Full Paper

A LabVIEW Based On-Line DC Servo Control Experiments For Students' Laboratory

K.S. Alli
Department of Electronic & Electrical Engineering
Obafemi Awolowo University, Ile- Ife, Nigeria

A.M. Jubril
Department of Electronic & Electrical Engineering
Obafemi Awolowo University, Ile- Ife, Nigeria
ajubril@yahoo.fr

ABSTRACT

In this paper, the development of DC control experiments using ilab interactive architecture is presented. The study is to design and develop Graphic User Interfaces (GUIs) using LabVIEW, to enable students in performing position control, speed control and system modeling of the Quanser DC Motor Control System experiments platform. This platform is implemented online using Massachusetts Institute of Technology (M.I.T) interactive laboratory architecture. The scope of this project is to apply virtual instruments to implement the Proportional-Integral-Derivative (PID) controller in position control, speed control and modeling of the control system. PID controller was used to control the output response and stabilize the control system. Interactive software is developed to help the students to visualize and analyze the system.

Key words: *LabVIEW, iLab, PID Controller*

1. INTRODUCTION

The field of automation control system and engineering is largely unexplored in the underdeveloped and the developing countries. Few universities undergo course work in control system and engineering and even fewer have labs for servo control system (Akinwale, 2010). This is the case despite the fact that the field of automation control system and engineering stands as one of the most promising of the future. Even, the economy of the world depends on servo control. The world is gearing up for the next level of automation where every device would be smart and able to do lots of things on their own without human participation. Typical areas of application of servo control system are found in the manufacturing industry, in scanners, printers, cameras, robots, CD players, vehicles and instrumentation. A characteristic feature of motion control is that, it is often possible to obtain mathematical models of the systems from first principles, possibly with a few complementary experiments. A simple DC motor can be used to illustrate motion control. Typical experiments are to control the speed or the motor angle in desired ways.

There are many other types of control problems. Stabilization of an unstable system is one task, damping of a swinging load on a crane, motion planning for a moving robot, traction control of cars are examples (Laubwald, 2000).

This research was prompted by the limited number of online lab experiments and lectures in the field of control system and engineering. The focus of this study was to expand the set of online experiments in the control engineering field by familiarizing users to control system. Hence, the availability of online experiment setups and laboratories can go a long way in facilitating the development of any technological system and hence, country or region.

The set of online experiments which this research work put up are based on a servomechanism control system implementing PID controller.

Three experiments have been set up at the Obafemi Awolowo University by this research effort. The experiments are:

i. The position control experiment
ii. The speed control experiment
iii. The system modeling experiment

2. REVIEW

Previous works have been done on development of real-time online control laboratories and virtual instrumentation. These researches that are relevant to this project are discussed next to demonstrate continuity from previous researches:

Below is an account of some of the existing online control engineering laboratories.

Imran *et al.* (2003) described a recently developed DC motor position control experimental setup that can be accessed via the Internet. The experiment consists of two primary elements i.e. a server and a client computer communicating with each other. A server consisting of a low-cost microcontroller, Parallax's 40-pin Basic Stamp 2 (BS2P40), interfaced with an embedded Ethernet IC, Cirrus Logic's Crystal CS8900A, and the client computer sends/receives data to/from the microcontroller using User Datagram Protocol (UDP) packets. The client computer connects to the server using Java applets that allow the user to command the position of the motor via a graphical user interface.

Akinwale (2010) developed a robust ilab platform for conducting robotic arm experiments. Three experiments were designed on the robotic arm platform: position control experiment, an experiment on the effect of gravity on the control of the robotic arm and a trajectory planning experiment. The experiments were designed using LabVIEW programming. This platform was implemented online using Massachusetts Institute of Technology (M.I.T) interactive ilab architecture. The existing batched ilab architecture of Obafemi Awolowo University, Ile-Ife, Nigeria was also upgraded to support the interactive architecture and provide visual feedback to the users.The major weakness about this work was that there was error of non-linearity in the motor movement of the robotic arm in response to the linear input. This problem could actually be solved by implementing PID controller to the robotic arm system which would give the users control over the torques of the individual motors.

Aaron *et al.* (2002) have designed a "Closed-loop Position Control System using Labview". A Closed-loop DC motor control system is developed using National Instrument's (NI) Data Acquisition (DAQ) Board (Model MIO 6040E), LabVIEW software package and DAQ signal Accessory Board for smooth and accurate positioning. Reading the motor position and sending voltage to motor circuit are accomplished via the DAQ's input and output ports respectively. The system is tested in the laboratory for different angular displacements and initial position of the motor.

3. The Architecture Of The Lab

3.1. The Three-Tiered iLab Architecture:

The laboratory which has been developed and reported in this paper was developed using the three-tiered architecture developed by the Massachusetts Institute of Technology (MIT) iLab research team. A second architecture is also possible, the two-tiered architecture. In the two-tiered architecture, there is a lab server and a client (forming the two tiers). This is shown in figure 1(a). In the three-tiered architecture however, a third tier is included between the lab server and the client. This third tier is called the Service Broker. In the two-tiered architecture, all authentications, user arbitration, database management and control of the laboratory equipment are handled by the laboratory server. In the three-tiered architecture however, the laboratory server is left solely with the responsibility of controlling the laboratory equipment. Hence, the lab server parses user requests (i.e. experiment specifications which the user sends in to the lab), executes requests which are deemed okay and returns the results of the experiment. The Service Broker takes on the function of authenticating users, arbitrating the use of the laboratories and providing access to stored results from previous lab sessions. The Service Broker also determines what features of the laboratory each user has access to during his log-on session. For instance, a user classified as a student may be given access to one type of experiments only. Another person classified as an instructor may have access to the same experiment and also stored data from lab sessions of the students Harward *et al.* (2006a).

The Quanser DC Motor Trainer lab uses the ilab Interactive Architecture. In the interactive architecture, this service is handled by what is called the Laboratory-side Scheduling Server (LSS). When the laboratory owner has specified the periods of the day during which the laboratory would be available to a user system, the need for a second service surfaces.

There is the need for another web service which specifies how long each user can spend in the laboratory performing experiments. In the interactive architecture, this service is called the User-side Scheduling Server (USS). This server defines the size of each time block which a user can be allocated and the maximum number of time blocks which a particular user can reserve for a particular experiment. This server also, having a database, stores the user reservations so that when the particular user logs on to the ISB, the ISB can contact the USS to find out if he has any reservations for the particular experiment which he is requesting permission to run. A third service is also included in the interactive architecture, but which has no bearing with scheduling sessions for experiments. This service is called the Experiment Storage Server (ESS). The ESS is responsible for, as its name implies, storing results of each laboratory session for retrieval by users with appropriate permissions.

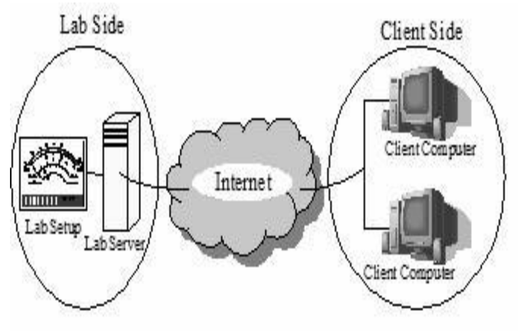

Fig.1(a): The Two-Tiered Architecture

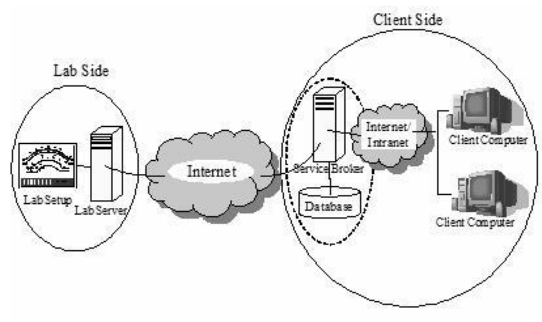

Fig.1(b): The Three-Tiered Architecture

Two main advantages of the three-tiered architecture are the flexibility it gives to the design of the online laboratory system and the reduction of the burden for the laboratory designer. Hence, each of these additional three services or three servers described, though they could have been coupled into the ISB and ILS, they are decoupled as individual servers with individual databases, with the ISB serving as the central processing unit, authentication and verification system, and router for data. Having two separate scheduling servers, one on the laboratory side and the other on the clients' side gives more flexibility to the system Harward *et al.* (2006b).

The geographical location of these servers in the Interactive Architecture is flexible.

4. Quanser Dc Motor Control System

The DC Motor control Trainer consists of a direct-current motor with an encoder and an inertia wheel on the motor shaft. The motor is driven using a pulse-width modulated (PWM) power amplifier. The power to the amplifier is delivered using the bulk power cable from a wall transformer and the encoder is powered by the ELVIS (Educational Laboratory Virtual Instruments) unit which converts the analogue rotation of the motor shaft to digital form. The inbuilt Data Acquisition (DAQ) card in the NI ELVIS II ensured Data to be read from the encoder. The control variable is the voltage to the drive amplifier of the system and the output is either the wheel speed or the angle of the wheel. Disturbances are introduced manually by manipulating the wheel or digitally through LabVIEW. The set up processes are controlled using a PC running the National Instruments Programming environment LabVIEW. The processes are connected to the computer using the National Instruments Educational Laboratory Virtual Instrumentation suite (NI ELVIS). As shown in figure 2, the QNET board slides with the NI ELVIS II. The QNET is compatible with both the traditional NI ELVIS, called NI ELVIS I. The NI ELVIS II has its own DAQ device that connects the PC via USB. Using LabVIEW, it is possible to implement user interfaces that are easy to used (Quanser, 2009).

5. THE CLIENTS

The front panels of the experiment engines created served as the laboratory clients on the user's end. The Remote Panels feature of LabVIEW enables VIs to be embedded in web pages. The front panels of the experiment engines was embedded in WebPages which the user accesses remotely and interactions of the user with these embedded VIs were received and worked on directly by the block diagram of the VI on the laboratory side.

6. THE LABVIEW RUNTIME INSTALLATION

The assessment of system performance was from the students of Electronic and Electrical Engineering Department, Obafemi Awolowo University, Ile-Ife, Nigeria who were asked to run the experiments at any of their locations.

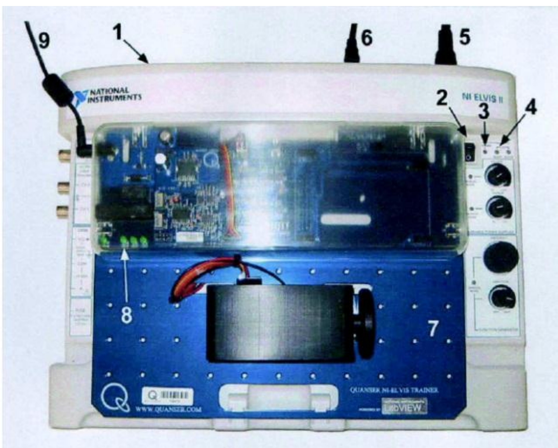

Fig 2: QNET Board Slides with NI ELVIS II

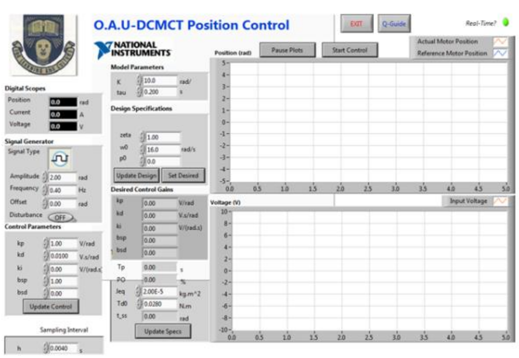

Fig. 3a: The front panel of the position control experiment

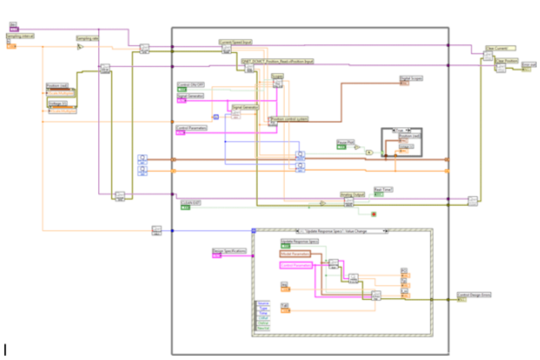

Fig. 3b: The block diagram of position control experiment

About 30% of students complained that they did not have access to a computer which was connected to the internet on which they could install the LabVIEW Runtime to perform the experiments.

The other 70% of the students installed the LabVIEW Runtime, zero percent downloaded the LabVIEW Runtime from the laboratory server. Two students copied the Runtime from us and others got it from them. To ease the LabVIEW Runtime download problem, a copy of the Runtime was placed on OAU's intranet so that students from OAU could download it at up to 8 MB/s and hence download the Runtime in seconds.

The slim installation supports the features necessary for Remote Panels. Using this version would ameliorate the download problems, but not solve it as 28.7 MB is still large for low bandwidth networks.

Another problem which surfaced on the need for the LabVIEW Runtime was the fact that it was practically impossible to perform the experiments from a cybercafé as one would have to have the Runtime installed on the computer there.

No cybercafé around was eager to have some extra software installed on their computers -extra software to further clog their registries and reduce computer speeds bearing in mind that they mostly use minimum spec computers. Hence, as long as LabVIEW clients were to be used, it was decided that students performing the experiments would have to use computers over which they have some amount of control or dedicated computers provided by their faculty or departments. LabVIEW Runtime was integrated well into Internet Explorer, Mozilla Firefox, Opera web browser, Google Chrome and Apple Safari, are all able to load the LabVIEW Runtime to launch the lab client. The laboratory has not been tried in the Linux environment. Users complained they could not covered the whole experiments within the time being allocated and this was due to the traffic on their networks which caused time delay for the Quanser system to response to their commands. Also, a number of users complained that they could not schedule the correct time to launch the lab client interface. It was discovered that differences in the times between the individual computers and the Laboratory-Side Scheduling Server were causing these disparities. This was resolved by telling them to reset the time zone setting on their computer to that of Nigeria (GMT+1).

7. CONCLUSION

An online laboratory has been set up at the Obafemi Awolowo University, available at http://ilab.oauife.edu.ng/sb. The laboratory has three experiments which are performed on a servo control system. The three experiments are position control experiment, speed control experiment and system modeling experiment. The position control experiment introduces users to the workings of a servo control system. The effect of PID controller to reduce the overshoot, steady-state error and settling time were also investigated. The speed control experiment allows students to observe the effect of PI controller to reduce steady-state error on the speed of DC motor.

The system modeling experiment is a medium whereby users specified some set of system parameters to some test input to model the DC motor. From the experimental results, it was observed that a proportional controller (K_p) reduced the effect of rise time and steady-state error. A derivative control (K_d) have the effect of increasing the stability of the system, reduced the overshoot and improved the transient response while, an integral control (K_i) have the effect of eliminating the steady-state error, reduced rise-time but increased settling time. Hence, work lies ahead on ways of minimizing the laboratory's bandwidth requirements while still presenting the necessary visual, auditory and possibly haptic feedback to the user Bethea *et al.* (2004) and minimizing simulations which could give students the feel of being in a virtual laboratory instead of an online laboratory. Experience gained from the setting up of this laboratory showed that the overall development time to deploy a new laboratory could be drastically reduced if LabVIEW is used for the development of the experiment engines and the clients. One reason for this is that one does not need to develop separate applications for the client application and the experiment engine. Secondly, in creating a LabVIEW application, one automatically creates a GUI for it hence; there is no need for creation of a GUI for the client application. The laboratory design was made to be robust so that it is safe for use without human supervision.

REFERENCES

Aaron, K. R., Foster, N. L., Hazel, D. P. and Hasanul-Basher, A. M., "Closedloop position control system using labview", Master's thesis, South Carolina State University, Orangeburg, 2002.

Akinwale, O. B., "Development of a robust Ilab platform for conducting robotic arm experiments", Master's thesis, Obafemi Awolowo University, Ile-Ife, Nigeria, 2010.

Bethea, B. T., Okamura, A. M., Kitagawa, M., Fitton, T. P., Cattaneo, S. M., Gott, V. L., Baumgartner, W. A. and Yuh, D., "Application of Haptic Feedback to Robotic Surgery", Laparoendoscpic Advanced Techniques 3: 191–195, 2004.

Harward, J., Mao, T. and Jabbour, I., "iLab Interactive Services– Overview", http://icampus.mit.edu/iLabs/architecture/downloads, (2006a).

Harward, J., Mao, T. T. and Jabbour, I., "ilab Interactive Services-Overview", (2006b).

Laubwald, E., "Servo control systems 1,visiting consultant", control systems principles.co.uk, (2000).

Imran, A., Hong, W. and Vikram, K., "Internet-based remote control using a microcontroller and an embedded Ethernet board", Department of Mechanical Aerospace and Manufacturing Engineering, Polytechnic University, Brooklyn, NY 11201, 2003.

Quanser, E. T., "QNET Practical Control Guide", Quanser Inc. and National Instrument (NI-ELVIS), 2009.