

Aula 8 - Laboratório de Controle - 2023/1

Modelagem do motor CC e teste de um controlador proporcional

Nomes: João Gabriel, Lucas Tessari

```
if ~exist('obj')
    z=seriallist;
    comPort=z{length(z)};
    obj=serial(comPort, 'BaudRate', 9600);
    obj.Terminator='CR';
    fopen(obj);
end
datetime
```

```
ans = datetime
    10-May-2016 07:18:37
```

Atividade 1: Obtenção da resposta em malha aberta do motor CC.

A seguir aplica-se uma sequência de degraus em malha aberta que faz o motor variar sua velocidade em toda faixa de operação.

```
Ts=20;
Tempo=1;
U0=[50 90 130 170 210 250];
[y,t, yr, tau] = arduino_coleta_motor(obj,U0,Tempo);
```

Undefined function 'arduino_coleta_motor' for input arguments of type 'serial'.

```
seta_saida_motor(obj,0);
figure;
plot(y);title('Figura 1. Resposta para a aplicação de degraus de mesma amplitude em MA');
xlabel('Amostras');ylabel('RPM')
u0=ones(50,1);
U=[];
for i=1:6
    U=[U;u0*U0(i)];
end
dat0=iddata(y(45:end)-mean(y(43:47)),U(45:end)-U(45),0.02);
figure;
plot(U);title('Figura 2. Degraus aplicados em MA');
xlabel('Amostras');ylabel('PWM')
```

A Tabela 1 resume os ganhos e as constantes de tempo de cada resposta ao degrau, do instante 50 ao 300.

O ganho é calculado pela variação da rotação dividida por 40, que é a amplitude do degrau.

A constante de tempo é obtida verificando o tempo necessário para a saída atingir 63% do valor do regime.

```

RPM=yr(1:5); % velocidades em RPM
RPM0=RPM;
U00=U0(1:5);
ganhos=diff(yr)/40'; % ganhos
taus=tau(2:6)'; % constantes de tempo
Tabela1=table(RPM, ganhos, taus)

```

1.1 Como se comporta o ganho para diferentes velocidades do motor? Como isso pode ser observado na Figura 1?

1.2 Como se comporta a constante de tempo para diferentes velocidades do motor?

Atividade 2: Análise do comportamento do motor CC para uma velocidade de operação.

Escolha a velocidade de operação do motor usando a Tabela 1 (valores entre 1000 e 3400rpm). A função `interp1` calculará por interpolação o valor PWM que deve ser aplicado para obter aproximadamente esta velocidade.

O objetivo agora é verificar o comportamento da velocidade em torno de uma região de operação, definida pela velocidade igual a `Ref`.

Pequenas variações do sinal PWM são agora aplicados, mantendo a rotação próxima à escolhida (`Ref`).

```

Ref=0; % Escolha um valor de referência para a velocidade olhando a Tabela 1
Tempo=2;
r0=floor(interp1(RPM0,U0(1:5),Ref));
d1=[0 0 5 10 5 8];
U20=r0+d1;
u0=ones(100,1);
U2=[];
for i=1:6
    U2=[U2;u0*U20(i)];
end
[y,t,yr,tau] = arduino_coleta_motor(obj,U20,Tempo);
seta_saida_motor(obj,0);

U2=U2(100:end);
Y2=y(100:end);
t=[0:(length(Y2)-1)]*Ts/1000;

y0=mean(Y2(1:50));
yn=Y2-y0;
dat1=iddata(yn,U2-U2(1),Ts/1000); % Dados do ponto de operação escolhido (Ref)

figure;

```

```
plot(t,Y2);title('Figura 3. Resposta para a região de operação escolhida');
xlabel('Amostras');
figure;
plot(t,U2);title('Figura 4. Degraus aplicados');
xlabel('Amostras');
```

```
RPM=yr(2:5);
ganhos=abs(diff(yr(2:6)))./[d1(3:6)]';
Tabela2=table(RPM, ganhos)
```

2.1 Observe os ganhos ($\frac{RPM}{PWM}$) para este caso, comparando com a Tabela 1.

Atividade 3. Obtenção do modelo do motor para a região de operação escolhida

Primeiro é feita a preparação dos dados, subtraindo o valor inicial para que comecem em zero, gerando dat0 e dat1.

Para obter o modelo com o comando procest, usa-se os comandos

g1=procest(data,'P1') para modelo de ordem 1

e

g2=procest(data,'P2') para o de ordem 2 .

Adicionando D estima-se também o atraso, se houver (exemplo, 'P1D').

Também pode-se estimar um modelo discreto ARX (AutoRegressive with eXogeneous Input):

gN=arx(data,[N N d]),

onde N é a ordem do modelo (1,2,3,...) e d é o atraso entre saída e entrada: d=1 significa que não há tempo morto, pois a saída discreta sempre está atrasada pelo menos 1 instante da entrada. Exemplo: g1=arx(data, [1 1 1]).

Caso uma função não dê um bom modelo, deve-se testar com a outra (procest e arx).

A avaliação do modelo é feita com o comando compare(data,m), avaliando o modelo m usando os dados em data.

Também se pode comparar vários modelos ao mesmo tempo, compare(data,m1,m2,m3).

3.1 Estime um modelo de ordem 1 sem atraso usando todos os dados (dat0). Depois, estime outro modelo usando os dados do ponto de operação escolhido (dat1). Compare os dois modelos justificando sua diferença de qualidade (FIT).

```
% comandos para estimar e comparar: usar figure antes de plotar via compare
```

3.2 Repita para um modelo de ordem 2, e verifique se houve um ganho significativo para o FIT.

```
% comandos para estimar e comparar: usar figure antes de plotar via compare
```

Atividade 3: Resposta em malha fechada variando o ganho proporcional.

Vamos analisar a resposta do motor em malha fechada, apenas com um controlador proporcional K_p .

Substitua abaixo $K_{p1}=1/g(0)$ e $K_{p2}=5/g(0)$, sendo $g(0)$ o ganho do modelo estimado.

Os testes são feitos alterando a referência de Ref para Ref+100.

Fique atento ao efeito do ruído no erro em regime!!!

3.1 Avalie a diferença na sobreelevação e no erro em regime para os dois ganhos.

3.2 Baseado nestas respostas, escolha uma especificação de projeto para um controlador para a rotação do motor contendo: sobreelevação, tempo de estabelecimento, erro em regime.

```
Tempo=2;
Ki=0;
Kp1=0; % Calcular
[y,u,t] = arduino_controle_PI_motor(obj,Ref,1, Kp1, Ki);
[y,u,t] = arduino_controle_PI_motor(obj,Ref+100,Tempo, Kp1, Ki);
seta_saida_motor(obj,0);
figure;
plot(t,y);
ss=sprintf('Figura 5. Resposta para ganho k = %.1f',Kp1);
title(ss);
xlabel('Tempo(s)');
ylabel('RPM');

Kp2=0; % Calcular
[y,u,t] = arduino_controle_PI_motor(obj,Ref,1, Kp2, Ki);
[y,u,t] = arduino_controle_PI_motor(obj,Ref+100,Tempo, Kp2, Ki);
seta_saida_motor(obj,0);
figure;
plot(t,y);
```

```
ss=sprintf('Figura 6. Resposta para ganho k = %.1f',Kp2);  
title(ss);  
xlabel('Tempo(s)');  
ylabel('RPM');
```