Aula 3 - Laboratório de Controle - 2021/1

Identificação de modelos a partir de dados

Nome: Lucas Neves Gobbo

Atividade 1: Análise da resposta do motor

O modelo do motor utiizado é mostrado na Figura 1. A entrada é o sinal PWM e a saída agora é a rotação, em RPM. Um sinal de ruído é adicionado a saída.

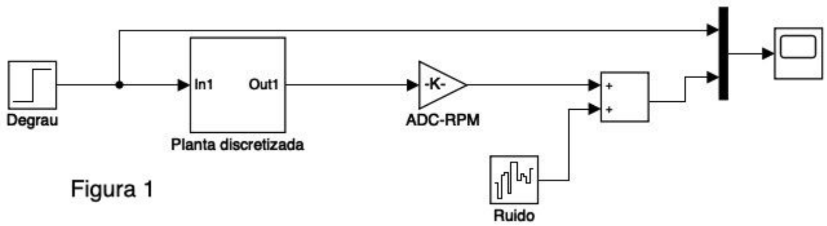
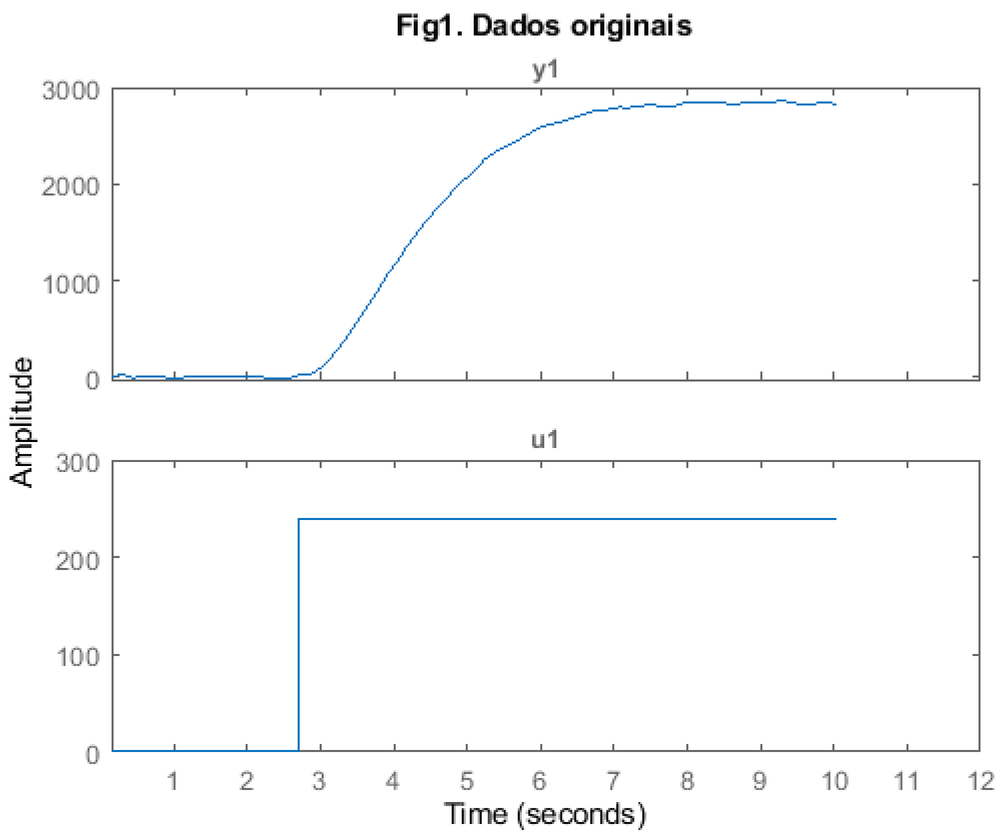


Figura 1

```
Tempo = 10;
arquivo='aula3_2018.slx';
Stepdelay=0.25*Tempo;
I = 5;Turma = 5;
[wn,PWM,rpm_r,Ts]=init(I,Turma);
warning('off','all');
[y,u,t]=simula_slx(arquivo,Tempo);
dat=iddata(y,u,Ts);
plot(dat); title('Fig1. Dados originais')
```



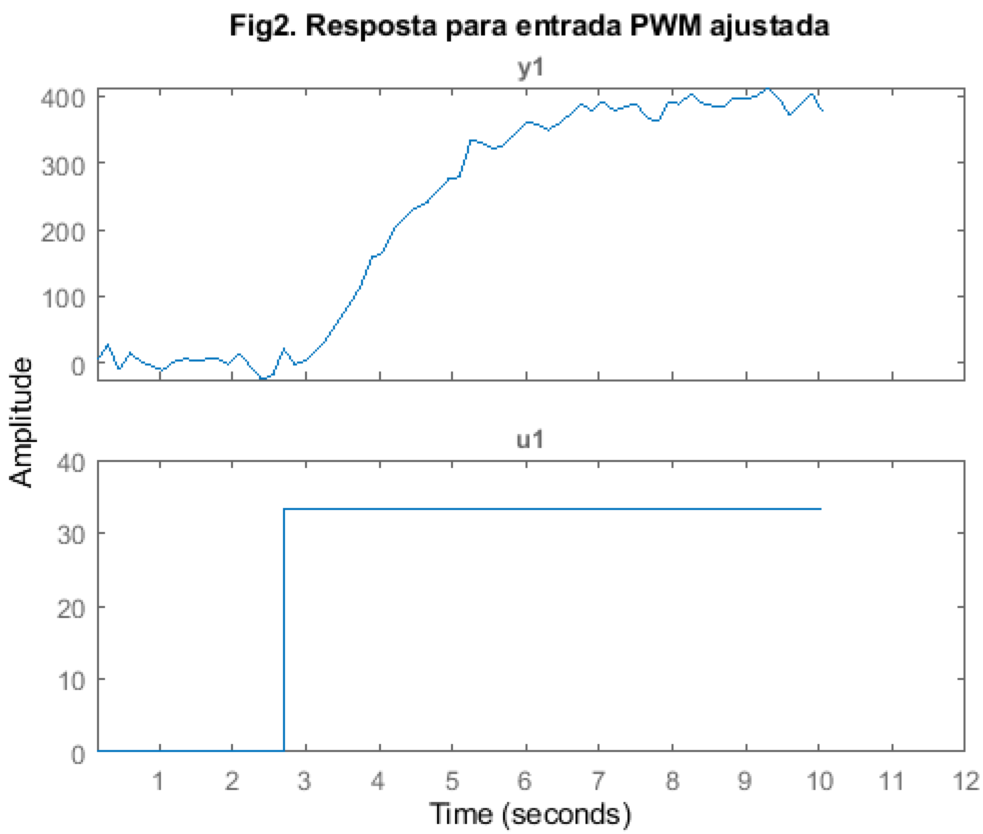
1.1 Explicar o que representam as duas curvas da Figura 1.

A curva de cima mostra a saída do sinal para a entrada do degrau da figura abaixo. Pode-se observar que o sistema demora cerca de 5s para pode chegar no valor de regime. Além disso, vemos o efeito do ruído adicionado à saída do sistema.

1.2 Calcular o valor da entrada PWM para ter em regime uma saída de velocidade = rpm_r. Explicar o cálculo e informar como obter o valor da saída em regime na figura 2.

Para acharmos o valor do PWM, temos que calcular o ganho primeiro, fazendo a relação de saída/entrada com as figuras. 2800/240 = 12. Para ter um rpm_r de 400, conforme solicitado, temos que ter um PWM de 400/12 = 33.3

```
PWM=33.3;
[y,u,t]=simula_slx(arquivo,Tempo);
dat=iddata(y,u,Ts);
plot(dat);title('Fig2. Resposta para entrada PWM ajustada')
```



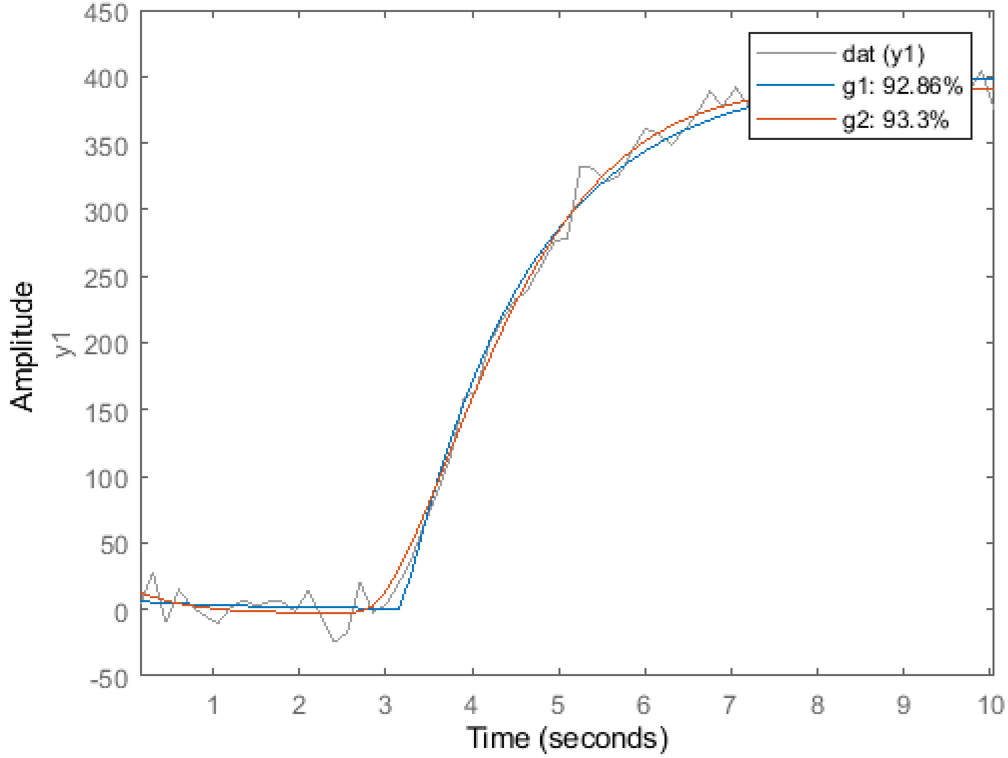
Atividade 2: Identificação dos modelos contínuos de ordem 1 e 2

Ler o apêndice sobre a identificação de funções de transferência e o sobre o índice fit para medir a qualidade do modelo. Depois, use os comandos abaixo para estimar a FT g1 de ordem 1 e a FT g2 de ordem 2.

```
delay=t(sum(y<0.1*y(end)))-Stepdelay;
g1=tfest(dat,1,0,delay);
g2=tfest(dat,2,0);

figure;compare(dat,g1,g2);
title('Fig3. Comparação das FTs estimadas g1 e g2 com os dados originais');
```

Fig3. Comparação das FTs estimadas g1 e g2 com os dados originais



2.1 Explique o comando tfest e analise o FIT de g1 e g2, comparando.

O tfest faz com que consigamos estimar a função de transferência utilizando dados coletados na saída. Ele pode estimar o tempo morto, mas a precisão fica melhor se informarmos. Podemos também informar com quantos pólos queremos que a precisão tenha

Comparando G1 e G2, vemos que o FIT de G2 é maior do que o de G1, portanto, o G2 é um modelo melhor aproximado do real do que o G1. Para este caso, vimos que o de segunda ordem é melhor do que o primeiro.

2.2 Compare esse método de estimar g1 ao utilizado na aula 2 (atividade 2).

Na atividade 2, nós fizemos uma estimativa de um sistema de primeira ordem avaliando a norma euclidiana do erro de dez valores de tempo morto, a partir daí, vemos qual é o melhor tau para aproximar o modelo. Além disso, nós calculamos o ganho fazendo uma relação entre a saída e a entrada. O dessa atividade nós utilizamos o tfest utilizando os dados que nos foram fornecidos, com exceção do delay, e ele nos fornece o valores de K e de tau.

2.3 Compare g1 com a FT G de primeira ordem+tempo morto $G = \frac{e^{ds}K}{\tau s + 1}$ e obtenha o ganho K e a constante de tempo τ.

Analisando e comparando as equações, vemos que K é 12.0821 e tau é 0.6884

2.4 Compare g2 com o protótipo de segunda ordem em malha fechada $M(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + wn^2}$ e obtenha ζ e w_n.

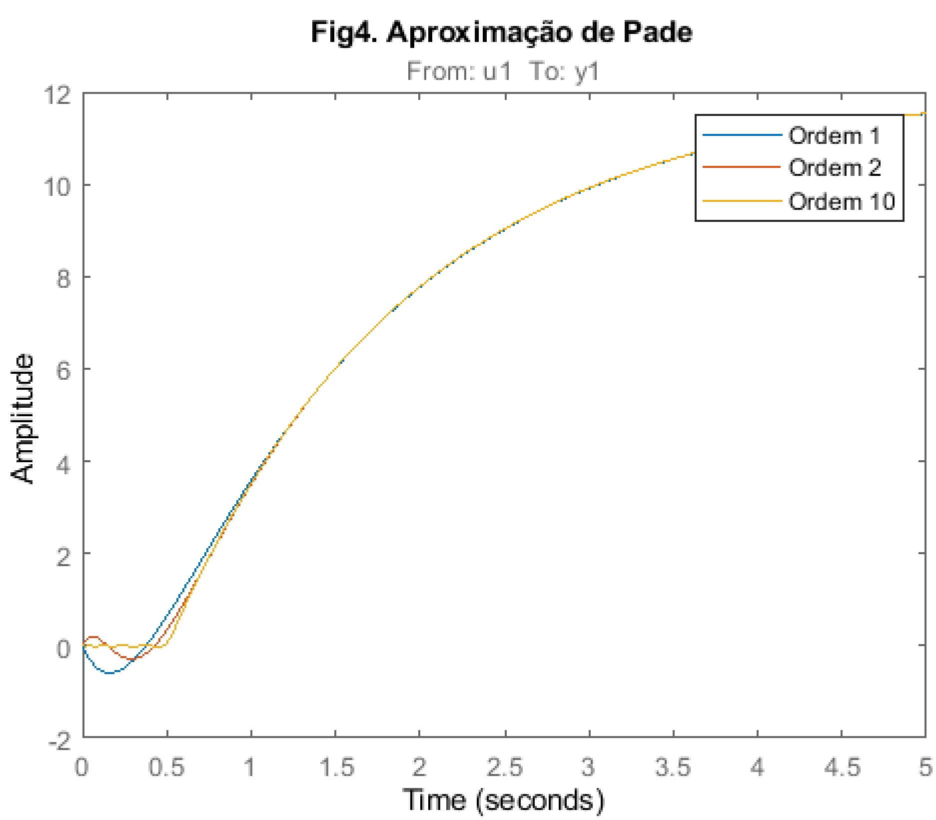
Resposta: w_n = 3.5114e ζ = 0.2606

Atividade 3 - Aproximação de Padé do tempo morto

A aproximação de Padé permite aproximar o atraso e^{-ds} por polos e zeros, e assim obter FTs racionais na presença de atrasos (ver apêndice 2)

No script a seguir, usa-se aproximação de Padé de ordem 1,2,10 para aproximar o tempo morto de g1 por polinômios.

```
g1p=pade(g1,1);
g2p=pade(g1,2);
g10p=pade(g1,10);
figure(7);step(g1p , g2p, g10p, Tempo/2 );title('Fig4. Aproximação de Pade');legend('Ordem 1'
```



3.1 Use a aproximação de Padé para obter uma FT de malha fechada m1 de g1 e obtenha seus polos (ver links de ajuda na aula 2).

```
m1 = feedback(g1p,1);
m2 = feedback(g2p,1);
m10 = feedback(g10p,1);
```

```
pole(m1)
pole(m2)
pole(m10)
```

```
Polos de malha fechada g1p
ans = 2×1 complex
   1.8139 + 5.7213i
   1.8139 - 5.7213i
Polos de malha fechada g2p
ans = 3×1 complex
 -23.6239 + 0.0000i
   1.3088 + 4.0728i
   1.3088 - 4.0728i
Polos de malha fechada g10p
ans = 11×1 complex
10^2 ×
  -1.5109 + 0.0000i
  -0.3117 + 0.7369i
  -0.3117 - 0.7369i
  -0.0531 + 0.4241i
  -0.0531 - 0.4241i
  -0.0247 + 0.2816i
  -0.0247 - 0.2816i
  -0.0126 + 0.1564i
  -0.0126 - 0.1564i
   0.0124 + 0.0403i
      ⋮
```

3.2 Use o comando rlocus para explicar o que ocorre com as raízes de 1+kg1(s)=0 (polos de malha fechada) quando o ganho k varia de zero a infinito (coloque apenas texto e informações do LR, sem comandos ou figuras).

Analisando os rlocus gerados, vemos que nos três casos, quando K vai para o infinito, o sistema fica instável. Vemos também que o modelo mais preciso é o g10p, mas ele possui tantos polos e zeros, que a sua análise fica muito complexa que ganho por causa da aproximação que não compensa.

No desenho com g1p, vemos que se o K for maior que 0.564, o sistema já fica instável. O sistema tem agora dois polos e um zero e por causa disso, temos uma assíntota.

No desenho com g2p, vemos que se o K for maior que 0.427, o sistema já fica instável. O sistema tem agora três polos e dois zeros e por causa disso, temos uma assíntota.

No desenho com g10p, vemos que se o K for maior que 0.383, o sistema já fica instável. O sistema tem agora onze polos e dez zeros e por causa disso, temos uma assíntota.

Conforme já mencionado, fazer a análise do root locus do g10p é tão complicado que o modelo com g2p já é satisfatório para aproximar e fazer uma boa análise.

```
datetime('now')
```

```
ans = datetime
   02-Jul-2021 18:49:08
```

```
pwd
```

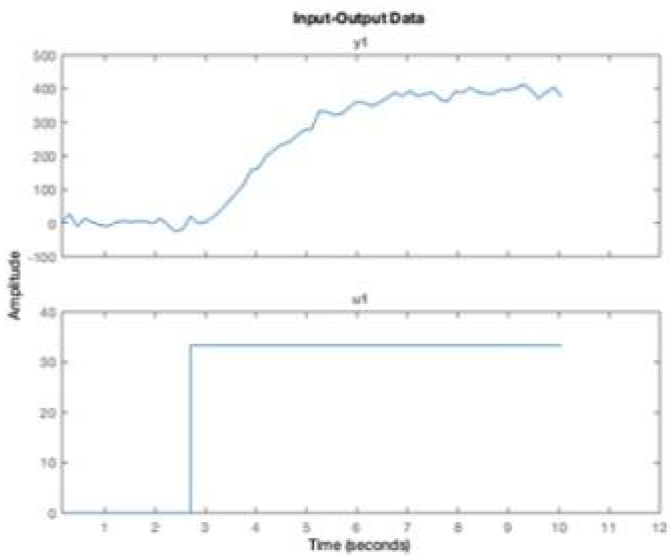
```
ans =
'C:\Users\LucasGobbo\Desktop\LCA\Aula3'
```

Apêndice 1

Identificação de modelos no Matlab

Seja o conjunto de dados de entrada-saída coletados e mostrados abaixo. Os métodos de identificação de sistemas permitem estimar os parâmetros de funções de transferência que aproximem a relação mostrada pelos dados.
A função tfest minimiza o erro do modelo para uma função de transferência com o número de polos e zeros especificados.
Exemplos:

```
g=tfest(dat,2,1); Estima uma FT com 2 polos e 1 zero
g=tfest(dat,1,0,d); Estima uma ft com 1 polo, nenhum zero e atraso=d
O numerador e denominador do modelo são acessados via g.Numerator e g.Denominator
Os dados usados por tfest são colocados na struct dat com o comando
dat=iddata(y,u,Ts);
```



Qualidade do modelo: usa-se a métrica fit, calculada por $fit = 100 * (1 - \frac{\|erro\|}{\|y-\bar{y}\|})$, sendo y a saída e $\bar{y}$ seu valor médio. Um valor de fit próximo de 100 indica que o erro é muito pequeno comparado ao sinal de saída y.

Ao estimar uma FT `g=tfest(dat,2,1),` o valor do FIT de g é obtido via `g.Report.Fit.FitPercent`

Apêndice 2

Aproximação de Padé

Seja a FT de primeira ordem+tempo morto $G(s)=\frac{Ke^{-ds}}{\tau s+1}$. Não é possível obter a FT de malha fechada $M(s)=\frac{G(s)}{1+G(s)}$, devido a presença do atraso no numerador, pois $M(s)$ não é uma FT racional (polinômio no numerador e no denominador).

A aproximação de Padé do atraso é uma solução para o problema. A aproximação de primeira ordem é

$$e^{-ds}\cong\frac{-s+2/d}{s+2/d}$$

e teríamos $G(s)\cong\frac{K(-s+\frac{2}{d})}{(\tau s+1)(s+\frac{2}{d})}$.

No Matlab, basta fazer g1p=pade(g1,N), sendo N a ordem da aproximação. g1p será a aproximação de Padé de ordem N de g1.