NIST
United States Department of Commerce
Technology Administration
National Institute of Standards and Technology

NIST Technical Note 1428

High-Dimensional Empirical Linear Prediction (HELP) Toolbox

Users's Guide, Version 2.2

Andrew D. Koffman, T. Michael Souders, Gerard N. Stenbakken, and Hans Engler

The National Institute of Standards and Technology was established in 1988 by Congress to "assist industry in the development of technology . . . needed to improve product quality, to modernize manufacturing processes, to ensure product reliability . . . and to facilitate rapid commercialization . . . of products based on new scientific discoveries."

NIST, originally founded as the National Bureau of Standards in 1901, works to strengthen U.S. industry's competitiveness; advance science and engineering; and improve public health, safety, and the environment. One of the agency's basic functions is to develop, maintain, and retain custody of the national standards of measurement, and provide the means and methods for comparing standards used in science, engineering, manufacturing, commerce, industry, and education with the standards adopted or recognized by the Federal Government.

As an agency of the U.S. Commerce Department's Technology Administration, NIST conducts basic and applied research in the physical sciences and engineering, and develops measurement techniques, test methods, standards, and related services. The Institute does generic and precompetitive work on new and advanced technologies. NIST's research facilities are located at Gaithersburg, MD 20899, and at Boulder, CO 80303. Major technical operating units and their principal activities are listed below. For more information contact the Publications and Program Inquiries Desk, 301-975-3058.

Office of the Director

- National Quality Program
- International and Academic Affairs

Technology Services

- Standards Services
- Technology Partnerships
- Measurement Services
- Technology Innovation
- Information Services

Advanced Technology Program

- Economic Assessment
- Information Technology and Applications
- Chemical and Biomedical Technology
- Materials and Manufacturing Technology
- Electronics and Photonics Technology

Manufacturing Extension Partnership Program

- Regional Programs
- National Programs
- Program Development

Electronics and Electrical Engineering Laboratory

- Microelectronics
- Law Enforcement Standards
- Electricity
- Semiconductor Electronics
- Electromagnetic Fields[1]
- Electromagnetic Technology[1]
- Optoelectronics[1]

Chemical Science and Technology Laboratory

- Biotechnology
- Physical and Chemical Properties[2]
- Analytical Chemistry
- Process Measurements
- Surface and Microanalysis Science

Physics Laboratory

- Electron and Optical Physics
- Atomic Physics
- Optical Technology
- Ionizing Radiation
- Time and Frequency[1]
- Quantum Physics[1]

Materials Science and Engineering Laboratory

- Intelligent Processing of Materials
- Ceramics
- Materials Reliability[1]
- Polymers
- Metallurgy
- NIST Center for Neutron Research

Manufacturing Engineering Laboratory

- Precision Engineering
- Automated Production Technology
- Intelligent Systems
- Fabrication Technology
- Manufacturing Systems Integration

Building and Fire Research Laboratory

- Structures
- Building Materials
- Building Environment
- Fire Safety Engineering
- Fire Science

Information Technology Laboratory

- Mathematical and Computational Sciences[2]
- Advanced Network Technologies
- Computer Security
- Information Access and User Interfaces
- High Performance Systems and Services
- Distributed Computing and Information Services
- Software Diagnostics and Conformance Testing
- Statistical Engineering

[1] At Boulder, CO 80303.

[2] Some elements at Boulder, CO.

NIST Technical Note 1428

High-Dimensional Empirical Linear Prediction (HELP) Toolbox

Users's Guide, Version 2.2

Andrew D. Koffman,
T. Michael Souders, and
Gerard N. Stenbakken

Electricity Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899-0001

and

Hans Engler

Department of Mathematics,
Georgetown University,
Washington, DC 20057

May 1999

U.S. Department of Commerce
William M. Daley, Secretary

Technology Administration
Gary Bachula, Acting Under Secretary for Technology

National Institute of Standards and Technology
Raymond G. Kammer, Director

National Institute of Standards
and Technology
Technical Note 1428
Natl. Inst. Stand. Technol.
Tech. Note 1428
54 pages (May 1999)
CODEN: NTNOEF

U.S. Government Printing Office
Washington: 1999

For sale by the Superintendent of
Documents
U.S. Government Printing Office
Washington, DC 20402

HELP Toolbox User's Guide

CONTENTS

I. Introduction

The High-dimensional Empirical Linear Prediction (HELP) Toolbox is an optimization tool designed specifically to meet the requirements of test and measurement engineers. For many electronic devices and instruments, it is not physically or economically feasible to perform exhaustive testing. Therefore, test engineers must formulate abbreviated test plans that are economical to execute but still yield accurate measures of the overall performance of the tested products. The HELP Toolbox incorporates a new approach for optimizing the testing of electronic devices and instruments. The method, high-dimensional empirical linear prediction, is currently being used by mixed-signal integrated circuit manufacturers to reduce the costs of testing their products, and it is also being used at the National Institute of Standards and Technology (NIST) to reduce customer's costs for selected calibration services. Examples of products that can benefit from this approach range from multi-range precision instruments to programmable filters to integrated circuit analog-to-digital (A/D) and digital-to-analog (D/A) converters. However, devices that are completely digital (digital inputs as well as outputs), are not supported.

The approach is based on a simple mathematical model that relates the device response at all candidate test conditions to a set of underlying variables. Once an accurate model has been developed, algebraic operations on the model are used to:

a) select an optimal set of test points that will minimize the test effort required to achieve a specified level of confidence,

b) estimate the parameters of the model from measurements made at the selected test points,

c) predict the response of the device at all candidate test points (from measurements made at the selected test points) as a basis for accepting or rejecting units, and

d) compute statistical intervals (uncertainty bounds) for the predicted response, and test the validity of the model, on-line.

The entire process including model development can be performed with the HELP Toolbox, a NIST-developed graphical software package for use with MATLAB®† specifically tailored to this application. While a general understanding of the underlying principles is desirable, no mathematical programming is required of the operator. HELP places special emphasis on empirical modeling using measurement data collected previously on devices similar to the units under test. Empirical models require no detailed knowledge of the internal device architecture, yet they can be both accurate and efficient.

In addition to test optimization, the Toolbox is also useful for exploring the structures that underlie the behavior of tested devices. For example, it can reveal how many variables are actually needed to explain the behavior, and what their characteristic signatures look like. It can warn production engineers when the manufacturing process undergoes hidden changes, and it can even be used to help diagnose the likely causes.

While the Toolbox is intended for production testing applications, it is not designed for on-line use. Models are developed and tested within the Toolbox off-line, usually from empirical data on representative test units, and an optimal set of test points is selected. Once created, the models and test point vectors can be exported to the test system's on-line processor which then drives the testing and calculates the predicted global responses of test devices from on-line

† In order to describe the procedures discussed in this paper, commercial products are identified. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology or that the materials or equipment specified are necessarily the best available for the purpose.

measurements at the selected test points. The required calculations can be executed very quickly with any up-to-date personal computer or workstation.

The Toolbox software has been developed using the MATLAB programming environment and consequently requires that MATLAB be installed on the host processor. While detailed knowledge of MATLAB is not required, the user will benefit from some familiarity with it.

Chapter II of this user's manual provides general background on the theory and algebraic tools that form the basis for the Toolbox, while chapter III gives a description of the Toolbox menus and variables that are available to the user. Brief descriptions of the software architecture, subroutines and the global variables that are used are found in chapter IV. Finally, section V takes the user through several typical modeling and analysis situations involving two different products, the first a multirange precision instrument, and the second an integrated circuit A/D converter. Real measurement data are used in both examples.

II. Empirical Linear Modeling: An Overview

1. The Need for Efficient Testing

Testing is a critical step for assuring the quality of electronic devices. For complicated devices, the cost of testing is quite significant and may exceed 20 percent of the purchase price of a device. Efficient yet reliable testing strategies can therefore result in substantial savings. On the other hand, it is important to assure the quality of every individual device. This need obviously rules out statistical techniques such as deciding about the quality of an entire lot of devices from the results of testing a sample: Every device must be tested in some form.

For efficient testing strategies, the key observation is that the number of test points is often much larger than the number of parameters that is expected to determine device behavior. For example, a 13-bit A/D converter has $2^{13} = 8192$ possible test points. However, only a few dozen parameters are expected to determine the behavior for such a device. In fact, examining the circuit topology often results in an overestimate for the number of device parameters, due to production processes in which components are manufactured simultaneously. If device behavior is determined by a relatively small number of parameters, it should also be possible to predict its behavior and to decide about the quality of a device from a reduced set of measurements.

Efficient testing strategies try to identify the parameters that govern the error behavior of a device type and build a mathematical model for it. For a given new device, these parameters are then determined from measurements at a reduced set of test points, and the mathematical model is used to compute the device response at all test points. This approach raises the issues of how to construct the model, how to assess its accuracy, how to select an optimal measurement subset, how to find device parameters from the subset of measurements, and how to assess the reliability of a decision about the quality of a device that has been reached in this way.

2. Linear Models

We consider a device whose behavior can be exhaustively measured at m different test points. (If the test space is continuous, m represents some reasonably dense sampling of that space.) The actual behavior at each of these test points differs from the nominal one by a quantity that is here called the device response. The goal of linear modeling is to produce a "condensed" description of the response patterns of a device and to use this to predict the response of an individual device from a suitably selected set of test points with known reliability. We assume that the set of m candidate test points is specified *a priori*; we do not consider here how it should be chosen.

We denote the true device response by the column vector $\bar{y}$ with m components, the measured device response by y, and the vector of measurement errors by ε'. Then the equation $y = \bar{y} + \varepsilon'$ holds. It is assumed that any measurement bias has been identified and corrected beforehand; therefore, the averages of the elements of ε' for a very large number of repeated measurements are zero. For the same reason, some information about the size of ε', such as the (order of magnitude of the) standard deviations, is expected to be known. The units at each test point should be chosen such that the uncertainties across the test-point set are as similar in magnitude as possible. This step may require that the data be renormalized from test point to test point. If this is not done, then test points with data that are particularly large in magnitude, due to large test-point standard deviation, will have a stronger influence on the test outcome than other points.

In linear modeling, it is assumed that the true device response of any fixed device can be expressed in the form $\bar{y} = Ax + \boldsymbol{R}$. Here A is an $m \times n$ model matrix. It is specific to the device type and incorporates information that depends on the device design, its components, its production process, etc. The $n \times 1$ vector, x, consists of parameters that are specific to an individual device. $\boldsymbol{R}$ denotes a remainder term. At the outset, neither the matrix A nor the number n of its columns is known.

The parameter vector x sometimes has some actual physical meaning. For instance, it may reflect some properties of the components of a device. In this case, some nonlinear model can often describe the true response, and by linearizing about the nominal behavior, a linear model is obtained. Models that are derived on the basis of such considerations are called **physical models.**

In other cases, the columns of A represent typical response patterns that are based on engineering considerations. For example, they may describe contributions from single bit errors or simple superposition errors for A/D converters, with the components of x describing the contribution from each response pattern to the actual device response. Models that are derived in this form are called ***a priori* models.**

If the components of A are determined purely empirically from the responses of a "modeling set" of devices that have been tested exhaustively, the result is called an **empirical model.** Typically, the model matrix A in this case is not uniquely determined, the parameter vector x has no clear meaning, and statistical methods have to be used to construct the model [1].

If combinations of these approaches are used, the result will be called a **mixed model.**

The Toolbox has provisions for incorporating physical and *a priori* model information with empirical information to construct mixed models. However, the main focus of the Toolbox is on empirical linear models and their use. In such cases, new devices must be sufficiently similar to the devices in the modeling set, and the modeling set must be sufficiently homogeneous in this respect.

To be able to work with a linear model, the remainder term y is lumped together with the measurement error ε'. The result is denoted by ε. This model error is treated as a random quantity, and its statistical properties must therefore be determined. Thus, the linear model can be described by the equation $y = Ax + \varepsilon$.

3. Construction of Empirical Linear Models

To construct an empirical model, we assume that an $m{\times}p$ matrix of modeling data $\overline{A}$ is given. Its columns are vectors of complete measurements, one for each of the p devices in the modeling set. We want to extract an n-dimensional approximation of these response patterns, i.e., an $m{\times}n$ model matrix A such that the columns of $\overline{A}$ can (nearly) be expressed in terms of the columns of A. There may also be an additional set of validation data, i.e., q empirical vectors of exhaustive measurements in a matrix Y that are to be used to assess the quality of the model.

The number of parameters, n, must be determined along with the model matrix, A. This number should be kept small, but it should be large enough to explain all but a small portion of a typical device response. The choice of n depends on the amount of noise included in the modeling data. One expects to obtain the columns of the model matrix A as linear combinations of the modeling data in $\overline{A}$.

In order to construct an empirical model matrix from the matrix of modeling data $\overline{A}$, the Toolbox first computes its singular value decomposition $\overline{A} = USV^T$ [2]. Here U has size $m{\times}p$ with orthonormal columns, V has size $p{\times}p$ with orthonormal columns, and the matrix $S = \mathbf{diag}(s_1, s_2, \ldots, s_p)$ contains the singular values s_i that are non-negative and decreasing. One then chooses the model matrix as $A = U_1$, consisting of the first n columns of the left orthogonal factor U. It is known that no model matrix with n columns gives a better approximation of the modeling data $\overline{A}$ with respect to a number of approximation criteria. The columns of U_1 can be viewed as the principal patterns in the device behavior, and the numbers s_i describe the size of the contributions of these patterns.

The key problem now is the choice of n, the number of parameters. This choice is usually based on the sequence $s_1, s_2, \ldots$. The reason is that when the model A that is computed in this form is fitted to the modeling data, the mean squared sizes of the residuals are given by

$$\frac{1}{mp}\sum_{i=n+1}^{p} s_i^2 .$$

These numbers are expected to be smaller than the residuals for new devices, but the bias can be corrected. There may also be a validation set available, i.e., an $m \times q$ matrix Y of complete measurements for q devices.

To choose n, one begins with a visual inspection of a plot of the singular values, s_i. Such a plot sometimes reveals an "elbow", i.e., a fairly steep decline of the s_i up to a value s_n, followed by a nearly flat part. This characteristic may become more visible if the logarithms of the s_i are plotted. The latter plot is sometimes called a log of eigenvalue (LEV) plot and is provided by the Toolbox. It is recommended [3] to choose n such that the first value of the "flat" portion is still included. It may also occur that the LEV plot has a "step" shape. In that case, the parameters up to and including the bottom of the step should be chosen.

For a selection of n that can be justified more rigorously under certain statistical assumptions, one sets

$$Q(n) = \frac{\left(\sum_{i>n} s_i^2\right)^2}{\sum_{i>n} s_i^4} \qquad \text{for } n = 1,2,\ldots,p-1.$$

The number $Q(n)$ is never larger than p-n, which is the dimension of the set of residuals of modeling data from a model with n parameters. It can be viewed as a statistical estimate of the dimension of the set of residuals. Thus, if $Q(n)$ is small, the residuals "look" as if they are concentrated on a low-dimensional set. One should then increase the number of parameters in order to capture the few additional patterns that appear to dominate the residuals. On the other hand, the $Q(n)$ will decrease for large n. The Toolbox plots the $Q(n)$ in the Q-Max plot, and the recommendation is to pick the value of n for which the plotted quantity is maximal or the smallest value for which it reaches a plateau.

A third procedure for choosing n consists of selecting this number such that the expected residuals at individual test points for new devices are sufficiently small. For many applications, this criterion reflects the ultimate goal of achieving a given level of accuracy with the minimum number of parameters or test points. Based on the modeling set data, the Toolbox computes and plots an estimate of the root-mean-square residuals $c(n)$ that would result from fitting a new device to models with n parameters for $n = 1,2,\ldots.$ This computation is essentially the quantity

$$c(n) = \sqrt{\frac{1}{(m-n)(p-n)}\sum_{i=n+1}^{p} s_i^2} ,$$

up to a small bias correction. Under certain statistical assumptions, $c(n)$ can be shown to be the expected root mean squares (rms) of residuals for new devices, if these were to be described by the same n-dimensional model. The Toolbox plots $c(n)$ in the RMS Residuals plot. One can choose n such that the plotted quantity is no greater than a predetermined required uncertainty. If there is no clear choice, the plot can be used to predict the effect of changing the number of parameters. The plot will always indicate that increasing n decreases the size of the residuals. However, for large n, only more noise will be captured and predicted by the model. It should be kept in mind that the $c(n)$ computed in this way are estimates of the residuals that would occur if all test points were used to estimate the model parameters, x. For a reduced set of measurements, the rms residuals can be expected to be somewhat larger.

If a validation set, Y, is available, the Toolbox can give more accurate estimates of the residuals that occur when a reduced set of measurements is used. In this case, the Toolbox displays (in the Valid. Error Stats. display box) the maximum, minimum and rms residual error that occurs when the selected n-column model is used with a subset of k selected test points (see section II.6) to estimate the responses at all m test points. The displayed statistics are computed from all p devices in the validation set. The validation set can be used in this way to explore the tradeoffs between model size, n, and the size k of the reduced measurement set, and the errors that result.

In practice, the recommendations from the three plots are often not consistent or not clear. The Q-Max plot will usually give a maximum, but it may be much too large. This case may correspond to a fairly even decline in the RMS Residuals plot and to the absence of an elbow in the LEV plot. From numerical experiments, it appears that not much can be gained by increasing the number of parameters in such a case. The recommendation then is to choose the smallest value of n that one is still comfortable with [3]. There are a number of other graphical or statistical routines available in the Toolbox, but in an unclear situation these just add to the confusion.

************************************ **HELP Toolbox** **

Modeling data are loaded into the Toolbox via the [Data Sets] menu. There, the measurement data can be split between data designated for the modeling set and data for the validation set. If the data need to be normalized, an m×1 normalization vector must be entered [Data Set/Load Data File/Normalization Vector], whereupon the user is cued to choose which data sets (modeling or validation or both) are to be normalized.

To construct an empirical model matrix from a previously loaded matrix of modeling set data $\bar{A}$, the [Parameters and Test Points] menu is used (section II.3). The [Modeling Set Decomposition] submenu is used to perform the singular value decomposition (SVD), producing the left singular matrix, U , from which the first n significant columns will be chosen. To determine and select the best choice for n based on the SVD of the modeling set, the [Parameter Selection Plots] submenu is used. The three diagnostic plots are found there. The effects of a given choice of n can subsequently be tested via the validation set using the procedures outlined in section II.5, Assessing the Model.

****

4. Construction of Physical, *A Priori* and Mixed Models.

Physical models are usually developed outside of the HELP Toolbox and then imported. For example, a physical model of an electronic circuit could be developed through a sensitivity analysis using a circuit simulation tool such as SPICE. Such a model might be comprised of vectors of partial derivatives of the circuit response errors with respect to the electrical parameters (resistors, capacitors, transconductances, etc.) that define the circuit, evaluated at their nominal values for each candidate test point.

A priori models, also developed outside the HELP Toolbox, are usually based on engineering considerations that go beyond the scope of this discussion. However, for both physical and *a priori* models, it is often true that these will not be complete enough to explain the behavior of the modeled devices with accuracy sufficient for all applications. This is due to approximations in the modeling caused by nonlinear behavior, lack of detailed knowledge, unmodeled interactions among the components, effects of unmodeled parasitics, etc. In addition, it is often the case that the resulting models are not full rank, i.e., two or more of the columns may be linearly dependent. For example, components in cascaded gain stages give rise to identical or collinear sensitivity vectors. When the model A is not full rank, there will be ambiguity in the determination of the corresponding parameter vector, x . While this may not present any problems for subsequent response predictions, it could cause problems if the parameter estimate $\hat{x}$ is used in trimming procedures, i.e., to determine how much adjustment a component requires to bring the circuit response within specifications. The rank of a physical or *a priori* model can be checked using the Toolbox by performing an SVD on it, and looking at the diagnostic plots. The LEV plot will show an abrupt knee at n corresponding to actual rank; if n is less than the number of parameters in the model, then some of the parameter vectors are not independent. Note that if it is important to preserve the physical or engineering meaning of the final model, the SVD should only be used as a diagnostic rank test, and SHOULD NOT be used to create a new model from the physical or *a priori* model. This is because the vectors of the U matrix model that results from the SVD are each weighted combinations of the original vectors.

The accuracy of physical or *a priori* models can be improved while maintaining their descriptive advantages by augmenting them with empirical model data to produce a mixed model. To create an appropriate mixed model, an empirical model, $\boldsymbol{A}_e = [\boldsymbol{a}_{e1}\ \boldsymbol{a}_{e2} \ldots \boldsymbol{a}_{en_1}]$ with n_1 columns is first developed from modeling set data as described in the previous section (II.3). The vectors of the empirical model are each orthogonalized to the n_2 vectors of the physical or *a priori* model, $\boldsymbol{A}_p = [\boldsymbol{a}_{p1}\ \boldsymbol{a}_{p2} \ldots \boldsymbol{a}_{pn_1}]$, using the method of Gram-Schmidt orthogonalization:

$$\boldsymbol{a}_{oi} = \boldsymbol{a}_{ei} - \sum_{j=1}^{n_2} a_{ij}\, \boldsymbol{a}_{pj} \quad \textit{where} \quad a_{ij} = \frac{\boldsymbol{a}_{ei}^T \boldsymbol{a}_{pj}}{\boldsymbol{a}_j^T \boldsymbol{a}_{pj}}, \quad \textit{and} \quad \boldsymbol{A}_o = [\boldsymbol{a}_{o1}\ \boldsymbol{a}_{o2} \ldots \boldsymbol{a}_{on_1}].$$

The new orthogonalized empirical model, $\boldsymbol{A}_o$, describes only the behavior present in the modeling set that is not described by the physical or *a priori* model. An $m \times (n_2 + n_1)$ mixed model $\boldsymbol{A}_m$ is produced by augmenting $\boldsymbol{A}_p$ with $\boldsymbol{A}_o$: $\boldsymbol{A}_m = [\boldsymbol{A}_p\ \ \boldsymbol{A}_o]$.

*************************************** HELP Toolbox **

To create a mixed model in the Toolbox, first create the desired empirical model. Next, load in a previously created physical or *a priori* model using the [Data Sets/Load Data File/Full Model] menu. Orthogonalize the empirical model to the full model (physical or *a priori*) using menu item [Data Sets/Orthogonalize Modeling Set]. Append the desired number of columns of the orthogonalized empirical model to the full model using the menu selection [Params and Test Pts/Modeling Set Decomposition] and then [Params and Test Pts/Select Number of Parameters]. Be sure the "Append" box is selected, as opposed to the "Replace" box.

****

5. Assessing the Model

Suppose a linear model $y = \bar{y} + e = Ax + \varepsilon$ for the measured device response is given, with m test points, n device parameters, an $m \times n$ model matrix $\boldsymbol{A}$, and model error ε. Then the statistical properties of the model error, ε, are of interest. These properties are used to decide if the modeling data are adequately described by a linear model, to give bounds for the accuracy of prediction of new devices, and to detect devices for which the model is not adequate.

We first discuss the case where there is also a validation set, in the form of an $m \times q$ matrix, $\boldsymbol{Y}$. In this case, one fits $\boldsymbol{Y}$ to the n-dimensional model, using ordinary least squares. The residuals from this fit are the columns of a matrix, $\boldsymbol{R}$, that has the same dimensions as $\boldsymbol{Y}$.

From the matrix of residuals, $\boldsymbol{R}$, the Toolbox computes the estimated standard deviation, $\hat{\sigma}$, corrected for degrees of freedom and for prediction variance, as follows:

$$\hat{\sigma} = \sqrt{\frac{1}{q(m-n)} \sum_{i=1}^{q} \sum_{j=1}^{m} \frac{(y_j^i - \hat{y}_j^i)^2}{1 + P_{cj}}},$$

where y_j^i are elements of $\boldsymbol{Y}$, and $\hat{y}_j^i$ are elements of the matrix of predicted values, and P_{cj} contains the prediction variance components [4]. (For this purpose, the number of selected test points should be m -- see box below.)

One should first check to see if $\hat{\sigma}$ is approximately the size expected, i.e., on the order of the measurement standard deviation for the validation set data. If it is substantially larger, it should at least agree with the value $c(n)$ returned by the RMS Residuals diagnostic plot that was computed when the model was created. If $\hat{\sigma}$ is much larger than $c(n)$, it can be concluded that the validation set contains information that was not present in the modeling set. Plots of the residual vectors that comprise $\boldsymbol{R}$ may reveal certain devices in the validation set that do not adequately conform to the model. If $\hat{\sigma}$ agrees reasonably well with $c(n)$ but is substantially larger than the measurement standard deviation, this is an indication that the model could be improved by choosing a larger value for n. The decision to increase n will probably depend on whether or not the residuals are small enough to be acceptable. To decide this, a more accurate estimate of the expected performance of the model should be made after the test-point selection process (see section II.6), when the effects of a reduced set of test points are included. For most cases, the final decision about the model dimensions and composition should be based on the uncertainty bounds that are produced. These can be computed for the validation set and are discussed in section II.7.

** HELP Toolbox **************************************

For most purposes, the rms residuals value returned in the Valid. Error Stats. box can be used as a good approximation for $\hat{\sigma}$. The two will differ slightly because the rms residuals value has not been corrected for degrees of freedom and prediction variance. Since both quantities, $\hat{\sigma}$ and rms residuals, are computed from predictions $\hat{y}$, they are dependent on the number of test points that have been chosen. To compare these values with the expected value for $\hat{\sigma}$, and with the value $c(n)$ given in the rms residuals diagnostic plot, ALL test points should be selected. To get a better estimate of the actual errors that will result from the model in use on test devices, only the REDUCED set of test points should be used.

****

If there is no validation set available, one can use $c(n)$ to get a rough estimate of the size of the residuals. These results will only be approximate though since they do not take the reduced test points into account, and they are subject to the other approximations noted above.

When the model from the modeling data is used to predict the behavior of a new device under test, the uncertainty associated with this prediction comes from several different sources. These sources are the estimation error for the model (due to the fact that the modeling set data themselves contain measurement noise), a truncation error for the model (due to the fact that the true device behavior in all likelihood is determined by more than just n parameters), another truncation error for the device under test (since it is not expected to really fit any model with only n parameters), and finally the measurement error for the device under test. Only the last error varies if a device under test is measured repeatedly. However, it can be shown that the truncation errors and the estimation errors can also be treated as if these were random quantities. What makes these behave differently than just measurement errors is the fact that they contain multiplicative effects on the overall prediction error. The truncation error that occurs during the model construction has a stronger effect on the overall prediction error if the device itself differs substantially from its nominal behavior. Essentially, a device that is very close to its nominal behavior will be predicted very accurately by almost any linear model. This effect can be analyzed mathematically and is taken into account by the Toolbox algorithms.

6. Selecting a Reduced Test-Point Set

Suppose a linear model, $y = \bar{y} + \varepsilon = Ax + \varepsilon$, for the measured device response is given; with m test points, n device parameters, an $m \times n$ model matrix A, and model error ε. We assume that the model matrix has been determined from a modeling set as outlined above; thus the matrix A now has orthonormal columns. We want to choose a reduced test-point set J, i.e. a subset of the set of all test points (1,2, ..., m) such that the device behavior can be predicted reliably at all m test points from measurements at the test points in J.

Once J has been chosen, an estimate, $\hat{x}$, of the device parameters is found from the measurements at the test points in J with the method of least squares, and the predicted device behavior, $\hat{y}$, is computed. The differences between the estimate, $\hat{y}$, and the measured behavior, y, then depend on the error vector, ε, and on the choice of test-point set, J. Recall that ε incorporates both measurement errors and errors that are due to the use of a linear model. One would like to select J automatically and efficiently such that these differences are small with high reliability. This is a problem from experimental design [5].

In order to discuss the test-point selection method, let us assume that the elements of ε have homogeneous variances, i.e., that its standard deviations are the same at all test points and that these errors are uncorrelated. Both assumptions are questionable, since the measurement uncertainties are usually not exactly the same at all test points even after changing units, and since the entries of the vector ε may include remainder terms that come from fitting the measured response to the model. Such remainder terms will always be correlated. However, the effects of violating these assumptions can be analyzed and controlled.

Suppose a subset J with $k \geq n$ elements is desired. The Toolbox can select these test points automatically, using a method that employs two phases. In the first phase, a minimal subset of n test points is determined by applying the well-known QR factorization with column pivoting [2] to the transpose of the model matrix, A. The first phase can be summarized as follows: The first test point that is selected corresponds to the row of A that has the largest norm. All other rows are orthogonalized with respect to this row (a suitable multiple of this row is subtracted from all other rows such that each result is perpendicular to the row that was chosen first), and their norms are recomputed. The test point corresponding to the largest remaining norm is chosen next, and the process is repeated until n test points have been selected. At this point, all remaining rows have norms equal to zero, and the first phase is over. In the second phase, $k-n$ additional test points are selected. To explain the second phase, let us assume that the "minimal" reduced test-point set that was the result of the first phase is to be used for predicting the behavior of the entire device. This prediction will amplify the measurement errors at the selected test points, and the amplification factors (prediction variances) can be computed for each test point that has not been selected. (At the reduced test-point set, these amplification factors are equal to 1 at this stage, since the predictions there agree with the measurements.) Of those test points that have not yet been selected, the one with the largest prediction variance is included in the reduced test-point set. The prediction variances then are recomputed and the procedure is repeated, adding one test point at a time until the desired model size has been reached. We now delete all rows of A except those that correspond to the reduced test-point set J. The result is the reduced model matrix $\widetilde{A}$ with k rows and n columns.

There are several possible optimality criteria for the choice of the reduced test-point set J. These all can be interpreted as attempts to make some kind of confidence set for the prediction $\hat{y}$ as small as possible (minimizing its volume, its diameter, etc.). It turns out that the first phase of the selection method that is given above tries to minimize the volume of such a confidence set. The second phase tries to minimize the diameter of another type of confidence set. Thus, the two phases do not pursue the same optimality criterion. However, it is known that in certain limiting cases the two criteria are equivalent. There is also strong numerical evidence that the two stages are compatible. Moreover, the test-point sets that are found with this method usually are also very good selections for all sorts of other optimality criteria.

Theoretically, an optimal reduced test-point set for any optimality criterion could be found by searching through all possible subsets, computing the relevant performance criterion, and selecting the best. However, the enormous size of the set of all possible test-point sets makes this approach completely unfeasible. For example, if a device has $m = 200$ test points and the reduced set is to have $k = 30$ test points, then the number of candidate subsets is equal to the binomial coefficient

$$\binom{200}{30} \approx 4 \times 10^{35}.$$

It is clearly impossible to examine them all. Thus, approximate methods must be used. Even iterative methods that have been developed specifically for experimental design purposes are usually too expensive for problems that come from linear modeling. Then "one-pass" methods that build up the reduced test-point set, J, in a single sweep are the only alternatives remaining. The method that is used by the Toolbox is of this type. Numerical evidence shows that the results of the method are usually not optimal, but that the reduced test-point set that is chosen here is better than 99.98 percent or so of all possible choices and that it cannot be improved by much.

7. Prediction and Decision for New Devices

When the Toolbox has constructed a model and determined its properties, it is ready to use measurement data from a new device, taken at the reduced test-point set, to predict its measured behavior at all test points. The parameter vector, x, is first estimated from the reduced measurement data $\tilde{y}$ using the least squares method

$$\hat{x} = \left(\widetilde{A}^T \widetilde{A}\right)^{-1} \widetilde{A}^T \tilde{y}.$$

From the estimate, $\hat{x}$, the predicted behavior, $\hat{y}$, at all test points is given by

$$\hat{y} = A\hat{x}.$$

** HELP Toolbox **

Parameter vector, $\hat{x}$, and prediction, $\hat{y}$, are automatically computed for vectors in the validation set when the [Assess Model/Validate Model] menu item is selected. When measurement data on a device under test is available at the selected test points and has been loaded into the Toolbox, then $\hat{x}$ and $\hat{y}$ are automatically computed when the [Assess Model/Predict Calibration] menu item is selected. A plot of $\hat{y}$ can be obtained from the [Plot/Validation Analysis/Response Predictions] or [Plot/DUT Analysis/Response Predictions] menu items.

****

Along with this prediction, some information about the measurement uncertainty at each test point is also needed. The predicted behavior, together with an uncertainty estimate for the prediction, is used to decide whether the behavior of the device is within specified tolerance bounds.

It may happen that a device under test cannot be well described by a given linear model, since its error patterns include features that were not identified from the modeling set. This may occur, e.g., if a production method is changed, if components in a device are replaced, or simply if measurements at the device under test are performed with less precision. The result will show up in a larger than expected error at any test point. Such errors are called non-model errors. Referring back to section II.5, non-model errors are indicated when the residuals of the new device are significantly greater than $\hat{\sigma}$, after correcting for degrees of freedom. The Toolbox has a procedure that incorporates the effects of non-model error into the calculated uncertainty bounds, based on the size of the residuals at the measured test points only. Consequently, the bounds are designed to contain the true measured response with specified confidence, even when the model fits rather poorly.

It is assumed that the upper and lower tolerance bounds for an acceptable device are in the form of two vectors of length *m*. A device is acceptable if its measured behavior at each test point is within the corresponding bounds. The Toolbox computes prediction intervals for the measured values. (Prediction intervals are computed statistical bounds around the predicted values that are asserted to bound the measured behavior with a given confidence. Note that since measured behavior is true behavior plus measurement noise, the prediction intervals bound true behavior with even greater confidence.) These intervals are calculated from the residuals at the reduced test points as

$$\hat{y} \pm r_s\sqrt{1+P_c}\,(t_{1-\alpha/2}),$$

where r_s is the standard deviation of the residuals at the measured test points given by

$$r_s = \frac{1}{k-n}\sqrt{\sum_{j=1}^{k}(\tilde{y}_j - \hat{y}'_j)^2},$$

such that $\hat{y}'$ is the estimate of the device response at the reduced test points, where P_c is the $m \times 1$ prediction variance coefficient vector given by

$$\mathbf{diag}\left\{A\left(\tilde{A}^T\tilde{A}\right)^{-1}A^T\right\},$$

and $t_{1-\alpha/2}$ is the 1-(α/2) quantile of the *t* distribution with $k-n$ degrees of freedom. The Toolbox uses a coverage probability $1\text{-}(\alpha/2) = 0.9545$. Note that the square root operation is performed element-by-element in the above prediction interval calculation.

It is assumed here that these residuals at the measured test points can be treated as normally distributed random variables and include a "representative" sampling of the non-model error that exists, as well as the contributions from measurement noise, model truncation, etc. Tests of these intervals on several examples of real measurement data indicate that they tend to be somewhat conservative, i.e., 95% intervals typically bound 96% to 97% of the measurements.

These individual intervals can therefore be treated as type A measurement uncertainties [6] that would occur if the device under test was tested exhaustively. If one decides to reject a device as soon as its prediction interval extends beyond the tolerance bound at any test point, no more than a fraction of about $\alpha/2 = 0.0227$ of all "bad" devices will be accepted, that is, the type-I error probability is about $\alpha/2$.

The rationale behind the decision procedure is equivalent to making the hypothesis that a device is "bad", the null hypothesis [5]. This hypothesis means that the data have to prove that a device is good, not that it is bad. If the device under test were bad, its measured behavior would exceed the specified tolerance bounds at no less than one test point. The probability that its prediction interval at this test point is within both tolerance bounds is at most $\alpha/2$ and is much smaller if the behavior exceeds the tolerance bound substantially. Thus, even if a bad device exceeds the tolerance bounds at only one test point, it will escape detection with a probability of no more than $\alpha/2$.

A set of individual prediction intervals for a particular device does not contain all measured values for this device with probability $1-\alpha$, nor does a fraction of $1-\alpha$ of these intervals contain the measured behavior for this device. Intervals that come with a guarantee that they predict the measured behavior for all test points for all but a small fraction of devices under test are called simultaneous prediction intervals and must be wider than individual intervals. The Toolbox computes such intervals with the same specified coverage probability $1-\alpha$ for the new device. These bounds are just the predicted values plus or minus a suitable larger *t*-quantile times the standard deviations that are also computed for individual prediction intervals. These values are typically about twice as wide.

In simulations, it is observed that a larger portion of unacceptable devices is identified correctly than the advertised fraction $1-\alpha/2$. This observation is true regardless of the proportion of bad devices in the set that is under test. The reason is that many bad devices will in fact be identified almost certainly, since their behavior exceeds the bounds somewhere substantially. The decision procedure is designed to work reliably even for "marginal" devices. Since the prediction intervals are adjusted for each device under test by means of the standard deviation of the residuals, "bad" devices are still reliably identified if non-model errors appear, and the coverage probability of individual prediction intervals does not decrease. However, simultaneous prediction intervals are observed to be less reliable in the presence of non-model errors or increased measurement uncertainties. That is, their coverage probability may be less than the advertised value $1-\alpha$ if there are non-model errors or if the measurement uncertainty for devices under test increases.

Along with a large fraction of bad devices, a certain fraction of acceptable devices will also be rejected. In fact, a device that is just barely within the tolerance bounds will be rejected with probability close to $1-\alpha$, since it is almost indistinguishable from a "bad" device. The occurrence of such type-II errors is in the nature of statistical decision procedures. If the measurement uncertainty is increased for devices under test or if non-model errors appear, the probability of type-II errors will also increase. The result is a larger overall proportion of devices that are rejected, even if their true quality does not change. An indicator for this phenomenon is a value for $r_s/\hat{\sigma}$ that is substantially and consistently larger than unity, approaching and exceeding 2. This result may indicate that the model is truly no longer adequate and suggests that it be tested against a new validation set or that a new model be built.

III. Software Description

1. Introduction

The purpose of the High-dimensional Empirical Linear Prediction (HELP) Toolbox is to provide test engineers and technicians with a tool to help select optimal testing strategies for complex electronic devices and instruments in a very user-friendly manner. The software has been developed using the MATLAB® programming environment. It is designed for use without detailed knowledge of the theory and mathematics governing empirical linear prediction. The menus are set up for use in a general sequential, top-down, left-to-right fashion. User prompts and data-entry windows aid the user in following the correct sequence in developing and analyzing models.

The software has evolved through several generations of programs and subroutines to its present form. Originally, the software consisted of numerous subroutines requiring a great deal of mathematical knowledge to proceed sequentially through formulation of model and application and analysis of the model using empirical data. The present HELP Toolbox consists of a main program controlling the main HELP window and menu system with numerous subroutines that are called by the main program. The subroutines control graphical user interfaces (GUIs) that help the user move through the menus and includes additional subroutines that perform some of the mathematical algorithms. Most of the mathematics is contained within the main program; however, a few of the previously developed routines were maintained separately for ease of development.

2. Toolbox Architecture

The design of the HELP Toolbox incorporates what is known as the "switchyard" technique for developing GUIs. The switchyard technique is a programming device whereby if/elseif conditional structures are used to create separate sections of code that are accessed by calling the main program with a particular flag set. The flag triggers the running of the desired switchyard section. Fig. 3.1 shows the general switchyard structure of the program.

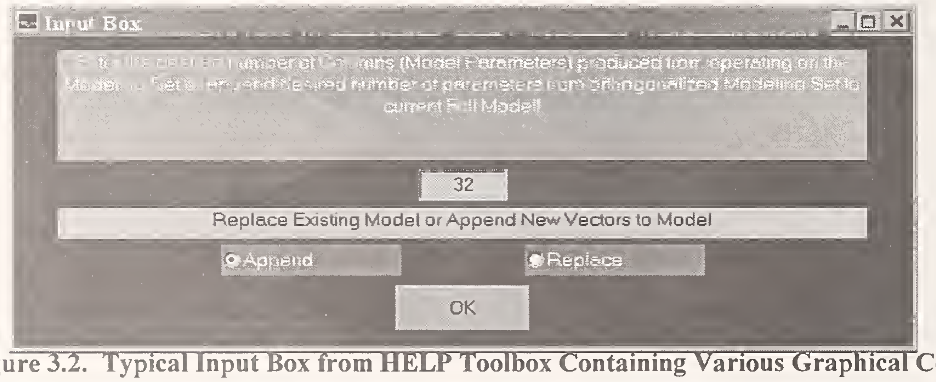
```
function help22(action);           Define a function called help22 with one input argument.
if nargin<1,
   action = 'start';               If there are no input arguments, set action = 'start'.
end
global (list of global variables)  Assign all the global variables
if strcmp(action,'start'),         This is the first switchyard section of code labeled 'start'.
          Initialize HELP Window and menu system.
          .
          .
elseif strcmp(action,'load_mbv'),  This is the second switchyard section labeled 'load_mbv'.
          Perform some operation.
          .
          .
elseif ... Numerous additional switchyard sections of code...
          .
          .
elseif strcmp(action,'help'),
          Bring up Help Window.
elseif strcmp(action,'done'),
          Close HELP Window and clear all global variables.
end
```

Figure 3.1. Switchyard Structure of the HELP Toolbox Main Program

When the HELP Toolbox main program is run from the MATLAB command window with no input arguments (no flags set), the program initializes the HELP Toolbox window with all its menus and graphical components and declares all the global variables used by the Toolbox. Selecting one of the Toolbox menus will call the main program again, this time, with one of the switchyard flags set. The appropriate switchyard section of code will run, performing the task requested. All global variables operated upon by the code, are saved and accessible to each subsequent call of the program.

The switchyard technique is the recommended method of developing multiple MATLAB routines that are to be controlled using a common GUI†. One characteristic of this programming method is that each switchyard section is not directly able to communicate with other sections without the use of global variables because each section is called with a separate running of the program. Manipulation of handle graphics allows the user to pass input parameters into a switchyard section but variables cannot be directly passed between sections of code. (Handle graphics is an object-oriented graphics system that provides the individual components necessary to create and manipulate computer graphics.) In order for variable values to pass between switchyard sections, the variables must be defined as global within all programs in which the programmer desires to maintain common variable values (see Local and Global Variables in the MATLAB Users Manual). Assigning numerous global variables has advantages and disadvantages. A disadvantage is that if a user assigns one of the HELP Toolbox global variables to be global within the MATLAB command window and then inadvertently changes the value of a global variable within the MATLAB command window, subsequent Toolbox commands using that variable will be affected (adversely). On the other hand, assigning global variables gives the user the power to access many parameters within the MATLAB command window, providing broad analysis capabilities to the HELP Toolbox. Taking advantage of global variables within the MATLAB command window requires the user to have some knowledge of MATLAB.

As mentioned above, the HELP Toolbox software consists of a main program and several types of subroutines that are called by the main program. There are subroutines that create data input windows to pass data into the Toolbox. There are subroutines that create error and information windows to help guide the user through the HELP procedures. There are also subroutines that perform the mathematical computations required by the procedures. All the subroutines are invisible to the user. Only their effects are visible. That is, the user sees the graphical user interface windows that appear and the user sees the model-related parameters that appear on the front panel of the main HELP window.



Enter the desired number of Columns (Model Parameters) produced from operating on the Modeling Set or append desired number of parameters from orthogonalized Modeling Set to current Full Model!

32

Replace Existing Model or Append New Vectors to Model

○ Append ● Replace

Figure 3.2. Typical Input Box from HELP Toolbox Containing Various Graphical Components

The subroutines that use graphical windows to input data or display error messages or information are programs that manipulate handle graphics. They create graphical objects with assigned properties such as pushbuttons, radio buttons static and dynamic text objects, and sliders. Fig. 3.2 shows an input box from the HELP Toolbox that contains static and dynamic text boxes, radio buttons and a pushbutton.

3. Glossary of Variables

3.1. Global Variables Used in the HELP Toolbox

ADAT	Data Set (any data loaded into HELP Toolbox later to be assigned to specific variable)
AFULL	Full Model (m×n)
ARED	Reduced Model (k×n)

† See the workbook for the 1995 MATLAB Conference Tutorial entitled "Building a Graphical User Interface with MATLAB."

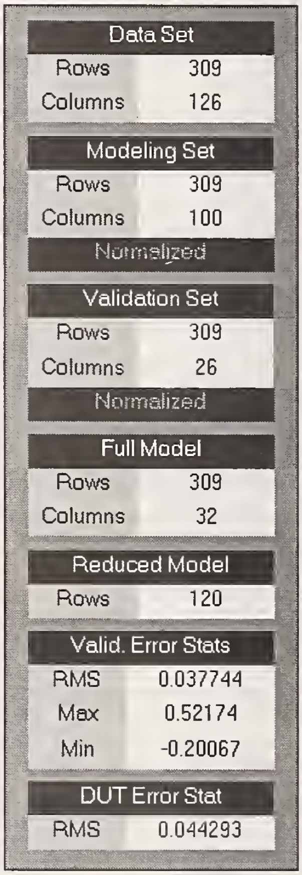
ATRAIN	Modeling Set (m×p) used to develop a device model
AVAL	Validation Set (m×q) used to test the device model
BINT1MEA	Upper bound for Simultaneous Measurement Prediction Interval (m×r)
BINT1VAL	Upper bound for Simultaneous Validation Prediction Interval (m×q)
BINT2MEA	Lower bound for Simultaneous Measurement Prediction Interval (m×r)
BINT2VAL	Lower bound for Simultaneous Validation Prediction Interval (m×q)
IINT1MEA	Upper bound for Individual Measurement Prediction Intervals (m×r)
IINT1VAL	Upper bound for Individual Validation Prediction Intervals (m×q)
IINT2MEA	Lower bound for Individual Measurement Prediction Intervals (m×r)
IINT2VAL	Lower bound for Individual Validation Prediction Intervals (m×q)
INDV	Index showing which columns of Data Set are to be assigned to the Validation Set
INTB	Absolute bound for Simultaneous Validation Prediction Interval (m×q)
INTBM	Absolute bound for Simultaneous Measurement Prediction Interval (m×r)
INTI	Absolute bound for Individual Validation Prediction Intervals (m×q)
INTIM	Absolute bound for Individual Measurement Prediction Intervals (m×r)
NMFACTORM	Nonmodel Factor for the Reduced Measurement Data (r×1)
NMFACTORV	Nonmodel Factor for the Validation (q×1)
NORM_VEC	Normalization Vector (m×1) used to normalize the Modeling Set and Validation Set
PREDVARVEC	Prediction Variance Coefficients used to produce prediction intervals
PR1	Pivot Vector (k×1) (lists test points selected for the Reduced Model)
RESMEAS	Residual Errors (k×r) (columns correspond to the Reduced Measurement Data)
RESVAL	Residual Errors (m×q) (columns correspond to the Validation Set)
SELECT_MOD_VECS	Index of Modeling Set vectors for plotting (r×1)
SELECT_VAL_EXTRACT	Indices of vectors from the validation set used to create reduced measurement data
SIGMA_MM	Standard deviation of the measurement noise for the Reduced Measurement Data (r×1)
SIGMA_MV	Standard deviation of the measurement noise for the Validation Set (q×1)
STDHAT	Nonmodel estimate of measurement noise (scalar)
S_SVD	(Absolute values of) Singular values of the Modeling Set (n×1)
TOTAL_MEA_OUT_B	Number of measurements from validation set that lie outside simultaneous prediction intervals
TOTAL_MEA_OUT_I	Number of measurements from validation set that lie outside individual prediction intervals
U_SVD	Left Singular Matrix (m×n) computed from singular value decomposition (SVD) of Modeling Set. Columns are linear combination of columns from Modeling Set. Full Model is subset of this matrix.
XHATMEAS	Parameter Coefficient Vectors for Measurement Data (n×r)
XHATVAL	Parameter Coefficient Vectors for Validation Set (n×q)
YHATMEAS	Response Predictions (m×r) (columns correspond to Reduced Measurement Data)
YHATVAL	Response Predictions (m×q) (columns correspond to Validation Set)
YMEAS	Reduced Measurement Data (k×r) (measurements at only the reduced test points)

3.2. Important Variable Dimensions

k	Number of test points selected for Reduced Model
m	Number of test points in Full Model (candidate test points)
n	Number of parameters selected for the model
p	Number of Modeling Set vectors
q	Number of Validation Set vectors
r	Number of Reduced Measurement vectors

3.3. HELP Toolbox Front Panel Parameters

Figure 3.3 shows the HELP Toolbox front panel displaying parameters from a particular modeling situation. The **Data Set** parameters give the row and column sizes for a matrix of data loaded into the Toolbox (using the menu selection [Data Sets/Load Data File/Modeling Set and Validation Set]) to be divided into a modeling set and a validation set. The user is prompted to select which columns of the data set is to be separated from the modeling set data and placed into a validation set.



Data Set
Rows 309
Columns 126

Modeling Set
Rows 309
Columns 100
Normalized

Validation Set
Rows 309
Columns 26
Normalized

Full Model
Rows 309
Columns 32

Reduced Model
Rows 120

Valid. Error Stats
RMS 0.037744
Max 0.52174
Min -0.20067

DUT Error Stat
RMS 0.044293

Figure 3.3. HELP Toolbox Parameter Display Panel

The **Modeling Set** parameters show the row and column sizes for data placed into the modeling set when data are loaded for modeling set and validation set or when data are loaded for modeling set only.

The **Validation Set** parameters show the row and column sizes for data placed into the validation set when data are loaded for modeling set and validation set or when data are loaded for validation set only.

The **Full Model** parameters show the row and column sizes for the full model. A full model can be created by algebraically manipulating the modeling set using three sequential menu selections: (1) [Params and Test Pts/Modeling Set Decomposition], (2) [Params and Test Pts/Select Number of Parameters], and (3) [Params and Test Pts/Test Point Selection/…]. Also, data can be loaded into the Toolbox to be used as a full model using the menu selection [Data Sets/Load Data File/Full Model].

The Rows-parameter under the **Reduced Model** label shows the number of test points selected from the full model to be used for the reduced model. The test points can be selected optimally using the menu selection [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. They can be selected manually using the menus [Params and Test Pts/Test Point Selection/Test Point Assignment/…]. Or they can be selected using a combination of optimally and manually selected points (if, for instance, the user wants to select the points optimally but force several additional points to be selected) using the menu selection [Params and Test Pts/Test Point Selection/Combined Optimization and Assignment]. The column size for the reduced model is always the same as the column size for the full model.

The validation error statistics are produced by applying the validation set to the model using the menu selection [Assess Model/Validate Model]. The **Valid. Error Stats** parameters show the root mean square value, the maximum value, and the minimum value of the residual errors. The residual errors equal the measurement minus the prediction at each of the candidate test points.

The **DUT Error Stat** RMS-parameter shows the root mean square value of the residual errors produced from applying measurement data from a device under test (DUT) to the model. The residual errors for the DUT equal the measurement minus the prediction at only the reduced test points. DUT measurement data can be loaded into the Toolbox using the menu item [Data Sets/Load Data File/Reduced Measurement Data] or it can be artificially produced by extracting the selected test points from one or more columns of the validation data using the menu item [Data Sets/Load Data File/Extract Reduced Meas. from Validation].

IV. Description of Toolbox Menus

Introduction to the Menus and Variables

The HELP Toolbox is a graphical-user-interface (GUI)-based software package for use with MATLAB® that facilitates the modeling and analysis required for testing and calibration of complex systems. High-dimensional empirical linear prediction requires a large number of device data sets for a common device or system, with each data set covering the identical measurement points, referred to as the candidate set of test points. (Note: Physical and *a priori* models may be used in place of empirical models but it has been experimentally found that empirical models are more robust.) The large number of data sets is termed the modeling set or training set. It is used to construct a linear model, referred to as the system model. The modeling set must be collected externally and loaded into the Toolbox. The modeling set is manipulated and operated upon algebraically within the Toolbox to produce the device or system model. The modeling allows for characterization of the device or system at all points within the candidate set of test points using measurements taken only at a subset of those candidate test points. Response predictions can then be made at the full set of candidate test points based upon measurements at the subset of test points. Various statistical analyses may be performed within the Toolbox to determine accuracy of the model, confidence bounds, etc.

The HELP Toolbox menu operations function in a general sequential manner from left to right, and top down, across the GUI window. The Toolbox is intended to be driven via the mouse or hotkeys with subsequent user-prompted keyboard entries for appropriate input. There are seven main menu headings, each containing sub-menus. The menus are, from left to right, Data Sets, Params and Test Pts, Assess Model, Quality Control, Plot, Help, and Exit. Figure 4.1 displays the Toolbox window with its menus and display panel (with a blank application screen). The panel displays various parameters relating to the user's current session of modeling and analysis. Some of the parameters are the sizes of the data set loaded into or derived within the Toolbox, the modeling set, the validation set, the full model, and the reduced model. The Toolbox panel also shows some error statistics obtained from applying the model to the validation set and the device under test (DUT).

The software for the HELP Toolbox consists of a main program and various subroutines that are accessed from the main program. Selection of a particular menu heading corresponds to execution of a section of code from the main HELP Toolbox program. Within the section of code executed, various variables are created and operated upon. The variables within the Toolbox code very often are mnemonic to the parameters they represent, unlike the variables within the mathematical overview presented in section II. Table 3.1 presents the mapping from the variable names in the mathematical overview to the variable names in the software.

1. Data Sets.

Data sets are used for many purposes within the HELP Toolbox. Typically, a set of empirical data in matrix form is loaded into the Toolbox as a modeling set, sometimes called a training set, to be algebraically transformed into the system model. Another matrix may be loaded into the Toolbox to serve as validation data (to test the model) once the model has been developed. If the magnitudes of the data are not close in size, then yet another data set, in the form of a vector, possibly consisting of tolerance data, may be loaded into the Toolbox to normalize the modeling and validation sets. (Normalization can be critical for producing prediction intervals, i.e., predicted uncertainties, that meet required specifications.) Additionally, variables created through operation of the Toolbox can be easily saved to the computer disk via the Data Sets menu. The Toolbox can load and save files in both ASCII text format and MATLAB binary format. MATLAB allows the loading of ASCII files with comma-, space-, or tab-delimited data. Files that are saved in the ASCII format are space-delimited. The MATLAB binary format is much more efficient and so the user may wish to use the binary format for larger data files. By convention, MATLAB binary files should use the '.mat' extension.

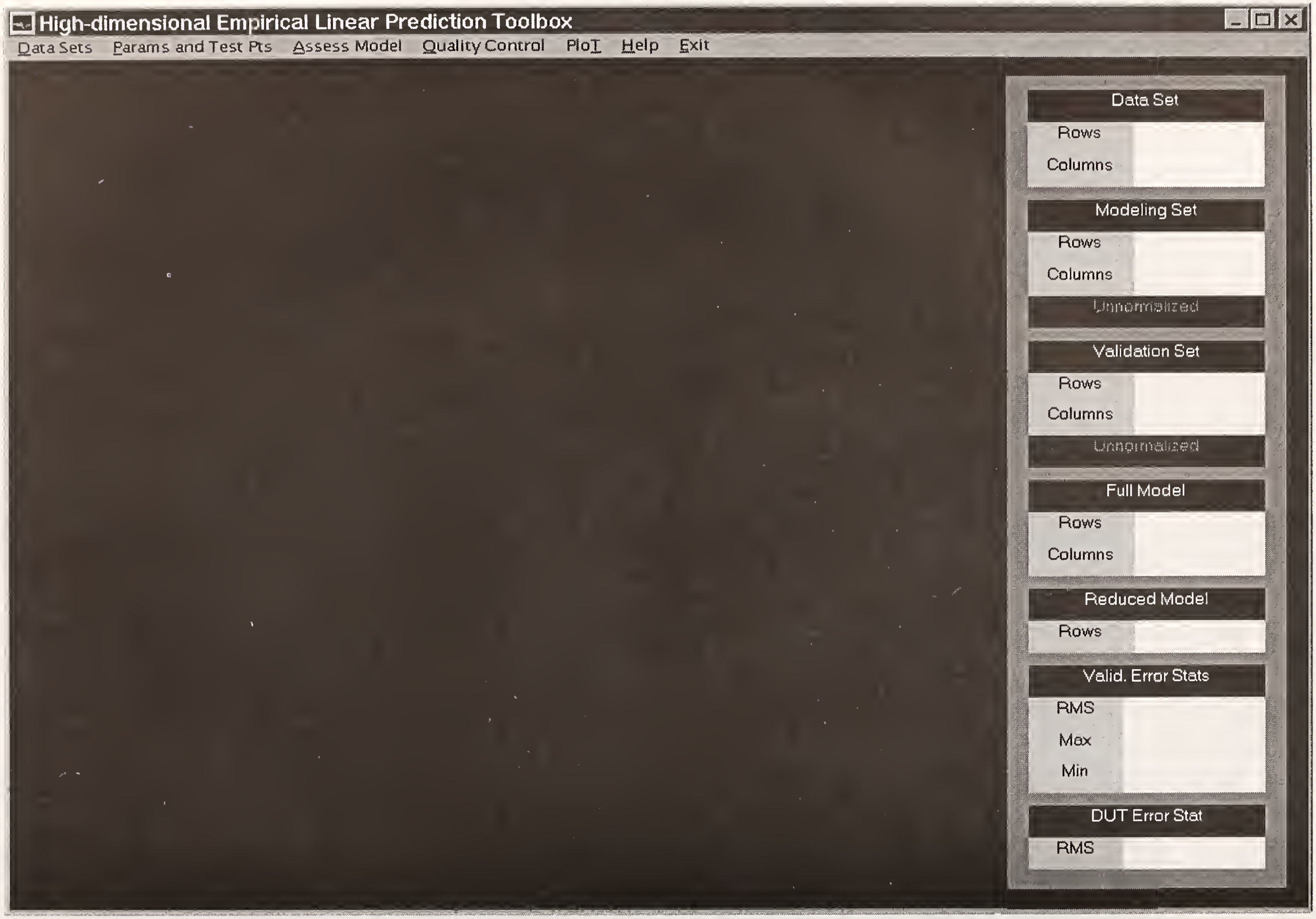


Figure 4.1. HELP Toolbox Window Showing Menus and Display Panel.

Table 4.1. Correspondence between Mathematical Overview Variables and Toolbox Variables

Math	Toolbox	Description
A	AFULL	m×n linear system model (called the full model)
U_1	AFULL	first n columns (m×n) of U (A equals U_1 during empirical modeling)
$\tilde{A}$	ARED	k×n reduced model matrix (called the reduced model)
$\bar{A}$	ATRAIN	m×p matrix of training data (called the modeling set)
Y	AVAL	m×q matrix of validation data (called the validation set)
$c(n)$	c	root mean squares of residual errors (used to determine n)
ε		m×1 device measurement errors (measured - true)
k	k	number of reduced test points selected for the model, i.e., m>k; generally selected such that k is 2 to 5 times n
m	m	number of measurement points (referred to as test points)
n	n	number of parameters in the model
p	p	number of training data sets
J	PR1	k×1 vector of reduced test points (called the pivot vector)
q	q	number of validation data sets
$Q(n)$	q	Q-Max value (used to determine n)
r	r	number of devices under test (DUTs)
	RESMEAS	k×r matrix of DUT errors (measured - predicted)
	RESVAL	m×q matrix of device errors (measured - predicted)
R		m×n remainder term
S	s	p×p diagonal matrix of singular values
	S_SVD	p×1 column vector of singular values
$\hat{\sigma}$	STDHAT	m×1 estimate of the rms of the residual errors in the validation set
U	U_SVD	m×p left singular matrix of orthogonal columns
V^T	v	p×p right singular matrix
	XHATMEAS	n×r matrix of parameter vector estimates for the devices under test
	XHATVAL	n×q matrix of parameter vector estimates for the validation set
x		n×1 true parameter vector (parameters are specific to a device)
$\hat{x}$		n×1 estimate of the parameter vector
	YHATMEAS	k×r prediction of device responses for the devices under test
	YHATVAL	m×q prediction of device responses for the validation set
$\tilde{y}$	YMEAS	k×r matrix of (reduced) measured responses for the DUTs
y		m×1 measured device response
$\bar{y}$		m×1 true device response
$\hat{y}$		m×1 estimate of the device response
$\hat{y}'$		k×1 estimate of the device response at reduced test points

1.1. Load Data File.

This menu item allows data files in either MATLAB binary or ASCII text format to be loaded into the Toolbox environment. The file must be assigned the status of one of the variables listed as options in the sub-menu (listed numerically below as 1.1.1. through 1.1.8.). The front panel on the Toolbox window displays the sizes of the data sets loaded into the Toolbox. Note that if data are loaded into the Toolbox for the modeling set and validation set and some of the data are set aside for validation, then the displayed sizes of the modeling and validation sets should add column-wise to equal the size of the data set displayed in the front panel. Note also that the full and reduced models are used in combination to 'form' the system model used to characterize the device or instrument under consideration. The rows of the reduced model correspond to the reduced set of test points which, when measured for a particular device, can be used with the system model to predict the response of the device at all of the candidate test points contained in the full model. The sizes of all the data sets are interrelated so that the sizes of the data sets not specifically listed within the front panel of the Toolbox window may be known by the sizes of the listed variables. In parentheses next to each menu heading below is listed the corresponding variable size.

For each of the sub-menu items below, a window will pop up prompting the user to state whether the file to be loaded is a MATLAB binary file or an ASCII text file. (The user may cancel at this point in order to make this determination.) Next, another window pops up asking for the name of the file to be loaded. Note that a MATLAB binary file must be of the form *filename.mat*, while an ASCII text file may have any extension or no extension.

1.1.1. Modeling Set (m×p) and Validation Set (m×q).

This menu item allows the data loaded from the selected file to be split by columns into two parts, a modeling set, from which to build a model, and a validation set, to test the model. After selecting the file to be loaded, the user is asked if some of the data are to be set aside for validation. If so, the user is asked whether the column indices are to be selected via a file (containing a vector of indices) or manually by typing the desired indices into an input box. The data file may contain 2-D or 3-D matrices.

To construct an empirical model from the modeling set, we assume that the modeling set that has been loaded is an m×p matrix. Its columns are vectors of exhaustive measurements, one for each of the p devices in the modeling set. The modeling set loaded into the Toolbox will be used to construct the system model. See section 2. Parameters and Test Points.

1.1.2. Modeling Set Only (m×p).

This menu selection designates that all the data loaded into the Toolbox from the selected file will be used for the modeling set. The modeling set can be manipulated algebraically to produce full and reduced models. The data file may contain 2-D or 3-D matrices (as produced within MATLAB).

1.1.3. Validation Set Only (m×q).

This menu selection designates that all the data from the selected file will be used for model validation once full and reduced models are obtained. The data file may contain 2-D or 3-D matrices (as produced within MATLAB).

1.1.4. Full Model (m×n).

This menu selection designates the data from the selected file to be assigned to the full model. A reduced set of test points must subsequently be selected to form a reduced model.

1.1.5. Reduced Model (k×n).

This menu selection designates the data from the loaded file to be assigned to the Reduced Model. The user must also load a Full Model into the Toolbox in order to make predictions on device behavior using the Toolbox.

1.1.6. Reduced Measurement Data (k×r).

Once Full and Reduced Models have been developed within or loaded into the Toolbox, measurements corresponding to the reduced test points from the Reduced Model can be measured and loaded into the Toolbox and used to predict the responses at the entire set of candidate test points. These data are referred to as Reduced Measurement Data.

1.1.7. Normalization Vector (m×1).

In a typical modeling case, it is helpful to normalize the Modeling Set and Validation Set to the tolerances required for the particular needs of the user. This step is especially true if different test points have different measurement uncertainties. Assignment of data using this menu item allows a Normalization Vector to be loaded into the

Toolbox for such a purpose. The user has the option of normalizing both the Modeling Set and the Validation Set or either set separately (normally both sets should be normalized). Descriptors directly below the Modeling Set size and Validation Set size on the front panel of the Toolbox indicate whether data has been normalized.

1.1.8. Extract Reduced Measurement from the Validation Set (r×n).

This option allows the user to artificially fabricate Reduced Measurement Data from the Validation Set. The user is prompted to select which vectors from the Validation Set are to be used to fabricate the Reduced Measurement Data. The user must have already loaded a Validation Set. Use of this option will write over any previously loaded Reduced Measurement Data.

1.2. Clear Current Data.

This menu item clears the Toolbox of all current data previously loaded into the Toolbox or previously computed within the Toolbox. It clears all global variables as well as the plot in the Toolbox Window and the parameter values displayed on the front panel.

Note that many variables are created as global variables within the HELP Toolbox program so that they may be passed between subroutines as the user progresses through the HELP modeling and analysis. The user should be aware when creating global variables within the MATLAB Command Window that the names below are used by the Toolbox and if changed could produce false modeling results.

Global Variables within the HELP Toolbox

ADAT	AFULL	ARED	ATRAIN
AVAL	BINT1MEA	BINT1VAL	BINT2MEA
BINT2VAL	IINT1MEA	IINT1VAL	IINT2MEA
IINT2VAL	INDV	INTB	INTBM
INTI	INTIM	NMFACTORM	NMFACTORV
NORM_VEC	PREDVARVEC	PR1	RESMEAS
RESVAL	SELECT_MOD_VECS	SELECT_VAL_EXTRACT	SIGMA_MM
SIGMA_MV	STDHAT	S_SVD	TOTAL_MEA_OUT_B
TOTAL_MEAS_OUT_I	U_SVD	XHATMEAS	XHATVAL
YHATMEAS	YHATVAL	YMEAS	

The variables shown above are defined in section 1.3. below and in a glossary in section III.3. Some of the variables are also described in Table 4.1 in the introduction to this section.

Additionally, a creative MATLAB user can make use of the global variables to access the Toolbox parameters within the MATLAB programming environment to increase the analysis capabilities of the Toolbox. To access the global variables, the user must enter a global statement within the MATLAB programming environment including the variables of interest (see MATLAB documentation on global variables).

1.3. Save Variable.

This menu item allows variables created within the HELP Toolbox to be saved to the disk as ASCII text files. The variables permitted to be saved and their corresponding filenames are listed below. In each case, a window pops up in which the user may name the saved file. The default filename will appear in the filename block. The user may type any name in place of the default name as well as select the desired directory in which to locate the saved file. The user should take care to name/locate the file uniquely so that subsequent saves do not overwrite a needed file.

1.3.1. Modeling Set (atrain.txt) (m×p).

This menu item allows the user to save the Modeling Set to the computer disk. The Modeling Set is sometimes referred to as training data.

1.3.2. Validation Set (aval.txt) (m×q).

This menu item allows the user to save the Validation Set to the computer disk.

1.3.3. Full Model (afull.txt) (m×n).

This menu item allows the user to save the Full Model to the computer disk.

1.3.4. Reduced Model (ared.txt) (k×n).

This menu item allows the user to save the Reduced Model to the computer disk.

1.3.5. Reduced Measurement Data (ymeas.txt) (r×n).

This menu item allows the user to save the Reduced Measurement Data to the computer disk.

1.3.6. Pivot Vector (test points indices, pr1.txt) (r×1).

This menu item allows the user to save the Pivot Vector to the computer disk. The Pivot Vector maps the reduced test points back into the Full Model (and therefore defines which points are to be measured for the device under test (DUT)).

1.3.7. Validation Parameter Coefficients (xhatval.txt) (n×q).

This menu item allows the user to save the Validation Parameter Coefficients to the computer disk. The parameter coefficients for each vector in the Validation Set determine the contribution of each parameter in the model to the predicted response of the device represented by that validation vector.

1.3.8. Measurement Parameter Coefficients (xhatmeas.txt) (n×r).

This menu item allows the user to save the Measurement Parameter Coefficients to the computer disk. The parameter coefficients for each vector within the Reduced Measurement Data determine the contribution of each parameter in the model to the predicted response of the measured device.

1.3.9. Validation Response Predictions (yhatmat.txt) (m×q).

This menu item allows the user to save the Validation Response Predictions to the computer disk. The Validation Response Predictions are predicted responses at all candidate test points for each vector in the Validation Set, based only upon knowledge of the reduced test points contained within the Validation Set for the particular device and the Full and Reduced Models. The model does not use the knowledge of the remaining measurements for each device contained within the Validation Set. See section II.7. on page 2.7 and following for details.

1.3.10. Measurement Response Predictions (yhatmeas.txt) (m×r).

This menu item allows the user to save the Measurement Response Predictions to the computer disk. The Measurement Response Predictions are predicted responses at all the candidate test points for each measurement vector loaded, based upon the reduced test points represented in the Reduced Model. The Full and Reduced Models are used to produce the predictions.

1.3.11. Validation Residual Errors (resval.txt) (m×q).

This menu item allows the user to save the Validation Residual Errors to the computer disk. The Validation Residual Errors are differences between the measured and predicted responses at all points for each vector in the Validation Set.

1.3.12. Measurement Residual Errors (resmeas.txt).

This menu item allows the user to save the Measurement Residual Errors to the computer disk. The Measurement Residual Errors are differences between the measured and predicted responses at the reduced test points for each vector of measurement data.

1.3.13. Individual Validation Prediction Intervals (iint1val.txt,iint2val.txt).

This menu item allows the user to save the Individual Validation Prediction Intervals to the computer disk. Individual Prediction Intervals provide a 95.45 percent bound for the predicted response at each test point produced using the HELP Toolbox. A 95.45 percent bound is a bound for which each prediction point has a 95.45 percent probability of falling within the upper and lower bound values. This corresponds to coverage of two standard deviations (2-sigma) for a normal distribution. The default filename containing the "1" stores the upper bound and the filename containing the "2" stores the lower bound.

1.3.14. Individual Measurement Prediction Intervals (iint1mea.txt,iint2mea.txt).

This menu item allows the user to save the Individual Measurement Prediction Intervals to the computer disk. Individual Prediction Intervals provide a 95.45 percent bound for the predicted response at each test point produced using the HELP Toolbox. A 95.45 percent bound is a bound for which each prediction point has a 95.45 percent probability of falling within the upper and lower bound values. This corresponds to coverage of two standard deviations (2-sigma) for a normal distribution. The default filename containing the "1" stores the upper bound and the filename containing the "2" stores the lower bound.

1.3.15. Simultaneous Validation Prediction Intervals (sint1val.txt,sint2val.txt).

This menu item allows the user to save the Simultaneous Validation Prediction Intervals to the computer disk. Simultaneous Prediction Intervals provide a 95.45 percent bound for the predicted response at all test points produced using the HELP Toolbox. A 95.45 percent bound is a bound for which there is a 95.45 percent probability that none of the predicted points lies outside. This corresponds to coverage of two standard deviations (2-sigma) for a normal distribution. The default filename containing the "1" stores the upper bound and the filename containing the "2" stores the lower bound.

1.3.16. Simultaneous Measurement Prediction Intervals (sint1mea.txt,sint2mea.txt).

This menu item allows the user to save the Simultaneous Measurement Prediction Intervals to the computer disk. Simultaneous Prediction Intervals provide a 95.45 percent bound for the predicted response at all test point produced using the HELP Toolbox. A 95.45 percent bound is a bound for which all prediction points taken together have a 95.45 percent probability of falling within the upper and lower bounds. This corresponds to coverage of two standard deviations (2-sigma) for a normal distribution. The default filename containing the "1" refers to the upper bound and the "2" to the lower.

1.4. Orthogonalize Modeling Set.

This menu item is useful for assigning a specific set of model vectors to the Full Model and using empirical vectors to augment the assigned vectors. For instance, if the user wanted to assign a set of Rademacher vectors to the Full Model in order to model an Analog-to-Digital Converter (ADC), and augment the set of Rademacher vectors with some measurement data taken from the ADC, the user would load the set of Rademacher vectors into the Full Model using the menu item [Data Sets/Load Data File/Full Model]. The user would then load the measurement data into the Modeling Set using the menu item [Data Sets/Load Data File/Modeling Set Only]. The user would next orthogonalize the Modeling Set to the Full Model using this menu item, i.e., [Data Sets/Orthogonalize Modeling Set]. The next step in the process would be to select parameters for the model using the menu item [Params and Test Pts/Make Parameter Selection]. When the input box appears into which the user will enter the desired number of parameters, the user must be sure the new vectors for the model are appended to the existing Full Model consisting of the Rademacher vectors, by pressing the radio button labeled "Append".

2. Parameters and Test Points.

This menu selection allows the user to mathematically manipulate the matrix of data referred to as the modeling set. The user must produce a full model and a reduced model before he can predict device behavior. Together, the full and reduced models are referred to as the system model because both are required to take a reduced set of device measurements and predict device behavior at all candidate test points.

Selecting model parameters and test points corresponds to selecting the number of unknowns and the number of equations, respectively, in a system of simultaneous linear equations. The user must first determine how many unknowns (model parameters) to use in the system of equations and then establish an appropriate number of equations (test points) to accurately solve the system.

2.1. Modeling Set Decomposition.

This menu item performs the computations necessary for selecting the parameters for the system model. The computations performed are also necessary to the algorithms that correspond to the parameter selection plots used to help select parameters. Therefore, this menu item must be selected prior to attempting to use any of the parameter selection plots and prior to selecting parameters or test points.

A singular value decomposition (SVD) is performed on the modeling set, $\boldsymbol{Y}$. The SVD factors the modeling set into three matrices, $\boldsymbol{Y} = \boldsymbol{USV}^T$, where $\boldsymbol{U}$ is a matrix of orthogonal column vectors that span the same space as the modeling set matrix, $\boldsymbol{S}$ is a diagonal matrix of singular values, and $\boldsymbol{V}$ is an orthogonal matrix that is not used by the Toolbox. The singular values give a quantitative description of how much information from the modeling set is contained in successive vectors of the matrix $\boldsymbol{U}$. The singular values (and thus the columns of $\boldsymbol{U}$) are in decreasing order according to their contribution to the information contained in the modeling set.

2.2. Parameter Selection Plots.

This menu item allows the user to determine the number of parameters significantly affecting device behavior. The number of parameters selected becomes the number of columns in the full and reduced models. The user has several parameter selection plots available (listed below) along with a plot of the singular values (in the Plot menu) in order to make an appropriate parameter size selection. Selecting n parameters corresponds to assigning the first n columns of the matrix $\boldsymbol{U}$ (see Modeling Set Decomposition) as the full model.

2.2.1. Diagnostic Plots.

This menu item plots the Log of Eigenvalues (LEV) Plot, the RMS of Residual Eigenvalues Plot, and the Q-Max Plot (see comments on each below), with a brief note to help in determining the number of parameters for the model.

In practice, the recommendations from the three plots are sometimes inconsistent or unclear. The Q-Max Plot will usually give a maximum, but it may be too large. This may correspond to a fairly even decline in the RMS of Residual Eigenvalues Plot and to the absence of an elbow in the LEV Plot. From numerical experiments, it appears that not much can be gained by increasing the number of parameters in such a case. The recommendation then is to choose the smallest number of parameters that provide sufficient accuracy in the prediction. There are a number of other plots available in the Toolbox, but in an unclear situation they may just add to the confusion. The three plots that are combined in 2.2.1. as well as others are described individually below.

2.2.2. Scree Plot (Eigenvalues).

The eigenvalues are the squares of the singular values (see Modeling Set Decomposition). A typical plot contains an "elbow", i.e., a fairly steep decline up to a particular eigenvalue, followed by a flat part. It is recommended that the user include all parameters up to and including the first eigenvalue in the flat region.

2.2.3. Log of Eigenvalues (LEV).

The LEV Plot is simply the log of the Scree Plot with similar selection characteristics. The LEV Plot may contain an "elbow" as in the Scree Plot or it may have a "step" shape. In that case, the parameters up to and including the bottom of the step should be selected.

2.2.4. Fraction of Variation Explained.

This plot shows the fraction of the cumulative variation explained with increasing parameter selections. The user could use this plot to determine what percentage of information existing within the modeling set is contained in a full model of a certain number of parameters. This plot can be used in conjunction with the Scree Plot to show what percentage of information is "drowned out" by the noise.

2.2.5. RMS of Residual Eigenvalues.

This plot shows the root mean square of the residual eigenvalues. It allows the user to choose the number of parameters such that the expected residuals at individual test points for new devices are sufficiently small. The user can choose the number of parameters such that the plotted quantity is no greater than a predetermined required uncertainty. The plot can alternatively be used to predict the effect of changing the number of parameters. The plot will always indicate that increasing the number of parameters decreases the size of the residuals. However, as the number of parameters grows large, the model may capture more noise.

2.2.6. Q-Max Plot.

This plot shows a statistical estimate for the dimension of the residuals of training data from a model with n parameters. The number of parameters selected should correspond to the maximum value of the plot or the smallest value for which the plot reaches a plateau.

2.3. Select Number of Parameters

Once the user has used the Parameter Selection Plots to determine the desired number of parameters, selecting this menu item produces a pop-up window in which the user can enter the desired number of model parameters.

2.4. Test Point Selection.

For a given number of model parameters, test points must be selected to solve the system of equations. Based upon experience with empirical model building, the Toolbox authors suggest using a quantity of test points equal to two to five times the number of model parameters. The test points can be chosen arbitrarily by the user or selected using the prediction variance optimization routine described below.

2.4.1. Prediction Variance Optimization.

Optimally select test points to be included in the reduced model by computing the ratio of the variance of the prediction to the variance of the measurement noise at all points and choosing the test point with the highest ratio value to be included in the reduced model. This ratio will change after each additional test point so the ratio is recalculated for each iteration until the desired reduced model size is established (see above theory).

2.4.2. Test Point Assignment.

This menu item allows the user to assign specific test points to the reduced model. The vector of the assigned test points is referred to as the pivot vector.

2.4.2.1. Assign Test Point File.

This menu item allows the user to load a file into the Toolbox environment to specify which of the candidate test points are to be used in the reduced model. The file must contain a vector of indices, the largest of which must be less than or equal to the number of candidate test points.

2.4.2.2. Manual Entry.

The user may use this menu item to manually type into a data entry window the indices of the test points to be included in the reduced model. The largest entry must be less than or equal to the number of candidate test points.

2.4.3. Combined Optimization and Assignment.

This menu selection allows the user to use both optimally select test points and manually select test points. The first n points selected optimally are selected using the QR factorization. Additional optimally selected test points are selected using prediction variance optimization. Manually assigned points are selected in addition to the optimal points and may repeat points selected optimally. Upon selection of this menu item, the user must choose how many optimal points are to be selected and type into the input window the indices of the test points that are to be manually assigned. Press the OK button and the Toolbox will produce the appropriate model.

3. Assess Model.

This menu item computes the response predictions for either the validation set or the reduced measurement data (evaluating actual or simulated Device Under Test (DUT)). The validation error statistics and the DUT error statistics are displayed on the front panel of the HELP Toolbox. Also, the Plot menu item will allow the user to display the predicted response, actual response (for the case of validation), and residual errors.

3.1. Validate Model.

Predict the response at all candidate test points in the validation set based on calculations using only the reduced model points extracted from the validation set and compute the residual errors (measured response - predictions) at all candidate test points for each vector/data set within the validation set.

3.2. Predict Calibration.

Predict the response at all unmeasured candidate test points and compute the residual errors (measured response - predictions) for all measured points. Reduced measurement data for calibration can be loaded into the Toolbox environment via the [Load Data] menu item.

3.3. Multiple Model RMS Error Results.

This menu item computes the rms of the residual errors produced for applying numerous models to the validation set. The Toolbox will prompt the user for a vector of parameter sizes and a vector of test-point sizes which the user is interested in testing. A matrix will be generated in which for each combination of parameter size and test-point size, the corresponding rms of the residual error will be computed for each validation vector. Combinations with more parameters than test points will be ignored by the software. The user may look at a plot of the results to determine optimal model sizes with respect to rms error values. Note that this menu item may take many minutes or even hours depending on the number of parameter and test point combinations and depending on the computer system used.

4. Quality Control

Once device predictions are made using a constructed model, it is desirable to know the uncertainty associated with the prediction. This menu item allows the user to compute prediction intervals around the predictions that provide 95.45 percent bounds (coverage of twice the standard deviation of a normal distribution).

4.1. Individual Prediction Intervals.

This menu item allows the user to compute prediction intervals for each test point represented within the system model.

4.1.1. Validation.

Compute an interval around each predicted response for the validation set that has a 95.45 percent statistical probability of containing the measured value. The prediction intervals (and thus the model) can be tested by checking to see what percentage of the measurements at points not included in the reduced model fall inside and outside of the prediction intervals. At least 95.45 percent of the points should lie within the intervals. Note that this check of the model and intervals assumes that the data in the modeling and validation sets are statistically representative of the system under test.

4.1.2. Measurement.

Compute an interval around each response prediction that has a 95.45 percent statistical probability of containing the measured value. The Reduced Measurement Data consist of measurements at a subset of all candidate test points represented in the Full Model.

5. Plot

View plots of quantities previously created within or loaded into the HELP Toolbox.

5.1. Measurement Vectors.

Allows the user to select vectors from one of several sets to plot. The user must enter the indices corresponding to the columns of the set that the user desires to plot. The user must enter an index or set of indices that are contained within the data set or an error will occur. The user may choose to plot multiple vectors horizontally on top of one another or vertically as separated columns. The user may also select between the plotting of normalized or unnormalized data.

5.1.1. Modeling Set.

Plot columns from the modeling set.

5.1.2. Model Vectors.

Plot columns from the full model.

5.1.3. Validation Set.

Plot columns from the validation set.

5.1.4. Reduced Measurement Data.

Plot columns from the reduced measurement set.

5.2. Normalization Vector.

Plot the vector used to normalize the modeling set and validation set. This vector will typically be a tolerance or uncertainty vector for the modeled device.

5.3. Singular Values.

Plot the singular values produced from performing a singular value decomposition on the modeling set. This plot gives an indication of the contribution of successive parameters to the model.

5.4. Validation Analysis.

This section allows the user to plot vectors computed from applying the model to the validation set. Prior to plotting from the validation analysis sub-menu, the user must have created full and reduced models from data loaded into the Toolbox or loaded them into the Toolbox directly, and then performed a validation of the model using a validation set that was loaded into the Toolbox.

5.4.1. Response Predictions.

Plot the response predictions computed for the validation set. The user must select the validation vector(s) for which the corresponding response predictions are to be plotted. The user has the option of plotting normalized or unnormalized data.

5.4.2. Residual Errors Statistics.

When the user validates the model, error statistics are computed for the predictions made at the total candidate set of test points based only on the reduced test points.

5.4.2.1. Residual Errors.

Plot the residual errors (measurement-prediction) computed for the validation set. The user must select the validation vectors for which the corresponding residual error vectors are to be plotted. The user has the option of plotting normalized or unnormalized data.

5.4.2.2. True Maximum Per Device.

Plot the maximum of the residual errors computed for each validation vector.

5.4.2.3. True Minimum Per Device.

Plot the minimum of the residual errors computed for each validation vector.

5.4.2.4. Absolute Maximum Per Device.

Plot the absolute maximum computed for each validation vector.

5.4.2.5. RMS Per Device.

Plot the root-mean-squares of the residual errors computed for each validation vector.

5.5. Analysis of DUT.

5.5.1. Response Predictions.

Plot the response predictions for the reduced measurement data. The user must select the measurement data vector for which the corresponding response predictions are to be plotted. If measurement data for a single device under test has been loaded into the Toolbox, the user must enter 1. The user may also select between plotting of normalized or unnormalized data.

5.5.2. Residual Error Statistics.

5.5.2.1. Predicted Residual Errors.

Plot the predicted residual error for the reduced measurement data. The user must select the measurement data vector for which the corresponding predicted residual error vector is to be plotted. If measurement data for a single device under test has been loaded into the Toolbox, the user must enter 1. The user may also select between the plotting of normalized or unnormalized data.

5.5.2.2. RMS Per Device.

Plot the root-mean-squares of the predicted residual error for the reduced measurement data.

5.6. Parameter Coefficient Vector.

5.6.1. Validation.

Plot the parameter coefficient vector computed for the model for the particular validation vector selected by the user.

5.6.2. Measurement.

Plot the parameter coefficient vector computed for the model and particular reduced measurement vector selected by the user.

5.7. Prediction Variance Vectors.

Plot the prediction variance vectors produced successively by selecting test points for the reduced model. Successive vectors decrease in magnitude. The maximum value for all previously unselected test points corresponds to the next test point selected in the optimal test point selection process.

5.8. Prediction Intervals.

5.8.1. Individual.

Plot 2-sigma (95.45 %) individual prediction intervals for a validation vector or a reduced measurement vector. The user may plot the intervals either normalized or unnormalized and either separately or with the predictions added to the prediction intervals. In all cases, the predictions are plotted to allow the user to view the coverage probability. The exact coverage percentage may be obtained by defining the variable TOTAL_MEAS_OUT_I global within the MATLAB command window. This gives the number of predictions not covered by the prediction intervals. To obtain the percentage, take 1 minus the ratio of this number to the total number of predictions and multiply by 100.

5.8.1.1. Validation.

Select the validation vector for which individual prediction intervals are plotted.

5.8.1.2. Measurement.

Select the measured data vector for which individual prediction intervals are plotted.

5.9. Multiple Model RMS Error Results.

Plot rms error results for the models tested with the menu item [Assess Model/Multiple Model RMS Error Results].

5.10. Control Plot Axes.

This menu item allows the user to control the range for the x- and y-axis settings.

5.11. Hold Plot Axes.

Fix current plot axes so that plots may be placed on the same axes. The command does not freeze the axes, but rather flexibly allows multiple plots on the same axes. Once Hold On is selected, Hold Off must be selected prior to viewing a new plot separately.

5.11.1. Hold On.

This menu item causes all subsequent plots to be placed on the same axes.

5.11.2. Hold Off.

This menu item causes subsequent plots to clear previous plots before displaying.

5.12. Grid

Place a grid on the plot axes.

5.13. Clear Plot.

Clear the plot axes.

5.14. Print Figure.

Print a plot of the current figure.

5.15. Print Bitmap of Entire Window.

Print a bitmap of the entire HELP Toolbox Window (everything below the menu labels).

6. Help.

Call the HTML navigator and the HTML help pages for the HELP Toolbox.

7. Exit.

Exit the HELP Toolbox.

V. HELP Toolbox Tutorial

1. Example: Modeling an Analog Instrument with an Empirical Model

To start the HELP Toolbox from within the MATLAB® Command Window, type

```
>>help22
```

The High-dimensional Empirical Linear Prediction Toolbox window will appear. As you browse the Toolbox, you will see the menu headings [Data Sets], [Params and Test Pts], [Assess Model], [Quality Control], [Plot], [Help], and [Exit]. Each of these main menu items has corresponding submenus. (For the sake of clarity and consistency, all references to menu labels are enclosed by square brackets.)

To begin using the HELP Toolbox, we must load some previously collected data into the Toolbox. Select the menu item [Data Sets/Load Data File/Modeling Set and Validation Set].

You will be prompted to select the file format for the file you wish to enter.

Select MATLAB® Binary Format (*.mat). Within the Load Data Window that pops up, change the folder directory to C:\HelpData\792A (or where ever the data exist). Select the filename "m792_309.mat" by single-clicking the filename and choosing the "Open" button or by double-clicking the filename.

You will see the size of the data file displayed as 309 rows and 126 columns. This is referred to as a matrix of size 309×126. You will next be asked if you would like to assign some of the data for model validation.

Choose the "Yes" button. Then, within the prompt window, select the button labeled "Manually". Place your cursor in the light blue box, click the mouse, and type "[101:126]" (without the quotes). Choose the "OK" button.

This tells the Toolbox to select the 101st through the 126th vectors from the data set and assign them to the validation set. You will see the sizes of the modeling and validation sets displayed as 309×100 and 309×26, respectively. Notice underneath both the modeling set sizes and validation set sizes, there are text objects labeled "Unnormalized". This informs the user that neither matrix has been normalized.

Select the menu item [Data Sets/Load Data File/Normalization Vector] in order to normalize the modeling and validation sets to a calibration tolerance vector. Choose MATLAB® Binary Format (*.mat). Change the folder directory to C:\HelpData\792A. Select the filename "fluk_792.mat". Note that when the user selects MATLAB Binary File type, the Toolbox looks specifically for *.mat files so the ".mat" extensions are not listed in the directory window.

Another small window will appear asking if you would like to normalize the modeling and validation sets.

Normalize both the modeling and validation sets by choosing the "Both" button.

After choosing the "Both" button, you will notice the text object labels change to "Normalized". Any subsequent normalization of the modeling and validation sets will not change this normalization flag so pay attention to the normalization flag prior to using it.

Now that you have a modeling set, you are ready to build an empirical model. Keep in mind that the goal is to build a model that characterizes the device or instrument of interest with as few test points as necessary to achieve the accuracy and confidence levels desired.

Select the menu item [Params and Test Pts/Modeling Set Decomposition].

This performs a factorization of the modeling set so that the column dimension of the model may be reduced.

Next, select the menu item [Params and Test Pts/Parameter Selection Plots/Diagnostic Plots].

The plots shown within the Toolbox give some graphical information helpful in determining the desired number of parameters for the model. The interpretation of these plots is discussed in the section III.

Description of Toolbox Menus and Variables. The Toolbox window with the plots displayed is shown in Figure 5.1. Next, we will select 20 parameters for the model.

Determining the desired number of parameters for a model can depend on the user's definition of an acceptable error level for the device under test. It is often beneficial to construct many models with differing numbers of model parameters and look at the resulting errors. Next we consider the effects of varying the number of test points selected for the model. A general guideline is to select test points totaling between two and five times the number of parameters chosen. Having an overdetermined system (more test points than parameters) helps to reduce the measurement noise and flag errors in the model and/or in the device under test. Given that there must be significantly more test points than model parameters, adding additional parameters to the model will require a greater number of test points. Since reducing the number of test points is the means by which the cost of testing is lessened, this reduction in model size must be weighed with the user's error requirements.

Select the menu item [Params and Test Pts/Select Number of Parameters]. An input box will appear. Click on the blue area of the input box, type "20", and press the "OK" button.

Notice on the front panel that a full model of size 309×20 has been created.

The next step is to choose a set of test points for the model so select the menu item [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. An input box containing a slider will appear. Enter the value "80" into the blue area showing the slider value or move the slider until "80" appears, and press the "OK" button.

You have just selected 80 test points creating a Reduced Model of size 80×20. Notice that the number of rows for the Reduced Model displays "80" on the front panel.

Next validate the model by selecting the menu item [Assess Model/Validate Model].

Notice that the Validation Error Statistics are filled in on the front panel. Recall that you set aside some of the initial data for validation. This portion of the data was not used in creating the model.

Plot the first vector in the Validation Set using the menu item [Plot/Measurement Vectors /Validation Set]. Click on the blue area, enter "1" to select the first validation vector, and press the "OK" button. Next we want to plot the predicted response for the first vector in the Validation Set right on top of the present plot so hold the current axes in place by selecting [Plot/Hold Plot Axes/Hold On]. Then select [Plot/Validation Analysis/Response Predictions], enter "1" in the blue area (because we want the first vector again), and press the "OK" button. Plot the Residual Errors on the same plot by selecting [Plot/Validation Analysis/Residual Error Statistics/Residual Errors]. Select [Plot/Hold Plot Axes/Hold Off] to free the plot axes for later plots.

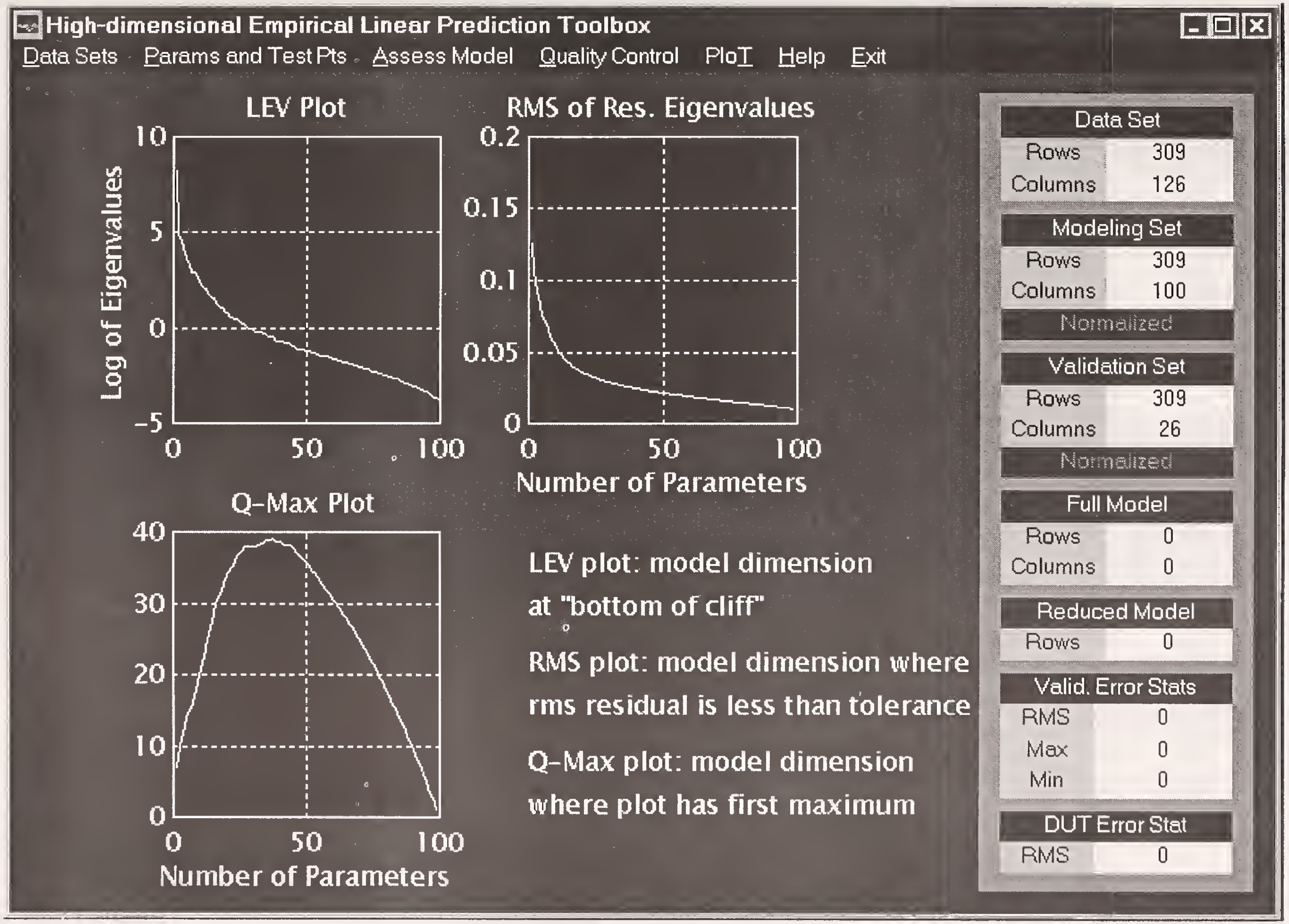
This plot now shows the set of measurements represented in the first validation set, the predictions based on measuring only the 80 test points selected, and the residual errors for those predictions. Figure 5.2 shows an image of the Toolbox window with all its parameters displayed on the front panel and the plots displayed in the graph portion of the window.

Suppose you want to determine whether the model accurately characterizes a validation device. This determination can be made using prediction intervals.

Compute prediction intervals by selecting the menu item [Quality Control/Individual Prediction Intervals/Validation].

Plot the prediction intervals for the first validation set using [Plot/Prediction Intervals/Individual/Validation]. Click in the blue area, type a "1" and press the "OK" button. Select [Plot/Control Plot Axes], change the x-axis range to [0 50], and press the "Apply" button.

The prediction intervals can be seen to be about plus and minus 20 percent of the tolerance (normalization vector) about the predictions. Figure 5.3 contains a picture of the intervals with the modified x-axis.



LEV plot: model dimension at "bottom of cliff"

RMS plot: model dimension where rms residual is less than tolerance

Q-Max plot: model dimension where plot has first maximum

Data Set	
Rows	309
Columns	126

Modeling Set	
Rows	309
Columns	100
Normalized	

Validation Set	
Rows	309
Columns	26
Normalized	

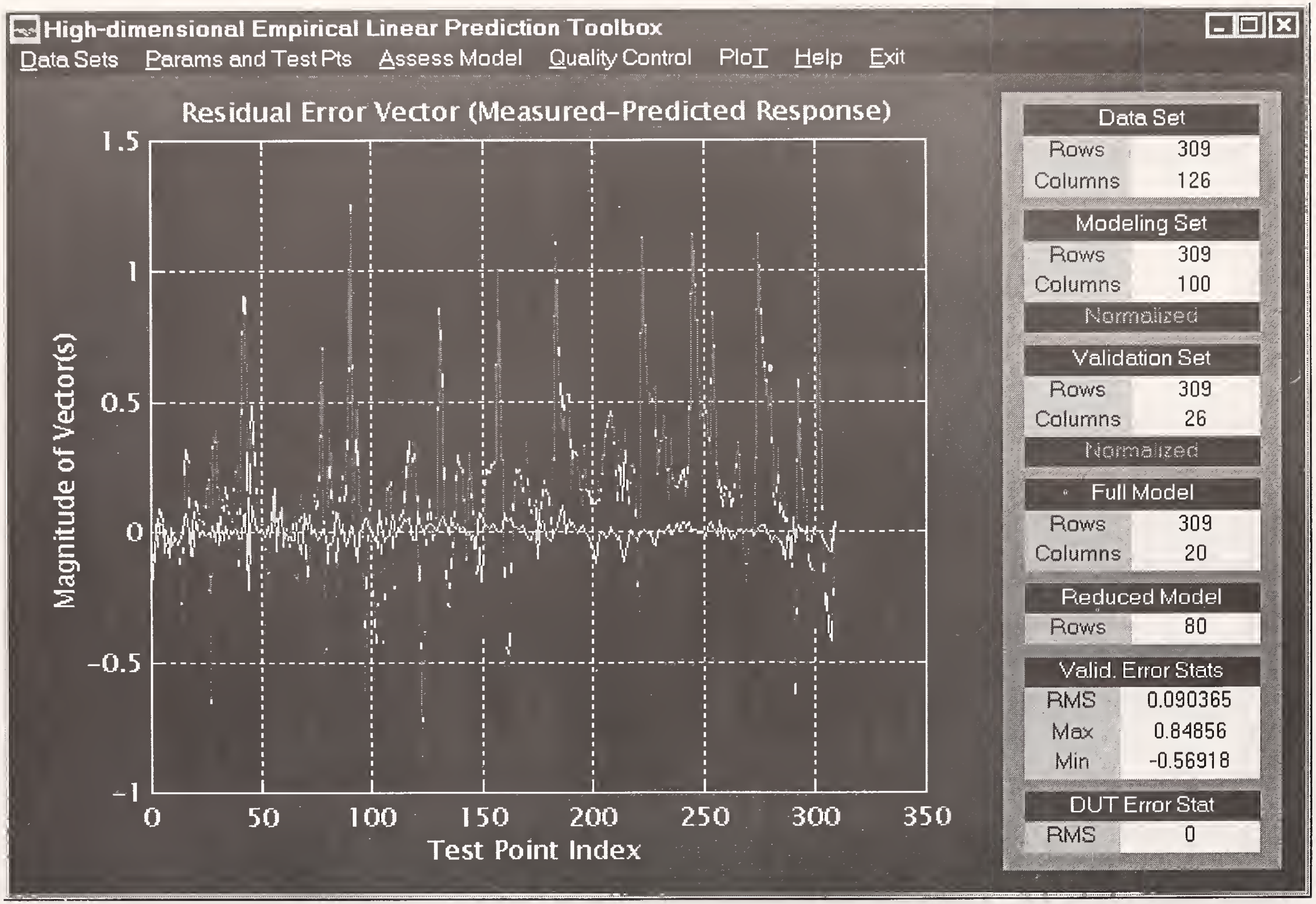
Full Model	
Rows	0
Columns	0

Reduced Model	
Rows	0

Valid. Error Stats	
RMS	0
Max	0
Min	0

DUT Error Stat	
RMS	0

Figure 5.1. Diagnostic Plots (Determining Number of Model Parameters)



Data Set
Rows 309
Columns 126

Modeling Set
Rows 309
Columns 100
Normalized

Validation Set
Rows 309
Columns 26
Normalized

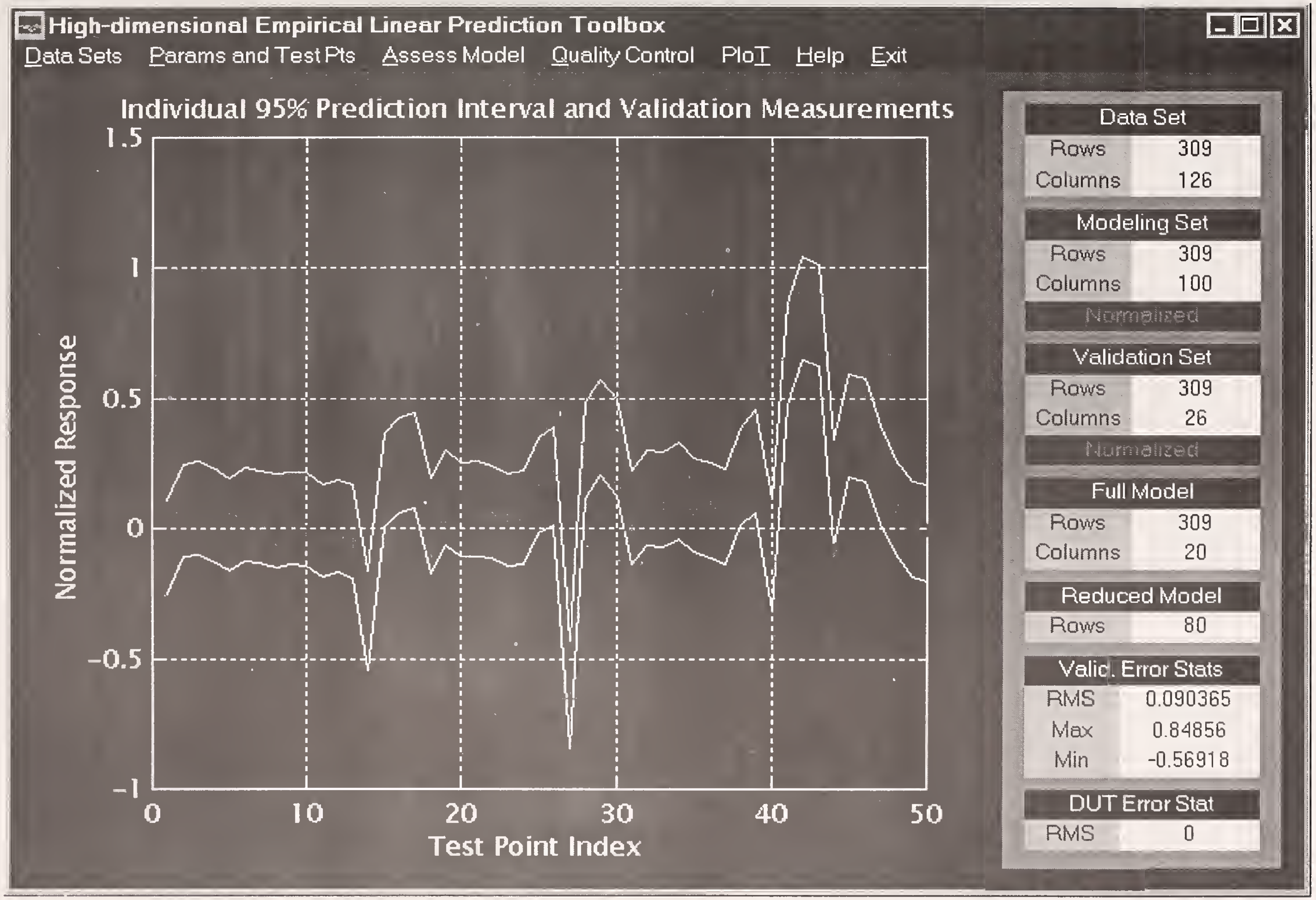
Full Model
Rows 309
Columns 20

Reduced Model
Rows 80

Valid. Error Stats
RMS 0.090365
Max 0.84856
Min -0.56918

DUT Error Stat
RMS 0

Figure 5.2. Measurement, Prediction, and Residual Error.



Data Set	
Rows	309
Columns	126
Modeling Set	
Rows	309
Columns	100
Validation Set	
Rows	309
Columns	26
Full Model	
Rows	309
Columns	20
Reduced Model	
Rows	80
Valid. Error Stats	
RMS	0.090365
Max	0.84856
Min	-0.56918
DUT Error Stat	
RMS	0

Figure 5.3. Prediction Interval with Plot Axes Modified.

The errors produced from applying the model to the validation set may be observed using any of the menu items under the label [Plot/Validation Analysis/...].

The previous model was constructed using arbitrary selections from the data for various parameters, such as modeling and validation set divisions, model parameters, and test points. Next, we will construct a model using statistical and engineering criteria for making such selections, as required to properly apply the HELP approach. The modeling and validation sets will be selected according to time of measurements so as to place broad parameter variability in both sets. The number of parameters will be varied to investigate the errors produced over varying numbers of model parameters.

Select [Data Sets/Clear Current Data] from the Toolbox menu. Notice that the entire plot portion of the window is cleared and all parameter listings in the front panel of the window are set to zero. Additionally, the normalization flags for the modeling and validation sets are reset to "Unnormalized". Now reload the same data set using [Data Sets/Load Data File/Modeling Set and Validation Set]. Select MATLAB Binary Format (*.mat) for the type of file to be loaded. Find the file under C:\HelpData\792a\M792_309.mat. Again, recall that all ".mat" extensions will be hidden because the Toolbox is looking specifically for these files. Select the filename "m792_309.mat" by either double clicking the filename or choosing the "OK" button.

You will see the size of the data file displayed as 309×126. Next we will assign a more appropriate portion of the data for model validation than previously assigned.

Choose the "Yes" button. Then, within the prompt window, select the button labeled "Manually". Place your cursor in the light blue box, click the mouse, and type "[1:5:126]" (without the single quotes)(square brackets indicate a vector or matrix and may be omitted if no comma is needed to separate elements). Choose the "OK" button.

This tells the Toolbox to separate every fifth vector from 1 through 126 from the data set and assign it to the validation set. You will see the sizes of the modeling and validation sets displayed as 309×100 and 309×26, respectively. Again, notice underneath both the modeling set size and validation set size, there are text objects labeled "Unnormalized". This informs the user that neither matrix has been normalized.

Select the menu item [Data Sets/Load Data File/Normalization Vector] in order to normalize the modeling and validation sets to a calibration tolerance vector. Choose MATLAB Binary Format (*.mat). Change the folder directory to C:\HelpData\792a. Select the filename "fluk_792.mat" either by selecting the file and clicking on the "OK" button or double-clicking the filename.

Normalize both the modeling and validation sets by choosing the "Both" button.

After choosing the "Both" button, you will notice the text object labels change to "Normalized".

Select the menu item [Params and Test Pts/Modeling Set Decomposition].

This performs a factorization of the modeling set so that the column dimension of the model may be reduced.

Next, select the menu item [Params and Test Pts/Parameter Selection Plots/Diagnostic Plots].

The plots shown within the Toolbox give graphical information helpful in determining the desired number of parameters for the model.

Select the menu item [Params and Test Pts/Select Number of Parameters]. An input box will appear. Click on the blue area of the input box, type "25", and press the "OK" button.

Notice on the front panel that a full model of size 309×25 has been created.

Now select the menu item [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. An input box containing a slider will appear. Enter the value "80" into the blue area showing the slider value or move the slider until "80" appears, and press the "OK" button.

You have just selected 80 test points creating a Reduced Model of size 80×25.

Next validate the model by selecting the menu item [Assess Model/Validate Model].

Notice that the Validation Error Statistics are filled in on the front panel. (Recall that you set aside some of the initial data for validation.)

Note that the RMS, Max, and Min for the residual errors are 0.041152, 0.50888, and -0.21042, respectively.

Now, look at the true maximum, true minimum, absolute maximum, and rms of the residual errors from the validation set using the menu item [Plot/Validation Analysis/Residual Error Statistics/...]. Next hold the plot axes fixed and plot all the plots on top of each other for comparison and contrast by selecting [Plot/Hold Plot Axes/Hold On] after the first plot. Remember to select [Plot/Hold Plot Axes/Hold Off] afterwards to release the plot axes. The Toolbox window containing these plots is shown in Figure 5.4.

Next, create reduced measurement data from the validation set using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click in the blue text field, type "1", and hit "OK". Another window will pop up to make sure you want to overwrite any existing reduced measurement data. We have not yet assigned any reduced measurement data ... hit the "OK" button. Predict the measurement response using [Assess Model/Predict Calibration]. Note the DUT Error Stat on the front panel contains a value. Take a look at the measurements by selecting [Plot/Measurement Vectors/Reduced Measurement Data]. Click on the mouse in the blue area, type "1" (there is only one reduced measurement vector), and press the "OK" button. Hold the plot axis using [Plot/Hold Plot Axes/Hold On]. Plot the predicted response using [Plot/Analysis of DUT/Response Predictions]. Click on the blue text filed, enter "1", and hit "OK". The actual measurements are shown as light blue circles and the predictions are displayed as a red curve. Plot the predicted residual errors using [Plot/Analysis of DUT/Residual Error Statistics/Predicted Residual Errors]. Click in the blue text field, type "1", and hit "OK". The entire window containing these plots is shown on the following page. Select [Plot/Hold Plot Axes/Hold Off] to release the plot axes. Figure 5.5 shows the Toolbox window containing the plot created.

Now go back and create a model of size 80×32 and compare the residual statistics.

Select [Params and Test Pts/Select Number of Parameters]. Enter "32" in the blue area. Be sure to check the "Replace" button within the parameter selection box so that instead of appending vectors to the present model, a new model is created. Note on the front panel that the Reduced Model row-size, the Valid. Error Stats, and the DUT Error Stat are all set to zero, indicating that a new reduced model must be selected. To choose a new set of reduced test points, select the menu item [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. An input box containing a slider will appear. Move the slider until the value "80" appears or enter "80" into the blue text-input area. Then press the "OK" button.

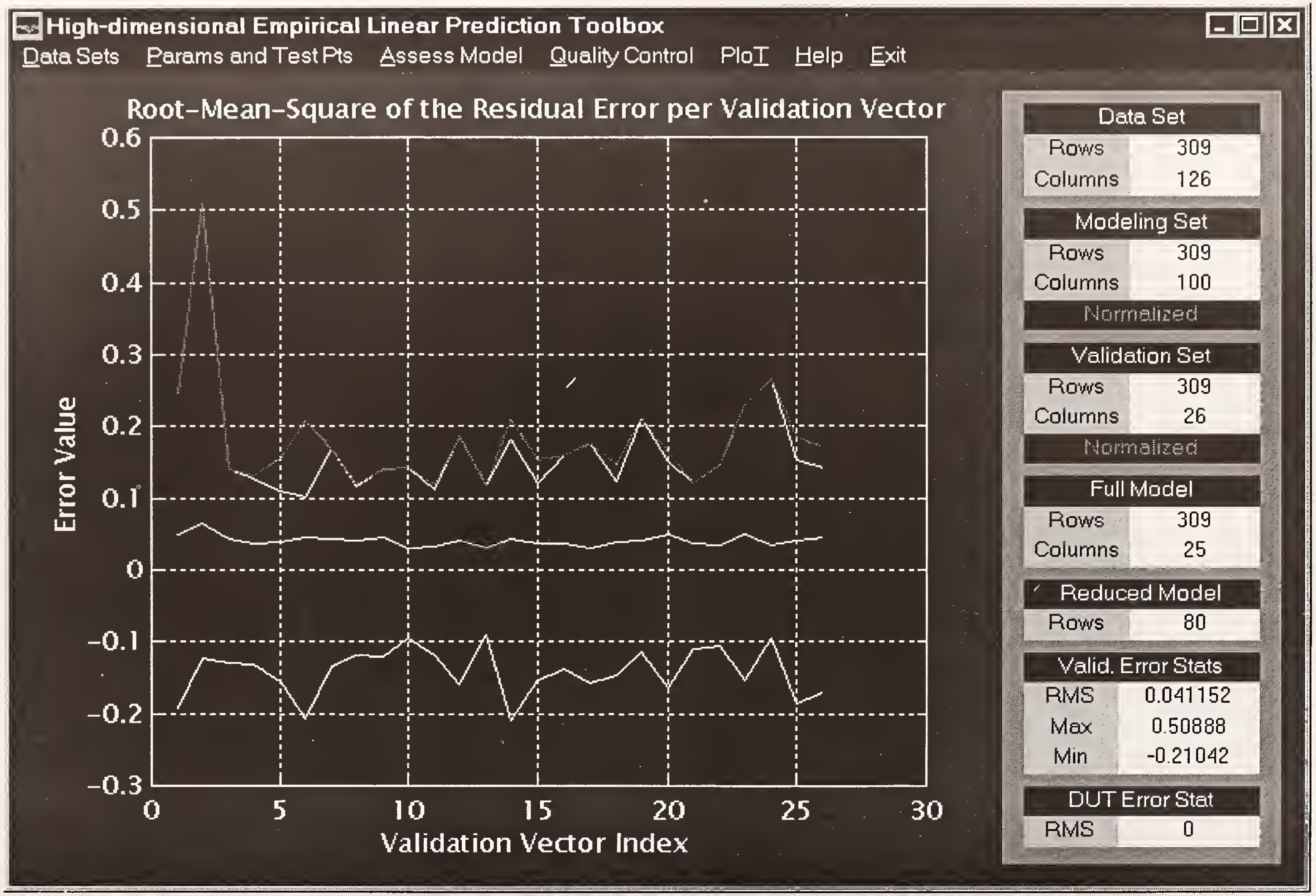
You have just selected 80 test points creating a reduced model of size 80×32.

Next, validate the model using [Assess Model/Validate Model].

Now the RMS, Max, and Min are 0.040057, 0.51302, and -0.21188, respectively. The changes in the RMS (slight decrease), Max, and Min are all negligibly small.

Next, create reduced measurement data from the validation set using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click in the blue text field, enter "1", and hit "OK". Again, you will be asked if you are sure you want to overwrite already-existing reduced measurement data. Hit the "OK" button. Predict the measurement response using [Assess Model/Predict Calibration]. Again, try plotting the reduced measurement vector, the predicted response, and the predicted residual errors using [Plot/Measurement Vectors/Reduced Measurement Data] and [Plot/Analysis of DUT/...].

Repeat the procedure with 40 model parameters and 80 test points. Increasing the model size to 80×40 produces RMS, Max, and Min of 0.038099, 0.49591, and -0.21158, respectively. The RMS error value continues to decrease, but very slowly. Next, select the menu item [Params and Test Pts/Parameter Selection Plots/Diagnostic Plots].



Data Set
Rows 309
Columns 126

Modeling Set
Rows 309
Columns 100
Normalized

Validation Set
Rows 309
Columns 26
Normalized

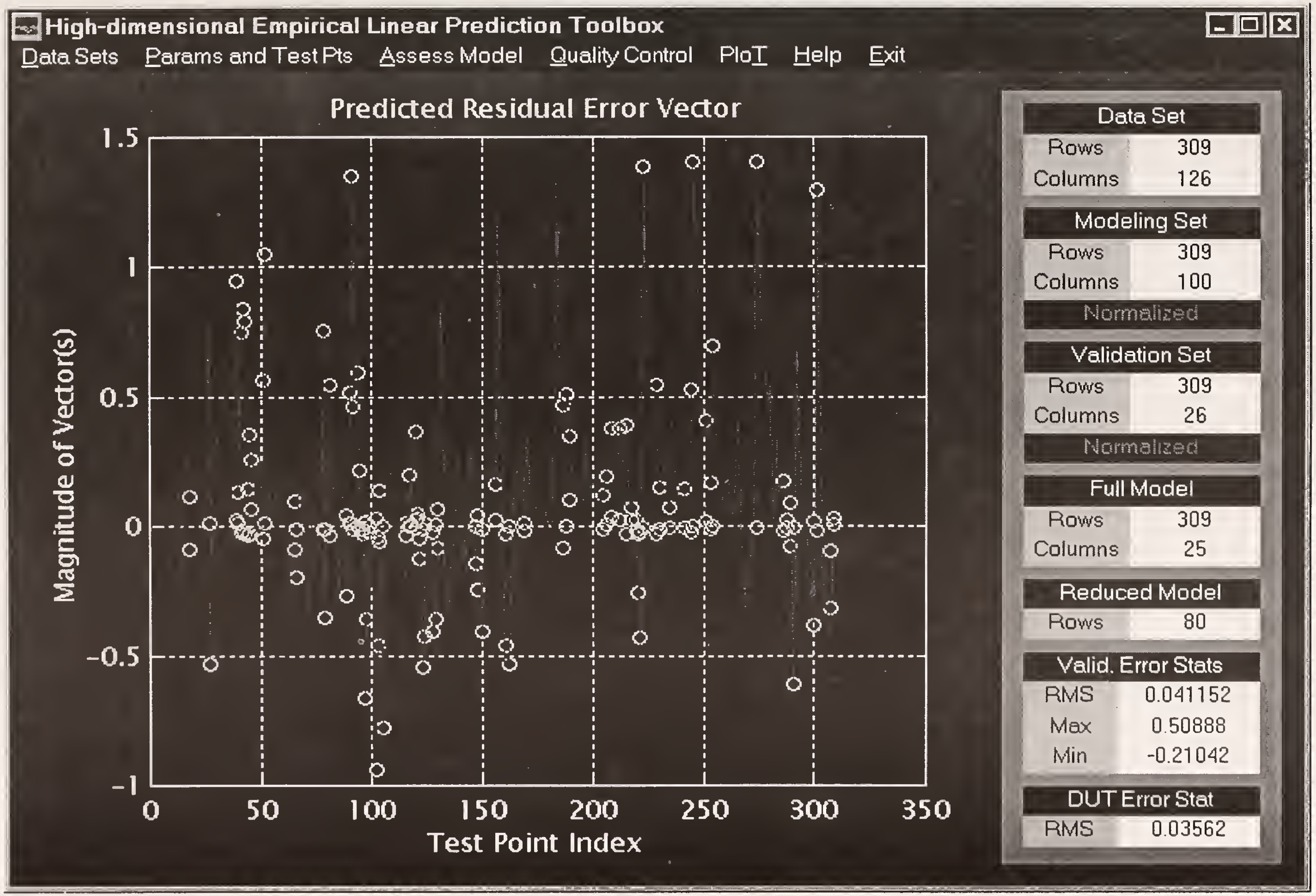
Full Model
Rows 309
Columns 25

Reduced Model
Rows 80

Valid. Error Stats
RMS 0.041152
Max 0.50888
Min -0.21042

DUT Error Stat
RMS 0

Figure 5.4. Max, Min, Absolute Max, and RMS of Residual Error.



Data Set
Rows 309
Columns 126

Modeling Set
Rows 309
Columns 100
Normalized

Validation Set
Rows 309
Columns 26
Normalized

Full Model
Rows 309
Columns 25

Reduced Model
Rows 80

Valid. Error Stats
RMS 0.041152
Max 0.50888
Min -0.21042

DUT Error Stat
RMS 0.03562

Figure 5.5. Reduced Measurement, Prediction, and DUT Residual Errors.

These plots give graphical information to enable the user to determine the desired number of parameters for the model. We will choose 32 parameters for the model and investigate what happens when the number of reduced test points is varied.

Select the menu item [Params and Test Pts/Select Number of Parameters]. An input box will appear. Click on the blue area of the input box and type "32". Remember to select the "Replace" button instead of "Append" so that we create a new model. Press the "OK" button. Next, select the menu item [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. An input box containing a slider will appear. Enter the value "70" into the blue area showing the slider value or move the slider until "70" appears, and press the "OK" button.

The Toolbox front panel should display the full model size as 309×32 and the number of rows for the reduced model as 70.

Next validate the model by selecting the menu item [Assess Model/Validate Model].

Note that the RMS, Max, and Min for the residual errors are 0.040624, 0.51887, and -0.20719, respectively.

Now, look at the validation analysis plots using the menu item [Plot/Validation Analysis/...]. Again, try holding on the plot axis to place the plots on top of each other for comparison and contrast by selecting [Plot/Hold Plot Axes/Hold On] after the first plot. Remember to select [Plot/Hold Plot Axes/Hold Off] before proceeding to a different set of plots.

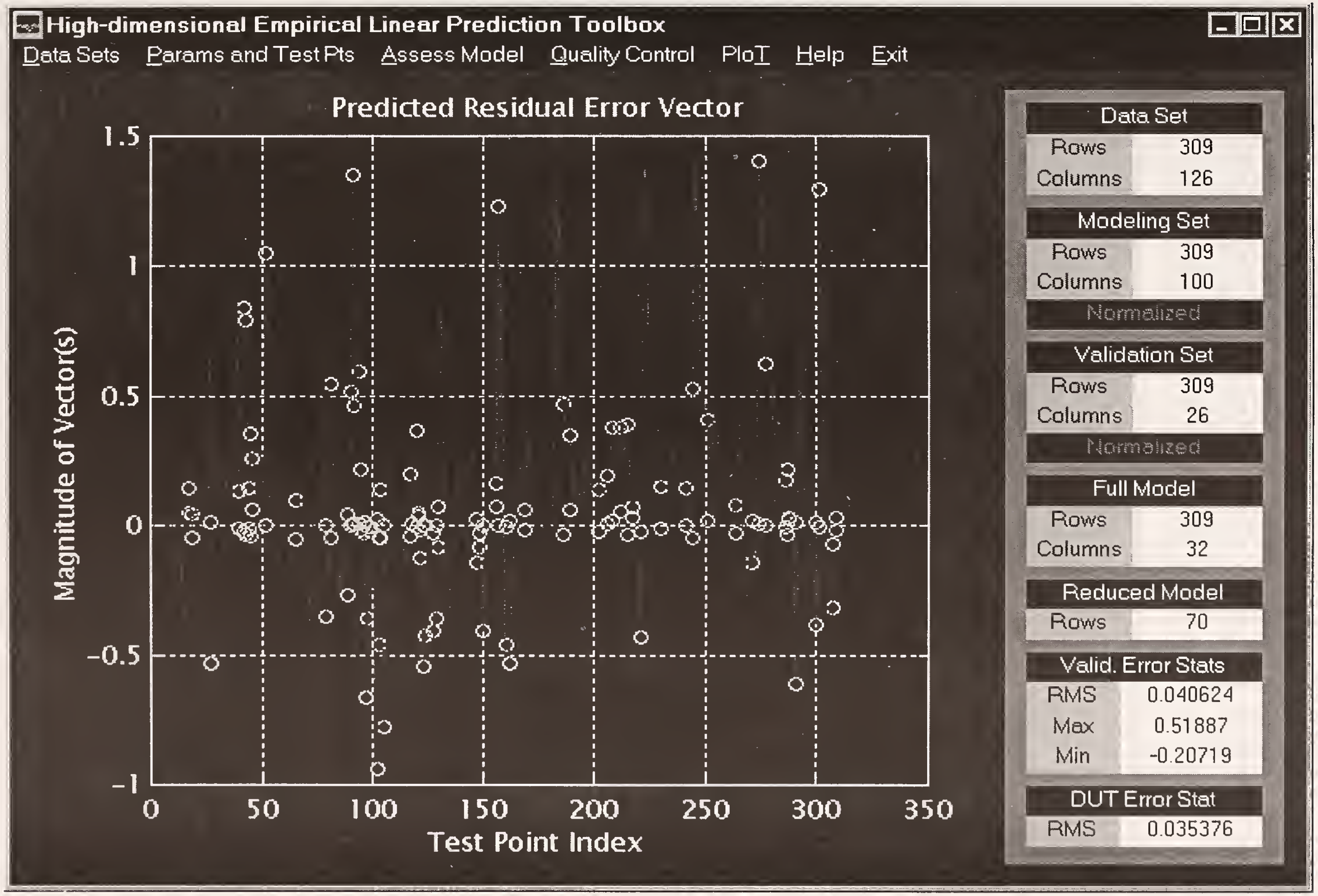
Take a look at the measured values (validation) and prediction on the same plot. Select [Plot/Measurement Vector/Validation Set]. Enter a "1" in the blue area. Select [Plot/Hold Plot Axes/Hold On] to fix the plot axes. Select [Plot/Validation Analysis/Response Predictions] and enter a "1" in the blue area. Select [Plot/Validation Analysis/Residual Error Statistics/Residual Errors] and enter a "1" in the blue area. You have plotted the measurements, predictions, and residual errors for the first validation vector based on a model of size 70×32.

Now create reduced measurement data from the validation set using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click the mouse in the blue text field, enter "1" to use the first validation vector, and hit the "OK" button. This operation extracts the 70 selected test points from the first vector in the validation set and uses it as a device under test. Selecting the menu item [Assess Model/Predict Calibration], the Toolbox predicts the response at all 309 points using only knowledge of the 70 selected measurement points. Now plot the reduced measurement vector using [Plot/Measurement Vectors/Reduced Measurement Data]. Click the mouse in the blue text field, enter "1", and click on the "OK" button. Fix the plot axes using [Plot/Hold Plot Axes/Hold On]. Now plot the predicted response for the DUT using [Plot/Analysis of DUT/Response Predictions]. Click in the blue text field, enter "1", and hit "OK". The light blue circles are the actual measurements at the 70 selected points and the red curve is the predicted device response at all 309 points based on knowledge of only the 70 measurements. Plot the predicted residual errors for the DUT using [Plot/Analysis of DUT/Residual Error Statistics/Predicted Residual Errors]. Again, click in the blue text field, enter "1" (there is only one DUT vector), and hit "OK". The Toolbox window containing plots of the reduced measurement data, the predicted response, and the predicted errors is shown in Figure 5.6.

Now go back and select 80 test points for a reduced model of size 80×32 and compare the residual statistics with those of the 70×32 model.

Select [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. Enter "32" in the blue area. Next, validate the model using [Assess Model/Validate Model].

Note that the RMS, Max, and Min for the residual errors are 0.040057, 0.51302, and -0.21188, respectively. The RMS and Max values have improved only slightly.



Data Set	
Rows	309
Columns	126

Modeling Set	
Rows	309
Columns	100

Normalized

Validation Set	
Rows	309
Columns	26

Normalized

Full Model	
Rows	309
Columns	32

Reduced Model	
Rows	70

Valid. Error Stats	
RMS	0.040624
Max	0.51887
Min	-0.20719

DUT Error Stat	
RMS	0.035376

Figure 5.6. Reduced Measurement, Prediction, and DUT Residual Errors.

Again, try all the plotting combinations for the residual errors using [Plot/Validation Analysis/Residual Error Statistics/...]. Plot a validation vector with its response prediction and residual error.

Next, create reduced measurement data using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click in the blue text field, enter "1", and hit "OK". Hit "OK" again. Predict the measurement response using [Assess Model/Predict Calibration]. The RMS of the predicted errors is displayed on the front panel as 0.037182 which is comparable to the value of 0.040057 for the RMS of the errors for the validation set.

Next, select 90 test points test points for the model using [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. Validate the model using [Assess Model/Validate Model]. Increasing the model size to 90×32 produces RMS, Max, and Min of 0.039182, 0.51244, and -0.20298, respectively. The RMS error value has decreased with more test points. Again, create a reduced measurement vector using [Data Sets/Load Data File/Extract Reduced Meas. from Validation], predict the calibration using [Assess Model/Predict Calibration], and check out the error statistics. Note that the RMS of the predicted residual errors for the DUT is 0.039039.

Try 120 test points. The RMS, Max, and Min are now 0.037744, 0.52174, and -0.20067, respectively. Look at the residual errors for the validation set using [Plot/Validation Analysis/...]. Create reduced measurement data from the first column of the validation set using the menu item [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Enter "1" in the blue text field and press the "OK" button. A prompt-window will appear to confirm the overwrite of any pre-existing Reduced Measurement Data. Press "OK" again. Perform mathematical analysis on the reduced measurement data using the menu item [Assess Model/Predict Calibration]. The RMS of the predicted residual errors for the DUT is 0.044293.

Now try using 309 test points (the complete set). The RMS, Max, and Min of the residual errors for the validation set are now 0.035487, 0.42446, and -0.20529, respectively. The residual errors continue to decrease with the addition of more test points. Now create reduced measurement data using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Note that this is theoretically not reduced measurement data since all points are included in the model. There are no savings in measurements with this model!!! The RMS of the predicted residual errors is 0.04446.

Select the entire validation set as the reduced data set by using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click in the blue area and enter "1:26" to select all validation vectors, and press the "OK" button twice. Then evaluate all the vectors by selecting [Assess Model/Predict Calibration]. Notice that the DUT Error Stat value is 0.035487, identical to the Valid. Error Stats RMS value. This should be the case since the calculation used the entire data set in both analyses.

2. Example: Modeling a 10-Bit Analog-to-Digital Converter with Empirical and Mixed Models

To start the HELP Toolbox from within the MATLAB® Command Window, type

```
>>help22
```

The High-dimensional Empirical Linear Prediction Toolbox window will appear. As you browse the Toolbox, you will see the menu headings [Data Sets], [Params and Test Pts], [Assess Model], [Quality Control], [Plot], [Help], and [Exit]. Each of these main menu items has corresponding submenus. (For the sake of clarity and consistency, all references to menu labels are enclosed by square brackets.)

To begin using the HELP Toolbox, load some previously collected data into the Toolbox. Select the menu item [Data Sets/Load Data File/Modeling Set and Validation Set].

You will be prompted to select the file format for the file you wish to enter.

Select MATLAB® Binary Format (*.mat). Within the Load Data Window that pops up, change the folder directory to C:\Help2Data\inl10. Select the filename "Adc10.mat" by single-clicking the filename and choosing the "Open" button or by double-clicking the filename.

The Toolbox is looking for *.mat files so the .mat extension will not appear within the directory menu. You will see the size of the data file displayed as 1024 rows and 89 columns. This is referred to as a matrix of size 1024×89. You will next be asked if you would like to assign some of the data for model validation.

Choose the "Yes" button. Then, within the prompt window, select the button labeled "Manually". Place your cursor in the light blue box, click the mouse, and type "[3:3:89]" (without the single quotes). Choose the "OK" button.

This tells the Toolbox to select every third vector from three through 89 from the data set and assign them to the validation set. You will see the sizes of the modeling and validation sets displayed as 1024×60 and 1024×29, respectively. Notice underneath both the modeling set sizes and validation set sizes, there are text objects labeled "Unnormalized". This informs the user that neither matrix has been normalized.

Select the menu item [Params and Test Pts/Modeling Set Decomposition].

This performs a factorization of the modeling set in order to reduce the column dimension of the model.

Next, select the menu item [Params and Test Pts/Parameter Selection Plots/Diagnostic Plots].

The plots shown within the Toolbox give graphical information helpful in determining the desired number of parameters for the model. Figure 5.7 shows the Toolbox window containing these plots. All three diagnostic plots are in agreement that about 8 to 12 parameters should model this data. Intuition might tell us that we need at least 10 parameters. Try 8 first.

Select the menu item [Params and Test Pts/Select Number of Parameters]. An input box will appear. Click the mouse in the blue text field of the input box, type "8", and press the "OK" button.

Notice on the front panel that a full model of size 1024×8 has been created.

Next, select the menu item [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. An input box containing a slider will appear. Enter the value "40" into the blue area showing the slider value or move the slider until "40" appears, and press the "OK" button.

You have just selected 40 test points to go along with the 8 parameters, for a 40×8 Reduced Model.

Next validate the model by selecting the menu item [Assess Model/Validate Model].

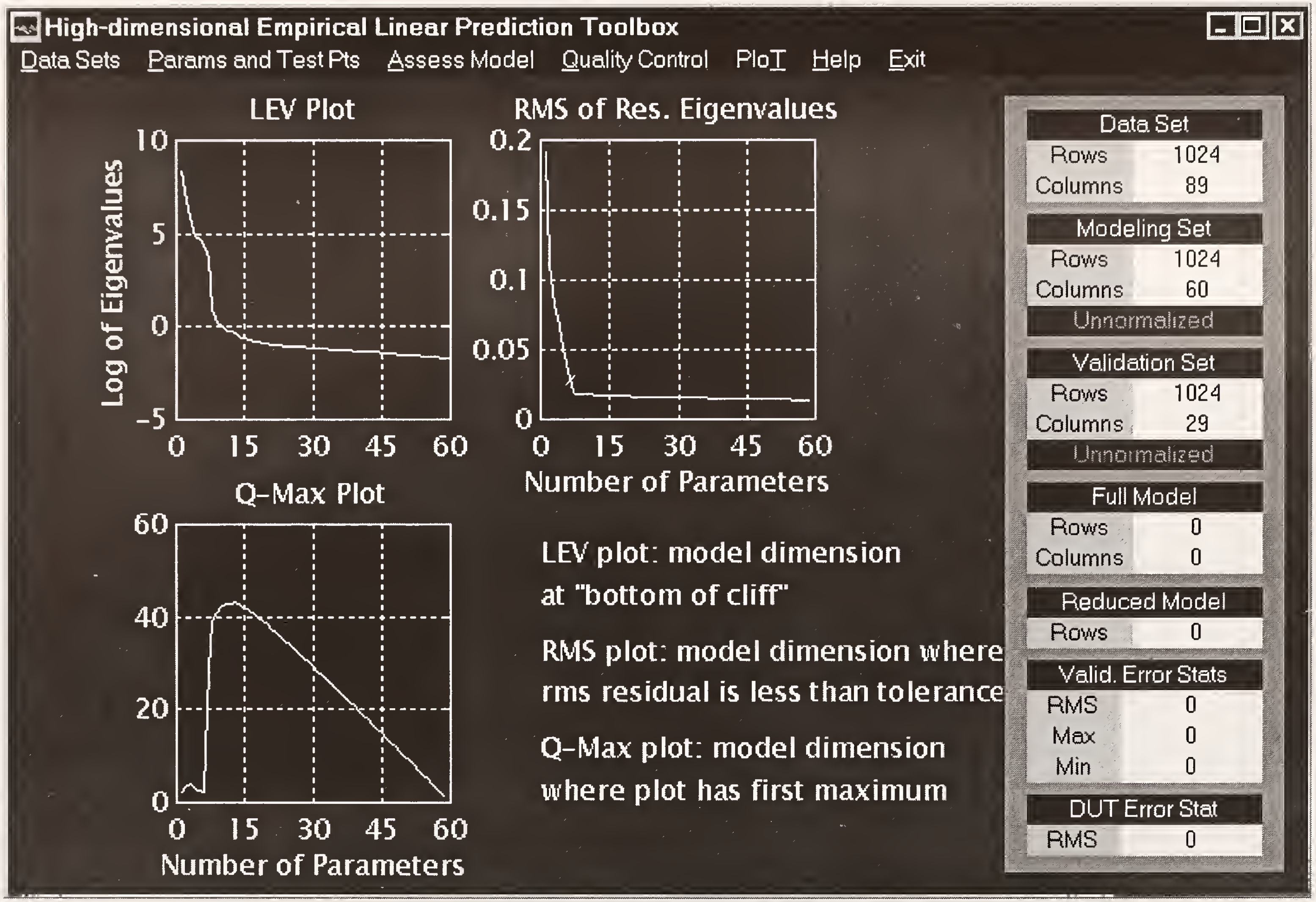
Notice that the Validation Error Statistics are filled in on the front panel. The RMS, Max, and Min for the residual errors are 0.021283, 0.096661, and -0.083969, respectively.

Now, look at the true maximum, true minimum, absolute maximum, and rms of the residual errors from the validation set using the menu item [Plot/Validation Analysis/Residual Error Statistics/...]. Try holding on the plot axis and plotting all these plots on top of each other for comparison and contrast by selecting [Plot/Hold Plot Axes/Hold On] after the first plot. Remember to select [Plot/Hold Plot Axes/Hold Off] before trying a new plot.

Figure 5.8 shows the Toolbox window containing these plots. Compare the statistics contained within the front panel of the Toolbox window with the values plotted.

Take a look at the measurement (validation) and prediction on the same plot. Select [Plot/Measurement Vector/Validation Set]. Select the "Unnormalized" button then click the mouse in the blue text area, enter a "1", and select the "OK" button. Select [Plot/Hold Plot Axes/Hold On] to fix the plot axes. Select [Plot/Validation Analysis/Response Predictions], select the "Unnormalized" button, click the mouse in the blue text area, enter a "1", and select the "OK" button. To add the residual errors to the same plot, select [Plot/Validation Analysis/Residual Error Statistics/Residual Errors], select the "Unnormalized" button, click in the blue text area, enter a "1", and press "OK". Remember to select [Plot/Hold Plot Axes/Hold Off] prior to creating any new plots.

You are looking at the measurements and predictions for the first validation vector based on a model of size 40×8. If the user does not selects the "Unnormalized" button when plotting any of the parameters, the Toolbox will inform the user that the data has not been normalized and that unnormalized data will be plotted.



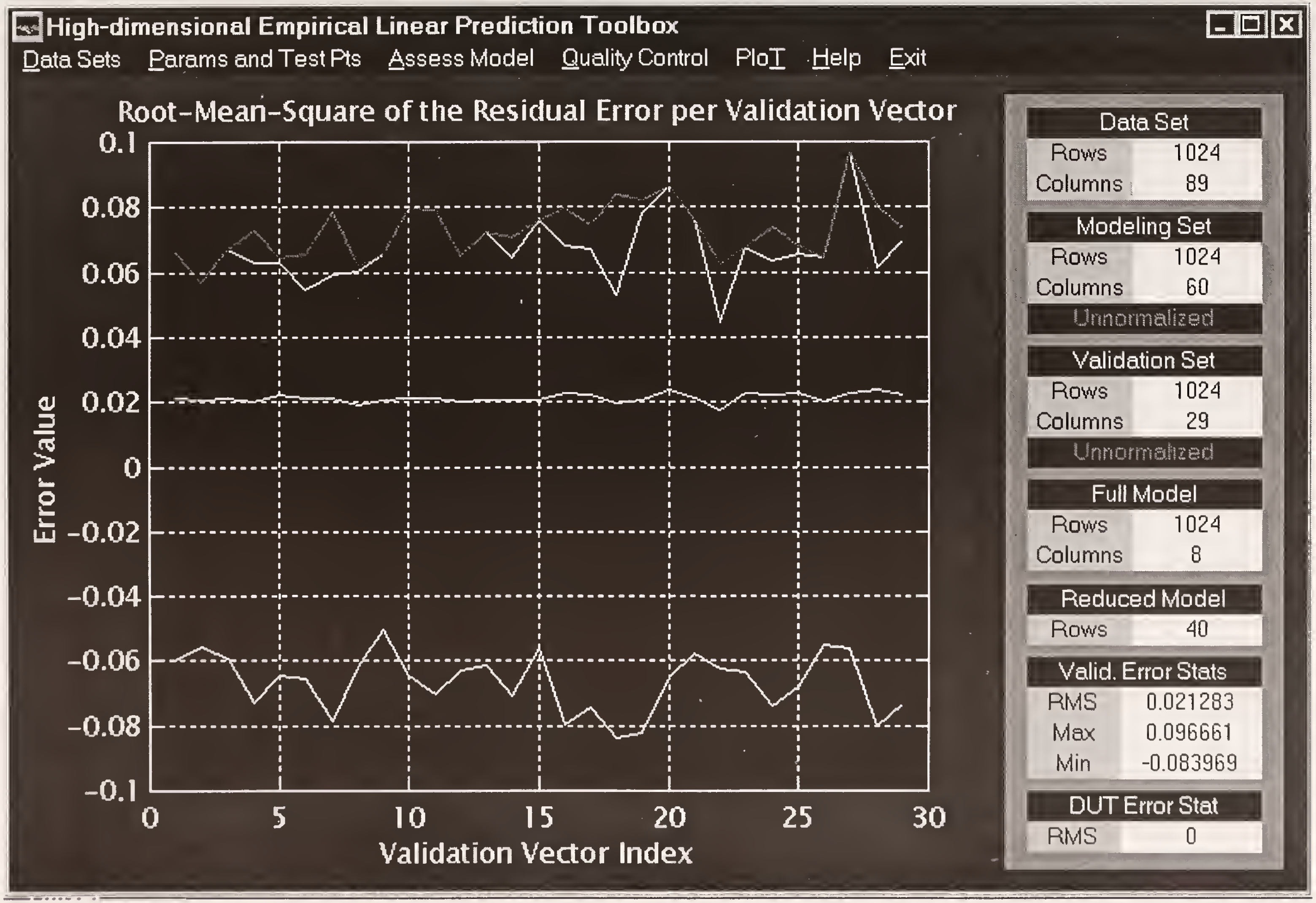
LEV plot: model dimension at "bottom of cliff"

RMS plot: model dimension where rms residual is less than tolerance

Q-Max plot: model dimension where plot has first maximum

Data Set	
Rows	1024
Columns	89
Modeling Set	
Rows	1024
Columns	60
Unnormalized	
Validation Set	
Rows	1024
Columns	29
Unnormalized	
Full Model	
Rows	0
Columns	0
Reduced Model	
Rows	0
Valid. Error Stats	
RMS	0
Max	0
Min	0
DUT Error Stat	
RMS	0

Figure 5.7. Diagnostic Plots Showing Parameter Information.



Data Set	
Rows	1024
Columns	89

Modeling Set	
Rows	1024
Columns	60
Unnormalized	

Validation Set	
Rows	1024
Columns	29
Unnormalized	

Full Model	
Rows	1024
Columns	8

Reduced Model	
Rows	40

Valid. Error Stats	
RMS	0.021283
Max	0.096661
Min	-0.083969

DUT Error Stat	
RMS	0

Figure 5.8. Max, Min, Absolute Max, and RMS of Residual Error.

Next, create reduced measurement data using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click the mouse in the blue text field, enter "1", and press the "OK" button. Again, press the "OK" button, this time to allow overwrite of data (no data presently exists to overwrite). Predict the measurement response using [Assess Model/Predict Calibration]. Take a look at the measurements by selecting [Plot/Measurement Vectors/Reduced Measurement Data]. Click the mouse in the blue text field, type "1" (there is only one reduced measurement vector), and press the "OK" button. Hold the plot axis on using [Plot/Hold Plot Axes/Hold On]. Plot the predicted response using [Plot/DUT Analysis/Response Predictions]. Click on the blue text field, enter "1", and press "OK". Plot the predicted residual errors produced from applying the model to the reduced measurement data on the same axes with the menu item [Plot/DUT Analysis/Residual Error Statistics/Predicted Residual Errors]. Again, click on the blue text field, enter "1", and hit "OK". Remember to select [Plot/Hold Plot Axes/Hold Off] prior to the next plot.

> The light blue circles are the actual measurements at the 40 selected test points. The red curve contains the predicted responses at all 1024 points. The light blue circles are the predicted residual errors at the 40 selected points. Figure 5.9 shows the Toolbox window containing these plots. Now go back and create a model of size 40×10 and compare the residual statistics.

Select [Params and Test Pts/Select Number of Parameters]. Enter "10" in the blue area. Be sure to check the "Replace" button within the parameter selection box so that instead of appending vectors to the present model, a new model is created. Select 40 test points for the model using [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. Enter 40 in the blue text field and hit "OK". Next, validate the model using [Assess Model/Validate Model].

> Note that the RMS, Max, and Min for the residual errors are 0.02095, 0.082252, and -0.084122, respectively. The RMS and Max values decreased but the Min actually became larger in absolute terms.

Again, try plotting the residual errors in all forms using the menu item [Plot/Validation Analysis/Residual Error Statistics/…].

Next, create reduced measurement data using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click the mouse in the blue text field, enter "1", and select the "OK" button. Another window will appear that asks the user to verify the overwrite of any existing reduced measurement data. Select the "OK" button. Predict the measurement response using [Assess Model/Predict Calibration].

> The RMS value of the predicted errors is displayed on the front panel as 0.015105.

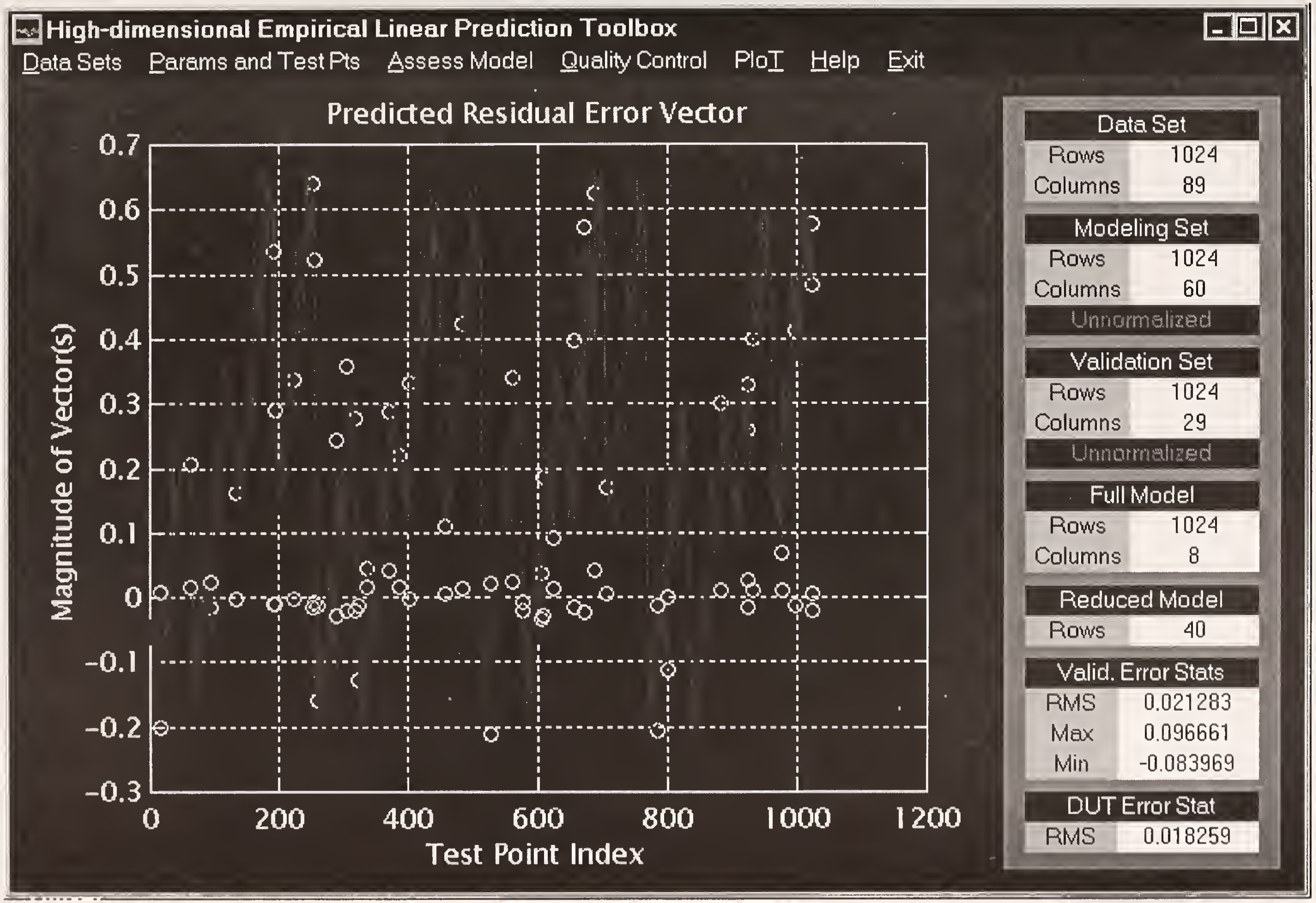
Repeat the procedure with 12 model parameters and 40 test points. Increasing the model size to 40×12 produces RMS, Max, and Min of 0.021253, 0.080644, and -0.086893, respectively.

> The RMS error value has increased in this case so that more model vectors is not better. The model is now including more noise per parameter vector than true information.

Create a reduced measurement vector using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click the mouse in the blue text field, enter "1", and select the "OK" button. Next check out the error statistics. Note that the RMS of the predicted residual errors for the DUT is 0.015547, which is comparable to the RMS of the error for the validation set.

> Now that several empirical models have been built and tested, we shall build and test a mixed model. The normal empirical modeling procedure takes many empirical vectors and creates a model of linear combinations of all or nearly all of the vectors in the modeling set. This approach cannot be used if the user wants to evaluate a particular vector corresponding to a physical or *a priori* characteristic of a device. In this case, the physical or *a priori* vector or vectors of interest are assigned to the full model. The same modeling set is then used to augment the full model according to the normal empirical modeling procedure.

Load a full model (previously created and located on the computer disk) into the Toolbox by selecting [Data Sets/Load Data File/Full Model]. Press the button labeled "MATLAB Binary Format (*.mat)". Change the folder



Data Set	
Rows	1024
Columns	89

Modeling Set	
Rows	1024
Columns	60
Unnormalized	

Validation Set	
Rows	1024
Columns	29
Unnormalized	

Full Model	
Rows	1024
Columns	8

Reduced Model	
Rows	40

Valid. Error Stats	
RMS	0.021283
Max	0.096661
Min	-0.083969

DUT Error Stat	
RMS	0.018259

Figure 5.9. Max, Min, Absolute Max, and RMS of Residual Error.

directory to C:\HelpData\Inl10, choose the file named "rad10.mat", and press the "OK" button. (Note that the .mat extension does not appear in the directory menu.)

The front panel now displays the full model size as 1024×10.

Plot the full model using [Plot/Measurement Vectors/Model Vectors]. Click on the blue area in the window that appears and enter [1:10], press the "Vertical" button and the "Unnormalized" button and then press the "OK" button. Do not plot the test points on this plot (they do not correspond to the new model). You will see the first 10 vectors from the set referred to as Rademacher vectors. Notice that the last few vectors are difficult to view at the current axis resolution. To get a better view of the last few vectors select [Plot/Control Plot Axes], change the x-axis range to [0 16] and the y-axis range to [0 200] in the window that appears, and press the "Apply" button and then the "Close" button.

Figure 5.10 shows the model vectors as plotted in the Toolbox window. The set of Rademacher vectors is a set of orthonormal vectors that characterize binary behavior. We are going to orthogonalize the (empirical) modeling set to the Rademacher vectors in the full model and augment the model.

Select the menu item [Data Sets/Orthogonalize Modeling Set]. Hit the "OK" button in the window that appears.

The modeling set has now been orthogonalized to the full model. This changes the modeling set! If the user desires to perform any additional mixed modeling, the full model must be reloaded into the Toolbox. If the user wants to perform additional empirical modeling, the modeling set must be reloaded and all HELP steps must be followed.

Select [Params and Test Pts/Modeling Set Decomposition] and then [Params and Test Pts/Select Number of Parameters]. Click in the blue area, enter "6", (make sure the "Append" button is selected,) and press the "OK" button.

This sequence adds six linear combinations of empirical vectors to the Rademacher vectors to form a mixed model of size 1024×16.

Select 40 test points for the model using the menu item [Params and Test Pts/Test Point Selection/Prediction Variance Optimization]. Enter 40 into the blue text field and press the "OK" button.

The front panel in the Toolbox window displays the full model size as 1024 rows and 16 columns and the number of reduced rows as 40.

Validate the model using [Assess Model/Validate Model].

Note the error statistics that are displayed in the front panel. The RMS, Max, and Min of the residual errors are 0.02095, 0.085391, and -0.083079, respectively.

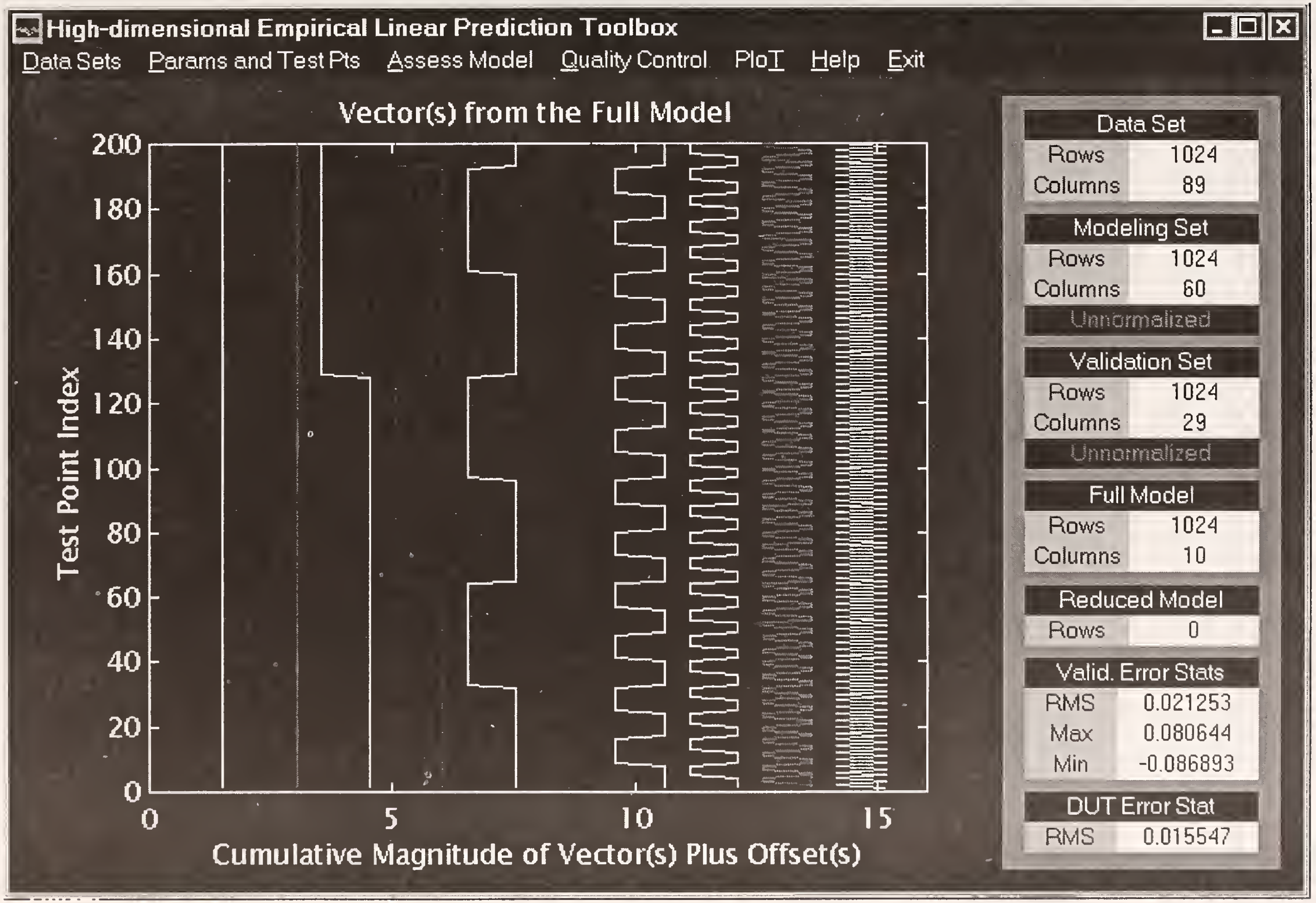
Look at the residual error plots using [Plot/Validation Analysis/Residual Error Statistics/…].

Create reduced measurement data using [Data Sets/Load Data File/Extract Reduced Meas. from Validation]. Click in the blue text field, enter "1", and hit the "OK" button. Another window will appear asking the user to verify the overwrite of any existing reduced measurement data. Hit the "OK" button. Predict the measurement response using the menu item [Assess Model/Predict Calibration].

The front panel displays the RMS of the predicted residual error as 0.015228.

Plot the reduced measurement data using [Plot/Measurement Vectors/Reduced Measurement Data]. Hold the plot axes using [Plot/Hold Plot Axes/Hold On]. Plot the predicted response using [Plot/Analysis of DUT/Response Predictions]. Click on the blue area, type "1", and press the "OK" button. Use [Plot/Control Plot Axes] to get a better look at how the predictions line up with the measurements.

Construction and testing of several model sizes is recommended in order to evaluate the decreased errors against the increased measurement costs as more test points are required. Table 5.1 shows the RMS, Max, and Min residual error sizes for the various models that have been constructed thus far as well as a 40×13 mixed model created with the same ten Rademacher vectors fixed in the full model. The user must reload


High-dimensional Empirical Linear Prediction Toolbox
Data Sets
Params and Test Pts
Assess Model
Quality Control
PloT
Help
Exit
Vector(s) from the Full Model
Test Point Index
200
180
160
140
120
100
80
60
40
20
0
0
5
10
15
Cumulative Magnitude of Vector(s) Plus Offset(s)

Data Set
Rows 1024
Columns 89

Modeling Set
Rows 1024
Columns 60
Unnormalized

Validation Set
Rows 1024
Columns 29
Unnormalized

Full Model
Rows 1024
Columns 10

Reduced Model
Rows 0

Valid. Error Stats
RMS 0.021253
Max 0.080644
Min -0.086893

DUT Error Stat
RMS 0.015547

Figure 5.10. Model Vectors (Rademacher Vectors).

the full model to produce this second mixed model. However, the user must NOT reselect [Data Sets/Orthogonalize Modeling Set]. Selecting this menu item changes the modeling set prior to construction of the model, and the same model is not produced as if the user had started from scratch and produced a 40×13 mixed model. Rather, the user should proceed to the menu item [Params and Test Pts/Modeling Set Decomposition]. If the sequence is confusing, reload all data sets and begin again. Note that the 40×13 mixed model produces much smaller errors than the 40×16 mixed model. This is because most of the information contained in the empirical data set is contained within the set of ten Rademacher vectors and going beyond three additional empirical vectors adds predominantly noise to the model.

Statistics	Empirical Models			Mixed Models	
Model Size	40×8	40×10	40×12	40×16	40×13
Val RMS	0.021283	0.020950	0.021253	0.020950	0.020277
Val Max	0.096661	0.082252	0.080644	0.085391	0.092211
Val Min	-0.083969	-0.084122	-0.086893	-0.083079	-0.094547
DUT RMS		0.015105	0.015547	0.015228	0.013931

Table 5.1 Error Statistics for Several Models

The Toolbox contains additional functions helpful in determining the appropriateness of a particular model. The [Quality Control] menu heading allows the user to compute individual and simultaneous prediction intervals for either the validation set or, more importantly, the device under test. The Toolbox computes 2-sigma prediction (uncertainty) intervals for all points that are predicted.

VI. References

1. T. W. Anderson, Estimating Linear Statistical Relationships, Ann. Stat. 12, pp. 1-45, 1984.

2. G. H. Golub, C. F. Van Loan, Matrix Computations, 2nd edition, Johns Hopkins University Press, Baltimore, 1989.

3. J. E. Jackson, A User's Guide to Principal Components, John Wiley & Sons, New York, 1991.

4. G. N. Stenbakken, "Effects of Nonmodel Errors on Model-Based Testing," IEEE Trans. Instrum. Meas., vol. 45, April 1996.

5. G. E. P. Box, W.G. Hunter, J.S. Hunter, Statistics for Experimenters; An Introduction to Design, Data Analysis and Model Building, John Wiley & Sons, New York, 1978.

6. Barry N.Taylor and Chris E. Kuyatt, "Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results," NIST Technical Note 1297, 1994.

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published bimonthly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program in support of the efforts of private-sector standardizing organizations.

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (**FIPS PUB**)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency or Internal Reports (**NISTIR**)—The series includes interim or final reports on work performed by NIST for outside sponsors (both government and nongovernment). In general, initial distribution is handled by the sponsor; public distribution is handled by sales through the National Technical Information Service, Springfield, VA 22161, in hard copy, electronic media, or microfiche form. NISTIR's may also report results of NIST projects of transitory or limited interest, including those that will be published subsequently in more comprehensive form.

U.S. Department of Commerce
National Institute of Standards
and Technology
Gaithersburg, MD 20899–0001

Official Business
Penalty for Private Use $300