GODDARD SPACE FLIGHT CENTER'S STRUCTURAL DYNAMICS DATA ACQUISITION SYSTEM

Christopher McLeod
ManTech International Corporation

ABSTRACT

Turnkey Commercial Off The Shelf (COTS) data acquisition systems typically perform well and meet most of the objectives of the manufacturer. The problem is that they seldom meet most of the objectives of the end user. The analysis software, if any, is unlikely to be tailored to the end users specific application; and there is seldom the chance of incorporating preferred algorithms to solve unique problems. Purchasing a customized system allows the end user to get a system tailored to the actual application, but the cost can be prohibitive. Once the system has been accepted, future changes come with a cost and response time that's often not workable. When it came time to replace the primary digital data acquisition system used in the Goddard Space Flight Center's Structural Dynamics Test Section, the decision was made to use a combination of COTS hardware and in-house developed software. The COTS hardware used is the DataMAX II Instrumentation Recorder built by R.C. Electronics Inc. and a desktop Pentium 4 computer system. The in-house software was developed using MATLAB from The MathWorks. This paper will describe the design and development of the new data acquisition and analysis system.

INTRODUCTION

The Old System

Testing is an essential part of the life cycle in preparing experiments or payloads for spaceflight. Analyzing the results of these tests is the work of the data acquisition system and analysis software. In the early 1990's Goddard Space Flight Center's Structural Dynamics Test Section contracted a vendor to customize a multi-channel data acquisition and analysis system to acquire and process dynamic test data. This system had a modular configuration that allowed use as two independent systems or as one system with a total channel count of 192 channels. All setup, data acquisition, and processing was controlled from a workstation that issued commands over a network to the data acquisition hardware. The system worked well for four years, but when the need for improved performance became an issue it was necessary to go back to the original vendor to obtain the upgrades. While the upgrade involved more than a simple replacement of old hardware, some felt that the performance increases and time required for the upgrade did not completely justify the cost. Added to this was the fact that as users of this system we had no access to the underlying analysis algorithms. If a question arose concerning the results of processing test data, we were not in a position to explain the vendor's choice of algorithms.

As smaller data acquisition systems were purchased to satisfy other data acquisition needs none of the COTS systems that were available provided data in a format that was compatible with our existing system. None of the COTS systems provided complete or satisfactory solutions to all of the various types of data processing that were needed.

The New System

In early 2002 the decision was made to find a replacement for the nearly eight years old primary data acquisition and analysis system used by Goddard's Structural Dynamics Section. In a break from the previous practice of purchasing custom built data acquisition systems for the large channel count systems, the decision was made to develop the system in-house. The goal was to develop a system that would make use of stable easily available COTS hardware components; be highly configurable to accommodate both small and large channel count tests; allow remote setup and operation of the data acquisition hardware from a central location; use a commonly available computer operating system; and allow the use of in-house developed analysis routines to process the data. The software would be developed using an industry standard platform that was capable of advanced mathematical processing, digital signal processing, displaying data graphically, and providing the user a comprehensive Graphical User Interface (GUI). Since multiple developers would contribute to the software, the software platform would have to be conducive to a multi-developer environment. For the hardware, many currently available COTS data acquisition systems for which the file formats were documented or available would have worked for this application. We selected the DataMAX I/II Instrumentation Recorder manufactured by R.C. Electronics Inc. as the data acquisition hardware and MATLAB from The MathWorks, Inc. was selected as the software platform.

SYSTEM DESCRIPTION

Hardware Components

The DataMAX Instrumentation Recorder is available in many configurations. The DataMAX I units are small highly portable systems that can easily be carried around the facility or off site for small data acquisition requirements. The DataMAX II units are mounted in portable racks which can be rolled around the facility as needed. Each DataMAX Instrumentation Recorder has a Windows NT operating system and data files are stored in a standard Windows file system on the DataMAX. Multiple DataMAX I and DataMAX II units can be linked together to configure a data acquisition system with as many as 1620 channels.

The configuration selected for use at Goddard consists of two DataMAX I units, each capable of acquiring 32 channels of data, and two DataMAX II units, each capable of acquiring 64 channels of data. At Goddard our current setup gives us the capability to configure 192 channels for data acquisition. In this multi-unit configuration one unit acts as the master for the purpose of starting and stopping data acquisition.

For security and performance reasons, the data acquisition system operates on a private Local Area Network (LAN) that is separated from the Internet by firewall. This LAN is accessible throughout parts of the three building complex that houses the Vibration, Modal, and Acoustic test facilities. When there is a need to operate off-site or on-site in a location not serviced by the LAN, a fully functional LAN can easily be setup to operate independent of any other network system.

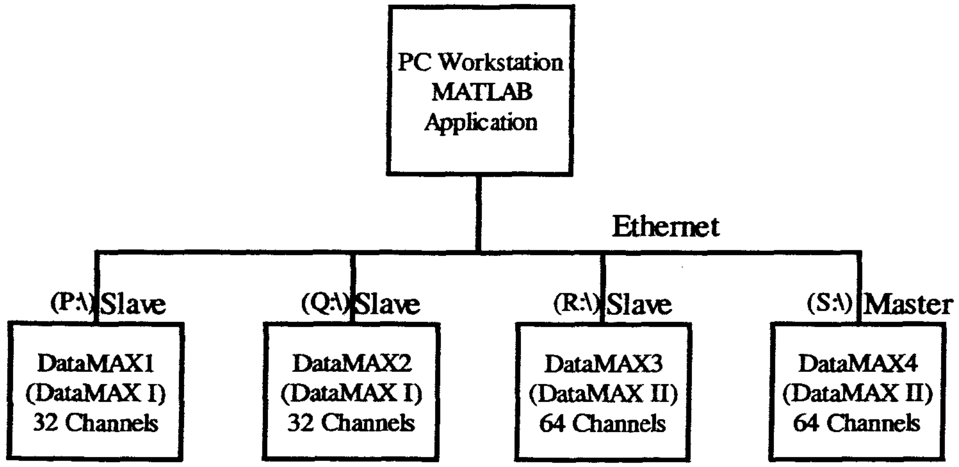

Figure 1: Block diagram of the Structural Dynamics Data Acquisition System

All of the MATLAB based analysis software resides on a standard Pentium 4 PC. This same computer system is used as the central point of control for the entire data acquisition system. Using a freely available Virtual Private Network (VPN) package, an operator sitting at this computer console can control the setup and data acquisition of each DataMAX unit as if he were at the console of each unit. Once data acquisition is complete, the data files are transferred to the PC, converted to the in-house developed MATLAB data structure and processed.

Software Components

MATLAB is a technical computing package that specializes in mathematical computation, analysis, visualization, and algorithm development. It can be used in an interactive mode in which the system responds immediately to commands entered by a user or it can be used to write and execute programs that perform more complex operations. The MathWorks provides additional software packages or Toolboxes to address specific problem areas.

The Goddard Dynamic Analysis Software (GDAS) application makes use of the MATLAB Signal Processing Toolbox, Microsoft Visual C/C++, and the programming capabilities of MATLAB. In some cases, the analysis routines that are implemented for GDAS are standard MATLAB function calls with some additional lines of MATLAB code to setup the MATLAB function calls. In other cases, the analysis routines are completely original solutions to the analysis problem. All of the functionality of the MATLAB based GDAS application is easily available to the operator via a user friendly and intuitive Goddard developed GUI.

All of the work on this application was done using MATLAB Release 13. MATLAB Release 14 is now available and is reported to offer several improvements over Release 13. This application will be ported to Release 14 in the near future.

In some cases low-level functions are better performed outside of the MATLAB environment. For this GDAS application, low level file conversions were written in C and converted to MATLAB callable routines by the MEX function. C-MEX files are dynamically linked subroutines that the MATLAB interpreter can automatically load and execute.

The DATAMAX user interface for setting up the channels is adequate for small channel count or simple configurations. The GDAS application was designed to make use of many channel related parameters that the DataMAX does not use and had no facility for entering. To get around this short coming, a format was developed that would embed additional information into the fields provided by the DataMAX user interface. The additional information gets stored in the DataMAX file and is later extracted by the MATLAB analysis software during the file conversion. A Microsoft Excel spreadsheet has been developed to aid in setting up the DataMAX for the more complex or large channel count data acquisitions. The setup of the spreadsheet can be performed on any PC and downloaded to the appropriate DataMAX system.

Application Design

The application design centers around a GUI that allows the user to import files from different data acquisition systems, open files that have already been imported, process data, save processed data, and print the results of the data processing. The GUI, as shown in Figure 2, has buttons to initiate the various types of data processing and menus to perform other tasks on the data file.

The end result of the data processing can be a set of printed plots, or if the customer chooses, the data can be provided in electronic format. The choices at this time are files in Universal File format, Microsoft Excel format or our MATLAB Project format.

The process of importing files into the GDAS application stores the data in a hierarchical Project data structure. This Project data structure has been designed to enhance the ability to keep track of channel and test parameters that aid in identifying and characterizing the data.

The following is an example of the Project structure for an X-Axis Sine Sweep. As we move down into the structure the information is more specific to the run, channel, and processing performed on the data.

Each of the analysis routines extracts the information it needs from the Project structure and writes the results of its processing to the Analysis sub-structure below the Channel sub-structure. The processed data can be recalled later for plotting or comparison to other processed results.

The Project Data Structure:

```
>> Project
         Name: 'GLAST'
    Test_Item: [1x1 struct]

>> Project.Test_Item
    TI_Name: 'LAT - ACD Structure'
         WD: 5318
        Run: [1x1 struct]

>> Project.Test_Item.Run
      Test_Date: '9-May-2004'
      Test_Type: 'Sweep'
     Test_Level: '2.5G max'
           Axis: 'X'
       Time_Ref: [1x35640 double]
       Num_Chan: 112
    Sample_Rate: 400
       Low_Pass: 80
     Setup_File: 'XSWP09'
       ADF_File: ' XSWP09.VDF'
        Channel: [1x192 struct]
        Descrip: 'Run Description'
        Run_Num: 9

>> Project.Test_Item.Run.Channel(2)
         Module: 0
        ChanNbr: 2
      Trans_Loc: 'Input Monitor'
     Coordinate: '100'
      Direction: 'X'
    Orientation: '+'
    Sensitivity: 56.44
             EU: 'G'
     Trans_Type: '8791'
      Cable_Ext: 'A006X'
     Serial_Num: '53276'
         Bypass: 'Off'
       Coupling: 'DC'
      Max_Volts: 1
         Max_EU: 5
       Time_His: [1x35640 single]
       Analysis: [1x1 struct]

>> Project.Test_Item.Run.Channel(2).Analysis
       Type: 4
        Abs: [1x229 double]
        Ord: [1x229 double]
     FPoint: 17
     LPoint: 74
    Descrip: 'Sine Mag : 5-50Hz : 4 o/m : 2.5G max'
```

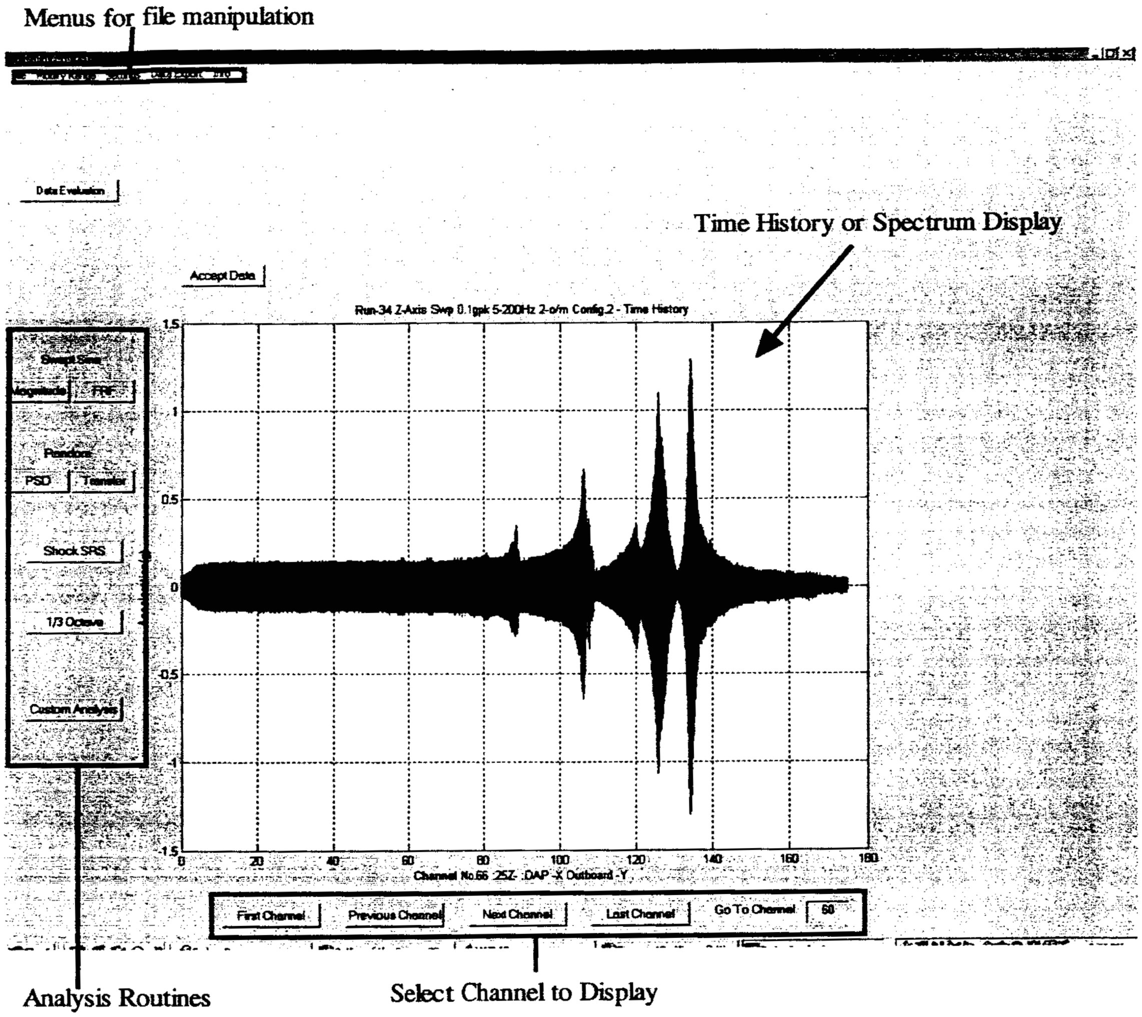

Figure 2: The GDAS Graphical User Interface

DATA ANAYSIS ROUTINES

Sine Magnitude

During a Swept Sine test a frequency reference signal is recorded, along with the response data from the test item. The accurate analysis of Swept Sine data requires an accurate determination of the frequency throughout the entire test. Various methods have been suggested to determine the instantaneous frequency of a sinusoidal signal whose frequency is constantly changing. The Hilbert transform is useful in calculating instantaneous attributes of a time series, especially the amplitude and frequency. The hilbert() function from the MATLAB Signal Processing Toolbox was used for this calculation. The results of the Sine Magnitude processing are stored in the Analysis sub-structure of the Project data structure. This method using the Hilbert transform to determine the instantaneous frequency has significant advantages over other methods that attempt to determine the instantaneous frequency.

Sine Frequency Response Function

Like the Sine Magnitude analysis for Swept Sine, the Sine Frequency Response Function uses the Hilbert transform to determine the instantaneous frequency. The user is presented with a dialog box to allow the selection of a data channel to be used as the reference channel for the Sine Frequency Response

Function. The results of Sine Frequency Response Function are stored in the Analysis sub-structure of the Project data structure.

Power Spectral Density

The pwelch() function from the MATLAB Signal Processing Toolbox is the heart of the Power Spectral Density processing routine. The in-house developed analysis routine extracts information from the Project data structure, provides dialog boxes to prompt the user, passes parameters to the pwelch function, stores the processed results in the Project data structure, and provides the necessary flow control to process every data channel.

Transfer Function

The tfe() function from the MATLAB Signal Processing Toolbox is the heart of the Frequency Response Function processing routine. Like the Power Spectral Density routine, this routine manages the data going to and from the tfe function.

Shock Response Spectra

The MATLAB functions to perform shock analysis were contributed by David O. Smallwood of Sandia National Laboratories, Albuquerque NM. Additional MATLAB code was added to prompt the user for input and to integrate the functions into the GDAS application.

One Third Octave Band Spectra

The 1/3 Octave Band analysis calculates and displays the spectrum of the power distribution of an input signal in 1/3 octave bands. The 1/3 Octave analysis routine is intended for processing acoustic data.

THE DEVELOPMENT PROCESS

The development of the MATLAB based GDAS involved a team of five part-time developers. Each of these part-time developers is an engineer who performs other functions in the day-to-day operations of the Structural Dynamics Section at the Goddard Space Flight Center.

The first phase of the development process demonstrated the feasibility of using MATLAB to develop a C-MEX function to import data from an existing data acquisition system, develop a Graphical User Interface, perform bulk processing of data, design a custom plot format, and address the issue of determining the instantaneous frequency of a changing sinusoidal signal.

The second phase of the development process involved deciding on an overall design philosophy, refining the Graphical User Interface, developing an application framework, and developing the Project data structure. With this done, each individual was assigned to develop a specific part of the application. During this phase a first pass at each of the analysis routines was completed, a new C-MEX function to import data from the DataMAX system was developed, and the software to generate printed copies of each of the various types of data processing was developed. At the end of this phase, a functioning though somewhat buggy version of the MATLAB based application was in place.

The third phase of the development process involved extensive testing, debugging, and enhancing the application. During this phase, procedures were written for the complete setup and use of the system, the system saw its first end-to-end use as a fully configured system, and was used in one of the Gamma Ray Large Area Space Telescope (GLAST) vibration tests.

There are source control systems available that should have worked with the MATLAB code, but it was thought that such a system would add an unnecessary layer to the development process. It was therefore necessary to implement a process in which each developer provided their code to the one developer who was responsible for integrating the specific analysis routines into the application framework, testing the entire application, and making the updated version available to everyone else. In hindsight, not using a formal source control system may not have been the best decision.

CONCLUSION

While work continues on the Goddard Data Analysis Software, this software development and system integration effort has been a complete success. The initial goals of developing a system to handle the entire data acquisition and processing cycle have all been met. The expensive workstations running proprietary operating systems can now be replaced with commonly available low cost desktop PC's running an operating system that nearly everyone is familiar with. The very expensive, custom built and difficult to upgrade front-end data acquisition hardware can now be replaced with easily available COTS units. The manufacturer provided custom data acquisition and analysis software that was impossible to modify can now be replaced with software that has been tailored to meet our specific requirements and can grow as our requirements change. All in all, this development effort has worked quite well.

Developing this data acquisition system in-house allowed us to avoid the "Serial Number 1" problem encountered on several previous data acquisition systems. In each of those cases the manufacturer either custom built or greatly modified an existing system to meet our specification. Once the system had been delivered, the cost and inconvenience of correcting problems on our unique system often resulted in the issue never being completely resolved. Instead, we would learn to live with the quirks while chalking it up to the cost of being at the cutting edge of data acquisition technology. In truth, many of these problems could have been resolved if we'd had in-depth knowledge of the inner workings of the systems. As the end users, and not the developers, we were in no position to probe deeply into the guts of an expensive turnkey system.

So is any widely used in-house developed system ever truly finished? Probably not. Certainly there are stages where development is frozen to provide a stable application for the end users, but one of the advantages of doing the application development in-house is the ability to grow the application as the need develops. In such a case good configuration management is crucial. In the past when customers made special data requests we were somewhat limited in what we could provide given the typically short lead times for testing. With our new ability to easily integrate new solutions into the analysis package we are limited only by our imagination and the customer's willingness to pay.