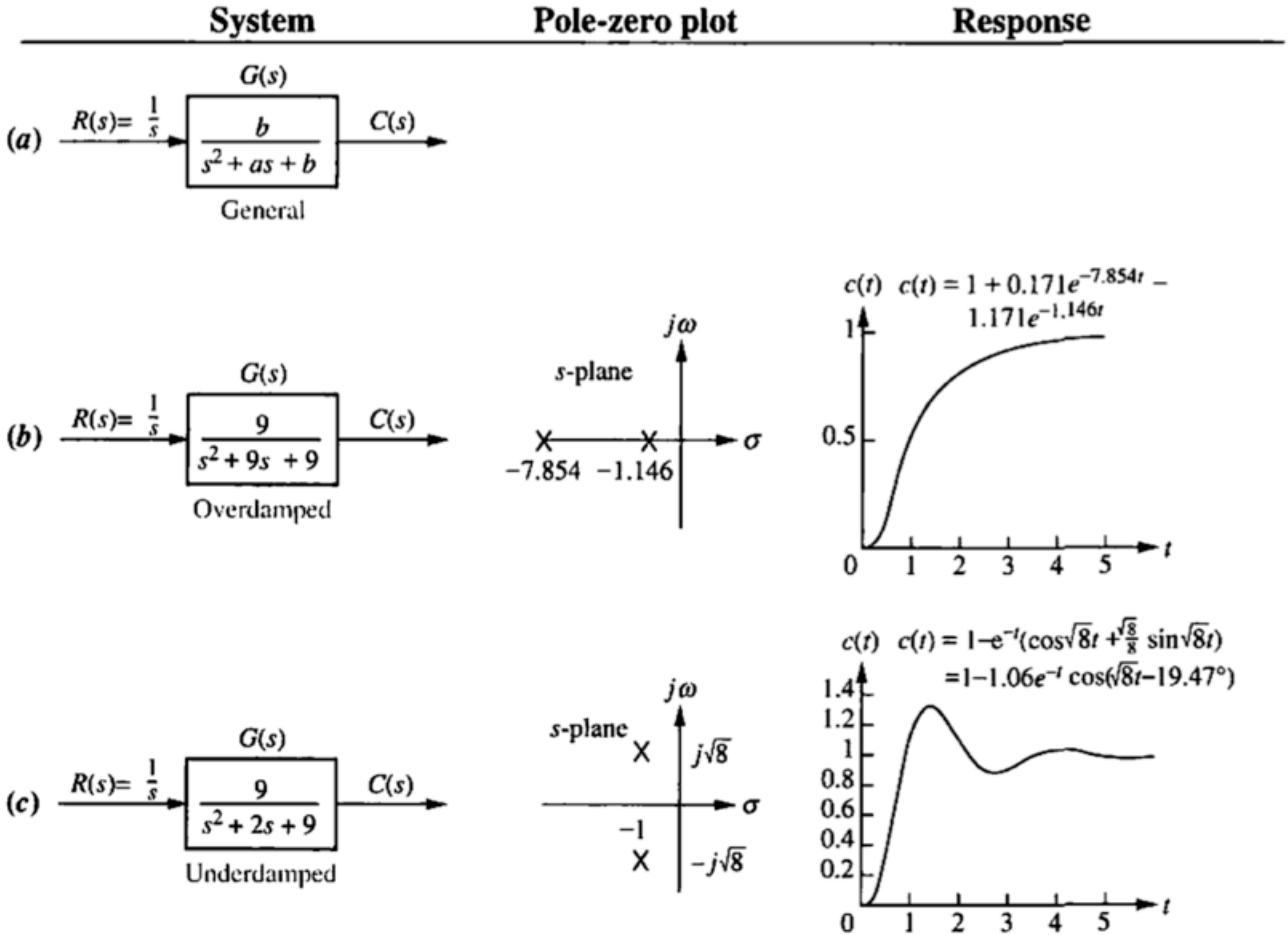
Respostas de Sistemas de 2a-ordem

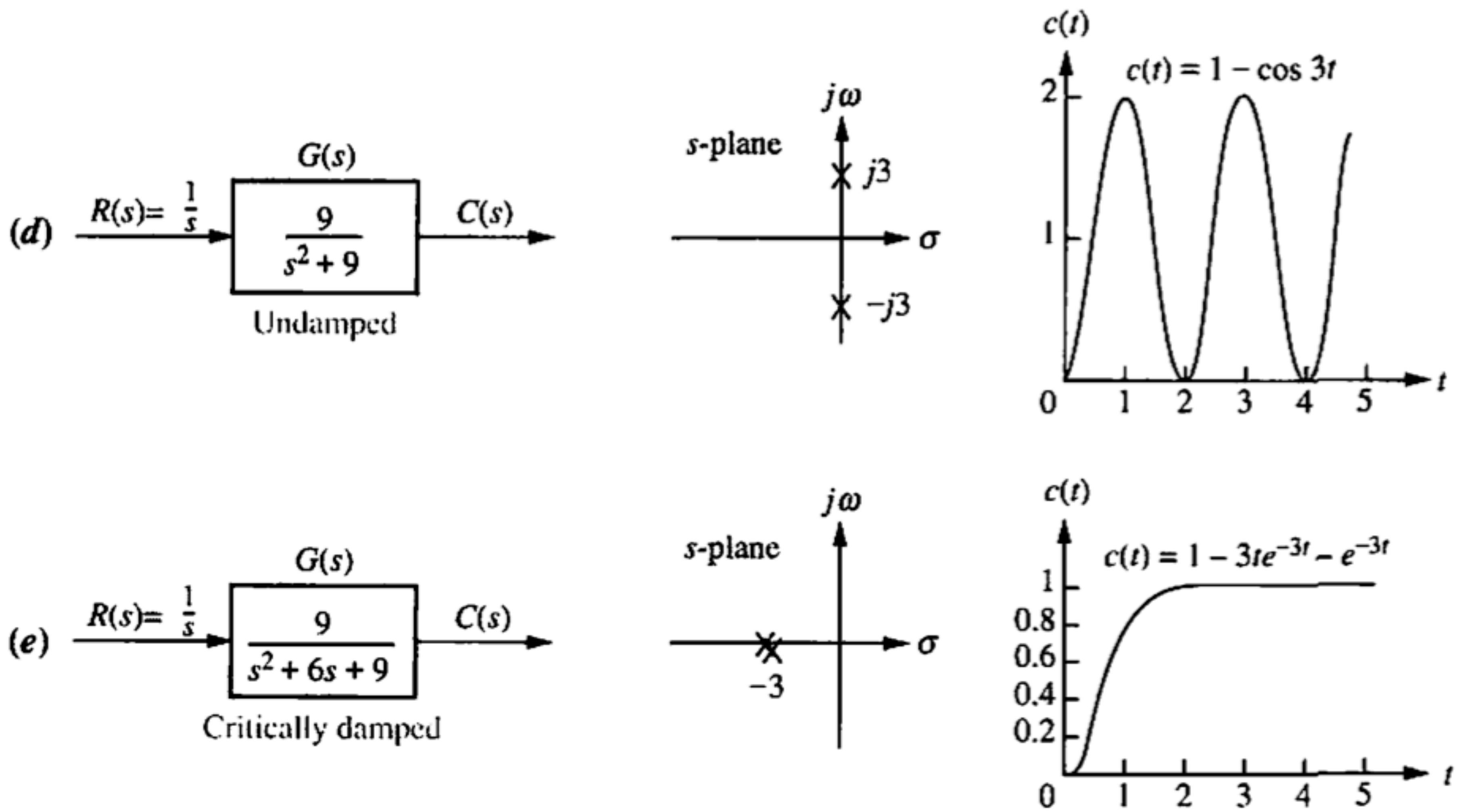
Controle Automático Digital

Prof. Fernando Passold



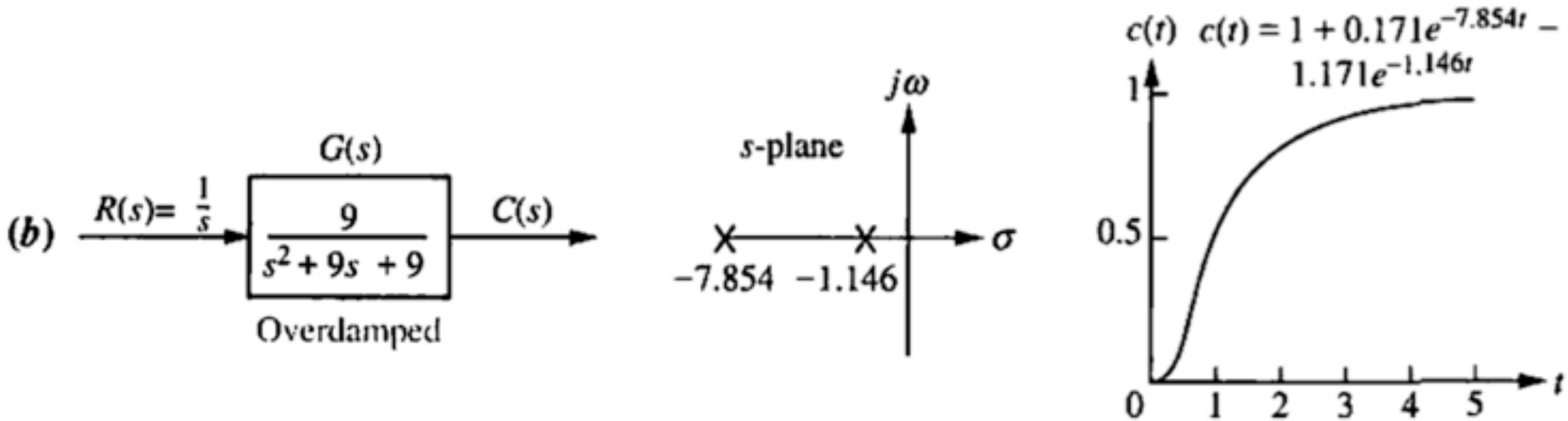
NISE, Control Systems Engineering, 6th. ed., Cap. 4.

Respostas típicas de sistemas de 2a-ordem:



NISE, Control Systems Engineering, 6th. ed., Cap. 4.

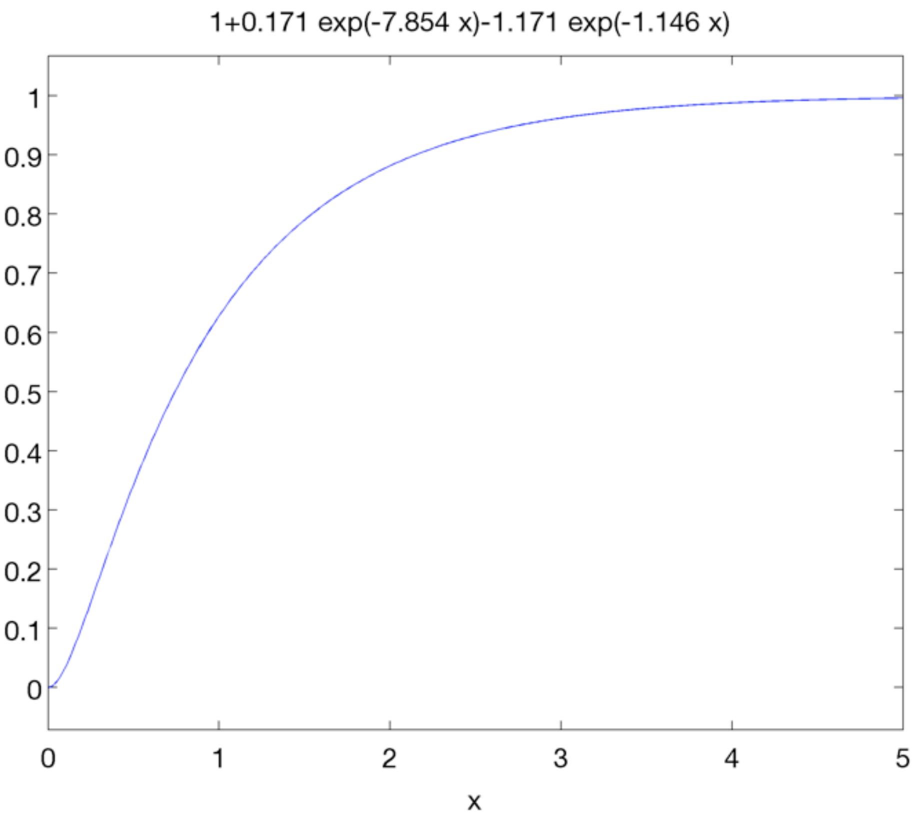
Caso: “Overdamped”

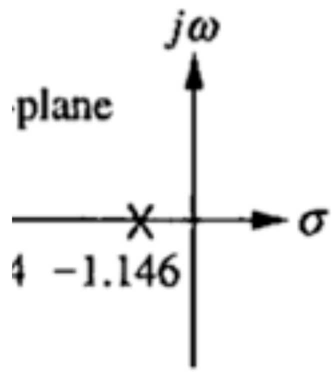


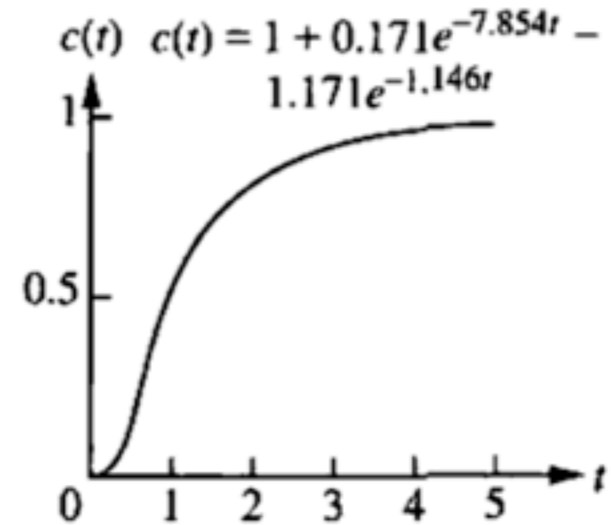
- Função: $$C(s) = \frac{9}{s(s^2+9s+9)} = \frac{9}{s(s+7,854)(s+1,146)}$$

$$\mathcal{L}^{-1}\{C(s)\} = c(t) = K_1 + K_2 e^{-7,854t} + K_3 e^{-1,146t}$$

Caso: “Overdamped”







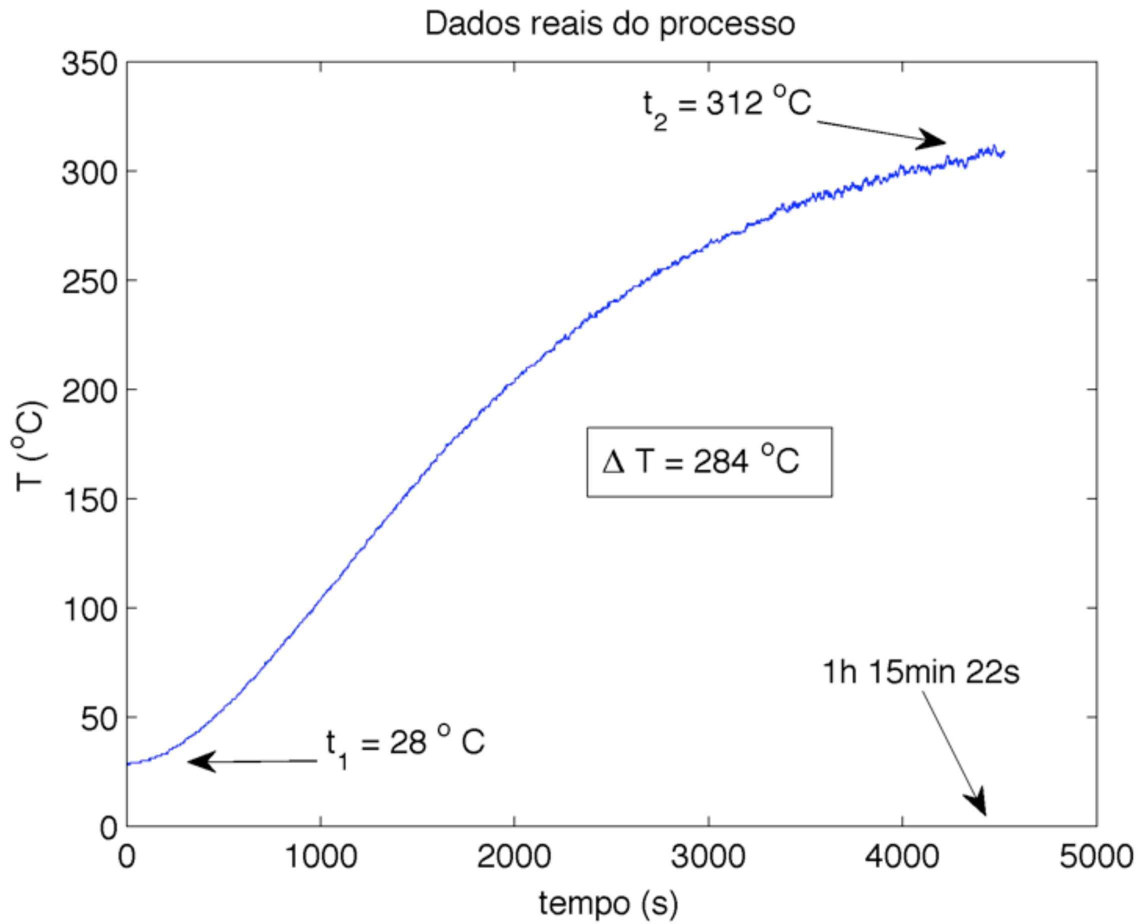
$$\frac{}{+ 9)} = \frac{9}{s(s + 7,854)(s + 1,146)}$$

$$= K_1 + K_2 e^{-7,854t} + K_3 e^{-1,146t}$$

```
>> ezplot('1+0.171*exp(-7.854*x)-1.171*exp(-1.146*x)',[0 5])
```

Caso da Estufa

- Dados Capturados:



- Parâmetros à serem levantados:

$$G(s) = \frac{b}{s^2 + as + b}$$

$$y(t) = K_1 + K_2 e^{-p_1 t} + K_3 e^{-p_2 t}$$

$$y(t) = K_1 e^{-p_1 t} + K_2 e^{-p_2 t} \longleftarrow \text{2 pólos reais!}$$

Generalizando

$$G(s) = \frac{b}{s^2 + as + b}$$

$$G(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2}$$

onde:

$$w_n = \sqrt{b} \quad \therefore \quad b = w_n^2$$

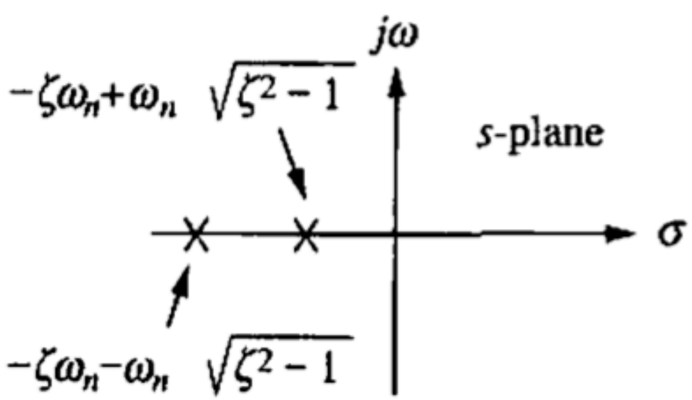
w_n = freq. natural de oscilação

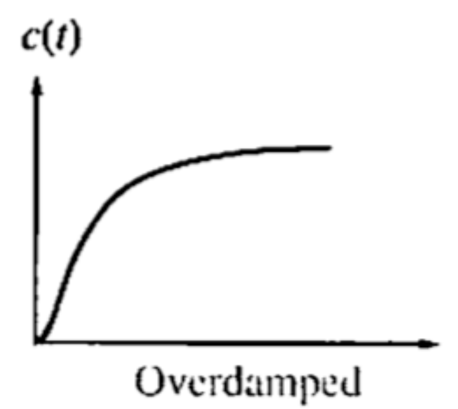
$$\zeta = \frac{|\sigma|}{w_n} = \frac{a/2}{w_n} \quad \therefore \quad a = 2\zeta w_n$$

σ, parte real dos pólos $= -a/2$

$$p_{1,2} = -\zeta w_n \pm w_n\sqrt{\zeta^2 - 1}$$

$\zeta > 1$





Transformada Inversa de Laplace:

Sistema *overdamped*: 2 pólos reais

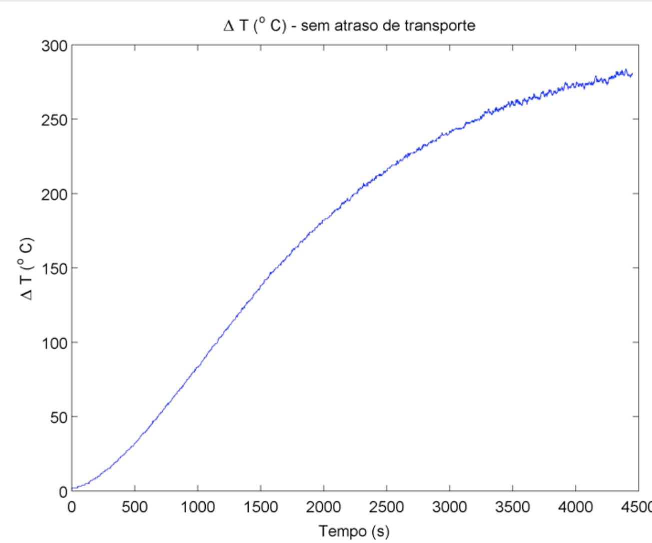
$$G(s) = \frac{K}{(s+p_1)(s+p_2)}$$

$$y(t) = \mathcal{L}^{-1}\left\{ \underbrace{U(s)}_{\text{Degrau Unitário}} \cdot G(s) \right\}$$

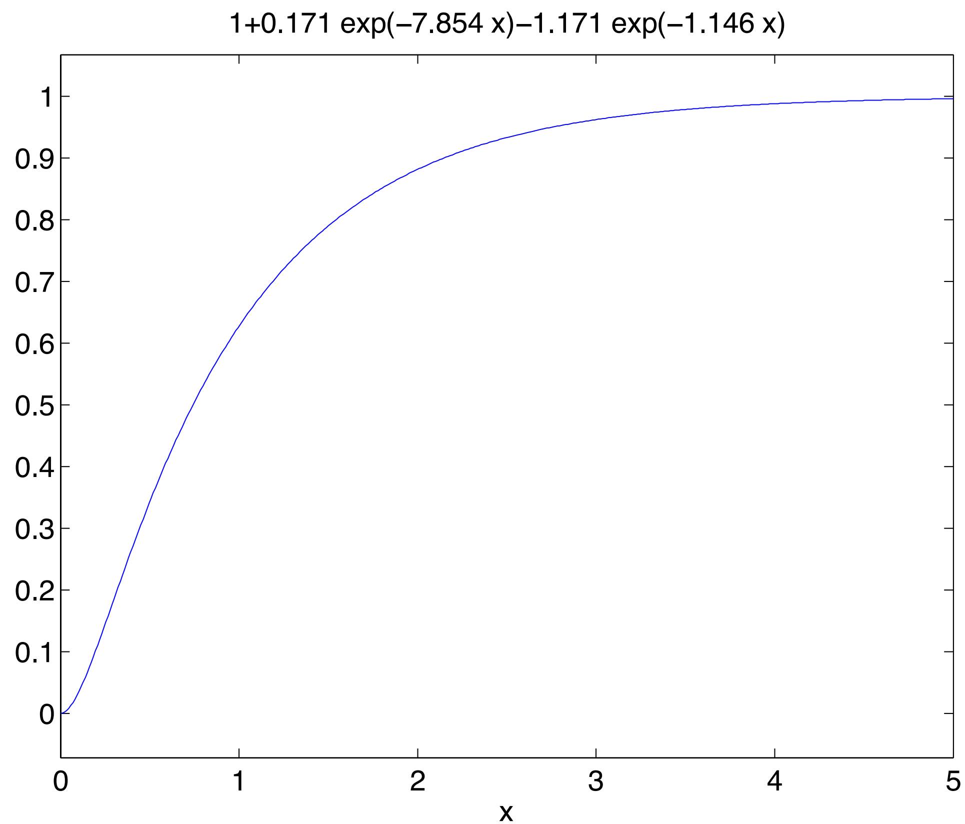
Resolvendo usando MATLAB:

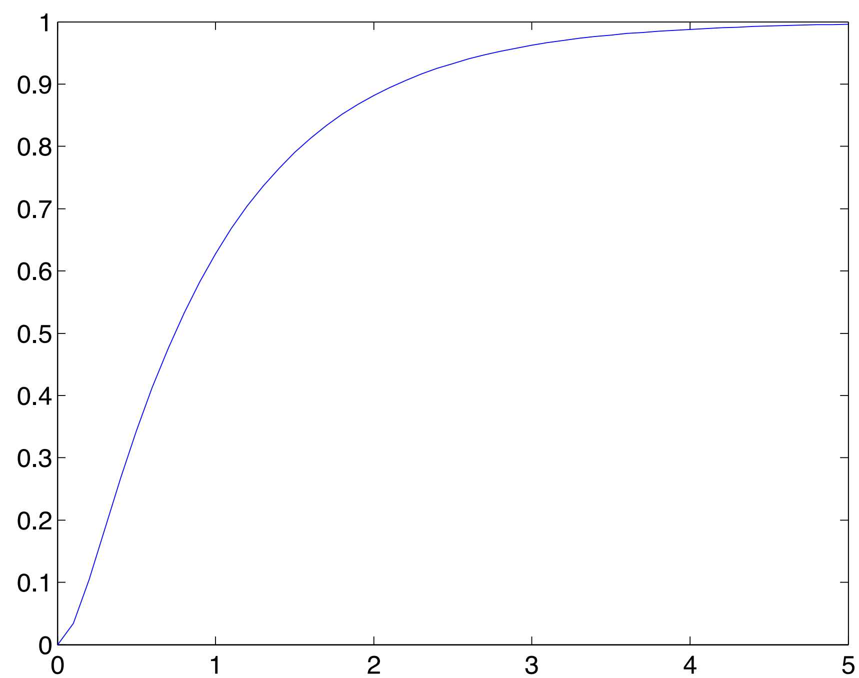
$$y(t) = 1 + K_1 e^{-p_1 t} + K_2 e^{-p_2 t}$$

```
function y = resp2polosreais(par,t)
% y(t) = K_1*exp(-p_1*t)+K_2*exp(-p_2*t)
% y(t) = par(1)*exp(-par(2)*t)+par(3)*exp(-par(4)*t)
% caso de resp. de 2a-ordem sistema overdamped (2 p?los reais)
y = 1 + par(1)*exp(-par(2)*t) + par(3)*exp(-par(4)*t);
```



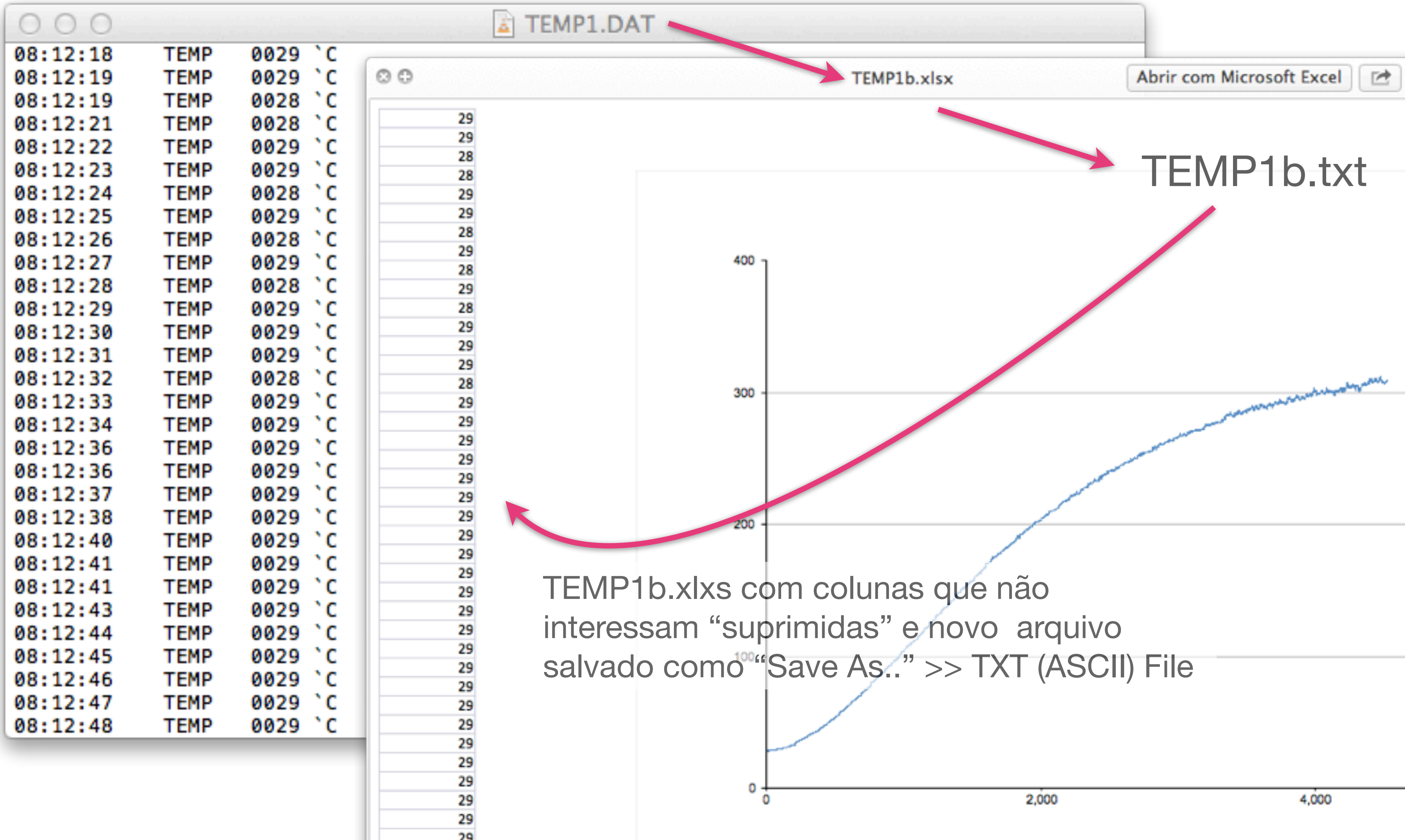
Testando ‘resp2polosreais.m’






```
>> ezplot('1+0.171*exp(-7.854*x)-1.171*exp(-1.146*x)',[0 5])
>> par=[0.171 7.854 -1.171 1.146]; % exemplo do NISE, Cap 4.
>> y4=resp2polosreais(par,t4);
>> figure; plot(t4,y4)
```

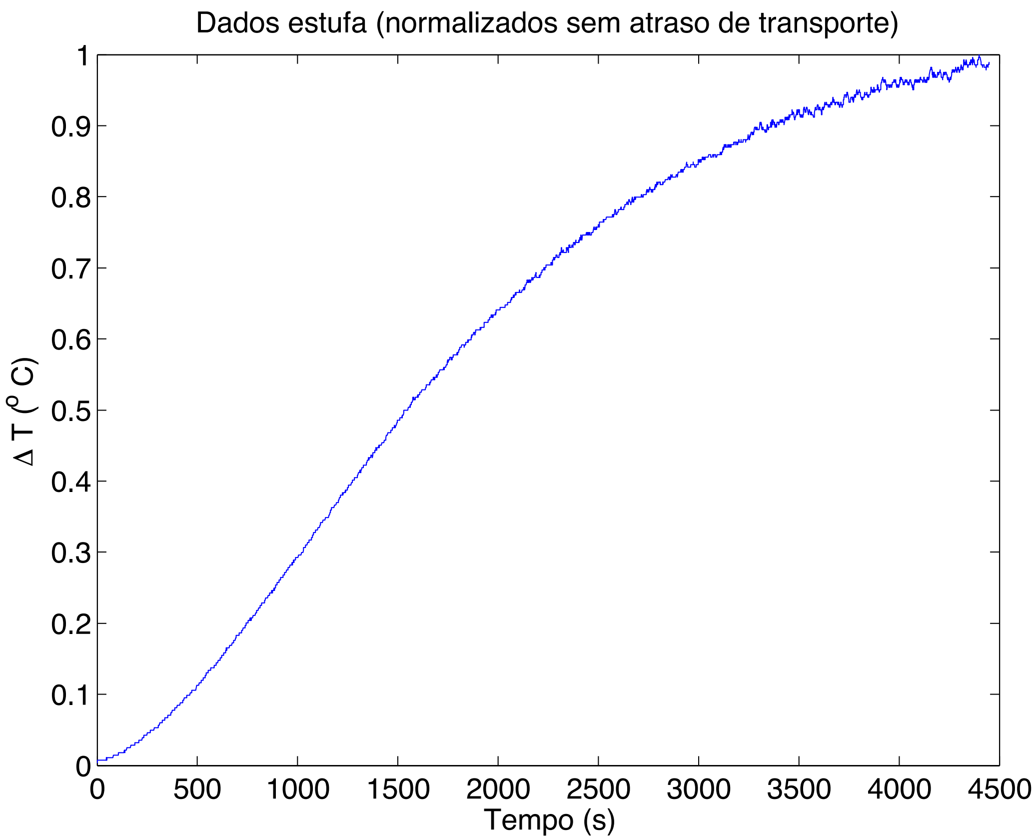
Dados da Estufa



TEMP1b.txt

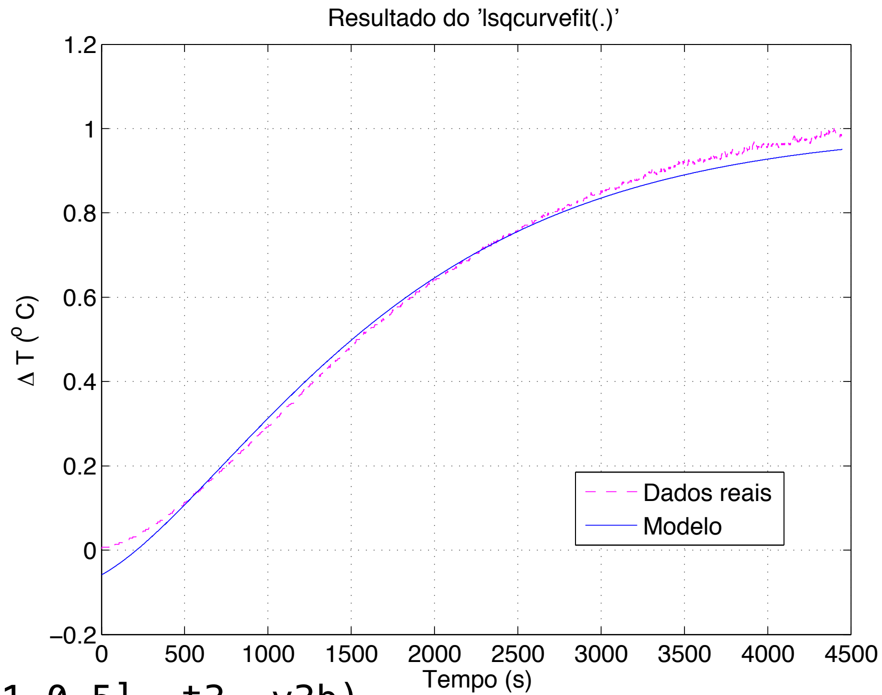
TEMP1b.xlxs com colunas que não interessam “suprimidas” e novo arquivo salvado como “Save As..” >> TXT (ASCII) File

Normalizando dados da Estufa




```
>> load -ascii temp1b.txt
>> % detalhe: temos que "normalizar dados da estufa"
>> y3b=y3/delta_t;
>> figure; plot(t3,y3b)
>> title('Dados estufa (normalizados sem atraso de transporte)');
>> xlabel('Tempo (s)')
>> ylabel('\Delta T (^o C)')
```

Rodando ‘lsqcurvefit(.)'




```
>> par=lsqcurvefit('resp2polosreais',[0.1 1 -1 0.5], t3, y3b)

Solver stopped prematurely.

lsqcurvefit stopped because it exceeded the function evaluation limit,
options.MaxFunEvals = 400 (the default value).


par =

   24.4479    0.0011  -25.5068    0.0010

>> % testando parâmetros encontrados...
>> y3teste=resp2polosreais(par, t3);
>> figure; plot(t3,y3b,'m--', t3,y3teste,'b-')
```

par(1)=63,5097
par(2)=1,0207x10^{-4}
par(3)=-64,5224

Laplace

```
y = 1 + par(1)*exp(-par(2)*t) + par(3)*exp(-par(4)*t);
```

par(1)=63,5097
par(2)=1,0207x10^{-4}
par(3)=-64,5224
par(4)=1,0606x10^{-4}

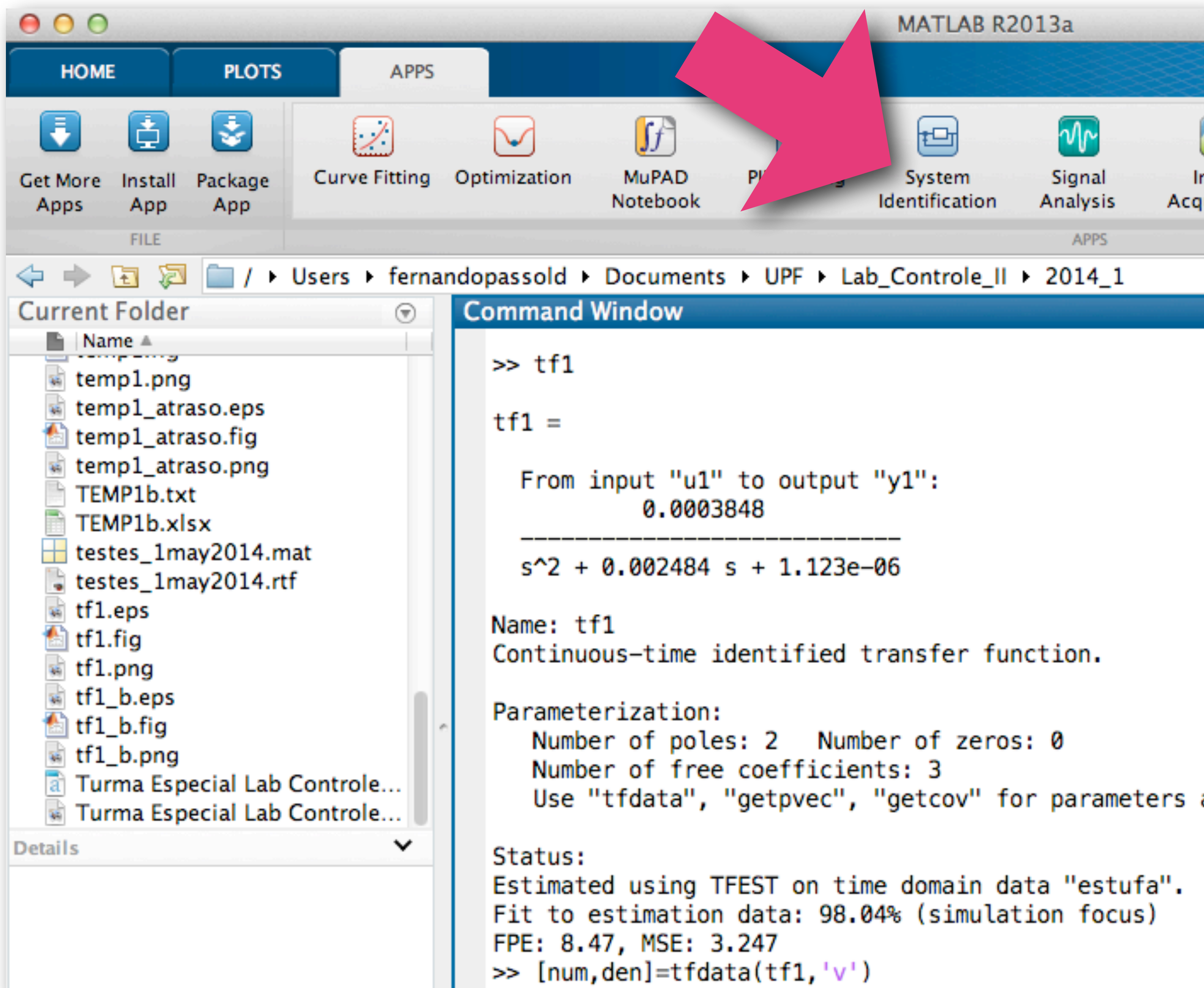
$$\mathcal{L}\left\{1 + a\,e^{-b\,t} + c\,e^{-d\,t}\right\} = \frac{a}{(s+b)} + \frac{c}{(s+d)} + \frac{1}{s}$$

$$Y(s) = R(s) \cdot G(s) = \frac{1}{s} \cdot G(s)$$

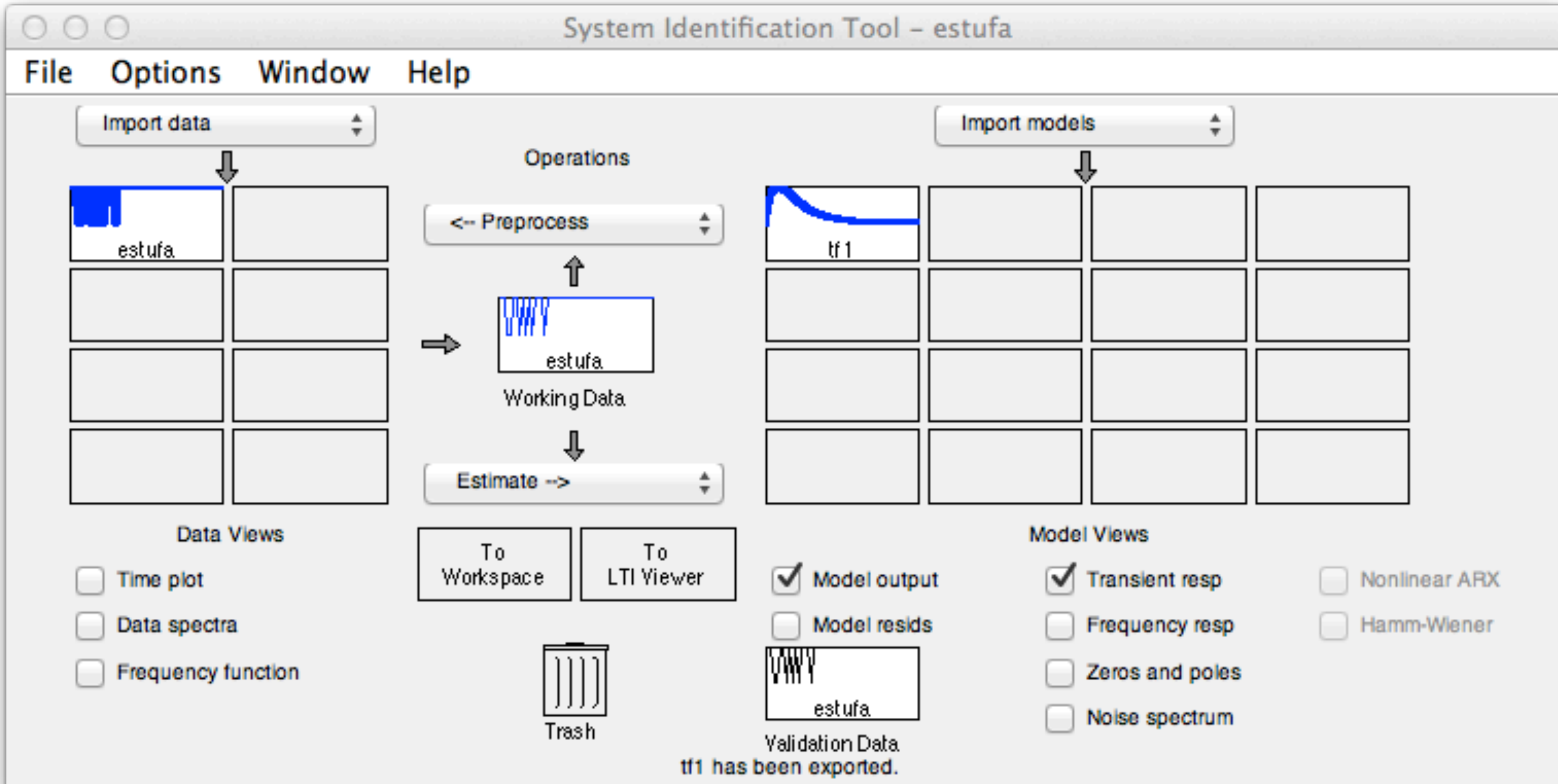
$$G(s) = \frac{K}{(s+b)(s+d)}$$

Modelo Esperado

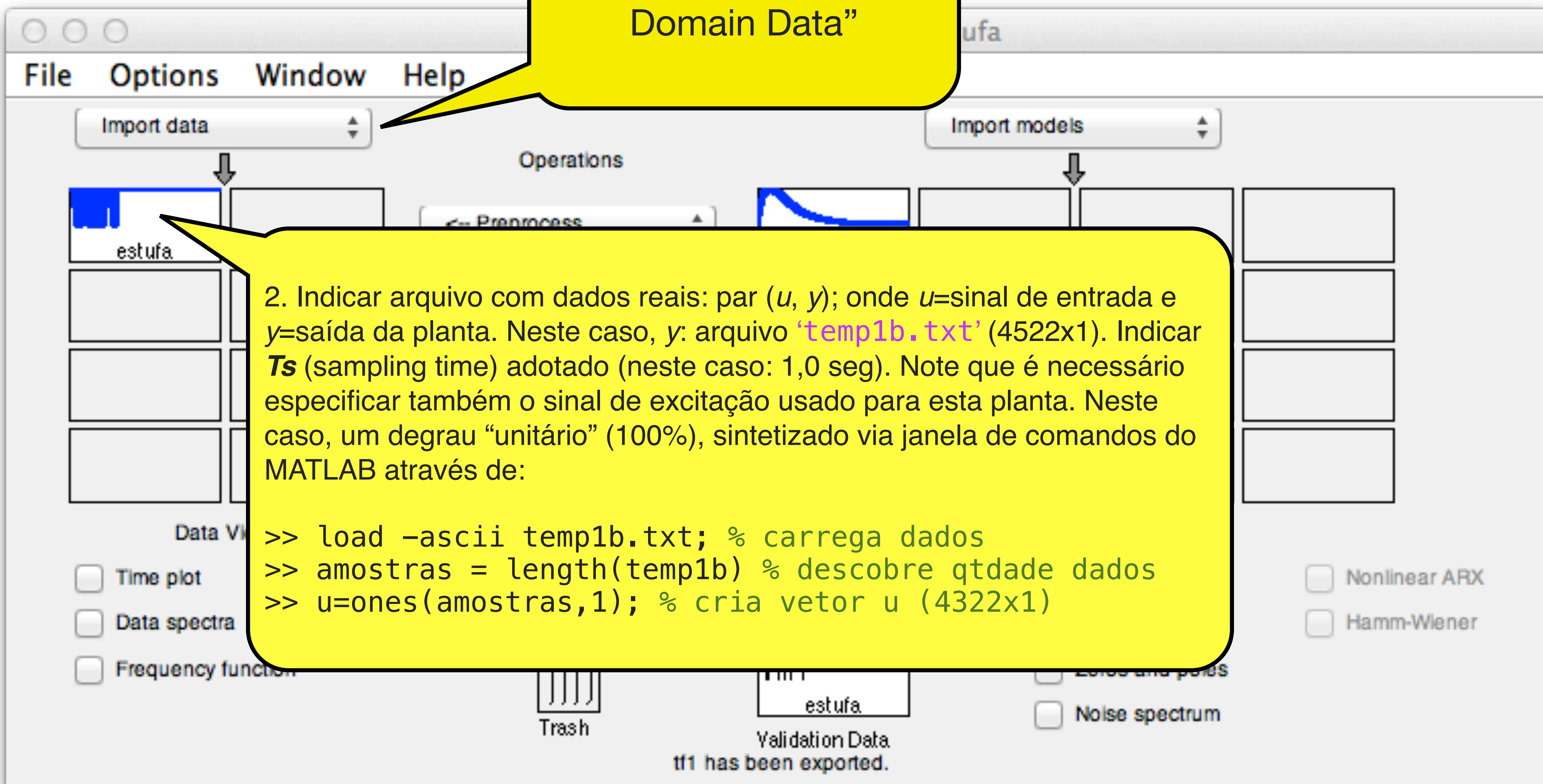
- Uso do “System Identification” toolbox do Matlab



System Identification tool



System Identification tool

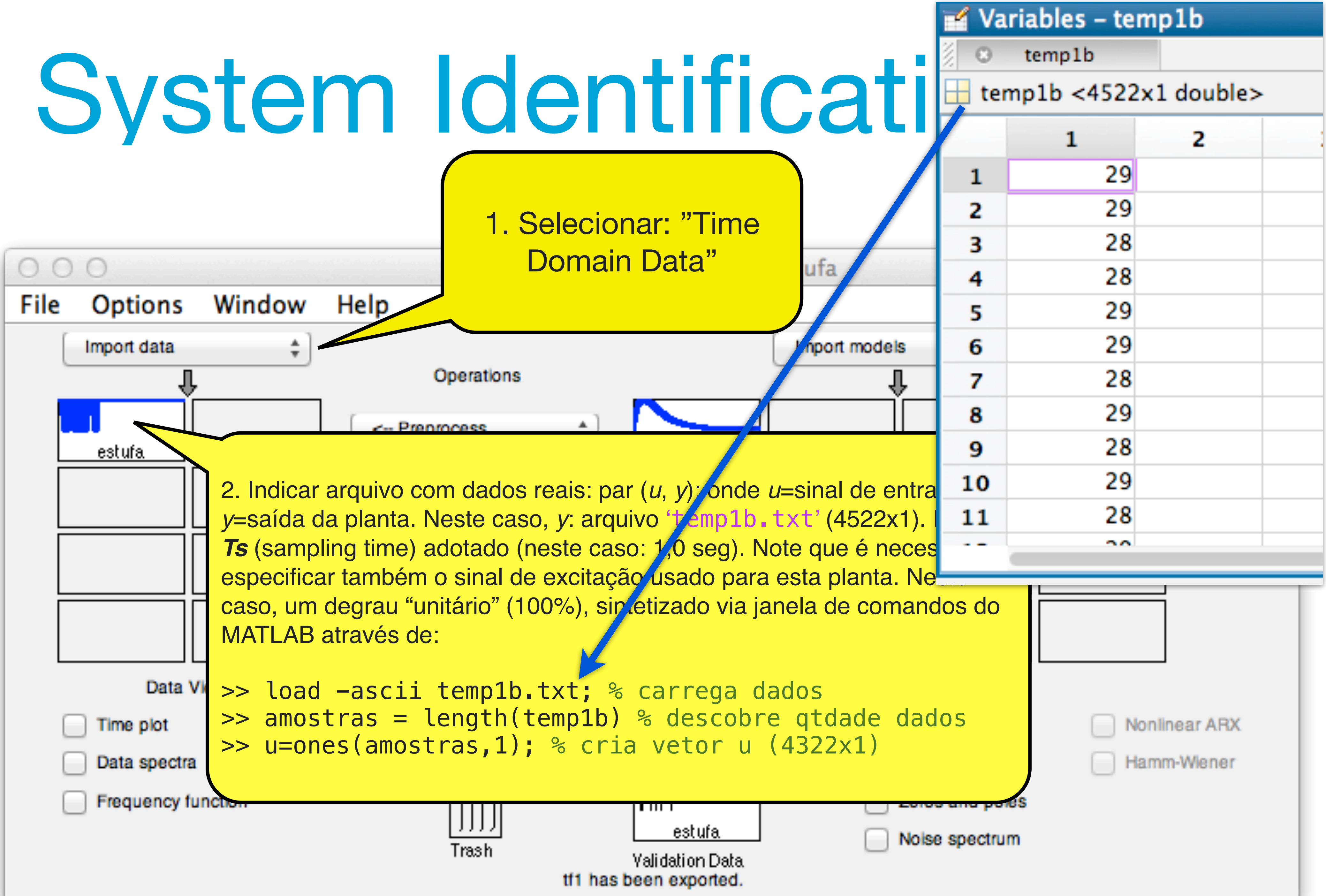


1. Selecionar: "Time Domain Data"

2. Indicar arquivo com dados reais: par (*u*, *y*); onde *u*=sinal de entrada e *y*=saída da planta. Neste caso, *y*: arquivo `'temp1b.txt'` (4522x1). Indicar ***Ts*** (sampling time) adotado (neste caso: 1,0 seg). Note que é necessário especificar também o sinal de excitação usado para esta planta. Neste caso, um degrau "unitário" (100%), sintetizado via janela de comandos do MATLAB através de:

```
>> load -ascii temp1b.txt; % carrega dados
>> amostras = length(temp1b) % descobre qtdade dados
>> u=ones(amostras,1); % cria vetor u (4322x1)
```

System Identificati



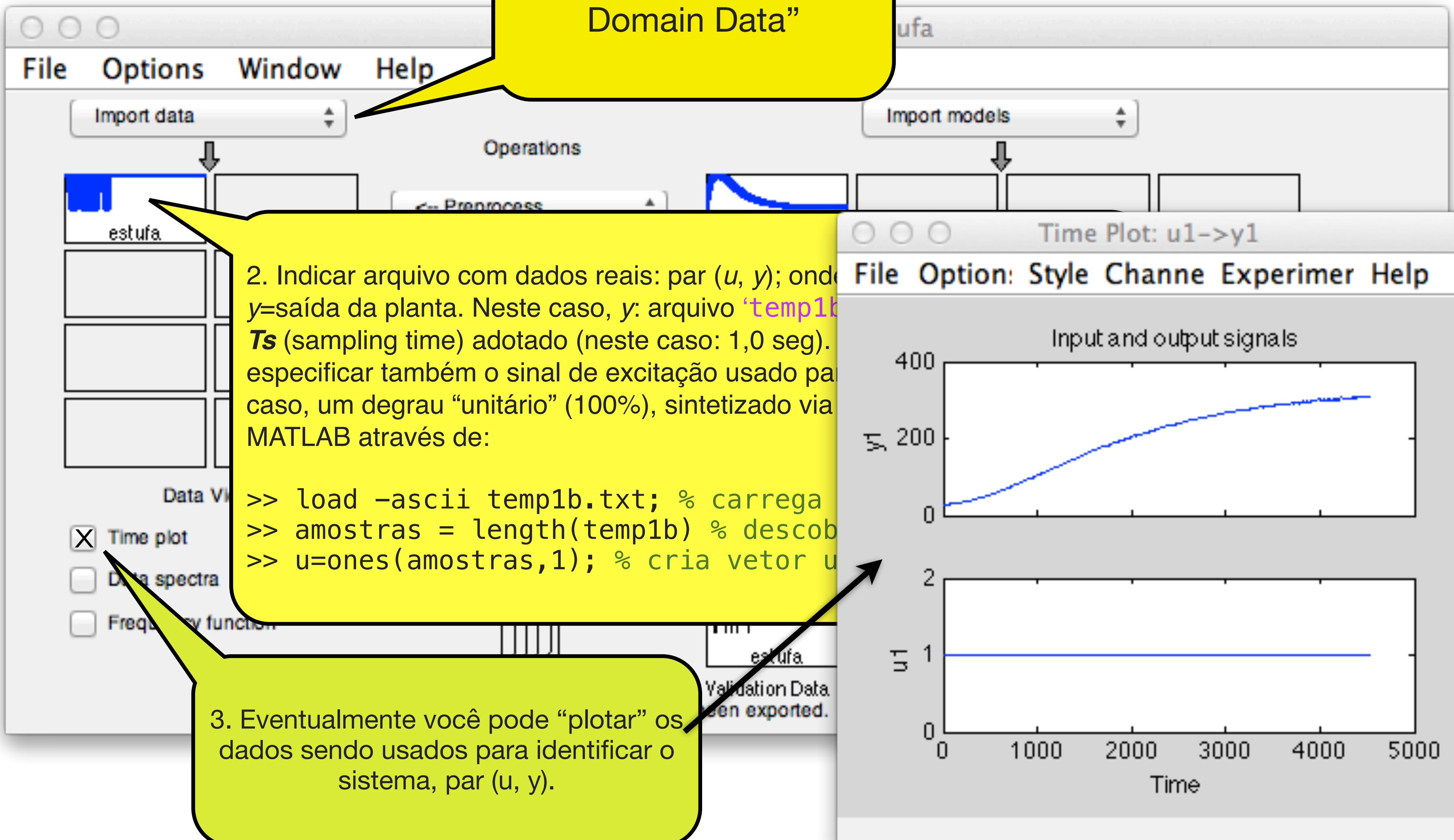
1. Selecionar: ”Time Domain Data”

2. Indicar arquivo com dados reais: par (*u*, *y*); onde *u*=sinal de entra *y*=saída da planta. Neste caso, *y*: arquivo ‘temp1b.txt’ (4522x1). I ***Ts*** (sampling time) adotado (neste caso: 1,0 seg). Note que é neces especificar também o sinal de excitação usado para esta planta. Ne caso, um degrau “unitário” (100%), sintetizado via janela de comandos do MATLAB através de:

```
>> load -ascii temp1b.txt; % carrega dados
>> amostras = length(temp1b) % descobre qtdade dados
>> u=ones(amostras,1); % cria vetor u (4322x1)
```

System Identification tool

1. Selecionar: ”Time Domain Data”

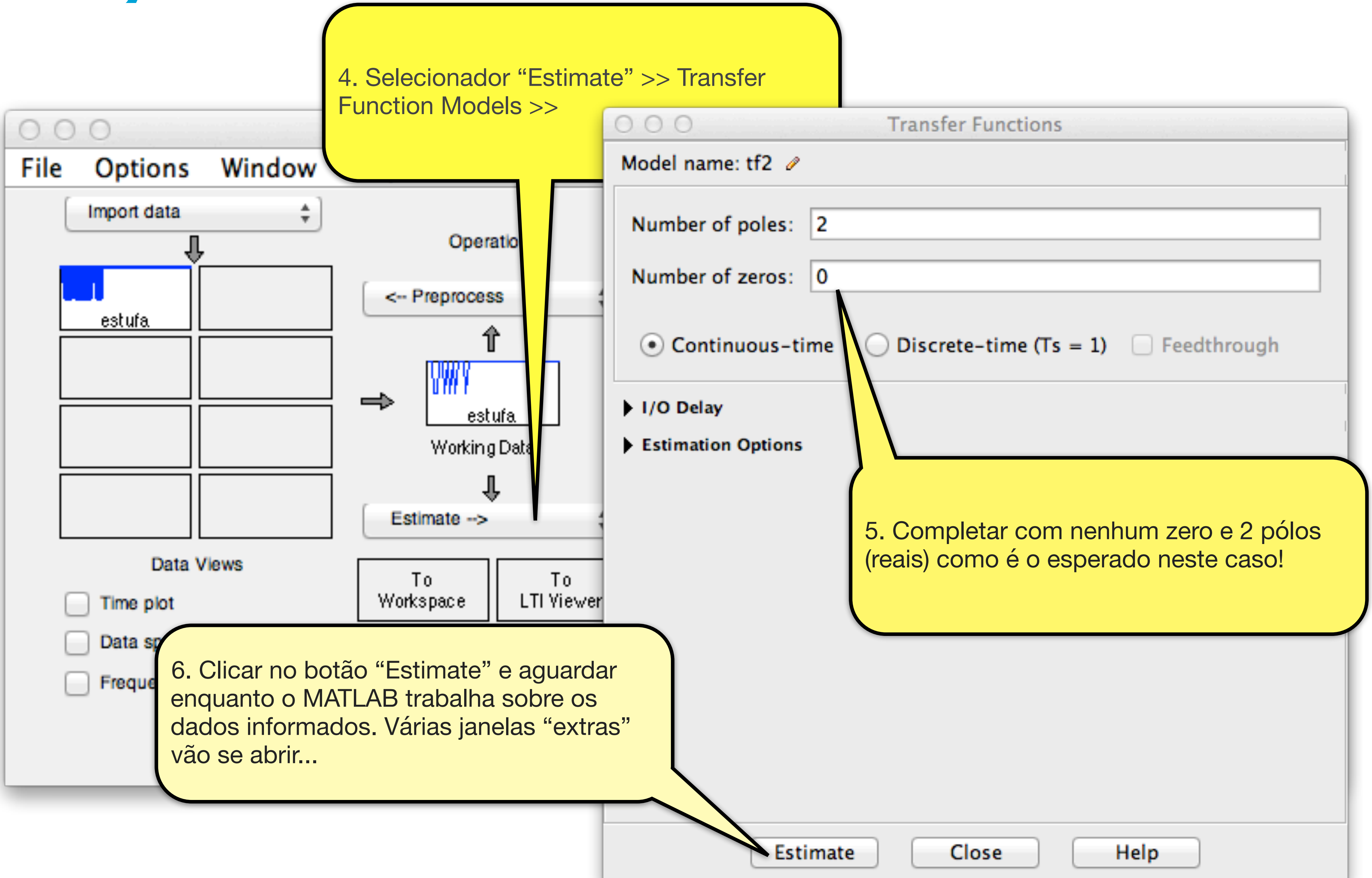


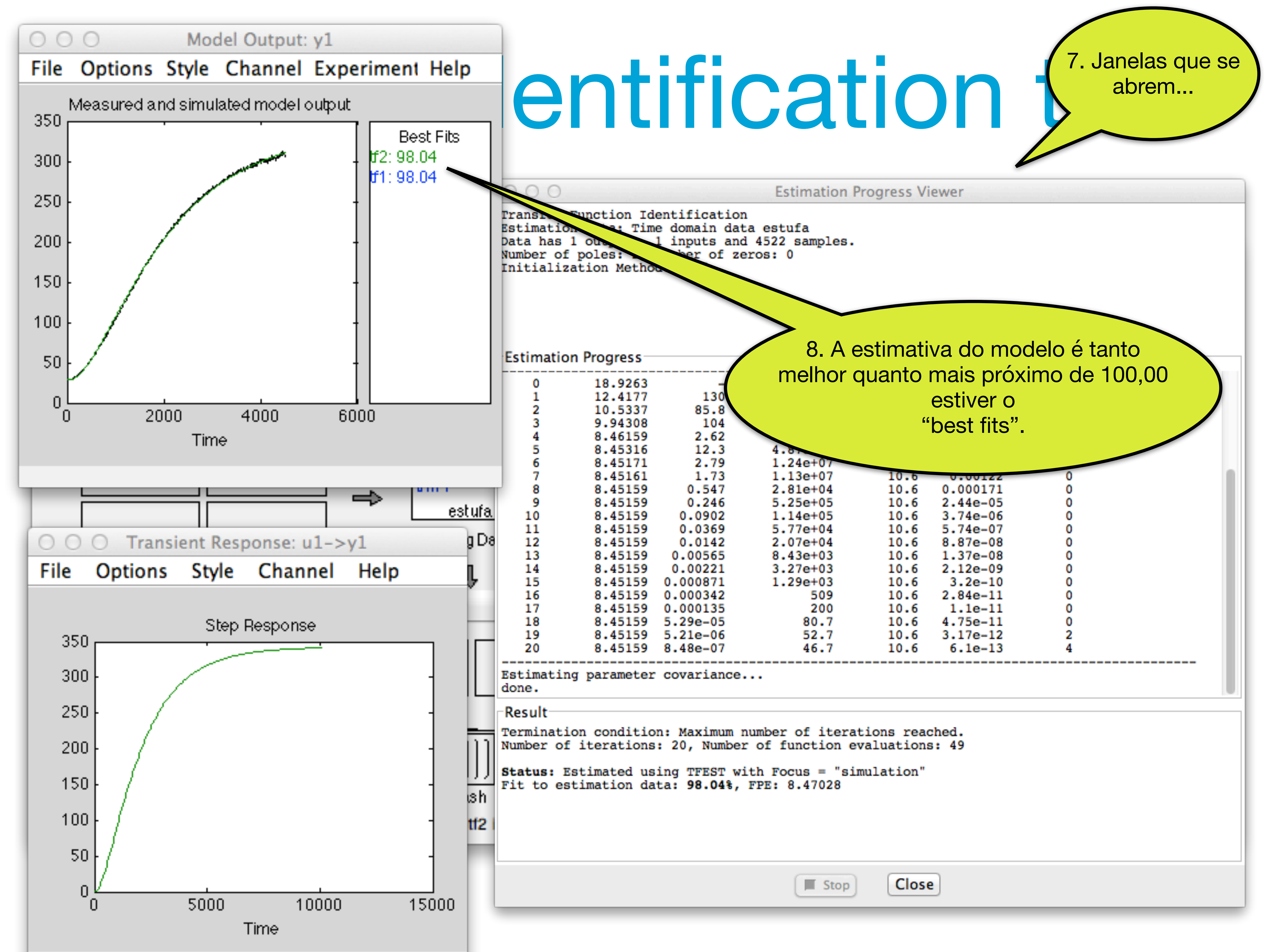
2. Indicar arquivo com dados reais: par (*u*, *y*); onde *y*=saída da planta. Neste caso, *y*: arquivo ‘temp1b *Ts* (sampling time) adotado (neste caso: 1,0 seg). especificar também o sinal de excitação usado pa caso, um degrau “unitário” (100%), sintetizado via MATLAB através de:

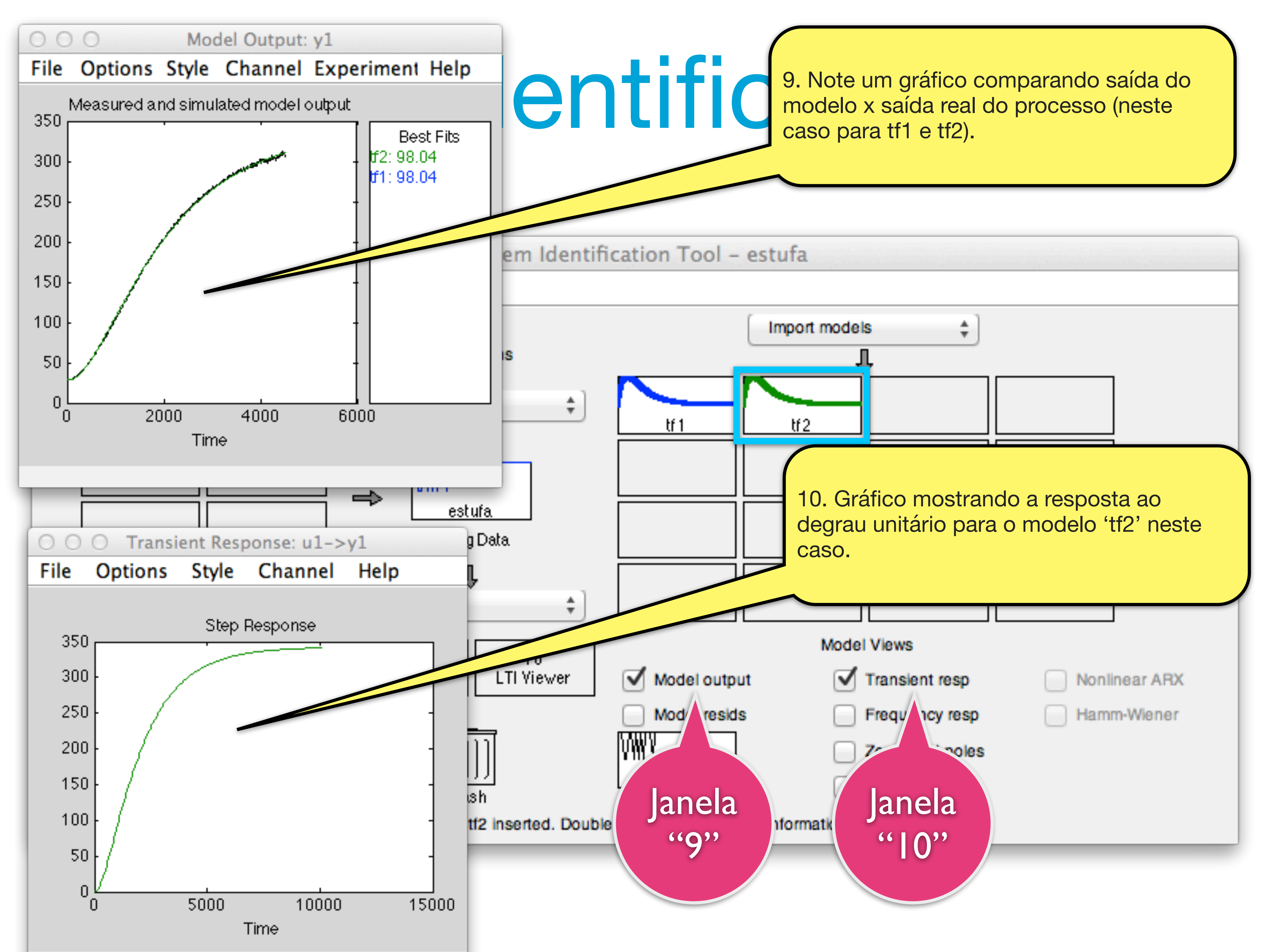
```
>> load -ascii temp1b.txt; % carrega
>> amostras = length(temp1b) % descob
>> u=ones(amostras,1); % cria vetor u
```

3. Eventualmente você pode “plotar” os dados sendo usados para identificar o sistema, par (u, y).

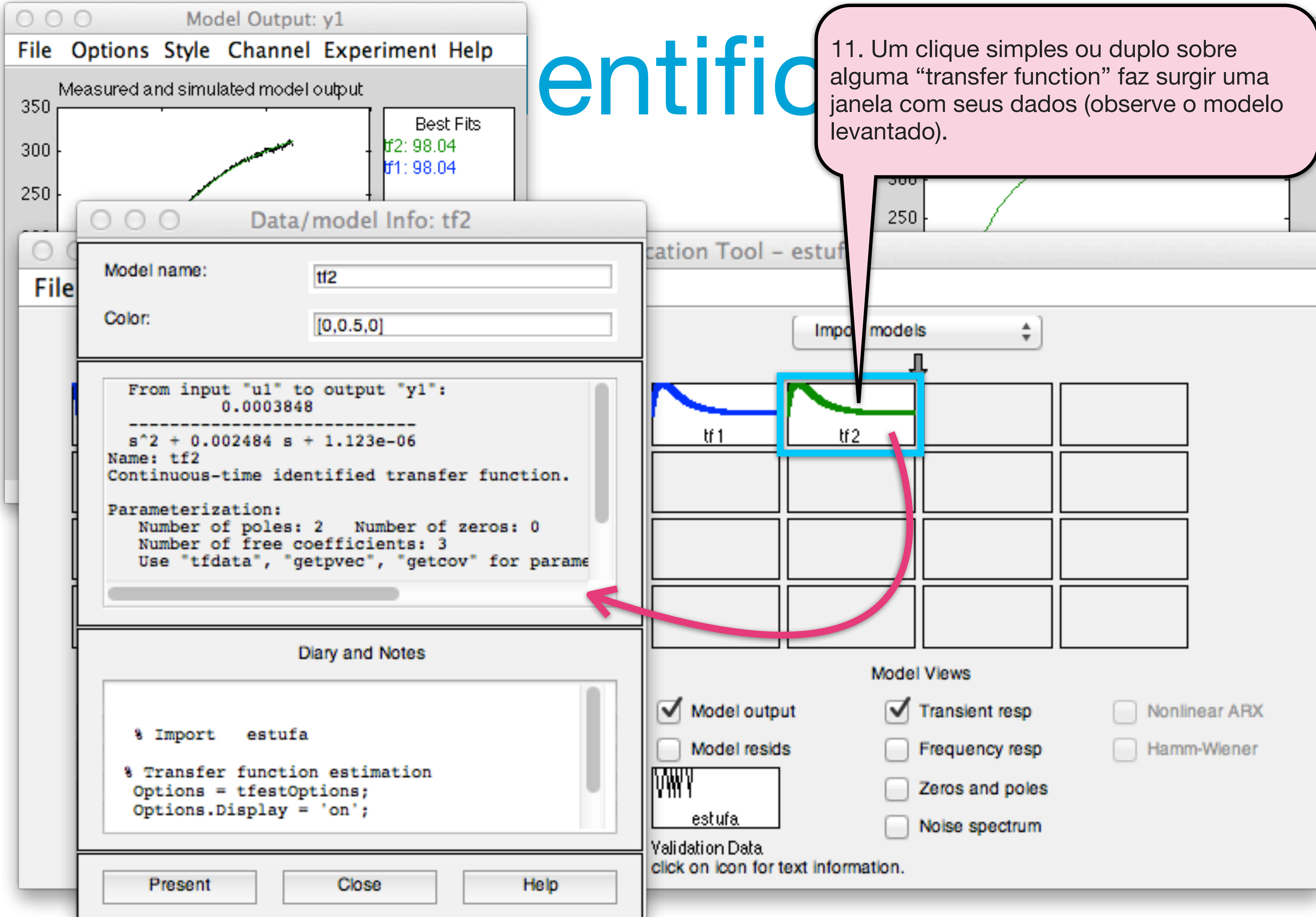
System Identification tool




entification t
7. Janelas que se abrem...
8. A estimativa do modelo é tanto melhor quanto mais próximo de 100,00 estiver o “best fits”.
Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
tf2: 98.04
tf1: 98.04
Time
Transient Response: u1->y1
File Options Style Channel Help
Step Response
Time
Estimation Progress Viewer
Transfer Function Identification
Data has 1 outputs, 1 inputs and 4522 samples.
Number of poles: ... Number of zeros: 0
Estimation Progress
Estimating parameter covariance...
done.
Result
Termination condition: Maximum number of iterations reached.
Number of iterations: 20, Number of function evaluations: 49
Status: Estimated using TFEST with Focus = "simulation"
Fit to estimation data: 98.04%, FPE: 8.47028
Stop
Close

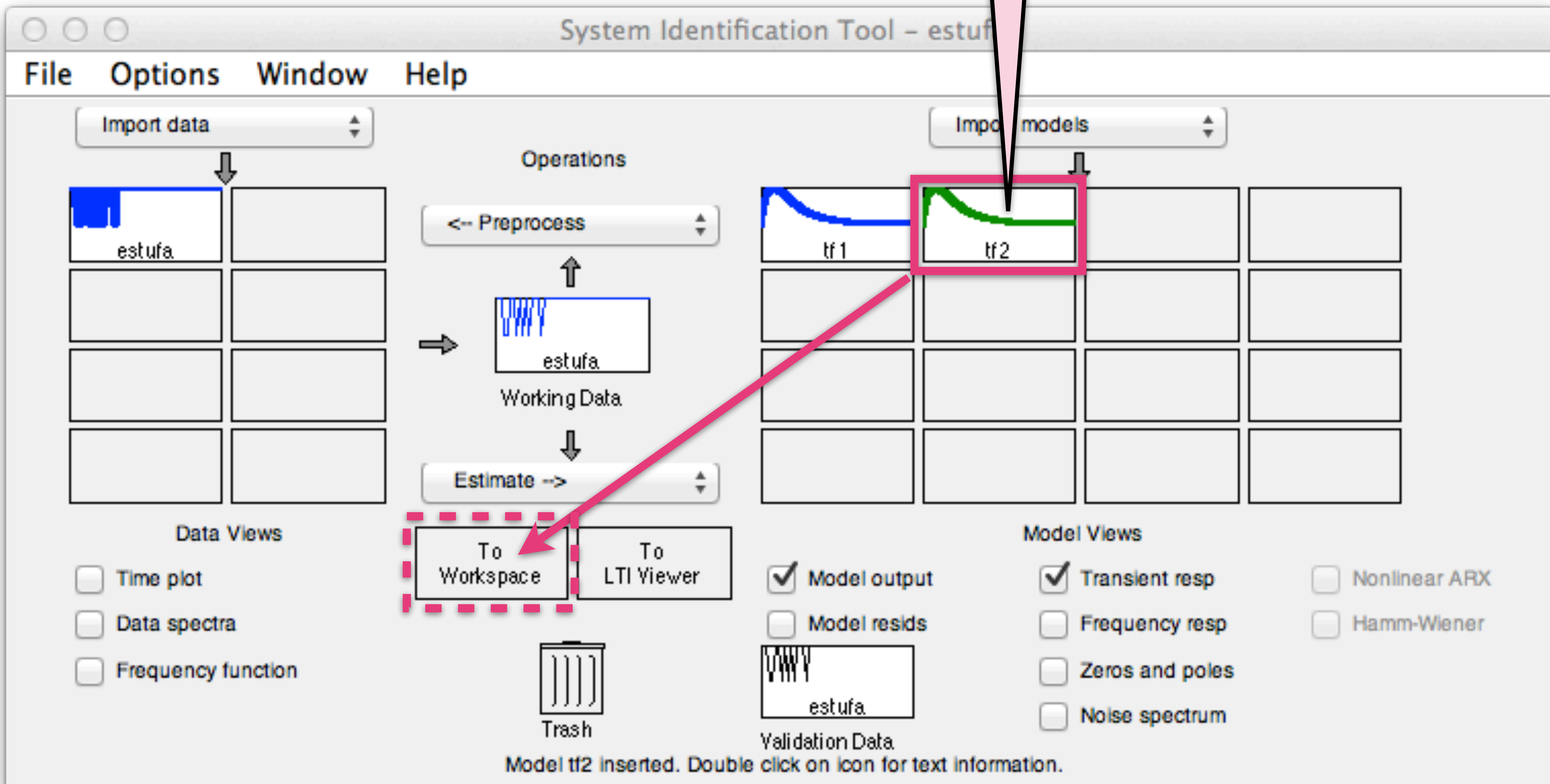


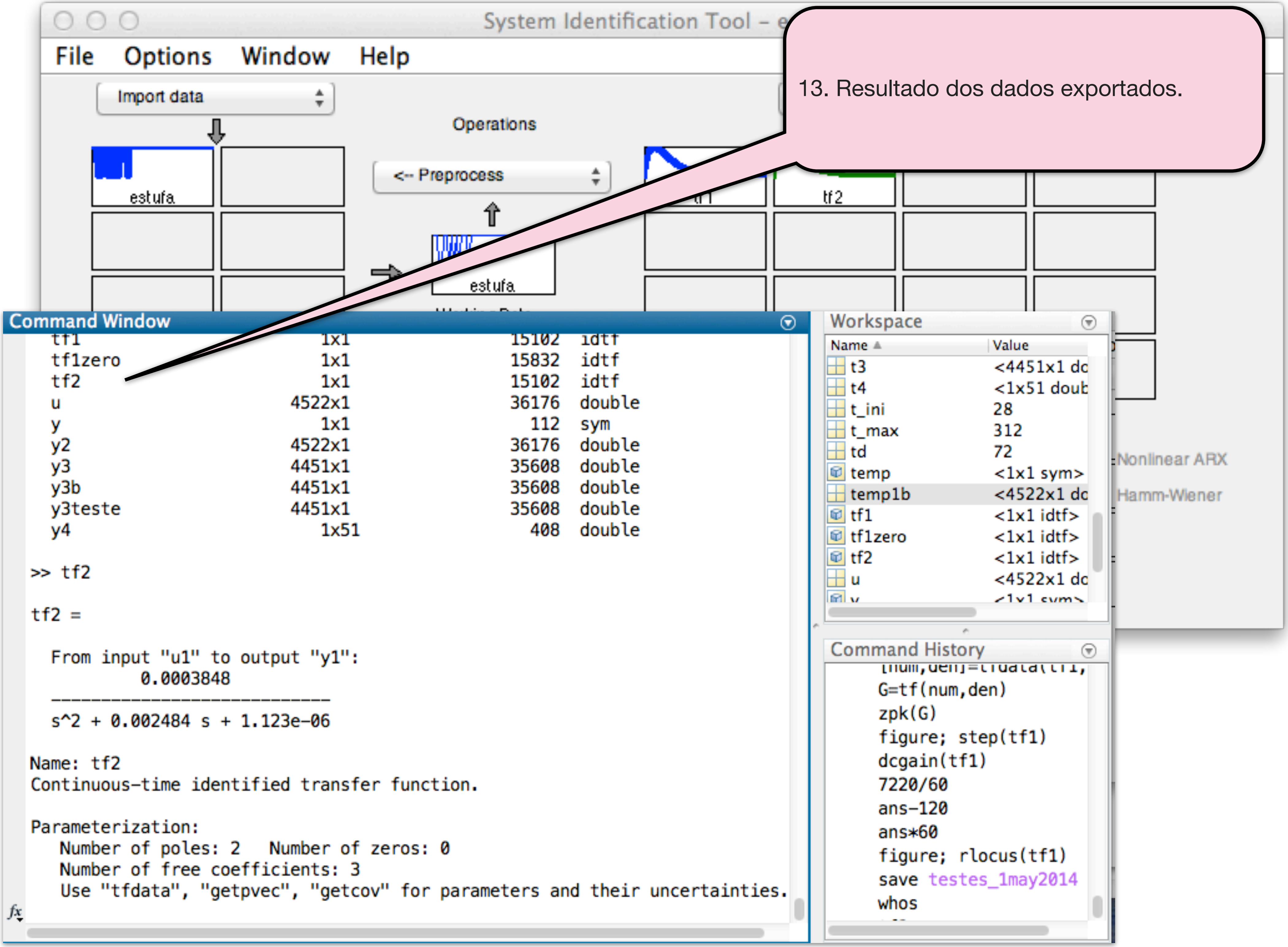
entific
9. Note um gráfico comparando saída do modelo x saída real do processo (neste caso para tf1 e tf2).
10. Gráfico mostrando a resposta ao degrau unitário para o modelo ‘tf2’ neste caso.
Janela “9”
Janela “10”
Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
tf2: 98.04
tf1: 98.04
Time
Transient Response: u1->y1
File Options Style Channel Help
Step Response
Time
em Identification Tool – estufa
Import models
tf1
tf2
estufa
LTI Viewer
Model Views
Model output
Transient resp
Nonlinear ARX
Hamm-Wiener

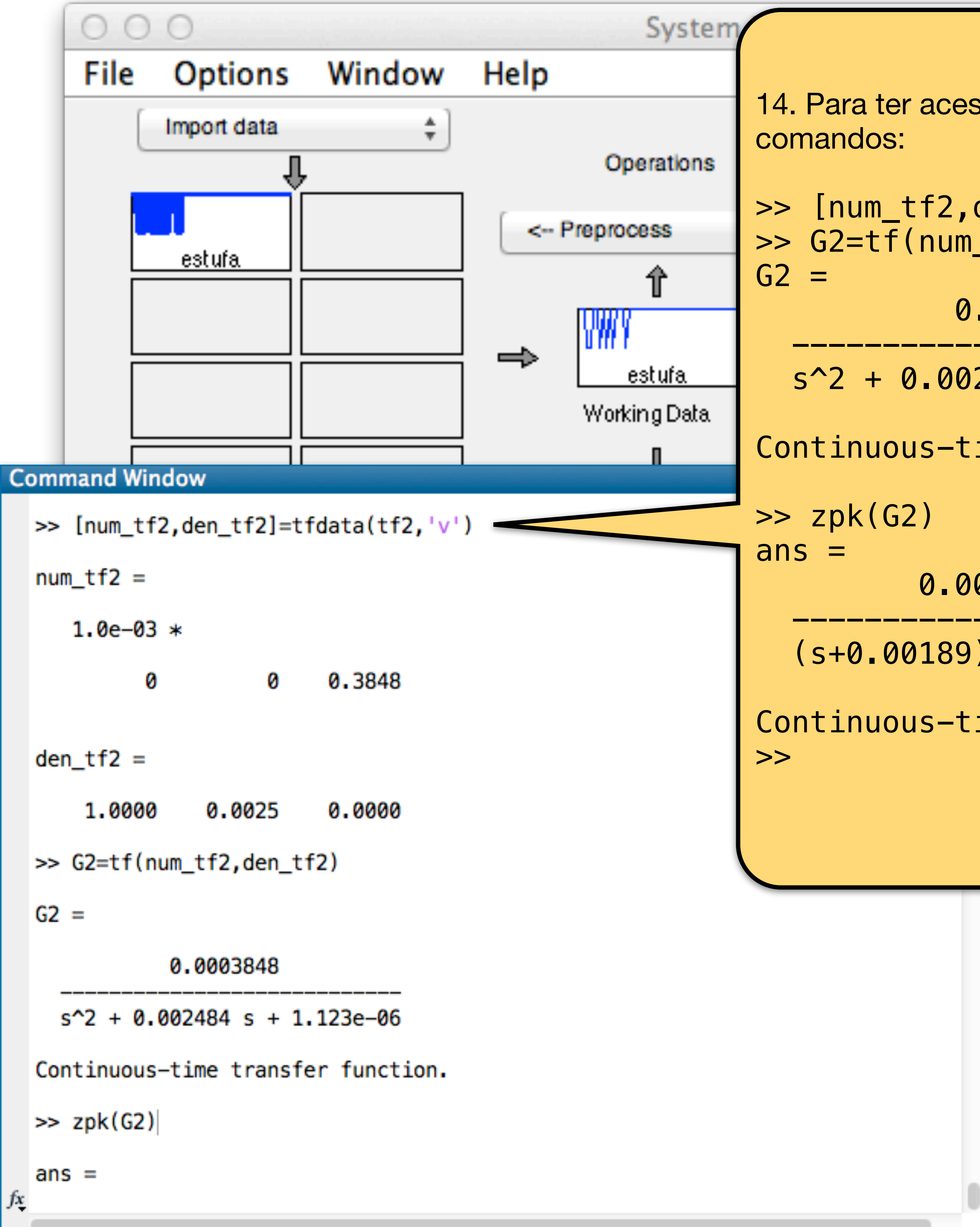

entific
11. Um clique simples ou duplo sobre alguma “transfer function” faz surgir uma janela com seus dados (observe o modelo levantado).
Model Output: y1
File Options Style Channel Experiment Help
Measured and simulated model output
Best Fits
tf2: 98.04
tf1: 98.04
Data/model Info: tf2
Model name:
tf2
Color:
[0,0.5,0]
From input "u1" to output "y1":
0.0003848
----
s^2 + 0.002484 s + 1.123e-06
Name: tf2
Continuous-time identified transfer function.
Parameterization:
Number of poles: 2 Number of zeros: 0
Number of free coefficients: 3
Use "tfdata", "getpvec", "getcov" for parame
Diary and Notes
% Import estufa
% Transfer function estimation
Options = tfestOptions;
Options.Display = 'on';
Present
Close
Help
cation Tool - estuf
File
models
tf1
tf2
Model Views
Model output
Transient resp
Nonlinear ARX
Model resids
Frequency resp
Hamm-Wiener
Zeros and poles
Noise spectrum
estufa
Validation Data
click on icon for text information.

System Identific

12. Para “exportar” os dados para o “WorkSpace” do MATLAB é necessário “arrastar” o modelo selecionado (neste caso ‘tf2’ para o quadro “To Workspace”




13. Resultado dos dados exportados.
System Identification Tool
File Options Window Help
Import data
Operations
<-- Preprocess
estufa
tf2
estufa
Nonlinear ARX
Hamm-Wiener
Command Window
tf1zero 1x1 15832 idtf
tf2 1x1 15102 idtf
u 4522x1 36176 double
y 1x1 112 sym
y2 4522x1 36176 double
y3 4451x1 35608 double
y3b 4451x1 35608 double
y3teste 4451x1 35608 double
y4 1x51 408 double
>> tf2
tf2 =
From input "u1" to output "y1":
0.0003848
s^2 + 0.002484 s + 1.123e-06
Name: tf2
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.
Workspace
Name Value
t3 <4451x1 do
t4 <1x51 doub
t_ini 28
t_max 312
td 72
temp <1x1 sym>
temp1b <4522x1 do
tf1 <1x1 idtf>
tf1zero <1x1 idtf>
tf2 <1x1 idtf>
u <4522x1 do
Command History
G=tf(num,den)
zpk(G)
figure; step(tf1)
dcgain(tf1)
7220/60
ans-120
ans*60
figure; rlocus(tf1)
save testes_1may2014
whos



14. Para ter acesso aos dados do modelo exportado, usar comandos:

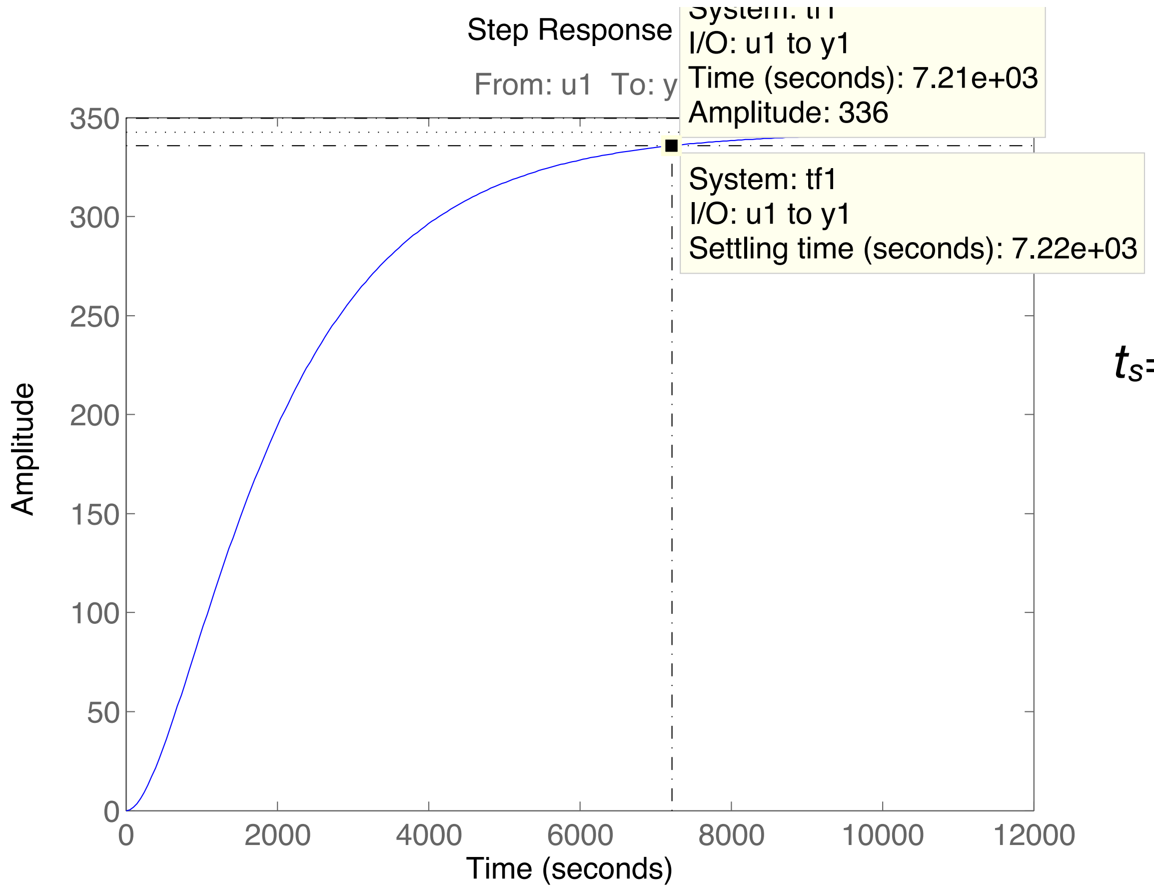
```
>> [num_tf2,den_tf2]=tfdata(tf2,'v')
>> G2=tf(num_tf2,den_tf2)
G2 =
           0.0003848
  ----------------------------
  s^2 + 0.002484 s + 1.123e-06

Continuous-time transfer function.

>> zpk(G2)
ans =
         0.00038476
  -------------------------
  (s+0.00189) (s+0.0005942)

Continuous-time zero/pole/gain model.
>>
```

Modelo Levantado

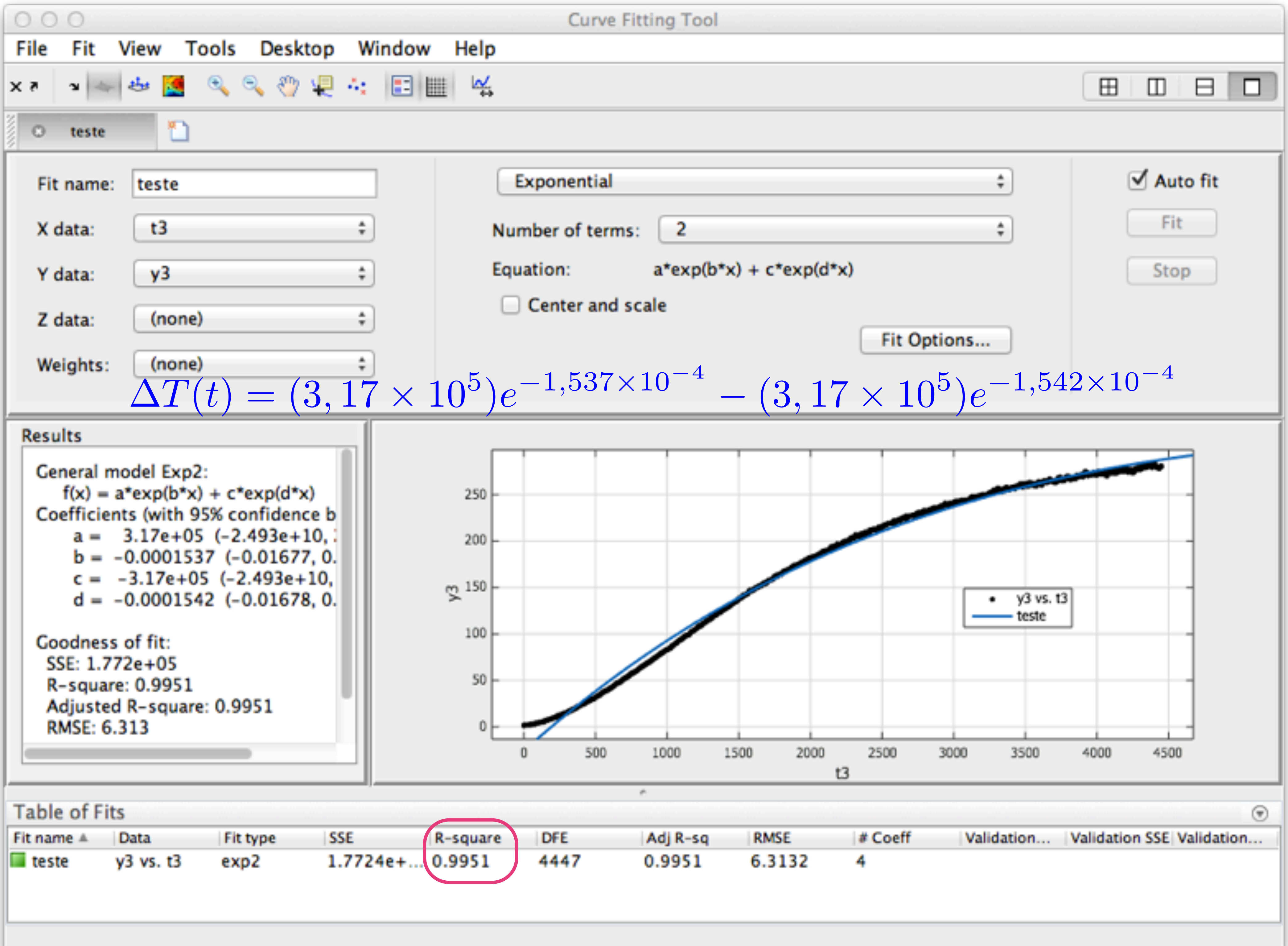
$$G(s) = \frac{3.848 \cdot 10^{-4}}{s^2 + 0,002484s + 1,123 \cdot 10^{-6}}$$

$$G(s) = \frac{3.8476 \cdot 10^{-4}}{(s + 0,00189)(s + 0,0005942)}$$

```
>> dcgain(tf1)
ans =
  342.6445
```



t_s=2h20min



$$\Delta T(t) = (3,17 \times 10^5)e^{-1,537\times 10^{-4}} - (3,17 \times 10^5)e^{-1,542\times 10^{-4}}$$

Laplace

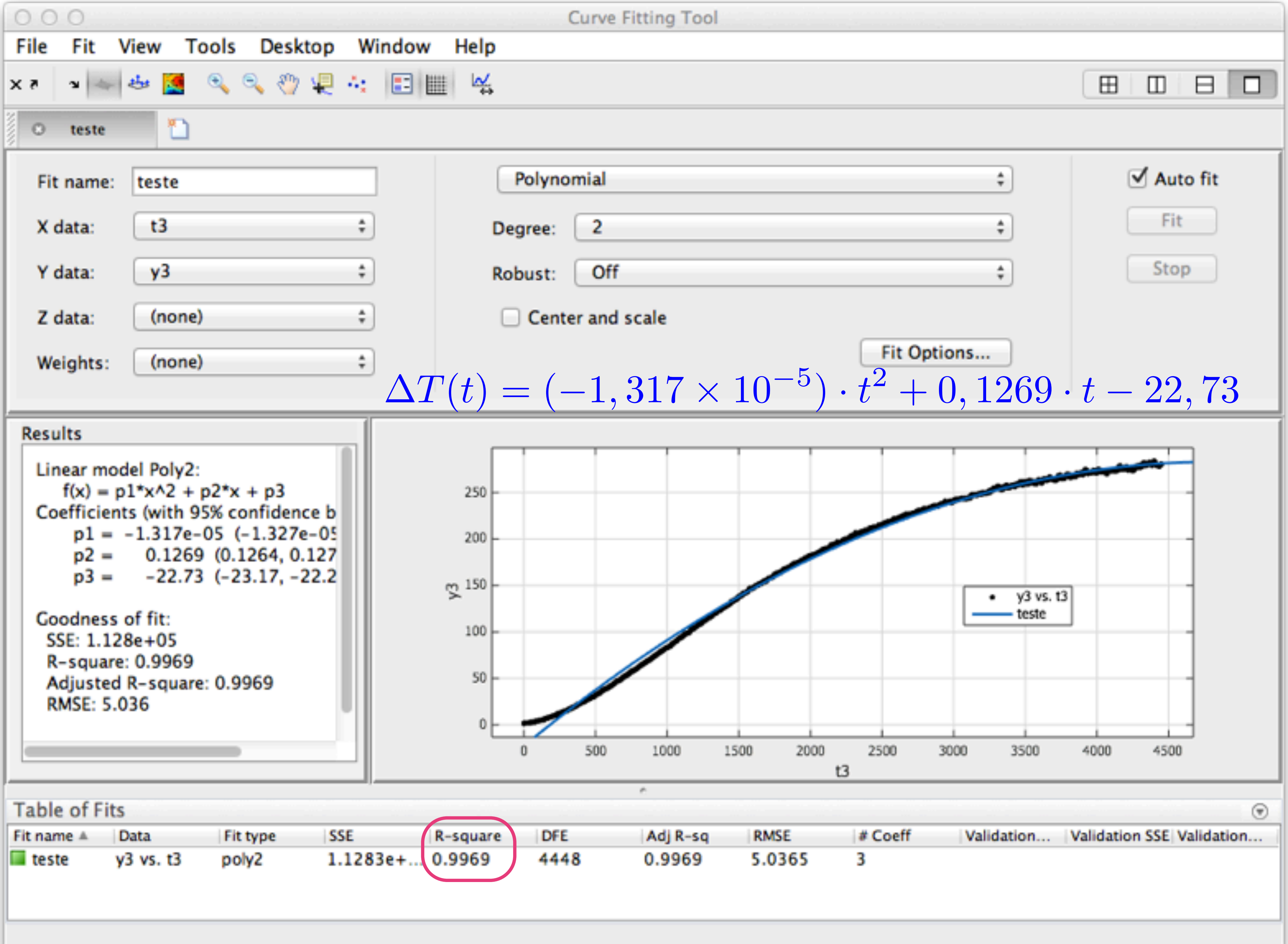
$$\Delta T(t) = (3,17 \times 10^5)e^{-1,537\times 10^{-4}} - (3,17 \times 10^5)e^{-1,542\times 10^{-4}}$$

$$\mathcal{L}\left\{a\,e^{-b\,t} + c\,e^{-d\,t}\right\} = \frac{a}{(s+b)} + \frac{c}{(s+d)}$$

$$\Delta T(s) = \frac{3,17 \times 10^5}{(s + 1,537 \times 10^{-4})} - \frac{3,17 \times 10^5}{(s + 1,542 \times 10^{-4})}$$

$$G(s) = \frac{-1.0049 \times 10^{11}}{(s + 0.0001542)(s + 0.0001537)}$$

Erro. Verificar como foi considerado Y(s)=R(s)*G(s) !


Curve Fitting Tool
File Fit View Tools Desktop Window Help
teste
Fit name: teste
X data: t3
Y data: y3
Z data: (none)
Weights: (none)
Polynomial
Degree: 2
Robust: Off
Center and scale
Fit Options...
Auto fit
Fit
Stop
$\Delta T(t) = (-1,317 \times 10^{-5}) \cdot t^2 + 0,1269 \cdot t - 22,73$
Results
Linear model Poly2:
f(x) = p1*x^2 + p2*x + p3
Coefficients (with 95% confidence b
p1 = -1.317e-05 (-1.327e-05
p2 = 0.1269 (0.1264, 0.127
p3 = -22.73 (-23.17, -22.2
Goodness of fit:
SSE: 1.128e+05
R-square: 0.9969
Adjusted R-square: 0.9969
RMSE: 5.036
y3 vs. t3
teste
y3
t3
Table of Fits
Fit name | Data | Fit type | SSE | R-square | DFE | Adj R-sq | RMSE | # Coeff | Validation... | Validation SSE | Validation...
teste | y3 vs. t3 | poly2 | 1.1283e+... | 0.9969 | 4448 | 0.9969 | 5.0365 | 3

Laplace

$$\Delta T(t) = (-1,317 \times 10^{-5}) \cdot t^2 + 0,1269 \cdot t - 22,73$$

$$\mathcal{L}^{-1}\left\{p_1\, t^2 + p_2\, t + p_3\right\} = \frac{2\, p_1}{s^3} + \frac{p_2}{s^2} + \frac{p_3}{s}$$

Erro: na eq. acima faltou considerar y(t) = 1 + ...