2

The Graphical User Interface

The Big Picture 2-2

Handling Data 2-7

Estimating Models 2-15

Examining Models 2-28

Some Further GUI Topics 2-35

The Big Picture

The System Identification Toolbox provides a graphical user interface (GUI). The GUI covers most of the toolbox's functions and gives easy access to all variables that are created during a session. It is started by typing

```
ident
```

in the MATLAB command window.

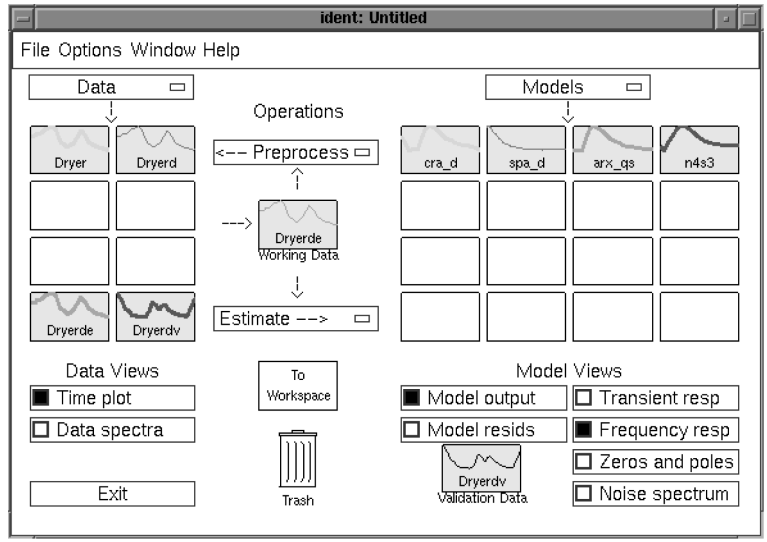

Figure 2-1: The Main ident Information Window

The Model and Data Boards

System Identification is about data and models and creating models from data. The main information and communication window **ident**, is therefore dominated by two tables:

- A table over available data sets, each represented by an icon
- A table over created models, each represented by an icon

These tables will be referred to as the *Model Board* and the *Data Board* in this chapter. You enter data sets into the Data Board by:

- Opening earlier saved sessions.
- Importing them from the MATLAB workspace.
- Creating them by detrending, filtering, selecting subsets, etc., of another data set in the Data Board.

Imports are handled under the pop-up menu **Data** while creation of new data sets is handled under the pop-up menu **Preprocess**. “Handling Data” on page 2-7 deals with this in more detail.

The models are entered into the summary board by:

- Opening earlier saved sessions.
- Importing them from the MATLAB workspace.
- Estimating them from data.

Imports are handled under the pop-up menu **Models**, while all the different estimation schemes are reached under the pop-up menu **Estimate**. More about this in “Estimating Models” on page 2-15.

The Data and Model Boards can be rearranged by dragging and dropping. More boards open automatically when necessary or when asked for (under menu **Options**).

The Working Data

All data sets and models are created from the Working Data set. This is the data that is given in the center of the **ident** window. To change the Working Data set drag and drop any data set from the Data Board on the Working Data icon.

The Views

Below the Data and Model Boards are buttons for different views. These control what aspects of the data sets and models you would like to examine, and are described in more detail in “Handling Data” on page 2-7 and in “Examining Models” on page 2-28. To select a data set or a model, so that its properties are displayed, click on its icon. A selected object is marked by a thicker line in the icon. To deselect, click again. An arbitrary number of data/model objects can be

examined simultaneously. To have more information about an object, double-click (or right-click or **Ctrl**-click) on its icon.

The Validation Data

The two model views **Model Output** and **Model Residuals** illustrate model properties when applied to the Validation Data set. This is the set marked in the box below these two views. To change the Validation Data, drag and drop any data set from the Data Board on the Validation Data icon.

It is good and common practice in identification to evaluate an estimated model's properties using a "fresh" data set, that is, one that was not used for the estimation. It is thus good advice to let the Validation Data be different from the Working Data, but they should of course be compatible with these.

The Work Flow

You start by importing data (under pop-up menu **Data**); you examine the data set using the **Data Views.** You probably remove the means from the data and select subsets of data for estimation and validation purposes using the items in the pop-up menu **Preprocess**. You then continue to estimate models, using the possibilities under the pop-up menu **Estimate**, perhaps first doing a quickstart. You examine the obtained models with respect to your favorite aspects using the different **Model Views**. The basic idea is that any checked view shows the properties of all selected models at any time. This function is "live" so models and views can be checked in and out at will in an online fashion. You select/deselect a model by clicking on its icon.

Inspired by the information you gain from the plots, you continue to try out different model structures (model orders) until you find a model you are satisfied with.

Management Aspects

Diary: It is easy to forget what you have been doing. By double-clicking on a data/model icon, a complete diary will be given of how this object was created, along with other key information. At this point you can also add comments and change the name of the object and its color.

Layout: To have a good overview of the created models and data sets, it is good practice to try rearranging the icons by dragging and dropping. In this way models corresponding to a particular data set can be grouped together, etc. You

can also open new boards (**Options** menu **Extra model/data boards**) to further rearrange the icons. These can be dragged across the screen between different windows. The extra boards are also equipped with notepads for your comments.

Sessions: The Model and Data Boards with all models and data sets together with their diaries can be saved (under menu item **File**) at any point, and reloaded later. This is the counterpart of save/load workspace in the command-driven MATLAB. The four most recent sessions are listed under **File** for immediate open.

Cleanliness: The boards will hold an arbitrary number of models and data sets (by creating clones of the board when necessary). It is however advisable to clear (delete) models and data sets that no longer are of interest. Do that by dragging the object to the **Trash Can**. (Double-clicking on the trash can will open it up, and its contents can be retrieved.) Empty the can if you run into memory problems.

Window Culture: Dialog and plot windows are best managed by the GUI's `close` function (submenu item under **File** menu, or select **Close**, or check/uncheck the corresponding View box). It is generally not suitable to iconify the windows – the GUI's handling and window management system is usually a better alternative.

Workspace Variables

The models and data sets created within the GUI are normally not available in the MATLAB workspace. Indeed, the workspace is not at all littered with variables during the sessions with the GUI. The variables can however be exported at any time to the workspace, by dragging and dropping the object icon on the **To Workspace** box. They will then carry the name in the workspace that marked the object icon at the time of export. You can work with the variables in the workspace, using any MATLAB commands, and then perhaps import modified versions back into the GUI. Note that models and data are exported as the toolbox's objects `idmodel`, `idfrd`, and `iddata`. For how to extract information and work with these objects, see Chapter 3, "Tutorial" and "Model Conversions" on page 4-6 of the "Command Reference "chapter.

The GUI's names of data sets and models are suggested by default procedures. Normally, you can enter any other name of your choice at the time of creation of the variable. Names can be changed (after double-clicking on the icon) at any time. Unlike the workspace situation, two GUI objects can carry the same name (i.e., the same string in their icons).

Help Texts

The GUI contains some 100 help texts that are accessible in a nested fashion, when required. The main **ident** window contains general help topics under the **Help** menu. This is also the case for the various plot windows. In addition, every dialog box has a **Help** push button for current help and advice.

Handling Data

Data Representation

In the System Identification Toolbox, signals and observed data are represented as column vectors, e.g.,

$$u = \begin{bmatrix} u(1) \\ u(2) \\ \dots \\ \dots \\ u(N) \end{bmatrix}$$

The entry in row number k, i.e., $u(k)$, will then be the signal's value at sampling instant number k. It is generally assumed in the toolbox that data are sampled at equidistant sampling times, and the sampling interval T is supplied as a specific argument.

We generally denote the input to a system by the letter u and the output by y. If the system has several input channels, the input data is represented by a matrix, where the columns are the input signals in the different channels:

$$u = \begin{bmatrix} u1 & u2 & \dots & um \end{bmatrix}$$

The same holds for systems with several output channels.

The observed input-output data record is represented in the System Identification Toolbox by the `iddata` object, that is created from the input and output signals by

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

where `Ts` is the sampling time

The `iddata` object can also be created from the input and output signals when the data are inserted into the GUI.

Getting Input-Output Data into the GUI

The information about a data set that should be supplied to the GUI is as follows:

1. The input and output signals
2. The name you give to the data set
3. The sampling interval

In addition to this mandatory information, you may add further properties that will help in the bookkeeping:

4. The starting time for the sampling
5. Input and output channel names
6. Input and output channel units
7. Periodicity and intersample behavior of the input
8. Data notes: These are notes for your own information and bookkeeping that will follow the data and all models created from them.

As you select the pop-up menu **Data** and choose the item **Import**, a dialog box will open, where you can enter the information items 1 - 8, just listed. This box has five fields for you to fill in.

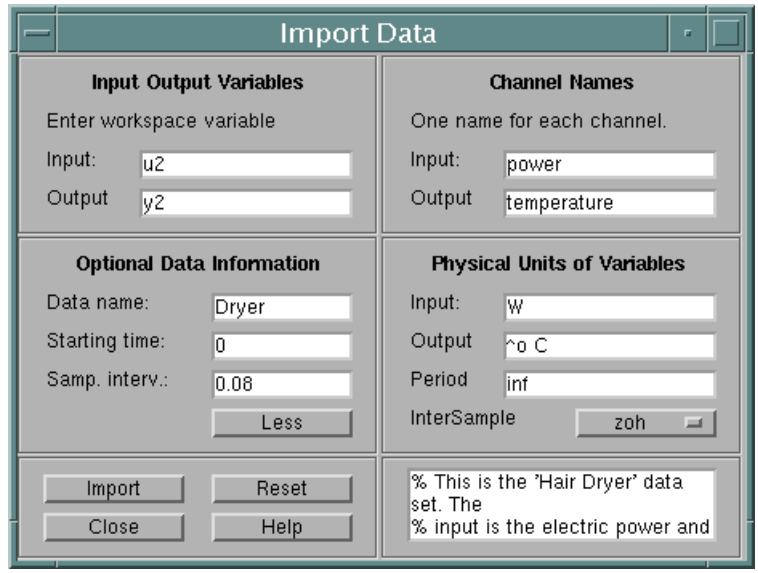

Figure 2-2: The Dialog for Importing Data into the GUI

By pressing **More,** six more fields will become visible.

Input and **Output**: Enter the variable names of the input and output respectively. These should be variables in your MATLAB workspace, so you may have to load some disk files first.

Actually, you can enter any MATLAB expressions in these fields, and they will be evaluated to compute the input and the output before inserting the data into the GUI.

Data name: Enter the name of the data set to be used by the GUI. This name can be changed later on.

Starting time and Sampling interval: Fill these out for correct time and frequency scales in the plots.

On the extra page you optionally can fill out

Channel names: Enter strings for the different input and output channels names. Separate the strings by comma. The number of names must be equal to the number of channels. If these entries are not filled out, default names, y1,y2, ..., u1, u2 ..., will be used.

Channel units: Enter, in analogous format, the units in which the measurements are made. These will follow to all models built from data, but are used only for plot information.

Period: If the input is periodic, enter here the period length. 'Inf' means a non-periodic input, which is default.

Intersample: Choose the intersample behavior of the input as one of ZOH (zero-order hold, i.e., the input signal piecewise constant between the samples) or FOH (first-order hold, i.e., the input signal is piecewise linear between the samples) or BL (Band-limited, i.e., the continuous time input signal has no power above the Nyquist frequency). ZOH is default.

The box at the bottom is for **Notes,** where you can enter any text you want to accompany the data for bookkeeping purposes.

Finally, select **Import** to insert the data into the GUI. When no more data sets are to be inserted, select **Close** to close the dialog box. **Reset** will empty all the fields of the box.

The procedure just described will create an `iddata` object, with all its properties. If you already have an `iddata` object available in the workspace, you can import that directly by selecting the data format **Iddata Object** in the pop-up menu at the top of the **Import Data** dialog.

Taking a Look at the Data

The first thing to do after having inserted the data set into the Data Board is to examine it. By checking the **Data View** item **Time plot**, a plot of the input and output signals will be shown for the data sets that are selected. You select/deselect the data sets by clicking on them. For multivariable data, the different combinations of input and output signals are chosen under menu item **Channel** in the plot window. Using the `zoom` function (drawing rectangles with the left mouse button down) different portions of the data can be examined in more detail.

To examine the frequency contents of the data, check the **Data View** item **Data spectra**. The function is analogous to **Time plot**, but the signals' spectra are shown instead. By default the periodograms of the data are shown, i.e., the absolute square of the Fourier transforms of the data. The plot can be changed to any chosen frequency range and a number of different ways of estimating spectra, by the **Options** menu item in the spectra window.

The purpose of examining the data in these ways is to find out if there are portions of the data that are not suitable for identification, if the information contents of the data is suitable in the interesting frequency regions, and if the data have to be preprocessed in some way, before using them for estimation.

Preprocessing Data

Detrending

Detrending the data involves removing the mean values or linear trends from the signals (the means and the linear trends are then computed and removed from each signal individually). This function is accessed under the pop-up menu **Preprocess**, by selecting item **Remove Means** or **Remove Trends**. More advanced detrending, such as removing piecewise linear trends or seasonal variations cannot be accessed within the GUI. It is generally recommended that you remove at least the mean values of the data before the estimation phase. There are however situations when it is not advisable to remove the sample means. It could for example be that the physical levels are built into the underlying model, or that integrations in the system must be handled with the right level of the input being integrated.

Selecting Data Ranges

It is often the case that the whole data record is not suitable for identification, due to various undesired features (missing or "bad" data, outbursts of disturbances, level changes etc.), so that only portions of the data can be used. In any case, it is advisable to select one portion of the measured data for estimation purposes and another portion for validation purposes. The pop-up menu item **Preprocess > Select Range...** opens a dialog box, which facilitates the selection of different data portions, by typing in the ranges, or marking them by drawing rectangles with the mouse button down.

For multivariable data it is often advantageous to start by working with just some of the input and output signals. The menu item **Preprocess > Select**

Channels... allows you to select subsets of the inputs and outputs. This is done in such a way that the input/output numbering and names remains consistent when you evaluate data and model properties, for models covering different subsets of the data.

Prefiltering

By filtering the input and output signals through a linear filter (the same filter for all signals) you can, e.g., remove drift and high frequency disturbances in the data, that should not affect the model estimation. This is done by selecting the pop-up menu item **Preprocess > Filter...** in the main window. The dialog is quite analogous to that of selecting data ranges in the time domain. You mark with a rectangle in the spectral plots the intended passband or stop band of the filter, you select a button to check if the filtering has the desired effect, and then you insert the filtered data into the GUI's Data Board.

Prefiltering is a good way of removing high frequency noise in the data, and also a good alternative to detrending (by cutting out low frequencies from the pass band). Depending on the intended model use, you can also make sure that the model concentrates on the important frequency ranges. For a model that will be used for control design, for example, the frequency band around the intended closed-loop bandwidth is of special importance.

If you intend to use the data to build models both of the system dynamics and the disturbance properties, it is recommended to do the filtering at the estimation phase. That is achieved by selection the pop-up menu item **Estimate > Parametric Models,** and then select the estimation **Focus** to be **Filter**. This opens the same filter dialog as above. The prefiltering will however apply only for estimating the dynamics from input to output. The disturbance model is determined from the original data.

Resampling

If the data turn out to be sampled too fast, they can be decimated, i.e., every k-th value is picked, after proper prefiltering (antialias filtering). This is obtained from menu item **Preprocess > Resample**.

You can also resample at a faster sampling rate by interpolation, using the same command, and giving a resampling factor less than one.

Quickstart

The pop-up menu item **Preprocess > Quickstart** performs the following sequence of actions: It opens the **Time plot Data view**, removes the means from the signals, and it splits these detrended data into two halves. The first one is made Working Data and the second one becomes Validation Data. All the three created data sets are inserted into the Data Board.

Multi-Experiment Data

The toolbox allows the handling of data sets that contain several different experiments. Both estimation and validation can be applied to such data sets. This is quite useful to deal with experiments that have been conducted at different occasions but describe the same system. It is also useful to be able to keep together pieces of data that have been obtained by cutting out "informative pieces" from a long data set. Multi-experiment data can be imported and used in the GUI as any `iddata` object. Selecting specific part of a multi-experiment data set is done from the pop-up menu item Preprocess > Select Experiment. To merge several data sets in the Data board (obtained, e.g., by cutting out nice portions from other data sets) use the pop-up menu item Preprocess > Merge Experiment.

Checklist for Data Handling

- Insert data into the GUI's Data Board.
- Plot the data and examine it carefully.
- Typically detrend the data by removing mean values.
- Select portions of the data for Estimation and for Validation. Drag and drop these data sets to the corresponding boxes in the GUI.

Simulating Data

The GUI is intended primarily for working with real data sets, and does not itself provide functions for simulating synthetic data. That has to be done in command mode, and you can use your favorite procedure in Simulink, the Signal Processing Toolbox, or any other toolbox for simulation and then insert the simulated data into the GUI as described above.

The System Identification Toolbox also has several commands for simulation. For example, should check `idinput` and `sim` in the "Command Reference" chapter for details. The following example shows how the ARMAX model

$$y(t) - 1.5y(t-1) + 0.7y(t-2) =$$
$$u(t-1) + 0.5u(t-2) + e(t) - e(t-1) + 0.2e(t-1)$$

is simulated with a random binary input u.

```
% Create an ARMAX model
model1 = idpoly([1 -1.5 0.7],[0 1 0.5],[1 -1 0.2]);
u = idinput(400,'rbs',[0 0.3]);
e = randn(400,1);
y = sim(model1,[u e]);
```

The input, u, and the output, y, can now be imported into the GUI as data, and the various estimation routines can be applied to them. By also importing the simulation model, `model1`, into the GUI, its properties can be compared to those of the different estimated models.

To simulate a continuous-time state-space model

$$\dot{x} = Ax + Bu + Ke$$
$$y = Cx + e$$

with the same input, and a sampling interval of 0.1 seconds, do the following in the System Identification Toolbox.

```
A = [-1 1;-0.5 0]; B = [1; 0.5]; C = [1 0]; D = 0; K = [0.5;0.5];
Model2 = idss(A,B,C,D,K,'Ts', 0) % Ts = 0 means continuous time
Data = iddata([],[u e]);
Data.Ts = 0.1
y=sim(Model2,Data);
```

Estimating Models

The Basics

Estimating models from data is the central activity in the System Identification Toolbox. It is also the one that offers the most variety of possibilities and thus is the most demanding one for the user.

All estimation routines are accessed from the pop-up menu **Estimate** in the **ident** window. The models are always estimated using the data set that is currently in the **Working Data** box.

One can distinguish between two different types of estimation methods:

- Direct estimation of the Impulse or the Frequency Response of the system. These methods are often also called nonparametric estimation methods, and do not impose any structure assumptions about the system, other than that it is linear.
- Parametric methods. A specific model structure is assumed, and the parameters in this structure are estimated using data. This opens up a large variety of possibilities, corresponding to different ways of describing the system. Dominating ways are state-space and several variants of difference equation descriptions.

Direct Estimation of the Impulse Response

A linear system can be described by the impulse response g_k, with the property that

$$y(t) = \sum_{k=1}^{\infty} g_k u(t-k)$$

The name derives from the fact that if the input *u(t)* is an impulse, i.e., *u(t)*=1 when *t*=0 and 0 when *t*>0, then the output *y(t)* will be $y(t) = g_t$. For a multivariable system, the impulse response g_k will be a *ny*-by-*nu* matrix, where *ny* is the number of outputs and *nu* is the number of inputs. Its *i-j* element thus describes the behavior of the i-th output after an impulse in the *j*-th input.

By choosing menu item **Estimate > Correlation Model** impulse response coefficients are estimated directly from the input/output data using so called

correlation analysis. The actual method is described under the command `impulse` in the "Command Reference" chapter. For a quick action, you can also just type the letter `c` in the **ident** window. This is the *hotkey* for correlation analysis.

The resulting impulse response estimate is placed in the Model Board, under the default name `imp`. (The name can be changed by double-clicking on the model icon and then typing in the desired name in the dialog box that opens.)

The best way to examine the result is to select the **Model View Transient Response**. This gives a graph of the estimated response. This view offers a choice between displaying the Impulse or the Step response. For a multivariable system, the different channels, i.e., the responses from a certain input to a certain output, are selected under menu item **Channel**.

The number of lags for which the impulse response is estimated, i.e., the length of the estimated response, is determined as one of the options in the Transient Response view.

Direct Estimation of the Frequency Response

The frequency response of a linear system is the Fourier transform of its impulse response. This description of the system gives considerable engineering insight into its properties. The relation between input and output is often written

$$y(t) = G(z)u(t) + v(t)$$

where G is the transfer function and v is the additive disturbance. The function

$$G(e^{i\omega T})$$

as a function of (angular) frequency ω is then the frequency response or frequency function. T is the sampling interval. If you need more details on the different interpretations of the frequency response, See "The System Identification Problem" on page 3-9 in the Tutorial chapter or any textbook on linear systems.

The system's frequency response is directly estimated using *spectral analysis* by the menu item **Estimate > Spectral Model**, and then selecting the **Estimate** button in the dialog box that opens. The result is placed on the Model Board under the default name `spad`. The best way to examine it is to plot it using the **Model View Frequency Response**. This view offers a number of

different options on how to graph the curves. The frequencies for which to estimate the response can also be selected as an option under the **Options** menu in this **View** window.

The Spectral Analysis command also estimates the spectrum of the additive disturbance *v(t)* in the system description. This estimated disturbance spectrum is examined under the **Model View** item **Noise Spectrum**.

The Spectral Analysis estimate is stored as an `idfrd` object. If you need to further work with the estimates, you can export the model to the MATLAB workspace and retrieve the responses directly from this object or by Nyquist and Bode. See `idfrd`, `bode`, and `nyquist` in the "Command Reference" chapter for more information. (A model is exported by dragging and dropping it over the **To Workspace** icon.)

Two options that affect the spectral analysis estimate can be set in the dialog box. The most important choice is a number, M, (the size of the lag window) that affects the frequency resolution of the estimates. Essentially, the frequency resolution is about $2\pi/M$ radians/(sampling interval). The choice of M is a trade-off between frequency resolution and variance (fluctuations). A large value of M gives good resolution but fluctuating and less reliable estimates. The default choice of M is good for systems that do not have very sharp resonances and may have to be adjusted for more resonant systems.

The options also offer a choice between the Blackman-Tukey windowing method `spa` (which is default) and a method based on smoothing direct Fourier transforms, `etfe`. `etfe` has an advantage for highly resonant systems, in that it is more efficient for large values of M. It however has the drawbacks that it requires linearly spaced frequency values, does not estimate the disturbance spectrum, and does not provide confidence intervals. The actual methods are described in more detail in the "Command Reference" chapter under `spa` and `etfe`. To obtain the spectral analysis model for the current settings of the options, you can just type the hotkey `s` in the **ident** window.

Estimation of Parametric Models

The System Identification Toolbox supports a wide range of model structures for linear systems. They are all accessed by the menu item **Estimate > Parametric Models...** in the **ident** window. This opens up a dialog box

Parametric Models, which contains the basic dialog for all parametric estimation as shown below.

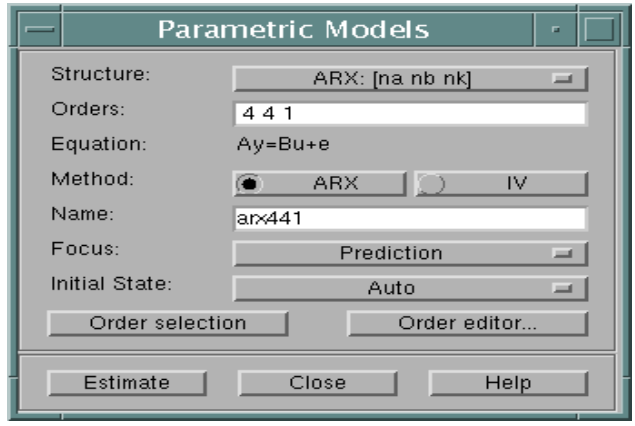

Figure 2-3: The Dialog Box for Estimating Parametric Models

The basic function of this box is as follows.

As you select **Estimate**, a model is estimated from the Working Data. The structure of this model is defined by the pop-up menu **Structure** together with the edit box **Orders**. It is given a name, which is written in the edit box **Name**.

The GUI will always suggest a default model name in the **Name** box, but you can change it to any string before selecting the **Estimate** button. (If you intend to export the model later, avoid spaces in the name.)

The interpretation of the model structure information (typically integers) in the **Order** box, depends on the selected **Structure** in the pop-up menu. This covers, typically, six choices:

- ARX models
- ARMAX model
- Output-Error (OE) models
- Box-Jenkins (BJ) models
- State-space models
- Model structure defined by Initial Model (User defined structures)

These are dealt with one by one shortly.

You can fill out the **Order** box yourself at any time, but for assistance you can select **Order Editor**. This will open up another dialog box, depending on the chosen **Structure**, in which the desired model order and structure information can be entered in a simpler fashion.

You can also enter a name of a MATLAB workspace variable in the order edit box. This variable should then have a value that is consistent with the necessary orders for the chosen structure.

Note For the state-space structure and the ARX structure, several orders and combination of orders can be entered. Then all corresponding models will be compared and displayed in a special dialog window for you to select suitable ones. This could be a useful tool to select good model orders. This option is described in more detail later in this section. When it is available, a button **Order selection** is visible.

Estimation Method

A common and general method of estimating the parameters is the *prediction error approach*, where simply the parameters of the model are chosen so that the difference between the model's (predicted) output and the measured output is minimized. This method is available for all model structures. Except for the ARX case, the estimation involves an iterative, numerical search for the best fit.

To obtain information from and interact with this search, select **Iteration control...** This is a button which is visible when an iterative estimation process has been selected. This also gives access to a number of options that govern the search process. (See "Algorithm Properties" on page 4-10 in the "Command Reference" chapter.)

For some model structures (the ARX model, and black-box state-space models) methods based on correlation are also available: Instrumental Variable (IV) and Sub-space (N4SID) methods. The choice between methods is made in the **Parametric Models** dialog box.

The dialog box also has three pop-up menus that offer further options**: Focus** allows you to choose between a frequency weighting that concentrates on the model's prediction or simulation performance. Another alternative is prefiltering, which was described on page 2-12. Moreover, the pop-up menu

InitialState gives options to estimate the initial state or to fix it to zero. The value **Auto** makes an automatic choice among these options. Finally, the menu **Covariance** allows the choice between **Estimate** and **None.** The normal situation is that the covariance of the model is estimated, so that various uncertainty measures can be displayed in the plots. However, for high order state-space models estimated by N4SID, or large multivariable ARX models, the computation of the covariance matrix may take quite a long time. Choosing **Covariance: None** will then greatly reduce the computation time.

Resulting Models

The estimated model is inserted into the GUI's Model Board. You can then examine its various properties and compare them with other models' properties using the **Model View** plots. More about that in "Examining Models" on page 2-28.

To take a look at the model itself, double-click on the model's icon (or right-click or **Ctrl**-click). The **Data/Model Info** window that then opens gives you information about how the model was estimated. You can then also select the **Present** button, which will list the model, and its parameters with estimated standard deviations in the MATLAB command window.

If you need to work further with the model, you can export it by dragging and dropping it over the **To Workspace** icon, and then apply any MATLAB and toolbox commands to it. (See, in particular, the commands `ssdata`, `tfdata`, `d2c`, and `get` in the "Command Reference" chapter.)

How to Know Which Structure and Method to Use

There is no simple way to find out "the best model structure"; in fact, for real data, there is no such thing as a *best* structure. Some routes to find good and acceptable model are described in "A Startup Identification Procedure" on page 1-14. It is best to be generous at this point. It often takes just a few seconds to estimate a model, and by the different validation tools described in the next section, you can quickly find out if the new model is any better than the ones you had before. There is often a significant amount of work behind the data collection, and spending a few extra minutes trying out several different structures is usually worthwhile.

ARX Models

The Structure

The most used model structure is the simple linear difference equation

$$y(t) + a_1 y(t-1) + \ldots + a_{na} y(t-na) =$$
$$b_1 u(t-nk) + \ldots + b_{nb} u(t-nk-nb+1)$$

which relates the current output *y(t)* to a finite number of past outputs *y(t-k)* and inputs *u(t-k)*.

The structure is thus entirely defined by the three integers na, nb, and nk. na is equal to the number of poles and nb–1 is the number of zeros, while nk is the pure time-delay (the dead-time) in the system. For a system under sampled-data control, typically nk is equal to 1 if there is no dead-time.

For multi-input systems nb and nk are row vectors, where the i-th element gives the order/delay associated with the i-th input.

Entering the Order Parameters

The orders na, nb, and nk can either be directly entered into the edit box **Orders** in the **Parametric Models** window, or selected using the pop-up menus in the **Order Editor**.

Estimating Many Models Simultaneously

By entering any or all of the structure parameters as vectors, using MATLAB's colon notation, like na=1:10, etc., you define many different structures that correspond to all combinations of orders. When selecting **Estimate**, models corresponding to all of these structures are computed. A special plot window will then open that shows the fit of these models to Validation Data. By clicking in this plot, you can then enter any models of your choice into the Model Board.

Multi-input models: For multi-input models you can of course enter each of the input orders and delays as a vector. The number of models resulting from all combinations of orders and delays can however be very large. As an alternative, you may enter one vector (like nb=1:10) for all inputs and one vector for all delays. Then only such models are computed that have the same orders and delays from all inputs.

Estimation Methods

There are two methods to estimate the coefficients a and b in the ARX model structure:

Least Squares: Minimizes the sum of squares of the right-hand side minus the left-hand side of the expression above, with respect to *a* and *b*. This is obtained by selecting ARX as the **Method**.

Instrumental Variables: Determines *a* and *b* so that the error between the right- and left- hand sides becomes uncorrelated with certain linear combinations of the inputs. This is obtained by selecting IV in the **Method** box.

The methods are described in more detail in the "Command Reference" chapter under arx and iv4.

Multi-Output Models

For a multi-output ARX structure with ny outputs and nu inputs, the difference equation above is still valid. The only change is that the coefficients *a* are ny-by-ny matrices and the coefficients *b* are ny-by-nu matrices.

The orders [NA NB NK] define the model structure as follows:

NA: an *ny*-by-*ny* matrix whose *i*-*j* entry is the order of the polynomial (in the delay operator) that relates the *j*-th output to the *i*-th output

NB: an *ny*-by-*nu* matrix whose *i*-*j* entry is the order of the polynomial that relates the *j*-th input to the *i*-th output

NK: an *ny*-by-*nu* matrix whose *i*-*j* entry is the delay from the *j*-th input to the *i*-th output

The **Order Editor** dialog box allows the choices

```
NA = na*ones(ny,ny)
NB = nb*ones(ny,nu)
NK = nk*ones(ny,nu)
```

where na, nb, and nk are chosen by the pop-up menus.

For tailor-made order choices, construct a matrix [NA NB NK] in the MATLAB command window and enter the name of this matrix in the **Order** edit box in the **Parametric Models** window.

Note that the possibility to estimate many models simultaneously is not available for multi-output ARX models.

See “Defining Model Structures” on page 3-35 for more information on multi-output ARX models.

ARMAX, Output-Error and Box-Jenkins Models

There are several elaborations of the basic ARX model, where different disturbance models are introduced. These include well known model types, such as ARMAX, Output-Error, and Box-Jenkins.

The General Structure

A general input-output linear model for a single-output system with input u and output y can be written

$$A(q)y(t) = \sum_{i=1}^{nu} [B_i(q)/F_i(q)]u_i(t - nk_i) + [C(q)/D(q)]e(t)$$

Here u_i denotes input #i, and A, B_i, C, D, and F_i, are polynomials in the shift operator (z or q). (Don’t get intimidated by this: It is just a compact way of writing difference equations; see below.)

The general structure is defined by giving the time-delays nk and the orders of these polynomials *(i.e.*, the number of poles and zeros of the dynamic model from u to y, as well as of the disturbance model from e to y).

The Special Cases

Most often the choices are confined to one of the following special cases.

ARX: $A(q)y(t) = B(q)u(t - nk) + e(t)$

ARMAX: $A(q)y(t) = B(q)u(t - nk) + C(q)e(t)$

OE: $y(t) = [B(q)/F(q)]u(t - nk) + e(t)$ (Output-Error)

BJ: $y(t) = [B(q)/F(q)]u(t - nk) + [C(q)/D(q)]e(t)$ (Box-Jenkins)

The “shift operator polynomials” are just compact ways of writing difference equations. For example the ARMAX model in longhand would be

$$y(t) + a_1y(t-1) + \ldots + a_{na}y(t - na) = b_1u(t - nk) + \ldots +$$
$$b_{nb}u(t - nk - nb + 1) + e(t) + c_1e(t-1) + \ldots + c_{nc}e(t - nc)$$

Note that *A(q)* corresponds to poles that are common between the dynamic model and the disturbance model (useful if disturbances enter the system "close to" the input). Likewise $F_i(q)$ determines the poles that are unique for the dynamics from input # *i*, and *D(q)* the poles that are unique for the disturbances.

The reason for introducing all these model variants is to provide for flexibility in the disturbance description and to allow for common or different poles (dynamics) for the different inputs.

Entering the Model Structure

Use the **Structure** pop-up menu in the **Parametric Models** dialog to choose between the ARX, ARMAX, Output-Error, and Box-Jenkins structures. Note that if the Working Data set has several outputs, only the first choice is available. For time series (data with no input signal) only AR and ARMA are available among these choices. These are the time series counterparts of ARX and ARMAX.

The orders of the polynomials are selected by the pop-up menus in the **Order Editor** dialog window, or by directly entering them in the edit box **Orders** in the **Parametric Models** window. When the order editor is open, the default orders, entered as you change the model structure, are based on previously used orders.

Estimation Method

The coefficients of the polynomials are estimated using a prediction error/ Maximum Likelihood method, by minimizing the size of the error term "e" in the expression above. Several options govern the minimization procedure. These are accessed by activating **Iteration Control** in the **Parametric Models** window, and selecting **Options**.

The algorithms are further described in Chapter 4, "Command Reference" under `armax`, `Algorithm Properties`, `bj`, `oe`, and `pem`. See also "Parametric Model Estimation" on page 3-25 and "Defining Model Structures" on page 3-35.

Note These model structures are available only for the scalar output case. For multi-output models, the state-space structures offer the same flexibility. Also note that it is not possible to estimate many different structures simultaneously for the input-output models.

State-Space Models

The Model Structure

The basic state-space model in innovations form can be written

$$x(t+1) = A\,x(t) + B\,u(t) + K\,e(t)$$
$$y(t) = C\,x(t) + D\,u(t) + e(t)$$

The System Identification Toolbox supports two kinds of parametrizations of state-space models: black-box, free parametrizations, and parametrizations tailor-made to the application. The latter is discussed below under the heading "User Defined Model Structures" on page 2-26. First we will discuss the black-box case.

Entering Black-Box State-Space Model Structures

The most important structure index is the model order; i.e., the dimension of the state vector *x*.

Use the pop-up menu in the **Order Editor** to choose the model order, or enter it directly into the **Orders** edit box in the **Parametric Models** window. Using the other pop-up menus in the **Order Editor**, you can further affect the chosen model structure:

- Fixing *K* to zero gives an Output-Error method; i.e., the difference between the model's simulated output and the measured one is minimized. Formally, this corresponds to an assumption that the output disturbance is white noise.

The delays from the input can be chosen independently for each input. It will be a row vector nk, with *nu* entries. When the delay is larger than or equal to one, the *D*-matrix in the discrete time model is fixed to zero. For physical systems, without a pure time delay, that are driven by piece-wise constant

inputs, nk = 1 is a natural assumption This is also the default. Note also that the delays can be directly entered into the **Orders** edit box.

Estimating Many Models Simultaneously

By entering a vector for the model order, using MATLAB's colon notation, (such as "1:10") all indicated orders will be computed using a preliminary method. You can then enter models of different orders into the Model Board by clicking in a special graph that contains information about the models.

Estimation Methods

There are two basic methods for the estimation:

PEM: Is a standard prediction error/maximum likelihood method, based on iterative minimization of a criterion. The iterations are started up at parameter values that are computed from n4sid. The parametrization of the matrices *A, B, C, D,* and *K* is free. The search for minimum is controlled by a number of options. These are accessed from the **Option** button in the **Iteration Control** window.

N4SID: Is a subspace-based method that does not use iterative search. The quality of the resulting estimates may significantly depend some options called N4Weight and N4Horizon. These options can be chosen in the **Order Editor** window. If N4Horizon is entered with several rows, the models corresponding to the horizons in each row are examined separately using the Working data. The best model in terms of prediction (or simulation, if K = 0) performance is selected. A figure is shown that illustrates the fit as a function of the horizon. If the N4Horizon box is left empty, a default choice is made.

See n4sid and pem in the "Command Reference" chapter for more information.

User Defined Model Structures

State-Space Structures

The System Identification Toolbox supports user-defined linear state-space models of arbitrary structure. Using the idmodel idss, known and unknown parameters in the *A, B, C, D, K,* and *X0* matrices can be easily defined both for discrete- and continuous-time models. The idgrey object allows you to use a completely arbitrary greybox structure, defined by an M-file. The model object properties can be easily manipulated. See idss and idgrey in Chapter 4,

"Command Reference" and "Structured State-Space Models with Free Parameters: the idss Model" on page 3-42

To use these structures in conjunction with the GUI, just define the appropriate structure in the MATLAB command window. Then use the **Structure** pop-up menu to select **By Initial Model** and enter the variable name of the structure in the edit box **Initial Model** in the **Parametric Models** window and select **Estimate**.

Any Model Structure

Arbitrary model structures can be defined using the System Identification Toolbox model objects:

- `idpoly`: Creates Input-output structures for single-output models
- `idss`: Creates Linear State-space models with arbitrary, free parameters
- `idgrey`: Creates completely arbitrary parametrizations of linear systems
- `idarx`: Creates multivariable ARX structures

In addition, all estimation commands create model structures in terms of the resulting models.

Enter the name of any model structure in the box **Orders (or Initial model)** in the window **Parametric Models** and then select **Estimate**. Then the parameters of the model structure are adjusted to the chosen Working Data set. The method is a standard prediction error/maximum likelihood approach that iteratively searches for the minimum of a criterion. Options that govern this search are accessed by the **Option** button in the **Iteration Control** window.

The name of the initial model must be a variable either in the workspace or in the Model Board. In the latter case you can just drag and drop it over the **Orders/Initial model** edit box.

Examining Models

Having estimated a model is just a first step. It must now be examined, compared with other models, and tested with new data sets. This is primarily done using the six **Model View** functions, at the bottom of the main **ident** window:

- Frequency response
- Transient response
- Zeros and poles
- Noise spectrum
- Model output
- Model residuals

In addition, you can double-click on the model's icon to get **Text Information** about the model. Finally, you can export the model to the MATLAB workspace and use any commands for further analysis and model use.

Views and Models

The basic idea is that if a certain **View** window is open (checked), then *all models* in the Model Summary Board that are selected will be represented in the window. The curves in the **View** window can be clicked in and out by selecting and deselecting the models in an online fashion. You select and deselect a model by clicking on its icon. An selected model is marked with a thicker line in its icon.

On color screens, the curves are color coded along with the model icons in the Model Board. Before printing a plot it might be a good idea to separate the line styles (menu item under **Style**). This could also be helpful on black and white screens.

Note that models that are obtained by spectral analysis only can be represented as frequency response and noise spectra, and that models estimated by correlation analysis only can be represented as transient response.

The Plot Windows

The six views all give similar plot windows, with several common features. They have a common menu bar, which covers some basic functions.

First of all, note that there is a zoom function in the plot window. By dragging with the left mouse button down, you can draw rectangles, which will be enlarged when the mouse button is released. By double-clicking, the original axis scales are restored. For plots with two axes, the x-axes scales are locked to each other. A single click on the left mouse button zooms in by a factor of two, while the middle button zooms out. The zoom function can be deactivated if desired. Just select the menu item **Zoom** under **Style**.

Second, by pointing to any curve in the plot, and pressing the right mouse button, the curve will be identified with model name and present coordinates.

The common menu bar covers the following functions.

File

File allows you to copy the current figure to another, standard MATLAB figure window. This might be useful, e.g., when you intend to print a customized plot. Other **File** items cover printing the current plot and closing the plot window.

Options

Options first of all cover actions for setting the axes scaling. This menu item also gives a number of choices that are specific for the plot window in question, like a choice between step response or impulse response in the **Transient response** window.

A most important option is the possibility to show confidence intervals. Each estimated model property has some uncertainty. This uncertainty can be estimated from data. By checking **Show confidence intervals**, a confidence region around the nominal curve (model property) will be marked (by dash-dotted lines). The level of confidence can also be set under this menu item.

Note Confidence intervals are supported for most models and properties, except models estimated using `etfe`, and the k-step ahead prediction-property. For `n4sid`, the covariance properties are actually not fully known. The Cramer-Rao lower limit for the covariance matrix is then used instead.

Style

The style menu gives access to various ways of affecting the plot. You can add gridlines, turn the zoom on and off, and change the linestyles. The menu also covers a number of other options, like choice of units and scale for the axis.

Channel

For multivariate systems, you can choose which input-output channel to examine. The current choice is marked in the figure title.

Help

The **Help** menu has a number of items, which explain the plot and its options.

Frequency Response and Disturbance Spectra

All linear models that are estimated can be written in the form

$$y(t) = G(z)u(t) + v(t)$$

where *G(z)* is the (discrete-time) transfer function of the system and *v(t)* is an additive disturbance. The frequency response or frequency function of the system is the complex-valued function $G(e^{i\omega T})$ viewed as a function of angular frequency ω.

This function is often graphed as a Bode diagram, i.e., the logarithm of the amplitude (the absolute value) of $G(e^{i\omega T})$ as well as the phase (the argument) of $G(e^{i\omega T})$ are plotted against the logarithm of frequency ω in two separate plots. These plots are obtained by checking the **Model View Frequency Response** in the main **ident** window.

The estimated spectrum of the disturbance *v* is plotted as a power spectrum by choosing the **Model View Noise Spectrum**.

If the data is a time series *y* (with no input *u*), then the spectrum of *y* is plotted under **Noise Spectrum**, and no frequency functions are given.

Transient Response

Good and simple insight into a model's dynamic properties is obtained by looking at its step response or impulse response. This is the output of the model when the input is a step or an impulse. These responses are plotted when the **Model View Transient Response** is checked.

It is quite informative to compare the transient response of a parametric model, with the one that was estimated using correlation analysis. If there is good agreement between the two, you can be quite confident that some essentially correct features have been picked up. It is useful to check the confidence intervals around the responses to see what "good agreement" could mean quantitatively.

Many models provide a description of the additive disturbance *v(t)*.

$$v(t) = H(z)e(t)$$

Here *H(z)* is a transfer function that describes how the disturbance *v(t)* can be thought of as generated by sending white noise *e(t)* through it. To display the properties of *H*, you can choose channels (in the **Channel** menu) that have noise components as inputs. The names of these channels are like `e@ynam`, for the noise component that directly affects the output with name `ynam`.

Poles and Zeros

The poles of a system are the roots of the denominator of the transfer function *G(z)*, while the zeros are the roots of the numerator. In particular the poles have a direct influence on the dynamic properties of the system.

The poles and zeros of *G* (and *H*) are plotted by choosing the **Model View Poles and Zeros**.

It is useful to turn on the confidence intervals in this case. They will clearly reveal which poles and zeros could cancel (their confidence regions overlap). That is an indication that a lower order dynamic model could be used.

For multivariable systems it is the poles and zeros of the individual input/output channels that are displayed. To obtain the so called transmission zeros, you will have to export the model and then apply the command `tzero`, provided you have the Control Systems Toolbox.

Compare Measured and Model Output

A very good way of obtaining insight into the quality of a model is to simulate it with the input from a fresh data set, and compare the simulated output with the measured one. This gives a good feel for which properties of the system have been picked up by the model, and which haven't.

This test is obtained by checking the **Model View Model Output**. Then the data set currently in the **Validation Data** box will be used for the comparison.

The fit will also be displayed. This is computed as the percentage of the output variation that is reproduced by the model. So, a model that has a fit of 0% gives the same mean square error as just setting the model output to be the mean of the measured output.

If the model is unstable, or has integration or very slow time constants, the levels of the simulated and the measured output may drift apart, even for a model that is quite good (at least for control purposes). It is then a good idea to evaluate the model's predicted output rather than the simulated one. With a *prediction horizon* of *k*, the *k*-step ahead predicted output is then obtained as follows:

The predicted value *y(t)* is computed from all available inputs $u(s)(s \le t)$ (used according to the model) and all available outputs up to time *t-k*, $y(s)(s \le t-k)$. The simulation case, where no past outputs at all are used, thus formally corresponds to $k=\infty$. To check if the model has picked up interesting dynamic properties, it is wise to let the predicted time horizon (kT, T being the sampling interval) be larger than the important time constants.

Note here that different models use the information in past output data in their predictors in different ways. This depends on the disturbance model. For example, so called Output-Error models (obtained by fixing K to zero for state-space models and setting `na=nc=nd=0` for input output models, see the previous section) do not use past outputs at all. The simulated and the predicted outputs, for any value of k, thus coincide.

Residual Analysis

In a model

$$y(t) = G(z)u(t) + H(z)e(t)$$

the noise source *e(t)* represents that part of the output that the model could not reproduce. It gives the "left-overs" or, in Latin, the *residuals*. For a good model, the residuals should be independent of the input. Otherwise, there would be more in the output that originates from the input and that the model has not picked up.

To test this independence, the cross-correlation function between input and residuals is computed by checking the **Model View Model Residuals**. It is wise to also display the confidence region for this function. For an ideal model the correlation function should lie entirely between the confidence lines for positive

lags. If, for example, there is a peak outside the confidence region for lag *k*, this means that there is something in the output *y(t)* that originates from *u(t-k)* and that has not been properly described by the model. The test is carried out using the Validation Data. If these were not used to estimate the model, the test is quite tough. See also “Model Structure Selection and Validation” on page 3-63.

For a model also to give a correct description of the disturbance properties (i.e., the transfer function *H*), the residuals should be mutually independent. This test is also carried out by the view **Model Residuals**, by displaying the auto-correlation function of the residuals (excluding lag zero, for which this function by definition is 1). For an ideal model, the correlation function should be entirely inside the confidence region.

Text Information

By double-clicking (right mouse button or Ctrl-click) on the model icon, a **Data/model Info** dialog box opens, which contains some basic information about the model. It also gives a diary of how the model was created, along with the notes that originally were associated with the estimation data set. At this point you can do a number of things:

Present

Selecting the **Present** button displays details of the model in the MATLAB command window. The model's parameters along with estimated standard deviations are displayed, as well as some other notes.

Modify

You can simply type in any text you want anywhere in the **Diary and Notes** editable text field of the dialog box. You can also change the name of the model just by editing the text field with the model name. The color, which the model is associated with in all plots, can also be edited. Enter RGB-values or a color name (like `'y'`) in the corresponding box.

LTI Viewer

If you have the Control System Toolbox, you will see an icon **To LTI Viewer** in the main window. By dragging and dropping a model onto this icon you will open the LTI Viewer. This viewer handles an arbitrary amount of models, but it requires all of them to have the same number of inputs and outputs.

Further Analysis in the MATLAB Workspace

Any model and data object can be exported to the MATLAB workspace by dragging and dropping its icon over the **To Workspace** box in the **ident** window.

Once you have exported the model to the workspace, there are many commands by which you can further transform it, examine it, and convert it to other formats for use in other toolboxes. Some examples of such commands are

`d2c`	Transform to continuous time
`ss, idss, ssdata`	Convert to state-space representation
`tf, tfdata`	Convert to transfer function form
`zpk, zpkdata`	Convert to zeros and poles

Note that the commands `ss`, `tf`, and `zkp` transform the model to the Control System Toolbox's LTI models. Moreover, if you have that toolbox many of its LTI-commands can be applied directly to the model objects of the Identification Toolbox. See "Connections Between the Control System Toolbox and the System Identification Toolbox" on page 3-87

Also, if you need to prepare specialized plots that are not covered by the **Views**, all the System Identification Toolbox commands for computing and extracting simulations, frequency functions, zeros and poles, etc., are available. See the "Tutorial" and "Command Reference" chapters.

Some Further GUI Topics

This section discusses a number of different topics.

Mouse Buttons and Hotkeys

The GUI uses three mouse buttons. If you have fewer buttons on your mouse, the actions associated with the middle and right mouse buttons are obtained by shift-click, alt-click or control-click, depending on the computer.

The Main ident Window

In the main **ident** window the mouse buttons are used to drag and drop, to select/deselect models and data sets, and to double-click to get text information about the object. You can use the left mouse button for all of this. A certain speed-up is obtained if you use the left button for dragging and dropping, the middle one for selecting models and data sets, and the right one for double-clicking (actually for the right button, (Ctrl-click) a single click is sufficient). On a slow machine a double-click from the left button might not be recognized.

The **ident** window also has a number of hotkeys**.** By pressing a keyboard letter when it is the current window, some functions can be quickly activated. These are:

- s: Computes **Spectral Analysis Model** using the current options settings. (These can be changed in the dialog window that opens when you choose **Spectral Model** in the **Estimate** pop-up menu.)
- c: Computes **Correlation Analysis Model** using the current options settings.
- q: Computes the models associated with the **Quickstart**.
- d: Opens a dialog window for importing **Data**

Plot Windows

In the various plot windows the action of the mouse buttons depends on whether the zoom is activated or not:

Zoom Active: Then the left and middle mouse buttons are associated with the zoom functions as in the standard MATLAB zoom. Left button zooms in and the

middle one zooms out. In addition, you can draw rectangles with the left button, to define the area to be zoomed. Double-clicking restores the original plot. The right mouse button is associated with special GUI actions that depend on the window. In the **View** plots, the right mouse button is used to identify the curves. Point and click on a curve, and a box will display the name of the model/data set that the curve is associated with, and also the current coordinate values for the curve. In the **Model Selection** plots the right mouse button is used to inspect and select the various models. In the **Prefilter** and **Data Range** plots, rectangles are drawn with this mouse button down, to define the selected range.

Zoom not active: The special GUI functions just mentioned are obtained by any mouse button.

The zoom is activated and deactivated under the menu item **Style.** The default setting differs between the plots. Don't activate the zoom from the command line! That will destroy the special GUI functions. (If you happen to do so anyway, "quit" the window and open it again.)

Troubleshooting in Plots

The function **Auto-range**, which is found under the menu item **Options**, sets automatic scales to the plots. It is also a good function to invoke when you think that you have lost control over the curves in the plot. (This may happen, for example, if you have zoomed in a portion of a plot and then change the data of the plot).

If the view plots don't respond the way you expect them to, you can always "quit" the window and open it again. By quit here we mean using the underlying window system's own quitting mechanism, which is called different things in the different platforms. The normal way to close a window is to use the **Close** function under the menu item **File**, or to uncheck the corresponding check box.

Layout Questions and idprefs.mat

The GUI comes with a number of preset defaults. These include the window sizes and positions, the colors of the different models, and the default options in the different **View** windows.

The window sizes and positions, as well as the options in the plot windows, can of course be changed during the session in the standard way. If you want the

GUI to start with your current window layout and current plot options, select menu item

```
Options > Save preferences
```

in the main **ident** window. This saves the information in a file `idprefs.mat`. This file also stores information about the four most recent sessions with **ident**. This allows the session **File** menu to be correctly initialized. The session information is automatically stored upon exit. The layout and preference information is only saved when the indicated option is selected.

The file `idprefs.mat` is created the first time you close the GUI. It is by default stored in the same directory as your `startup.m` file. If this default does not work, you are prompted for a directory where to store the file. This can be ignored, but then session and preference information cannot be saved.

To change or select a directory for `idprefs.mat`, use the command `midprefs`. See the "Command Reference" chapter for details.

To change model colors and default options to your own customized choice, make a copy of the M-file `idlayout.m` to your own directory (which should be *before* the basic `ident` directory in the `MATLABPATH`), and edit it according to its instructions.

Customized Plots

If you need to prepare hardcopies of your plots with specialized texts, titles and so on, make a copy of the figure first, using **Copy Figure** under the **File** menu item. This produces a copy of the current figure in a standard MATLAB figure format.

For plots that are not covered by the **View** windows, (e.g., Nyquist plots), you have to export the model to the MATLAB workspace and construct the plots there.

What Cannot be Done Using the GUI

The GUI covers primarily everything you would like to do to examine data, estimate models and evaluate and compare models. It does not cover:

- Generation (simulation) of data sets
- Model creation (other than by estimation)

- Model manipulation and conversions
- Recursive (on-line) estimation algorithms

To see what M-files are available in the toolbox for these functions, check the "Tutorial" chapter, as well as the "Simulation and Prediction", "Model Structure Creation", "Manipulating Model Structures", "Model Conversions", and "Recursive Parameter Estimation" tables in the beginning of the "Command Reference" chapter.

Note that at any point you can export a data set or a model to the MATLAB workspace (by dragging and dropping its icon on the **To Workspace** icon). There you can modify and manipulate it any way you want and then import it back into **ident**. You can, for example, construct a continuous-time model from an estimated discrete-time one (using `d2c`), and then use the model views to compare the two.