Development and Experiment of A Real Time Control System Using LabVIEW

Qing-Guo Wang, Tong Heng Lee, Jia Hong Shu and Kian Chee Ang
Department of Electrical Engineering
National University of Singapore

Abstract **- General commercial distributed control systems (DCS) have very specific functions and do not allow user to configure or modify the system. In addition, DCS usually do not provide useful information like the plant model parameters. In this paper, a user-friendly general-purpose real time control software that operates in the Windows environment will be presented. This software is able to control up to four channels of any plant linked to National Instruments DAQ hardware and allows user to configure it for convenience of research and development. The software is developed using graphical programming language, LabVIEW, a product from National Instruments.**

I. INTRODUCTION

Over last several decades, automatic control has played a vital role in the advance of engineering and science. As the dynamic characteristics of most control systems are not constant for several reasons, such as the deterioration of components as time elapses or the change in parameters and environment. Technology for adaptive control of industrial process in frequency domain has a great deal of appeal to the system designer [1]. As an adaptive control system, besides accommodating environmental changes, will also accommodate moderate engineering design errors or uncertainties and will compensate for the failure of minor system components, thereby increasing overall system reliability.

Adaptive control system has being comprehensively studied and extensive applied to various places such as cleanrooms, molding machines and even paper machines. However, these commercial distributed control systems (DCS) are usually catered to perform certain specific functions and do not have the facilities for upgrading or extensive modification. Thus, there is a lack of general-purpose real time control software to implement various adaptive control technologies for research and testing purposes. The rapid growth of Internet in recent years has also promoted the need of general-purpose control software. As with such software, control students and engineers will be able to configure instruments and data analysis remotely via software, thus facilitate sharing of expensive equipment, and may well be the next important step in remote distance learning.

In this paper the software will be developed using LabVIEW. LabVIEW is a revolutionary G programming language that is developed by National Instruments. LabVIEW has the flexibility of a powerful programming language without the associated difficulty and complexity, as its graphical programming methodology is inherently intuitive to scientists and engineers [2].

II. SYSTEM PLATFORM

The first concern of any software implementation is selection of operating platform. Since Microsoft Windows 95/98 operation system is currently the most popular operation system, this project should be developed based on its architecture. LabVIEW in combination with Windows 95/98/NT enpowers our development and enables user-friendly interface. For the compatibility to LabVIEW, PXI-1000B compact PC, which is also developed by National Instruments and shown in Fig. 1, is chosen as our computer for signal processing and real time control due to its robustness and flexibility [3].

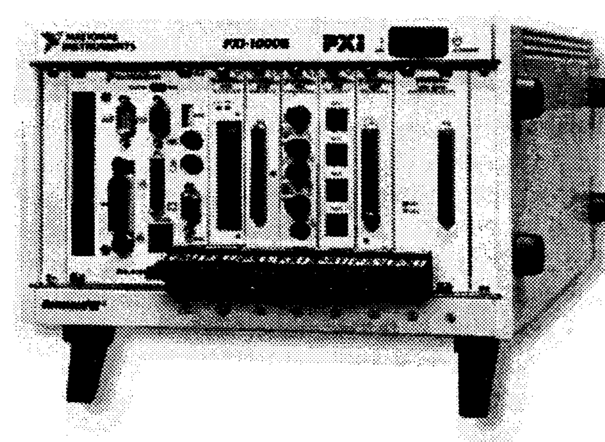

Fig. 1. PXI-1000B Compact PC

III. SOFTWARE DESIGN

Software design involves the design of software structure, which specify the functions of software and dataflow. This structure is also be used as a reference to ensure that the required functions specified are achieved in the future phases. As the nature of this project is very complex, it would be most efficient to divide the overall software (system) into manageable, logical pieces called modules. These modules will have clearly define functions which perform different tasks specify by users. Modular approach also promotes division of labor and the potential for code reuse. It is also easier to implement, maintain and test the software at module level then at the system level. Since it is easier to catch bugs in one small module than in a hierarchy of several modules.

The interrelations of individuals module in this software can be represented in the form of a structure chart. A structure chart is a hierarchical, tree shaped diagram in which all the modules in the project are represented as rectangular boxes. The calling relationships between modules are represented in the form of arrows connecting the boxes. Base on the software requirement, the software is broken into 12 separate modules and is related by the structure chart shown in Fig. 2.

The description of the individual module is as follows,

i) *Configuration*: This module allows user to configure their hardware and initialize the real time control software.
ii) *Main Menu*: This module is the core of the software as it will handle all the data acquisition, data computations and coordination between others modules.
iii) *SP*: This module allows user to control and monitor set point (SP) individual channels.
iv) *E*: This module allows user to monitor the noise band and errors of individual channels.
v) *MV*: This module allows user to control and monitor manipulate value (MV) individual channels.
vi) *PV*: This module allows user to monitor individual process value (PV) channels.
vii) *G*: This module allows user to set or alter the plant models for simulation purpose.
viii) *K*: This module allows user to set or alter the controller parameters.
ix) *Control*: This module allows user to simultaneously monitor or control individual channels MV, SP and PV.
x) *Plant Identification*: This module allows user to identify plant models with available data files.
xi) *Controller Design*: This module allows user to formulate controller parameters based on current plant models.
xii) *Auto-tuning*: This module allows user to auto-tune their process.

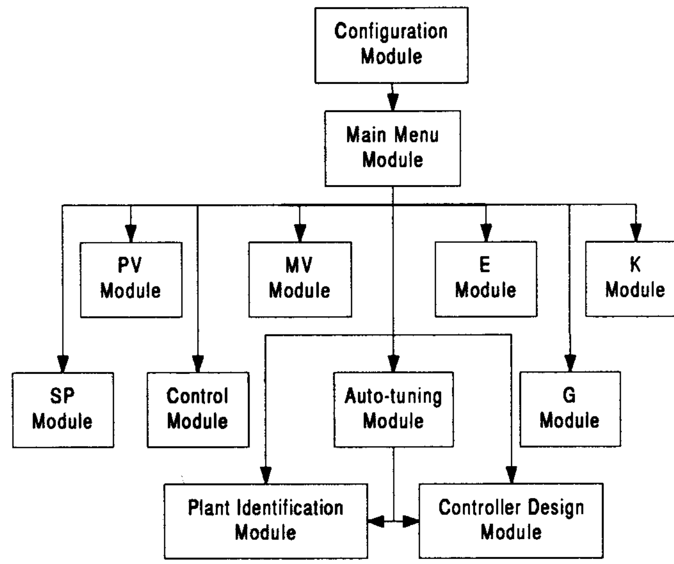

Fig. 2. Structure chart of Software

IV. SOFTWARE ILLUSTRATION

Besides providing functionality for users, this software also has non-functional requirements. For instance, the software should be easy to use and learn. As such, the structure and style of the software should be familiar to the users. In the implementation of the user-interface, various interaction styles and layout were considered, ensuring that this software is both user-friendly and easily reconfigurable. In addition, to ensure a close correspondence to Windows concepts, this software has also incorporated Windows application program interface API [4]. Descriptions and illustration of the user interface will be further elaborated in the following sections.

a) Configuration module

This will be the first module that the user encounter, as the name implies, this module allows user to specify the number of channels to control, set the sampling rate and scaling factor as shown in Fig. 3. By using the hardware setup option, user will be given the flexibility of matching their National Instruments peripheral ports or channels to the respective PV and MV. In Fig. 4 below, the device columns refer to the specific National Instruments DAQ cards and the channel columns refer to the specific input or output channel of that corresponding DAQ card.

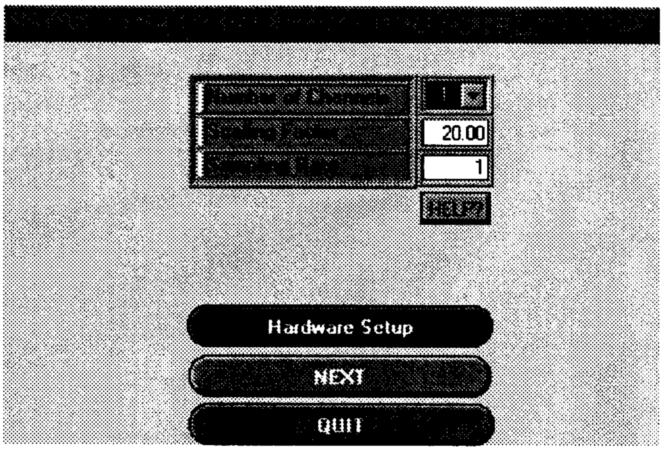

Fig. 3. Configuration display

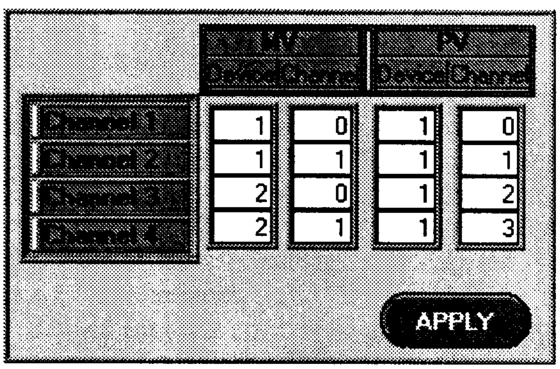

Fig. 4. Hardware setup display

b) Main Menu module

When the user click "Next" controls button in the configuration module, this module will be loaded. As shown in Fig. 5 below, user is allowed to perform various operations like setting SP, MV, PV, K and G of an actual closed loop system by clicking on the respective control buttons. User can also specify the controller type (PID/Higher order) or change the simulation mode. In addition, user can select functions like plant identification, controller design and auto-tuning using their respective control buttons.

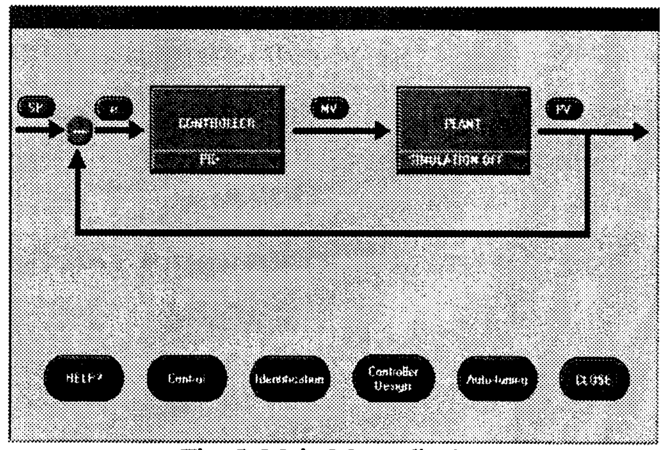

Fig. 5. Main Menu display

c) SP/MV/PV module

From Fig. 6, user is able to specify which channel SP/MV/PV to monitor or control by selecting the respective button inside the channel selection clusters. The SP value of the current selected channel will be display on both the graph and the bar indictor/control.. To control the current SP or MV value, the bar indictor/control can be used by moving the sliding bar or entering the new SP or MV value into the numeric control box. User can also simultaneously change the SP or MV values for all the channels by clicking the "syn-all setup" control button. This feature can be useful for the study of coupling between different channels. However, this synchronization option is only valid if all the channels are either in auto or manual mode. User can also print out the current SP/MV/PV graph by clicking the "print graph" control button. Lastly, to terminate this module, user just need to click the "close" control button.

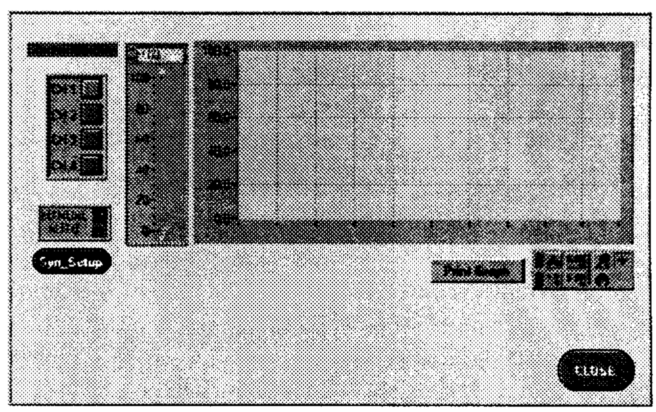

Fig.6. SP/MV/PV display

d) Error module

As Fig. 7, User can monitor error of individual channel by clicking the respective channel control button inside the channel selection cluster. The noise band, PV maximum and PV minimum of the individual channel will be displayed in their respective numeric indictors. However, the user is allowed to reset the noise band, PV maximum and PV minimum by clicking the "reset" control button.

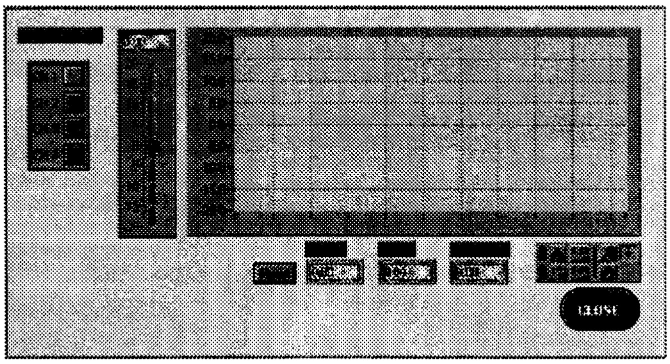

Fig. 7. Error display

e) Plant module

When plant module is chosen, user will see Fig. 8 below. This module enables user to simulate up to 10^{th} order multivariable plant models. To change or view these plant model parameters, user must click the desired plant selector. The selected plant model, which will be indicted by the current model indictor, can then be view or changed. Furthermore, an "apply" check box is implemented into this module to give user an option of terminating without saving the plant model.

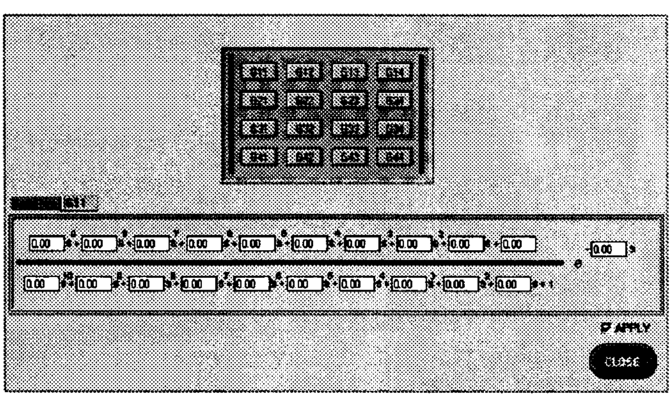

Fig. 8. Plant display

f) Controller module

When controller module is chosen, the user will see Fig. 9 and Fig. 10 below. As shown in Fig. 9 and Fig. 10, depending on the controller type selected in the Main Menu module, this software can accept either PID or up to 10^{th} order controller.

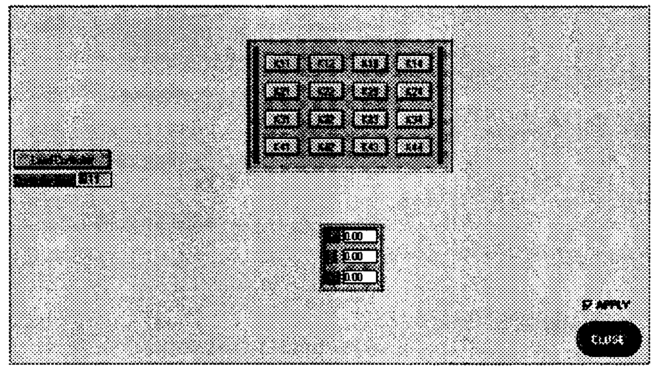

Fig. 9. PID controller display

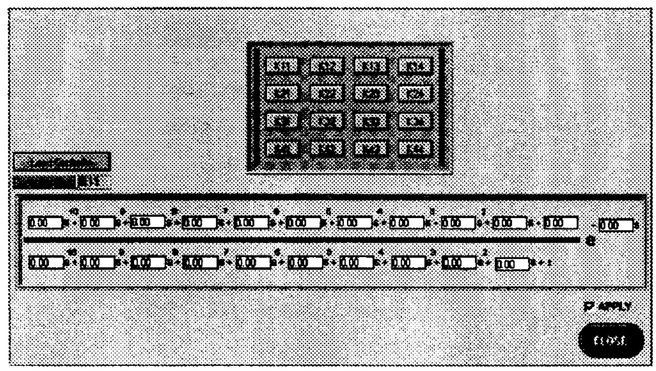

Fig. 10. Higher order controller display

g) Control module

From Fig. 11 below, user is able to specify which channel SP, MV & PV to monitor or control by clicking the respective button inside the channel selection clusters. The bar indictors/controls could be used to monitor or change the current SP or MV value, by moving the sliding bar or entering the new SP or MV value into the respective numeric control box. User can also simultaneously change the SP or MV values for all the channels by clicking the "syn_setup" control button.

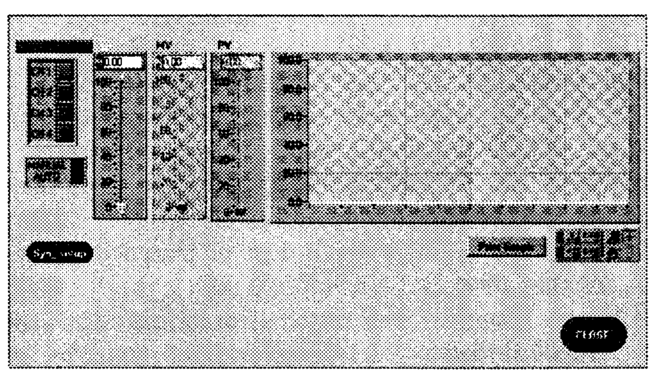

Fig. 11. Control display

h) Controller Design module

When this module is initially loaded, user will see the display as in Fig. 12 below.

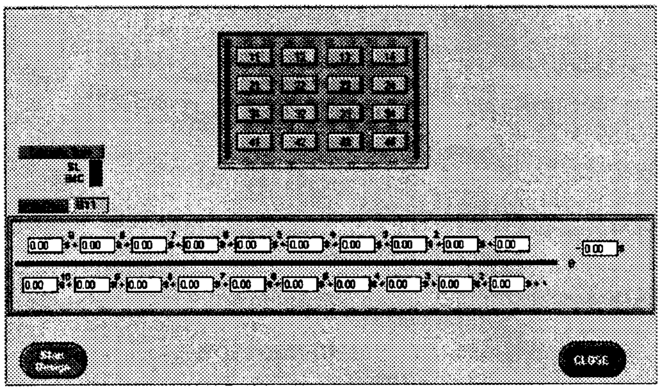

Fig. 12. Initial Controller Design display

After verify that all the plant model parameters are correct, user can click the "Start Design" control button to activate the Matlab algorithm for the formulation of controller parameters. After this phase, the display will change to depending on the controller type selected in the Main Menu module "Controller" control button, show as fig. 13.

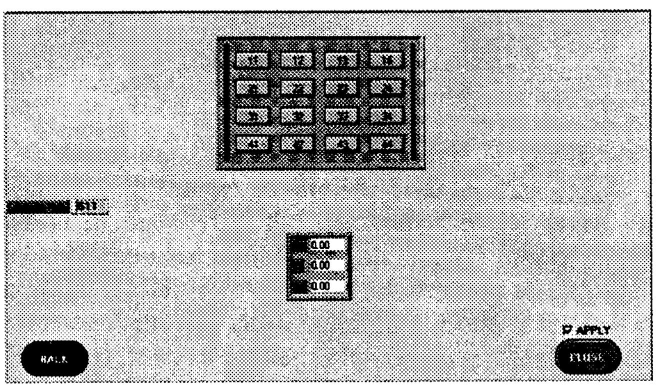

Fig. 13. PID controller display

i) Plant Identification module

This module allows user to identify 2nd or 1st order plant model using available data files. As shown in Fig. 14, user must first initialize the path of these data files, order of plant model to be identified (1st/2nd order), test type of data (step/relay) and data type (minimum/non-minimum). After user clicked the "Start Identification" control button, the plant identified using respective Matlab script will be shown in Fig. 15.

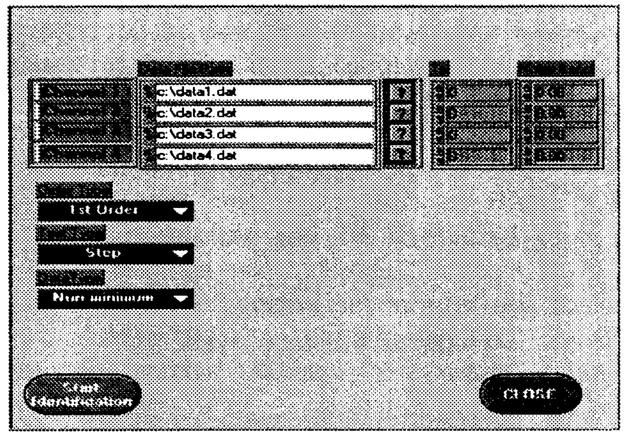

Fig. 14. Initial plant identification display

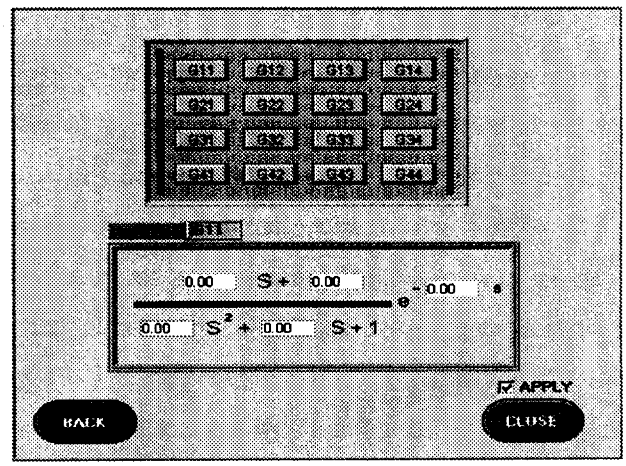

Fig. 15. Plant identification result display

j) Auto-tuning module

The module user interface consists of 4 steps. These four steps are Configuration, Acquisition, Plant Identification and Controller Design. In addition from section II, this module algorithm integrates both the Plant Identification module and Controller Design module. To enhance user-friendliness, the user interfaces of these two modules are incorporated into the 4 steps. In Fig. 16, the first step of this module requires user to configure the paths of data files for data files that will be created in step 2, type of test signals to be applied (step/relay) and amplitude of test signals.

After the "Next" control button is clicked, user is then given an option of sequencing the test signals that are applied to the different channels. User can also chose to return to step 1 by clicking on the "Back" control button. Once a channel has been selected, the data acquisitions will began and will only stop if the "Stop" control button is clicked as shown in Fig. 17.

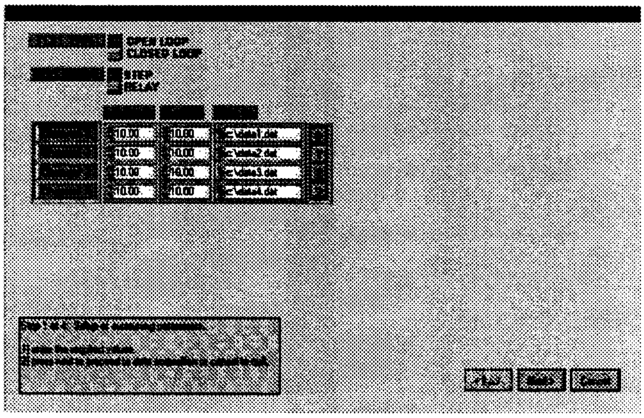

Fig. 16. Auto-tuning step 1 display

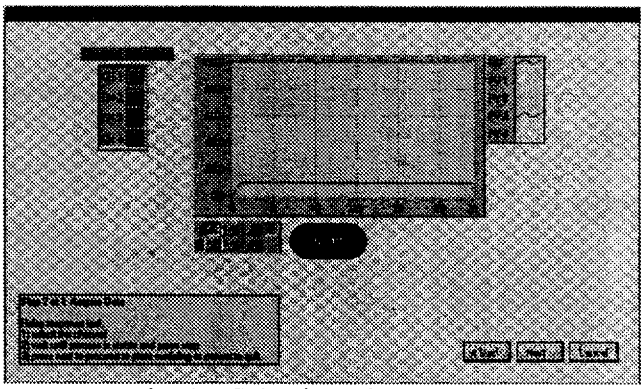

Fig. 17. Selection of test channel

When the "Stop" control button is clicked, the display will change to Fig.18. User is then given an option of specifying the data range to be saved. To enhance the user-friendliness an "undo" control button is added to allow easy undo of data range specified. After the "apply" control button is clicked, user can then choose to proceed to step 3 by clicking "Next" control button, return to step 1 by clicking "Back" control button or terminating this module by clicking "Cancel" control button.

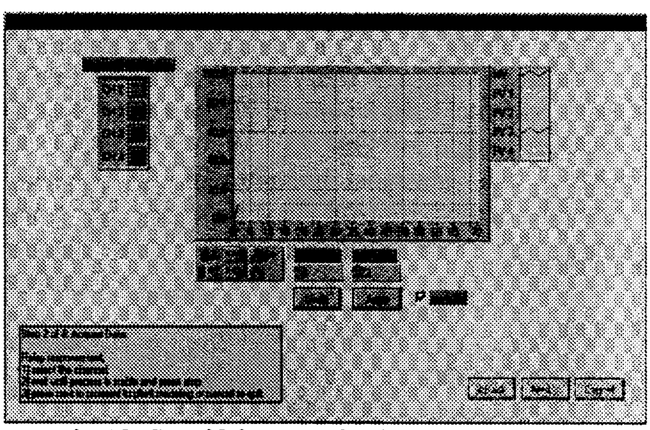

Fig.18. Specifying acquired data range

In step 3, the Plant Identification module will be activated and the data files that are created in step 2 will be used for the plant identification process. The default multivariable plant model identified is of type 2[nd] order. However, user is allowed to change these plant parameters.

In the last step, either PID or Higher Order controller parameters will be formulated using the multivariable plant model identified in step 3. Lastly, by clicking the "Next" control button, user will terminate this module or "Back" to return to step 3 or "Cancel" to terminate without saving the current controller parameters.

V. AN EXPERIMENT

Fluid level control is important in chemical process. It aims at keeping the fluid level at a certain value in storage tanks, chemical blending and reaction vessels. A coupled tank system was employed for illustration of our system. First step was to set the testing parameters. We chose a step test, as shown in Fig. 19. The step response is displayed in Fig. 20. These data were processed by modules of our identification and controller design to yield a controller. The controller was then commissioned and the resultant control performance is shown in Fig.21.

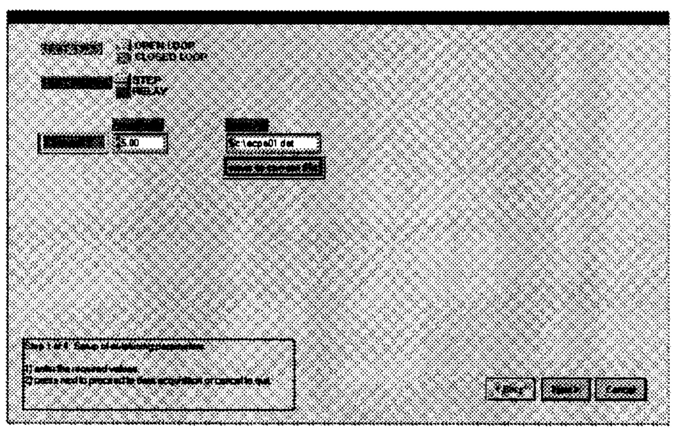

Fig. 19. Setup the testing parameters

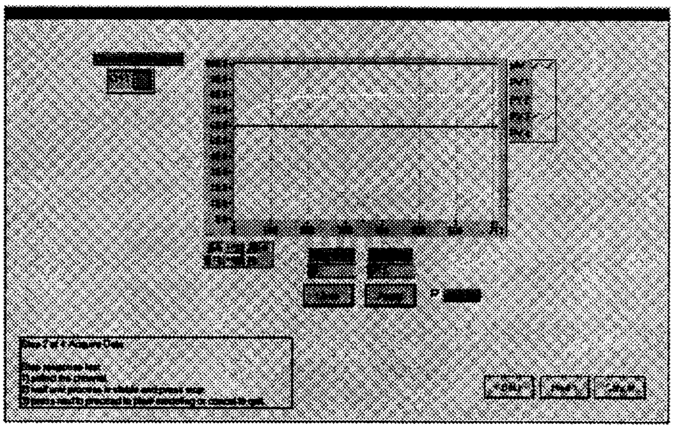

Fig. 20. The open loop step response of couple tank

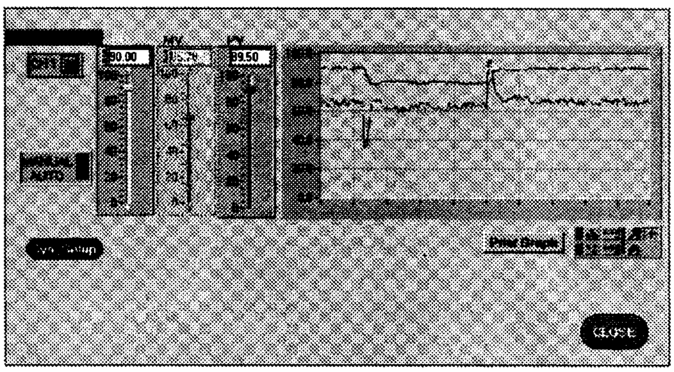

Fig. 21. The trend of couple tank

VI. CONCLUSION

This project involves the development of a user-friendly, easily restructurable general-purpose real time control software, something that is lacking in commercial distributed control system (DCS). The software will enable user to control up to four channels of a physical plant connected to any National Instruments Data Acquisition (DAQ) devices. By using National Instruments LabVIEW graphical programming language, user-friendly graphical user interface is built. The system is easy to use without preassumed knowledge of programming languages and advanced control; yet, users can apply advanced control functions available to enhance their modeling and control performance.

REFERENCES

[1] K. OGATA, "Modern Control Engineering", 2nd edition, Prentice-Hall, Singapore, 1998.

[2] National Instruments Corporation, "User Manual", 1998 edition, LabVIEW Version 5.1 CD, 1998.

[3] National Instruments Corporation, "Measurement and Automation Catalogue", 2000.

[4] National Instruments Corporation, "G Programming Reference Manual", 1998 edition, LabVIEW Version 5.1 CD, 1998.