DEVELOPMENT OF A LIQUID LEVEL CONTROLLER BY USING DIGITAL SIGNAL PROCESSING SYSTEM

G. K. G. S. Kumara[1], M. Narayana[2]

Department of Chemical & Process Engineering, Faculty of Engineering, University of Moratuwa, Sri Lanka.
Email: [1]sapumalkaar@gmail.com, [2]mahinsasa@uom.lk

ABSTRACT

Liquid level control is vital in the chemical and process industries. Some processes need liquids to be pumped to a certain desired level. This paper presents development of Proportional-Integral-Derivative (PID) controller for controlling the desired liquid level of a tank. Three conventional techniques of PID tuning that is Ziegler-Nichols, Z-N, method and Good Gain, G-G, method and Trial & Error method will be tested in order to obtain the PID controller parameters. Simulink is used to build the model and the liquid level of a tank is controlled by using TI Delfino F28355 Experimental kit which is a digital signal processing (DSP) control board. Arduino UNO board is used for real time data acquisition which is used to plot required graphs in Labview. The performance of the system is evaluated in terms of Rise Time (Tr), Settling Time (Ts), Steady State Error (SSE) and Overshoot (OS).

Key words: Digital Signal Processing (DSP) control board, Delfino F28355 Experimental kit, PID Controller, PID Tuning Method, Water Level Control, Simulink

1. INTRODUCTION

Real time monitoring and controlling are widely used in chemical and process industries. Usually it is required that liquid level is critically maintained at a specific height [1]. Feedback control structure is the most common control structure in industrial control system. In this study, feedback control strategy is used with Proportional Integral Derivative (PID) controller. Liquid level is detected by an ultrasonic sensor and then a pump is controlled accordingly to maintain required level of the tank. The experimental setup of the liquid level tank is shown in Figure 1.

Figure 1: Liquid level tank

PID controlling is used for error based controlling. Control signal ***u*** is expressed in the eq. (01). ***e*** is the error for feedback loop. Feedback control loop is shown in the Figure 2.

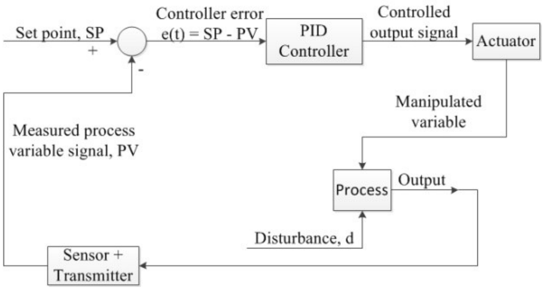

Figure 2: Feedback control loop

The PID control algorithm comprises three elements:

- **P**roportional - also known as Gain (K_P)
- **I**ntegral - also known as Automatic Reset or simply Reset (K_I)
- **D**erivative - also known as Rate or Pre-Act (K_D)

$$u = eK_P + K_I \int e dt + K_D \frac{de}{dt} \qquad (01)$$

This study intends to use digital signal processing (DSP) control boards for data acquisition and processing. Many digital signal processing (DSP) control boards are available for various controlling applications. TI Delfino F28355 Experimental kit (Figure 3) was used in this study, which is a high-performance floating-point microcontroller and it delivers up to 150 MHz

(6.67-ns Cycle Time) [2]. Therefore it is an ideal controller for high speed real time controlling applications.

Figure 3: TI Delfino F28355 Experimental kit

The biggest advantage of using this DSP board is that it is not necessary to have programming skills. This DSP board can be integrated with MATLAB/Simulink and then it can be utilize more complex mathematical tools, which are available in MATLAB. Instead of writing programming codes for controlling, it is easy to build a mathematical model by using Simulink which is a powerful modeling tool and then C-language code can be generated, which are comply with the IT controller.

Z-N is the widely used tuning method. At first K_D and K_I gain are set to zero. Then K_P is increased until the output of the loop starts to oscillate in the oscillation period [3].

Even though Good Gain method is not used as much as Z-N method, in many applications it gives better controlling and stability [4]. At first K_D and K_I gain are set to zero and then increase or decrease the K_p value until some overshoot and a barely observable undershoot are observed.

There is no mathematics in Try and Error method. Tuning procedure will takes a lot of time to obtain the best result.

2. METHODOLOGY

2.1. Experimental Setup

A lab scale liquid level controller was used in this study. Controlling and Real time data acquisition facility were setup with TI and Arduino UNO boards. Host computer communicates with these DSP boards by using serial ports. Schematic diagram of the experimental setup is shown in Figure 4.

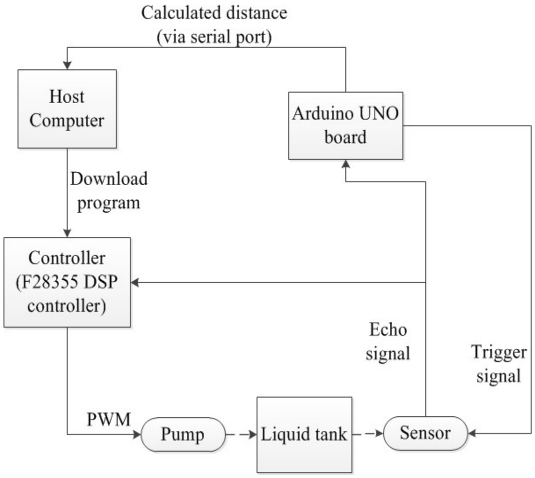

Figure 4: Experimental setup

2.2. Data acquisition and controlling

An ultrasonic sensor (HC-SR04) is used to measure the level of the tank. It works by trigger signal and time for receiving echo signal can be used to calculate the distance from the sensor to the water surface. Trigger signal to sensor is send by using an Arduino UNO board. The echo signal is received to both Delfino F28355 board and Arduino UNO board. Arduino board calculates the level height and send calculated values to Labview via serial port of the host computer in a 51 milliseconds interval. Using the real time data, graphs is obtained by the host computer.

The Simulink Model was developed to calculate the length of echo (pulse width, T: Refer Figure 5). The developed Simulink mathematical model is shown in Figure 6 and Figure 7. The main model is shown in Figure 6 and 'Calculate level (cm)' subsystem of the main model is shown in Figure 7.

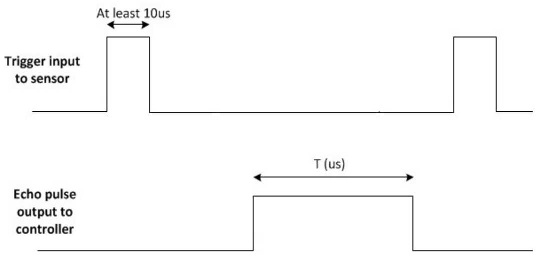

Figure 5: Trigger and Echo signals

'ADC' block converts analog echo signal to a digital values. The level height is calculated in 'Calculate level (cm)' subsystem. In 'Calculate level (cm)' subsystem of Figure 7, from 'In' to until 'If Action subsystem', T us (Refer Figure 5) is calculated. To convert T to the distance (cm), say 'h', T is multiplied by '1000000/58.2' (Gain). This calculated distance is the height between the sensor and the water surface of the tank. Because the tank height is 100 cm, '100-h' cm gives the liquid height in the tank. Then the 'Out' is the Height of the liquid in the tank. Then according to the error (Level height – Set point), PID controller control the percentage value (0 to 100%) of 'ePWM' block which controls the output voltage value of relevant PWM pin of F28355 board. From this output PWM signal, the pump is controlled.

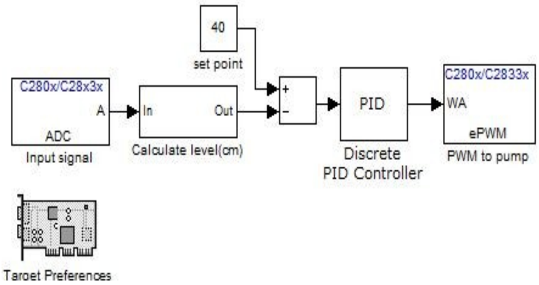

Figure 6: Control structure with PID Controller

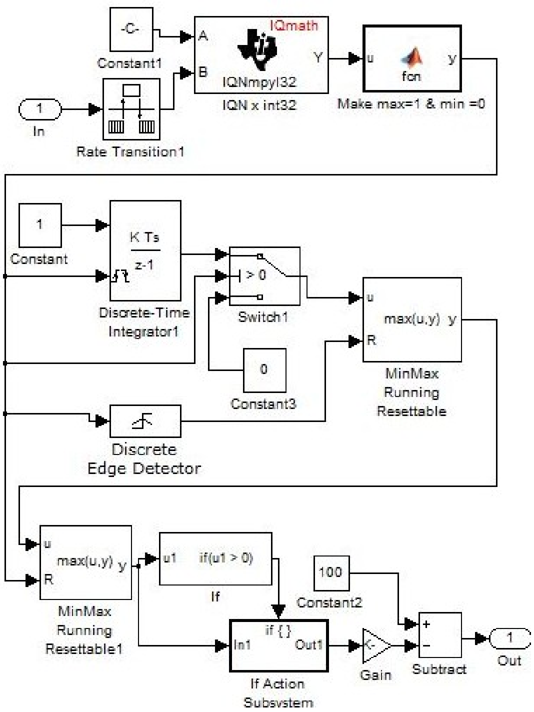

Figure 7: 'Calculate level (cm)' subsystem

The Code Composer Studio v3.3, which is an integrated development environment, can execute C code for the micro-controller. Then Code Composer Studio uploads the generated code to Delfino F28355 board.

3. RESULTS

PID controller was tuned by the Z-N method as well as by the G-G method. Tuning parameters (K_p, K_i and K_d) by each method are given in the Table 1. Performance of each tuning procedures are summarized in Table 2. Trial error methods was also used in this study, due difficulties of smooth outputs.

Table 1: Parameter of PID Controller

Method	Parameter		
	K_p	K_i	K_d
G-G	37.1	6.137	0
Z-N	15.75	3.885	0
Trial & Error	80	0.01	0

Table 2: Performance of PID controller

Method	Tr (s)	Ts (s)	OS (%)	SSE (cm)
G-G	8.98	23.94	4	0 to 0.7
Z-N	10.91	34.55	6.36	0 to 0.3
Trial & Error	10.13	12.2	0	0 to 0.3

Figure 8, 9 and 10 are the performance for each method respectively.

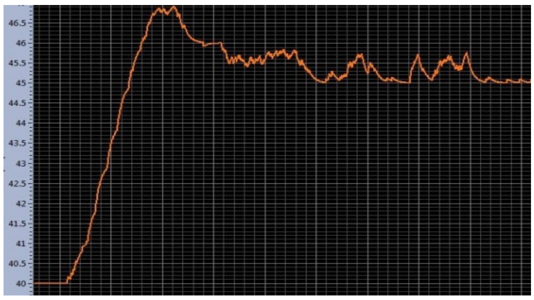

Figure 8: Performance response using G-G method

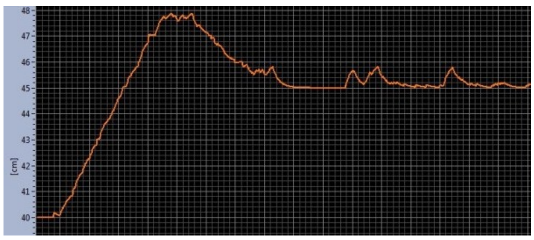

Figure 9: Performance response using Z-N method

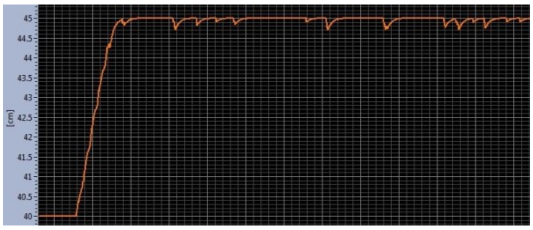

Figure 10: Performance response using Trial and Error method

4. DISCUSSION

Getting serial data through the F28355 DSP board for data plotting was very difficult and therefore an Arduino UNO board was used for that purpose.

It was quite difficult for finding the parameters. It required a lot of effort and experiences to obtain a good-gain of the controllers. For example getting a well-damped slight overshoot in G-G method and sustained oscillation in Z-N method were very difficult. The reason may be the power input to the pump is not proportional with flow rate and pump characteristics are nonlinear.

Trial & Error method had the fastest time for the system to reach the stable condition in the system. From the tuning method of G-G, the response of the system is the second fastest compared to Z-N method. However, G-G method shows that it has the fastest time in the rise time in the system followed by Trial and Error and Z-N. For Z-N method had the highest percentage of the overshoot and Trial & Error method had the lowest percentage of the overshoot (almost zero). Therefore it is clear that Trial & Error method gives the better performance to the system over other methods. Performances of each tuning methods are shown in the Table 2. Steady state error of Trial & Error method is much better than others.

5. CONCLUSION

TI Delfino F28355 DSP board wad used to control the level of a tank by PID controlling. A Simulink model for PID controlling was developed and uploaded to the DSP board by Code Composer studio v3.3. The parameters of the PID controller were tuned by the traditional method such as G-G, Z-N and Trial and Error. Even though it is quite difficult to find tuning parameters, satisfactory performance can be achieved by above all tuning methods. However, Trial & Error method gives the better performance to the system over other methods.

6. REFERENCES

[1] H. I. Jaafar, S. Y. S Hussien, N. A Selamat, M. S. M. Aras, and Z. *A. Rashid, " Development of PID controller for controlling Desired Level of Coupled Tank System*", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 3, issue 9, pp. 2278-3075, 2014.

[2] "*TMS320F28335 Delfino microcontroller* ", Texas Instruments, At: http://www.ti.Com/product/ tms320 f28335, [Retrived 3rd March 2015].

[3] J. G. Ziegler, and N. B. Nichols, "*Optimum setting for automatic controllers*", Transactions of ASME, vol. 64, pp. 759-768, 1942.

[4] F. Haugen, "The good gain method for PI(D)", from TechTeach, At: http://techteach.no/ publications/articles/good_gain_method/good_gain_method.pdf. [Retrieved 3rd march 2015].