

A Developed Controller Software for Computer Automation of a Triaxial Testing System

Tanay KARADEMIR

Abstract — A controller software with user interface was developed in Visual Basic that provides user friendly interfaces to control test phases and to enter input test parameters as well as programming capability to control sophisticated and to run high quality triaxial tests for measuring stress-strength-strain properties of soils in obtaining consistent and reliable results from the performed tests. When designing the controller software to administer the entire triaxial test process, the major variables considered primarily and desired to be controlled during the testing progress were: i) boundary drainage conditions (either drained or undrained); ii) imposed stress path; iii) magnitude of axial deformation; and iv) time rate. The software can consistently control the triaxial system applying either undrained or drained linear stress paths as well as performing isotropically (CI) and anisotropically consolidated (CA) or one-dimensionally (CKo) consolidated tests on soil specimens. Additionally, the software enables the user to access measurement instrumentation readings as well as to determine the specimen's current stress-strain state through on-test transducers' measurements.

Keywords — Automated Controller Software, Computer Automation, Triaxial Testing System, Visual Basic Code, User Interface

I. Introduction

Research in soil mechanics is primarily concerned with the development of theories for strength and deformation predictions as well as the advancements of laboratory and field techniques for obtaining proper soil design parameters. To this end, precise and accurate control as well as consistent operational performance of laboratory testing devices is one of the most important issues in geotechnical laboratory practice related to obtaining reliable design parameters from laboratory test results to be able to predict soil strength and deformation properties under loading in the field.

II. Background and Relevant Literature

Commercially produced, computer controlled soil testing equipments are in lack of certain operational performance capabilities in terms of running sophisticated and research oriented long-duration stress-path tests particularly on soft soils (i.e. clays) for which performance tests to investigate the stress-strength-deformation behavior takes significant time duration and the testing duration consists of different test phases (i.e. test stages). With advances in technology and reduced costs of electro-mechanical devices and personal computers (PC), many researchers in the past developed automated triaxial testing systems specialized for their research purpose to obtain accurate and robust test data to be used in the subsequent steps of their study. Atkinson et al. [1], Baldi et al. [2], Menzies [3], Ampadu and Tatsuoka [4] and Pradhan et al. [5] are among others. However, it is noted that their efforts were mostly on the design of testing equipment components and the development of supplementary hardware for the automation of triaxial systems. Little attention has been given to the development of user friendly controller programs incorporating user interfaces and designed menu options in order to: i) control the current state of testing, ii) proceed through different test stages, and iii) enter initial test input parameters as well as specimen properties for attaining reliable test results and accurate characteristic curves in terms of stress-strength-deformation (i.e. strain) behavior of the soil specimens tested. To this end, developing an up-to-date controller software with user interface for controlling the triaxial testing equipment and for administrating different triaxial test phases during overall testing progress will accompany with those previous efforts that were primarily on the computer-automation hardware development and related to the test instrumentation component design works for triaxial systems.

Tanay KARADEMIR
Istanbul Bilgi University
Department of Civil Engineering
Istanbul, TURKEY

III. Controller Software and Computer Automation

A. Introduction

The controller software developed to perform entire triaxial testing process was coded in Visual Basic platform based on the principles including: i) control test phases, ii) display test progress on screen, and iii) allow information interactions with the user. Further, it is noted that the software enables the user to access measurement instrumentation readings as well as to determine the specimen's current stress-strain state during the course of the entire testing duration.

B. Software Execution Procedures and Operational Functions

The Visual Basic environment provides user interface and programming capability to develop robust, user-friendly software. The programmer can easily create interfaces to enable interaction by developing form modules, used to “fire” or initiate tasks handled by the code existing in class modules. The primary functional programming and controlling code is written by creating these class modules. In addition, a main module can be developed to define public variables in software, which are those variables that can be used by all of the class modules. Each class module contains variables, functions and subroutines constituting the controlling software. The major difference between functions in form modules and those in class modules is that a function in a class module can be accessed and used by other interfaces and class modules; this is the prototypical characteristic of object-oriented programming. The functions of a form can only be accessed by the form itself.

The triaxial system controlling software is organized into three parts: i) the main module, ii) class modules, and iii) forms. The main module defines variables, and declares functions and objects in order to access and control the A/D and D/A conversion devices. The controlling software for all test phases of triaxial testing process was developed using class modules. In addition, communication with A/D and D/A boards, and their conversion processes are executed through class modules. Forms were designed as user interfaces to initiate the process being handled by class modules according to pre-scheduled parameters. In other words, forms enable user interaction directly to the test process and progress. In order to progress through different test tasks, the buttons on forms are selected, leading to the next stages of testing. When the software is first run, a control panel appears to prompt the user to proceed to the test information interface, where instrumentation zeros and conversion factors, specimen dimensions, motor performance (“gain”) factors, and other system information is entered. The user can then return to the main control panel to process test stages. Figure 1 shows a comprehensive flowchart for the controller software in which the operational principals of the controller program execution is depicted.

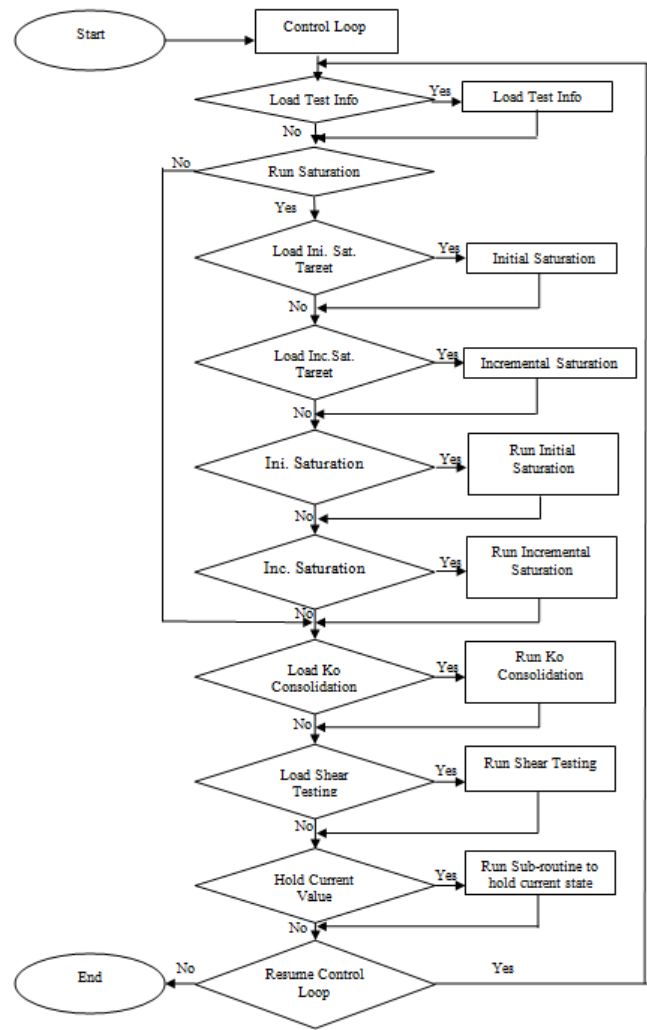


Figure 1. Flowchart of The Controller Software

C. Testing Current State Control and Test Continuation

The process of test control primarily involves the following. The software makes a comparison between the specimen's current stress-strain-time state and the pre-scheduled values. It then determines corrective actions to send to the control motors to close any gap between the actual and scheduled state at a particular time. As such, after checking the current state of testing progress and the test environment (triaxial cell), the correction process is accomplished through an empirically derived, linear relationship. For instance, the cell pressure must be constant throughout either undrained or drained shearing. If the cell pressure deviates from this constant value, the motor controlling the cell pressure PVCC (pressure-volume-control-cylinder) is actuated to correct the error, as follows:

$$\text{Required-Cell-Steps} = (\text{Cell-Pressure-Gain}) * (\text{Cell-Pressure-Error}) \quad (1a)$$

Where:

$$\text{Cell-Pressure-Error} = (\text{Current-Cell-Pressure}) - (\text{Pre-Scheduled-Cell-Pressure}) \quad (1b)$$

and Cell-Pressure-Gain, that is in units of stepper motor steps per 1 kPa change in cell pressure, is a constant and pre-determined during calibrating the system for the step motor controlling cell pressure. The gain value is input when test process information is entered into the controller software prior to initiating test progress.

As per controlling the specimen's axial deformation (i.e. displacement) at a constant rate, the number of motor steps for the loading frame is determined from a calibration of load frame vertical displacement versus motor steps, and the user-defined strain rate. Equation 2 gives the computation method for the calculation of specimen axial deformation with time.

$$\text{Required-Steps} = (\text{Axial-Deformation-Per-Steps}) * (\text{Total-Time-Interval}) \quad (2a)$$

$$\text{Axial-Deformation-Per-Steps} = (\text{Loading-Gain}) * (\text{Strain-Rate}) * (\text{Specimen-Height}) \quad (2b)$$

$$\text{Total-Time-Interval} = (\text{Current-Time}) - (\text{Last-Comparison-Time}) \quad (2c)$$

Where; Loading-Gain is a pre-determined value from the calibration of load frame vertical displacement versus motor steps.

D. Controller Program Module for A/D Hardware Driver

Two class modules were written to interface with the A/D and D/A converters. The code to read the measurement instrumentation voltages via the A/D converter was developed according to the register-level I/O map of the A/D conversion device. In the computer I/O space, the device occupies four consecutive base addresses. In addition, the A/D conversion card utilized offers three different voltage gains, which are 1, 10 and 100. These gains are software programmable for the input ranges of 0 to 3.276 Volts. A gain of 1 was set for all 8 channels to reduce the reading error. For a gain of 1, the input voltage of 0 to 3276 mV can be measured with an error of 100 mV, for gain 10, 0 to 327.6 mV with error of 10 mV, and gain of 100, 0 to 32.76 mV with error of 1 mV.

The steps that the program follows for converting analog signals to digital form are as follows. The status of the register is initially checked to see if the A/D converter is busy or not. If not, the program then writes to the device control set the channel and gain to be read. In order to initiate A/D conversion, the program writes to the A/D register base address for interrupt mode and polarity data. Then, the high byte and sign in base address and low byte in base address are obtained, and later, these were assembled into one signed data word. In the control software, the class module called A/D conversion controls and performs the operations mentioned

above (Figure 2). Further, in order to write and read registers, DLL functions of D1PortReadUchar and D1PortWriteUchar provided by the A/D card manufacturer are used. The purpose of this function is to scan 8 channels twice consecutively and then compute the average of two readings. In this way, errors in data conversion can be compensated. The card can perform 15 conversions per second based on the acceptable time interval of a control loop cycle.

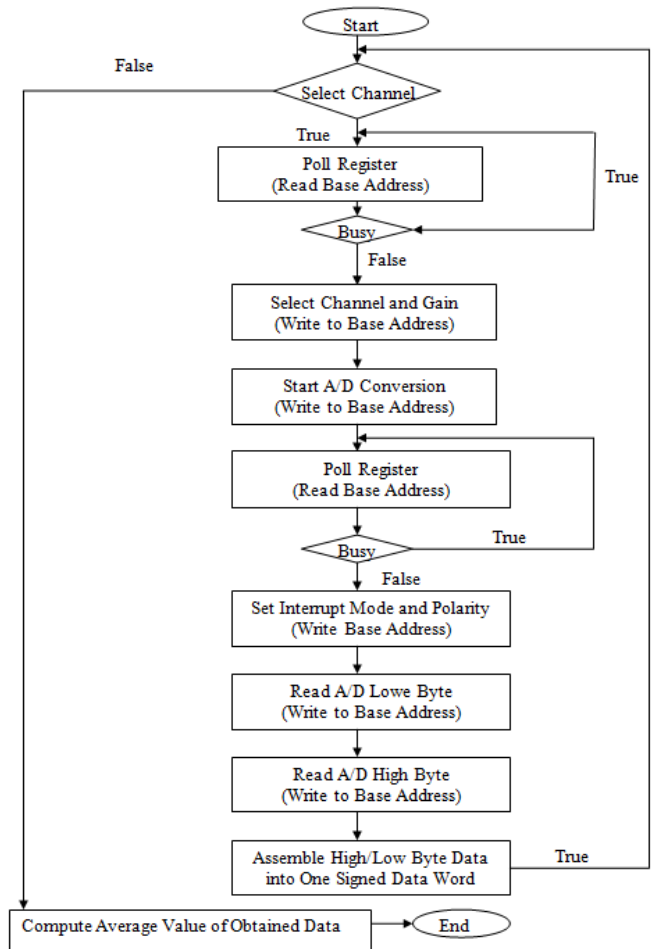


Figure 2. Algorithm of Class Module For A/D Conversion

E. Controller Program Module for D/A Hardware Driver

The class module called D/A conversion (Figure 3) interacts with motor controller card in order to actuate and drive cell pressure, back pressure and axial load motors. Initially, the motor controller card needs to be opened to enable communication of software with the controller card. Further, this class module converts the number of steps and direction to move, generated by the control algorithm, into card-compatible commands by using the command library and the DLL functions, provided by the card manufacturer, to access the card in Visual Basic environment. Then the interaction between the module and the controller card is

enabled to send the converted commands to the motor controller card board. After that, the software checks for the feedback information as a response from the controller card to double-check if the preceding command was performed successfully. Otherwise, the volume of the buffer could be enlarged. Later, motors start to move according to the number and direction of steps. Finally, the controller card is closed for the completion of one-time access. It is estimated that the total time for A/D conversion, processing through software and D/A conversion for the movement of motors takes 3 seconds, meaning that the closed feedback control loop of automated testing system completes in seconds.

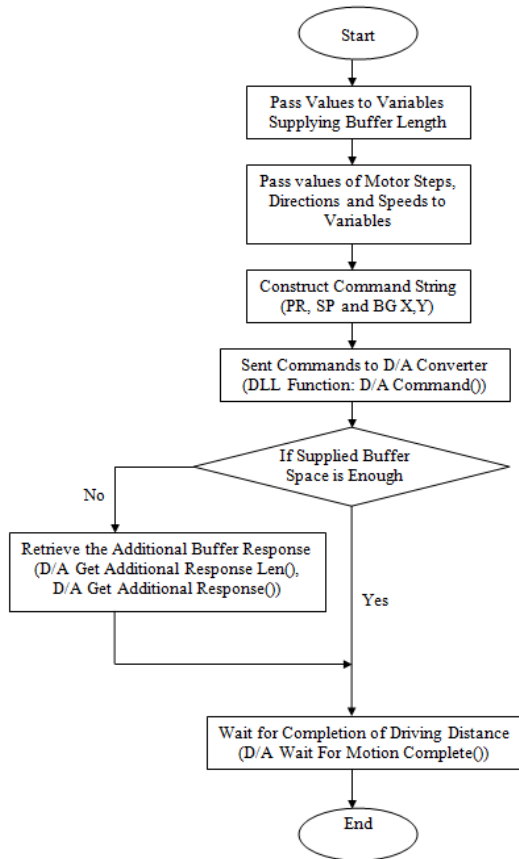


Figure 3. Algorithm of Class Module For D/A Conversion

iv. Conclusion

This paper provides detailed knowledge on the development of controller software with user interface to administer different test phases as well as to control computer-automated triaxial testing system. Further, a detailed description on programming components was provided and the functionality of the controller software was discussed. Those programming components were particularly developed to facilitate consistent long-duration stress-path testing of soils with accurate test-duration readings obtained through robust transducers' measurements for on-time determining current

state of testing progress regarding stress-strength-strain behavior of the soil specimen tested.

The controller program coded and developed in Visual Basic platform has shown its capability in accomplishing multiple functions and duties simultaneously when controlling test processes, including: i) controlling test phases; ii) displaying test progress on screen; and iii) allowing information interactions with the user. As such, the triaxial specimen stress-strain state is measured by the instrumentation, and the instrumentation signals are read into the PC using an analog-to-digital (A/D) converter, and these signals are converted to engineering units. This current actual stress-strain-time state is compared with pre-scheduled values. If a difference occurs between these two at a given time, the control algorithm computes and actuates the stepper motors via the digital-to-analog (D/A) device to minimize or eliminate any differences that exist. In this way, the triaxial tests with consistent controller software performance, and thus equipment stability, particularly during long testing durations, was achieved and resulted in obtaining repeatable, reliable test data through the performed triaxial tests on the soil specimens.

References

- [1] Atkinson, J.H., Evans, J.S. and Scott, C.R. (1985). "Developments in microcomputer controlled stress path testing equipment for measurement of soil parameters." *Ground Engineering*, 18(1), 15-22.
- [2] Baldi, G., Hight, D.W. and Thomas, G.E. (1988). "State-of-the-Art: A re-evaluation of conventional triaxial test methods." *Advanced Triaxial Testing of Soil and Rock*, ASTM STP 977, Robert T. Donaghe, Ronald C. Chaney, Marshall L. Silver, Eds., ASTM International, West Conshohocken, PA, 219-263.
- [3] Menzies, B.K. (1988). "A computer controlled hydraulic triaxial testing system." *Advanced Triaxial Testing of Soil and Rock*, ASTM STP 977, Robert T. Donaghe, Ronald C. Chaney, Marshall L. Silver, Eds., ASTM International, West Conshohocken, PA, 82-94.
- [4] Ampadu, S. and Tatsuoka, F. (1989). "An automated stress-path control triaxial system." *Geotechnical Testing Journal*, 12(3), 238-243.
- [5] Pradhan, T.B.S., Tatsuoka, F., Mohri, Y. and Sato, Y. (1989). "An automated triaxial testing system using a simple triaxial cell for soils." *Soils and Foundations*, 29(1), 151-160.

About Author:



Tanay Karademir, Ph.D. is an Assistant Professor in the Department of Civil Engineering at Istanbul Bilgi University in Istanbul, Turkey. His research interests are in the field of Geotechnical and Geoenvironmental Engineering under the profession of Civil Engineering.