

## **VB for Control & Instrumentation Program**

**W. Feng, A. Gomez-Rivas**

**Engineering Technology Department  
University of Houston-Downtown  
One Main Street  
Houston, TX 77002**

### **Abstract:**

This paper assesses the unique role of Visual Basic (VB) in the curriculum development for a four-year degree program in Control & Instrumentation Engineering Technology. A VB program is presented which was developed in-house and has been used to demonstrate the fundamentals of control system theory and techniques. Also presented is a course project in which a VB program was successfully used for a real-time temperature control system. It is concluded that, by embedding VB programming to the Control and Instrumentation curriculum, we have created an effective and efficient teaching and learning approach and, as a result, students not only have gained the technical knowledge but also have significantly developed their skills for computer applications.

### **I. Introduction**

Over the last two decades we have seen a strong growing trend of computer applications, especially PC applications, in the control and instrumentation industry. In response to the needs of current industry and graduate job market, we have revamped our old Electrical Engineering Technology program into a new Control and Instrumentation Engineering Technology program, by shifting the emphasis from hardware and component oriented courses to hardware/software interfacing and integrated systems oriented courses.

Visual Basic (VB) has become the most widely used programming language for its easiness of graphic user interface and language features. VB has been used for application developments in business as well as science and engineering communities<sup>1</sup>. With each new edition, new features have been added to the language and it is becoming increasingly powerful. When our students start their freshmen level, a significant number of them have little experience with computer usage. Some of them are even reluctant to touch the computer keyboard. It is at this point that they are introduced to computer applications via VB programming, and the results are amazing. After one semester, students' confidence with computer applications has been boosted and they become eager to apply and expand their knowledge and skills in computer programming. In view of this

situation, it only logic to continue with this momentum, and use VB as a programming tool for teaching and learning to maximize students' learning outcomes.

In the spirit of this philosophy, a VB based Control Fundamentals Tutorial has been developed for the course Automation and Control. The tutorial covers time-domain as well as frequency-domain analysis and design. Basic concepts and techniques are embodied in the tutorial. At the clicks of mouse buttons, the cause-effect relationship can be instantly revealed and studied among various related items, such as first- and second-order dynamics, dc gains and time constants, poles and zeros, Bode and Nyquist plots, root loci, step or ramp response. As the source codes of the tutorial program are made available to students, they can better understand the underlying computational issues, the typical example is the frequency response and Bode plots which will be discussed in more details in the following section of this paper. How to implement a continuous control law in a computer controlled system? Here is the clue. Just by studying the segment of the codes for the closed-loop control system simulation, student can see why and how the continuous control law can be discretised for implementation in a digital environment. It should be pointed out that throughout the course, students are involved into more intensive learning activities than just clicking the mouse button and playing with the tutorial program. They are required to develop a similar VB program on a smaller scale, so the traditional hand-calculations and plot sketching are largely replaced by programming exercises. As a result, students have the opportunity to gain more valuable computer programming experience while learning various topics of Automation and Control. At the final stage of the course, they use their newly gained knowledge and skill to carry out their course projects, and the experience turned out to be the most beneficial and rewarding.

In an industrial process control environment, computers are used for process monitoring, data-logging, and real-time control. For the Process Control Systems course projects, students are required to build a simple computerized process control system. The process involves a temperature control problem, and K-type thermocouples are used for temperature measurements. A VB program is used for user interface, data acquisition, and process control. The project demonstrates that, for process control systems where the dynamics is relatively slow, VB programs can adequately be used for real-time control purpose. The details of the project are presented in the Section III of this paper.

We are at the early stage of this experiment of using VB for the control and instrumentation curriculum. The conclusion so far and some of the thoughts for future work will be discussed in Section IV.

## II. The Control Fundamentals Tutorial

It is widely agreed that the classical control theory and analysis methods involve a significant amount of mathematics that can be tedious and boring to learn as well as to teach. As an effort to initiate an effective and efficient teaching and learning approach, various software tools are adopted as learning aids for Control Engineering related

courses. The most common one being used at a number of schools is the MATLAB/SIMULINK environment. MATLAB has the advantages of requiring short learning curves and large collections of functions and toolboxes. However, we decided to use VB to develop a tutorial for the course Automation and Control. First of all, VB is a standard and widely used programming language, and our students already have had some VB programming experience, it would be beneficial to consolidate and expand their VB programming experience. Secondly, simplicity and clarity are critical in the tutorial program. Unlike highly cryptic languages, such as C/C++, it is much easier to use simple and straightforward VB codes to implement algorithms and user interface.

The front-end of the tutorial is shown in Figure 1. A standard feedback control block diagram is used and the following elements are included:

1. Plant (constant, first-order, second-order, or transfer function)
2. Compensator1 (P, PI, PD, PID, or transfer function)
3. Sensor (constant, or first-order)
4. Reference (step, ramp, sine wave, or input file)
5. Disturbance(step, sinewave, or input file)

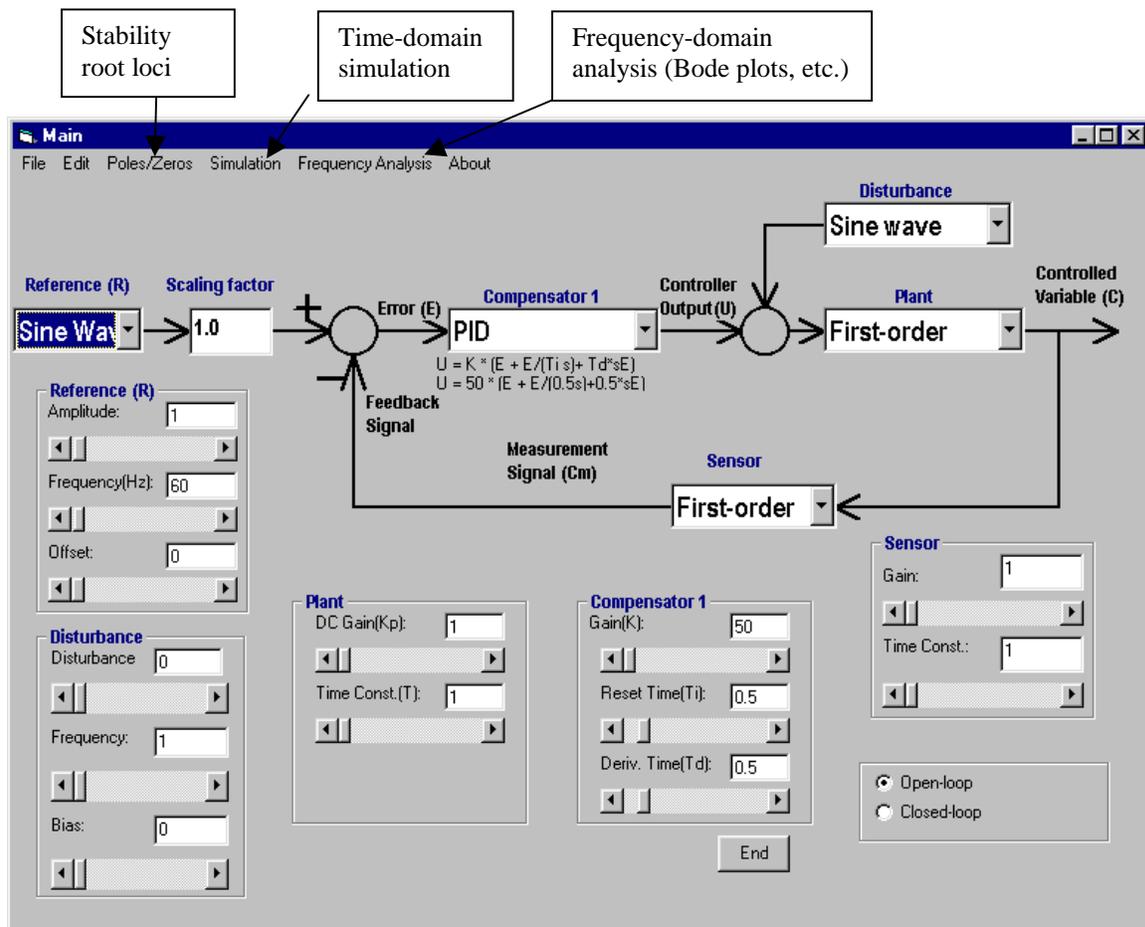


Figure 1: Front-end user interface

As it is in Figure 1, the plant is of a first-order dynamics (with a dc gain of 1 and time constant of 1), the compensator is a PID controller. To set the types of plant, controller, reference input, sensor, etc., just click the corresponding combo box and choose the required items (types). Once element types are selected, the parameter default values of elements appear on the screen automatically, and default values can be changed by rolling the corresponding scroll bars or simply typing in the text boxes. Furthermore, the control system can be configured as either an open-loop or a closed-loop system. To summarize, it takes less than a couple of minutes to specify a complete control system with this user interface.

After the system is defined, analysis can be carried out in its frequency response characteristics, stability, and time domain response. Traditionally, generate frequency response data from a given transfer function for Bode or Nyquist plots can be a tedious task, however, the following code segment shows that it would be done in a more productive manner:

```

` Input frequency range
LowFreq = Val(frmFrequencies.Text1.Text)
HighFreq = Val(frmFrequencies.Text2.Text)
FreqPoints = Val(frmFrequencies.Text3.Text)

` Discretise the frequency range
FreqInterval = Exp(Log(HighFreq / LowFreq) / FreqPoints)
For j = 0 To FreqPoints
    If j = 0 Then
        Frequencies(j) = LowFreq
    Else
        Frequencies(j) = Frequencies(j - 1) * FreqInterval
    End If
CurrentFreq = Frequencies(j)

`Calculate the complex value of the transfer function G(jw)=(c+jd)/(a+jb)
n = TF.DenominatorOrder
m = TF.NumeratorOrder

a = 0
b = 0
c = 0
d = 0

For i = 0 To n
    If (i / 2 - Fix(i / 2)) > 0 Then
        b = b + (-1) ^ ((i - 1) / 2) * TF.DenominatorCoeff(i) * CurrentFreq ^ i
    Else
        a = a + (-1) ^ (i / 2) * TF.DenominatorCoeff(i) * CurrentFreq ^ i
    End If
Next i

For i = 0 To m
    If (i / 2 - Fix(i / 2)) > 0 Then
        d = d + (-1) ^ ((i - 1) / 2) * TF.NumeratorCoeff(i) * CurrentFreq ^ i
    Else
        c = c + (-1) ^ (i / 2) * TF.NumeratorCoeff(i) * CurrentFreq ^ i
    End If
Next I

`Determine the rectangular and polar forms of G(jw):
TF.RealPart(j) = (a * c + b * d) / (a ^ 2 + b ^ 2)
TF.ImagPart(j) = (a * d - b * c) / (a ^ 2 + b ^ 2)
TF.Magnitude(j) = 10 * Log((c ^ 2 + D ^ 2) / (a ^ 2 + b ^ 2)) / Log(10)

If (TF.RealPart(j) = 0 And TF.ImagPart(j) > 0) Then

```

```

TF.Phase(j) = pi / 2
ElseIf (TF.RealPart(j) = 0 And TF.ImagPart(j) < 0) Then
    TF.Phase(j) = -pi / 2
ElseIf TF.RealPart(j) > 0 Then
    TF.Phase(j) = Atn(TF.ImagPart(j) / TF.RealPart(j)) 'First and forth quadrant
ElseIf TF.RealPart(j) < 0 Then
    TF.Phase(j) = Atn(TF.ImagPart(j) / TF.RealPart(j)) - pi 'Second and third quadrant
End If

```

The above code segment is used in the tutorial for calculating frequency response data. Students have been engaged in a similar programming exercise, they programmed for the routine to calculate the frequency response of second-order transfer functions. The Bode plots of a second-order under-damped system are displayed in the fashion as shown in Figure 2 (a resonance frequency of 9.55 rad./s is indicated).

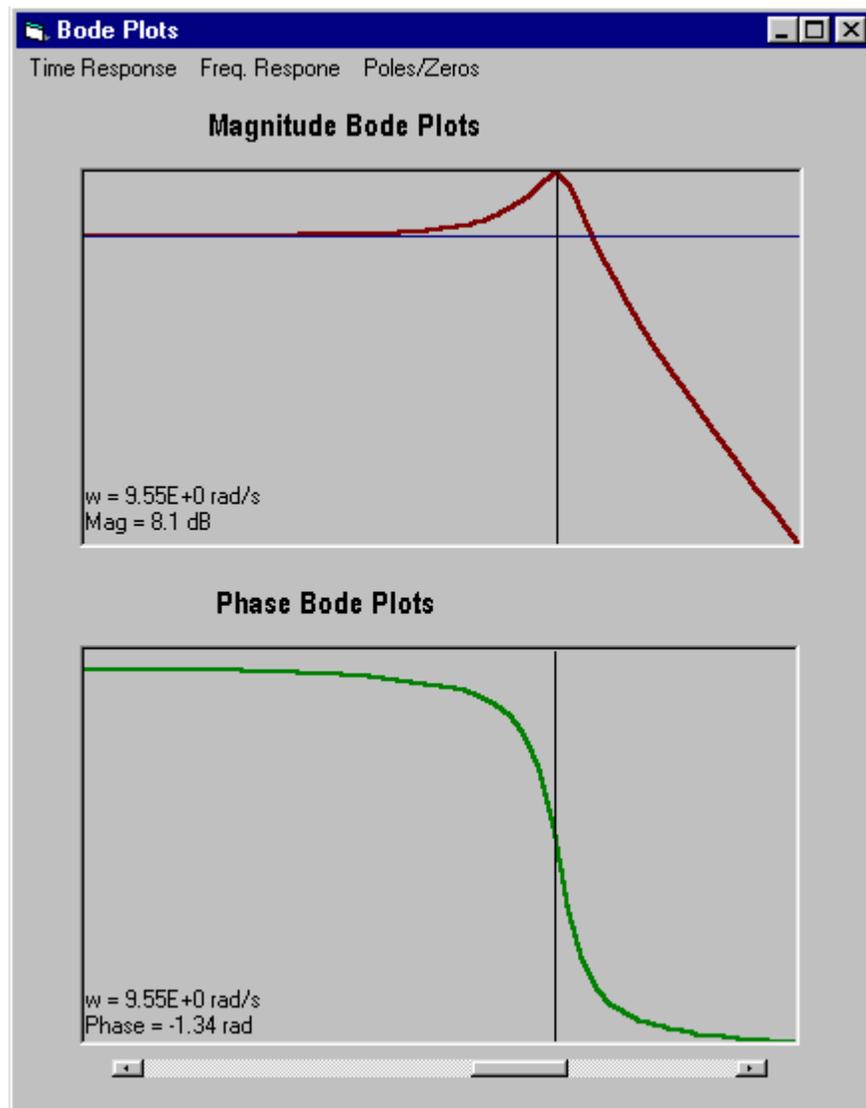


Figure 2: Bode plots

It is worth noting that the horizontal scroll bar at the bottom of the graph display is used to display the frequency response data at a specific frequency point. This feature can be

used to find out dc gain, gain margin and phase margin, bandwidth, and high frequency roll-off rate of the system.

Due to the interactive nature of the tutorial program, any change in the system setting or the values of parameters can be reflected in graphic display immediately. For example, the effect of damping ratio on the magnitude response can be easily demonstrated.

### III. A Real-time Process Control Project

In this section, a Process Control System course project is presented to illustrate how VB can be used for real-time applications. The main objective of the project is to control the temperature of a housing enclosure (Figure 3.) with a PC based control system (Figure 4).



Figure 3: A housing enclosure



Figure 4: PC based Control System

In this control system, two thermocouples are used, one for interior, the other for ambient, temperature measurement. The measurement signals are then amplified,

sampled, and then read into the PC through the parallel port. Based upon the difference between set-point and the actual measurement of the interior temperature, the control algorithm determines the switching of a solid state relay which in return control the “ON” or “OFF” of the heating element (a hair dryer).

It is noted that the professional edition of Visual Basic allows communication between PC and external devices via serial port only, but applications with VB can be extended to communication via parallel port by introducing Windows Application Programming Interface (API). A Dynamic Link Library file IOPORT.DLL (Download from: <http://www.stellarresearch.com/>) was used in this control system, and the following segment of codes show the link to the IOPORT.DLL:

```
Dim DataPort As New Port
Dim ControlPort As New Port
Dim StatusPort As New Port

'The Memory Port address for the printers Control port is &h37A
'setup port width for the control port
ControlPort.Address = &H37A
ControlPort.BusWidth = IOBW1Byte
'The Memory Port address for the printers Data port is &h378
'setup port width for the data port
DataPort.Address = &H378
DataPort.BusWidth = IOBW1Byte
'The Memory Port address for the printers Status port is &h379
'setup port width for the status port
StatusPort.Address = &H379
StatusPort.BusWidth = IOBW1Byte
```

At the front end of PC screen display (Figure 4), users can specify the process set-point and control tolerance level, the historic data of temperature measurement is graphically displayed.

Due to the fact that the process is a PC based control system, data logging can be done without requiring any additional hardware device, and different aspects of process operation can be inferred from the operation data. In addition, different control laws can be implemented without any change on the system hardware.

#### IV. Summary

So far our brief experiment with incorporating VB into our Control and Instrumentation curriculum is very rewarding. Although the Control Fundamental Tutorial is still at its early development stage and can be refined in many ways, positive feedback has been received from students as they find the learning process is a whole new experience. They have not only learned the subject matters but also gained more experience with computer applications. A number of course projects have been successfully carried out by applying VB programs for data acquisition and process control.

It can be envisaged that by applying advanced features of VB in the areas such as networking<sup>1</sup> and Windows API<sup>2</sup>, we are ready to venture into more challenging aspects of Control and Instrumentation engineering.

## Bibliography

1. Russo, M. F. & Echols, M. M. *Automating Science and Engineering Laboratories with Visual Basic*. John Wiley & Sons (1999).
2. Bradley, J. C. & Millspaugh, A. C. *Advanced Programming Using Visual Basic 6*. McGraw-Hill (2001).

## Acknowledgement

The authors would like acknowledge the work carried out by Bruce Hunter and Michelle Shuck in relation to the temperature control system.

## ALBERTO GOMEZ-RIVAS

Dr. Alberto Gomez-Rivas is a professor of Structural Engineering and the Chair of Department of Engineering Technology, University of Houston-Downtown. Dr. Gomez-Rivas received an MA and a Ph.D. in economics from Rice University in 1976 and 1979, a Ph.D. in Structural Engineering from University of Texas in 1968.

## WEINING FENG

Dr Weining Feng is an Associate Professor of Process Control and Instrumentation at Engineering Technology Department, University of Houston-Downtown. Dr. Feng received a Ph.D. from the Department of Electrical and Electronic Engineering, University of Strathclyde, Scotland, in 1990.