
AC 2012-4629: NETWORK-BASED DATA COLLECTION FOR A PROJECT-BASED FRESHMAN CLASS

Dr. Samuel Bogan Daniels, University of New Haven

Dr. Daniels is an associate professor of mechanical engineering with more than 20 years of experience teaching laboratory classes. He also teaches in the multidisciplinary engineering foundation spiral curriculum at the University of New Haven. Research interests are in engineering education and renewable energy systems.

Dr. Cheryl Q Li, University of New Haven

Dr. Cheryl Qing Li joined University of New Haven in the fall of 2011, where she is a senior lecturer of the Industrial, System & Multidisciplinary Engineering Department. Dr. Li earned her first Ph.D. in Mechanical Engineering from National University of Singapore in 1997. She served as Assistant Professor and subsequently Associate Professor in Mechatronics Engineering at University of Adelaide, Australia and Nanyang Technological University, Singapore respectively. In 2006, she resigned from her faculty job and came to Connecticut for family reunion. Throughout her academic career in Australia and Singapore, she had developed a very strong interest in learning psychology and educational measurement. She then opted for a second Ph.D. in Educational Psychology, specialized in Measurement, Evaluation and Assessment at University of Connecticut. She earned her second Ph.D. in 2010. Dr. Li has a unique cross-disciplinary educational and research background in Mechatronics Engineering, specialized in Control and Robotics, and Educational Psychology, specialized in Statistical Analysis and Program Evaluation.

Dr. W. David Harding P.E., University of New Haven

Network-based Data Collection for a Project-based Freshman Class

The hands-on experience for engineering students is often lacking with class sizes increasing and classroom resources under constant pressure. Even with smaller class sizes and studio classrooms there is only so much equipment and technical support to run experiments and allow for student interaction. Network-based data collection and control may alleviate some of these issues and allow for more hands-on experiences, although remote, in the classroom. This paper discusses the interesting problems and not so obvious pitfalls of setting up some simple network-based data acquisition systems for a multidisciplinary freshman level engineering course. The course as currently taught includes some sensor calibration and feedback/control projects using LabVIEW that are introduced in the last six weeks of the term. In groups of three or four students, they develop a basic understanding of interfacing to sensors and using LabVIEW through a series of guided exercises. The projects assigned are all team based, so it is always a concern that each student in the team develops a good understanding of sensor calibration, data acquisition and control fundamentals. Too often one student will operate the system and the other students just stand to the side watching. With the limited number of control experiments and data acquisition systems, there have been few options for improving the experience. The introduction of network-based data collection for a simple thermistor exercise improved the class organization and structure but removed the student team aspect of the exercise and drew consistent student disapproval with a clear preference for team-based lab exercises.

LabVIEW has been used for over two decades as an experimental simulation tool and programming environment with faculty teaching physics labs¹ or teaching programming². The applications span senior level work with Bluetooth and serial communications^{6,8}, mid-level work with LabVIEW as both a remote experimental interface and learning environment at Sonoma State University⁵, to freshman level feedback and control labs⁷. The advent of network-based data collection has even seen some significant development of cross-continental collaboration using MIT's "ilab" to African Universities^{3,9}. The recent work at Virginia Tech⁴ for a freshman level class to collect remote data on water quality is perhaps the closest comparison to the work described in this paper.

There are currently two freshman level multidisciplinary courses taken by nearly all entering freshman. These courses include topics from all of the disciplines and are taught with an emphasis on either common disciplinary topics or a collection of topics from different disciplines so students can gain a better understanding of the different majors before entering the major classes. One of these two freshman courses is quite unique and has few comparable courses at other universities, EAS109 Project Planning & Development. There are two major projects for the course, a manufacturing floor simulation and a feedback control project. The manufacturing floor simulation is done with small robots and conveyor belts where AutoCAD drawings, control programs and project management documentation are required. The students learn personal organization and team/project organization and planning as they work through a very fun and enjoyable activity. Sadly the same cannot be said of the second project.

The feedback control project consists of a sequence of laboratory experiments aimed at interfacing to and calibrating sensors and computer control of motors and other systems. This sequence eventually culminates in the application of those calibration and feedback control skills to one of five desktop scale control projects: water tank level, salinity of fluid stream level, chamber temperature level, ball and beam position, and light tracking position. The last two are Quanser experiments while the others were built by the instructors. Each and every engineering discipline has examples of sensors and feedback control that are necessary in industry. The time constants may differ but the basic concepts and techniques are common so one would think that this would be an ideal project for freshman engineering students.

The difficulty that students face with the control projects is multifaceted. First, a low level graphical programming environment organized much like a program flow chart is used to program the robot and conveyor belt in the first project. This allows students who have never programmed before to gain some experience before shifting to the industry standard program, National Instruments LabVIEW, used for the feedback control projects. The shift to LabVIEW is significant enough that the students have difficulty learning the interface and the structure of the programming environment. Next, the idea of calibration of sensors, a necessary first step to running a control program based on sensor feedback, is entirely new to students. The process of collecting data points experimentally then deriving a model for the sensor behavior is in itself a rather big step for freshman. The need for multiple points and accessing the fit and function are details that are introduced but not emphasized. Interfacing to sensors always involves some wiring and this class used the NI-ELVIS platform to setup the sensor and connect to the computers and devices for control. Seven stations are setup in a studio style classroom with 24 student seats with computers in the center of the room and the seven experimental stations setup on the outside perimeter.

The normal classroom experience has students grouped with those in adjacent seats and asked to move to the outside wall for testing, calibration and programming. The difficulty lies with 3-4 students attempting to access the same equipment and computer on the adjacent counter top. With only three students per group, one student can run the computer while another modifies the test conditions for the calibration and the third collects the data. This type of approach works well but compartmentalizes the experience of the students into active and passive participants. This can obviously lead to one student taking on the difficult task of running the calibration or writing the control program while the others stay to the side and watch. An undesirable outcome that may also occurs to some degree for the first project.

In the first project the students all gain experience in class with all of the tasks thanks to the 24 computer seats in the center of the studio classroom. All students learn how to program the robots using small motors running on gantries. Each student develops a program and then independently tests that program with the gantry motor. For AutoCAD, each student has some mandatory drawings they must do for the class and all students must independently develop project management documents for a small project as homework. This in class or homework experience with the different aspects of the first project is lacking in the second project and thus the need for a common calibration, programming and feedback control experience.

The need for individual student experience running LabVIEW is already a part of the class and a simulated signal is used to try and teach calibration and feedback control. An improvement on this situation, or so we thought, would be for the students to directly gain experience working with the sensors and devices without having to wait their turns at the seven setups around the room. The network features of LabVIEW were considered the only reasonable solution to this problem.

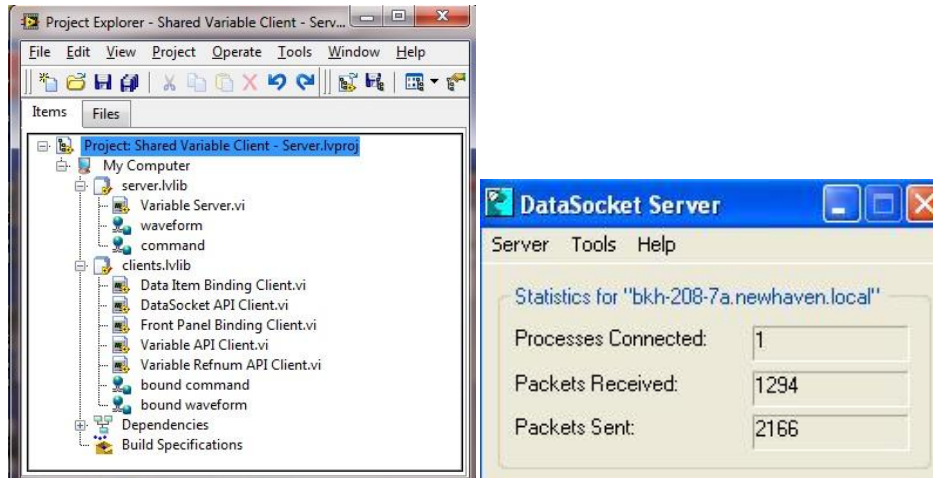


Figure 1- Data socket server in LabVIEW project

LabVIEW support data streaming via a project with the use of a datasocket server (figure 1). The datasocket server used was an example found in the network section of the getting starting interface of LabVIEW. There are numerous examples and among them was the project file, “shared variable client- server”. This project was already setup to stream two variables, a waveform and a command. The waveform was a double precision floating point array appropriate for collecting data from a single channel. The datasocket server is initialized when the project is opened but the actual streaming is done using a separate VI that writes data to the waveform variable. There were also examples available that demonstrated how to write data to the datasocket server (figure 2) and to read data from a datasocket server (figure 4). Both the datasocket writer and reader had to be modified to the specific problem. The datasocket writer example used a virtual signal to create the data that streams over the network. The program also had some additional programming structures not needed for this example. The DAQ Assistant replaced the virtual signal and the extra structure was eliminated to create the simple program shown (figure 2). Once the “DS Writer” is running, the datasocket server window (figure 1) appears and tracks the total number of packets sent and number of processes, users on the network, that are connected. That number is controlled via settings in the project file. For our purposes, we had about twenty students accessing the data at the same time. The dstp protocol shown in the front panel of the “DS Writer” uses the local computer name to stream the signal so while the name “localhost” is used in the virtual instrument, in this case it means the network name of the computer which was “bkh-208-7a.newhaven.local”. It was the network name that had to be used in the datasocket read program. This took quite some time to figure out for a group of mechanical and chemical engineering professors but fortunately the LabVIEW documentation is excellent and relatively easy to understand.

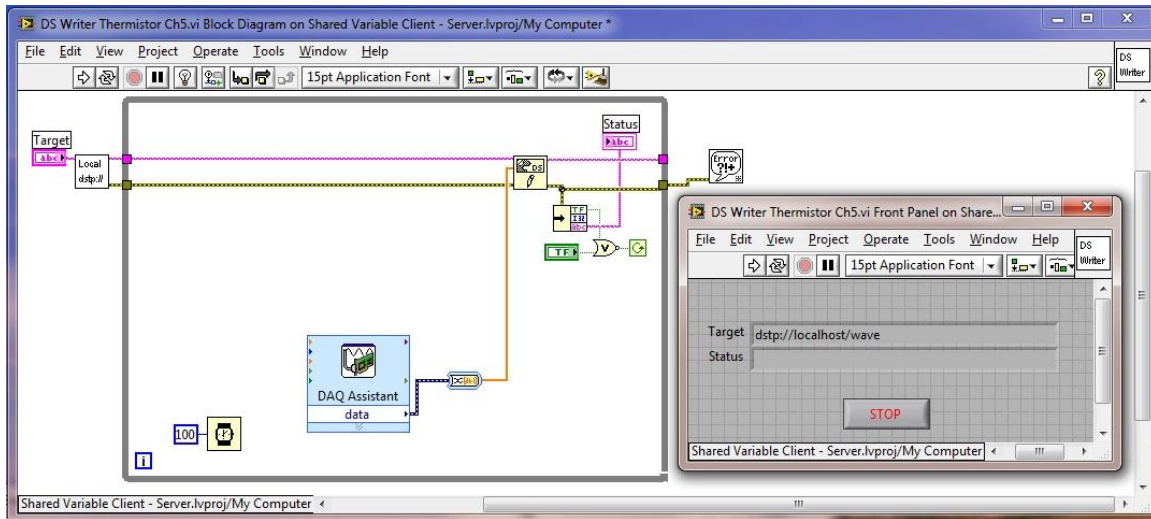


Figure 2 – A simple LabVIEW program is used to send data to the socket server

The classroom exercise was organized such that the instructor and student assistant were working in the front of the room and the students each collected data separately using LabVIEW via the datasocket read example provided. A template VI was posted online for the students to download. While the students were new to LabVIEW, the previous experience with graphical programming provided enough experience that only minimal help was needed to modify the VI (figure 4) and collect data manually in Excel. The thermistor experiment shown involved the recording of temperatures and voltages then plotting the data to generate a calibration curve that would later be used to create a virtual thermometer.

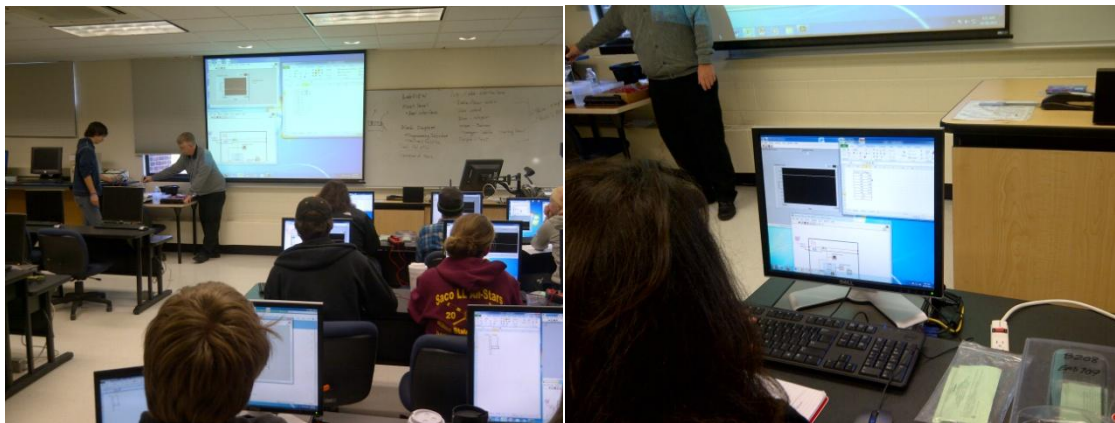


Figure 3- Data is streamed from a single experimental station to the network.

The students had to add the statistics VI to the program, type in the machine name, adjust the chart settings and add a numeric value to read from the average of each data set collected at a given temperature. Given that this was the first time using LabVIEW, they were also introduced to variable types, basic programming structures and how to navigate through the windows.

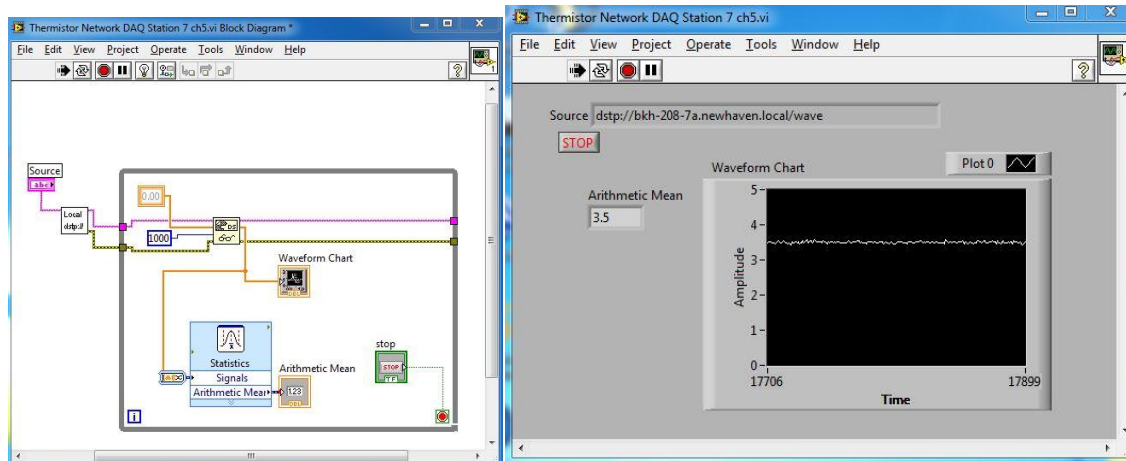


Figure 4 – Network data is read from the socket server using the local machine name

There were two classes that this approach was tested on and it was assumed that the individual access to LabVIEW with a real signal rather than simulated one would keep them interested and provide a better understanding of programming LabVIEW. It was also hoped that the elimination of the fairly chaotic nature of the lab experience with students crowded around a lab table would be a better experience for the students. A brief survey of the two classes was developed and given to the students after they had experienced that chaotic atmosphere in the next LabVIEW experiment where they develop an RPM indicator from a motor that operates a switch using a gear to generate a pulse wave.

The survey questions used a Likert scale and was setup so that the first two were in favor of data streaming and the second two opposed. The hope was that the average for the first two responses would be higher indicating a preference for data streaming of data and the second two would show a lower average value indicating disagreement with the questions and also a preference for network-based data collection. Oddly enough, the exact opposite occurred. The students clearly preferred the team-based collection of data rather than the network-based. Table 1 shows that while the students may have perceived that they gained some improved understanding of programming in LabVIEW by individually developing their own programs, they clearly preferred to work in teams. Figure 5 shows the distribution of responses that clearly show a bias toward teamwork and a slight bias against working individually to learn LabVIEW. These results were quite a surprise given the apparent calm and wonderful structure that was obvious during the networked-based data collection vs. the chaotic nature of the later experiment with three to four people around a single experiment. While the students did not appear to enjoy this approach, the question of gains in programming and understanding still remain unanswered. As a follow up to this activity, the integration of the network-based data collection with small teams of two can be assessed as an alternative and tested during the spring 2012 semester and compared to a class that uses separate lab stations and no networking of data. A test of the actual programming and conceptual gains will also be needed to see if this approach does provide better student outcomes.

There was another potential reason for the unfavorable responses in class 1. The LabView version posted for download was different from the version installed on the student computers. So during class 1, the students had to pass a thumb drive around to get the compatible LabVIEW

version of the template. As a further refinement, the initial introduction to LabVIEW will be given prior to this exercise so students can build the code from scratch, rather than starting with a template. Then the student will learn the data collection process simultaneously as the instructor goes through the programming details. We will test these alternatives in our future studies.

Student Survey Questions	Class 1 Average	Class 1 standard deviation	Class 2 Average	Class 2 standard deviation
Question 1: I gained a better understanding of LabVIEW using data streaming than I would have if I worked in a group at the lab stations.	3.17	0.99	3.21	0.92
Question 2: I prefer to work on lab experiments with data streaming on my own computer compared to working in a group on a single lab computer.	3.1	1.5	2.5	1.2
Question 3: By working in a group of students, I gained a better understanding of the DC Motor rpm exercise than I would have if the data was streamed and I worked individually.	3.5	1.2	3.8	1.1
Question 4: I gained a better understanding of the LabVIEW programming by working in a group during the DC Motor rpm experiment than I would have working individually with data streaming.	3.5	1.3	3.9	1.1

Table 1 – Results of student survey of network-based laboratory experience

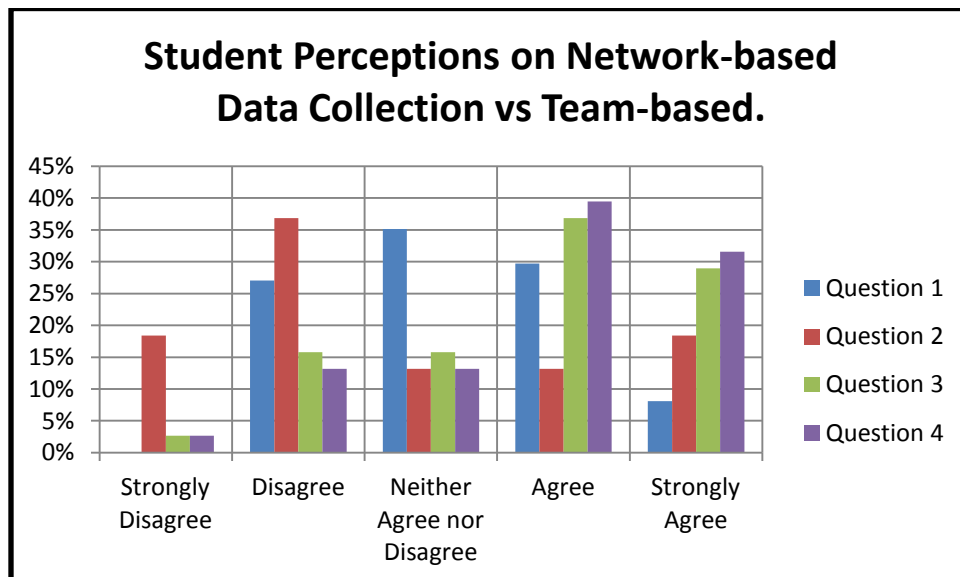


Figure 5 – Student perceptions of lab experience

References

1. Newmann, C., Lieberman, D., Engelberg, D., Flamholz A., Marchese, P., Tremberger, G., Cheung, T., "LabVIEW Graphical Programming in an Introductory Engineering Physics Course", ASEE Annual Conference 2006-394.
2. Hrynuik, J., Pennington, M., Illig, D., Dempsey, J., "Freshman Engineering: An Introductory Computer Course Teaching Matlab and Labview", ASEE Annual Conference 2008-1408.
3. Akinwale, O., Kehinde, L., Ayodele, K., Jubril, A., Jonah, O., Ilori, S., Chen, X., "A LabVIEW-Based On-Line Robotic Arm for Students' Laboratory", ASEE Annual Conference 2009-1179.
4. Lohani, V., Delgoshaei, P., Green, C., "Integrating Labview And Real-Time Monitoring into Engineering Instruction", ASEE Annual Conference 2009-1764.
5. Khan, S., Farahmand, F. , Moslehpour, S., "A LabVIEW-Based Integrated Virtual Learning Platform", ASEE Annual Conference 2010-1159.
6. Loker, D., Frampton, C., McElhaney, T., Mook, J., Sansone, A., "Remote Data Acquisition using Bluetooth", ASEE Annual Conference 2005-2147.
7. Daniels, S., Harding, D., Collura, M., "Introducing Feedback Control to First Year Engineering Students using LabVIEW", ASEE Annual Conference 2005-2161
8. Loker, D., "Remote Data Acquisition using LabVIEW", ASEE Annual Conference 2001-2259
9. Achelengwa, Edison and Oyebode, Olayemi. "IAP 2011 iLab Africa Report". Cambridge, MA, USA. February 24, 2011.