

Utilizando o toolbox Ident do Matlab

SEL0328 - Laboratório de Controle de Sistemas

Rafael F. Q. Magossi and Vilma A. Oliveira

Agosto 2016

1 Introdução

O Toolbox de Sistema de IdentificaçãoTM, doravante denominado Ident, fornece funções MATLAB, blocos Simulink®, e um aplicativo para a construção de modelos matemáticos de sistemas dinâmicos a partir de dados de entrada e saída medidos. Ele permite que você criar e usar modelos de sistemas dinâmicos não são facilmente modelados a partir de dados a priori ou especificações. É possível usar os dados no domínio do tempo e no domínio da frequência de entrada-saída para identificar modelos de espaço de estados, funções de transferência de tempo discreto e/ou tempo contínuo, etc. A caixa de ferramentas também fornece algoritmos para estimativa de parâmetros online.

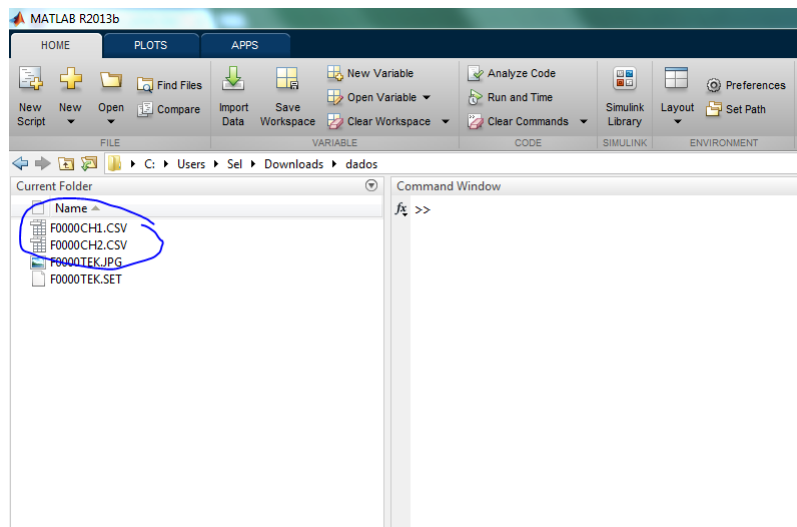
A caixa de ferramentas fornece técnicas de identificação, tais como a máxima verossimilhança, minimização de erros de previsão (PEM), e sistema de identificação de subespaço. Para representar a dinâmica do sistema não-linear, você pode estimar modelos Hammerstein-Wiener e modelos ARX não-lineares com a rede wavelet, partição árvore, e não-linearidades de rede sigmóide. A caixa de ferramentas executa sistema de identificação caixa-cinza para estimar parâmetros de um modelo definido pelo usuário. É possível usar o modelo identificado para a previsão de resposta do sistema e modelagem fábrica em Simulink. A caixa de ferramentas também suporta modelagem de dados de séries temporais e previsão de séries temporais.

2 Obtendo os dados a partir do arquivo .csv

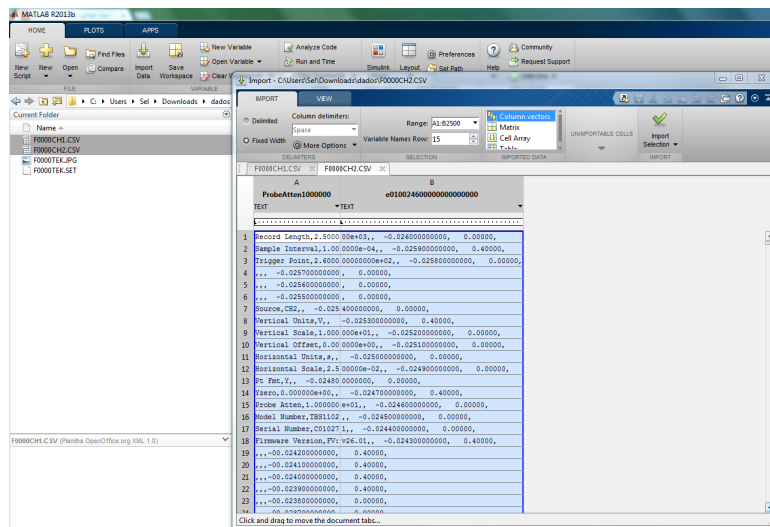
O osciloscópio Tektronix quando salva os arquivos em mídia móvel, usualmente salva: um arquivo .csv por canal (note a terminação CH1 significa canal 1), uma figura da tela do osciloscópio e um arquivo .set. O que nos interessa são os arquivos .csv. Os arquivos Comma-separated values, também conhecido como CSV, são arquivos de formato regulamentado pelo RFC 4180 que faz uma ordenação de bytes ou um formato de terminador de linha. Ele comumente usado em softwares como o Microsoft Excel.

Nome	Data de modificaç...	Tipo	Tamanho
F0000CH1	18/08/2016 21:16	Microsoft Excel Comma Separated Values File	77 KB
F0000TEK	18/08/2016 21:15	Arquivo PostScript encapsulado	30 KB
F0000TEK.SET	18/08/2016 21:17	Arquivo SET	2 KB

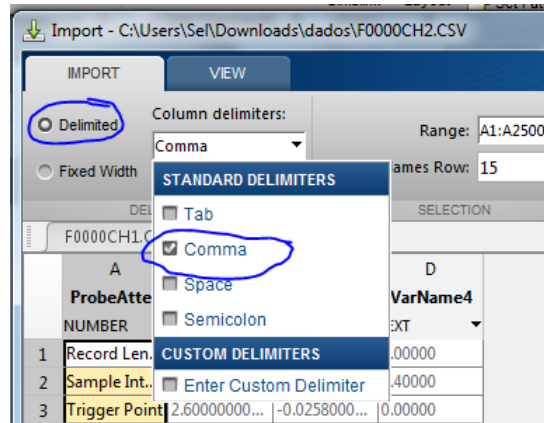
O Matlab conta com um GUI para importar arquivos .csv. Primeiramente, copie e cole seus arquivos para dentro da pasta de trabalho do Matlab.



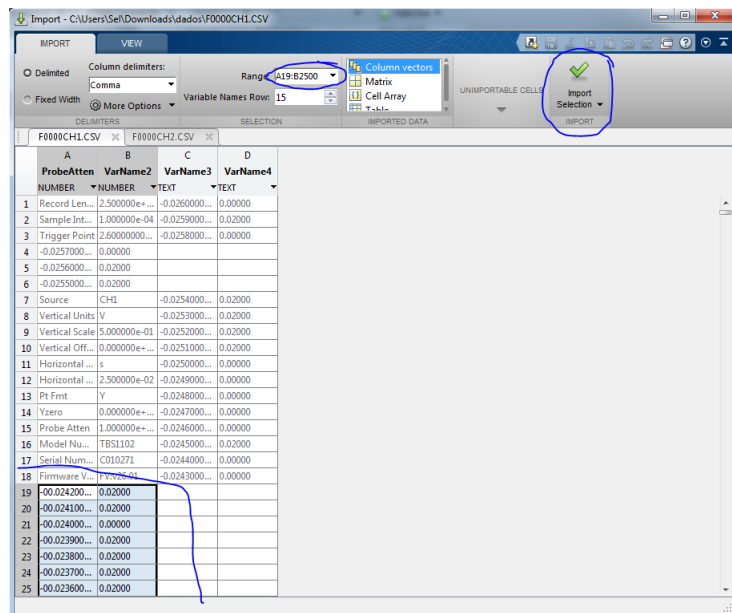
Para usá-lo, clique e arraste o arquivo .csv desejado para o workspace do Matlab. O GUI de importação será iniciado automaticamente.



O osciloscópio gera os dados em forma de texto. Para realizar a importação correta, vamos delimitar o arquivo por vírgulas.

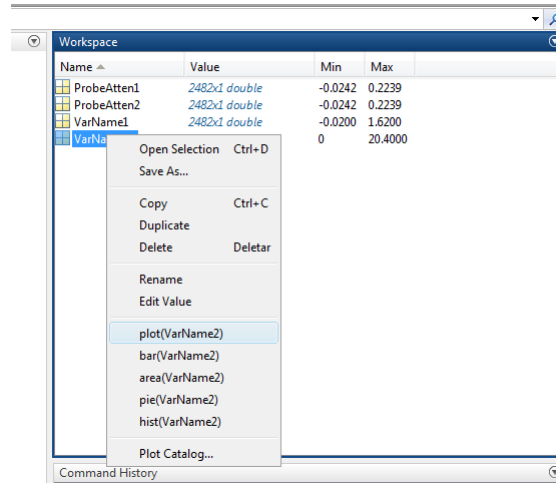


As colunas A e B representam o vetor de tempo e sinal medido. Encontre o início dos dados e selecione os dados dessas colunas até o fim (como no Excel). Caso seja necessário é possível mudar o nome da variável (VarName2 para Canal, por exemplo) clicando no nome abaixo da letra da coluna e renomeando como desejado. Quando finalizar clique em *Import Selection*. Repita o mesmo procedimento para quantos arquivos .csv você possuir.

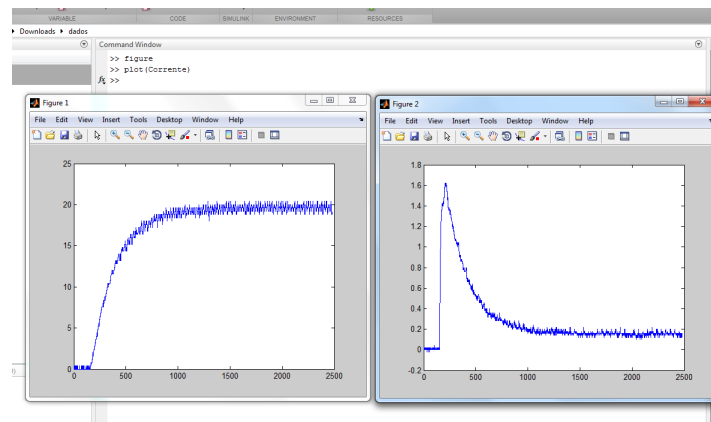


Note que os dados agora se encontram no seu *workspace* do Matlab. Se você

não tem ideia de quem são as variáveis que estão ali, clique com o botão direito do mouse em cima da variável que deseja descobrir o que é e clique em plot.

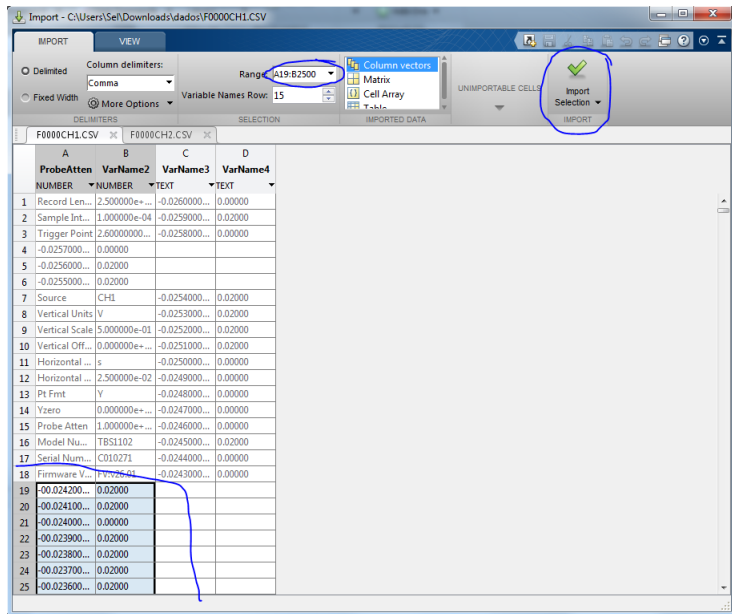


Faça isso para quantas variáveis forem necessárias. E verifique quem é o vetor de tempo e dados. Renomeie as variáveis após descobrir.



2.1 Descobrimdo o Sample Time

Os dados estão digitalizados, portanto, é necessário descobrir qual é o período de amostragem do seu sinal. Para isso, você pode utilizar o GUI de importação do Matlab. Na segunda linha encontra-se a informação sobre o sample time, $1e-4$ segundos.



Esse valor também pode ser descoberto utilizando a diferença entre quaisquer dois pontos do seu vetor de tempo.

```
>> SampleTime = Tempo(1001)-Tempo(1000)

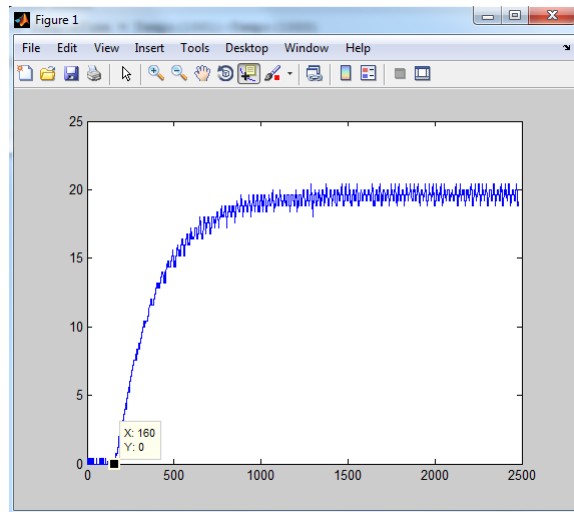
SampleTime =

    1.0000e-04

>> |
```

2.2 Obtendo o sinal de entrada

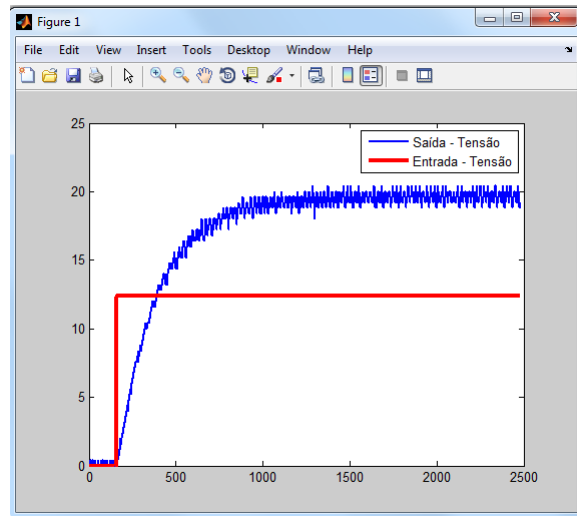
O ident necessitará do vetor de entrada que gerou a resposta de saída. Caso não tenha salvo o vetor de entrada no osciloscópio, será necessário gerar um sinal de entrada aproximado. Primeiramente, com o vetor de dados plotado, adicione um marcador de dados na imagem e tente identificar o início do degrau. Note que no eixo X não é tempo, mas sim posição do dado no vetor. No exemplo abaixo, o degrau se inicia próximo ao dado 160.



Uma vez descoberto isso, vamos gerar um sinal do tipo degrau, com amplitude igual a usada para colher os dados, utilizando a linha de comando abaixo.

```
>> Entrada = [zeros(160,1); 12.4*ones(length(Tensao)-160,1)];
```

Agora plote no mesmo gráfico para ver como ficou a sua entrada com a sua saída de dados.

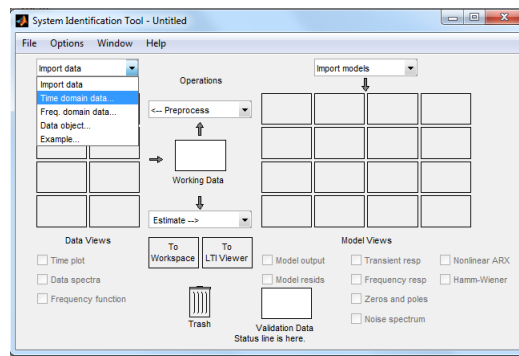


3 Utilizando o Ident

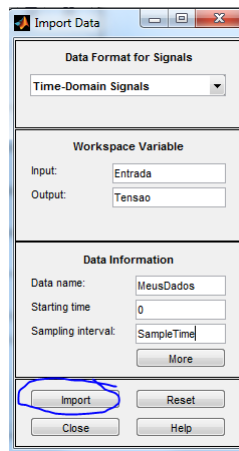
Para começar a utilizar o Ident, vamos primeiramente digitar na linha de comando do Matlab:

```
|> >> ident|
```

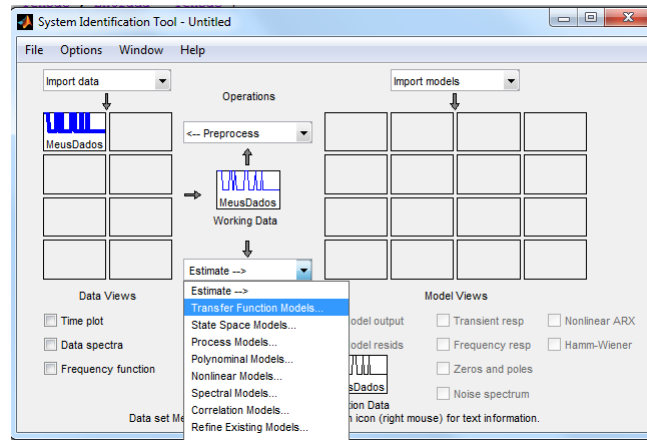
A seguinte tela aparecerá. Vamos importar os dados para trabalhar. Para isso vá em *import data* e depois em *Time domain data*, pois estamos com os dados todos no domínio do tempo.



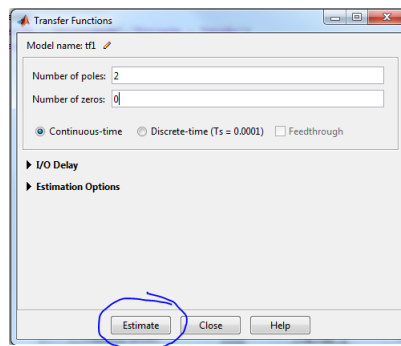
Uma nova janela para importação do dados aparecerá. Configure-a como mostrado. Em input será o vetor de entrada, obtido pelo osciloscópio ou criado como na seção anterior. Em output o vetor de dados de saída (nesse exemplo a tensão de saída do taco gerador de um motor CC). Se desejar troque o nome do seu conjunto de dados. O starting time é 0, pois nesse sistema não há delays. O sampling interval é o sample time que o seu osciloscópio forneceu. Clique em import para finalizar a importação dos dados.



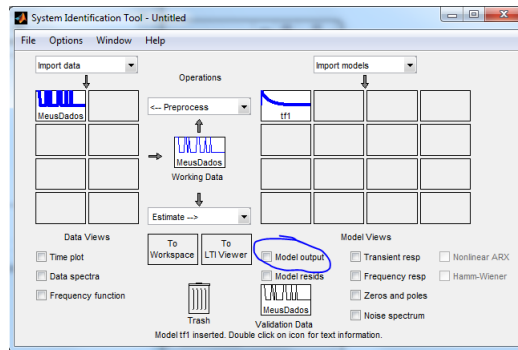
Agora vá em *Estimate* e clique em *Transfer Function Models...*



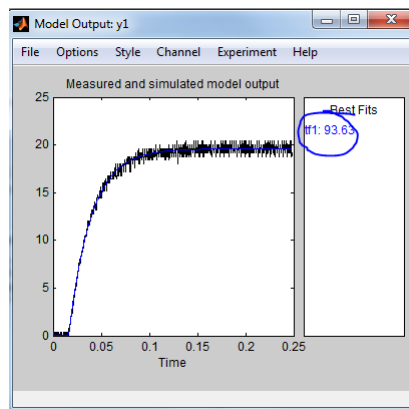
Agora configure quantos polos e quantos zeros você acredita que tenha sua planta. Na opção Continuous-time será estimada uma planta em s . Em Discrete-time será estimada uma planta em z . Nesse caso estamos interessados numa planta em s . Clique em *estimate*.



Depois de concluído a estimação, verifique qual boa foi. Para isso clique em *model output*.



Obteve-se uma estimação com aproximadamente 94% de confiança. Se não estiver satisfeito, repita o processo para diferentes combinações de zeros e polos. Lembre que o aumento da ordem de um sistema desnecessariamente pode lhe causar dores de cabeça ao projetar um controlador eficiente.



Para salvar o resultado clique e arraste *tf1* para *To Workspace*. Note que o resultado no seu *workspace* será um dado do tipo *idtf*.

```
>> tf1

tf1 =

From input "u1" to output "y1":
    3.264e05
-----
s^2 + 4859 s + 2.066e05

Name: tf1
Continuous-time identified transfer function.

Parameterization:
  Number of poles: 2  Number of zeros: 0
  Number of free coefficients: 3
  Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "MeusDados".
Fit to estimation data: 93.63% (simulation focus)
FPE: 0.1357, MSE: 0.1352
```

Para trabalhar somente como função de transferência, utilize o comando abaixo.

```
>> G = tf(tf1)
G =
From input "u1" to output "y1":
      3.264e05
-----+---
s^2 + 4859 s + 2.066e05
Name: tf1
Continuous-time transfer function.
```

Salve e use de acordo com seu interesse. Para salvar use “save G”. Pode usar G como entrada do bloco LTI system no simulink.