MATLAB® for Telemetry data analysis

D. Martins[1]

1: The MathWorks S.A.S., 20 rue Troyon, 92316 Sèvres Cédex, France

Abstract: Telemetry data analysis requires specific capabilities such as large data file handling, visualization and reporting. Furthermore, such a tool must be extendable to accommodate with the variety of data and situations (missiles, planes, torpedoes…). The latest evolutions of MATLAB make it a suitable environment to easily build a telemetry data handling tool.

1. Introduction

Telemetry tends to involve collecting data from test runs of missiles, planes, and torpedoes, primarily, but also other moving underwater and airborne vehicles. Usually, there is a data collection device on board the vehicle that collects many variables of data per run (could be thousands), while a separate collection process back at the site (on the "range") collects similar, parallel data for verification and comparison later on. These two sets are sometimes called "run data" and "range data", respectively. Sometimes, yet a third set of data is generated by a simulation program. After the collection is complete, the data sets are compared. For analysis purposes, discrepancies between the two or three data sets may also need to be reconciled.

The data collection process has two key purposes. The first one is the real-time monitoring of the test run. In this case, the goal of the collection is the monitoring and control of the vehicle itself, to ensure that it is performing the test as planned and that no out-of-control conditions are encountered. If they are, the test may be aborted. The second purpose of the data collection is the post-test analysis. This analysis both validates the test results and provides information that will help optimize the behavior of the vehicle in future tests or actual runs.

The bottom line is that the data is paramount. The data tells you whether the missile, torpedo, or aircraft is accurately following the instructions it received, where it is, whether you need to stop it, and whether it's safe to deploy it in non-range, true combat or reconnaissance situations.

Multiple groups in an organization may be involved in this process of collecting and analyzing the telemetry data. A typical scenario for the process flow might be: Instrumentation Group → Data Processing Group → Data Analysis Group.

Corresponding to those logical organization groups are three discrete steps:

1. Data collection and real-time monitoring
2. Data archival and conversion (to more usable forms of data for analysis)
3. Data reduction and analysis

At a high-level, the needs of the end-users performing data analysis and visualization overlap significantly with those of other MATLAB users in Aerospace/Defense, but also in Biotech, semiconductor and other areas. A key differentiator, however, is that these users tend to be more focused on developing a clear picture of the data itself and the connections within and between the runs and less focused on more advanced numeric computations that could be performed on the data. At least that is their initial goal. This means that tools to view segments of different variables or perhaps the same variables during the same 3 second time period, where, say, the missile reached its target, would be more appealing than advanced covariance algorithms that would allow the end user to statistically analyze the data.

This state of affairs has two drivers. The primary one is the difficulty of getting, viewing, and reporting on the data. Because it's so difficult and time-consuming, just accomplishing that becomes almost the exclusive focus. The second related but separate issue is the skill set of the average analyst. They are not typically experts in statistics or signal processing, nor are they necessarily even well-versed in these disciplines.

2. Requirements for a telemetry data analysis tool

Much has been written or presented in the past years with regards to the development of a modern software package for telemetry data analysis. Many organizations are evaluating MATLAB as the foundation for a potential telemetry data analysis package. While some organizations are focused on building a new, state-of-the-art software package, others are looking for tools which run on modern computing platforms. Among these organizations, a set of capabilities and features has been identified to provide the best tool for telemetry data analysis. Capabilities address what the tool allows analysts to do and features represent how users can work with it.

2.1 Key capabilities

The following *capabilities* have been identified as critical in a telemetry data analysis package.
- Compare disparate (but related) data sets, whether they are a single parameter recorded during multiple flights or runs, or multiple parameters recorded at different sample rates. The data may live in multiple files recorded in multiple formats.
- Handle asynchronously recorded data appropriately.
- Handle large (> 1GB) data sets efficiently.
- Interactive and scripting operation modes.
- Automatic reporting capabilities.
- Use automatically meta information stored along with the data. (e.g. implemented as data dictionaries which can be automatically created.)
- Perform basic data analysis (cross-channel arithmetic, spectral analysis).
- Customization to handle any data file format.*
- Open-ended extensibility.*

*It is evident that no single package will be able to provide all users with all desired functionality "out of the box." As such, it must be extensible, beyond simple scripting of repetitive operations. Extensibility may be as simple as adding a specific analysis capability, or as complex as adding appreciable new capabilities. As an example, a segment of the user community has requested multimedia functionalities. This may be critical to these users, but provide no value for others. Features such as this must be identified, prioritized, and planned for.

2.2 Favorite Features

There are many preferred features which should be strongly considered for a telemetry package.
- A point-and-click interface for data manipulation, plotting, and annotation.
- Plotting templates which are easy to create.
- Event searching capabilities.
- Additional built-in analysis capabilities.
- Improved graphics (greater choices, greater flexibility, making it easier to communicate and interpret results).
- Reduce proliferation of data formats.
- Analysis which can be extended and customized by end-users.
- The application can be customized on site, by developers more closely aligned with the final application (end users can build custom tools, local support departments can build and customize tools).

The last two points significantly leverage the large base of experienced MATLAB users within the telemetry data analysis community.

3. Capabilities of MATLAB for telemetry data analysis

Usage of MATLAB and MATLAB based tools have been increasing at a steady pace within the Aerospace and Defense industry. The last two years have shown an even steeper upward trend in this direction. Reasons for this include:

- MATLAB maturity combined with the number of years of its use in Aero/Defense
- The close mapping between the applications MathWorks[1] tools serve and the needs of Aero/Defense (Data analysis, test and measurements, signal and image processing, control design, …)
- New focuses on standardization and tools consistency within key areas of the Aero/Defense community.

Strictly speaking, The MathWorks does not offer specific tools for telemetry, in that it does not supply hardware, nor does it support telemetry-specific format files. However, it does provide tools for the post-test analysis portion of the analysis. All of the critical capabilities identified in the previous section for a Telemetry analysis tool can now be accommodated by the current MATLAB release (R14, MATLAB 7).

3.1 Data Access

MATLAB is an efficient platform for accessing data from files, other applications, databases, and external devices. Popular file formats are supported such as Microsoft Excel, ASCII text or binary files; image, sound, and video files; and scientific files, such as HDF and HDF5. Low-level binary file I/O functions allow to work with data files in any format even files greater than 2 GB (2^{31}-1 bytes).

External applications and languages, (such as C, C++, COM objects, DLLs, Java, Fortran, and Microsoft Excel) can be called from the MATLAB environment and their data directly imported.

Data acquisition from common hardware devices like computer serial port or sound card is available. Using the Data Acquisition Toolbox, live or measured data can be streamed directly into MATLAB for analysis and visualization. The Data Acquisition Toolbox provides a complete set of tools for analog input, analog output, and digital I/O from a variety of PC-compatible data acquisition hardware. The toolbox lets users configure their external hardware devices, read data into MATLAB for analysis, and send out data.

[1] The MathWorks is the world's leading developer of technical computing software for engineers and scientists in industry, government, and education. With an extensive product set based on MATLAB and Simulink, The MathWorks provides software and services to solve challenging problems and accelerate innovation in automotive, aerospace, communications, electronics, instrumentation and other industries.

Based on MATLAB, the toolbox allows easily customizing acquisitions, accessing the built-in features of hardware devices, and incorporating the analysis and visualization features of MATLAB and related toolboxes into the design.

Together, MATLAB and the Data Acquisition Toolbox offer a single, integrated environment to support the entire data acquisition and analysis process. You can easily analyze or visualize your data, save it for post-processing, and make iterative updates to test setup based on analysis results.

The Instrument Control Toolbox is useful for communicating with instruments, such as oscilloscopes, function generators, and analytical instruments, directly from MATLAB. Generated data in MATLAB are sent out to an instrument, or data can be read from MATLAB for analysis and visualization.

The toolbox provides a consistent interface to all devices independent of hardware manufacturer, protocol, or driver. The toolbox supports IVI, VXIplug&play, and MATLAB instrument drivers. Support is also provided for GPIB, VISA, TCP/IP, and UDP communication protocols.

3.2 Visualizing Data

MATLAB excels at handling most of the plotting and user interface requirements “out-of-the-box”. These include 2-D and 3-D plotting functions, 3-D volume visualization functions, tools for interactively creating plots, and the ability to export results to all popular graphics formats. Plots can be customized by adding multiple axes; changing line colors and markers; adding annotation and legends; and drawing shapes.

MATLAB provides interactive tools for designing and modifying graphics. From a MATLAB figure window, the following tasks are available:
- Drag and drop new data sets onto the figure
- Change the properties of any object on the figure
- Zoom, rotate and pan
- Add annotations and data tips
- Draw shapes
- Generate an M-code function that can be reused with different data

MATLAB lets users read and write common graphical and data file formats, such as GIF, JPEG, BMP, EPS, TIFF, PNG, HDF, AVI, and PCX. As a result, it is easy to export MATLAB plots to other applications, such as Microsoft Word and Microsoft PowerPoint, or to desktop publishing software. Before exporting, style templates can be created and applied, covering characteristics such as layout, font, and line thickness, to meet publication specifications.

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools greatly simplify the process of designing and building GUIs. Typical tasks using GUIDE tools are:

- Lay out the GUI.
Using the GUIDE Layout Editor, a GUI is designed by clicking and dragging GUI components -- such as panels, buttons, text fields, sliders, menus, and so on -- into the layout area. GUIDE stores the GUI layout in a FIG-file.
- Program the GUI.
GUIDE automatically generates an M-file that controls how the GUI operates. The M-file initializes the GUI and contains a framework for the most commonly used callbacks for each component -- the commands that execute when a user clicks a GUI component. Thanks to the M-file editor, code to the callbacks is written in order to perform the functions you want.

3.3 Performing Numeric Computation

MATLAB contains mathematical, statistical, and signal processing functions to support all common engineering and science operations needed for telemetry. These functions, developed by experts in mathematics, are the foundation of the MATLAB language. The core math functions use the LAPACK and BLAS linear algebra subroutine libraries and the FFTW Discrete Fourier Transform library. Because these processor-dependent libraries are optimized to the different platforms that MATLAB supports, they execute faster than the equivalent C or C++ code.

The Statistics Toolbox provides engineers and statisticians with a comprehensive set of tools to assess and understand their data. It includes functions and interactive tools for analyzing historical data, modeling data, simulating systems, developing statistical algorithms, and learning and teaching statistics.

The toolbox supports a wide range of tasks, from basic descriptive statistics to developing and visualizing multidimensional nonlinear models. It offers a rich set of statistical plot types and interactive graphics, such as polynomial fitting and response surface modeling.

The Curve Fitting Toolbox provides graphical user interfaces (GUIs) and command-line functions for a variety of curve-fitting applications. It includes integrated tools for previewing and preprocessing data, developing and comparing standard and custom models, fitting with standard and robust methods, and analyzing fits. Among others following functions are available:
- Interactive graphical user interface that unifies key curve-fitting tasks:
- Preprocessing routines, including data scaling, sectioning, smoothing, and removal of outliers,
- Extensive library of linear and nonlinear parametric fitting models, with optimized starting points and solver parameters for nonlinear models,

- Varied linear and nonlinear fitting methods, including least squares, weighted least squares, and robust fitting (all with or without bounds) ,
- Custom linear and nonlinear model development,
- Nonparametric fitting using splines and interpolants ,
- Interpolation, extrapolation, differentiation, and integration of fits.

The Signal Processing Toolbox is a collection of tools built on the MATLAB numeric computing environment. The toolbox supports a wide range of signal processing operations, from waveform generation to filter design and implementation, parametric modeling, and spectral analysis.

All toolbox functions are written in the open MATLAB language. This means that users can inspect the algorithms, modify the source code, and create their own custom functions.

MATLAB can perform arithmetic on a wide range of data types, including doubles, singles, and integers. This allows faster execution and reduces the need for memory.

3.4 Publishing Results

MATLAB includes export of results as plots or as complete reports. All popular graphics file formats are supported allowing import into other packages, such as Microsoft Word or Microsoft PowerPoint. Using the MATLAB Editor, MATLAB code is automatically published in HTML, Word, LaTeX, and other formats.

MATLAB Report Generator generates automatically document on tasks performed in MATLAB, such as analyzing and visualizing data and developing algorithms. It enables to run MATLAB code and capture the graphics and data as they are produced. The provided templates can be used directly or customized to fit specific report characteristics. The MATLAB Report Generator facilitates information exchange and helps keep documentation up to date with the workflow.

3.5 Deploying Applications

The MATLAB Compiler ensures the automatic conversion of MATLAB programs into self-contained applications and software components that can be shared with end-users. Applications and components created using the Compiler do not require MATLAB to run.

The MATLAB Compiler significantly reduces application development time by eliminating the process of manually translating MATLAB code into C or C++. Both stand-alone applications and software components can be compiled, packaged and distributed to end users who do not work in MATLAB.

Moreover, the MATLAB Builder products allow the conversion of MATLAB applications and algorithms into additional types of software components, such as Excel add-ins and COM objects, for use within other applications.

To summarize, the MATLAB Compiler has the following features:

- Automatically convert MATLAB programs including graphical user interfaces into executables that run outside the MATLAB environment

- Package and distribute stand-alone executables and software components at no extra charge

- Incorporate MATLAB based algorithms into applications developed using other languages and technologies, such as C or C++.

4. Conclusion

Data Analysis for telemetry requires handling of large data files and visualization tools to inspect these data. Thanks to its capabilities like ready-to-use mathematical functions, visualization and reporting tools, MATLAB is penetrating the area of telemetry very quickly. It helps users to construct new interfaces that display their data and allow complex analysis without a thorough knowledge in signal processing, statistics or programming. Thus the telemetry users can easily add new capabilities to their analysis in order to fit their needs (e.g. synchronizing playback of data, audio, and video). They can share their work by deploying the MATLAB application and by automatically generating a report. The result is that a small effort is provided to build the telemetry tool and the entire Data analysis group can benefit from it.

5. Acknowledgement

The author acknowledges the contribution of their colleagues to this work, especially Lisa Kempler and Scott Hirsch.