Implementing PID Temperature Control Using LabVIEW

Carlton Prewit, Rafic Bachnak

Department of Computing and Mathematical Sciences
Texas A&M University—Corpus Christi

Abstract

Feedback control is used by industry to regulate and improve responses and results of a variety of processes and systems. This paper describes the design and development of a feedback control system that maintains the temperature of a process at a desired set point. The system consists of a PC-based data acquisition unit that provides input and output interfaces between the PC and the sensor circuit and hardware. A proportional, integral, and derivative controller is implemented using LabVIEW. The paper will provide details about the data acquisition unit, shows the implementation of the controller, and present test results.

Introduction

Control systems can be found almost anywhere you look. They are used in automobiles, factories, homes and businesses, and are found in equipment used by any industry you can name. The job of a typical control system is to maintain something that is variable or changeable in a prescribed state. The human body for example has control systems that regulate heartbeat, maintain body temperature, and provide balance. Another common example is the temperature control system found in most homes. A control system keeps the air in a home at a certain temperature. It does this by sensing the ambient temperature, comparing it to the desired temperature, and producing a control action that will keep the ambient temperature at or near the desired temperature. This paper describes a project that involves designing, building, and demonstrating a regulatory control system. The system will maintain the temperature of a process at a desired set point. Another goal is to incorporate a PC-based data acquisition system (DAQ) along with LabVIEW, a graphics-based programming language by National Instruments. LabVIEW has been used in many control systems applications [1-5].

To demonstrate the functionality of the system a heating element that gives off a constant amount of heat was used. The surface temperature of the heating element is regulated by varying the amount of cooling it receives. A small electric fan is placed directly in line with the heating element so that cool air is forced over it. The amount of heat transfer from the element is directly proportional to the rate of airflow over it. The system

monitors the surface temperature of the element and controls it by varying the speed of the cooling fan.

DAQ System

This system uses a data acquisition system (DAQ) that is connected to a PC in the lab. It gathers input from the process and passes output signals to the control element. A control algorithm is implemented in software on the PC that is connected to the DAQ system. The LabVIEW software package from National Instruments is used to develop the custom data acquisition and control program. The program measures the temperature from the process, compares it to a desired set point, and issues the proper control signal to the final control element. The control signal adjusts the rotation speed of an electric fan. The fan rotation speed determines the rate of air flow over the heating element. Figure 1 below shows a block diagram of the system.
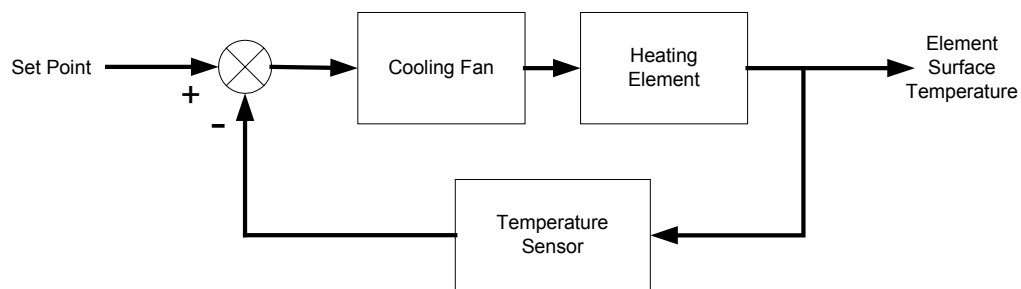
Figure 1. Feedback System Block Diagram

National Instruments SCXI

The I/O system is the Signal Conditioning eXtensions for Instrumentation (SCXI) system from National Instruments. This is a modular data acquisition system that has the ability to connect to a PC. This system includes an input module, output module, and terminal blocks that are housed in a chassis. The SCXI system provides a basic I/O interface along with signal conditioning. It is designed to connect to a DAQ card which is located in the PC. The DAQ card performs A/D and D/A conversion, as well as signal timing and multiplexing.

System Chassis

The SCXI-1000 is a 4-slot chassis designed to power and control up to four modules. It is expandable to allow more than one chassis to be daisy-chained as one system. The chassis provides power to the modules as well as a communication bus that is connected to the PC.

Data Acquisition Card

The DAQ device connects the SCXI chassis and modules to the PC. It performs the analog-to-digital and digital-to-analog conversion necessary for interfacing the I/O signals to the PC. The card is a National Instruments 6040E PCI card. It has 16 analog inputs, 2 analog outputs, 8 digital I/O lines, and two 24-bit counter/timers.

Analog Input

The analog input module is a SCXI-1102C 32-Channel Analog Input Module. It is ideal for measuring small voltage and current inputs, and includes a Cold Junction Compensation circuit for use with thermocouple sensors. Connected to the front of the SCXI-1102C is a terminal block. The terminal block provides the wiring terminals that external signals are connected to.

Analog Output

For analog output, a SCXI-1124 6-Channel Isolated Analog Output Module is provided. It is able to provide six channels of slowly varying DC voltage or current signals. The output voltage range is software selectable with the maximum swing between ±10 volts.

Software

Perhaps the most important piece of the system is the software. The software is the controller. It analyzes the input from the process and decides on an output action based on a control algorithm. The software is also the instrumentation. It displays data on the PC screen that a human operator can use to understand how the system is behaving. The software also enables an operator to control the process by entering a set point or by manually controlling the final control element.

National Instruments LabVIEW 7 Express

LabVIEW is a graphics-based software development package that incorporates data acquisition, measurement, analysis, presentation, and control. Programs are written in the form of Virtual Instruments, or "VI's" that can execute alone or can be used as "sub-VI's" in larger programs. The programming style is much like the C programming language, however, actual code input is done graphically rather than typed with a text editor. This makes it easy for non-programmers to quickly create and debug applications.

The software integrates seamlessly with data acquisition and measurement products from National Instruments. When used with the SCXI system, the result is very quick development of powerful measurement and control applications.

Perhaps the biggest benefit of the LabVIEW system is that it includes hundreds of VI's that are ready to use in a custom program. In the development of this project I took full advantage of ready-made VI's for control, acquisition, and analysis.

LabVIEW Software VI's

The software is perhaps the most important component of the system. It acts as the controller and also as instrumentation. The main routine, or VI, provides a front panel interface that allows the operator to control and monitor the system. It calls sub-VI's to perform functions that gather analog input, send analog output, and implement a PID control algorithm.

Front Panel

The front panel is what allows the operator to control and monitor the process. It includes software controls and indicators that mimic physical controls such as buttons, sliders, LEDs, and charts. Shown below as Figure 2 is a screenshot of the front panel.
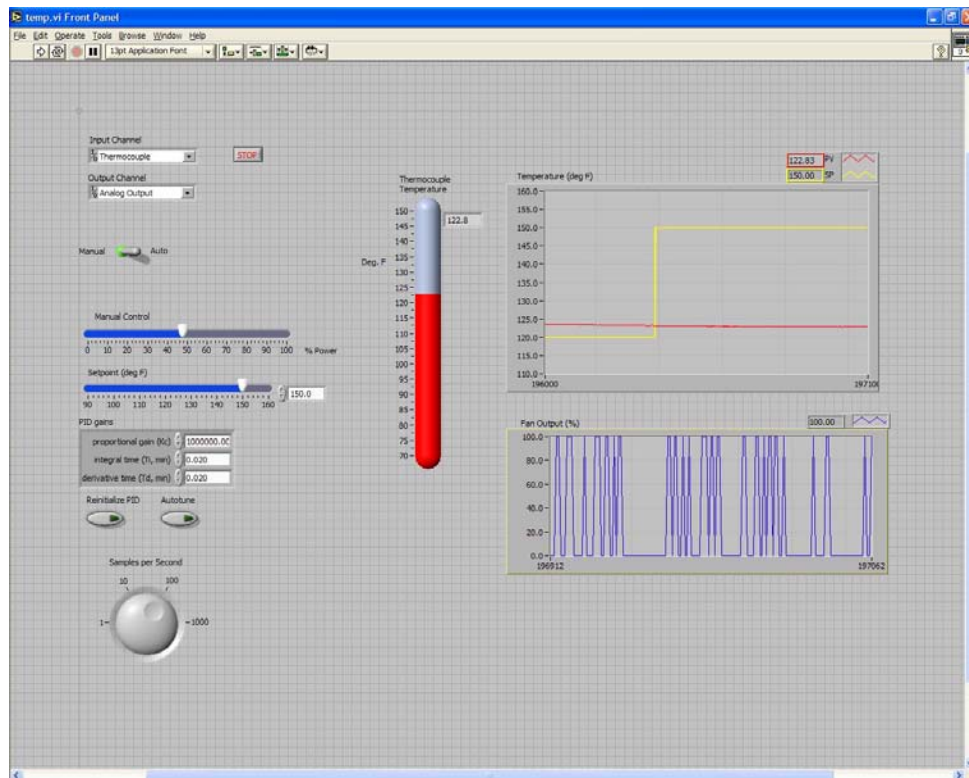

Figure 2. Screenshot Of LabVIEW Front Panel

The temperature of the process is displayed in a thermometer-style indicator as well as recorded on a strip chart. Also on the strip chart is the set point value. By showing both set point and measured values on the strip chart, one can easily see how the system responds to changes in set point. This is very helpful when trying to determine the correct PID constants. There is a slider for manually controlling the fan speed and one to adjust the temperature set point for automatic control. A toggle switch allows the operator to switch between manual and automatic control. There is a dial control that sets the sampling rate. It controls how fast the software loop runs. The PID values can be set in a numerical control box. Underneath the PID values box are two pushbutton switches. The one marked Autotune starts an automatic tuning routine. This routine attempts to find the best values for P, I, and D using the Zeigler-Nichols ultimate gain method. Upon completion, the new PID values are automatically entered into the control box.

Block Diagram

The block diagram is a graphical representation of the underlying software program. It consists of icons that represent typical programming elements such as constants, variables, subroutines, and loops. Figure 3 below is a screenshot of the block diagram for this system.
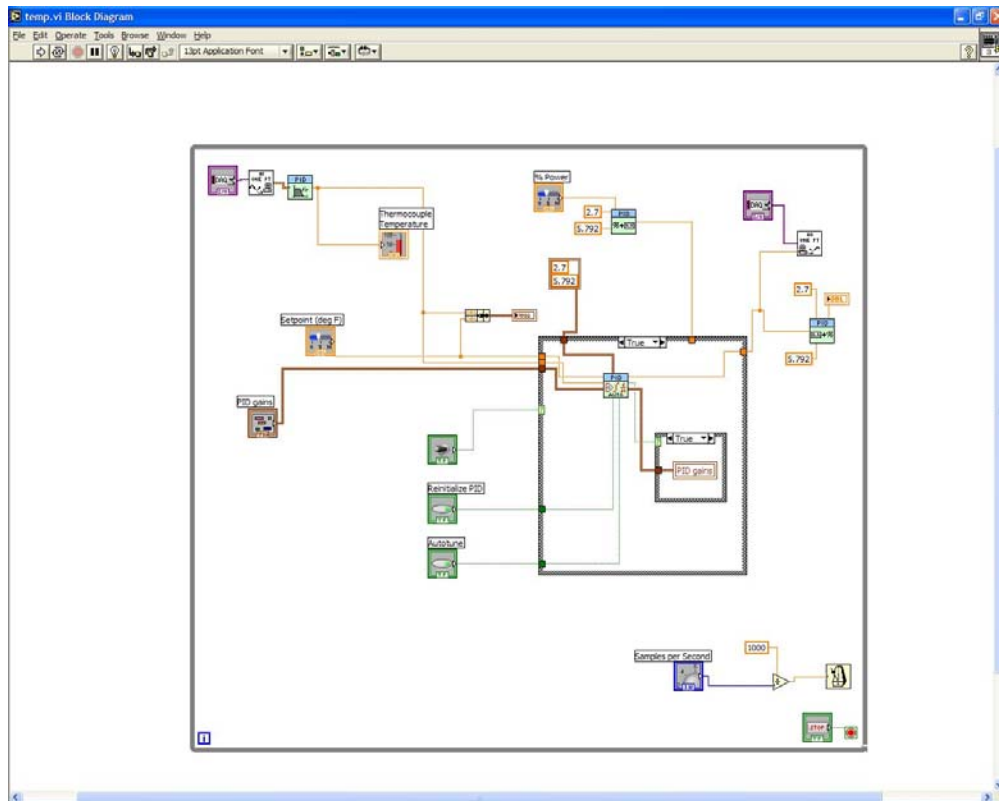

Figure 3. Screenshot of LabVIEW Block Diagram

Hardware and Circuits

System Power

To provide power for the electronics and fan, a 12-volt regulated supply is used. A simple voltage regulator IC is used to provide the positive 12-volt rails that power the fan and op-amp circuits. This way, only one external power connection is needed. A connection to the 15-volt laboratory power supply is all that is needed to supply regulated 12-volt power to the entire circuit.

Heat Circuit

A simple resistance heater circuit is used as the process heating element. It is made up of two 270Ω resistors wired in parallel. They are connected directly to the 12-volt power supply. The resistors heat up and dissipate 1.2 Watts of power, almost all of which is given up as heat. It is an easy way to model a heat source that is capable of reaching 160° F.

Temperature Sensor

A temperature sensor is attached to the surface of the heating element. It is what provides feedback to the control system. The sensor is a J-type thermocouple sensor which sees much use in industry. It is capable of sensing temperatures from 32-900 degrees Fahrenheit. It is well suited for use with the SCXI system because the signal conditioning system takes care of the cold junction compensation as well as the scaling necessary to get an accurate temperature reading in degrees Fahrenheit.

Fan Interface Circuit

A custom interface circuit is necessary to control the DC fan. It would be easiest to connect the fan directly to the SCXI-1124 output module. However, this is not possible for a couple of reasons. First, the output module is not designed to source the amount of current the fan needs to operate. The fan is designed to operate at 12 volts DC with around 60mA of current. When used as a voltage source, the output module can source at most 5mA of current. So, for the fan to work properly, an interface circuit is required that can provide the power needed by the fan. Even if the output module could deliver the current the fan needs, the module's voltage range is different than that required for the fan. Figure 4 shows how the fan is connected to the output module through the PWM circuit.

When designing an interface circuit, the most important thing to consider is the method that will be used to control the fan speed. The next sections discuss the possible interface methods.
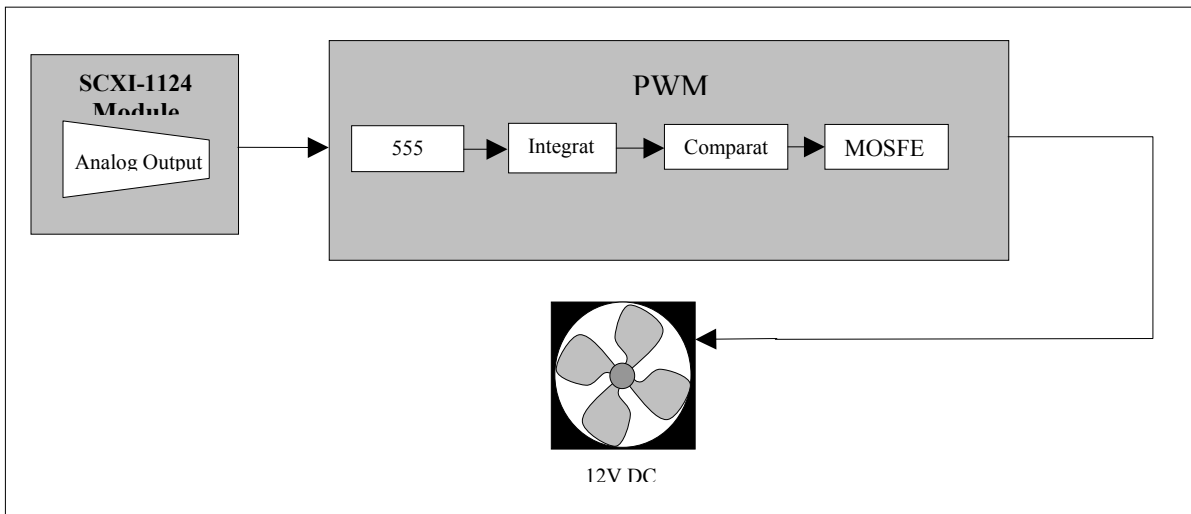
Figure 4. DC Fan Connected To Interface Circuit

Variable Voltage Method

The most straightforward method is to vary the input voltage to the fan between its minimum and maximum values. This would mean varying the voltage between 0-12 volts. A very simple circuit using an adjustable voltage regulator could be used. A typical regulator can provide up to 1 Amp of current which is more than enough needed to power the fan. The problem with this method is that a lot of power is wasted in the form of heat given off by the regulator. Also, the fan is designed to operate under full 12 volts. Operating the fan at voltages below this can shorten the life of its DC motor.

Pulse Width Modulation Method

A better method is to use PWM to control the amount of power going to the fan. This is accomplished by generating a square wave that will be used to switch a power MOSFET transistor. The MOSFET is capable of handling currents up to 4 Amps. The duty cycle of the square wave is modulated by the output signal coming from the analog output module. As the duty cycle changes, the "ON" time of the MOSFET is changed between 0 and 100%. When the MOSFET is switched like this at a relatively fast frequency, it effectively controls how much power is applied to the fan.

Building the System

The interface hardware was constructed using a breadboard and jumper wires. This allows changes to be made while testing the design.

Breadboard Circuit

The PWM, heat, and power circuits were constructed on a breadboard. Figure 5 below is a photograph of the completed circuit on a breadboard.
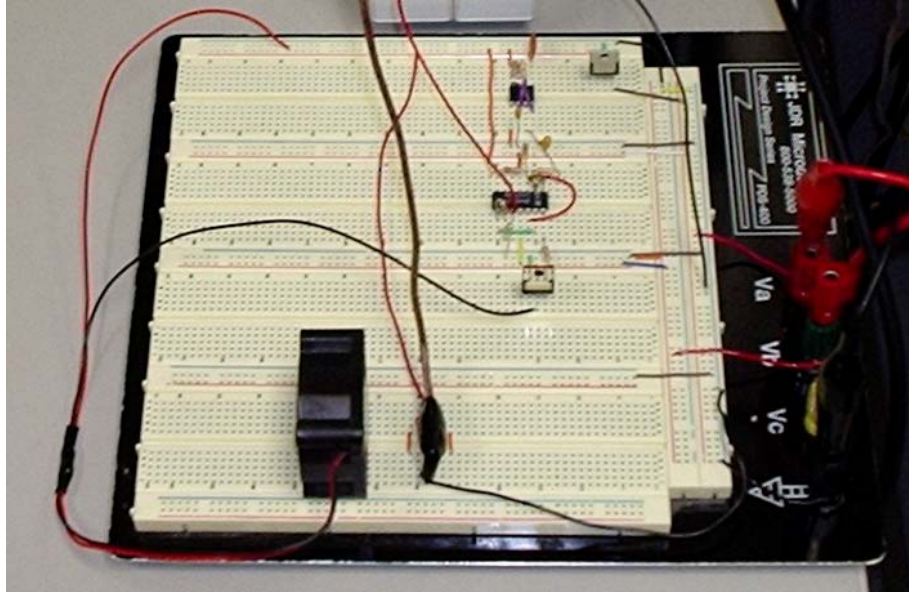

Figure 5. Breadboard Circuit

Writing and Debugging the Software

Since the LabVIEW environment uses a graphical programming language, it is very easy to write and debug the software. You don't have to memorize code because each code element and structure can be picked from a menu and dragged into your program. The program can be run and debugged in the same window. This makes it very quick to troubleshoot and fix problems. There is an "Execution Highlighting" option that acts as a debugger by slowly stepping through your program. The slow action allows you to see how the code is behaving.

Testing the System

Testing of the system was done in sections. For example, each component of the PWM circuit was tested as it was built. The ability to send and receive signals using the SCXI system was tested separately from the software. This approach makes problems easier to isolate when troubleshooting. Testing the complete system meant connecting the breadboard circuit to the SCXI modules, and then running the LabVIEW VI

Testing the SCXI hardware is pretty straightforward. There is a software configuration tool from National Instruments called Measurement and Automation Explorer. Using this software, you can easily configure input and output channels and test them. I configured and tested the input and output channels using this easy interface. When I was ready to use the channels in my LabVIEW program, I already knew they worked.

Since the system is controlled with a graphical front-panel interface, it is very easy and intuitive to learn. The controls and instruments are modeled after their physical counterparts. They have the same mechanical action as one would expect. To run the system, first make sure that the hardware circuit is powered on and connected to the SCXI modules. Start the LabVIEW software on the PC and run the VI. Everything can be controlled from the front panel.

Setup Requirements

The system needs a power supply to operate the breadboard circuitry and fan, a connection to the proper SCXI input and output modules, and a PC running the LabVIEW VI. One interesting thing is that the controlling PC need not be physically connected to the hardware circuit. As long as the hardware is connected to a networked PC via the SCXI system, the process can be monitored and controlled from any PC that can run the VI and communicate over the network. For example, I was able to do some of my software testing and development on a PC located upstairs from the actual SCXI system and process circuit.

Operation

To power the process circuit, a 15-volt DC power supply is needed. It should be able to provide a positive and negative 15-volts to the breadboard circuit. The system is simple to operate. Once the circuit is connected and powered up, the VI is started. The operator can see the current temperature and manually control the fan speed. Once the system is settled, control can be switched from manual to automatic. The PID algorithm is now used by the controller. The PID algorithm implements a bumpless transfer from manual to automatic. It also includes an auto-tuning function. Press the Autotune button to begin the self-tune process. The autotune function starts a 4-step "wizard" dialog that prompts you for the necessary input. Once the tuning process is complete, the new PID constants are automatically entered into the control box. The setpoint and chart scale can be adjusted while the program is running. To stop the program at any time, click on the Stop button.

Results

The system worked extremely well given the time constraints on design and construction. The fan is able to cool the heated element to a temperature of 110° F when running at is maximum speed. Using manual control, the operator is able to vary the fan speed from almost zero to full speed very smoothly. When operating under automatic control, the

system does a very good job of keeping the temperature within a few degrees of the set point. This was true even without tuning the loop for optimal control. For example, when set as proportional only control with unity gain, the temperature was held constant with an offset of five degrees from the setpoint. Shown below as Figure 6 is a screenshot showing a graph of the measured and setpoint values.
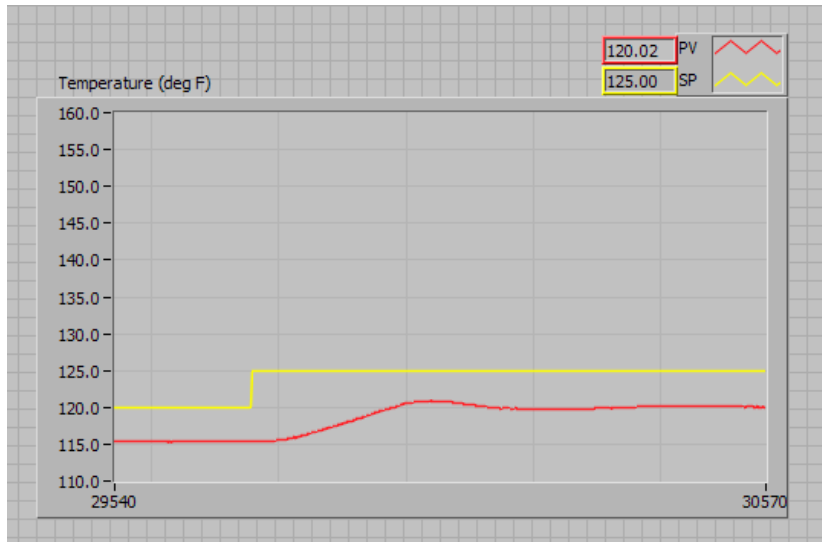

Figure 6. Proportional-only control, unity gain.

The figure above shows the response to a setpoint step change of 5 degrees. The temperature reaches a steady state, but there is a 5 degree offset. As the proportional gain is increased, the offset is reduced, however, oscillation is introduced into the system. This is shown as Figure 7.
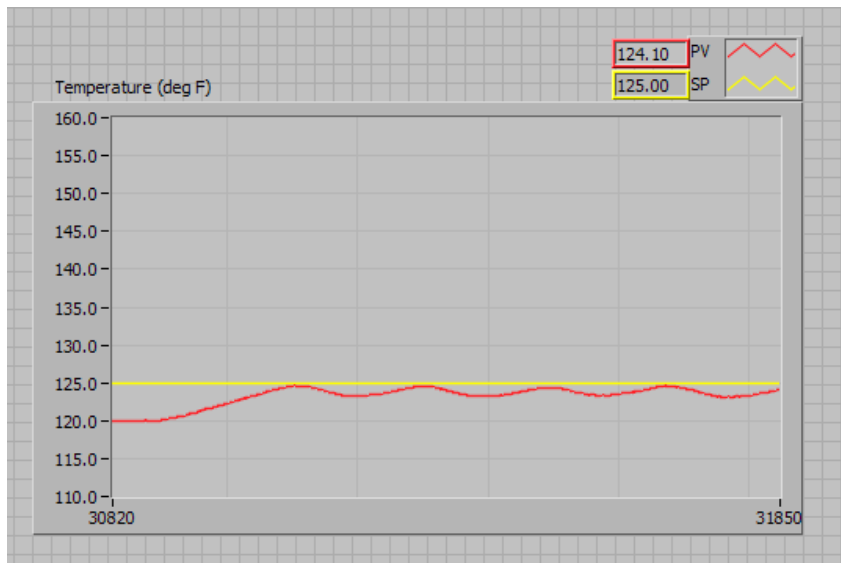

Figure 7. Minimal offset with oscillation.

The PID VI that was used includes an autotuning function. With the loop tuned, the controller was able to keep the process temperature within ±½ degree of the set point. The response to a three degree step change is shown below as Figure 8.
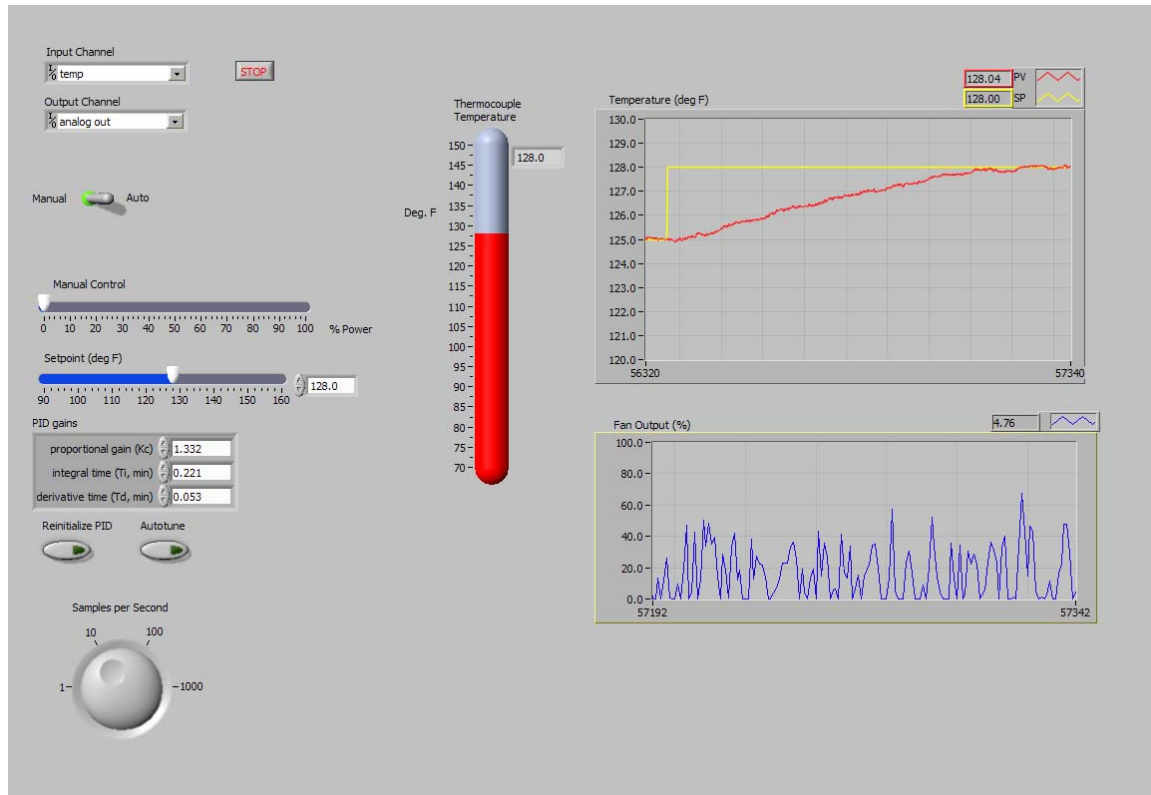

Figure 8. Autotuned PID Loop Response to Setpoint Step Change.

In this screenshot, you can also see the P, I, and D gains that were set by the auto-tuning function. Getting the loop perfectly tuned is not really an issue for this system because even the 1-degree variation of an un-tuned system is too small to matter for practical applications.

Conclusion

The system described in this paper illustrates the design and development of a feedback control system that employs a proportional, integral, and derivative controller implemented using LabVIEW. The system provides a very good tool for learning how to implement PID control and is being converted into an experiment for use in an upper-level control systems course. The setup is straightforward provided that the PWM method for powering the fan is supplied to the student. The LabVIEW environment is intuitive and allows much to be done in little time.

References

1. M. Naghedolfeizi, S. Arora, and S. Garcia, "Survey of LabVIEW technologies for building Web/Internet-enabled experimental setups," Proceedings of the 2002 ASEE Annual Conference and Exposition, June 16-19, 2002 Montreal, CA.
2. N. Swain, J. Anderson, M. Swain, and R. Korrapati, "State-space analysis of linear, time-invariant control systems using virtual instruments," Proceedings of the 2001 ASEE Annual Conference Proceedings, June 2001, Albuquerque, NM.
3. L. Sokoloff, "LabVIEW implementation of ON/OFF controller," Proceedings of the 1999 ASEE Annual Conference Proceedings, June 1999, Charlotte, NC.
4. R. Bachnak and C. Steidley, "An interdisciplinary laboratory for computer science and engineering technology," Journal of Computing in Small Colleges, Vol. 17, No. 5, April 2002, pp. 186-192.
5. K. Resendez and R. Bachnak, "LabVIEW programming for internet-based measurements," Journal of Computing in Smakk Colleges, Vol. 18, No. 4, April 2003, pp. 79-85.

CARLTON PREWIT
Carlton Prewit is a senior student in the Control Systems Engineering Technology program at Texas A&M University—Corpus Christi. Mr. Prewit has been working as an Undergraduate Research Assistant in the Department of Computing and Mathematical Sciences (CAMS) since Fall 2003. His research involves the design and development of custom-made depth sensors for use on a remotely controlled boat for data acquisition in shallow water areas. Mr. Prewit is expected to graduate with a B.S. degree in May 2004.

RAFIC BACHNAK
Rafic (Ray) Bachnak is a Professor in the Department of Computing and Mathematical Sciences (CAMS) at Texas A&M University-Corpus Christi (A&M-CC). He received his B.S., M.S., and Ph.D. degrees in Electrical and Computer Engineering from Ohio University in 1983, 1984, and 1989, respectively. Dr. Bachnak was previously on the faculty of Franklin University and Northwestern State University.