



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FEMEC 42060

CONTROLE DE SISTEMAS LINEARES

Relatório de Laboratório 3 - Sistemas de 1^a e 2^a ordem

Prof. Pedro Augusto

2 de junho de 2022

1 Objetivos

Na presente aula de laboratório verificar-se-ão as respostas de sistemas de primeira e segunda ordem ao degrau.

2 Introdução

A resposta de um sistema pode ser dividida em respostas transitória e em regime permanente (Figura 1).

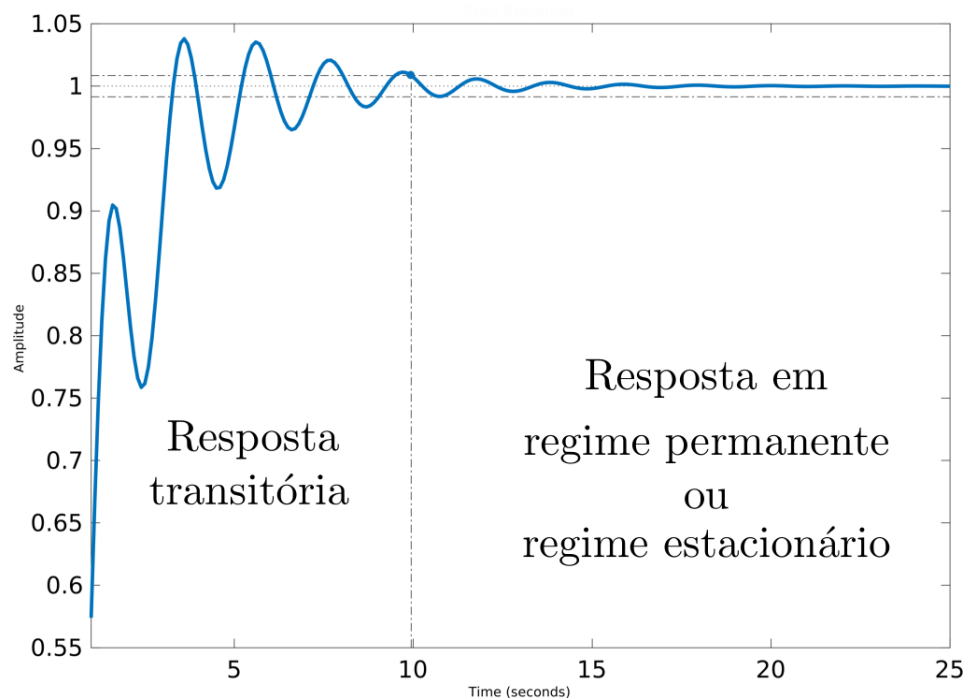


Figura 1: Resposta de sistemas dinâmicos.

Tipicamente, o projeto de um controlador envolve modificar as características de tais respostas. Para isso, definem-se critérios de desempenho para a resposta do processo considerando certos tipos de entrada. Por simplicidade, essa análise será realizada para sistemas de primeira e segunda ordem considerando uma excitação do tipo degrau. Então, índices de desempenho serão definidos e serão base de comparação entre diferentes controladores.

2.1 Sistemas de primeira ordem

Sistemas de primeira ordem podem ser descritos no domínio da variáveis complexa s por

$$Y(s) = \frac{k}{s+a} U(s)$$

Supondo uma entrada do tipo degrau com amplitude A $\left(U(s) = \frac{A}{s} \right)$, tem-se que

$$Y(s) = \frac{k}{s+a} \cdot \frac{A}{s} = \frac{-kA}{a} \cdot \frac{1}{s+a} + \frac{kA}{a} \cdot \frac{1}{s}$$

Utilizando a Transformada Inversa de Laplace, obtém-se

$$y(t) = \frac{kA}{a} - \frac{kA}{a} e^{-at}, \quad t \geq 0$$

A Figura 2 mostra a evolução temporal da saída.

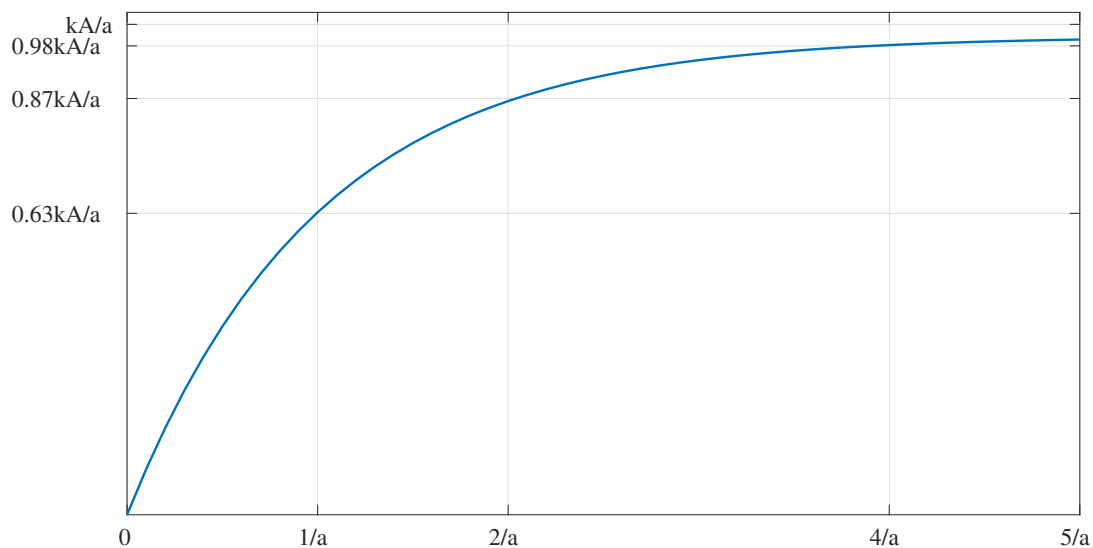


Figura 2: Resposta ao degrau de um sistema de primeira ordem.

O tempo $\tau = \frac{1}{a}$ s é denominado constante de tempo. Os requisitos de desempenho para sistemas desse tipo podem envolver uma determinada constante de tempo ou a precisão de rastreamento, isto é, a diferença entre a referência e a saída em regime permanente.

2.2 Sistemas de segunda ordem - caso subamortecido

Um sistema de segunda ordem pode ser representado pela seguinte função de transferência:

$$U(s) \rightarrow \boxed{\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}} \rightarrow Y(s)$$

em que $\omega_n > 0$ e ξ são denominados frequência natural não amortecida e coeficiente de amortecimento, respectivamente.

Os polos dessa função de transferência são

$$s_{1,2} = -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1}$$

quando $0 < \xi < 1$ tem-se que

$$s_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1 - \xi^2}$$

Esse cenário é denominado subamortecido. A forma geral da resposta de dinâmicas desse tipo é na Figura 3.

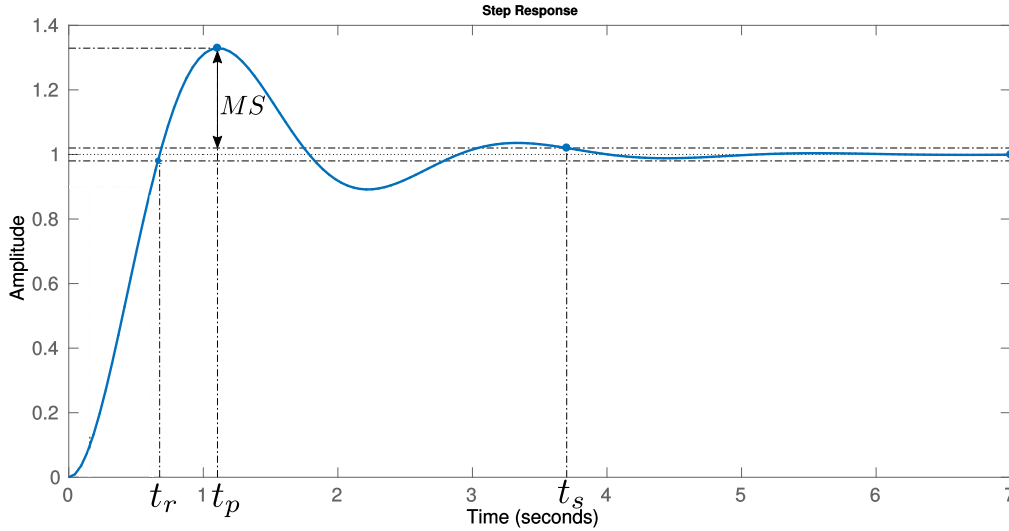


Figura 3: Resposta ao degrau de sistemas de segunda ordem.

A partir desse resposta, definem-se os seguintes índices de desempenho:

1. t_r : tempo de subida
2. t_p : tempo de pico
3. t_s : tempo de **estabelecimento**/acomodação/assentamento
4. MS : máximo sobressinal ou porcentagem de *overshoot*

Seguindo o devido procedimento matemático, é possível relacionar tais requisitos com ξ e ω_n , obtendo-se

$$t_r = \frac{\pi - \arccos \xi}{\omega_n \sqrt{1 - \xi^2}} \text{ s}$$

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \xi^2}} \text{ s}$$

$$MS = 100e^{-\frac{\xi\pi}{\sqrt{1 - \xi^2}}} \%$$

$$t_s = \frac{4}{\xi\omega_n} \text{ s}$$

Nesta aula verificaremos experimentalmente respostas de primeira e segunda ordem.

3 Lista de materiais

Os materiais para realização do presente laboratório são listados abaixo

- Arduino UNO
- Fios de conexão
- Fonte DC
- Motor DC
- Ponte H - L298N
- *Protoboard*

4 Procedimento experimental

Inicialmente será determinado um modelo para o motor a partir da resposta ao degrau. Posteriormente tal modelo será utilizado no projeto de uma lei de controle.

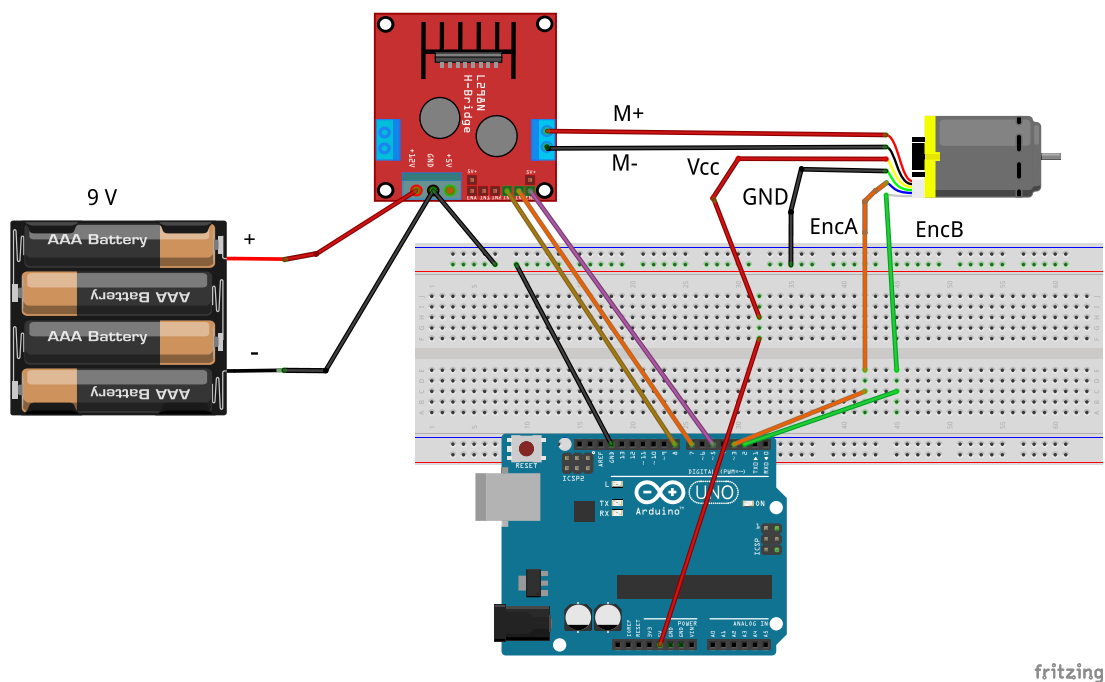
4.1 Sistemas de primeira ordem

Determinar-se-á um modelo entre *duty cycle* e velocidade do motor a partir da resposta ao degrau. Com esse propósito, siga o procedimento a seguir

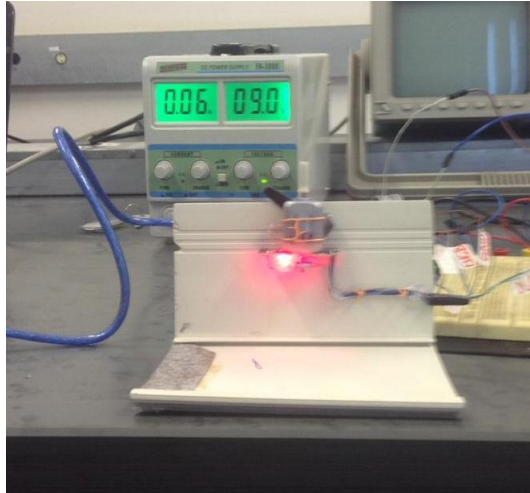
- Faça download da biblioteca Encoder.h em <https://github.com/PaulStoffregen/Encoder> e inclua a biblioteca em “Sketch”-> “Incluir Biblioteca”-> “Adicionar Biblioteca ZIP”

- Monte o circuito

MUITA ATENÇÃO NA CONEXÃO DOS CABOS VCC E GND DO ENCODER!!!



- Posicione o motor na posição indicada na Figura 4.1



- O código abaixo que permite controlar o *duty cycle* do motor e medir a velocidade

```
//Incluindo biblioteca para leitura do encoder
#include <Encoder.h>
Encoder meuEncoder(2, 3);

//Definindo variaveis
double velAng, theta = 0.0, thetaAnt = 0.0;
double tempo1 = 0.0, tempo2 = 0.0, dt, u;
long contEnc = 0.0;
double vel1 = 0.0, vel2 = 0.0, vel3 = 0.0, mediavel;

void setup() {

    //Inicializando comunicacao serial
    Serial.begin(115200);

    //Definindo saidas
    pinMode(5, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);

    //sentido horario
    digitalWrite(7, HIGH);
    digitalWrite(8, LOW);
}
```

```
void loop() {  
    //Salvando valores anteriores  
    tempo1 = tempo2;  
    thetaAnt = theta;  
  
    //Determinando leitura atual do encoder  
    contEnc = meuEncoder.read();  
  
    //Calculando theta a partir da leitura do encoder  
    theta = contEnc*2.0*3.14/(334*4);  
  
    //Determinando tempo atual  
    tempo2 = micros();  
  
    //Calculando diferenca de tempo  
    dt = tempo2 - tempo1;//em micro s  
    //Calculando velocidade angular  
    velAng = (theta - thetaAnt)/dt*1000000.0;//em rad/s  
  
    //calculando media movel dos tres ultimos valores  
    vel1 = vel2;  
    vel2 = vel3;  
    vel3 = velAng;  
    mediavel = (vel1 + vel2 + vel3)/3.0;  
  
    //Variando duty cycle  
    if(tempo2/1000000.0 >= 5.0){  
        u = xxxxx.0;  
    }else if(tempo2/1000000.0 >= 8.0){  
        u = xxxxxx.0;  
    }  
  
    analogWrite(5,255.0*u/100.0);  
  
    //Imprimindo os valores na porta serial  
    Serial.print(mediavel);  
    Serial.print(" ");  
    Serial.print(u);  
    Serial.print(" ");  
    Serial.println(tempo2/1000000);//em s  
}
```


- Modifique o código acima para aplicar um degrau de 80 % na entrada do motor após 5 s e zero a entrada após 8 s
- Monitore (Ctrl+Shift+M) e salve os dados da porta serial em um arquivo .txt
Nota: Conte 2 s após aplicar o degrau e desconecte o cabo de alimentação do Arduino
- Importe os dados do arquivo .txt para o Matlab (ou *software* similar) e gere um gráfico com a resposta do sistema. Para Matlab, pode-se utilizar o código a seguir

```
clear
close all
clc

%Carregando dados
load('NomeDoArquivo.txt')
Data = NomeDoArquivo;

%Criando figura
figure
plot(Data(:,3), Data(:,1), 'b-', 'LineWidth', 2)
grid on, hold on
% Plotando dados de controle
plot(Data(:,3), Data(:,2), 'r--', 'LineWidth', 2)

xlim([4.9 max(Data(:,3))])

%criando legenda
legend('y', 'u');

%Ajustando nome dos eixos
xlabel('t (s)');
```

- A partir do gráfico, estime uma função de transferência considerando um sistema de primeira ordem, isto é,

$$\frac{Y(s)}{U(s)} = G(s) = \frac{K}{s + a}$$

Nota: O degrau aplicado foi de 80 % isto é 80 unidades ($U(s) = 80/s$).

- Compare graficamente a resposta do sistema identificado com a resposta experimental. Para isso, empregue o código substituindo os valores de K e a calculados

```
clear
close all
clc

%Carregando dados
load('NomeArquivo.txt')
Data = NomeArquivo;

%Criando figura
figure
plot(Data(:,3), Data(:,1),'b-', 'LineWidth',2)
grid on, hold on

%Definindo FT
K = xxxxxxxx;
a = xxxxxx;
G = tf(K,[1 a]);

% Obtendo resposta ao degrau e imprimindo no grafico
[y,t] = step(G,8);
plot([0; t + 5],[0; y.*80], 'r--', 'LineWidth',2)

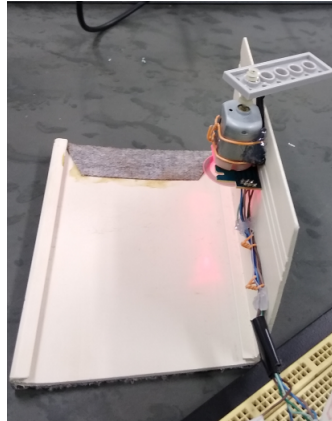
%Ajustando nome dos eixos e tamanho de letra
xlabel('t (s)');
ylabel('y (rad/s)');

xlim([4.9 max(Data(:,3))])

%legenda
h = legend('y_{real}', 'y_{est}');
```

5 Sistemas de segunda ordem

Nesta seção controlar-se-á a posição do eixo do motor. Para ficar mais claro a posição do eixo do motor, acople (com cuidado) uma haste no motor conforme mostrado na figura a seguir.



5.1 Controle de um sistema de segunda ordem

- A partir da função de transferência entre **velocidade** e *duty cycle* obtida na Seção 2.1, escreva um modelo entre *duty cycle* e a **posição**

Lembrete : A posição angular é a integral da velocidade angular.

- Projete um controlador proporcional de modo que $MS = 40\%$
- Empregue o código mostrado a seguir para controlar a posição do eixo do motor com um **controlador proporcional**

```
//Incluindo biblioteca para leitura do encoder
#include<Encoder.h>
Encoder meuEncoder(2, 3);

//Definindo variaveis
double theta = 0.0;
double tempo = 0.0, erro = 0.0, u;
long contEnc = 0.0;
double thetaRef = 0.0;

//Definindo ganho do controlador proporcional
double Kp = XXXXXXXXXXXX;

void setup() {
    //Inicializando comunicacao serial
    Serial.begin(115200);
```

```
    //Definindo Entradas da ponte H
    pinMode(5,OUTPUT);
    pinMode(7,OUTPUT);
    pinMode(8,OUTPUT);
}

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

    //Calculando theta a partir da leitura do encoder
    theta = contEnc*2.0*3.14/(334.0*4.0);

    //Determinando tempo atual
    tempo = micros();

    //Degrau de referencia e aplicado entre 5 e 6s
    if(tempo/1000000 > 5 ){
        thetaRef = XXXX;
    }
    //Obtendo erro de rastreamento
    erro = XXXXXXXXXXXXX;

    //Calculando controle
    u = XXXXX;

    //Alterando sentido de giro
    if (u >= 0){
        //sentido horario
        digitalWrite(7,HIGH);
        digitalWrite(8,LOW);
    }
    else{
        //sentido anti-horario
        digitalWrite(7,LOW);
        digitalWrite(8,HIGH);
        u = -u;
    }

    //Saturando na faixa linear
    u = min(u,65.0); //limitando superiormente
    u = max(u,0.0); //limitando inferiormente
}
```

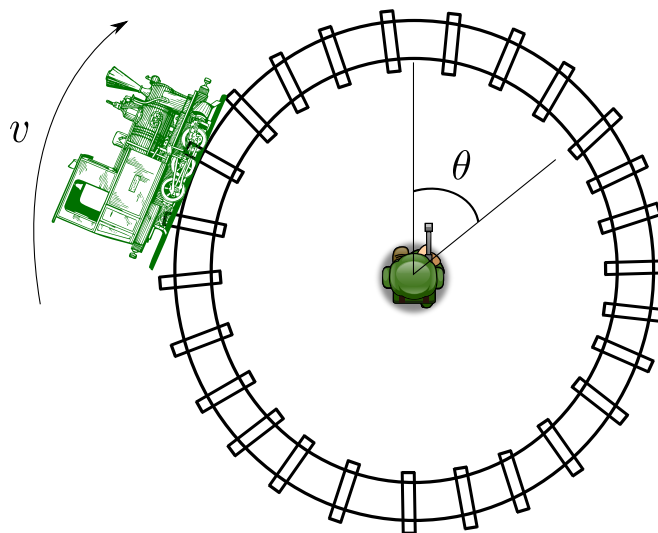
```
//Aplicando controle a planta
analogWrite(5,(u + 35.0)*255.0/100.0);

//Imprimindo os valores na porta serial
Serial.print(theta*180.0/3.14);
Serial.print(" ");
Serial.print(u);
Serial.print(" ");
Serial.println(tempo/1000000);
}
```

- Aplique um degrau de 2,0944 radianos (120°) na referência. Salve os dados em um arquivo .txt e analise o comportamento da planta por meio de um gráfico

5.2 Desafio valendo 15 pontos extras na prova de pior nota

- Considere que o trem se move em um percurso circular com **velocidade constante** v e que há um soldado posicionado no centro do círculo, conforme ilustrado seguir



- Utilizando os conceitos e os programas vistos nessa aula e no curso teórico, empregue um controlador para que a posição do trem seja rastreada pelo soldado (que é representado pelo motor)

Dica: Talvez seja necessário adotar ganhos elevados para o controlador

- A ideia correta para resolução desse problema vale 5 pontos e a implementação da ideia os outros 10.
- Para uma performance adequada, talvez seja necessário modificar o controlador (aumentar ganho)