Integration of Multiple Tools for Efficient Test Data Manipulation

Ralph D. Brillhart, ATA Engineering, Inc., San Diego, California

There are many hardware and software tools available to the test engineer. Many of these provide coverage for a broad scope of test applications. Nonetheless, there is no single tool available today that provides all of the capabilities needed for all aspects of every test. Test engineers must be resourceful and integrate appropriate tools to best provide efficient solutions. Examples of how a variety of test tools can be combined to provide more powerful capabilities are described. Suppliers to test engineers are encouraged to provide more open and versatile systems to foster future growth.

There are many software tools that can be used to define and measure a wide variety of test data. Sometimes these software tools can encompass most of the capabilities required for an entire test program, from planning to end results. It is seldom, however, that a single software package provides all of the capabilities required to complete an entire test program. As a result, there may be duplication of effort as data are entered in multiple locations and test data are passed from one software application to another. Inefficiencies and errors can result as information is transferred. The test engineer needs to find methods that allow minimal effort in integrating the various tools available while providing high-quality test results.

This article presents some approaches for integrating a wide variety of software packages to allow more effective testing activities to take place. Many of the software packages in use are commercial products. In some cases, special applications have been developed to allow commercial packages to work together more efficiently. Adding these special applications to the test engineer's tool kit can substantially improve test data quality and efficiency in test conduct. Specific software utilized and described in this article includes Microsoft Excel®, MATLAB®, Labview®, MTS I-DEAS Pro®, and some special applications developed to integrate information from these software applications. The key to effective integration is the capability of software packages to share data. The integration of these various software products is intended to take advantage of each of their individual strengths so that the whole is greater than the sum of the parts.

Test Preparations

In preparation for a test, the test engineer may perform several tasks to ensure that data are collected efficiently and accurately. In addition to typical test procedure development, test planning often includes identification of the types of instrumentation to be employed and locations where measurements are to be made. Adjustments during the test may need to be made but early identification of key parameters simplifies the setup process.

One tool available to the test engineer is the electronic spreadsheet. Various forms of these readily available tools have been used for years in conjunction with many data acquisition packages. The spreadsheet allows a listing of the test parameters to be cataloged and reviewed prior to test start. In some cases, specific transducer and sensitivity information can be incorporated in the spreadsheet and used leading up to the start of data collection.

Calibration data can be identified well in advance of a test. This is typically required for many test programs and efficient methods of cataloging and utilizing such data during the test

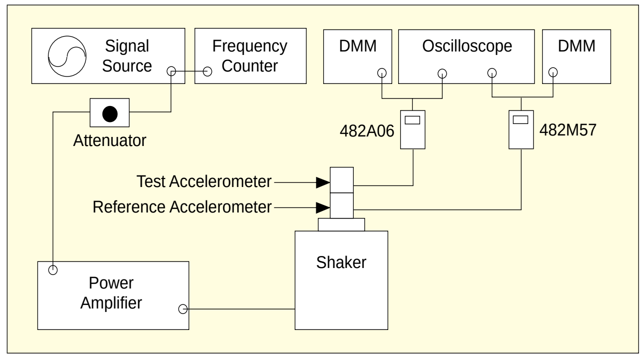


Figure 1. The calibration process typically involves the use of several instruments to measure and capture required response information.

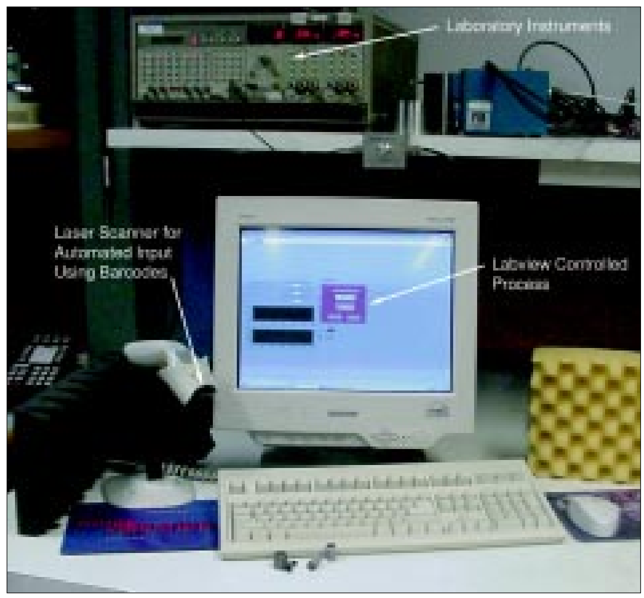


Figure 2. Labview provides an efficient software tool for performing instrumentation calibrations by controlling laboratory instruments. It can also include a bar code scanner for automating data entry.

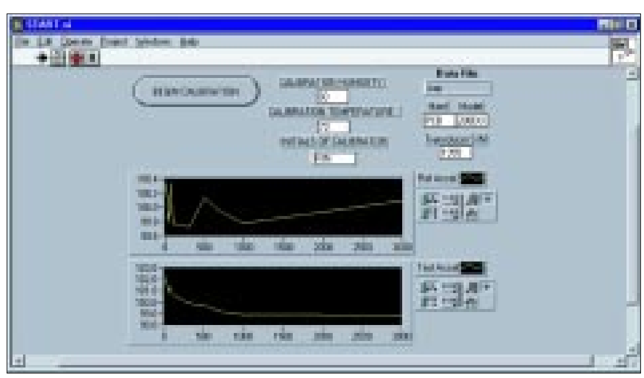
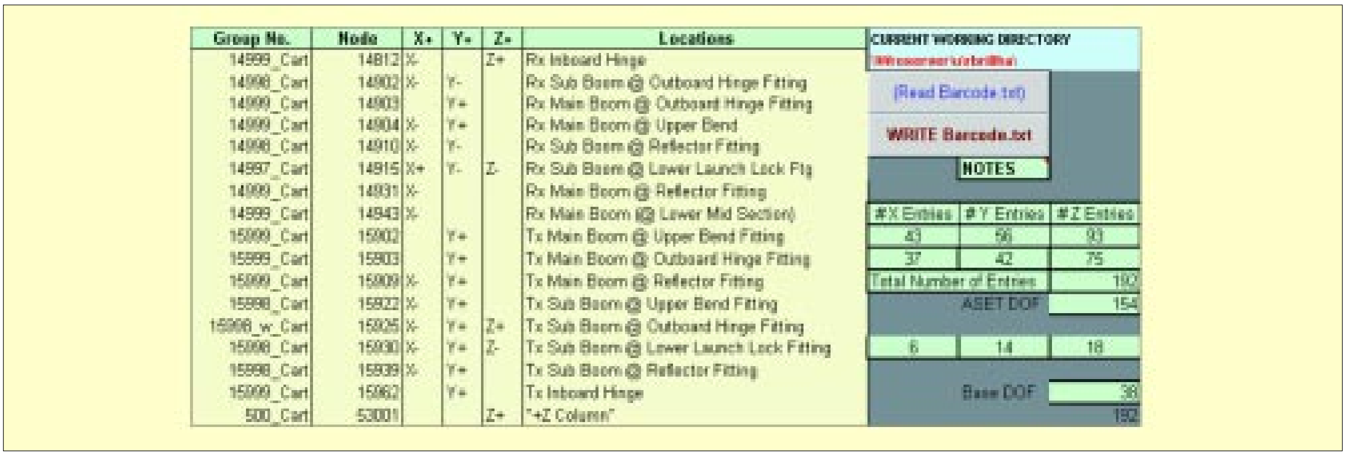


Figure 3. The Labview interface allows easy data entry and review of calibration data.

program can be critical. Excel is one spreadsheet program in widespread use that is very effective in cataloging data. However, it is not well suited to operating, controlling and collecting data from calibration equipment. Labview, on the other hand, is well suited to implementation in a calibration envi-

Based on a paper presented at IMAC-XX, the 20th International Modal Analysis Conference, Los Angeles, CA, February 2002.



Group No.	Node	X+	Y+	Z+	Locations
14999_Cart	14812	X-		Z+	Rx Inboard Hinge
14998_Cart	14902	X-	Y-		Rx Sub Boom @ Outboard Hinge Fitting
14999_Cart	14903		Y+		Rx Main Boom @ Outboard Hinge Fitting
14999_Cart	14904	X-	Y+		Rx Main Boom @ Upper Bend
14998_Cart	14910	X-	Y-		Rx Sub Boom @ Reflector Fitting
14997_Cart	14915	X+	Y-	Z-	Rx Sub Boom @ Lower Launch Lock Ftg
14999_Cart	14931	X-			Rx Main Boom @ Reflector Fitting
14999_Cart	14943	X-			Rx Main Boom (@ Lower Mid Section)
15999_Cart	15902		Y+		Tx Main Boom @ Upper Bend Fitting
15999_Cart	15903		Y+		Tx Main Boom @ Outboard Hinge Fitting
15999_Cart	15909	X-	Y+		Tx Main Boom @ Reflector Fitting
15998_Cart	15922	X-	Y+		Tx Sub Boom @ Upper Bend Fitting
15998_w_Cart	15925	X-	Y+	Z+	Tx Sub Boom @ Outboard Hinge Fitting
15998_Cart	15930	X-	Y+	Z-	Tx Sub Boom @ Lower Launch Lock Fitting
15998_Cart	15939	X-	Y+		Tx Sub Boom @ Reflector Fitting
15999_Cart	15962		Y+		Tx Inboard Hinge
500_Cart	53001			Z+	"+Z Column"

Figure 4. Test planning can be effectively performed using a spreadsheet program such as Excel where color highlights allow for easy reading and sorting of data.

ronment.

A typical setup for calibration of accelerometers or load cells is shown in Figure 1. The equipment shown can be computer controlled with an automated calibration process. Figure 2 shows calibration hardware where Labview is being used to perform calibration of accelerometers by controlling a set of instruments including a barcode scanning input device. A screen capture of the Labview user interface developed for the automated sequence is shown in Figure 3. The use of Labview for automating the control of calibration processes requires special scripting to be developed, as is the case for most Labview applications. Once this special application is developed, it can be used for full automation of the calibration procedure, helping reduce errors in this process. The Labview implementation allows calibration data to be output to Excel, which is more effective in allowing many users to access and use the information. This data base can be placed on a company internal website so that the most current calibration data are always available to all users regardless of location. The spreadsheet storage allows easy data transport at the start of any test program and also allows calibration history and status to be checked at any time.

In addition to calibration-type data, the test engineer can perform a substantial amount of test preparation prior to installing any instrumentation. In operational tests, this might include tape recorder channel information, as well as expected input ranges. In modal testing, this might include identification of the measurement locations and descriptions to be used in the test. Once again, Excel provides an efficient platform for cataloging and checking this information. Use of formatting and color-coding allow data to be easily reviewed and verified (Figure 4). The key to taking advantage of this efficiency is having the ability to transfer this test preparation information to the data collection system in use. Implementation of specialized tools such as Automated Test Setup (ATS)[1,2] allow the planning information to be transferred directly to the test program. Barcodes are generated to indicate measurement locations and directions and to provide all data configuration information to the data collection system. This information can be integrated with the calibration data previously defined. This approach also provides the data in a format that is easily incorporated in test planning documents that may be required in preparation for the test.

Installing instrumentation is a part of the test preparation process that can also benefit from integrating a variety of capabilities of different software programs. As transducers are installed, verification of installation can be done using bar code scanning devices (Figure 5). The data are then merged with other preparation data in Excel. Implementation of smart transducers with TEDS technology[3] (including software) allows actual instrumentation hookups to be extracted and merged with other setup data. Final setup verification checks can then be conducted using the data stored in Excel to assure that instrumentation has been properly connected. Visual feedback can again be provided as shown in Figure 6 where ATS (utilizing Excel) is used to consolidate all of the transducer hookup information and check for errors.

Figure 5. Barcode scanning using a hand held personal digital assistant verifies transducer installation and provides data for easy merging with test configuration files.

Collecting Data

While a spreadsheet program is quite effective at cataloging data for test measurements, it has no real capability for making the actual measurements by digitizing the data. Once again the integration of software can provide efficient, error-free transmission of information to improve the overall testing process. Since the instrumentation setup has been verified and cataloged, transfer of these data from the test-setup software (spreadsheet) to the data collection software is desired. Development of special application software allows the channel configuration data stored in Excel to be converted to a format that can be imported by test software such as MTS I-DEAS Pro.

Integration of the setup software to the measurement software is the best way to eliminate errors that can occur during manual data entry for the data collection system. This integration also involves facilitating a data transfer from one program to the other in an effective manner. This process is extremely important as the number of data channels continues to increase in most modern test systems.[4] This can be accomplished by using 'universal' files like ASCII that can be recognized by multiple programs. Implementation in a program such as ATS allows all transducer and channel configuration information to be transferred to the data collection system using Universal file format (UFF) dataset types 1807 and 1808. Complete setup in-

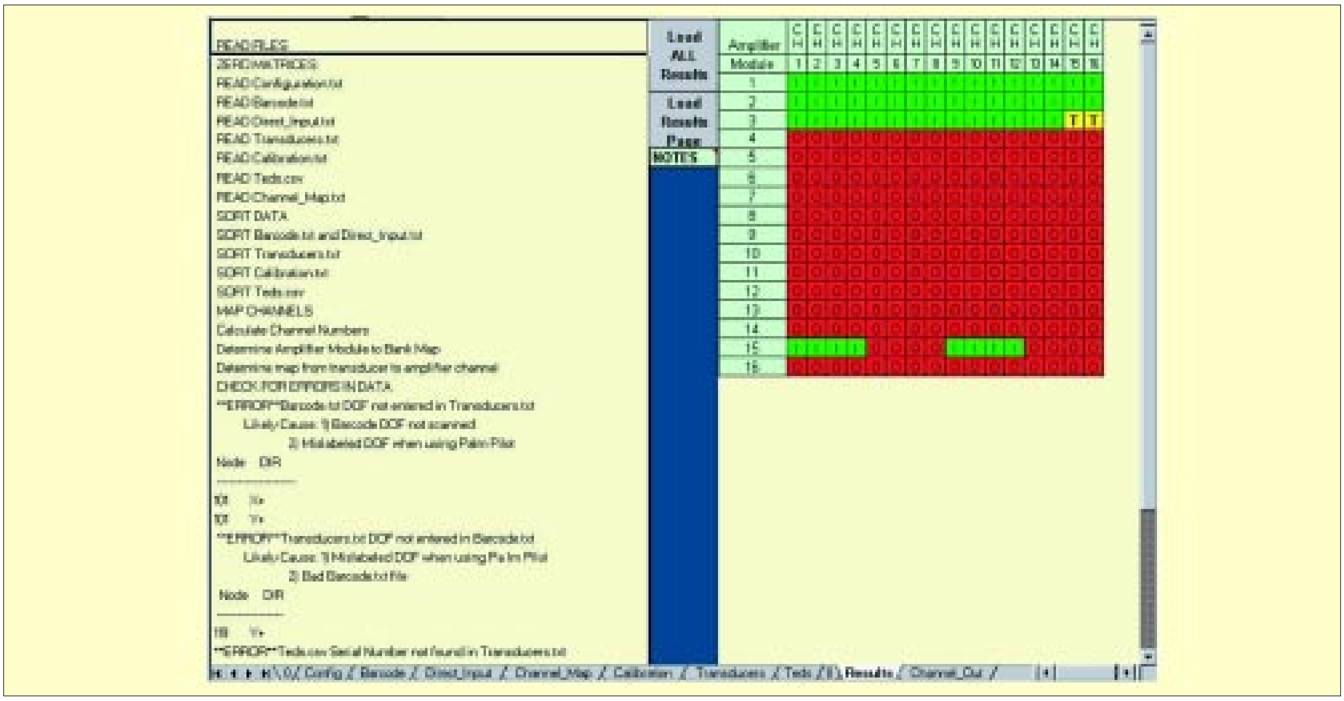


Figure 6. Visual feedback of instrumentation hookup along with potential problems can be achieved in software implementation using Excel.

formation is contained and confirmed in tabular form as shown in Figure 7, after which it is converted to UFF. The output of these data files from the Excel setup and import to MTS I-DEAS Pro allows data collection to commence as soon as the transfer is completed.

Engineering units are also handled during the transfer of data between the two software programs. This assures that the data scaling is handled properly as the measurements are made. This information is transferred from the initial transducer calibration activity so that the correct sensitivities are applied. The appropriate calibration information can be applied to each channel of data since the correct channel mapping is also transferred from the Excel spreadsheet and reflects the information obtained from the TEDS sensors and signal conditioning units.

As measurements are made, all appropriate data attributes are applied in MTS I-DEAS Pro. This step allows proper identification of the data to be presented later. This includes identification attributes such as ID lines showing where transducers were located on the structure. Anyone who has shared data by transferring Excel spreadsheet files knows that considerable data attribute information can be lost in this type of data transfer. Even something as simple as engineering units is not transferred automatically in spreadsheet file formats. These attributes are also important during subsequent data processing activities. All of this information must be carried consistently from the test setup spreadsheet to the final data.

Data Processing

Data processing can take place using the same software used for data collection. In many instances though, data will be shared among users who may each have different software for performing analysis operations. In addition, some software can perform select functions better than others. Having the ability to easily move data from one software product to another allows use of the best tool for each function.

It is possible to again export data from the data collection software into an ASCII data format, either as UFF or as comma separated data. It can also be transported in universal file binary format (UFB), which is more compact than its ASCII equivalent. The comma separated variable (CSV) and tab separated data files can be easily read into spreadsheet programs making them common methods for sharing data. However, the limitations of these formats are often too small to be of real practical use.

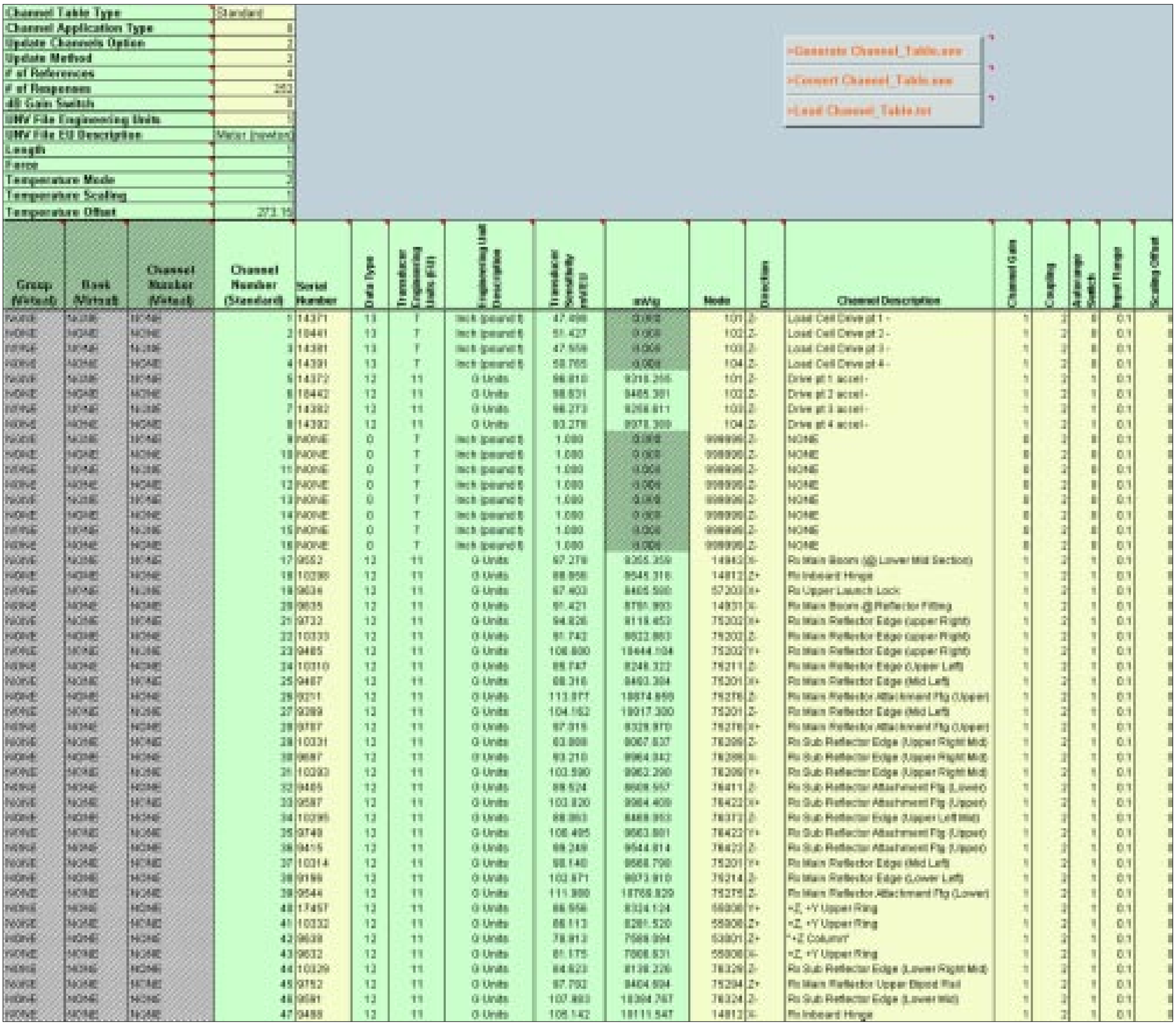
As mentioned previously, important data attributes are often lost when using spreadsheet or CSV files. This can lead to errors in computations or mislabeled data. This usually means that advanced data processing must take place in more sophisticated programs that are designed for such use. MTS I-DEAS Pro provides such capabilities in addition to data collection. MATLAB also offers a tool that is widely used in detailed data analysis. In order to accommodate the transmission of data between software packages and to allow special data-manipulation tool kits to be used in MATLAB, specialized software such as IMAT[5,6] has been developed to convert data directly between I-DEAS and MATLAB. This allows for direct data access in MATLAB as indicated in Figure 8. One of the features that evolved in MATLAB is the tracking of data attributes that are consistent with those utilized in I-DEAS. As a result, all of this information as well as the data can be accessed and used in either program during the data processing. This ensures that there is no lost information in the data transfers, allowing engineers to develop specialized tools in MATLAB to improve data analysis software capabilities.[7]

There are some instances in which data level evaluations can be performed more efficiently with Excel than with other programs. In these cases, data listed by one program can be pasted into a spreadsheet where quick evaluations can be performed. An example of this is presented in Figure 9. Acoustic data collected in I-DEAS were listed for third-octave bands and then presented in Excel where the data's relationship to the desired levels could be shown. Easy data transfer between programs also makes it simpler to document test results. Data tabulated and formatted in Excel is often easier to incorporate in a report and convey in a useful fashion. IMAT implementation allows sophisticated data analysis to be performed using MATLAB with easy transfer to other programs for final documentation.

Conclusion

There are many software tools available to the test engineer. In most cases, no one software package can provide all of the capabilities required for a complete test program. In order to maximize test performance efficiency, different software is integrated to take advantage of each program's strengths.

The desire to minimize errors, decrease test time and provide informative test results calls for effective use of each software package at the appropriate time and for making the soft-



Channel Table Type	Standard
Channel Application Type	8
Update Channels Option	2
Update Method	3
# of References	4
# of Responses	262
dB Gain Switch	0
UNV File Engineering Units	1
UNV File EU Description	Metric (newton)
Length	1
Force	1
Temperature Mode	2
Temperature Scaling	1
Temperature Offset	273.15

Group (Virtual)	Bank (Virtual)	Channel Number (Virtual)	Channel Number (Standard)	Serial Number	Data Type	Transducer Engineering Units (F/U)	Engineering Unit Description	Transducer Sensitivity mV/EU	mV/g	Node	Direction	Channel Description	Channel Gain	Coupling	Auto-range Switch	Input Range	Scaling Offset
NONE	NONE	NONE	1	14371	13	T	inch (pound f)	47.499	0.000	101	Z-	Load Cell Drive pt 1 -	1	2	0	0.1	0
NONE	NONE	NONE	2	10441	13	T	inch (pound f)	51.427	0.000	102	Z-	Load Cell Drive pt 2 -	1	2	0	0.1	0
NONE	NONE	NONE	3	14381	13	T	inch (pound f)	47.559	0.000	103	Z-	Load Cell Drive pt 3 -	1	2	0	0.1	0
NONE	NONE	NONE	4	14391	13	T	inch (pound f)	50.765	0.000	104	Z-	Load Cell Drive pt 4 -	1	2	0	0.1	0
NONE	NONE	NONE	5	14372	12	11	G Units	96.810	9316.266	101	Z-	Drive pt 1 accel -	1	2	1	0.1	0
NONE	NONE	NONE	6	10442	12	11	G Units	98.831	9465.301	102	Z-	Drive pt 2 accel -	1	2	1	0.1	0
NONE	NONE	NONE	7	14382	12	11	G Units	96.273	9266.811	103	Z-	Drive pt 3 accel -	1	2	1	0.1	0
NONE	NONE	NONE	8	14392	12	11	G Units	93.278	8978.369	104	Z-	Drive pt 4 accel -	1	2	1	0.1	0
NONE	NONE	NONE	9	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	10	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	11	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	12	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	13	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	14	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	15	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	16	NONE	0	T	inch (pound f)	1.000	0.000	999999	Z-	NONE	0	2	0	0.1	0
NONE	NONE	NONE	17	9552	12	11	G Units	97.278	9365.359	14842	X-	Rs Main Boom (@ Lower Mid Section)	1	2	1	0.1	0
NONE	NONE	NONE	18	10298	12	11	G Units	88.958	8545.316	14812	Z+	Rs Inboard Hinge	1	2	1	0.1	0
NONE	NONE	NONE	19	9634	12	11	G Units	87.403	8405.500	57203	X+	Rs Upper Launch Lock	1	2	1	0.1	0
NONE	NONE	NONE	20	9635	12	11	G Units	91.421	8791.993	14831	X-	Rs Main Boom @ Reflector Fitting	1	2	1	0.1	0
NONE	NONE	NONE	21	9732	12	11	G Units	94.826	9118.453	75202	X+	Rs Main Reflector Edge (upper Right)	1	2	1	0.1	0
NONE	NONE	NONE	22	10333	12	11	G Units	91.742	8822.863	75202	Z-	Rs Main Reflector Edge (upper Right)	1	2	1	0.1	0
NONE	NONE	NONE	23	9465	12	11	G Units	108.600	10444.104	75202	Y+	Rs Main Reflector Edge (upper Right)	1	2	1	0.1	0
NONE	NONE	NONE	24	10310	12	11	G Units	85.747	8246.322	75211	Z-	Rs Main Reflector Edge (Upper Left)	1	2	1	0.1	0
NONE	NONE	NONE	25	9487	12	11	G Units	88.316	8493.304	75201	X+	Rs Main Reflector Edge (Mid Left)	1	2	1	0.1	0
NONE	NONE	NONE	26	9211	12	11	G Units	113.877	10974.859	75276	Z-	Rs Main Reflector Attachment Ftg (Upper)	1	2	1	0.1	0
NONE	NONE	NONE	27	9289	12	11	G Units	104.162	10017.300	75201	Z-	Rs Main Reflector Edge (Mid Left)	1	2	1	0.1	0
NONE	NONE	NONE	28	9787	12	11	G Units	87.915	8329.970	75276	X+	Rs Main Reflector Attachment Ftg (Upper)	1	2	1	0.1	0
NONE	NONE	NONE	29	10331	12	11	G Units	83.909	8067.837	76209	Z-	Rs Sub Reflector Edge (Upper Right Mid)	1	2	1	0.1	0
NONE	NONE	NONE	30	9687	12	11	G Units	93.210	8964.342	76209	X-	Rs Sub Reflector Edge (Upper Right Mid)	1	2	1	0.1	0
NONE	NONE	NONE	31	10293	12	11	G Units	103.590	9962.290	76209	Y+	Rs Sub Reflector Edge (Upper Right Mid)	1	2	1	0.1	0
NONE	NONE	NONE	32	9465	12	11	G Units	89.524	8609.557	76411	Z-	Rs Sub Reflector Attachment Ftg (Lower)	1	2	1	0.1	0
NONE	NONE	NONE	33	9587	12	11	G Units	103.820	9984.409	76422	X+	Rs Sub Reflector Attachment Ftg (Upper)	1	2	1	0.1	0
NONE	NONE	NONE	34	10295	12	11	G Units	88.053	8468.053	76373	Z-	Rs Sub Reflector Edge (Upper Left Mid)	1	2	1	0.1	0
NONE	NONE	NONE	35	9749	12	11	G Units	108.495	9563.801	76422	Y+	Rs Sub Reflector Attachment Ftg (Upper)	1	2	1	0.1	0
NONE	NONE	NONE	36	9415	12	11	G Units	89.249	9544.814	76423	Z-	Rs Sub Reflector Attachment Ftg (Upper)	1	2	1	0.1	0
NONE	NONE	NONE	37	10314	12	11	G Units	98.140	9568.790	75201	Y+	Rs Main Reflector Edge (Mid Left)	1	2	1	0.1	0
NONE	NONE	NONE	38	9196	12	11	G Units	102.871	9873.910	75214	Z-	Rs Main Reflector Edge (Lower Left)	1	2	1	0.1	0
NONE	NONE	NONE	39	9544	12	11	G Units	111.899	10760.829	75275	Z-	Rs Main Reflector Attachment Ftg (Lower)	1	2	1	0.1	0
NONE	NONE	NONE	40	17457	12	11	G Units	86.556	8324.124	55606	Y+	+Z, +Y Upper Ring	1	2	1	0.1	0
NONE	NONE	NONE	41	10332	12	11	G Units	86.113	8281.520	55606	Z+	+Z, +Y Upper Ring	1	2	1	0.1	0
NONE	NONE	NONE	42	9638	12	11	G Units	78.913	7589.094	53801	Z+	"+Z Column"	1	2	1	0.1	0
NONE	NONE	NONE	43	9632	12	11	G Units	81.175	7806.831	55606	X-	+Z, +Y Upper Ring	1	2	1	0.1	0
NONE	NONE	NONE	44	10329	12	11	G Units	84.623	8138.226	76326	Z-	Rs Sub Reflector Edge (Lower Right Mid)	1	2	1	0.1	0
NONE	NONE	NONE	45	9752	12	11	G Units	87.792	8404.604	75294	Z+	Rs Main Reflector Upper Bipod Pad	1	2	1	0.1	0
NONE	NONE	NONE	46	9591	12	11	G Units	107.883	10394.767	76324	Z-	Rs Sub Reflector Edge (Lower Mid)	1	2	1	0.1	0
NONE	NONE	NONE	47	9488	12	11	G Units	106.142	10111.547	14812	X-	Rs Inboard Hinge	1	2	1	0.1	0

Figure 7. Complete channel configuration is tabulated and verified using an Excel spreadsheet.

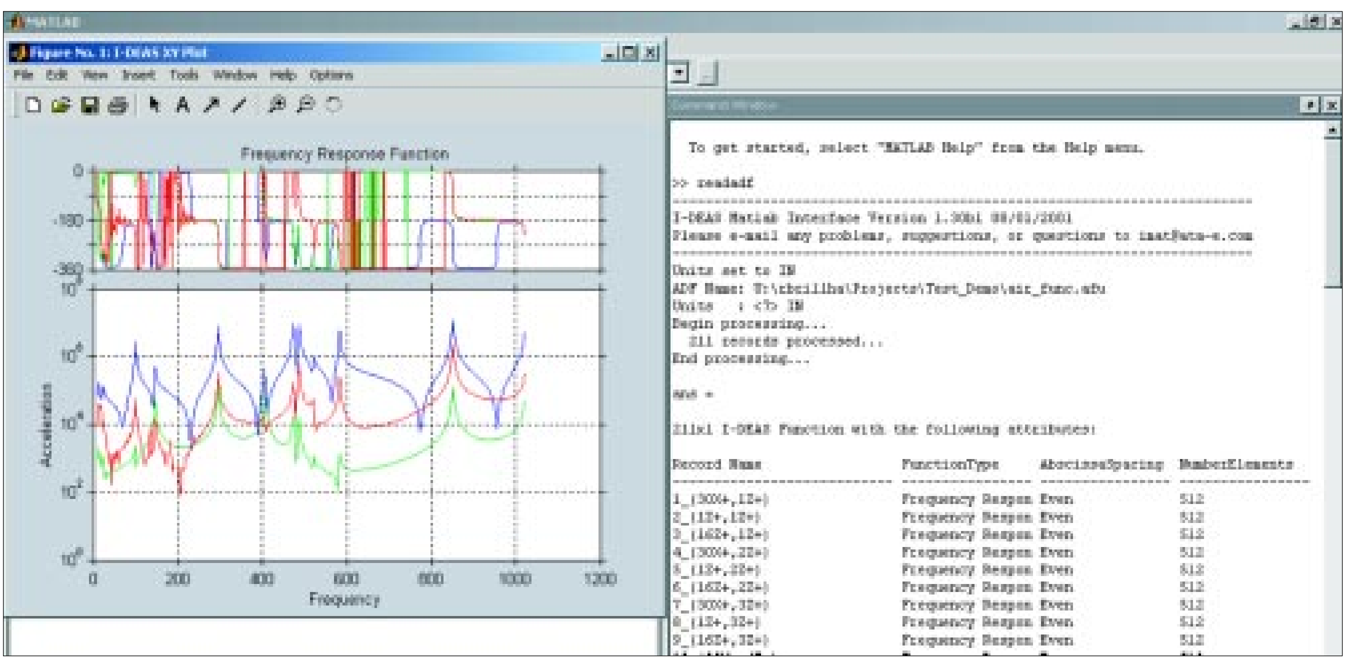


Figure 8. Efficient data conversion between I-DEAS and MATLAB allows for detailed data analysis to be performed in either software package and allows more effective data sharing.

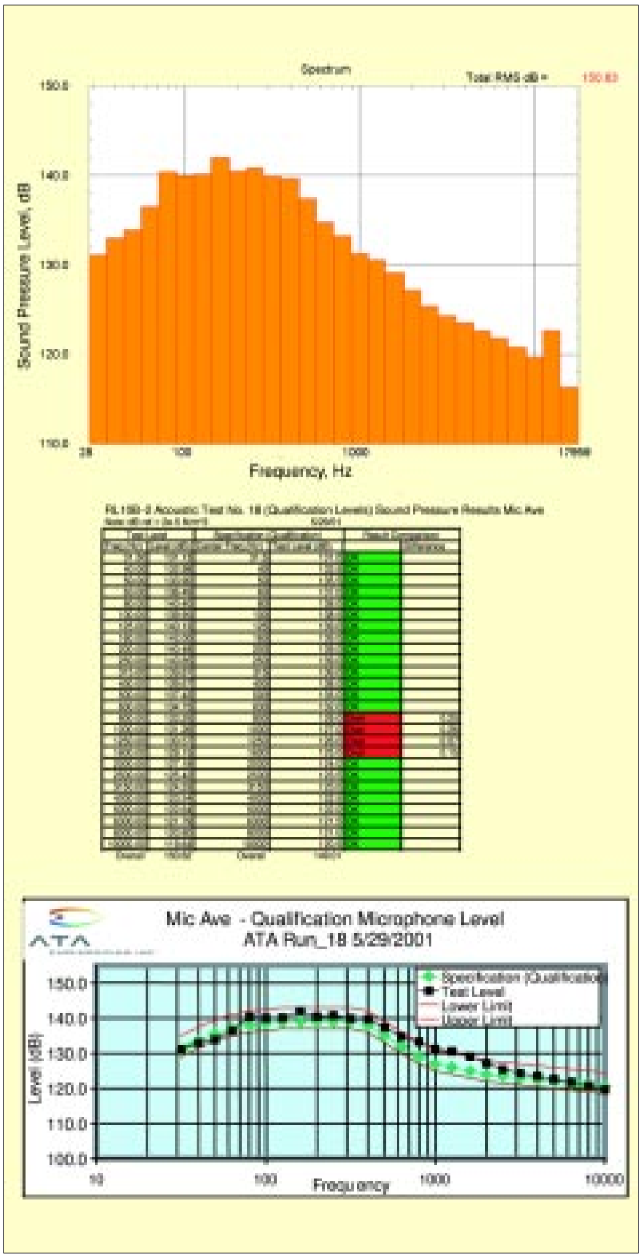


Figure 9. Data collected in I-DEAS can be listed and copied to Excel for further computation and level checking as in this example of acoustic data to check for meeting specification levels.

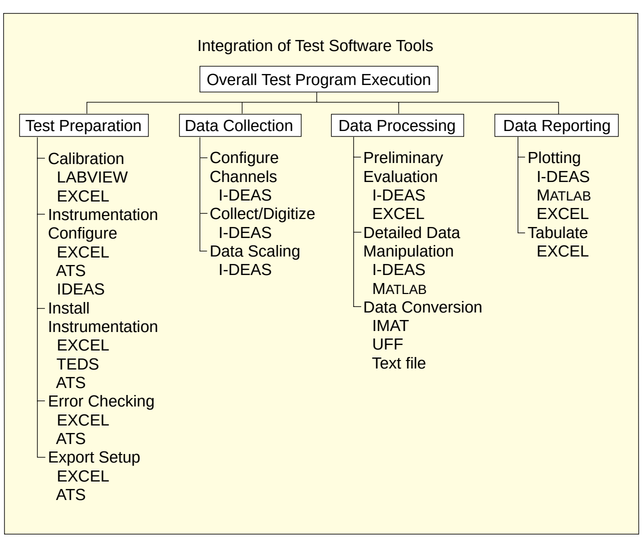


Figure 10. Steps in test program execution and potential associated software applications.

ware pieces work together. Figure 10 shows a schematic of when various software tools might be implemented in a typical test program. Some of the ways that these tools can be used together have been presented in this article. This is not a unique solution but presents some ways of solving the challenges facing the test engineer.

Hardware and software tools will continue to evolve with efforts to improve ease of use, increasing quantity of data and provide higher quality results. Many of these improvements will be based on how engineers implement today's tools while striving to make the data gathering process more efficient. As suppliers develop new capabilities for future test applications, the inclusion of open systems will make data sharing easier and will improve test performance.

References

1. Brillhart, R., Hensley, D., and Jenkins, R., "Scanning For Test Configuration Automation," 18th International Modal Analysis Conference, San Antonio, TX, February 7-10, 2000.
2. Dillon, M., and Brillhart, R., "Automated Test Setup In Modal Testing," 20th International Modal Analysis Conference, Los Angeles, CA, February 4-7, 2002.
3. Brillhart, R., "Improving Test Efficiency And Accuracy With TEDS," 20th Aerospace Testing Seminar, Manhattan Beach, CA.
4. Bono, R. W., Dillon, M. J., Gatzwiller, K. B., and Brown, D. L., "New Developments in Multichannel Test Systems," August 1999, *Sound & Vibration*, Volume 33/Number 8, 18-22.
5. Deiters, T., and Smith, K., "Faster, Easier Finite Element Model Reduction," 17th International Modal Analysis Conference, Kissimmee, FL, February 8-11, 1999.
6. Cunningham, T. J., and Hensley, D. P., "Using IMAT And MATLAB For Coriolis Flowmeter Design," 19th International Modal Analysis Conference, Kissimmee, FL, February 5-8, 2001.
7. Mayes, R. L., and Klenke, S. E., "Consolidation of Modal Parameters from Several Extraction Sets," 19th International Modal Analysis Conference, Kissimmee, FL, February 5-8, 2001. S/V

The author can be contacted at: ralph.brillhart@ata-e.com.