\int_0^{\infty} |g(t)| dt < \infty$$

sendo $g(t)$ a resposta impulsiva.

Note que estabilidade é uma propriedade da planta e não da entrada que está sendo aplicada.

Uma técnica para determinar se um sistema linear e invariante no tempo (SLIT) é estável ou não consiste em determinar a localização dos polos da função de transferência. Mais especificamente, um SLIT é estável se e somente se todos os polos da função de transferência tiverem parte real negativa, isto é,



2.2 Atraso de transporte

Sistemas de controle estão sujeitos a **atrasos**, isto é, diferenças entre a mudança na referência e a saída variar ou entre a leitura do sensor e a atualização da entrada da planta. Esses atrasos são gerados por diversos fatores, por exemplo, tempo computacional, tempo de resposta do atuador e tempo de transporte.

Como estamos empregando o Arduino para implementação do controlador, o atraso ocorrerá devido aos tempos computacionais para leitura de sensor, cálculo da ação de controle e comunicação com o computador. Esse atraso pode prejudicar o desempenho da malha de controle ou mesmo causar instabilidade.

Matematicamente, o efeito de um atraso de T_d segundos é dado por ¹

$$y(t) = g(t - T_d)u(t - T_d)$$

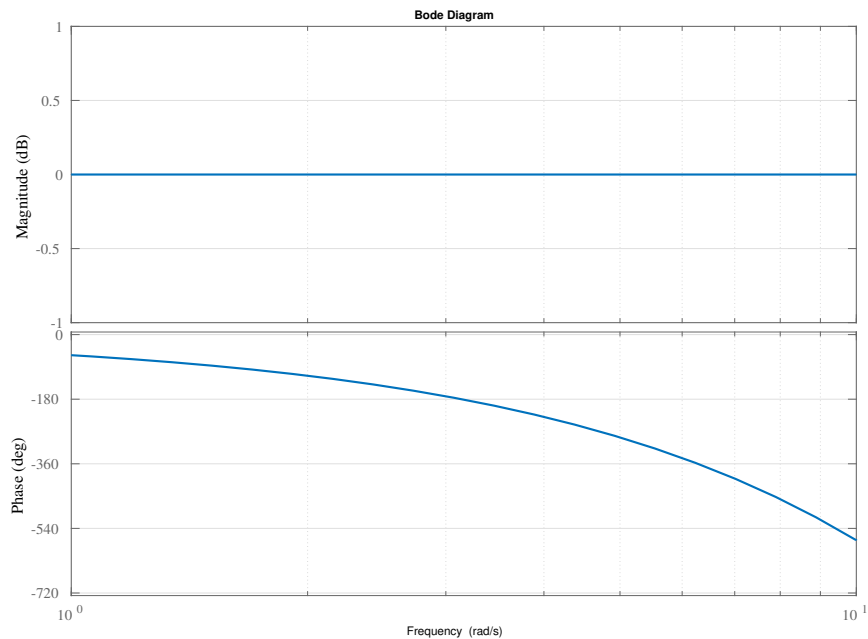
Aplicando a transformada de Laplace, pode-se escrever

$$Y(s) = e^{-sT_d}G(s)U(s)$$

Então, substituindo s por $j\omega$, em que ω é a frequência do sinal senoidal de entrada, é possível determinar o diagrama de Bode dessa função da transferência, isto é,

$$|G(j\omega)|_{dB} = 0 \quad \angle G(j\omega) = -\omega T_d$$

Graficamente, tem-se que



¹A demonstração pode ser encontrada no Cap. 2 do livro do Ogata

Portanto um dos efeitos do atraso é a diminuição da margem de fase. Quanto maior o tempo de atraso, menor é a margem de fase. Como consequência, tem-se um menor fator de amortecimento ξ e uma resposta do sistema mais oscilatória. Em casos extremos o atraso pode causar instabilidade.

2.3 Modelo linear do pêndulo invertido

Considere um pêndulo convencional mostrado na Figura 1, em que M_t é a massa total, g é a aceleração da gravidade e L é o comprimento da haste.

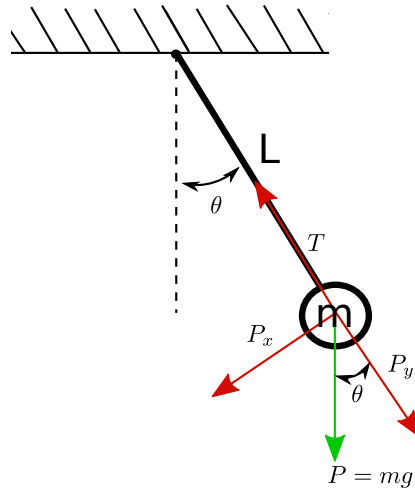


Figura 1: Digrama esquemático do pêndulo.

Note que a variável de entrada é o torque aplicado ao eixo.

A aceleração da massa pode ser descrita em função do ângulo θ da seguinte forma:

$$\ddot{\theta}(t) = \frac{1}{M_t L^2} T(t) - \frac{g}{L} \sin \theta(t)$$

Por meio da expansão em série de Taylor, é possível escrever a dinâmica em torno do ponto de equilíbrio $(\pi, 0)$, obtendo-se

$$\ddot{\theta}(t) = \frac{1}{M_t L^2} T(t) + \frac{g}{L} \theta(t)$$

Utilizando a transformada de Laplace obtém-se a função de transferência de $T(s)$ para $\Theta(s)$. Mais especificamente, tem-se que

$$\frac{\Theta(s)}{T(s)} = G(s) = \frac{1}{M_t L^2} \cdot \frac{1}{s^2 - \frac{g}{L}} \quad (1)$$

Os polos do sistema em MA são

$$s_{1,2} = \pm \sqrt{\frac{g}{L}}$$

Como que um dos polos tem parte real positiva, a dinâmica em torno do ponto de equilíbrio $(\pi, 0)$ (pêndulo invertido) é instável em MA. Neste experimento projetaremos um controlador utilizando o LGR para estabilizar a planta.

Vale lembrar que a estabilização de sistemas somente é possível adotando estruturas de controle em MF.

3 Lista de materiais

- Arduino UNO
- Fios de conexão
- Fonte DC
- Motor DC
- Ponte H - L298N
- *Protoboard*
- Balança
- Régua
- Peças para montagem do pêndulo invertido (rodas, haste e conexões)

4 Procedimento experimental

4.1 Obtenção dos dados do modelo

- Determine os parâmetros da função de transferência

$$\frac{\Theta(s)}{T(s)} = G(s) = \frac{1}{M_t L^2} \cdot \frac{1}{s^2 - \frac{g}{L}} \quad (2)$$

em que L o comprimento da haste em **metros**, $g = 9,81 \text{ m/s}^2$ e a massa total é dada por

$$M_t = m + \frac{m_h}{2} \quad (3)$$

sendo m a massa da roda em kg e m_h a massa da haste em kg.

- Monte o seguinte sistema:

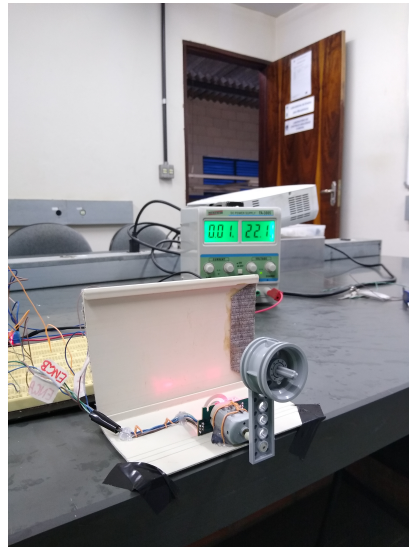


Figura 2: Pêndulo invertido.

4.2 Sistema em malha fechada com controlador proporcional

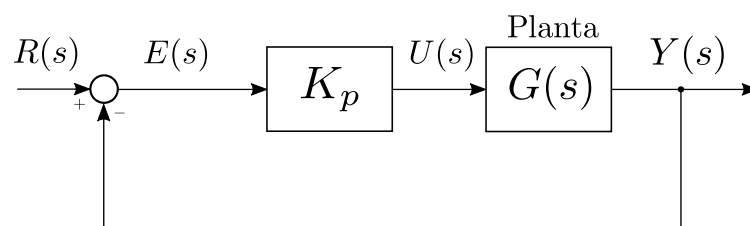


Figura 3: Sistema em malha fechada com controlador proporcional.

- Monte o circuito eletrônico mostrado abaixo

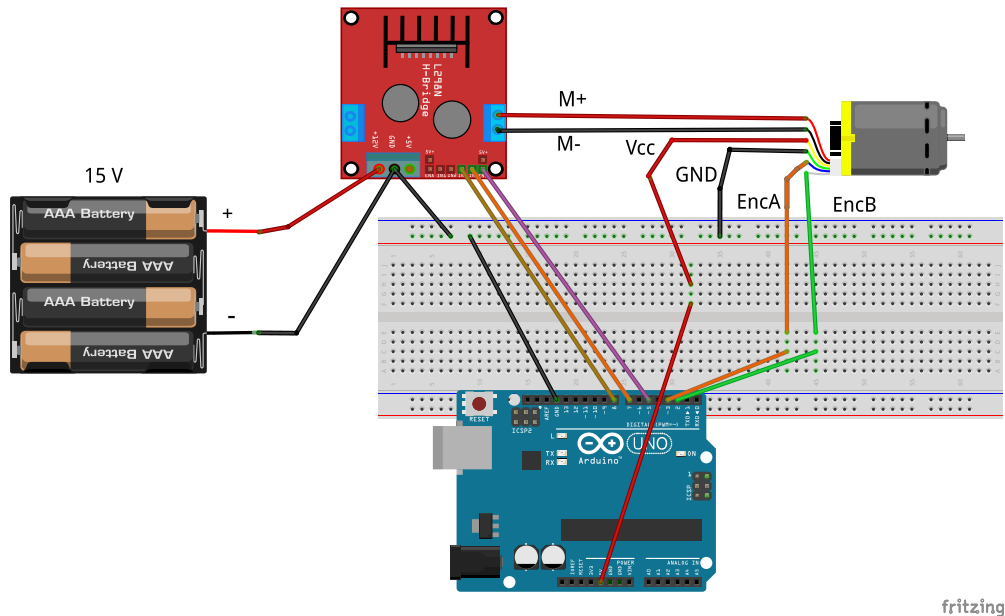


Figura 4: Montagem circuito eletrônico.

- Altere o código abaixo para implementar um sistema em malha fechada com **controlador proporcional**
- Tente estabilizar a planta para **três ganhos do controlador entre 0.05 e 1.0**

```
//Incluindo biblioteca para leitura do encoder
#include <Encoder.h>
//Definindo objeto meuEncoder
Encoder meuEncoder(2, 3);
//Definindo outras variaveis uteis
double theta = 0.0, erro = 0.0, u = 0.0, T = 0.0;
long contEnc = 0.0;

// Escolhendo referencia para a velocidade
double thetaRef = 0.0; //em radianos

void setup() {
    //Iniciando comunicacao serial
    Serial.begin(2000000);
```

```
//Definindo Entradas da ponte H
pinMode(5,OUTPUT); //velocidade de giro
}

void loop() {

    //Determinando leitura atual do encoder
    contEnc = meuEncoder.read();

    //Calculando theta a partir da leitura do encoder
    theta = contEnc*2*3.1415/(334.0*4.0);

    //Obtendo erro de rastreamento
    erro = XXXXXXXX;

    //Calculando acao do controlador
    T = XXXXXXXX;

    //Alterando sentido de giro de acordo com o sinal do
    controle
    if (T >= 0){
        //sentido horario
        digitalWrite(7,HIGH);
        digitalWrite(8,LOW);
    }
    else{
        //sentido anti-horario
        digitalWrite(7,LOW);
        digitalWrite(8,HIGH);
        T = -T;
    }

    // Gerando PWM a partir do torque
    u = 19442.0*T + 8.0;

    //Saturando na faixa linear
    u = min(u,100.0); //limitando superiormente
    u = max(u,0.0); //limitando inferiormente

    //Aplicando controle a planta
    analogWrite(5,u*255.0/100.0);

    //Imprimindo grandezas de interesse na porta serial
```



```
Serial.print(theta*180.0/3.1415);  
Serial.print(" ");  
Serial.print(T);  
Serial.print(" ");  
Serial.print(u);  
Serial.print(" ");  
Serial.println(micros()/1000000.0);  
}
```

- Responda as Seções 1 e 2 do Relatório.

4.3 Projeto do controlador

- Considerando a função de transferência da Seção 4.1, **projete** um compensador na forma

$$C(s) = K \frac{s+a}{s+b}$$

de modo que os **polos dominantes de malha fechada sejam** $-30 \pm 30j$.

- Estime o máximo sobressinal e o tempo de pico esperados a partir dos polos do item anterior

4.4 Implementação do controlador

O controlador será implementado digitalmente. Para isso, precisamos obter uma forma discreta do controlador. O passo-a-passo a seguir mostra como obter tal forma equivalente e implementar o controlador.

- Obtenha um equivalente digital para o controlador projetado na seção anterior por meio do Matlab utilizando o código a seguir

```
% Definindo uma variavel auxiliar  
s = tf([1 0],1);  
% Substitua K a e b com os valores projetados  
C = K*(s+a)/(s+b);  
%Obtendo um equivalente discreto por meio do metodo de  
Tustin  
Cd = c2d(C, 1e-3, 'tustin');
```

Nota: No Matlab ou em *software* similar, utilize o comando `zpk(Cd)` para representar o controlador na forma

$$\frac{T(z)}{E(z)} = C(z) = \bar{K} \frac{z + \bar{a}}{z + \bar{b}}$$

Então, o controlador pode ser implementado digitalmente por

$$\tau(k) = -\bar{b}\tau(k-1) + \bar{K}(e(k) + \bar{a}e(k-1)) \quad (4)$$

em que k é o índice de tempo discreto.

- Implemente o controlador obtido para manter o pêndulo invertido na posição de equilíbrio. Com esse propósito, altere o código mostrado abaixo com os parâmetros do controlador $C(z)$ conforme (4)

```
//Incluindo biblioteca para leitura do encoder
#include <Encoder.h>
//Definindo objeto meuEncoder
Encoder meuEncoder(2, 3);
//Definindo outras variaveis uteis
double theta = 0.0, Tant = 0.0, erroant = 0.0, TempoIni =
    0.0, erro = 0.0, u = 0.0, T = 0.0;
long contEnc = 0.0;

// Escolhendo referencia para a velocidade
double thetaRef = 0.0; //em radianos

void setup() {
    //Inicializando comunicacao serial
    Serial.begin(2000000);

    //Definindo Entradas da ponte H
    pinMode(5,OUTPUT); //velocidade de giro
}

void loop() {
    //Determinando tempo atual
    TempoIni = micros();

    //Determinando leitura atual do encoder
    contEnc = meuEncoder.read();
```

```
//Calculando theta a partir da leitura do encoder
theta = contEnc*2.0*3.1415/(334.0*4.0);

// degrau de referencia
if(TempoIni/1000000.0 > XXXX){
    thetaRef = XXXXX*3.1415/180.0; //em radianos
}

//Obtendo erro de rastreamento
erro = XXXXXXX;

//Calculando controle - eq. (5)
T = -bbarra*Tant + Kbarra*(erro + abarra*erroant);

//Salvando valores para o proximo instante de amostragem
    (Housekeeping)
erroant = erro;
Tant = T;

//Alterando sentido de giro de acordo com o sinal do
    controle
if (T >= 0){
    //sentido horario
    digitalWrite(7,HIGH);
    digitalWrite(8,LOW);
}
else{
    //sentido anti-horario
    digitalWrite(7,LOW);
    digitalWrite(8,HIGH);
    T = -T;
}

// Gerando PWM a partir do torque - eq. (4)
u = 19442.0*T + 8.0;

//Saturando na faixa linear
u = min(u,100.0); //limitando superiormente
u = max(u,0.0); //limitando inferiormente

//Aplicando controle a planta
// delay(5); //ja em ms - atraso
analogWrite(5,u*255.0/100.0);
```

```
//Imprimindo grandezas de interesse na porta serial
Serial.print(theta*180.0/3.1415);
Serial.print(" ");
Serial.print(T);
Serial.print(" ");
Serial.println(micros()/1000000.0);

//Esperando proximo instante de amostragem 1ms
while ((micros() - TempoIni) < 1000.0){ }
}
```

- Altere o código anterior para aplicar um degrau de referência de 20° em 8 s
- Salve os dados em um arquivo .txt e verifique o comportamento da resposta experimental. Para isso, pode-se empregar o código a seguir para gerar uma figura no Matlab

```
%Criando figura
load('NomeArquivo.txt')
Data = NomeArquivo;
%
% Plotando dados
% plot(Data(:,4), Data(:,1), 'b-', 'LineWidth', 2) % Sec. 4.2
plot(Data(:,3), Data(:,1), 'b-', 'LineWidth', 2)
grid on

%configurando grafico
xlabel('$t\sim(s)$', 'FontSize', 20, 'FontName', 'Times');
ylabel('$y\sim(deg)$', 'FontSize', 20, 'FontName', 'Times');
set(gca, 'FontSize', 20, 'FontName', 'Times')
```