



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FEMEC 42060

CONTROLE DE SISTEMAS LINEARES

---

## Laboratório 5 - Controlador PID - parte 1

---

Prof. Pedro Augusto

4 de julho de 2022

## 1 Objetivos

Na presente aula de laboratório verificar-se-ão os sinais de controle de um PID. Mais ainda, investigar-se-ão os efeitos de cada um desses sinais na saída da planta.

## 2 Introdução

## 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*

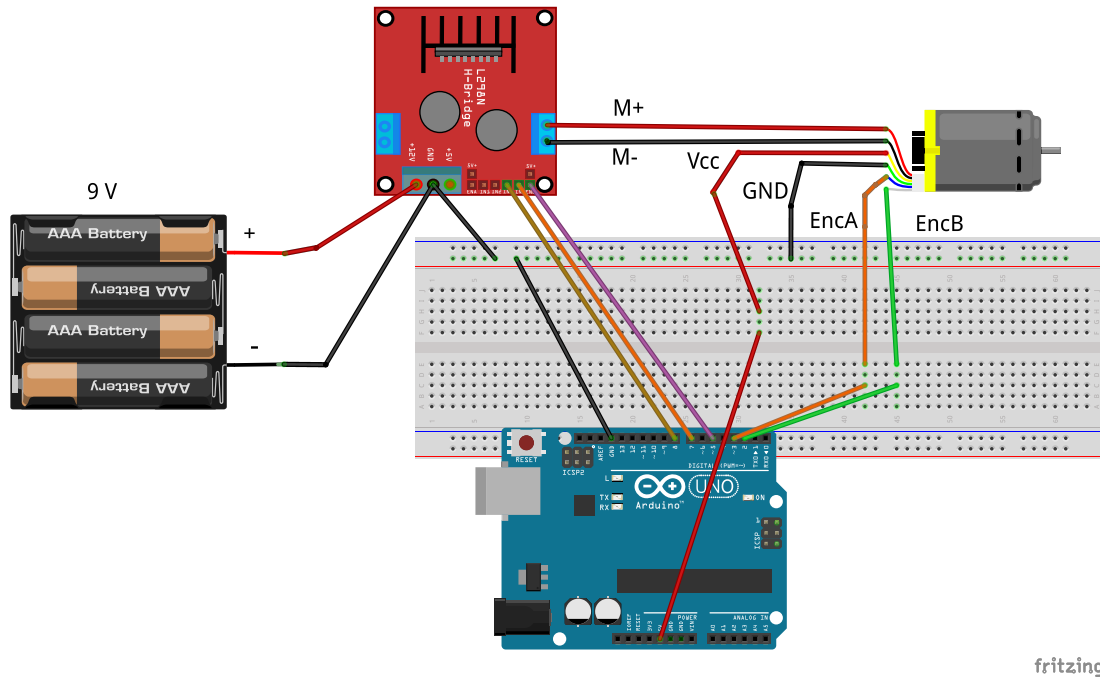
## 4 Procedimento experimental

Primeiramente, os sinais de controle de um PID gerados por uma entrada do tipo degrau serão estudados. Então, os efeitos de cada um desses termos na saída serão avaliados.

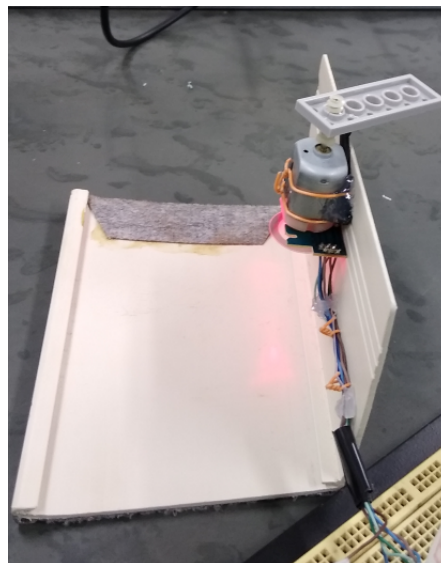
### 4.1 Sinais de controle

- Monte o circuito mostrado a seguir

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



- Posicione o motor na posição indicada na Figura 4.2 e acople a haste no motor



- Utilize o código abaixo para impor uma referência do tipo degrau motor e

monitorar as ações de controle

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

//Definindo outras variaveis uteis
double theta = 0.0, tempo1 = 0.0, tempo2 = 0.0, tempoini ,
      dt, u, up, ui, ud;
double erro = 0.0, erroAnt = 0.0, erroint = 0.0, erroderiv
      = 0.0;
long contEnc = 0.0;

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

//Definindo ganho do controlador proporcional
double Kp = XXXX, Kd = XXXX, Ki = XXXX;

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
}

void loop() {
    tempoini = micros();
    //Salvando valores anteriores de tempo
    tempo1 = tempo2; erroAnt = erro;

    //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); //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

//Degrau de referencia aplicado em 3.0s
if(tempo2/1000000.0 > 3.0 ){
    // Escolhendo referencia para a velocidade
    thetaRef = xxxxxxxxxxxx*3.1415/180.0; //em rad
}

//Obtendo erro de rastreamento
erro = xxxxxxxxxxxx;
//calculando a integral do erro
erroint = erroint + erro*(dt/1000000.0);
// calculando a derivada do erro
erroderiv = (erro - erroAnt)/(dt/1000000.0);

//Calculando acao controle
//proporcional
up = xxxxxxxxxxxx;
//integral
ui = xxxxxxxxxxxx;
//derivativo
ud = xxxxxxxxxxxx;
//controle total
u = xxxxxxxxxxxx;

//Imprimindo os valores de velocidade e tempo na porta
    serial
Serial.print(up);
Serial.print(" ");
Serial.print(ui);
Serial.print(" ");
Serial.print(ud);
Serial.print(" ");
Serial.print(u);
Serial.print(" ");
Serial.println(tempo2/1000000);
    //Esperando proximo instante de amostragem 1ms
while ((micros() - tempoini) < 1000){ }
```

- Verifique o comportamento das ações de controle para  $K_p = 10.0$ ,  $K_i = 8.0$  e  $K_d = 0.05$  considerando uma referência de  $120^\circ$  após 5 s
- Salve os dados em um arquivo .txt e, se possível, utilize o código abaixo obter um gráfico de cada termo de controle

```
%Criando figura
load('Dados.txt'); Data = Dados;
%
% Plotando dados
figure
subplot(2,2,1)
plot(Data(:,5), Data(:,1),'b-','LineWidth',2)

xlabel('t (s)','FontSize',20,'FontName','Times');
ylabel('u_p','FontSize',20,'FontName','Times');
set(gca,'FontSize',20,'FontName','Times')
grid on

subplot(2,2,2)
plot(Data(:,5), Data(:,2),'b-','LineWidth',2)

xlabel('t (s)','FontSize',20,'FontName','Times');
ylabel('u_i','FontSize',20,'FontName','Times');
set(gca,'FontSize',20,'FontName','Times')
grid on

subplot(2,2,3)
plot(Data(:,5), Data(:,3),'b-','LineWidth',2)

xlabel('t (s)','FontSize',20,'FontName','Times');
ylabel('u_d','FontSize',20,'FontName','Times');
set(gca,'FontSize',20,'FontName','Times')
grid on

subplot(2,2,4)
plot(Data(:,5), Data(:,4),'b-','LineWidth',2)

xlabel('t (s)','FontSize',20,'FontName','Times');
ylabel('u','FontSize',20,'FontName','Times');
set(gca,'FontSize',20,'FontName','Times')
grid on
```

- Responda ao item 1 do relatório

## 4.2 Efeito dos termos proporcional, integral e derivativo no comportamento do sistema

- Utilize o código abaixo para controlar a posição do eixo do motor utilizando um PID

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

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

//Definindo outras variaveis uteis
double theta = 0.0, velAng = 0.0, thetaAnt = 0.0, tempo1 = 0.0, tempo2 = 0.0, tempoini, dt;
double u, usat, up, ui, ud;
double erro = 0.0, erroint = 0.0, erroderiv = 0.0, erroAnt = 0.0;
long contEnc = 0.0;

//Referencia
double thetaRef = 0.0;

//Definindo ganho do controlador proporcional
double Kp = XXXXX, Kd = XXXXX, Ki = XXXXX;

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
}

void loop() {
    tempoini = micros();
    //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*2.0*3.14/(334.0*4.0); //em rad

//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;//em rad/s

//Degrau de referencia aplicado em 3.0s
if(tempo2/1000000.0 > 3.0 ){
    // Escolhendo referencia para a velocidade
    thetaRef = XXXXXXXXXXXXXXXX*3.1415/180.0; //em rad
}

//Obtendo erro de rastreamento
erro = XXXXXXXXXXXXXXXX;
//calculando a integral do erro
erroint = erroint + erro*(dt/1000000.0);
// calculando a derivada do erro
erroderiv = (erro - erroAnt)/(dt/1000000.0);

//Calculando acao controle
//proporcional
up = XXXXXXXXXXXXXXXX;
//integral
ui = XXXXXXXXXXXXXXXX;
//derivativo
ud = XXXXXXXXXXXXXXXX;
//controle total
u = XXXXXXXXXXXXXXXX;

//Alterando sentido de giro de acordo com o sinal do
    controle
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
    usat = min(u,65.0); //limitando superiormente
    usat = max(usat,0); //limitando inferiormente

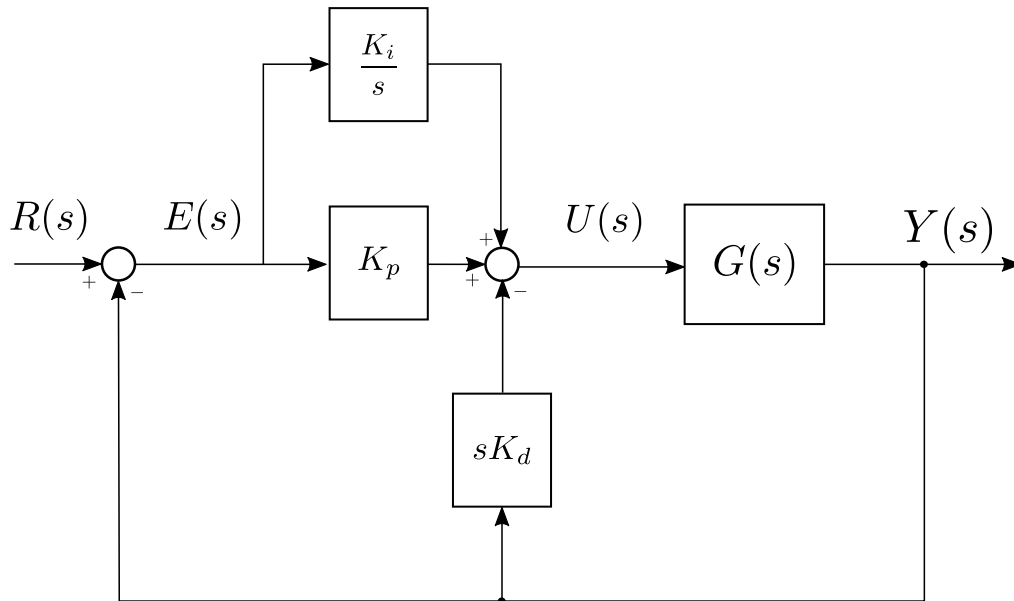
    //Aplicando controle a planta
    analogWrite(5,(usat+35)*255.0/100.0);

    //Imprimindo os valores de velocidade e tempo na porta
    serial
    Serial.print(theta*180.0/3.1415);
    Serial.print(" ");
    Serial.print(u);
    Serial.print(" ");
    Serial.print(usat);
    Serial.print(" ");
    Serial.println(tempo2/1000000.0);

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

- Controlador P: Verifique o comportamento da saída e da entrada para um degrau de  $120^\circ$  na referência com  $K_i = K_d = 0$  e para  $K_p \in \{5 \ 15 \ 30\}$
- Controlador PI: Verifique o comportamento da saída e da entrada para um degrau de  $120^\circ$  na referência com  $K_d = 0$ ,  $K_p = 10$  e  $K_i \in \{10 \ 30 \ 100\}$
- Controlador PID: Verifique o comportamento da saída e da entrada para um degrau de  $120^\circ$  na referência com  $K_i = 8.0$ ,  $K_p = 10.0$  e  $K_d \in \{0.001 \ 0.01 \ 1.0\}$

- Controlador PI+D ou PI+V: Mova a ação derivativa para o ramo de realimentação, conforme ilustrado abaixo, e verifique o comportamento da saída e da entrada para  $K_i = 8.0$ ,  $K_p = 10.0$  e  $K_d = 1.0$ . Nota: **Será necessário modificar o código anterior**



- Responda ao item 2 do relatório. Para isso, o código a seguir pode ser útil para comparar até três sinais diferentes

```
clear all; close all; clc
%Carregando dados do arquivo 1
load('DadosKi10.txt')
Data1 = DadosKi10;

%Carregando dados do arquivo 2
load('DadosKi30.txt')
Data2 = DadosKi30;

%Carregando dados do arquivo 3
load('DadosKi100.txt')
Data3 = DadosKi100;

%
% Plotando dados de posicao
subplot(1,2,1)
```

```
plot(Data1(:,4), Data1(:,1), 'b-', 'LineWidth', 2)
hold on, grid on
plot(Data2(:,4), Data2(:,1), 'r--', 'LineWidth', 2)
plot(Data3(:,4), Data3(:,1), 'm-.', 'LineWidth', 2)

xlabel('t (s)', 'FontSize', 20, 'FontName', 'Times');
ylabel('y (\circ)', 'FontSize', 20, 'FontName', 'Times');
legend('Dados1', 'Dados2', 'Dados3', 'location', 'southeast')
set(gca, 'FontSize', 20, 'FontName', 'Times')

%Limites do grafico - MUDAR CONFORME O ITEM
ylim([0 250])
xlim([3 4])

subplot(1,2,2)
plot(Data1(:,4), Data1(:,3), 'b-', 'LineWidth', 2)
hold on, grid on
plot(Data2(:,4), Data2(:,3), 'r--', 'LineWidth', 2)
plot(Data3(:,4), Data3(:,3), 'm-.', 'LineWidth', 2)

xlabel('t (s)', 'FontSize', 20, 'FontName', 'Times');
ylabel('u_{sat} (%)', 'FontSize', 20, 'FontName', 'Times');
legend('Dados1', 'Dados2', 'Dados3', 'Location', 'NorthWest'
)
set(gca, 'FontSize', 20, 'FontName', 'Times')

%Limites do grafico - MUDAR CONFORME O SINAL
ylim([0 65])
xlim([3 4])
```