SAND2014-17418R

New Software Architecture Options for the TCL Data Acquisition System

Emmanuel Valenton, UC Berkeley, B.S. Chemistry, est. June 2016
Mentors: Robert Barlow and Robert Harmon
Org. 8351, Reacting Flow Research Department
August 29, 2014

Abstract

The Turbulent Combustion Laboratory (TCL) conducts research on combustion in turbulent flow environments. To conduct this research, the TCL utilizes several pulse lasers, a traversable wind tunnel, flow controllers, scientific grade CCD cameras, and numerous other components. Responsible for managing these different data-acquiring instruments and data processing components is the Data Acquisition (DAQ) software. However, the current system is constrained to running through VXI hardware—an instrument-computer interface—that is several years old, requiring the use of an outdated version of the visual programming language, LabVIEW. A new Acquisition System is being programmed which will borrow heavily from either a programming model known as the Current Value Table (CVT) System or another model known as the Server-Client System. The CVT System model is in essence, a giant spread sheet from which data or commands may be retrieved or written to, and the Server-Client System is based on network connections between a server and a client, very much like the Server-Client model of the Internet. Currently, the bare elements of a CVT DAQ Software have been implemented, consisting of client programs in addition to a server program that the CVT will run on. This system is being rigorously tested to evaluate the merits of pursuing the CVT System model and to uncover any potential flaws which may result in further implementation. If the CVT System is chosen, which is likely, then future work will consist of build up the system until enough client programs have been created to run the individual components of the lab. The advantages of such a System will be flexibility, portability, and polymorphism. Additionally, the new DAQ software will allow the Lab to replace the VXI with a newer instrument interface—the PXI—and take advantage of the capabilities of current and future versions of LabVIEW.

Introduction

The Turbulent Combustion Laboratory (TCL), located in the Combustion Research Facility (CRF) at Sandia National Laboratories, studies flames in a turbulent environment. Turbulent combustion is found in a variety of applications such as inside the combustion chamber of a car or jet engine. A variety of turbulent combustion configurations is studied by shooting a laser through the flame and analyzing the Raman and Rayleigh scattering of the laser light. Based on the scattering, many different attributes of the flame, such as major species, OH concentration, and temperature, can be characterized along an imaged segment of the laser beam.

The TCL is well equipped for such research, housing nine pulsed lasers, seven scientific grade CCD cameras, more than ten computers for data collection and experimental control, customized optical signal collection systems, a traversing wind tunnel, and up to 28 mass flow controllers. Controlling and coordinating these different components is the DAQ software.

The DAQ software passes data from the Lab instruments to the data storage computer and other devices and then passes commands from various control panels back to the instruments. The current software was programmed using LabVIEW, a programming language geared for ease of development of instrument control and graphical user interfaces, and runs on the LabVIEW run-time engine.

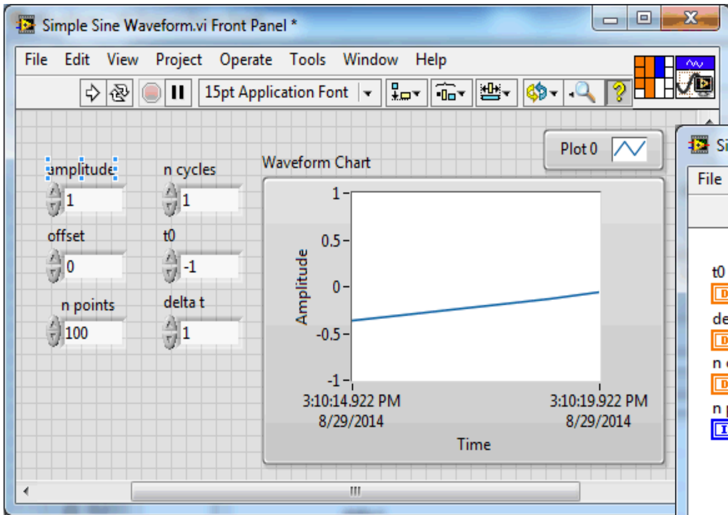
However, operation of the current DAQ software depends heavily on a VXI—an instrument interface computer—that is about fifteen years old. The VXI is no longer being supported by the manufacturers and can only run an older version of LabVIEW. As a result, the crucial components of the DAQ software running on the VXI are constrained to this older LabVIEW version and cannot take advantage of the improvements and features offered by newer LabVIEW versions.

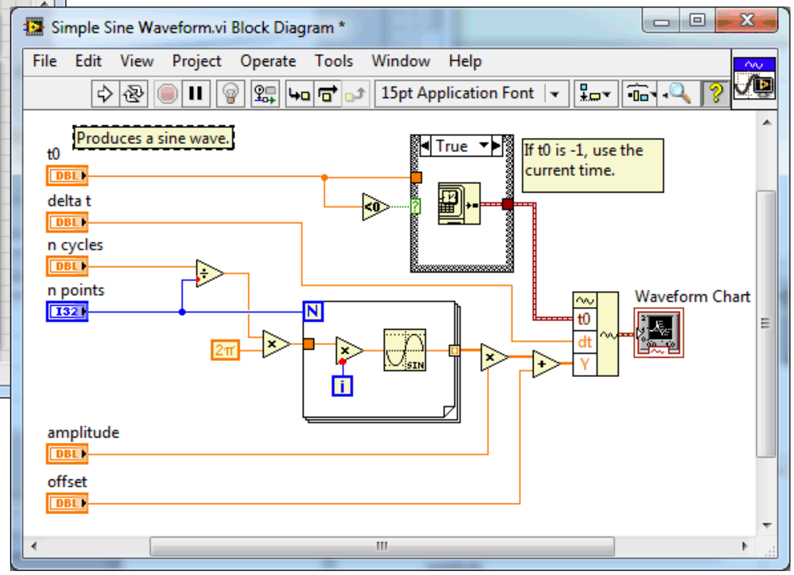
The objective of this project is to create a new DAQ software package for the TCL. The software will implement several improvements over the old software and allow the Lab to replace the VXI with a newer instrument interface—a PCI Extensions for Instrumentation device or PXI.

Software Design

Implementation of the new DAQ software was done with LabVIEW, a graphical programming language. LabVIEW relies on putting graphical symbols together on a screen instead of programming using text commands as traditional coding is done. These symbols represent traditional programming commands—for instance, a blue box represents an integer variable, while a triangle represents a mathematical or logical operation. These symbols are connected using wires to indicate the flow of information. Thus a fully-fledged program would consist of something looking very similar to an electronic circuit diagram.

Example LabVIEW Code





The use of LabVIEW yields several advantages over traditional coding. First, the LabVIEW language offers the advantage of ease of parallel programming and modularity. In essence, modularity means designing a section of code to perform a specific task, while parallel programming involves organizing sections of code to run at the same time—in parallel. Furthermore, using a graphical programming language allows the programmer to visualize the flow of actual data from an instrument. It is easier to follow the raw data from the instrument through signal processing to the data's destination, wherever that may be. Finally, LabVIEW makes it very easy to create graphical user interfaces—a very important asset when creating instrument control programs.

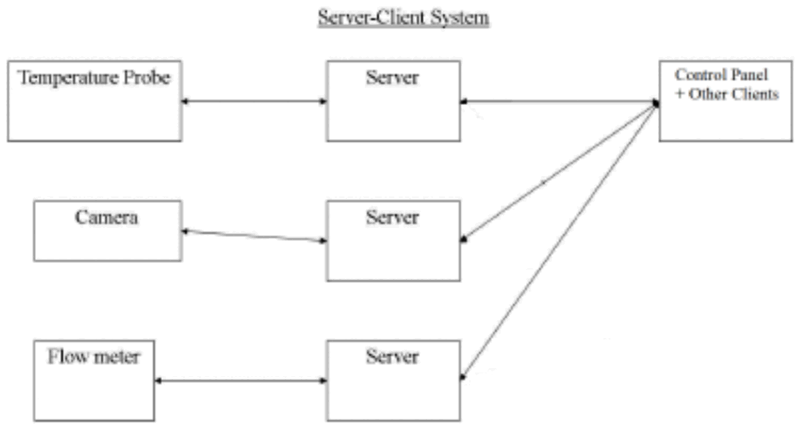
Towards the creation of the new DAQ software, two different designs were considered. The first was the Server-Client design and the Current Value Table (CVT) design. The server-client model operates very much like the way Internet servers and clients work and is relatively easy to implement. The CVT model on the other hand is based on a central "spreadsheet" and offers several advantages, such as flexibility, over the Server-Client model. However, implementation is a longer and more complicated process.

The server-client model is based on server-side and client-side data processing, in addition to the dynamic connections a client sets up with a server. For instance, a lab component, such as a gas flow meter, would read the gas flow rate of fuel gas running through a feed pipe and send a raw data signal to a PXI chipset. The PXI would then pass the signal off to the gas flow meter's designated server program running on one of the lab computers. The server program would interpret the signal as gas flow rate data and then format and store the data in memory. Next, a client program, such as a data display program, would connect to the server just like an Internet client connects to a web server. The gas flow meter server program would then send the gas flow data to the data display client, which would display the data on a chart.

Information would also flow in reverse. A client program such as a control panel would connect with the gas flow meter server. A user would enter a command—such as a stop command—and the client would bounce the command to the server. The server would receive the command and then send the appropriate signal through the PXI to the gas flow meter. The gas flow meter would then close, shutting off the flow of fuel gas. It is important to note that two clients can connect to the same server at once—so the data display client and the control panel client can connect to the gas flow meter server at the same time.

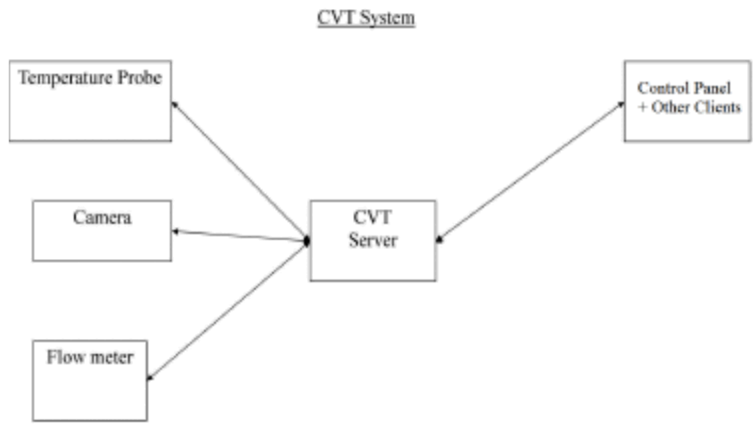
The server-client model would require a server program being created for each instrument of the Lab. In addition, a separate client program would have to be created for the different Lab components—such as the control panel for one—trying to access data or send commands. Implementation of the program would involve a separate computer to run the individual client and server programs--although it is possible to run many programs on the same computer.

The main advantage of following this server-client model is that the current DAQ Software to be replaced is based around this model—the foundation is already laid. The brunt of the work would be to tailor the server-client model to the different instruments of the lab and then install the software on the new PXI. Construction of a server-client DAQ Software would be a relatively straightforward process.



The other design was the Current Value Table (CVT). This system involves what in essence is a giant spreadsheet (the CVT), running on one central server. Next, each component of the lab would have its own client program—similar to the server-client system with clients running the lab instruments instead of server programs. Then, each client program would maintain a constant connection to the CVT Server and write data to the different spreadsheet cells on the Server.

From there, clients which process data or write commands, such as the control panel client, the data display client, and the data storage client would also maintain a constant connection with the Server and constantly access and download the data stored on the "spreadsheet". Furthermore, commands coming from the control panel, for example, would be written to the CVT Server. The Lab instrument clients would then access the commands stored on the Server. Each of the various clients would be constantly pinging the CVT server for information. As a result, data and commands would be flowing in real time across the DAQ Software.



At present, the CVT design is being investigated. A prototype CVT DAQ Software package has been created, which consists of the CVT server and an assortment of client programs which model a gas flow system. One client is designed to monitor the flow rate of the fuel, oxidizer, and masking gases going into the burner and send the flow rates to the CVT Server. Another will read gas flow commands from the server and adjust the gas flow accordingly. A third client will act as the control panel, writing commands to the CVT server, and a fourth client will display the gas flow data on a graph. The focus of this stage is to see how well the CVT model will serve the Lab's needs, so the prototype program is being tested by running the System for several days using randomly generated gas flow data.

A few issues are of concern with this program. One is the issue of lag. If there is a delay between the time an instrument generates data and the time that the data arrives at a different client program, caused by the trip that the data has to take through the CVT server, then the CVT System will have to be improved. The same goes for the question of whether the single CVT Server will be able to handle the many, many clients the fully-fledged DAQ System will have.

Future Work

It is very likely that the CVT DAQ Software model will be chosen over the Server-Client model. If the testing shows that the CVT model can handle the data flow of the Lab, then implementation will move to the next stage, which is to extend the program until it serves the entire laboratory. Towards this end, numerous client programs need to be created and tailored to each individual component of the lab. For instance, one client program will be made for the Traverse System, which controls the co-flow burner's vertical and horizontal position, while another client program will be made for the gas control system, which controls what gases are sent to the burner. Each of the Lab instrument client programs has to be customized to interpret the many different signals coming from the Lab instruments and also to process the commands coming in from the CVT server.

Not only will the fully implemented system operate the extent of the lab, but there are a number of planned capabilities that the current DAQ Software does not have. For one, the biggest advantage of the CVT System is that it allows the flexibility to add, remove, and modify client programs. This ability would greatly aid in diagnostics and troubleshooting because it would allow the Lab Technologist to install a control panel client on a laptop or even a tablet, for example, and then bring the tablet over to a problematic area of the lab, and then perform diagnostics while controlling the system from that one spot. Ease of rapid installation would also aid in one very noticeable flaw of the CVT System—the dependence on the central CVT Server. Every client in the system communicates through the Server, and all data and commands flow through the server. If there server experienced an error, then the entire system would be incapacitated. Ease of installation would alleviate the problem by shortening the time it takes to install the Server program on a new computer as a replacement.

Currently further development of the program requires only the current version of LabVIEW and a reasonably fast computer. These resources are all that is needed for the development phase that the program is in. Likewise, as the system progresses from prototyping to implementation into the lab, the only needed resources will be access to the Lab and the Lab computers. It is recommended, however, that additional PXI, just like the ones used in the Lab, be acquired so that a clean, separate environment may be setup for testing the new system, without having to go into the Lab and interrupting any work being done there.

Impact

Completion of this project would provide a great service to the Lab's goal of combustion research by ensuring smooth and rapid Laboratory operation. Use of this new system would provide several operational improvements over the old one. Likewise, implementation of the new system would allow the lab to replace the outdated VXI instrument interface with the new PXI hardware, removing the constraints of running the older LabVIEW software.

Conclusion

The net result of this project is a prototype Data Acquisition Software package that is based on the CVT System model and that forms the foundation of the fully-fledged Data Acquisition Software to come. The project has shown that the CVT System model has promise for coordinating multiple clients and facilitating communications between those same clients. Much still has to be done before the project's goal of overhauling the Turbulent Combustion Laboratory's Data Acquisition Software may be realized, but currently strong strides have been made in that direction.

Appendix

I. Participants
   - Robert Barlow—Project Leader and Intern Mentor
   - Robert Harmon—Lab Technician and Intern Mentor

II. Scientific Facilities

   The Turbulent Combustion Laboratory, located in the Combustion Research Facility of Sandia National Labs, served as the host site for this project.

Acknowledgements

"This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Community College Internship (CCI) program."

"Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP"