www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 1 January 2015, Page No. 9933-9940

Closed loop control of curing process using Lab VIEW

Mrs. V. Radhika[1], V.P. Priyadharshini[2], B. Shibila[3], V. Varsha[4]
Electronics and Communication Engineering, Anna University/Sri Ramakrishna Engineering College.

ABSTRACT
The objective of this project is to simulate a curing process chamber found in many industry settings. Curing involves any process where heat is used to catalyze or initiate chemical and molecular level structural changes in polymeric materials such as epoxies, phenolic, polyesters and silicones. The project aims to simulate the entire process of temperature control, its variations and as to how PID controller determines its response in LabVIEW. Each set point is attained and controlled for the specified time. On a waveform chart, all the important process parameters like set point, process variable and controller output are plotted using a DAQ card to the host computer running LabVIEW.

Keywords:Curing station, Final control element, LabVIEW, PID controller, Thermocouple.

I. INTRODUCTION

Closed loop control of curing process using LabVIEW emulates the curing process found in many industry settings. It introduces a set point change in control process and controls each set point for the specified time. Set point, Process variable and controller output all are displayed on the waveform chart. Active transducer (i.e.) Thermocouple, is used as the temperature sensor to sense the water bath temperature. Thermocouple output is amplified using a non-inverting Amplifier. The amplified output range is in the order of (0-5)V to make it easy for DAQ communication. The overall simulation process not only provides a solution for curing process but also for the Servo problems.

A. General block diagram

The block diagram of the Closed Loop Control for Curing Process using LabVIEW is shown in Fig.1.

It consists of four main functions. They are

- Temperature sensing
- Amplification unit
- Control and simulation.

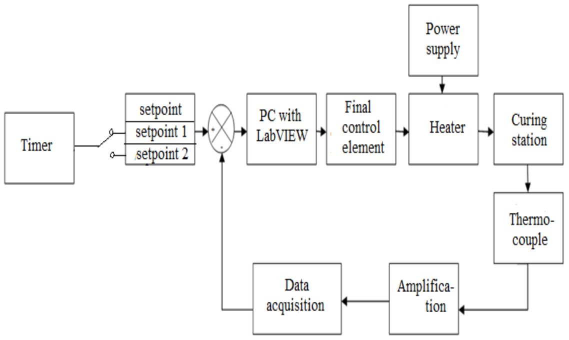


Fig.1 Block diagram of closed loop control of curing process.

The above individual sections are explained in detail in the following paragraphs.

B. Temperature sensing

The project makes use of a thermocouple to detect the temperature from the requisite test location, which can be a boiler, a furnace or any other industrial application needing temperature control. The sensed signal is then sent to the signal conditioned circuit.

C. Amplification unit

The signal received from the thermocouple setup is then amplified and then sent to the master Computer.

D.Control and simulation

The controller plays a major role in controlling the overall process. The PID controller is implemented using PID toolbox .The temperature sensed by thermocouple is controlled by the PID controller and given to the heater control circuit. If the temperature reaches above the set point, the controller controls the final control element which can be a TRIAC or an SCR to turn off the heater and if the temperature reaches below the set point it will make the TRIAC or SCR to turn on.

II. CONTROLLER DESCRIPTION

A. PID controller

The Proportional-Integral-Derivative (PID) algorithm’s the most common control algorithm used in industry. PID is used to control processes that include heating and cooling systems, fluid level monitoring, flow control, and pressure control. In PID control, a process variable and a set point should be specified by user. The process variable is the system parameter you want to control, such as temperature, pressure, or flow rate, and the set point is the desired value for the parameter you are controlling.

A PID controller determines a controller output value, such as the heater power or valve position. The controller applies the controller output value to the system, which in turn drives the process variable toward the setpoint value. The PID Control Toolset VIs with National Instruments Hardware are there to develop Lab VIEW control applications.

Use I/O hardware, like a DAQ device, Field Point I/O modules, or a GPIB board, to connect PC to the system you want to control. The I/O VIs provided in LabVIEW with the PID Control Toolset to develop a control application or modify the examples provided with the Toolset.

Using the VIs described in the PID Control section of the manual, one can develop the following control applications based on PID controllers:

- Proportional (P)
- Proportional-integral (PI)
- Proportional-derivative(PD)
- Proportional-integral-derivative (PID)
- Gain-scheduled PID
- PID auto tuning
- Error-squared PID
- Lead-Lag compensation
- Set point profile generation
- Multi loop cascade control
- Feed forward control
- Override (minimum/maximum selector) control
- Ratio/bias control

B. Proportional term

The proportional term produces an output value that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain (1).

$$Pout = Kp\ e(t) \qquad (1)$$

The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output.
The proportional term is given by the following equation.

A high proportional gain results in a large change in the output for a given change in the error. If the proportional gain is too high, the system can become unstable. In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

C. Integral term

The contribution from the integral term is proportional to both the magnitude of the error and the duration of the error. The integral in a PID controller is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously.

The accumulated error is then multiplied by the integral gain (Ki) and added to the controller output.
The integral term is given by the following equation (3):

$$Iout=Ki\int_0^t e(t)dT \quad (3)$$

The integral term accelerates the movement of the process towards set point and eliminates the residual steady-state error that occurs with a pure proportional controller.The derivative of the process error calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain Kd. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, Kd. The derivative term is given by the following equation (2):

D. Derivative term

$$Dout=Kd\ \frac{d}{dt}e(t) \qquad (2)$$

The derivative term slows the rate of change of the controller output. Derivative control is used to reduce the magnitude of the overshoot produced by the integral component and improve the combined controller-process stability. However, the derivative term slows the transient response of the controller. Also, differentiation of a signal amplifies noise and thus this term in the controller is highly sensitive to noise in the error term, and can cause a process to become unstable if the noise and the derivative gain are sufficiently large .In contrast, a small gain results in a small output response to a large input error, and a less responsive or less sensitive controller. If the proportional gain is too low, the control action may be too small when responding to system disturbances.

E.PID controller block diagram

The general block diagram of a PID controller is shown in the Fig.2.

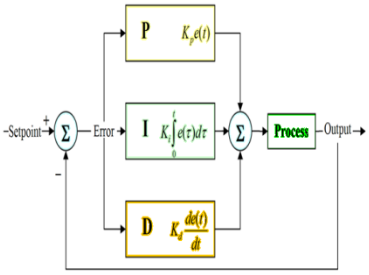


Fig.2 Block diagram of PID controller

F. PID algorithm

The equation (4) represents the PID controller that compares the set point (SP) to the process variable (PV) to obtain the error(*e*).

$$e = SP - PV \qquad (4)$$

Then the PID controller calculates the controller action, u(t) which is represented in the equation (5) ,where Kc is controller gain.

$$u(t) = Kc\{e + \frac{1}{Ti}\int_0^t edt + Td\frac{de}{dt}\} \qquad (5)$$

If the error and the controller output have the same range, –100% to 100%, controller gain is the reciprocal of proportional band. *Ti* is the integral time in minutes, also called the reset time, and *Td* is the derivative time in minutes, also called the rate time.

The equation (6) represents the proportional action.

$$uP(t)=Kc(e) \qquad (6)$$

The equation (7) represents the integral action.

$$uD(t)=KcTd(de/dt) \qquad (7)$$

The following equation (3.8) represents the derivative action.

$$uI(t)=(Kc/Ti)\int edt \qquad (8)$$

III. TEMPERATURE MEASUREMENT

A. Introduction

The temperature measurement device used here is the Thermocouple type. A thermocouple is a device consisting of two different conductors (usually metal alloys) that produce a voltage proportional to a temperature difference between either end of the pair of conductors. Thermocouples are small, rugged, relatively inexpensive, and operate over the widest range of all temperature sensors. They are especially useful for making measurements at extremely high temperatures (up to +2300°C) in hostile environments. They produce only millivolts of output, however and require precision amplification for further processing. They are more linear than many other sensors, and their non-linearity has been well characterized. The most common metals used are Iron, Platinum, Rhodium, Rhenium, Tungsten, and Copper.

The e.m.f produced in a thermocouple circuit is given by:

$$E=a(\Delta\theta)+b(\Delta\theta)^2$$

(9)

$\Delta\theta$ = difference in temperature between the hot thermocouple junction and the reference junction of the thermocouple.

And a, b=constants.

a is usually very large as compared with band therefore thermocouple emf is

$$E=a(\Delta\theta)d\theta \qquad (10)$$

The thermo electric emf depends upon the difference in temperature between the hot junction and the reference junction, the temperature of the latter should remain absolutely constant in order that the calibration holds good and there are no errors on account of change in ambient temperature. The temperature of reference junction is controlled for this purpose. The reference junction temperature is usually 0 C. The thermocouples are used for measurement of temperature upto 1400 C.

B. Thermocouple

The below shown Fig.4 represents the thermocouple temperature sensor.

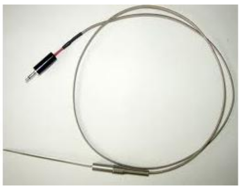


Fig.4 Image of thermocouple

C. Thermocouple characteristics

In this circuit the Thermocouple is used to measure the temperature.The thermocouple used here is type J: Fe-K. Type J covers temperature range between (−40 to +750) °C and has a sensitivity of about 55 µV/°C.

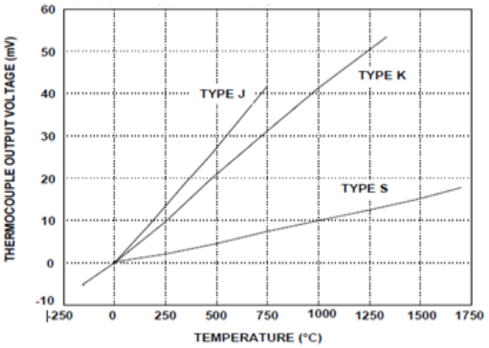


Fig.3 Thermocouple characteristics

If the two dissimilar metals are at any temperature above absolute zero, there will be a potential difference between them. If the two wires are joined at two places, two junctions are formed .If the two junctions are at different temperatures, there will be a net e.m.f. in the circuit, and a current will flow determined by the e.m.f. and the total resistance in the circuit If one of the wires is broken, the voltage across the break will be equal to the net thermoelectric e.m.f. of the circuit, and if this voltage is measured. It can be used to calculate the temperature difference between the two junctions.

D. Advantages of thermocouple

- Thermocouples are active transducers unlike RTD and thermistors which are passive transducers.
- A thermocouple is best for your application, because of its lower cost, wider temperature range, faster response time, and simpler construction, plus its many physical sizes of configurations available.
- Thermocouples follow the temperature changes with a small time lag and as such are suitable for recording comparatively rapid changes in temperature.
- Thermocouples are very convenient for measuring the temperature at one particular point in a piece of apparatus.

IV. CIRCUIT DESCRIPTION

A. Basic non –inverting operational amplifier circuit

The basic non-inverting operational amplifier circuit, to which the signal is applied to the non-inverting input of the op-amp. However the feedback is taken from the output of the op-amp via a resistor to the inverting input of the operational amplifier where another resistor is taken to ground. It is the value of these two resistors that govern the gain of the operational amplifier circuit.

As the input to the op-amp draws no current this means that the current flowing in the resistors R1 and R2 is the same. The voltage at the inverting input is formed from a potential divider consisting of R1 and R2, and as the voltage at both inputs is the same, the voltage at the inverting input must be the same as that at the non-inverting input. This can be written as shown in equation (11).

$$Vin = Vout \times R1 / (R1 + R2) \qquad (11)$$

Hence the voltage gain of the circuit GAIN (Av) as is shown in equation (12):

$$GAIN(Av) = 1 + R2 / R1 \qquad (12)$$

There are several reason for this:

1)for the same gain non-inverting amplifier has higher bandwidth

2)input offset voltage and input referred noise at op-amp input get higher gain (1+R2/R1) than input voltage (-R2/R1) in case of inverting amplifier and hence degrading Signal to noise ratio at output.

B. Amplification circuit

The non-inverting operational amplifiers that are cascaded to give necessary gain factor for amplification is shown in the Fig.5. Here two non-inverting operational amplifiers are used. The input voltage is given by the power supply unit in the process for the same gain non-inverting amplifier has higher bandwidth.Non inverting amplifier is always better than inverting amplifier for the same gain if the input voltage variation is within op-amp input common mode range and if sign of gain is irrelevant.

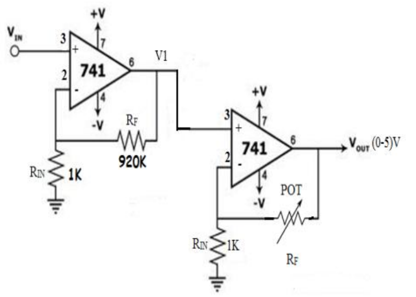


Fig.5 Circuit of non-inverting operational amplifier

C. Graphical representation

The characteristics of temperature in degree Celsius and signal conditioned output in Volts is given in the Fig.6.

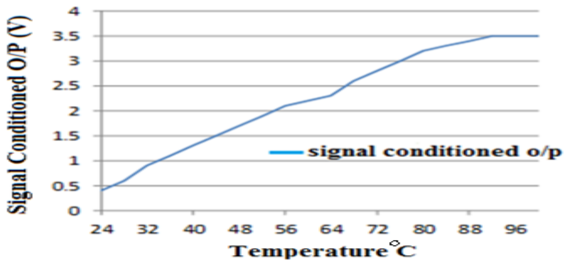


Fig.6 Temperature Vs Signal conditioned output graph

The characteristics of temperature in degree Celsius and thermocouple output in millivolts is shown in the Fig.6.The characteristics of the thermocouple is an important procedure in controlling the curing process.

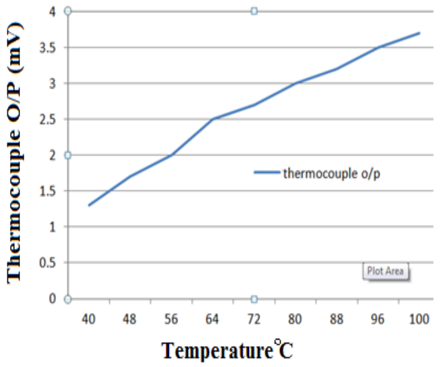


Fig.7 Temperature Vs Thermocouple output graph

V. SOFTWARE DESCRIPTION

A. Introduction

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine the order of program execution, LabVIEW uses dataflow programming, where the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions. VIs, or virtual instruments, is LabVIEW programs that imitate physical instruments.

In LabVIEW, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

LabVIEW is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and data acquisition control, vision, and motion control devices. LabVIEW also has built-in features for connecting your application to the Internet using the LabVIEW web server and software standards such as TCP/IP networking and ActiveX.

Using LabVIEW, you can create 32-bit compiled applications that give you the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions. You also can create stand-alone executables and shared libraries, like DLLs, because LabVIEW is a true 32-bit compiler.

LabVIEW contains comprehensive libraries for data collection, analysis, presentation, and storage. LabVIEW also includes traditional program developmenttools. You can set breakpoints, animate program execution, and single-step through the program to make debugging and development easier.

LabVIEW also provides numerous mechanisms for connecting to external code or software through DLLs, shared libraries, ActiveX, and more. In addition, numerous add-on tools are available for a variety of application needs.

B. Virtual instruments

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. A VI contains the following three components:

- Front panel Serves as the user interface.
- Block diagram contains the graphical source code that defines the functionality of the VI
- Icon and connector pane identifies the interface to the VI so that you can use the VI in another VI.

C. Front panel

The front panel is the user interface of the VI. You build the front panel using controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input mechanisms. Indicators are graphs, LEDs, and other output displays. Controls simulate instrument input mechanisms and supply data to the block diagram of the VI. Indicators simulate instrument output mechanisms and display data the block diagram acquires or generates.

D. Block diagram

After you build the front panel, you add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code, also known as G code or block diagram code. Front panel objects appear as terminals on the block diagram.

E. LabVIEW environment

It is possible to use the LabVIEW palettes, tools, and menus to build the front panels and block diagrams of VIs.LabVIEW includes three palettes: the Controls palette, the Functions palette, and the Tools palette. LabVIEW also includes the Getting Started window, the Context Help window, the Project Explorer window, and the Navigation window.

1)while loop

The while loop which repeats the sub diagram inside it until the conditional terminal, an input terminal, receives a particular Boolean value is shown in the Fig.8. The Boolean value depends on the continuation behaviour of the While Loop. Right-click the conditional terminal and select Stop if True or Continue if True from the shortcut menu. You also can wire an error cluster to the conditional terminal, right-click the terminal, and select Stop on Error or Continue while Error from the shortcut menu. The While Loop always executes at least once. The iteration **(**i)terminal provides the current loop iteration count, which is zero for the first iteration.

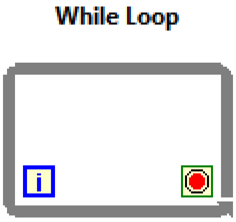


Fig.8 Icon of while loop

2) Case structure

The case structure given in the Fig.9 shows that it has one or more sub diagrams, or cases, exactly one of which executes when the structure executes. The value wired to the selector terminal determines which case to execute and can be Boolean, string, integer, enumerated type, or error cluster. Right-click the structure border to add or delete cases.

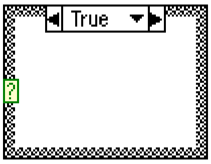


Fig.9 Icon of case structure

3) PID advanced VI

The PID Advanced VI shown in Fig.10 implements a PID controller using a PID algorithm with advanced optional features. The advanced PID algorithm includes the features of the algorithm the PID VI uses, as well as manual mode control with bump less manual-to-automatic transitions, nonlinear integral action, two degree-of-freedom control, and error-squared control.

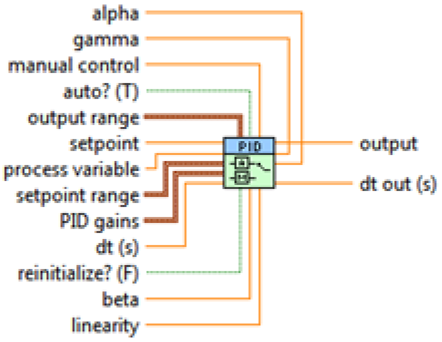


Fig.10 Icon of PID Advanced VI

4) PID VI

The PID VI shown in Fig.11 implements a PID controller using a PID algorithm for simple PID applications or high speed control applications that require an efficient algorithm. The PID algorithm features control output range limiting with integrator anti-windup and bump less controller output for PID gain changes.

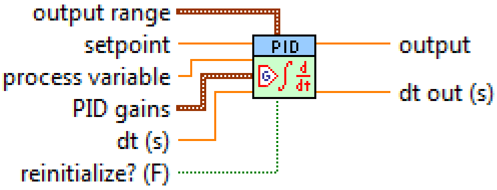


Fig.11 Icon of PID VI

5) wait until next ms multiple function

The wait until next ms multiple is shown in the Fig.12 that waits until the value of the millisecond timer becomes a multiple of the specified millisecond multiple. This function is called in a loop to control the loop execution rate. However, it is possible that the first loop period might be short. Wiring a value of 0 to the milliseconds multiple input forces the current thread to yield control of the CPU.

This function makes asynchronous system calls, but the nodes themselves function synchronously. Therefore, it does not complete execution until the specified time has elapsed.

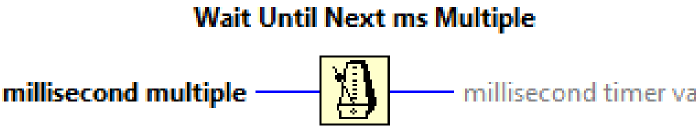


Fig.12 Icon of wait until next ms multiple function

6) PID set point profile

The PID set point profile shown in Fig.13 generates set point values over time in a control loop for ramp and soak types of control applications. The inputs of the PID set point profile are a set point profile cluster, change in time(s) and a boolean input which is used for reinitialisation.

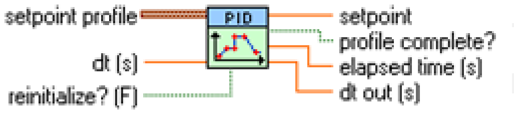


Fig.13 Icon of PID setpoint profile

VI. TUNING PROCEDURE

A. Closed loop (ultimate gain) tuning procedure

Although the closed-loop (ultimate gain) tuning procedure is very accurate, process must be put in steady-state oscillation and observe the PV on a strip chart. Complete the following procedure to perform the closed-loop tuning process.

1. Set both the derivative time and the integral time on your PID
Controller to 0.
2. With the controller in automatic mode, carefully increase the proportional gain (*Kc*) in small increments. Make a small change in SP to disturb the loop after each increment. As you increase *Kc*, the value of PV should begin to oscillate. Keep making changes until the oscillation is sustained, neither growing nor decaying over time.
3. Record the controller proportional band(PB_u) as a percent, where PB_u= 100/*Kc*.
4. Record the period of oscillation (T_u) in minutes.
5. Multiply the measured values by the factors shown in Table 1 and enter the new tuning parameters into your controller. Table 1provides the proper values for a quarter-decay ratio.
If you want less overshoot, increase the gain

Table 1- Closed-Loop–Quarter-Decay Ratio Values

Controller	PB (%)	RESET(min)	RATE(min)
P	2.00PBu	-	-
PI	2.22PBu	0.83Tu	-
PID	1.67PBu	0.50Tu	0.125Tu

B. Controller tuningThe Ziegler Nichols tuning procedure output graph is shown in the Fig.14. The ultimate time and the gain can be determined from this graph.

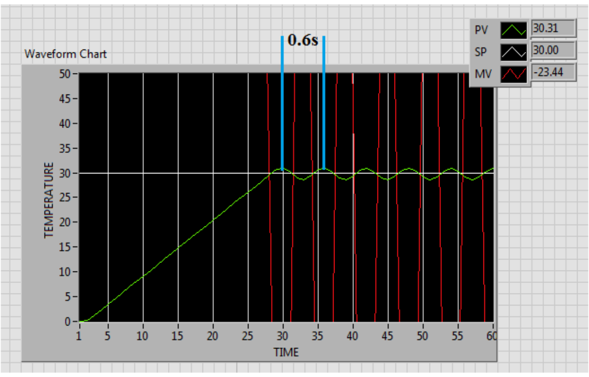
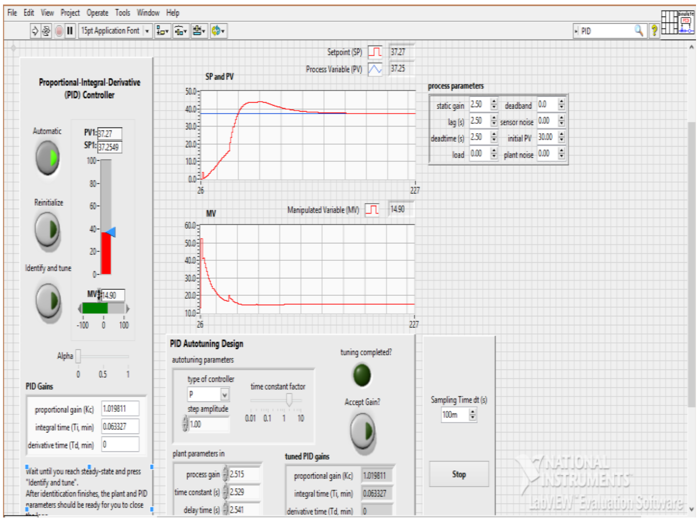


Fig.14 Output graph of Ziegler Nichols controller tuning

C. Tuned PID gain value

The default set point, process variable, and output ranges for the PID VIs correspond to percentage of full scale. In other words, proportional gain (*Kc*) relates percentage of full-scale output to percentage of full-scale input. This is the default behaviour of many PID controllers used for process control applications. To implement PID in this way, you must scale all inputs to percentage of full scale and all controller outputs to actual engineering units. The integral time (Ti) and derivative time (Td) are in minutes. When these tuned values are given as PID gains to the controller, we can observe settling of process variable without any oscillation.



Proportional-Integral-Derivative (PID) Controller

Setpoint (SP) 37.27
Process Variable (PV) 37.25
Manipulated Variable (MV) 14.90

PV1: 37.27
SP1: 37.2549
MV: 14.90

process parameters

static gain	2.50	deadband	0.0
lag (s)	2.50	sensor noise	0.00
deadtime (s)	2.50	initial PV	30.00
load	0.00	plant noise	0.00

PID Gains

proportional gain (Kc)	1.019811
integral time (Ti, min)	0.063327
derivative time (Td, min)	0

Wait until you reach steady-state and press "Identify and tune".
After identitication finishes, the plant and PID parameters should be ready for you to close

PID Autotuning Design

autotuning parameters

type of controller P
step amplitude 1.00

plant parameters in

process gain	2.515
time constant (s)	2.529
delay time (s)	2.541

tuned PID gains

proportional gain (Kc)	1.019811
integral time (Ti, min)	0.063327
derivative time (Td, min)	0

Sampling Time dt (s) 100m

Fig.15 Output response in LabVIEW platform

VII. CONCLUSION

The curing process is implemented and controlled using LabVIEW. From the output response it is concluded that this process which has sufficient flexibility to yield excellent results in a wide variety of applications.

In curing process heat is used to catalyze or initiate chemical and molecular level structural changes. Hence different temperature settings are necessary for different time period. Therefore this simulation process can provide a solution for curing process.

VIII. FUTURE SCOPE

The future scope of this process is to implement the simulation in real time. This process can further be implemented for other process variables like pressure, level, flow and the servo problem associated with them. Controllers like Model Predictive Controller and Model Reference Adaptive Controller can also be used.

IX. REFERENCES

[1] Chenhui Zhao, Guang Cheng Zhang and Lei Zhao, (2012)'Effect of curing agent and temperature on the Rheological Behavior of epoxy resin systems', Journal of Composite MaterialsVol.17, No.3, pp.8587-8594.

[2] Harsh Kaji, ShrutiAnnigeriandProf.PrafullaPatil, (2013)'Designing PID Controller using LabVIEW for Controlling Fluid Level of Vessel', International Journal of Engineering Research and Applications Vol.3, No.2, pp.1329-1333.

[3] Keyu Li, (2013) 'PID Tuning for Optimal Closed-Loop Performance with Specified Gain and Phase Margins', IEEE Transaction on Control Systems Technology Vol.21, No.3, pp.132-140.

[4] Melba Mary, P. Marimuthu, N.S. and Albert Singh, N. (2007) 'Design of Intelligent Self- Tuning Temperature Controller for Water Bath Process', International Journal of Imaging Science and Engineering Vol.1, No.4, pp.935-943.

[5] Mote, T.P. and Lokhande, S.D. (2012) 'Temperature Control System Using ANFIS', International Journal of Soft Computing and Engineering Vol.2, No.1, pp.231-236.

[6] Ranjit Kumar, C. HarshaVardhan, A. and SaiBharadwaj, A. (2013) 'Temperature Control System Using Artificial Neural Network', International Journal of Engineering Research and Applications Vol.3, No.4, pp.672-675.

[7] SachinWaigaonkar, Babu.BJC. and Amit Rajput,(2011)'Curing studies of unsaturated polyester resin used in FRP products', Indian Journal of Engineering & Materials sciences Vol.18, No.4, pp.31-39.

[8] Yasmine Begum, A. VenkataMarutheswar, G.V. and AyyappaSwamy, K. (2013) 'Tuning of PID Controller for superheated Steam Temperature System using Modified Zeigler-Nichols Tuning Algorithm', International Journal of Engineering and Advanced Technology Vol.2, No.2, pp.541-543.

[9] Yusoff, R. Aroua, M.K. Nesbitt, A. and Day, R.J. (2007) 'Curing of polymer composites using microwave resin transfer moulding',Journal of Engineering Science and Technology Vol.2,No.2,pp.151-163.

[10] Zhang, J.Xu,Y.C. and Huang, P. (2009) 'Effect of cure cycle on curing process and hardness for epoxy resin ', eXPRESS Polymer Letters Journal of Composite Materials Vol.3,No.9, pp.534-541.

Authors

Mrs. V.Radhika is the Assistant Professor(Sr.G.) in the department of Electronics and Instrumentation Engineering. She is having teaching experience of more than 8 years. She did her under graduate in Electronics and Instrumentation Engineering from R.V.S College of Engineering and Technology, Dindigul and her post graduate in Applied Electronics from Kumaraguru College of Technology, Coimbatore. She has published papers in 02 International Journals, 01 International Conferences and 02 National Conferences. Her area of interest is Low Power VLSI and Embedded Systems.

V.P Priyadharshiniisstudying BEin Electronics and Instrumentation Engineering from Sri Ramakrishna Engineering College, Coimbatore under Anna University Chennai.

B. Shibilaisstudying BEin Electronics and Instrumentation Engineering from Sri Ramakrishna Engineering College, Coimbatore under Anna University Chennai.

V. Varsha is studying B.E in Electronics and Instrumentation Engineering from Sri Ramakrishna Engineering college, Coimbatore under Anna University Chennai.