LabVIEW™

System Identification Toolkit User Manual

NATIONAL INSTRUMENTS™

April 2004 Edition
Part Number 371001A-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada (Calgary) 403 274 9391, Canada (Ottawa) 613 233 5949, Canada (Québec) 450 510 3055, Canada (Toronto) 905 785 0085, Canada (Vancouver) 514 685 7530, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427, India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Malaysia 603 9131 0918, Mexico 001 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 095 783 68 51, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227, Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

Except as specified herein, National Instruments makes no warranties, express or implied, and specifically disclaims any warranty of merchantability or fitness for a particular purpose. Customer's right to recover damages caused by fault or negligence on the part of National Instruments shall be limited to the amount theretofore paid by the customer. National Instruments will not be liable for damages resulting from loss of data, profits, use of products, or incidental or consequential damages, even if advised of the possibility thereof. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

LabVIEW™, National Instruments™, NI™, ni.com™, NI-DAQ™, and Xmath™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED “SYSTEM FAILURES”). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Conventions

The following conventions are used in this manual:

»

The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

💡

This icon denotes a tip, which alerts you to advisory information.

📝

This icon denotes a note, which alerts you to important information.

bold

Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic

Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.

`monospace`

Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

`monospace italic`

Italic text in this font denotes text that is a placeholder for a word or value that you must supply.

Contents

Chapter 1
LabVIEW System Identification

Chapter 2
System Identification Case Study

Appendix A
References

Appendix B
Technical Support and Professional Services

1

LabVIEW System Identification

System identification is a technique of building mathematical models of a dynamic system based on a set of measured stimulus and response data samples. You can use the models to simulate the system or to design a model-based controller for the system. System identification is fundamental for communication and control engineering, and it also plays important roles in many other areas, such as meteorology and economics. This manual describes how you can apply the LabVIEW System Identification Toolkit to your system identification needs.

LabVIEW System Identification Toolkit Overview

The LabVIEW System Identification Toolkit provides an interactive tool for identifying discrete-time multi-input and multi-output linear systems. You can think of single-input and single-output as a special case of multi-input and multi-output. This toolkit encompasses the entire identification process from raw data analysis to validation of identified models. You can identify large multivariable models of high-order systems from large amounts of data. With National Instruments hardware and the LabVIEW Real-Time Module, you can readily build a system for real-time system identification applications.

The LabVIEW System Identification Toolkit provides VIs encompassing the entire identification process, which includes the following:

- Data preprocess
- Parametric estimation
- Nonparametric estimation
- Recursive model estimation
- Model validation
- Model presentation
- Model conversion
- Model management
- Utility

System Requirements

- Windows 2000/NT/XP
- LabVIEW 7.1 Full or Professional Development System
- NI-DAQ 7.1 or later if the application requires NI hardware

Note Refer to the *LabVIEW Release Notes* for LabVIEW system requirements.

Tip System Identification is a memory-intensive task. Increasing the amount of RAM can significantly increase system performance.

Installation

Complete the following steps to install the LabVIEW System Identification Toolkit.

Note Some virus detection programs might interfere with installing the LabVIEW System Identification Toolkit. Disable any automatic virus detection programs before you install. After installation, check the hard disk for viruses and enable any virus detection programs you disabled.

1. On Windows 2000/NT/XP, log in as an administrator or as a user with administrator privileges.
2. If it is not already installed, install LabVIEW 7.1.
3. Install the LabVIEW System Identification Toolkit. If the LabVIEW System Identification Toolkit installation program does not run automatically, you can run the installer by running `x:/setup.exe`, where *x* is the CD-ROM drive on the computer.

The Basic Concepts of System Identification

System identification is an iterative process that includes acquiring, formatting, processing, and identifying raw data from a real-world system based on a mathematical model you select. You then determine if the resulting model fits the observed system behavior. If the results are unsatisfactory, you revise the parameters and iterate through the process. Figure 1-1 shows a typical system identification flowchart.

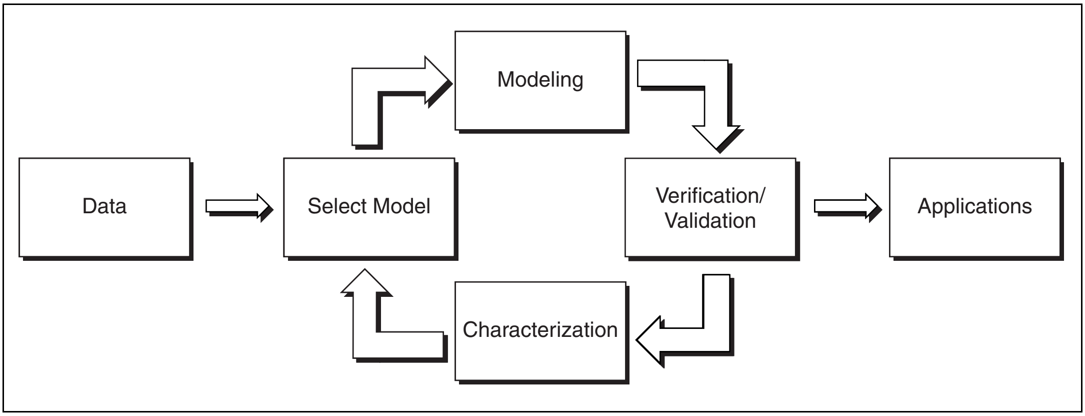


Figure 1-1. System Identification Application Flowchart

System identification applications include data acquisition and preprocessing, model estimation, and validation and verification.

Real-world systems seldom have one model that perfectly fits all of the observed behaviors because system identification involves many variables, such as sampling frequency, type of mathematical model, model order, and so on. You usually have a number of models you can use. Each of the models describe the system behavior only to some extent.

You might have multiple applicable algorithms available for the same model. For example, the LabVIEW System Identification Toolkit includes four different adaptive schemes for recursive system identification and five different algorithms for AR model estimation. The algorithms you select depend on the model structure, stochastic assumptions, and numerical properties of the algorithm.

You can use system identification in a wide range of applications, including chemical engineering, mechanical engineering, biology, physiology, meteorology, and economics. You must have a good understanding of the system under test, such as a motor, to devise an applicable control strategy. Designers use a system model of the relationship between the fuel flow and the shaft speed of turbojet engine to optimize the efficiency and operational stability of the jet engine. Biology and physiology researchers use system identification techniques to make progress in areas such as eye pupil response and heart rate control. Meteorologists and economists build mathematical models based on historical data for use in forecasting.

Data Acquisition and Preprocessing

The first step in system identification is collecting two sets of input-output data samples. You use one set of samples to estimate the mathematical model of the system. You use the second set of samples to verify the resulting model. If the resulting model does not meet the predefined specifications, such as prediction error, modify the settings and iterate the processing.

To verify the behavior of a system in a real-world environment, you can intentionally inject perturbations into the input data. For example, you can study the characteristics of an electric motor by varying the input voltage and observing how the motor responds. However, experimentation might be prohibited in many cases. For example, it might not be practical to introduce arbitrary perturbations to the input of a chemical plant because of safety concerns.

For the best results, you might need to preprocess the raw data samples you acquire, such as removing the trend, reducing random noise, and so on.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Data Preprocess VIs.

Model Estimation

The next step in system identification is computing the mathematical model of the unknown system. The LabVIEW System Identification Toolkit models the system as linear, which must be in a state-space or polynomial form. The two most common techniques for estimating linear time-invariant systems are the nonparametric method and the parametric method.

Nonparametric Model Estimation

A nonparametric method does not require the same amount of information as the parametric method, which requires the model type and model order. The nonparametric method is simple and efficient, but it is often less accurate than the parametric method. Nonparametric estimation uses the impulse response and frequency response methods to estimate the system.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Nonparametric Model Estimation VIs.

Parametric Model Estimation

Parametric estimation requires the model type and model order, which typically comes from some detailed knowledge of the system you are attempting to model. Parametric estimation is more involved and less efficient than nonparametric estimation, but it can yield more accurate results if you properly select the model type and model order. Prior knowledge of the real system is very useful when selecting the correct model type and the right order of the model. There are few systems in the real world that you can completely formulate using a linear mathematical model, so you usually need to try various models and model orders to obtain the best estimation. Refer to Chapter 2, *System Identification Case Study*, for information about adjusting the order of the model.

The LabVIEW System Identification Toolkit supports the following model types for parametric estimation:

- AR
- ARX
- ARMAX
- Output-Error (OE)
- Box-Jenkins (BJ)
- General Linear (GL)
- State-Space (SS)

You can represent linear discrete-time systems in state-space or polynomial forms. All of the above models are polynomial in form except the state-space model. From an input and output point of view, the model class that you choose is irrelevant because you can convert a polynomial model to a state-space form and a state-space form to a polynomial model. Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Model Conversion VIs.

From a results point of view, the model class that you choose is important because the results from a direct model and the results from a converted model are usually different. For example, the results from a polynomial model and the results from a state-space model that you convert to a polynomial model are usually different because of different parameter settings and different estimation approaches.

The state-space model is preferred for multiple input or multiple output systems because it is suitable for complex model structures, particularly when a state-space system is in canonical form. There is a one-to-one mapping between the model parameters and the input-output behavior within a model set when the state-space system is in canonical form. Input and output data samples can uniquely identify parameters.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Parametric Model Estimation VIs.

Recursive Model Estimation

System characteristics can evolve over time and cause the previous estimated characteristics to be invalid. In this case, you need to use real-time system identification or online system identification. The LabVIEW System Identification Toolkit includes the Recursive Model Estimation VIs, which allow you to work with time-varying systems to achieve real-time system identification.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Recursive Model Estimation VIs.

Validation and Verification

The final step in system identification is validating the results. You have two ways to validate the resulting models. You can apply Bode, Nyquist, and pole-zero plots and examine the dynamic characteristics of the estimated model. You also can compare the estimated model against the actual system. For example, you can apply part of the acquired input data samples to the estimated model. Compare the difference between the response signal and the corresponding output samples to determine if the estimated system models the actual system with acceptable accuracy. Reestimate the model if the accuracy is unsatisfactory.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about the Model Presentation VIs and the Model Validation VIs.

2

System Identification Case Study

This case study compares various order estimation techniques and the ARX model and state-space model estimation methods for data sets from a flexible robot arm. The data was adopted from version 7.0 of the *Xmath Interactive System Identification Module, Part 2*, which was originally collected by Hendrik Van Brussel and Jan Swevers of the Laboratory of Production Manufacture and Automation at Katholieke Universiteit Leuven.

The input data is the reaction torque of the structure on the ground, which is a multi sinewave with 200 frequency points equally spaced over the frequency band from 0.122 Hz to 24.4 Hz.

The output data is the acceleration of the flexible arm that contains information about the flexible resonance and anti-resonance modes.

Each set contains 4,096 samples at a sampling rate of 500 Hz.

The VIs for this case study are located in `examples\system identification\SICaseStudy1.llb`. As you read the discussion in this chapter, open the VIs in `SICaseStudy1.llb` as you encounter them in the text and run them to better evaluate the resulting plots and the System Identification VIs used on the block diagram.

Refer to the *LabVIEW Help*, available by selecting **Help»VI, Function, & How-To Help**, for information about specific System Identification VIs in this case study.

Data Preprocessing

The Flexible Arm Data Set VI in Figure 2-1 shows the time waveform of torque, which is the measured input sample, and acceleration, which is the measured output sample. There is no obvious nominal, trend, or outlier values in the input or output time waveforms.

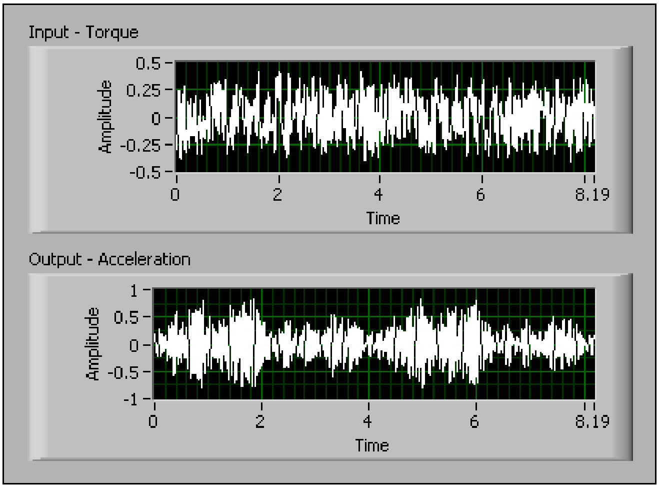


Figure 2-1. Flexible Arm Data Set in the Time Domain

The Non-Parametric FRF VI in Figure 2-2 shows the frequency response of the measured output signal. There are at least three resonances and two anti-resonances in the frequency domain. One resonance is at approximately 42 Hz. Noise or the system's non-linear behavior cause the peak at approximately 42 Hz because the frequency of the input data is between 0.122 Hz and 24.4 Hz.

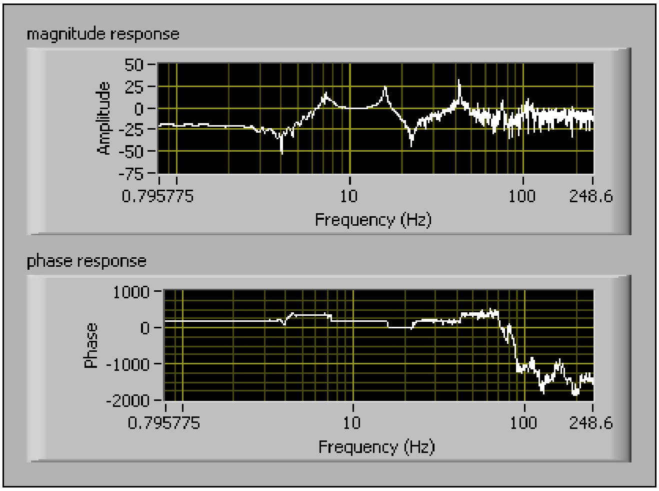


Figure 2-2. Flexible Arm Data Set in the Frequency Domain

The corresponding block diagram in Figure 2-3 uses the SI Estimate Frequency Response VI to compute the frequency response of the measured output signal. The **window length** input parameter is 4096 and specifies the length of the window used in the frequency computation.

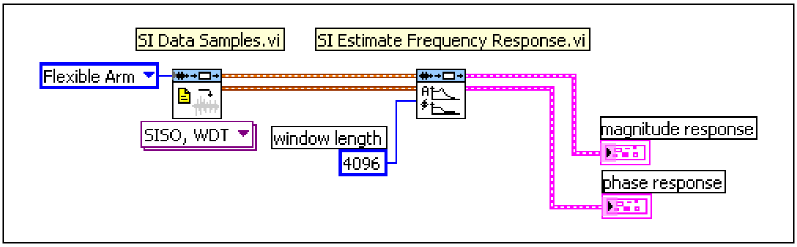


Figure 2-3. Block Diagram for Computing Frequency Response

You can apply a lowpass filter to the raw data sample before performing system identification to eliminate high frequency noise. According to sampling theory, you can perform downsampling to reduce the number of samples and the computation complexity when you reduce the frequency bandwidth to 25 Hz.

The Non-Parametric FRF with Prefiltering and Down Sampling VI in Figure 2-4 shows the frequency response after downsampling.

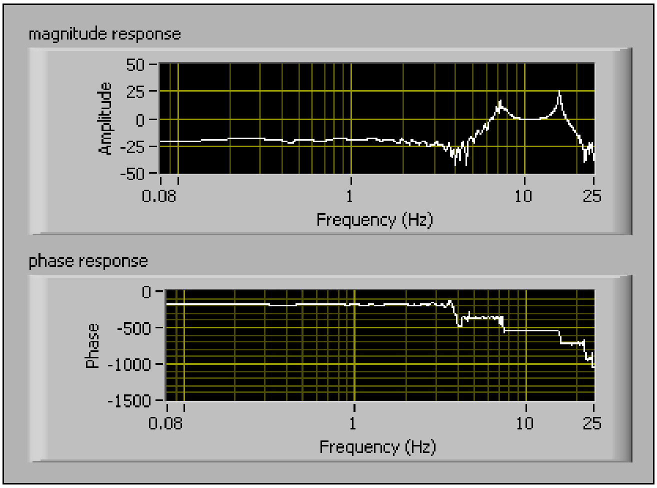


Figure 2-4. Frequency Response after Downsampling

The corresponding block diagram in Figure 2-5 adds the SI Lowpass Filter VI and the SI Down Sampling VI to eliminate the fake resonance at approximately 42 Hz and substantially reduce the number of samples to process.

The SI Lowpass Filter VI applies a lowpass filter before downsampling to avoid non-realistic behavior. Set the **cutoff frequency** input parameter to 25 so the effective frequency bandwidth is one-tenth of the original sampling frequency. The downsampling ratio can be as high as 10. The SI Down Sampling VI reduces the sampling rate using a decimation factor of 10. The lowpass filter and downsampling avoids the high frequency disturbance and makes the process faster and more efficient.

The **window length** input parameter of the SI Estimate Frequency Response VI in Figure 2-5 is 400 instead of 4096 as Figure 2-3 shows because the data set is downsampled by a factor of 10. The number of samples is one-tenth of the raw data.

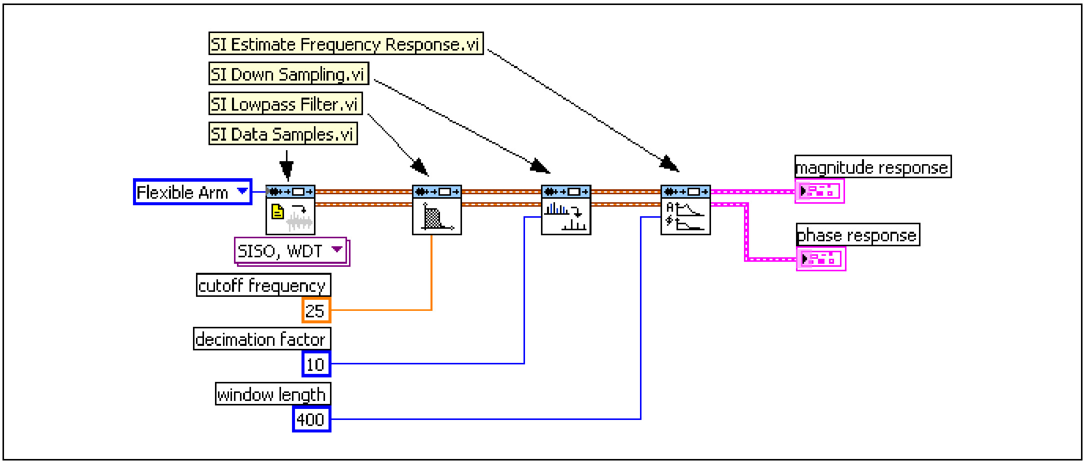


Figure 2-5. Block Diagram for Downsampling

Order Estimation

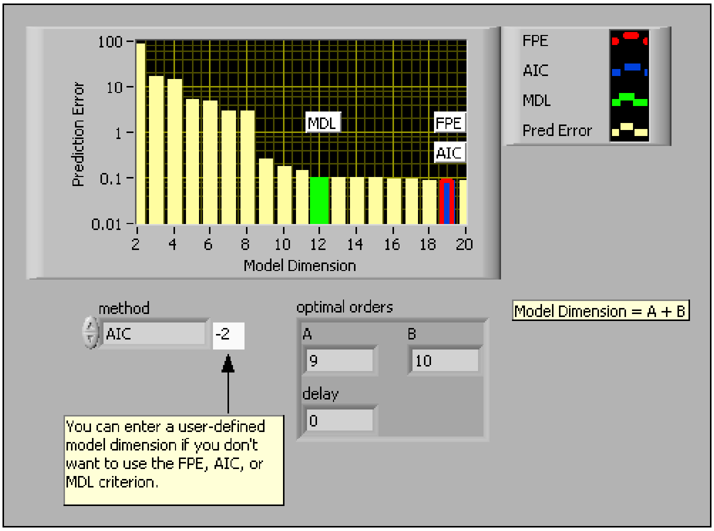
One of the biggest challenges in model estimation is selecting the correct order. The LabVIEW System Identification Toolkit supports three different criteria to estimate the orders.

- FPE—Akaike's final prediction error criterion
- AIC—Akaike's information theoretic criterion
- MDL—Minimum description length criterion

Sometimes the results you obtain with these three criteria might be inconsistent. You can use a pole-zero plot for further investigation and to verify the results.

The Estimate Orders of ARX Model VI in Figure 2-6 shows a prediction error plot. The y-axis is the prediction error and the x-axis is the model dimension. For an ARX model, the model dimension is equal to A order + B order. The three different color bars on the chart represent the FPE, AIC, and MDL criteria. The lowest prediction error usually corresponds to the optimal order. Figure 2-6 uses the AIC criterion so the prediction error corresponds to the following optimal orders:

- A order equals 9
- B order equals 10
- delay equals 0



method: AIC -2

optimal orders
A: 9
B: 10
delay: 0

You can enter a user-defined model dimension if you don't want to use the FPE, AIC, or MDL criterion.

Figure 2-6. Prediction Error Plot for an ARX Model

The corresponding block diagram in Figure 2-7 uses the SI Estimate Orders of System Model VI for order estimation. The **optimal orders** output parameter contains the A order and B order. The SI Split Signal VI partitions the preprocessing data samples into a portion for model estimation and a portion for model validation. The **1st portion** input parameter is 66, which means 66% of the data samples are used for estimation and the remainder of the data samples are used for validation. The SI Estimate Orders of System Model VI generates the prediction error plot.

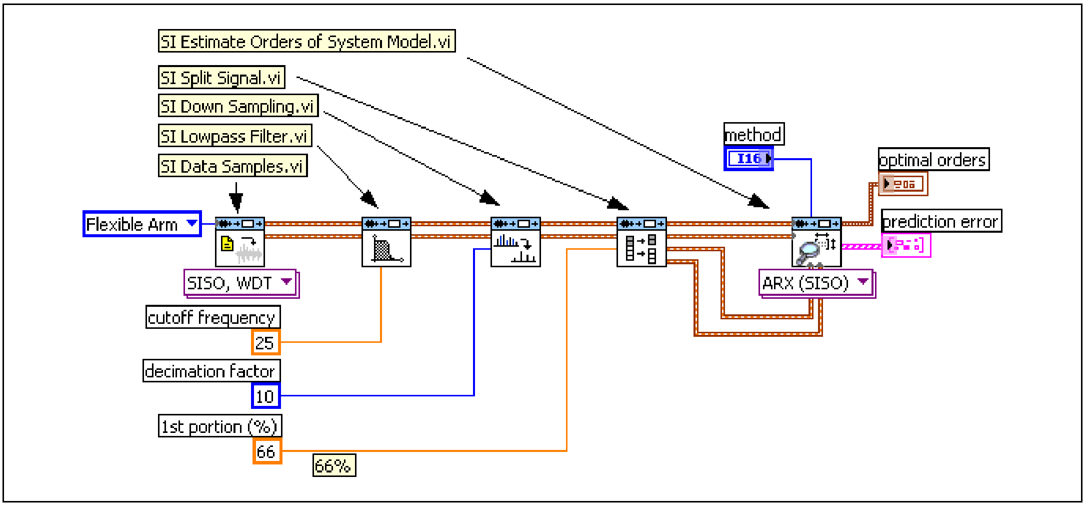


Figure 2-7. Block Diagram for Order Estimation for an ARX Model

The Pole-Zero Cancellation VI in Figure 2-8 shows a pole-zero plot with three close pole-zero pairs. You can use a pole-zero plot to determine redundancy. If a pole and a zero overlap, the pole and zero cancel out each other, which indicates that the estimated optimal order is too high.

Due to numerical error, it is unlikely that a zero and a pole perfectly overlap. You can use the confidence region to justify whether the pole and the zero cancel out each other.

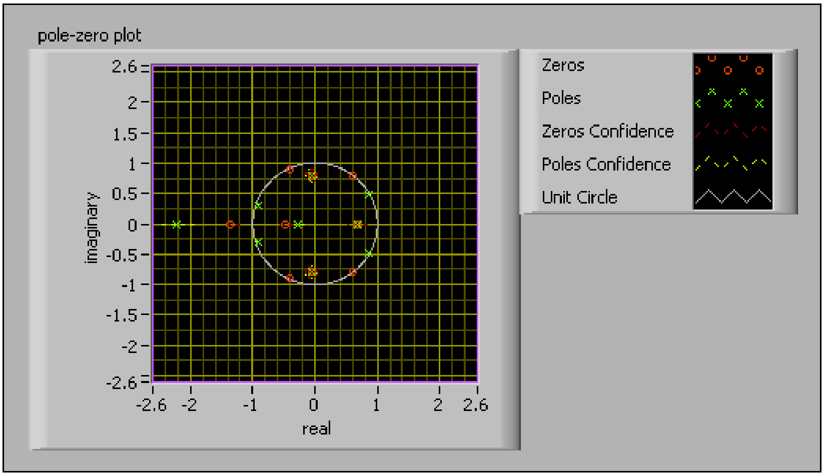
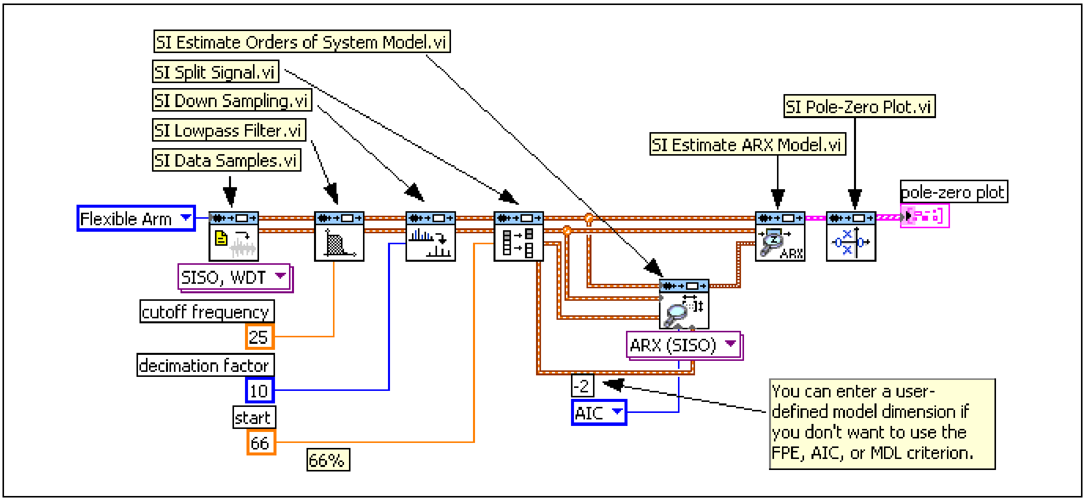


Figure 2-8. Pole-Zero Plot for an ARX Model



You can enter a user-defined model dimension if you don't want to use the FPE, AIC, or MDL criterion.

Figure 2-9. Block Diagram of a Pole-Zero Plot for an ARX Model

The Pole-Zero Cancellation VI in Figure 2-10 shows a pole-zero plot with one pole outside the unit circle. This pole-zero plot uses the MDL criterion so the prediction error corresponds to the following optimal orders:

- A order equals 6
- B order equals 6
- delay equals 0

Compare Figure 2-10, which uses the MDL criterion and Figure 2-8, which uses the AIC criterion. The three close pairs in Figure 2-8 have disappeared. This implies that the MDL criterion fits better than the AIC criterion in this particular example.

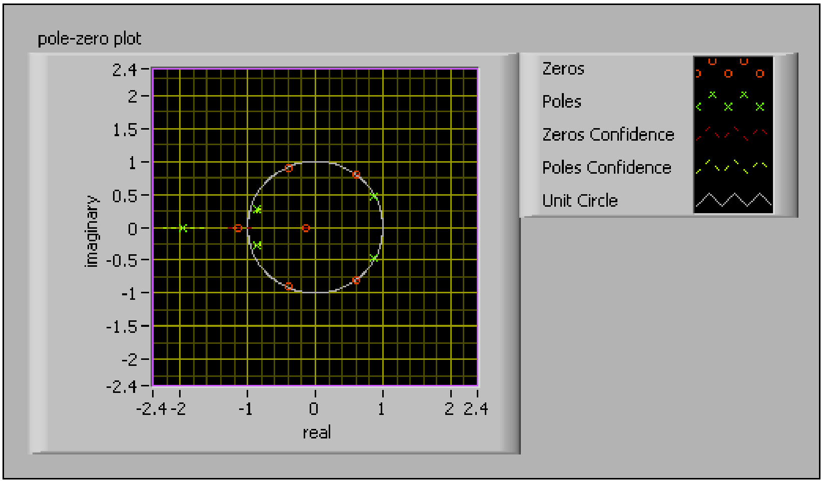


Figure 2-10. Pole-Zero Plot for an MDL Model

In addition to examining redundancy, you also can use the pole-zero plot for other purposes. For example, Figure 2-8 and Figure 2-10 show poles outside the unit circle. This implies that the ARX system based on the AIC or MDL criteria might be unstable. One way to stabilize the system is to change the order. In addition to the FPE, AIC, and MDL criteria, you can set user-defined orders in the SI Estimate Orders of System Model VI.

Figure 2-11 shows a pole-zero plot with user-defined orders. This pole-zero plot uses the following optimal orders:

- A order equals 4
- B order equals 5
- delay equals 0

Compare Figure 2-11 with Figure 2-8 and Figure 2-10. The poles outside the unit circle in Figure 2-8 and Figure 2-10 have disappeared.

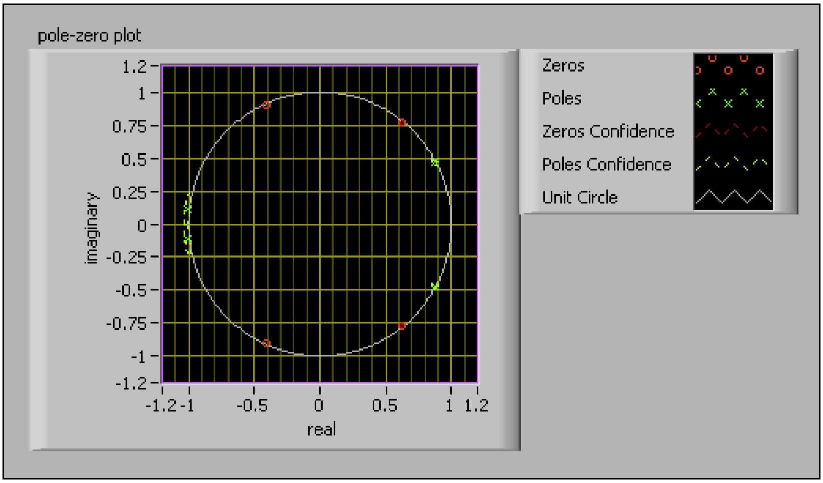
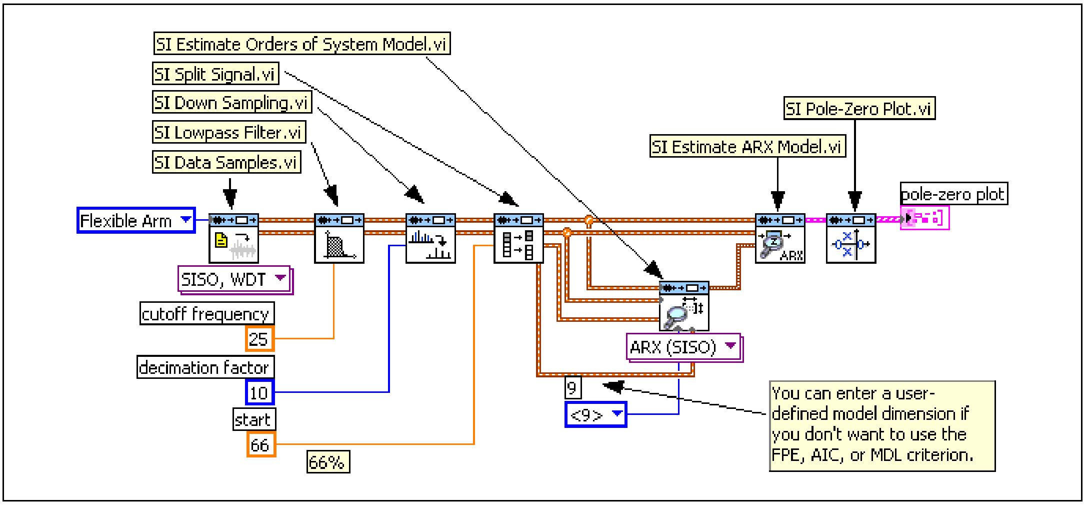


Figure 2-11. Pole-Zero Plot with User-Defined Order

The corresponding block diagram in Figure 2-12 uses a user-defined model dimension of 9 instead of the AIC model dimension of –2 used in Figure 2-9.



You can enter a user-defined model dimension if you don't want to use the FPE, AIC, or MDL criterion.

Figure 2-12. Block Diagram for Pole-Zero Plot with User-Defined Order

ARX Model Estimation and Verification

The Simulation & Prediction with ARX Model VI in Figure 2-13 shows simulation and prediction to verify an ARX model.

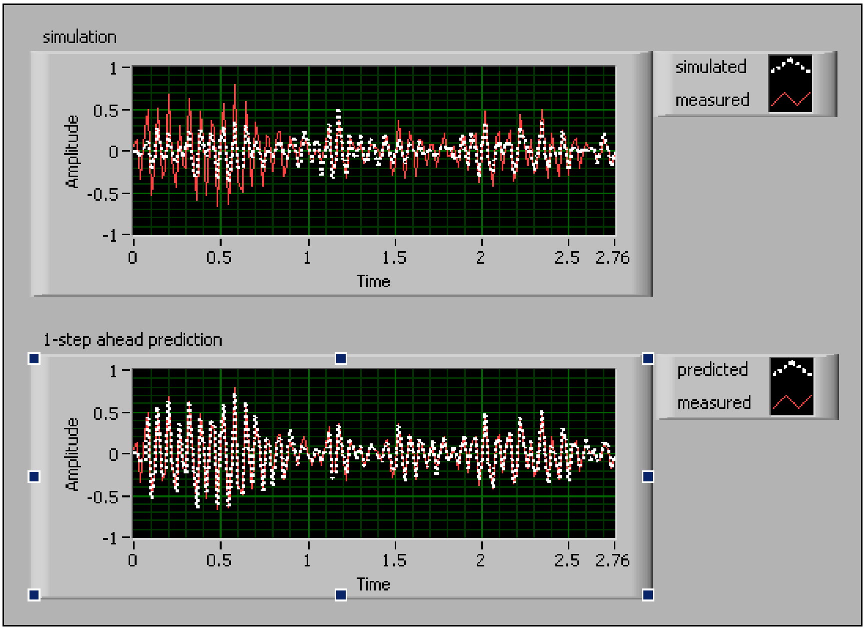


Figure 2-13. Simulation and Prediction for an ARX Model

The corresponding block diagram in Figure 2-14 uses the SI Model Simulation VI to simulate the system model and the SI Model Prediction VI to perform prediction of the system model.

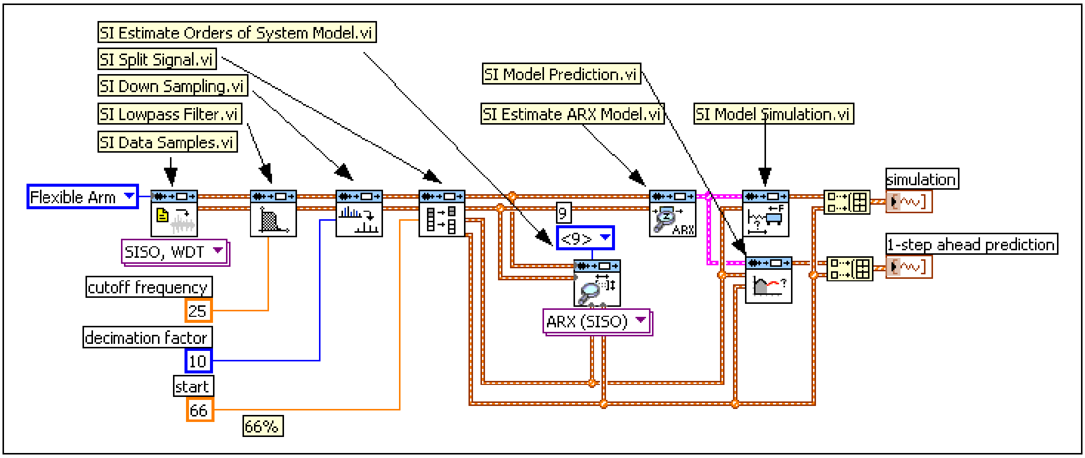


Figure 2-14. Block Diagram for Model Simulation and Prediction for an ARX Model

In addition to simulation and prediction, you also can apply a residual analysis to verify the system model. Residual analysis tests whether the prediction error correlates to the stimulus signal. Prediction errors are usually uncorrelated with all stimulus signals in an open-loop system. For a closed-loop system, prediction errors usually are uncorrelated with past stimulus signals.

The Residual Analysis VI in Figure 2-15 shows ideal results, where both auto-correlation and cross-correlation are inside the confidence region except those in the vicinity of $\tau = 0$. Insufficient system order can cause a correlation between the stimulus and the residue. Figure 2-15 uses the following optimal orders:

- A order equals 4
- B order equals 5
- delay equals 0

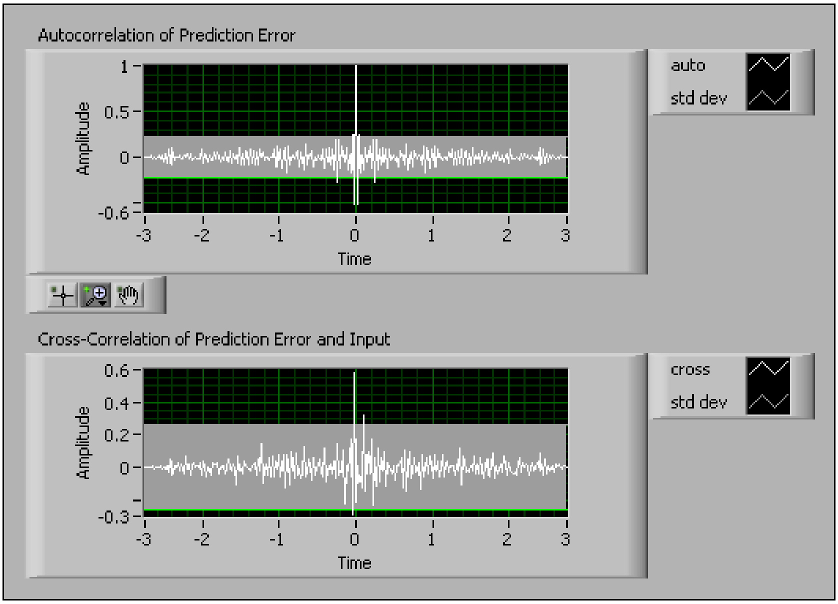


Figure 2-15. Residual Analysis for A order = 4, B order = 5, and delay = 0

The Residual Analysis VI in Figure 2-16 shows the auto-correlation and cross-correlation residual analysis for the following orders:

- A order equals 2
- B order equals 2
- delay equals 0

The auto-correlation and cross-correlation indicates the residue is not white noise and correlates with the stimulus signal.

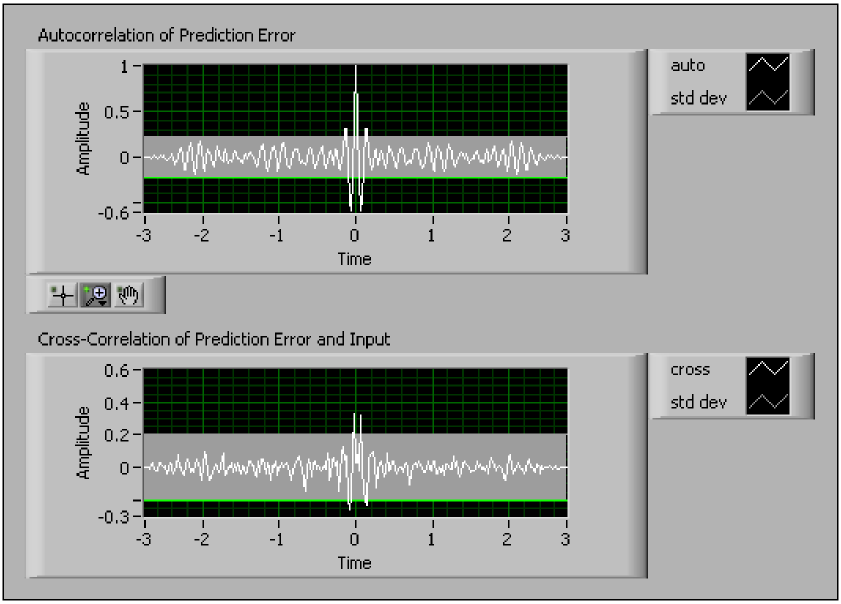


Figure 2-16. Residual Analysis for A order = 2, B order = 2, and delay = 0

The corresponding block diagram in Figure 2-17 uses the SI Model Residual Analysis VI to compute and analyze the prediction error.

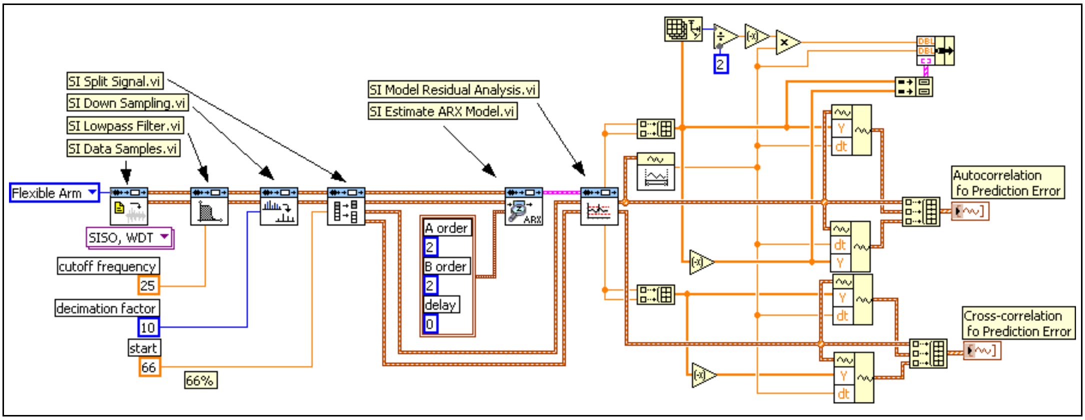
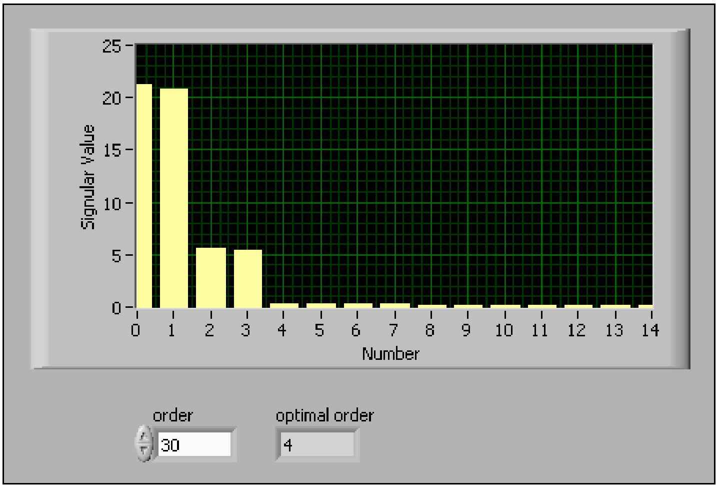


Figure 2-17. Block Diagram of Residual Analysis for A order = 2, B order = 2, and delay = 0

State-Space Model Estimation

Order estimation is always a challenge with the ARX model. In the state-space model, order estimation is usually equal to the number of significant singular values.

The Estimate Orders of State-Space Model VI in Figure 2-18 shows a singular value plot with four leading singular values. This is the number of states.



order 30

optimal order 4

Figure 2-18. Singular Value Plot for State-Space Model

The corresponding block diagram in Figure 2-19 uses the SI Estimate Orders of System Model VI to estimate the number of states.

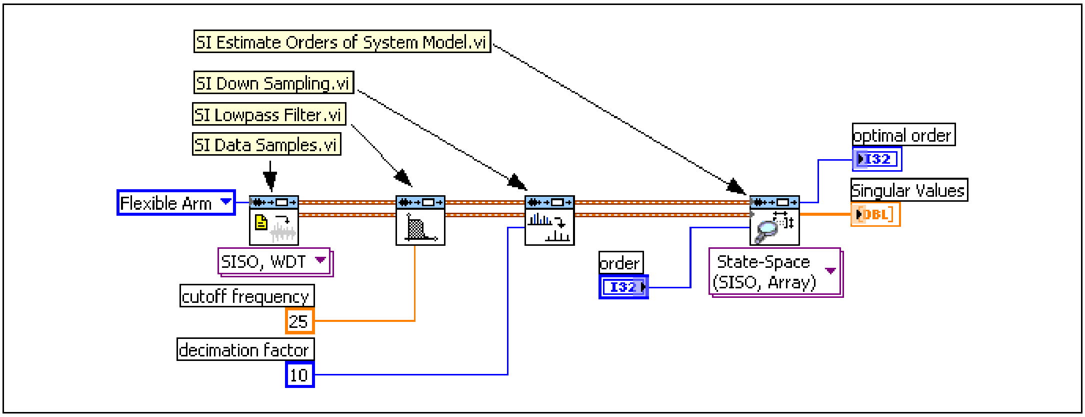


Figure 2-19. Block Diagram for Estimating the Number of States

The Simulation & Prediction with State-Space Model VI in Figure 2-20 shows simulation and prediction plots for a state-space model to verify the quality of a system. Compare the state-space model in Figure 2-20 to the ARX model in Figure 2-13. The state-space model has less simulation and prediction errors.

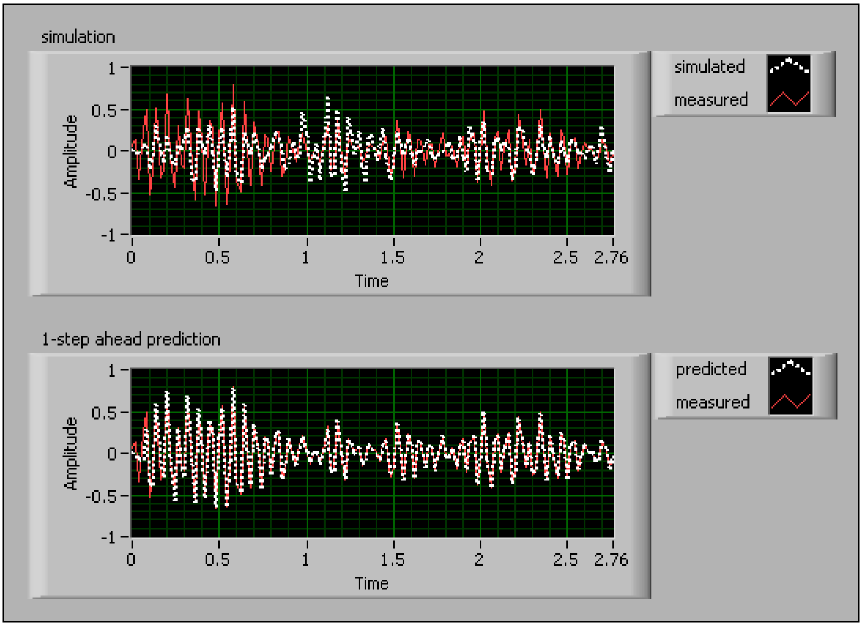


Figure 2-20. Simulation and Prediction for a State-Space Model

The corresponding block diagram in Figure 2-21 shows a complete state-space model estimation and validation block diagram.

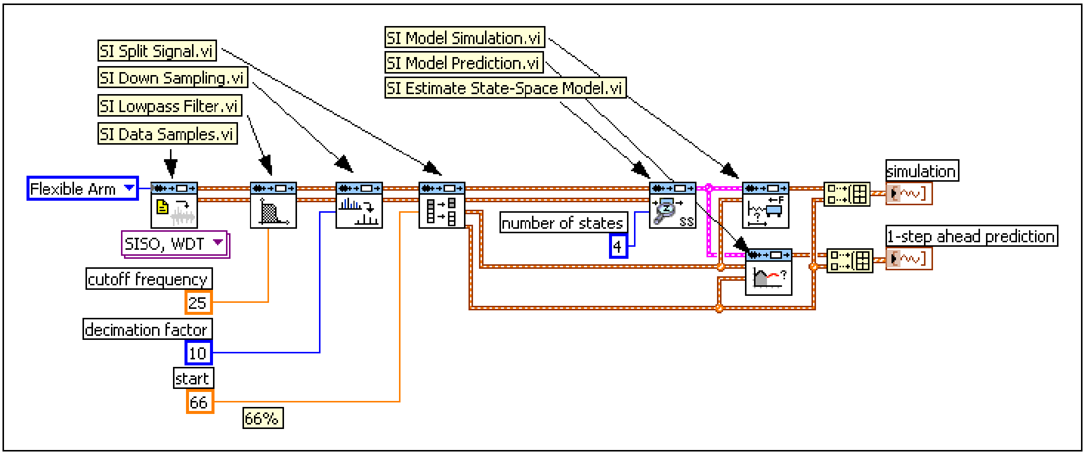


Figure 2-21. Block Diagram of Estimation and Validation for a State-Space Model

A

References

This manual contains information about using the LabVIEW System Identification Toolkit VIs in LabVIEW applications. In most cases, the underlying theory is omitted. If you are interested in the rigorous mathematical treatment of system identification techniques and algorithms, refer to the following textbooks and technical papers, which National Instruments used to develop this toolkit.

Astrom, K. J., and B. Wittenmark. 1974. *Adaptive control*. 2d ed. Addison-Wesley.

De Moor, B. 1988. *Mathematical concepts and techniques for modelling of static and dynamic systems*. Ph.D. thesis. Katholieke Universiteit Leuven, UDC 519.17.

Gill, P. E., W. Murray, and M. H. Wright. 1981. *Practical optimization*. Academic Press.

Goodwin, G. C., and R. L. Payne. 1977. *Dynamic system identification: Experiment design and data analysis*. Academic Press.

Kailath, T. 1980. *Linear systems*. Prentice Hall.

Ljung, L. 1999. *System identification theory for the user*. 2d ed. Prentice Hall.

Ljung, L., and T. Glad. 1994. *Modeling of dynamic systems*. Prentice Hall.

Ljung, L., and T. Söderström. 1983. *Theory and practice of recursive identification*. MIT Press.

Oppenheim, A. V., A. S. Willsky, and I. T. Young. 1997. *Signals and systems*. Prentice Hall.

Oppenheim, A. V., and R. W. Schafer. 1989. *Discrete-time signal processing*. 2d ed. Prentice Hall.

Overschee, P. V., and B. De Moor. 1993. *N4SID: Subspace algorithms for the stochastic identification problem*. Automatica 29, no. 3:649–660.

———. 1994. *Subspace algorithms for the identification of combined deterministic-stochastic systems*. Automatica 30, no. 1:75-93.

Powell, M. J. D. 1964. *An efficient method for finding the minimum of a function of several variables without calculating derivatives*. Computer Journal 7:155–162.

Priestley, M. 1981. *Spectral analysis and time series*. Academic Press.

B

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at `ni.com` for technical support and professional services:

- **Support**—Online technical support resources at `ni.com/support` include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Application Engineers worldwide in the NI Developer Exchange at `ni.com/exchange`. National Instruments Application Engineers make sure every question receives an answer.
- **Training and Certification**—Visit `ni.com/training` for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit `ni.com/alliance`.

If you searched `ni.com` and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of `ni.com/niglobal` to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.