Boeing Team B CS397 Project Guide

Implementation of Table Based Volumetric Compensation

Alex Bertels, Charles Ortman, Joseph "Gus" Steurer, Kevin Zheng

Project Files

In the root directory, the following folders are found:

- DOCUMENTATION: Contains information collected for the project including client proposals and team design documents.
- LIB: MATLAB DLLs (.Net Assemblies) used by the project for compilation. This directory currently includes the transformation script and will, once compiled, contain the ILM and GA DLLs.
- MATSCRIPTS: The .m files used to generate MATLAB DLLs with NE Builder.
- SAMPLEDATA: Sample inputs to the VEC software application.
- VECTool: The Visual Studio project directory. Contains project source code.

Project Source Code

In the directory VECTool/VECTool the following directories and files will be found:

- bin & obj: These directories contain the binaries and compiler artifacts generated by compilation. They should not be present in the git repository and should never be committed.
- Properties: This folder is managed by Visual Studio and should not be modified by the user. It must be committed with the project whenever changes are made.
- Resources: Contains resources such as images used by the application. Files are added using Visual Studio.
- App.xml: This file contains configuration information managed by Visual Studio. It should not be manipulated by the user.
- Program.cs: Entry point for the application. It is automatically generated as part of a Visual Studio project. It serves as the VECTool's main function, and should only instantiate the main VECGUI window and the VECState class. It is not documented as part of the project design because it is a C# specific entity.
- VECGUI.Designer.cs: This file is automatically generated and maintained when a GUI is manipulated in Visual Studio's designer tool. It should never be manipulated by the user.
- VECGUI.resx: Contains configuration information for the VECGUI designer module. This file is managed by Visual Studio and should not be manipulated by the user.
- VECTool.csproj: Contains configuration information for the project. This file is managed by Visual Studio and should not be manipulated by the user.

The following files correspond to modules specified by project design requirements. Further information can be found in the project documentation folder. They are listed in order corresponding to runtime workflow.

- VECState.cs: This module maintains global state information for the VECTool's runtime lifecycle. It is manipulated to contain program input, which is processed as the

application moves throughout its phases. This class corresponds to FR4.
- VECGUI.cs: The main application window for supply user inputs and providing runtime feedback. It also manages runtime lifecycle corresponding to each phase of preprocessing. This class corresponds to FR1, FR2, FR3, FR7, FR11, and FR12.
- ToolMeasurementHandler.cs: Responsible for aligning raw input such that they fit within a margin of error. This class corresponds to FR5, FR6, and FR8.
- CommandMeasurementHandler.cs: Ensures that predicted measurements match existing actual measurements. This class corresponds to FR9 and FR10.
- ReportGenerator.cs: This module is responsible for formatting application output. It corresponds to FR13, FR14, and FR15.
- FR11, FR12, FR13, FR14, and FR15 have not yet been implemented. The client was unable to supply the required MATLAB modules used for these steps due to complications with MATLAB compilation to .Net libraries and Intellectual Property concerns. These requirements are staged for a future project.

Compilation Requirements
- The MATLAB Compiler runtime must be installed in order to compile or run the VEC software [1].
- The Microsoft Visual Studio .net Framework version 4.5 must be installed for compilation and run time.
- The file VECTool.sln should be opened in microsoft Visual Studio version 2010 or 2012.
- Version control is handled with the GIT version control software. A remote repository can be found at [2] or obtained from [3].
- Compilation is accomplished by satisfying all Compilation Requirements, opening the project in Visual Studio (VECTool.sln), and selecting 'Build'.
- Compilation of MATLAB DLLs requires MATLAB r2012a, MATLAB Compiler, and NEBuilder. Compilation and integration is detailed at [4].

Testing Policy

As a requirement is implemented, developers will resolve compilation errors, supply input to functions, and observe output from the VECGUI log window. If the expected output does not match the actual output observed through the VECGUI log window the issue is resolved and retested. Developers attempt to anticipate cases that might violate input constraints and resolve corresponding issues. A small set of input data has been supplied by the client. Throughout development, each team member used this set of input data to test their implementation.

A very limited subset of input information was used to test the application, as this information is frequently not preserved by the manual process and the client was unable to supply multiple sources of data. For future testing on the project, sample input and output must be collected for different input scenarios. This test data must be supplied by the client.

Footnotes
[1]
http://www.mathworks.com/supportfiles/MCR_Runtime/R2012a/MCR_R2012a_win64_installer.exe

[2]https://jasteurer@bitbucket.org/jasteurer/vec.git

[3] Joseph August "Gus" Steurer
gsteurer@gmail.com
asjxc9@mst.edu
314.448.7779

[4] http://www.mathworks.com/videos/using-matlab-builder-ne-68758.html