MODELLING AND SIMULATIONS

FILTERING

AALBORG UNIVERSITY
DENMARK

FIR and IIR filters - RECAP

- A **finite impulse response (FIR)** filter is a filter whose impulse response (or response to any finite length input) is of *finite* duration, because it settles to zero in finite time.
- A **infinite impulse response (IIR)** filters, which may have internal feedback and may continue to respond indefinitely
- Advantage/disadvantage – The internal feedback in the IIR filter, reduces the order of the filter compared to a FIR filter, with the same design specs. The FIR filter have a constant group delay, rather the IIR filter have a filter delay dependent on the frequency of the signal.

	FIR	IIR
Computation vs. performance	- Often more computation for the same magnitude response as IIR	+ Less computation for a given magnitude response
Phase	+ Can have exactly linear phase + Other phase responses possi-ble	- Nonlinear phase leads to phase distortion of signal (distortion of "waveshape")
Stability	+ Guaranteed stability	- Must verify stability of final design; no guarantee of stability
Effect of limited number of bits for coefficients and math	+ Noise and errors are generally lower than for IIR	- More sensitive to quantization of coefficients and noise from rounding off calculations
Use of analog filter as model	- No direct conversion from an analog design to FIR (indirect methods exist)	+ Several methods for convert-ing analog designs (this is the most common "pencil and paper" design method)
Arbitrary filter specifications	+ Possible even when no analog equivalent is possible	- (Much) more difficult to pro-duce arbitrary designs
Implementation	+ Straightforward implementa-tion; most DSP hardware sup-ports directly and efficiently	- Large filters tend to be broken down into smaller stages; a bit more complicated than FIR - For optimum performance, careful design of filter stages is necessary
Design	- Optimum designs require com-puter programs (no closed-form solutions) + Some design methods are fairly simple	+ Well-known design processes, can be done manually

AALBORG UNIVERSITY
DENMARK

Bode Plot

- Developed by Hendrik Bode
- Plot magnitude curves using logarithmic scale and phase curves, using a linear scale
  - We present the *magnitude* in dB versus $\log(\omega)$
  - We present the *phase* in dB versus $\log(\omega)$

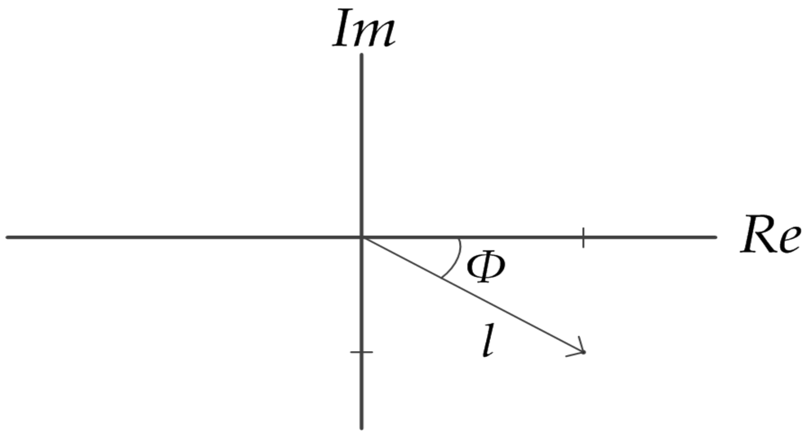
- Some benefits of Bode plots:
  - Evaluate the systems frequency response for a wide range of frequencies
  - Bode plots can be determined experimentally
  - Bode plots of systems in series can be added together
  - Log scale allows for convenient plot of a wide frequency band

- Bode plot by hand
  - Evaluate a system quickly by hand and validate the computers response

Frequency response - Bode Plot

- Frequency analysis (of the previous system):
  - Calculate the steady state frequency response, i.e. $s = j\omega$
  - **Such that $\boldsymbol{H(j\omega) = real + imag \cdot j}$**

- Visualize the whole frequency spectrum, wrt:
  - **Magnitude**: $M(\omega) = |H(j\omega)| = \sqrt{real^2 + imag^2}$
  - **Phase**: $\phi(\omega) = \arg(H(j\omega)) = atan2(imag, real)$



Magnitude to Decibels - Bode Plot

*To calculate the **magnitude** we have to convert into dB*

$relative\ magnitude\ in\ dB = 10 \cdot \log\left(\frac{P_a}{P_b}\right), P_a \,\&\, P_b$ are measures of power of two signals

$$Power = \frac{voltage^2}{resistance}$$

$relative\ magnitude\ in\ dB = 20 \cdot \log\left(\frac{V_a}{V_b}\right), V_a \,\&\, V_b$ are voltages of two signals

$$power \propto Amplitude^2$$

Thus:

$$1\ decibel = 10\log(Amplitude^2) = 20\log(Amplitude)$$

Simple example: Sketch Bode plot by hand

Our system in steady state:

$$\frac{1}{s} \Rightarrow \frac{1}{j\omega}$$

$$Real\ component = 0$$
$$imaginary\ compoent = j\omega$$

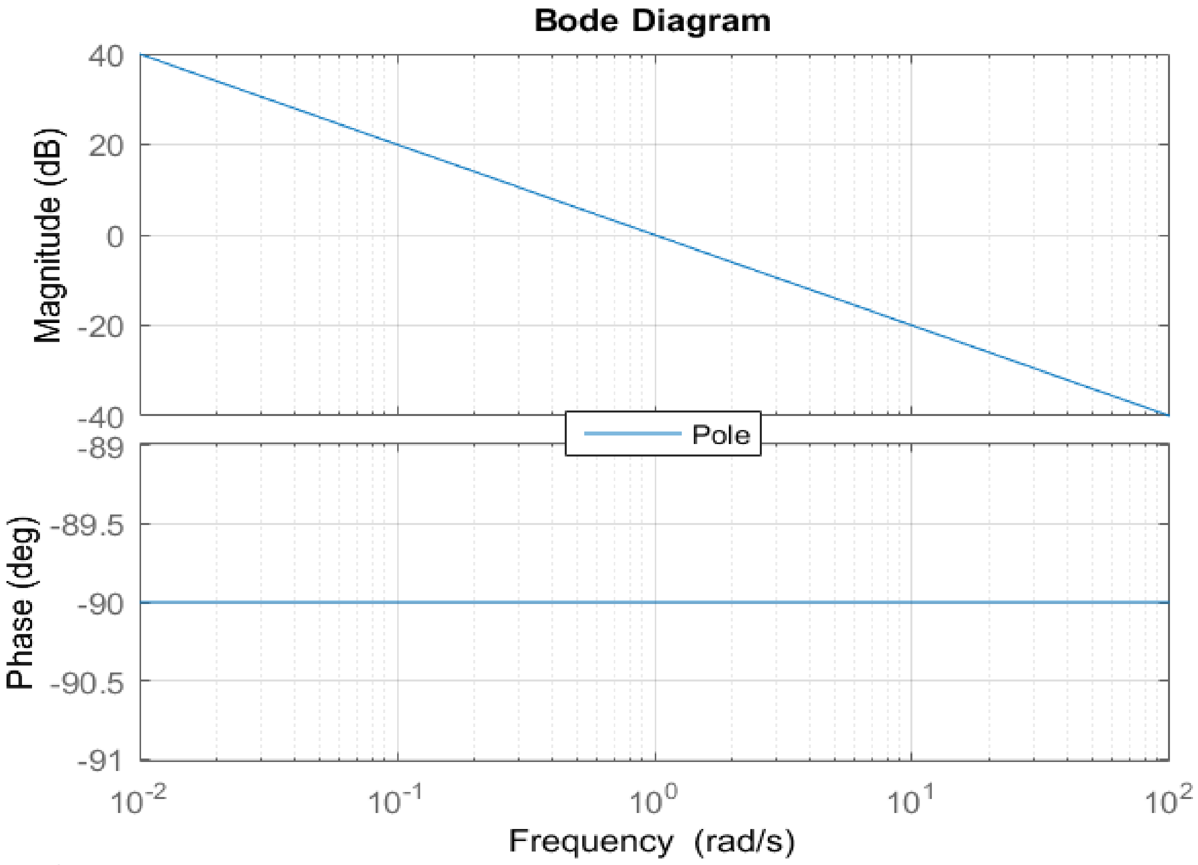
Use the blackboard

AALBORG UNIVERSITY
DENMARK

Pole - Zero Bode plot example

Transfer_function_example.m

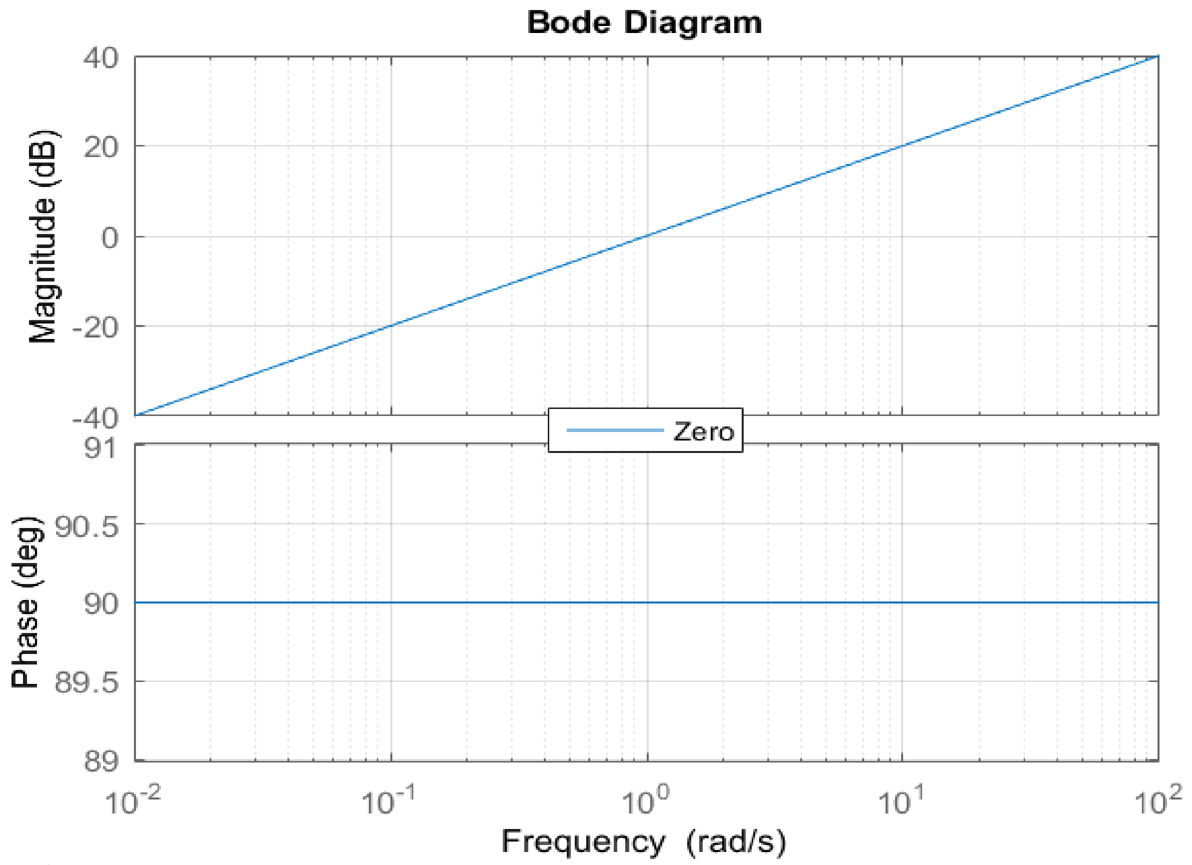
- Single Pole



Pole - Zero Bode plot example

Transfer_function_example.m

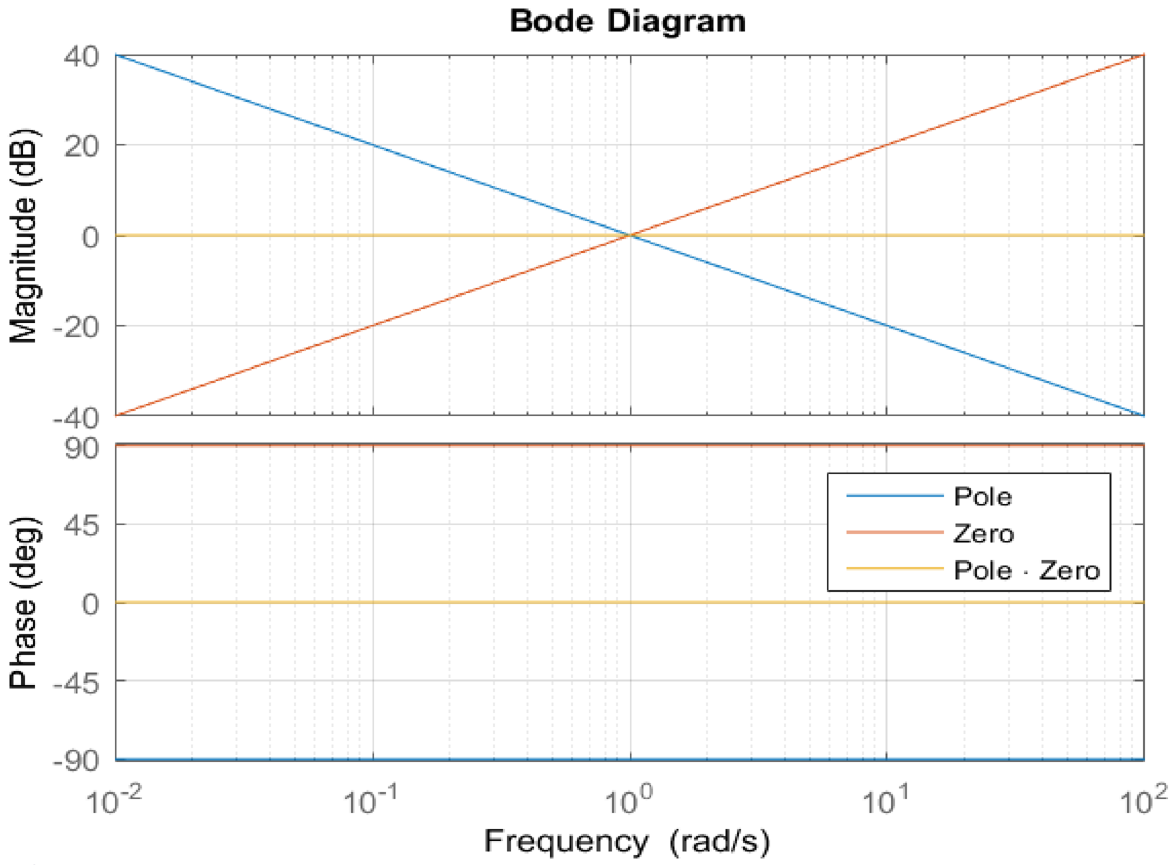
- Single Pole
- Single Zero



Pole - Zero Bode plot example

Transfer_function_example.m

- Single Pole
- Single Zero
- Multiplication of pole and zero



Example: How Bode plot is done in Matlab

Transfer_function_example.m

- Input a sine wave to the system and find the amplitude and phase which is plotted in a Bode Plot
- To gain the entire Bode plot for all desired frequencie

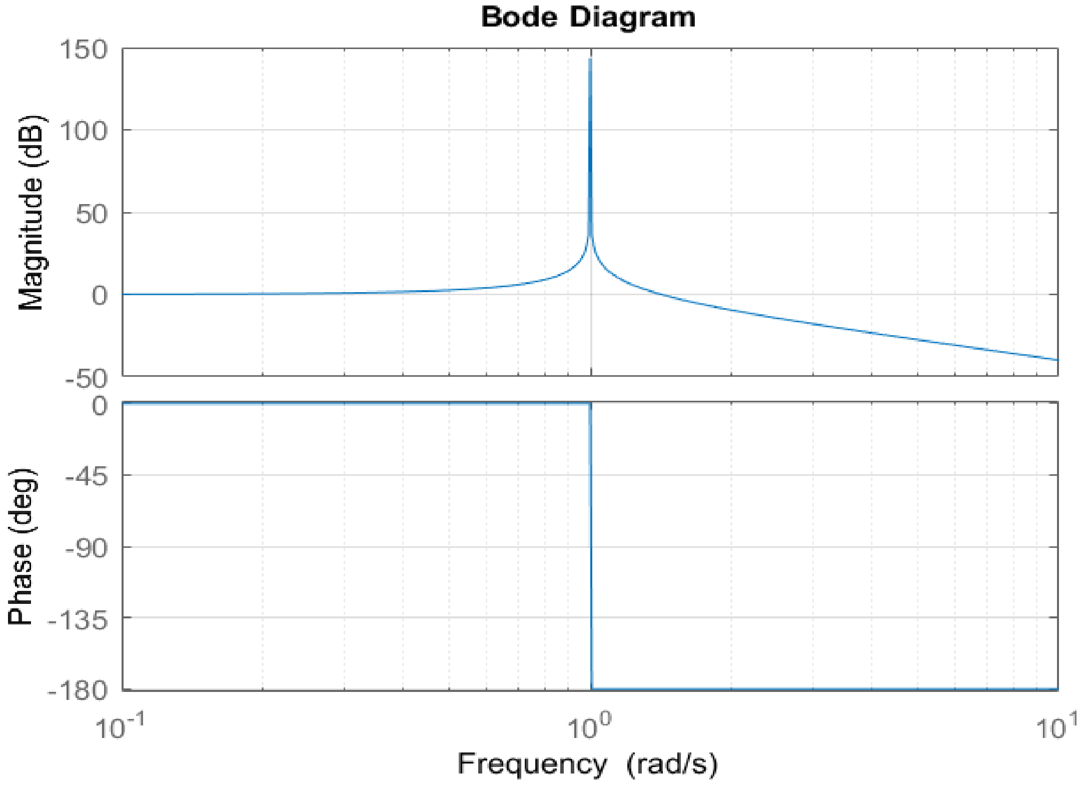
For our example:

$$\frac{1}{ms^2+k}$$

Where:

$m = 2$

$k = 1$



MODELLING AND SIMULATIONS

EXPERIMENT DESIGN & PARAMETER ESTIMATION

AALBORG UNIVERSITY
DENMARK

Intro

- Not all system constants can be modeled using known physical laws
- Some parts have to be identified from data collected from the ‘real’ system
- Data required:
    - Inputs, output, and/or disturbances
- The techniques is called **System Identification**

Parameterized Models

- **Tailor-made model:** constructed from basic physical principles. Unknown parameters have physical interpretation **(*grey-box*)**
- **Ready-made model:** describe the properties of the input-output relationships without any physical interpretation **(*black-box*)**

Model based system identification

- *Grey-box*
  - Can be done by conventional physical experimentation and measurement methods, e.g.,
  - Example for first order system:
    - Estimate the time constant using step response
    - Estimate the DC-gain using steady-state response
- *Black-box*
  - Box-Jenkins (BJ) model
  - Output error (OE) model
  - ARMAX model
  - ARX model

Modeling Phases

- Phase 1
  - Divide system into subsystems, which variables are important and interact, what is the use of the model.
  - Result is a description of the system with eq. block diagrams
- Phase 2
  - Relationship between the variable and constraints in the subsystems are formed
  - Approximations and idealizations are introduced
  - Parts of the model are set to be identified if necessary (eg. unknown dynamics)
- Phase 3
  - Organization of the equations and relationships, can use a computer algebra program (Maple, Matlab, Mathcad…)
  - Prepare the model for simulations
  - End up with eq. a State Space model

Step Response Analysis *or* Transient Analysis

- Through this approach we can gain knowledge about
  - Cause and affect relationships, time delays, time constants (τ) and static gains
- Good industrial practice
- A lot of information is hidden
- Approximated model around a specific operating point
- Requires ability to manipulate the system inputs, and do it sufficiently to achieve a:
  - **Satisfactory Impulse Response**

Experiment example - Transient response

- Often used for system identification

$$Complete\ response\ = natural\ response\ (zero\ input)\ +\ forced\ response\ (zero\ state)$$

- Several possibilities
  - Step response
    - Start from equilibrium and give some step input (eg. Electric motor from 0 to 12V)
  - Impulse response (less used)
    - Give a pulse of energy to the system (eg. "hit it with a hammer")
  - "Natural" response
    - Start from some initial condition away from equilibrium (eg. letting a pendulum fall from some angle)

Step Response Analysis *or* Transient Analysis

Vary the inputs

$$u(t) = u_0, \qquad t < t_0$$
$$u(t) = u_1, \qquad t \geq t_0$$

In the meanwhile measure the outputs:

$$y_0(t), \qquad t < t_0$$
$$y_1(t), \qquad t \geq t_0$$

Or let the inputs have a pule of short duration.

What can we learn from stepping the input

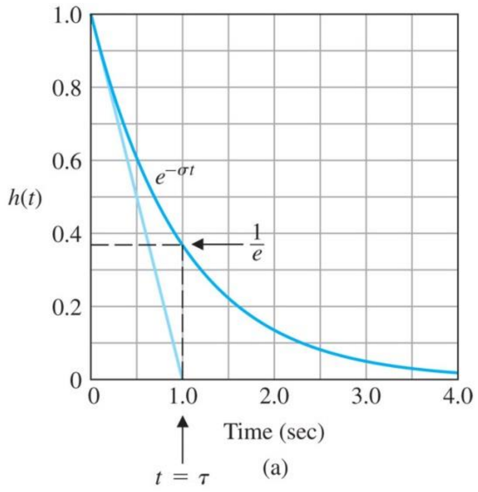
- **Its** effect on the system (i.e. which dynamics does it influence and what is significant to model)
  - Example: *Voltage to a motor on a quadrotor would influence the angle of the body, resulting in it have a specific acceleration, which can be translated into a force through Newtonian laws*.
- Time Constants
  - Which dynamics are most significant
    - Example: *Voltage input to a valve governing a level in a tank; the dynamic of the valve's position is insignificant compared to the level in the tank*
- Characteristics of the system (Oscilaion, damping, monotony…)

1st order model from Step Response

$$G(s) = \frac{K}{s\tau + 1}$$

Give the step input

$$u(t) = \begin{cases} 0, & t < 0 \\ 1, & t \geq 0 \end{cases}$$



The time constant τ is the rate of decay, it is $\tau = t$ when $\frac{1}{e} \approx 0.3679$

If we wish to find the response of a system we search for this point.

The gain of the system is K

Example of Step Response

Find the TF of the following step response.

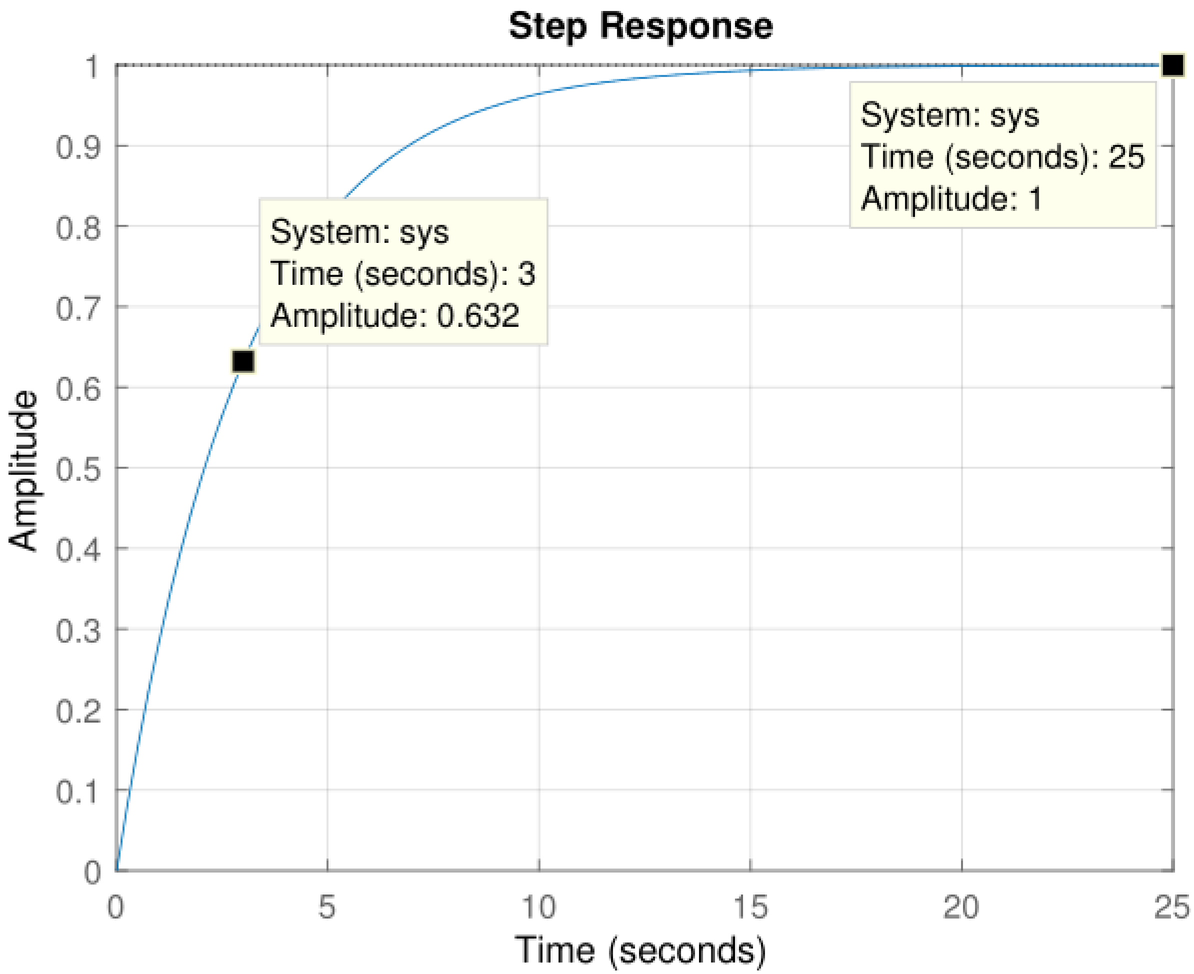
No time delay

Time Constant $\tau = 3$

DC gain $= \ 1$

Thus we can construct our TF as follows:

$$G(s) = \frac{K}{s\tau+1}$$



Example of Step Response

Find the TF of the following step response.

No time delay

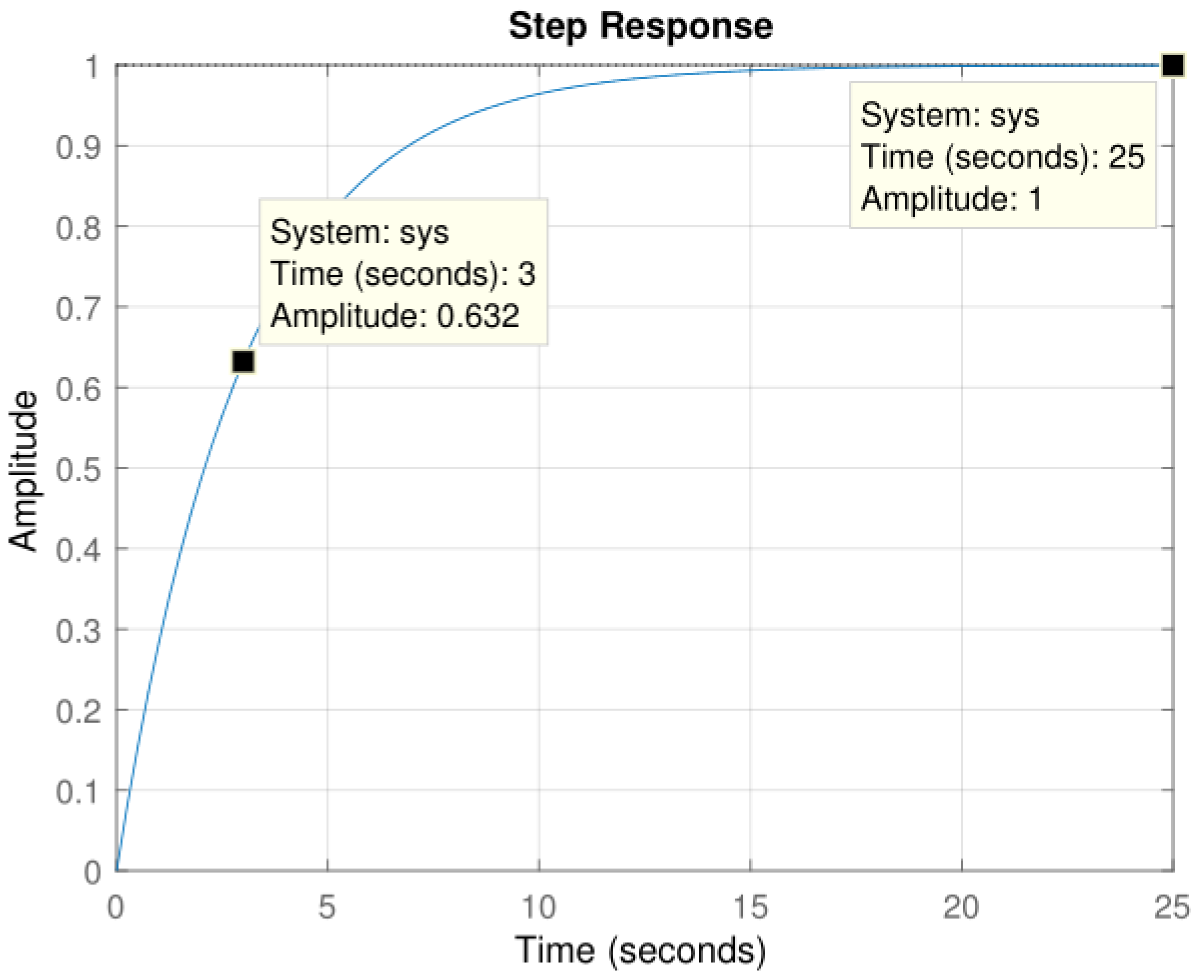
Time Constant $\tau = 3$

DC gain $= 1$

Thus we can construct our TF as follows:

$$G(s) = \frac{K}{s\tau+1}$$

$$G(s) = \frac{1}{3s+1}$$



TAILOR MADE MODELS AND *READY MADE MODELS*

AALBORG UNIVERSITY
DENMARK

Tailor made models

- System constants with unknown values
- Can be one or more

Example

- Valve coefficient
- DC Motor’s moment of inertia

Tailor made models

- The state space model with the unknown parameter vector θ, is represented as:

$$\dot{x}(t) = f(x(t), u(t), \theta)$$
$$y(t) = h(x(t), u(t), \theta)$$

- Where the unknown parameter vector is defined as:

$$\theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}$$

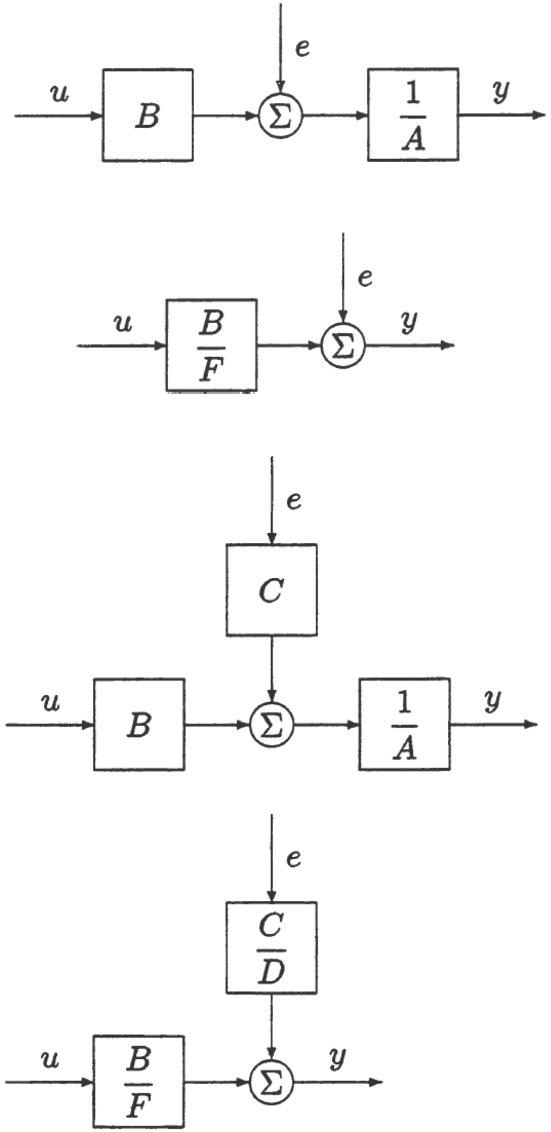
READY MADE MODELS

Ready Made Models

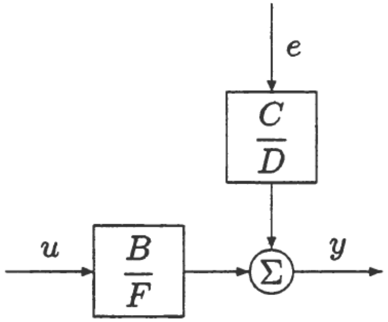
- No knowledge of the physics or
- Too complicated to model using physical relationships
- Find a standard model that fits the dynamics of the data
  - Choose model structure
  - Choose order
  - Preferably linear
- Discrete time models (data is sampled)

Most Common Model Structures

- *ARX*
- *OE (output error model)*
- *ARMAX*
- *BJ*



Box Jenkins Model (BJ)



Box Jenkins Model (BJ)

- General linear time discrete model:

$$y(t) = \eta(t) + w(t)$$

Where $w(t)$ is the **disturbance** and $\eta(t)$ is the **noise–free output:**

$$\eta(t) = G(q, \theta)u(t)$$

And $G(q, \theta)$ is a rational function

$$G(q, \theta) = \frac{B(q)}{F(q)} = \frac{b_1 q^{-nk} + b_2 q^{-nk-1} + \cdots + b_{nb} q^{-nk-nb+1}}{1 + f_1 q^{-1} + \cdots + f_{nf} q^{-nf}}$$

Where q is the shift operator,

BJ model

The disturbance term can be written as:

$$w(t) = H(q, \theta)e(t)$$

Where $\mathrm{H}(q, \theta)$:

$$H(q, \theta) = \frac{C(q)}{D(q)} = \frac{1 + c_1 q^{-1} + \cdots + c_{nc} q^{-nc}}{1 + d_1 q^{-1} + \cdots + d_{nd} q^{-nd}}$$

Where $e(t)$ is white noise and the final model can be summarized to:

$$y(t) = G(q, \theta)u(t) + H(q, \theta)e(t)$$

Where θ contains the parameters of the transfer functions: b_i, c_i, d_i, f_i

This ready-made model is described by 5 structural parameters nb, nc, nd, nf, nk, which are to be chosen And the parameters of θ are to be adjusted to collected data.

OJ model

Disturbance signals are not modeled

The noise model $H(q) \equiv 1, \qquad i.e \qquad nc = nd = 0$

And Thus

- *The noise is equal to the error i.e.:* $e(t) = w(t)$*, which will be the difference (error) between the actual output and the noise-free output*

ARMAX model

If he same denominator is used for both the $G\ and\ H$, than:

$$F(q) = D(q) = A(q) = 1 + a_1 q^{-1} + \cdots + a_{na} a^{-na}$$

Multiplying out by $A(q)$:

$$A(q)y(t) = B(q)u(t) + C(q)e(t)$$

Where$A(q)y(t)$ represent the **A**uto **R**egression and $C(q)e(t)$ a **M**oving **A**verage of the white noise and $B(q)u(t)$ represents the e**X**tra input or the exogenous variable

Good choice if the dominating disturbances enter together with the input

ARX model

Here $C(q) \equiv 1, \quad i.e. \quad nc = 0$

We have:

$$A(q)y(t) = B(q)u(t) + e(t)$$

Model construction

- When a model structure has been picked the orders of na, nb, nc, nd, nf, nk is decided

Pretreatment of Data

Sampling Interval

Filtering of data

Offset

Pretreatment of Data

Sampling Interval

- choose sufficient sampling interval
  - From frequency analysis find the bandwidth of the signal (FFT)

Pretreatment of Data

Sampling Interval

Filtering of data

Make an FFT of the signal, and use this as a baseline for filtering the signal

Design the filter and apply to your signal:

$$y_F(t) = L(q)y(t), \qquad u_F(t) = L(q)y(t)$$

Where the filter $L(q)$ is applied on both the input and output data

Be aware, The filter dynamics will influence your models dynamics

- Carefully design the filter
- Only filter the necessary frequencies, not too low:
    - Risk of removing some important system features
    - If the filtering too severe, some responses will be damped/prolonged

Be aware of filter delay!

- Not important for offline analysis but very important for filters used in control solution

Pretreatment of Data

Sampling Interval

Filtering of data

Offset

- Remove means
- Add the offset when simulating the data

Closed Loop System

- Avoid Closed loop systems
- In some cases this is unavoidable
  - Eg. Control of valve position, from signal input
  - Make sure that the dynamics of the controller are far faster than the ones of the system characteristic which you wish to model

Choice of Model Order

- This can be done automatically in the software
- Be aware of too high order, may not be necessary
  - The noise can be modeled by the higher orders
- Use pole zero plot to check for cancelations
  - Surviving poles and zeros determine the necessary order to describe the dynamics

Fitness of Data

This is a guess, or a prediction of $y(t)$ at time $(t-1)$

$$\hat{y}(t|\theta)$$

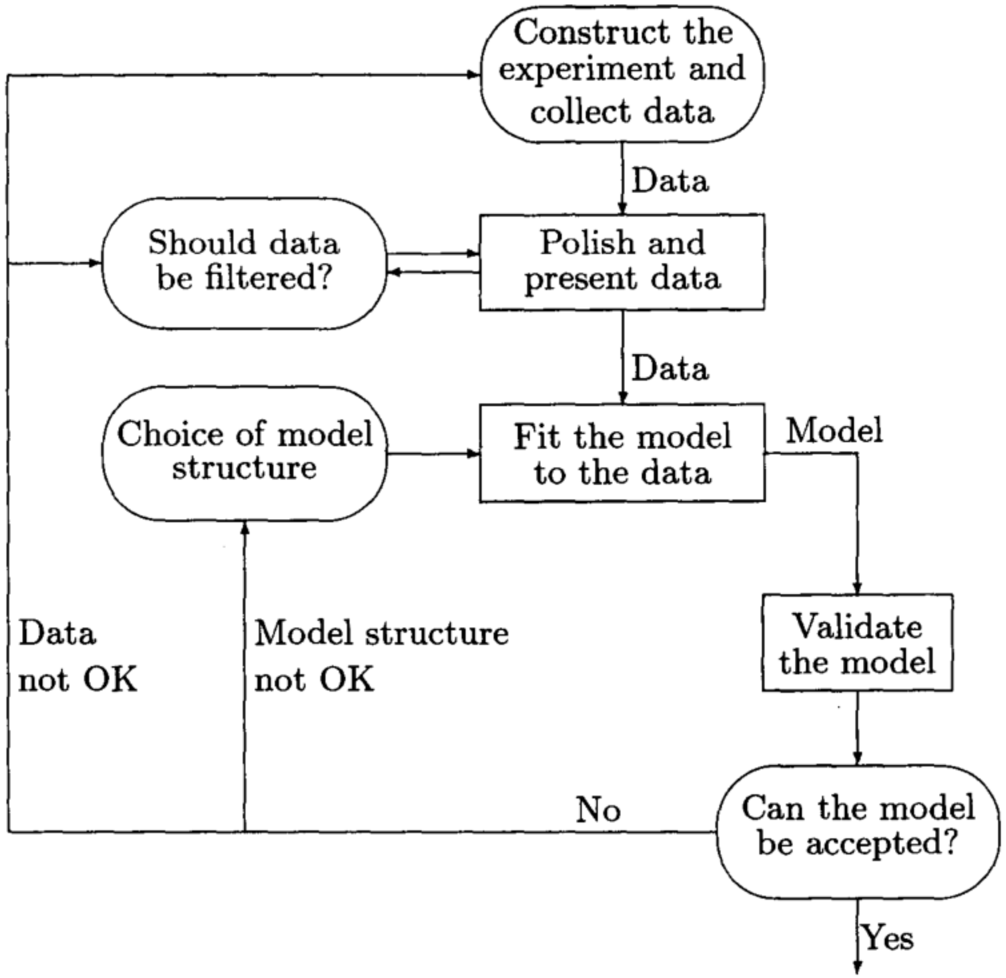
Where θ is the parameter vector which contains the parameters of the identified model, where parameters of θ are to be adjusted to collected data.

Then we can calculate the prediction error

$$\epsilon(t,\theta) = y(t) - \hat{y}(t|\theta)$$

Using SYS ID Software

1. Specify model structure
2. Best model fit is calculated by the software
3. Evaluate model using inbuilt function in the toolbox



AALBORG UNIVERSITY
DENMARK

System Identification Toolbox

System identification toolbox:

IDENT

Provides:

A. Handling of data, plotting, and the like:

Filtering of data, removal of drift, choice of data segments, and so on

B. Nonparametric identification methods

Estimation of covariances, Fourier transforms, correlation and spectral analysis, and so on.

C. Parametric estimation methods **:**

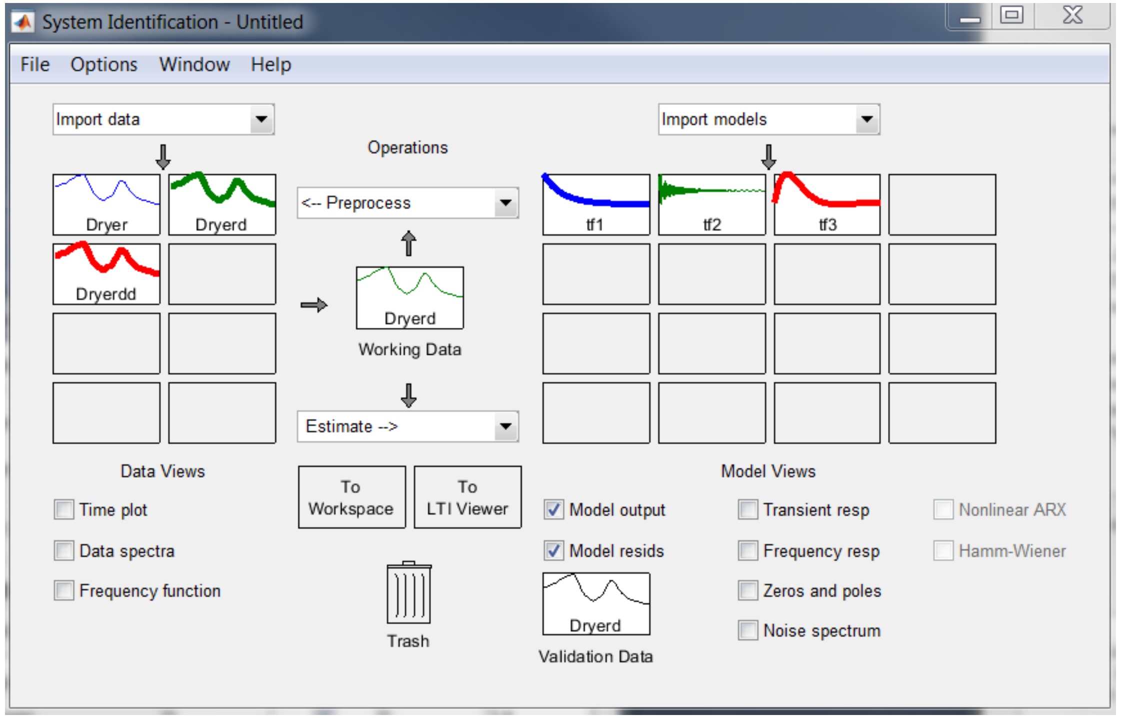
Calculation of parametric estimates in different model structures

D. Presentation of models:

Simulation of models, estimation and plotting of poles and zeros, computation of frequency functions and plotting in Bode diagrams, and so on

E. Model validation:

Computation and analysis of residuals $\left(\epsilon(t, \hat{\theta}_N)\right)$ comparison between different models' properties, and the like



AALBORG UNIVERSITY DENMARK

Prepare Data for Toolbox

```
data = iddata(y,u,Ts)
```

Creates an `iddata` object containing a time-domain output signal `y` and input signal `u`, respectively. `Ts` specifies the sample time of the experimental data.

Estimation Result

- Example of a Transfer Function Estimation Result from the System Identification Toolbox

```
Transfer Function Identification
Estimation data: Time domain data Dryerd
Data has 1 outputs, 1 inputs and 1000 samples.
Number of poles: 1, Number of zeros: 0
Initialization Method: "iv"
```

Estimation Progress

```
Initializing model parameters...
Initializing using 'iv' method...
done.

Initialization complete.

Nonlinear least squares with automatically chosen line search method
----------------------------------------------------------------------------------------
                            Norm of      First-order      Improvement (%)
Iteration      Cost       step         optimality    Expected  Achieved   Bisections
----------------------------------------------------------------------------------------
    0       0.159458         -            1.19     0.00056         -         -
    1       0.159458  0.000732        0.00865     0.00056   8.92e-05         0
----------------------------------------------------------------------------------------
Estimating parameter covariance...
done.
```

Result

```
Termination condition: Near (local) minimum, (norm(g) < tol).
Number of iterations: 1, Number of function evaluations: 3

Status: Estimated using TFEST
Fit to estimation data: 51.85%, FPE: 0.160417
```

Model Validation

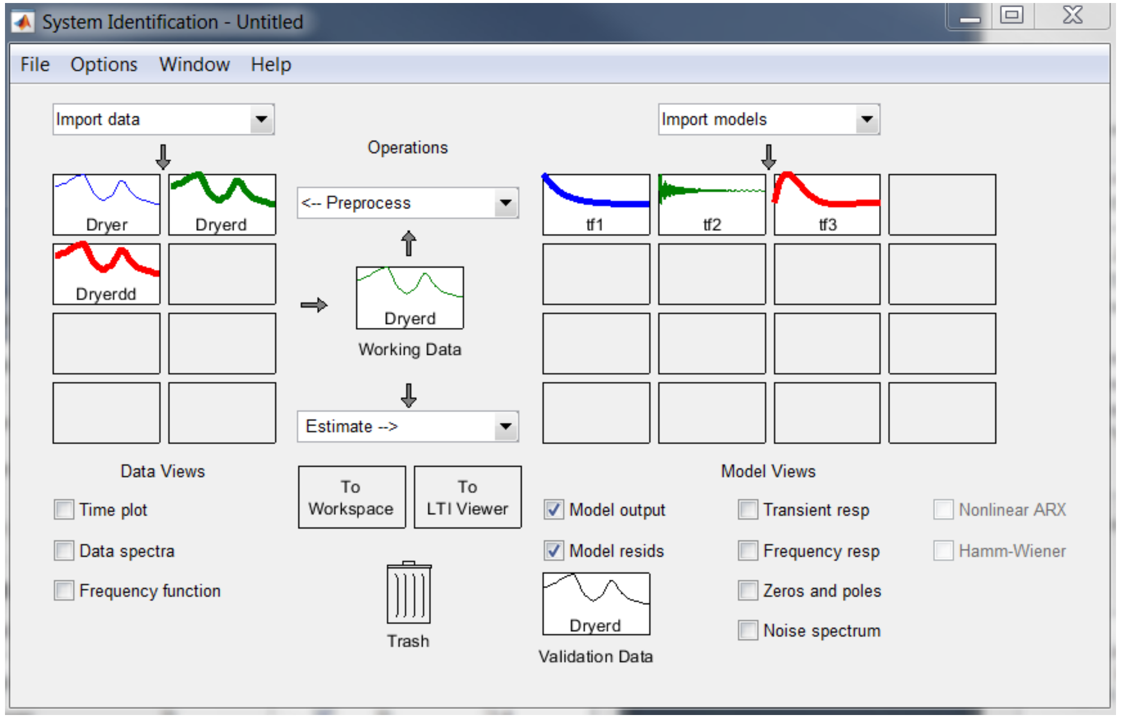
1. Model Quality
    1. Stability (Bode analysis, frequency function)
    2. Ability to reproduce system behavior
        1. Input output behavior comparison (simulated to real(new data set))
        2. k-step prediction (prediction horizon k larger than τ)
            1. Can be done using the residuals (`resid` ‘1 step ahead prediction errors’)
2. Residual Analysis
3. Is the model working well
    1. Simulations
    2. Analysis
    3. Control Design
    4. …

Practical Session

- Identify a systems Transfer Function in the System Identification Toolbox:

IDENT

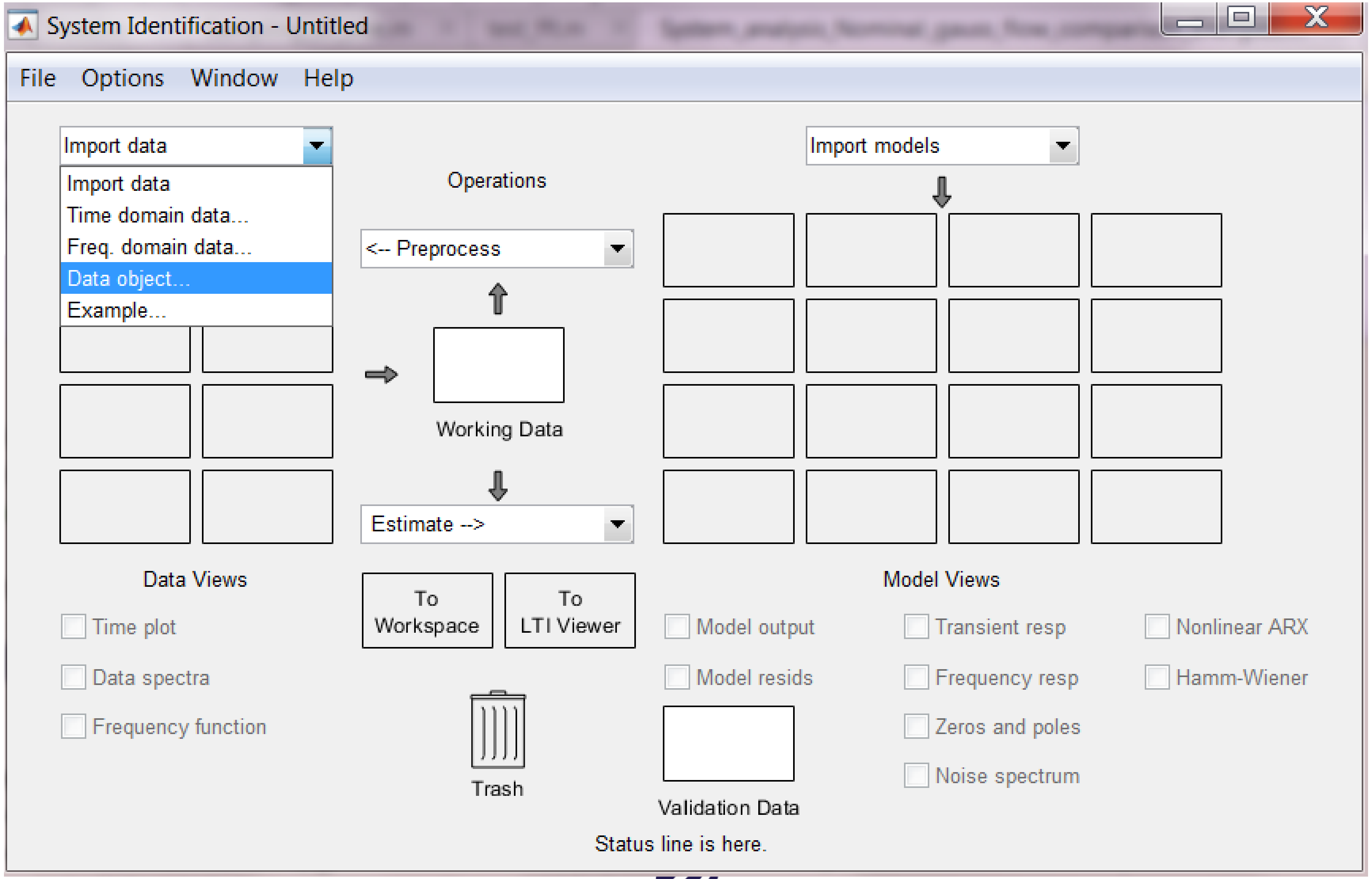
- We import a data set
- Pretreat the data
- And crate a transfer function of a order of our choice
- Analyze the data



AALBORG UNIVERSITY
DENMARK

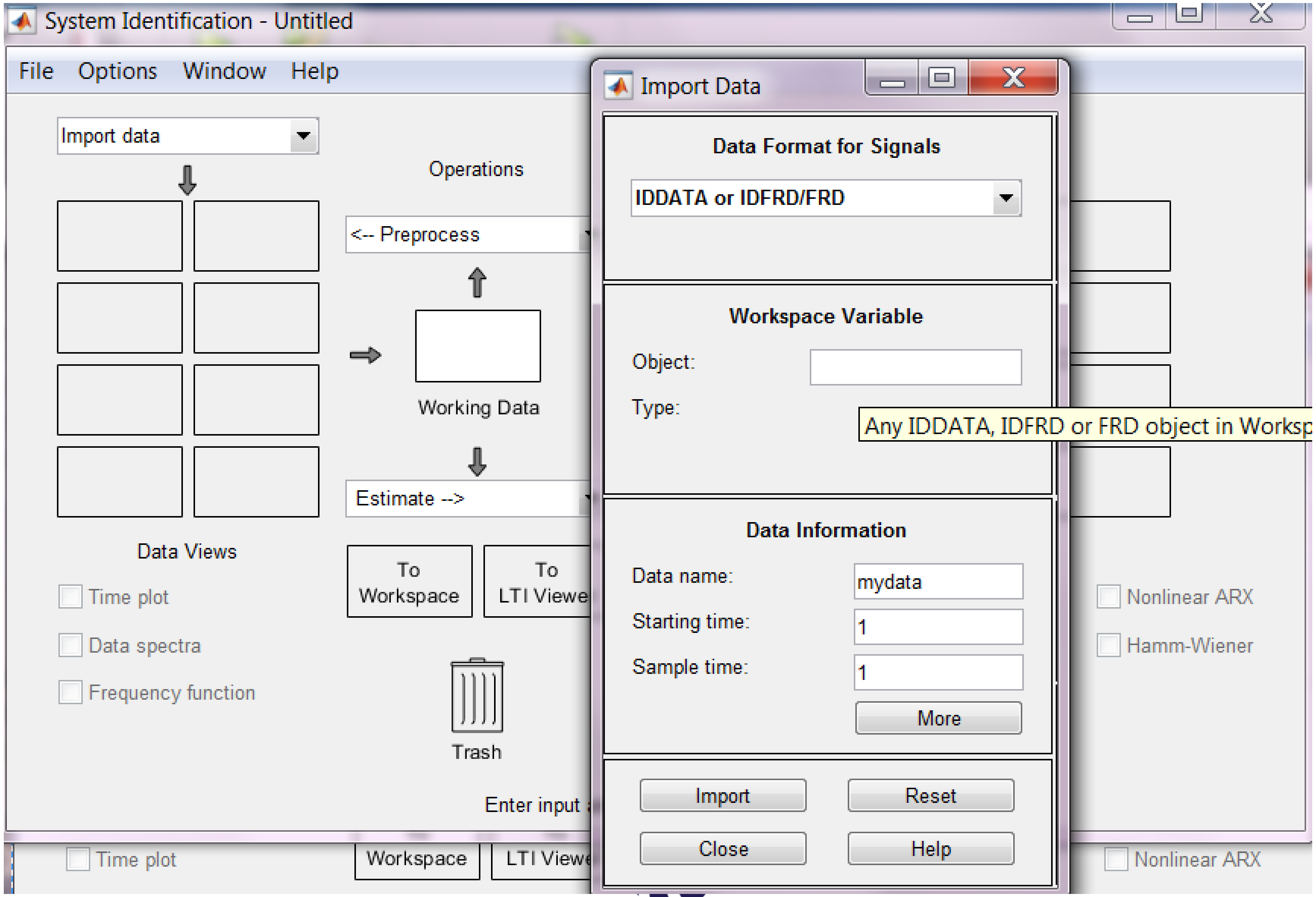
..

- ...


System Identification - Untitled
File
Options
Window
Help
Import data
Import data
Time domain data...
Freq. domain data...
Data object...
Example...
Operations
<-- Preprocess
Working Data
Estimate -->
Import models
Data Views
Time plot
Data spectra
Frequency function
To Workspace
To LTI Viewer
Trash
Validation Data
Model Views
Model output
Model resids
Transient resp
Frequency resp
Zeros and poles
Noise spectrum
Nonlinear ARX
Hamm-Wiener
Status line is here.

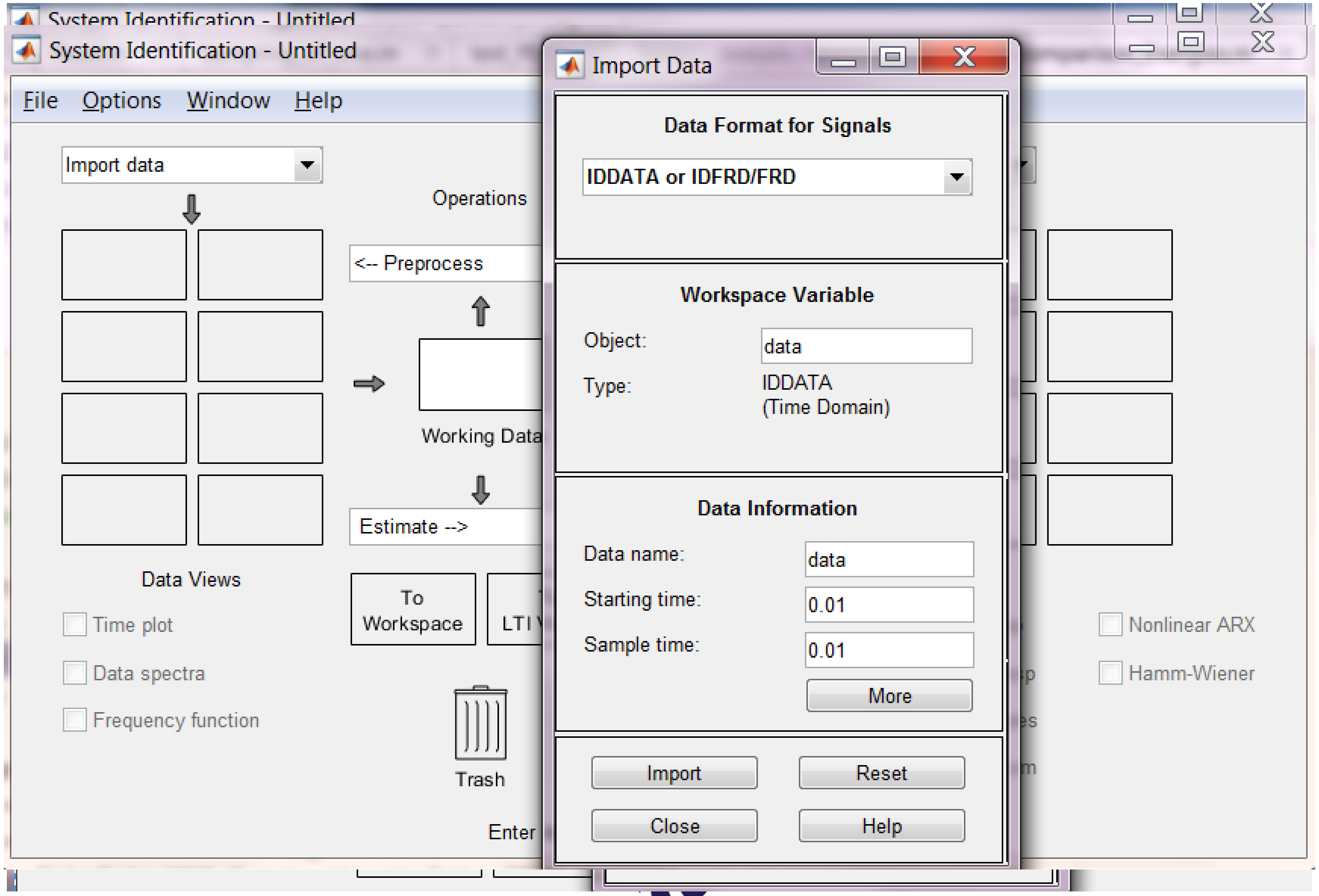
..

- ...


System Identification - Untitled
File
Options
Window
Help
Import data
Operations
<-- Preprocess
Working Data
Estimate -->
Data Views
To Workspace
To LTI Viewer
Time plot
Data spectra
Frequency function
Trash
Enter input
Nonlinear ARX
Hamm-Wiener
Import Data
Data Format for Signals
IDDATA or IDFRD/FRD
Workspace Variable
Object:
Type:
Any IDDATA, IDFRD or FRD object in Worksp
Data Information
Data name:
mydata
Starting time:
1
Sample time:
1
More
Import
Reset
Close
Help

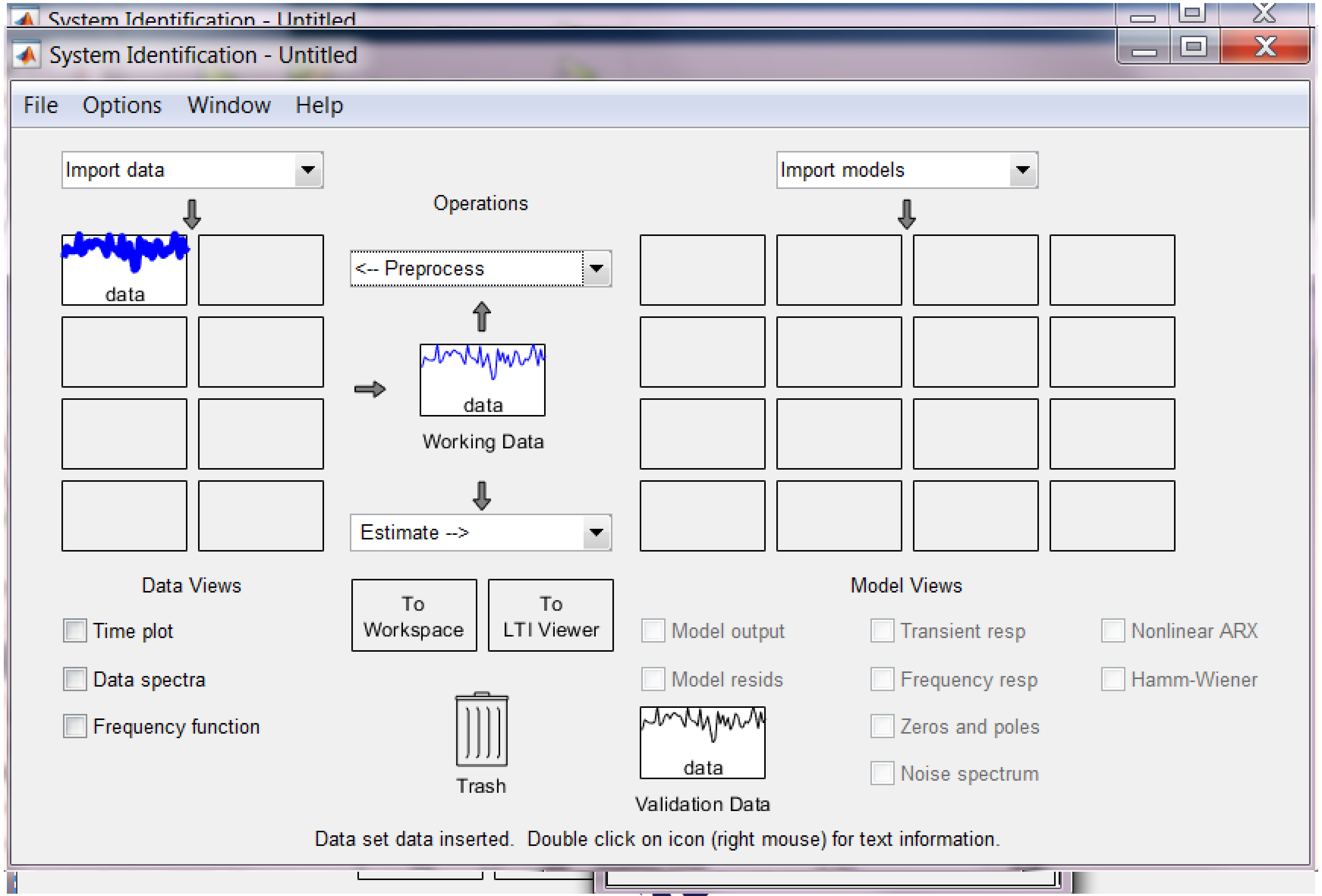
..

- ...


System Identification - Untitled
File
Options
Window
Help
Import data
Operations
<-- Preprocess
Working Data
Estimate -->
Data Views
Time plot
Data spectra
Frequency function
To Workspace
Trash
Enter
Nonlinear ARX
Hamm-Wiener
Import Data
Data Format for Signals
IDDATA or IDFRD/FRD
Workspace Variable
Object:
data
Type:
IDDATA
(Time Domain)
Data Information
Data name:
data
Starting time:
0.01
Sample time:
0.01
More
Import
Reset
Close
Help

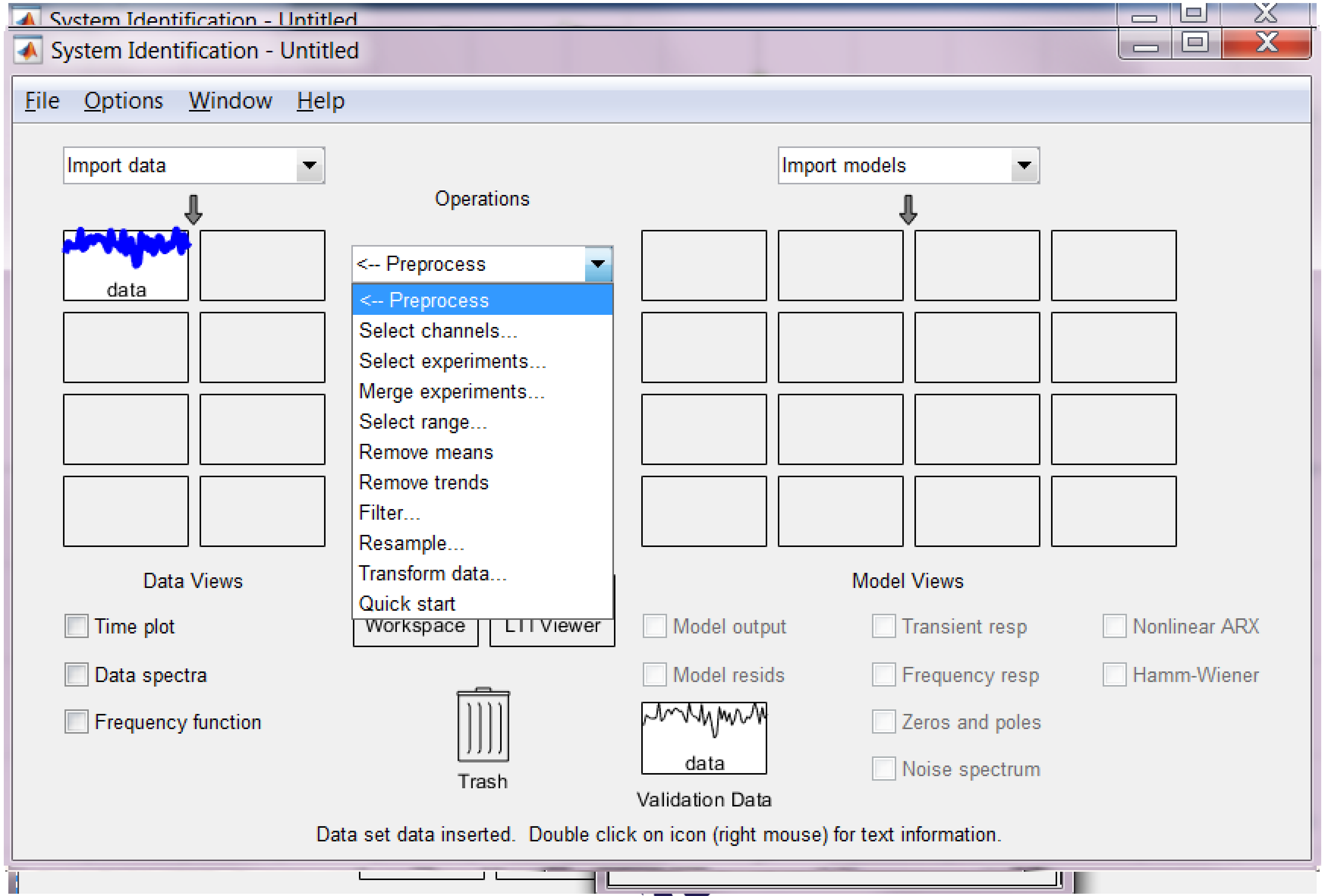
..

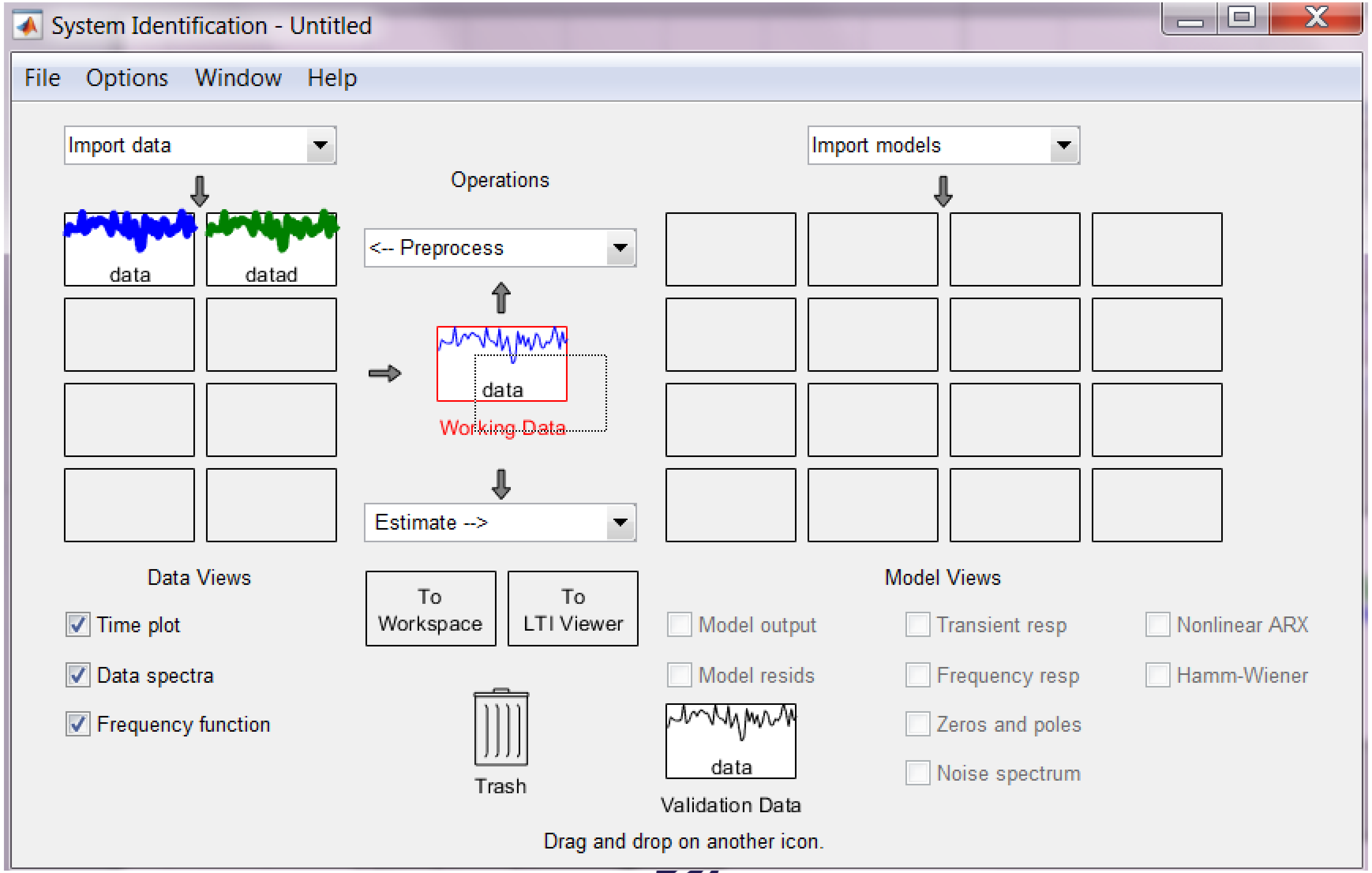
- ...


System Identification - Untitled
System Identification - Untitled
File
Options
Window
Help
Import data
Operations
Import models
data
<-- Preprocess
data
Working Data
Estimate -->
Data Views
Time plot
Data spectra
Frequency function
To Workspace
To LTI Viewer
Trash
data
Validation Data
Model Views
Model output
Model resids
Transient resp
Frequency resp
Zeros and poles
Noise spectrum
Nonlinear ARX
Hamm-Wiener
Data set data inserted. Double click on icon (right mouse) for text information.

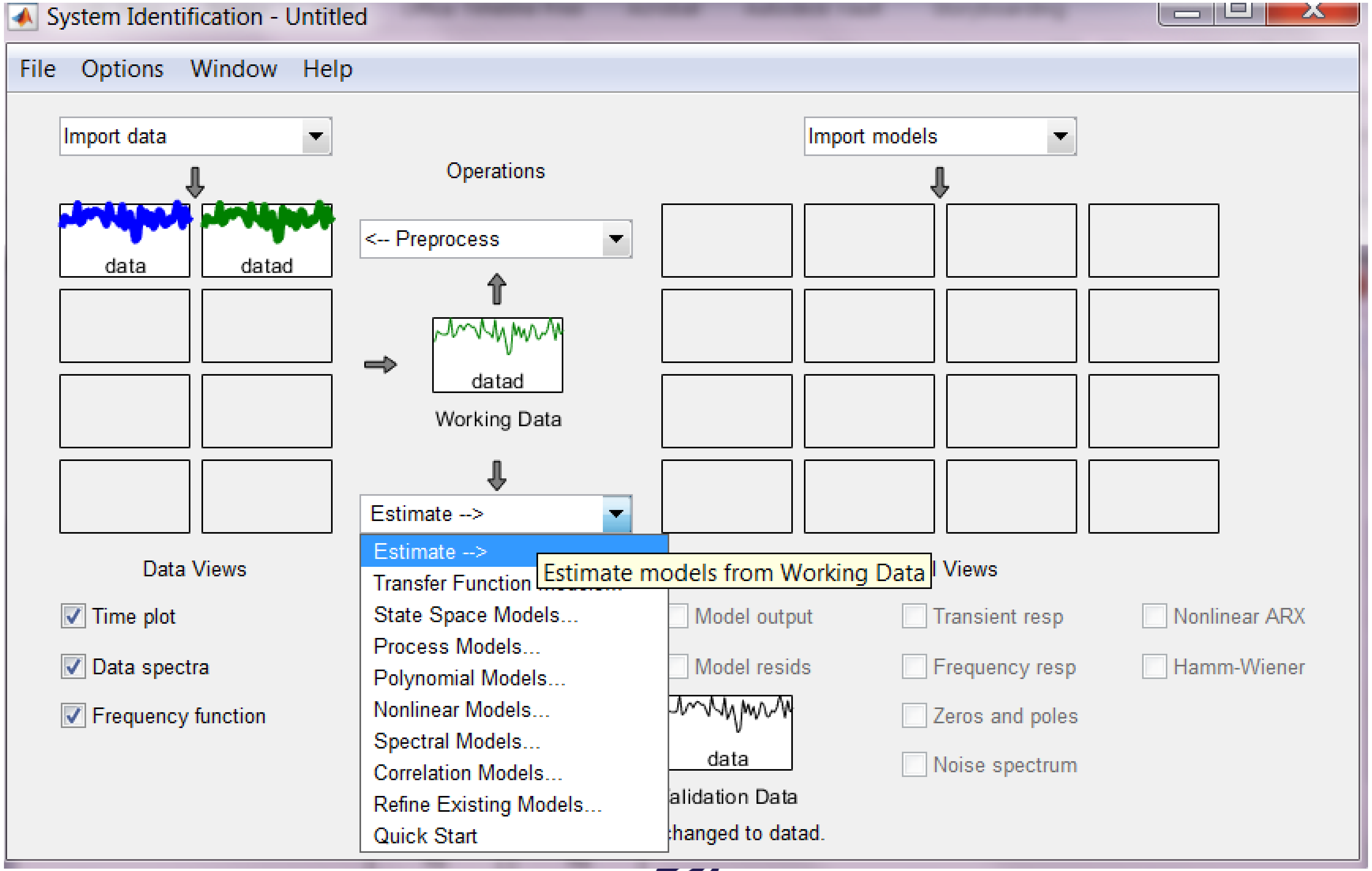
..

- ...

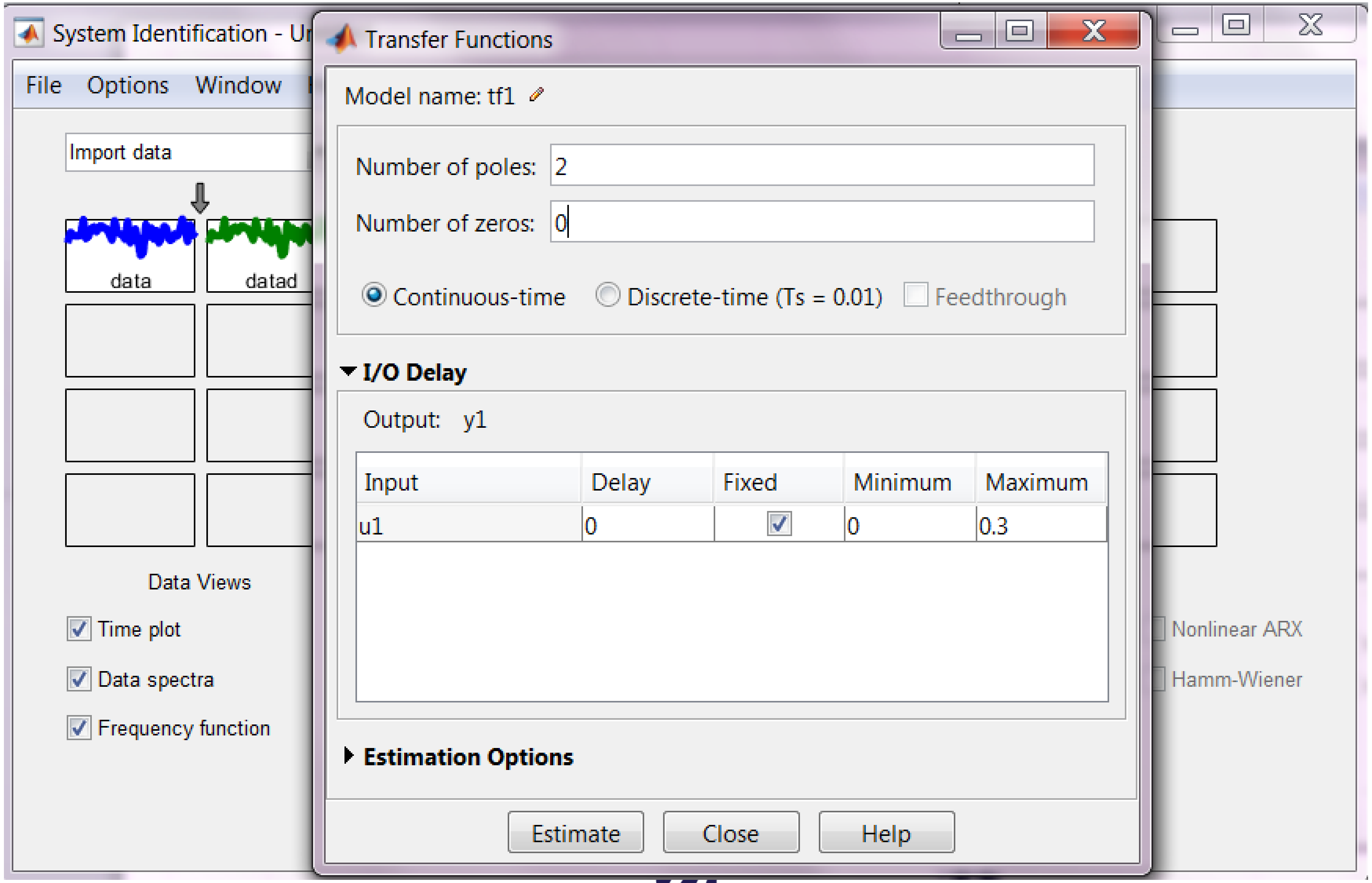

System Identification - Untitled
File Options Window Help
Import data
Import models
Operations
data
<-- Preprocess
<-- Preprocess
Select channels...
Select experiments...
Merge experiments...
Select range...
Remove means
Remove trends
Filter...
Resample...
Transform data...
Quick start
Workspace
LTI Viewer
Data Views
Model Views
Time plot
Data spectra
Frequency function
Trash
Model output
Model resids
Transient resp
Frequency resp
Zeros and poles
Noise spectrum
Nonlinear ARX
Hamm-Wiener
data
Validation Data
Data set data inserted. Double click on icon (right mouse) for text information.



AALBORG UNIVERSITY
DENMARK



AALBORG UNIVERSITY
DENMARK


System Identification - U
File
Options
Window
Import data
data
datad
Data Views
Time plot
Data spectra
Frequency function
Nonlinear ARX
Hamm-Wiener
Transfer Functions
Model name: tf1
Number of poles: 2
Number of zeros: 0
Continuous-time
Discrete-time (Ts = 0.01)
Feedthrough
I/O Delay
Output: y1
Input
Delay
Fixed
Minimum
Maximum
u1
0
0
0.3
Estimation Options
Estimate
Close
Help

Plant Identification Progress

```
Transfer Function Identification
Estimation data: Time domain data datad
Data has 1 outputs, 1 inputs and 8001 samples.
Number of poles: 2, Number of zeros: 0, I/O delay: 0.1
Initialization Method: "iv"
```

Estimation Progress

8	0.000775517	1.05e+04	4.05e+04	157	0.0834	0
9	0.000775254	5.69e+03	3.89e+05	47	0.0339	0
10	0.000774933	1.72e+04	5.49e+05	37.7	0.0414	0
11	0.000774678	2.34e+04	2.02e+05	31	0.0329	0
12	0.000774616	59.2	8.51e+05	15.9	0.00795	0
13	0.000774519	1.99e+04	1.57e+06	25.8	0.0125	0
14	0.000774372	59.1	9.93e+05	18.7	0.019	0
15	0.000774314	8.77e+03	9.37e+05	12.5	0.00757	0
16	0.000774294	839	7.54e+04	4.23	0.00253	0
17	0.000774289	2.12e+03	1.59e+05	1.6	0.000699	0
18	0.000774288	532	1.56e+04	0.383	0.000142	0
19	0.000774287	570	3.3e+04	0.133	5.05e-05	0
20	0.000774287	208	8.03e+03	0.036	1.27e-05	0

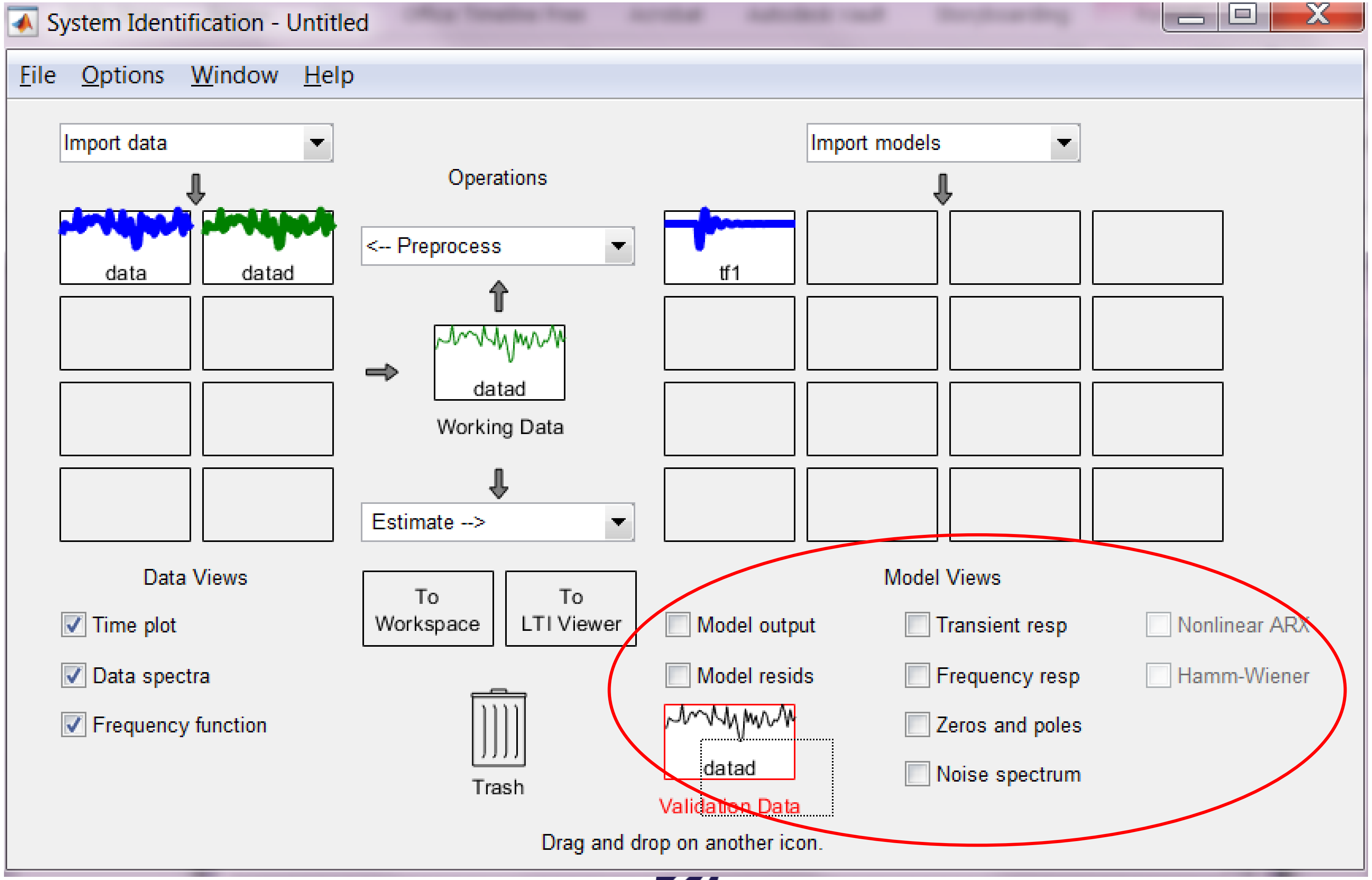
```
Estimating parameter covariance...
done.
```

Result

```
Termination condition: Maximum number of iterations reached.
Number of iterations: 20, Number of function evaluations: 50

Status: Estimated using TFEST
Fit to estimation data: 51.84%, FPE: 0.000775255
```

Stop   Close


System Identification - Untitled
File
Options
Window
Help
Import data
Import models
Operations
data
datad
<-- Preprocess
tf1
datad
Working Data
Estimate -->
Data Views
Model Views
To Workspace
To LTI Viewer
Time plot
Data spectra
Frequency function
Model output
Model resids
Transient resp
Frequency resp
Zeros and poles
Noise spectrum
Nonlinear ARX
Hamm-Wiener
Trash
datad
Validation Data
Drag and drop on another icon.