System Identification Toolbox Release Notes

Note The System Identification Toolbox does not introduce any significant updates for Release 14. It does incorporate the updates in Version 6.0, which was released in Web download form after Release 13 with Service Pack 1 was released.

If you are upgrading from a version earlier than Version 6.0, see

- "System Identification Toolbox 6.0 Release Notes" on page 1-1
- "System Identification Toolbox 5.0 Release Notes" on page 2-1

Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.

Contents

1 System Identification Toolbox 6.0 Release Notes

2 System Identification Toolbox 5.0 Release Notes

1

System Identification Toolbox 6.0 Release Notes

New Features

This section introduces the new features and enhancements added in the System Identification Toolbox 6.0 since the System Identification Toolbox 5.0 (Release 12).

If you are upgrading from a version earlier than Version 5.0, see "New Features" on page 2-2 of the Version 5.0 release notes.

New Model Object

A new model object, `idproc`, has been introduced. It contains simple continuous time process models, characterized by static gain, possible dead time, and dominating time constant(s). See

```
help idproc
```

and

```
idprops idproc
```

for a complete description of the model, or run the demo `iddemopr`. A quick insight in what is does is obtained by

```
m = pem(data,'p1d')
```

There is also a new graphical user interface (GUI) that supports this object. It is opened from the main Ident GUI.

Estimation and Validation in the Frequency Domain

The toolbox now supports estimation and validation using frequency domain data. Inputs and outputs as frequency domain data in the `iddata` object are covered, as well as frequency response data, obtained, for example, from a frequency analyzer.

Frequency response data should be packaged as an `frd` or `idfrd` object. All estimation, simulation and validation routines accept frequency domain data and frequency response data as inputs quite analogously to time domain data. New routines `fft/ifft` transform between the time and frequency domains. A new routine `spafdr` allows estimation of frequency responses using frequency dependent resolution.

See

```
help iddata
```

and

```
idprops data
```

for complete descriptions and `iddemofr` for demonstrations. The main Ident GUI also supports frequency domain data.

Storing Continuous-Time Data as Frequency Domain Objects

You can now store continuous-time data as frequency domain data objects. This means that continuous-time Fourier transformed data are stored at a finite number of arbitrary frequencies. For such data, direct estimation of continuous-time models is possible. See, for example,

```
help oe
```

Simulink Support of iddata and idmodel Objects

There is direct Simulink support of `iddata` and `idmodel` objects. The command `slident` opens a Simulink block library that allows the simulation of any `idmodel` with or without noise. This library also contains data sources and sinks for `iddata` objects.

Advice Command

A new command `advice` can be applied to any `iddata` and any `idmodel` object. It gives the user advice about the quality, problems and options of the data set or the model. See

```
help iddata/advice
```

and

```
help idmodel/advice
```

for more information.

Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving from the System Identification Toolbox 4.0.5 to the System Identification Toolbox 6.0.

Theta Models No Longer Supported

Theta models (matrices) are no longer supported in the System Identification Toolbox 6.0. Existing code that uses functions such as `th2par` and `th2ss` to access the theta model data will continue to work in the System Identification Toolbox 6.0. However, if you have code that directly indexes into the theta matrix (e.g., `th(1,3)`), that code will no longer work.

2

System Identification Toolbox 5.0 Release Notes

New Features

This section introduces the new features and enhancements added in the System Identification Toolbox 5.0 since the System Identification Toolbox 4.0.5 (Release 11.0).

Object-Based Design

Based on MATLAB object technology, the System Identification Toolbox 5.0 provides functions for creating objects directly associated with your models and data. Some quick examples illustrating this feature are

```
z = iddata(y,u,Ts);
sys = pem(z);
```

The new object-based syntax makes it much easier to perform analysis activities beyond what the System Identification Toolbox visual interfaces support. The use of an object-based design by the System Identification Toolbox 5.0 makes it much easier to work with Control System Toolbox objects seamlessly, including converting back and forth between the two toolbox's objects and applying the relevant analysis tools to objects from both toolboxes.

For an overview of the features included in this new object-based approach, type

```
help idhelp
```

You do not need to rewrite any code you wrote using an earlier version of the System Identification Toolbox; the earlier command-line syntax is still supported in the System Identification Toolbox 5.0.

Advanced Feature Enhancements

The System Identification Toolbox 5.0 includes several enhancements to some of the toolbox's more advanced features:

- Free parameterization for state-space models is now supported. For example, you can simply use

  ```
  m = pem(z,4)
  ```

  to obtain a fourth order state-space model with a well-conditioned parameterization.

- You can now add initial filter conditions. This yields much better performance for slow dynamics. See the `'InitialState'` property of `idmodel` objects for further information.
- You can now use the `SearchDirection` and `Advanced` properties of `idmodel` objects to access several variants of iterative search algorithms. For more information, type

  ```
  idprops algorithm
  ```

- You can now focus the model approximation inherent in system identification to various frequency regions, by using the `Focus` property. The values for the `Focus` property include `'Prediction'`, `'Simulation'`, or any idmodel or LTI object that uses the frequency weighting of that system.