Model Output Plots

System Identification Toolbox lets you validate all linear parametric models and nonlinear models by checking how well the simulated or predicted output of the model matches the measured output.

> **Note** For nonparametric models, including impulse-response, step-response, and frequency-response models, model output plots are not available. For time-series models, you can only generate model-output plots for parametric models using time-domain time-series (no input) measured data.

If you plan to use the model for simulation applications, validate the model by comparing simulated output to the validation data. However, if you plan to use the model for prediction, compare the *k*-step-ahead predicted output to the validation data. For example, if you are modeling a plant for a control system, your model must perform for prediction over a horizon that corresponds to the time-constant of the system. For information about the difference between simulation and prediction, see the introductory chapter in the *Getting Started with System Identification Toolbox*.

This section discusses the following topics:

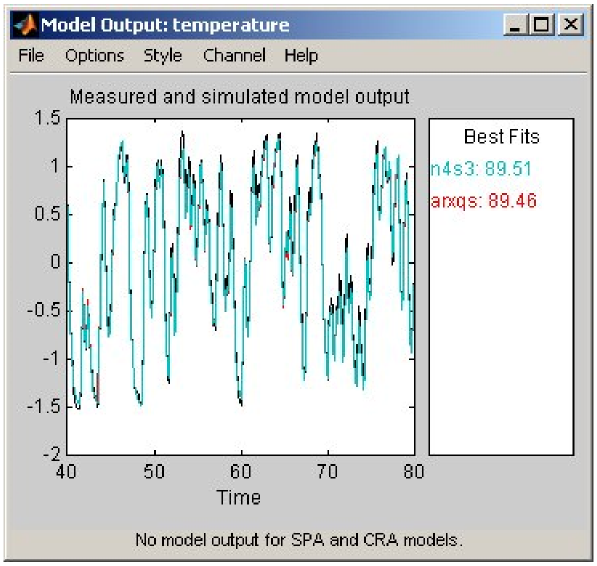
- What Does a Model Output Plot Show?
- Choosing Simulated or Predicted Output
- Displaying the Confidence Interval
- Plotting Model Output Using the GUI
- Using Functions to Plot Model Output

What Does a Model Output Plot Show?

The model output plot shows different information depending on the domain of the input-output validation data, as follows:

- For time-domain validation data, the plot shows simulated or predicted model output.
- For frequency-domain data, the plot shows the simulated complex-valued amplitude of the model output. The complex-valued amplitude is equal to the product of the Fourier transform of the input and the model frequency function.
- For frequency-response data, the plot shows the simulated amplitude of the model frequency response.

The following figure shows a sample Model Output plot, created in the System Identification Tool GUI.



No model output for SPA and CRA models.

For linear models, you can estimate a model using time-domain data, and then validate the model using frequency domain data. For nonlinear models, you can only use time-domain data for both estimation and validation.

Choosing Simulated or Predicted Output

How you validate the model output should match how you plan to use the model. If you plan to use the model in simulation applications, then compare the simulated model output to the validation data. However, if you plan to use the model for prediction, then compare the predicted output from the model to the measured output.

The main difference between simulation and prediction is whether System Identification Toolbox uses measured or computed previous outputs for calculating the next output.

Using a model for prediction is common in controls applications where you want to predict output a certain number of steps in advance. When you use System Identification Toolbox to *predict* model output, the algorithm uses both the measured and the calculated output data values in the difference equation for computing the next output.

The predicted value *y(t)* is computed from all available inputs *u(s)*, where $s \le (t-k)$, and all available outputs *y(s)*, where $s \le (t-k)$.

Simulating models uses the input-data values from a data set to compute the output values. When you *simulate* the model output, System Identification Toolbox computes the first output value using the initial conditions and the inputs. Then, System Identification Toolbox feeds this computed output into the differential (continuous-time) or difference (discrete-time) equation for calculating the next output value. In this way, the simulation progresses using previously calculated outputs in the difference equation to produce the next output; with an *infinite prediction horizon* (*k*=∞), the simulation has no limit on how far out in time it computes output values. Thus, no past outputs are used in the simulation.

To check whether the model has picked up interesting dynamic properties, let the predicted time horizon *kT* be larger than the important time constants, where *T* is the sampling interval.

> **Note** Output-error models, obtained by fixing *K* to zero for state-space models and setting `na=nc=nd=0` for polynomial models, do not use past outputs. Therefore, for these models, the simulated and the predicted outputs are the same for any value of *k*.

To learn how to display simulated or predicted output, see the description of the plot settings in Plotting Model Output Using the GUI.

Displaying the Confidence Interval

In the GUI, you can display a confidence interval on the plot to gain insight into the quality of a linear model. To learn how to show or hide confidence interval, see the description of the plot settings in Plotting Model Output Using the GUI.

The *confidence interval* corresponds to the range of output values with a specific probability of being the actual output of the system. System Identification Toolbox uses the estimated uncertainty in the model parameters to calculate confidence intervals and assumes the estimates have a Gaussian distribution.

For example, for a 95% confidence interval, the region around the nominal curve represents the range of values that have a 95% probability of being the true system response. You can specify the confidence interval as a probability (between 0 and 1) or as the number of standard deviations of a Gaussian distribution. For example, a probability of 0.99 (99%) corresponds to 2.58 standard deviations.

> **Note** The calculation of the confidence interval assumes that the model sufficiently describes the system dynamics and the model residuals pass independence tests.

Plotting Model Output Using the GUI

To create a model output plot for parametric linear and nonlinear models in the System Identification Tool window, select the **Model output** check box in the **Model Views** area. For general information about creating and working with plots, see Working with Plots.

To include or exclude a model on the plot, click the corresponding model icon in the System Identification Tool window. Active models display a thick line inside the Model Board icon.

The right side of the plot displays the percentage of the output that the model reproduces (**Best Fit**), computed using the following equation:

$$\text{Best Fit} = \left(1 - \frac{|y - \hat{y}|}{|y - \bar{y}|}\right) \times 100$$

In this equation, *y* is the measured output, $\hat{y}$ is the simulated or predicted model output, and $\bar{y}$ is the mean of *y*. 100% corresponds to a perfect fit, and 0% indicates that the fit is no better than guessing the output to be a constant ($\hat{y} = \bar{y}$).

Because of the definition of **Best Fit**, it is possible for this value to be negative. A negative best fit is worse than 0% and can occur for the following reasons:

- The estimation algorithm failed to converge.
- The model was not estimated by minimizing $|y - \hat{y}|$. **Best Fit** can be negative when you minimized 1-step-ahead prediction during the estimation, but validate using the simulated output $\hat{y}$.
- The validation data set was not preprocessed in the same way as the estimation data set.

The following table summarizes the Model Output plot settings.

Model Output Plot Settings

Action	**Command**
Display confidence intervals. **Note** Confidence intervals are only available for simulated model output of linear models. Confidence internal are not available for nonlinear ARX and Hammerstein-Wiener models.	• To display the dashed lines on either side of the nominal model curve, select **Options** > **Show confidence intervals**. Select this option again to hide the confidence intervals. • To change the confidence value, select **Options** > **Set % confidence level**, and choose a value from the list. • To enter your own confidence level, select **Options** > **Set confidence level** > **Other**. Enter the value as a probability (between 0 and 1) or as the number of standard deviations of a Gaussian distribution.
Change between simulated output or predicted output. **Note** Prediction is only available for time-domain validation data.	• Select **Options** > **Simulated output** or **Options** > **k step ahead predicted output**. • To change the prediction horizon, select **Options** > **Set prediction horizon**, and select the number of samples. • To enter your own prediction horizon, select **Options** > **Set prediction horizon** > **Other**. Enter the value in terms of the number of samples.
Display the actual output values (**Signal plot**), or the difference between model output and measured output (**Error plot**).	Select **Options** > **Signal plot** or **Options** > **Error plot**.
(Time-domain validation data only) Set the time range for model output and the time interval for which the **Best Fit** value is computed.	Select **Options** > **Customized time span for fit** and enter the minimum and maximum time values. For example: `[1 20]`

Action	Command
(Multiple-output system only) Select a different output.	Select the output by name in the **Channel** menu.

Using Functions to Plot Model Output

You can plot simulated and predicted model output using the `compare`, `sim`, and `predict` functions.

Simulation and prediction requires input data, a model, and the values of the initial states. If you estimated the model using one data set, but want to simulate the model using a different data set, the initial states of your simulation must be consistent with the latter data set.

By default, `sim` and `predict` use the initial states that were derived from the data you used to estimate the model. These initial states are not appropriate if you are simulating or predicting output using new data.

To use `sim` or `predict` with a data set that differs from the data you used to estimate the model, first estimate the new initial states `X0est` using `pe`:

```
[E,X0est]=pe(model,data)
```

Next, specify the estimated initial states `X0est` as an argument in `sim` or `predict`. For example:

```
y=sim(model,data,'InitialState',X0est)
```

> **Note** The `compare` function automatically estimates the initial states from the data and ensures consistency.

Function	Description	Example
`compare`	Plots simulated or predicted model output on top of the measured output. You should use an independent validation data set as input to the model.	To plot five-step-ahead predicted output of the model `mod` against the validation data `data`, use the following command: `compare(data,mod,5)` **Note** Omitting the third argument assumes an infinite horizon and results in simulation.
`sim`	Plots simulated model output only.	To simulate the response of the model `model` using input data `data`, use the following command: `sim(model,data)`
`predict`	Plots predicted model output only.	To perform one-step-ahead prediction of the response for the model `model` and input data `data`, use the following command: `predict(model,data,1)`

◀ Overview of Model Plots and Validation Residual Analysis Plots ▶