



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FEMEC 42060

CONTROLE DE SISTEMAS LINEARES

Relatório de Laboratório 2 - Malha Aberta *vs* Malha Fechada

Prof. Pedro Augusto

19 de maio de 2022

1 Objetivos

Neste laboratório comparar-se-ão as estruturas de controle em malha aberta e em malha fechada.

2 Introdução

Controlar um sistema pode ser definindo como impor um comportamento desejado a um processo. Tipicamente isso envolve guiar a saída para o entorno de uma referência com determinados requisitos de velocidade e precisão. Com esse propósito, é possível adotar estruturas de controle em malha aberta (MA) ou em malha fechada (MF).

Em MA, manipula-se diretamente a entrada na planta para resultar na saída desejada. Para que isso funcione adequadamente, a relação entrada-saída deve ser conhecida com precisão, não podendo haver descasamentos de modelo ou perturbações externas. Um diagrama de blocos do controle em MA é mostrado na Figura 1.

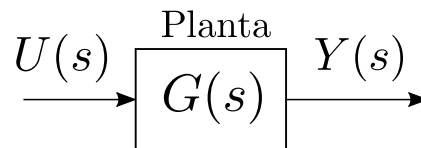


Figura 1: Estrutura de controle em malha aberta.

Algumas vantagens dessa estrutura são simplicidade de construção e manutenção, implementação barata e conveniência em situações nas quais não se é possível medir a saída do processo. Em contrapartida, os efeitos de perturbações ou descasamentos de modelo afetam diretamente a saída da planta. Mais ainda, para manter um desempenho adequado, é possível que uma recalibração frequente seja necessária.

Já em MF (Figura 2) as informações da saída são levadas em consideração no cálculo da entrada (realimentação). Em malha fechada, a variável manipulada pelo operador é o valor desejado para a saída. O controlador é responsável por calcular a entrada de acordo com o erro de rastreamento. Com efeito, tem-se uma maior robustez a perturbações externas e descasamentos de modelo. Por outro lado, o custo de implementação desse controlador é maior, bem como a complexidade de manutenção.

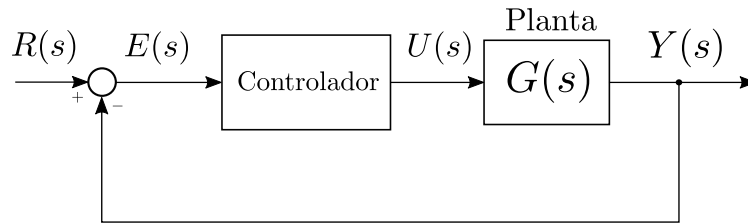


Figura 2: Estrutura de controle em malha fechada.

No que se segue, as estruturas de controle em MA e MF serão comparadas experimentalmente.

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

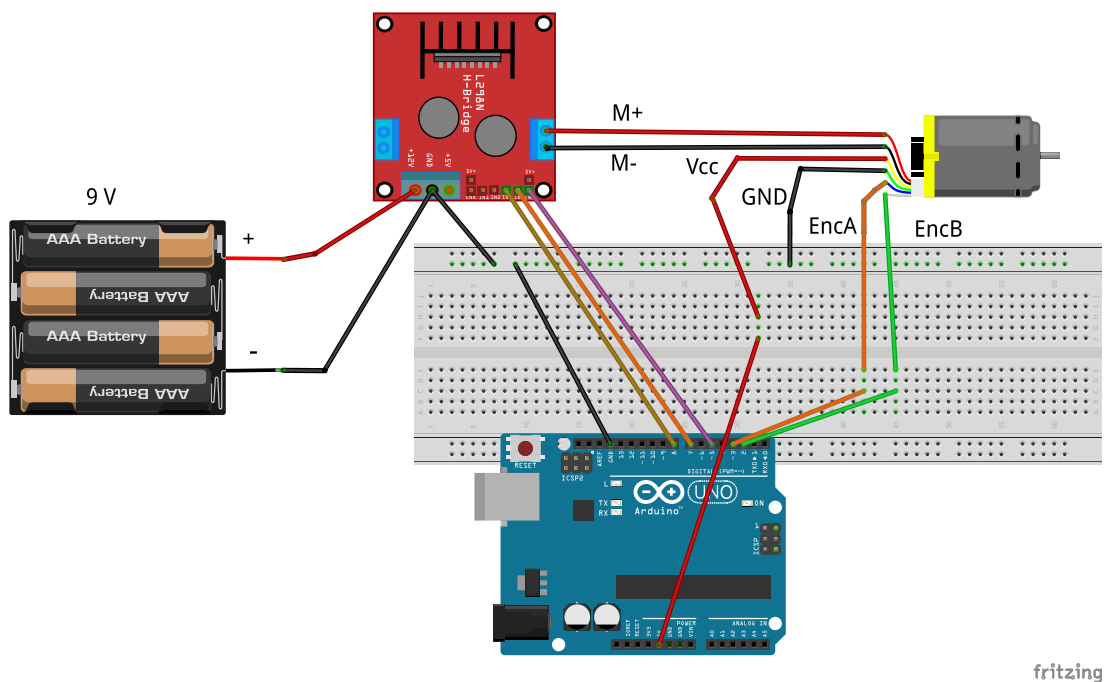
Na sequência circuitos eletrônicos e códigos para implementação de controladores em MA e MF serão mostrados.

4.1 Controle de sistemas em malha aberta

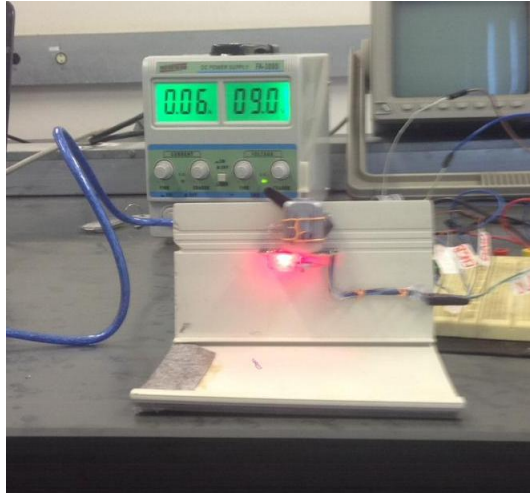
Neste item, variar-se-á a entrada da planta (ciclo ativo de onda PWM) e monitorar-se-á a velocidade do motor. A partir dessas informações, um sistema de controlador em MA será implementado.

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

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



- Posicione o motor na posição indicada na Figura 4.1



- Utilize o código abaixo para variar o *duty cycle* de entrada e medir a velocidade do motor

```
//Incluindo biblioteca para leitura do encoder
#include <Encoder.h>

//Definindo objeto meuEncoder
Encoder meuEncoder(2, 3);

//Definindo outras variaveis uteis
double velAng, theta = 0.0, thetaAnt = 0.0, tempo1 = 0.0,
      tempo2 = 0.0, dt, u;
long contEnc = 0.0;

//variaveis da media movel
double vel1 = 0.0, vel2 = 0.0, vel3 = 0.0, mediavel;
void setup() {

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

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

    pinMode(7,OUTPUT); //sentido de giro
    pinMode(8,OUTPUT); //sentido de giro
```

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

void loop() {
    //Salvando valores anteriores de tempo e posicao
    tempo1 = tempo2;
    thetaAnt = theta;

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

    //Calculando theta a partir da leitura do encoder
    theta = contEnc*360/(334*4); //resolucao do encoder e
    334

    //Determinando tempo atual
    tempo2 = micros();

    //Calculando diferenca de tempo ente instante atual e
    instante da ultima leitura
    dt = tempo2 - tempo1; //em micro s
    //Calculando velocidade angular
    velAng = (theta - thetaAnt)/dt*1000000*0.1667; //em rpm

    //calculando media movel dos tres ultimos valores
    vel1 = vel2;
    vel2 = vel3;
    vel3 = velAng;
    mediavel = (vel1 + vel2 + vel3)/3.0;

    //Variando duty cycle
    u = 0.0;
    analogWrite(5,255.0*u/100.0);

    //Imprimindo na porta serial
    Serial.print(mediavel);
    Serial.print(" ");
    Serial.print(u);
    Serial.print(" ");
    Serial.println(tempo2/1000000); //em s
}
```

- Modifique o código acima para preencher a tabela abaixo.

Nota 1: os valores de velocidade podem ser obtidos monitorando a porta serial: “Ferramentas”->“Monitor serial” ou Ctrl+Shift+M, ou “Ferramentas”->“*Plotter* serial” ou Ctrl+Shift+L

Tabela 1: Relação entre *duty cycle* e temperatura

| <i>duty cycle</i> % | <i>y</i> (rpm) |
|---------------------|----------------|
| 0 | 0 |
| 25 | 0 |
| 35 | 825 |
| 50 | 2345 |
| 75 | 3790 |
| 85 | 4180 |

Nota 2: Em valores de *duty cycle* muito baixos, pode ser necessário partir manualmente o motor devido à presença de atrito seco (dinâmica não linear que negligenciaremos)

- Construa um gráfico entre velocidade (rpm) e *duty cycle* (%) e determine a equação que relaciona essas grandezas (**regressão linear**)
- Utilizando a relação velocidade-*duty cycle* obtida, complete a tabela abaixo

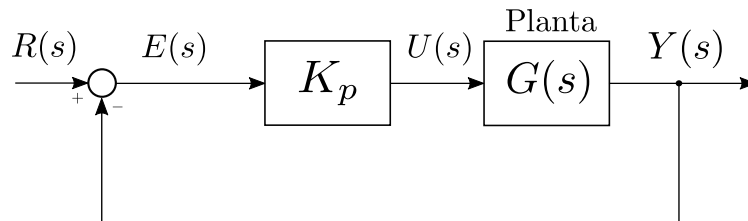
Tabela 2: Desempenho de sistema de controle em MA

| Referência (rpm) | <i>duty cycle</i> (%) | <i>y</i> (rpm) - valor medido |
|------------------|-----------------------|-------------------------------|
| 1000 | 30 | 0 |
| 1500 | 38,93 | 1400 |
| 4000 | 83,57 | 4130 |

- Salve os dados para responder à Seção 1 do relatório

4.2 Controlador proporcional

- Altere o código apresentado nessa seção para implementar controlador proporcional em MF conforme ilustrado na figura a seguir.



```

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

//Definindo outras variaveis uteis
double velAng, theta = 0.0, thetaAnt = 0.0, tempo1 = 0.0,
      tempo2 = 0.0, dt, u;
long contEnc = 0.0;

// Variaveis da media movel
double vel1 = 0.0, vel2 = 0.0, vel3 = 0.0, mediavel;

// Referencia
double Kp, erro = 0.0, ref = 0.0;

// Integral do erro
double erro_int;

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

    //Definindo Entradas da ponte H
    pinMode(5,OUTPUT); //velocidade de giro
    pinMode(7,OUTPUT); //sentido de giro
    pinMode(8,OUTPUT); //sentido de giro

    //sentido horario

```



```
    digitalWrite(7,HIGH); digitalWrite(8,LOW);
}

void loop() {
    //Salvando valores anteriores de tempo e posicao
    tempo1 = tempo2;
    thetaAnt = theta;

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

    //Calculando theta a partir da leitura do encoder
    theta = contEnc*360/(334*4);

    //Determinando tempo atual
    tempo2 = micros();

    //Calculando diferenca de tempo ente instante atual e
    // instante da ultima leitura
    dt = tempo2 - tempo1;//em micro s
    //Calculando velocidade angular
    velAng = (theta - thetaAnt)/dt*1000000*0.1667;//em rpm

    //calculando media movel dos tres ultimos valores
    vel1 = vel2; vel2 = vel3; vel3 = velAng;
    mediavel = (vel1 + vel2 + vel3)/3.0;

    //tempo do degrau
    if (tempo2/1000000 >= XXXX){
        ref = XXXX; //rpm
    }

    // Calculando erro de rastreamento
    erro = xxxxxx;

    // Implementando lei de controle proporcional
    u = xxxxxx;

    // Limitando o controle entre 0 e 90
    u = min(u, 90.0);    u = max(u, 0.0);

    // Enviando controle saturado para porta analogica
    analogWrite(5, 255.0*u/100.0);
}
```

```
//Imprimindo na porta serial
Serial.print(mediavel);
Serial.print(" ");
Serial.print(u);
Serial.print(" ");
Serial.println(tempo2/1000000); //em s
}
```

- Verifique o comportamento da planta para um degrau na referência de 1000 **rpm** após 2 s com $K_p = 0,1$, $K_p = 0,25$, $K_p = 0,5$ e $K_p = 1$

- Para cada ensaio, salve os dados na porta serial (“Ferramentas”->“Monitor serial”) em um arquivo .txt

Nota: deve-se copiar todos os dados lidos na porta serial (Ctrl+A e Ctrl+c) e colá-los em um arquivo de .txt (Ctrl+v). **Desconectar o cabo USB de alimentação do Arduino pode ser útil nesse processo**

- Observe os gráficos de velocidade e controle. Com esse propósito, sugere-se utilizar o código abaixo para gerar as imagens

```
clear, close all, clc
%Carregando dados
load('NomeDoArquivo.txt')
Data = NomeDoArquivo;

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

%Ajustando nome dos eixos e tamanho de letra
xlabel('t (s)'); ylabel('y (rpm)');
xlim([0 max(Data(:,3))])

subplot(1,2,2)
% Plotando dados de entrada
plot(Data(:,3), Data(:,2), 'b-', 'LineWidth', 2)
```

```
grid on, hold on

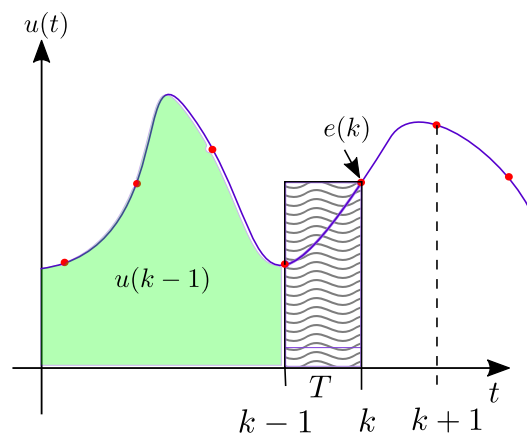
%Ajustando nome dos eixos e tamanho de letra
xlabel('t (s)'); ylabel('u (\%)');
xlim([0 max(Data(:,3))])
```

- Salve os dados para responder à Seção 2 do relatório

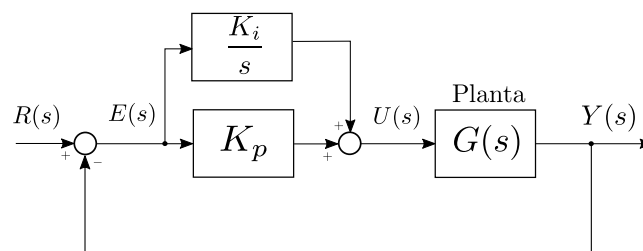
4.3 Exercício

- A integral de um sinal $u(t) = \int_0^\infty e(t)dt$ pode ser realizada a tempo discreto da seguinte forma

$$u(k) = u(k-1) + e(k)\Delta t \quad (1)$$



- Utilizando essa aproximação numérica, implemente a lei de controle ilustrada a seguir



Lembre-se: a variável dt está em microssegundos!!

- Ajuste $K_p = 0,15$ e $K_i = 0,08$
- Altere a referência para **3000 rpm** e verifique a resposta da planta
- Salve os dados para responder à Seção 3 do relatório.