Simulation, Design and Practical Implementation of IMC tuned Digital PID controller for liquid level control system.

Sandip A. Mehta, Jatin Katrodiya and Bhargav Mankad

***Abstract*-- The paper is mainly concerned on Liquid level control systems which are commonly used in many process control applications to control, for example, the level of liquid in a tank. Liquid enters the tank using a pump, and after some processing within the tank the liquid leaves from the bottom of the tank. The requirement in this system is to control the rate of liquid delivered by the pump so that the level of liquid within the tank is at the desired set point. IMC tuned PID algorithm is implemented in MICROC programming language and it is loaded into the PIC microcontroller. The controller generate the output according to the error signal and derives the system towered the zero error. It is also interfaced with PC through MAX232 and DB9 connector for system identification and the observation of output of the system.**

***Index Terms*—ADC,DAC, IMC, MICROC, PIC, PID**

I. INTRODUCTION

The Internal Model Control (IMC) philosophy relies on the Internal Model Principle, which states that control can be achieved only if the control system encapsulates, either implicitly or explicitly, some representation of the process to be controlled. In particular, if the control scheme has been developed based on an exact model of the process, then perfect control is theoretically possible. In practice, however, process-model mismatch is common; the process model may not be invertible and the system is often affected by unknown disturbances. Thus the above open loop control arrangement will not be able to maintain output at setpoint. Nevertheless, it forms the basis for the development of a control strategy that has the potential to achieve perfect control. This strategy, known as Internal Model Control (IMC) has the general structure depicted in Fig. 1

IMC-based PID Controller

The internal model control (IMC) algorithm is based on the fact that an accurate model of the process can lead to the design of a robust controller both in terms of stability and performance [1]. The basic IMC structure is shown in Figure 1 and the controller representation for a step perturbation is described by (1).

$$G_q(s) = \frac{G_f(s)}{G_{mm}(s)} \tag{1}$$

where

$G_{mm}(s)$ is the inverse minimum phase part of the process model and

$G_f(s)$ is a n$^{\text{th}}$ order low pass filter $1/(\lambda s + \;)^n$. The filter's order is selected so that $G_q(s)$ is semi-proper and λ is a tuning parameter that affects the speed of the closed loop system and its robustness [2].

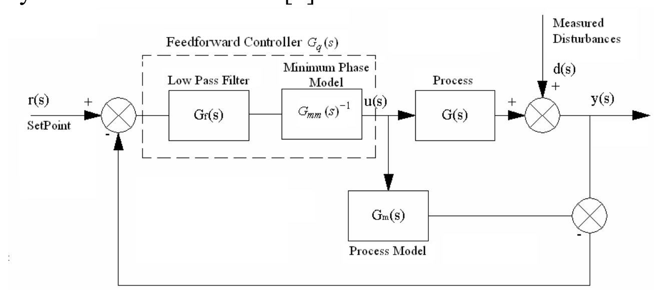

Figure 1: IMC control structure

However, there is equivalence between the classical feedback and the IMC control structure, allowing the transformation of an IMC controller to the form of the well-known PID algorithm.

$$G_c(s) = \frac{G_q(s)}{1 - \tilde{G}_m(s)G_q(s)} \tag{2}$$

The resulted controller is called IMC-based PID controller and has the usual PID form (3).

$$G_c(s) = K_p\left(1 + T_D s + \frac{1}{T_I s}\right) \tag{3}$$

IMC-based PID tuning advantage is the estimation of a single parameter λ instead of two (concerning the IMC-based PI controller) or three (concerning the IMC-based PID controller). The PID parameters are then computed based on that parameter [1]. Though for the case of a FOPDT process model, the delay time should be approximated first by a zero-order Padé (usually) approximation [3]. However, the IMC-based PID tuning method can be summarized according to the following Table 1 [2].

Controller	$K_P K_c$	T_I	T_D	λ/θ
IMC-based PID	$\frac{2\tau+\theta}{2\lambda+\theta}$	$\tau+\frac{\theta}{2}$	$\frac{\tau\theta}{2\tau+\theta}$	>0.8

Table 1: IMC-based PID tuning parameters of a FOPDT process

Recent [4-6] simulation studies shows that IMC tuned PID parameter provides better performance than S-F based or ZN based tuning methods. In this paper we design a practical set up for implementation and verification of digital IMC tuned PID algorithm on PIC microcontroller.

II. Practical Implementation Of IMC Tuned PID Parameters

1) System Design

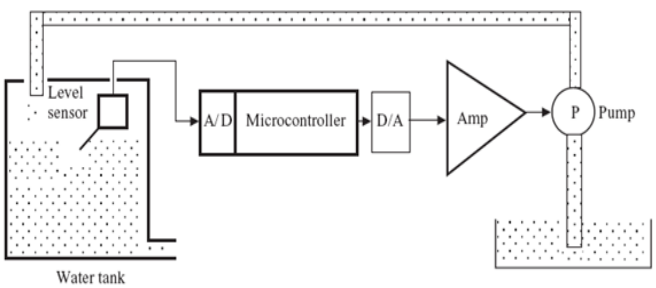

Fig.2 Basic block diagram of liquid level control system

The basic schematic diagram of the system is as shown in the figure 2. The list of components used are as follows.

1.1 Water tank:

This is the tank where the level of the liquid inside is to be controlled. Water is pumped to the tank from above and a level sensor measures the height of the water inside the tank. The microcontroller controls the pump so that the liquid is at the required level. The tank used in this setup is a stainless steel container.

1.2 DC Pump:

The pump is a small 12V dc water pump drawing about 3A when operating at the full-scale voltage

Figure 3 DC Pump

1.3 Level sensor:

Level Sensor Selection

When determining what type of level sensor should be used for a given application, there are a series of questions that must be answered:

- Can the level sensor be inserted into the tank or should it be completely external?

Should the sensor detect the level continuously or will a point sensor be adequate?

- Can the sensor come in contact with the process fluid or must it be located in the vapor space?
- Is direct measurement of the level needed or is indirect detection of hydrostatic head (which

or preferences of the particular plant or the

A rotary potentiometer type level sensor is used in this setup. The sensor consists of a floating arm connected to the sliding arm of a rotary potentiometer. The level of the floating arm, and hence the resistance, changes as the liquid level inside the tank is changed. A voltage is applied across the potentiometer and the change of voltage is measured across the arm of the potentiometer. The resistance changes from 100Ω when the floating arm is at the bottom to 10Ω when the arm is at the top.

The level sensor is shown in Figure 1.5.

Figure.4.Float type Level Sensor

1.4 Internal diagram & working of level sensor:

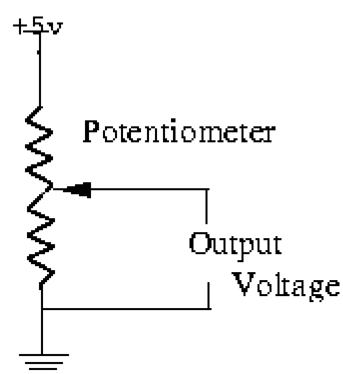

Figure.5 Principle of liquid level sensor.

A potentiometer type level sensor is a manually adjustable resistor. The way this device works is relatively simple. One terminal of the potentiometer is connected to a power source. Another is hooked up to ground (a point with no voltage or resistance and which serves as a neutral reference point), while the third terminal runs across a strip of resistive material. This resistive strip generally has a low resistance at one end; its resistance gradually increases to a maximum resistance at the other end. The third terminal serves as the connection between the power source and ground, and is usually interfaced to the user by means of a knob or lever. The user can adjust the position of the third terminal along the resistive strip in order to manually increase or decrease resistance. By controlling resistance, a potentiometer can determine how much current flow through a circuit. When used to regulate current, the potentiometer is limited by the maximum resistivity of the strip.

1.5 Microcontroller:

A PIC16F877 type microcontroller is used in this setup as the digital controller. In general, any other type of microcontroller with a built-in A/D converter can be used. The PIC16F877 incorporates an 8-channel, 10-bit A/D converter.

1.6 D/A converter:

An 12-bit MCP4921 type D/A converter is used in this setup. In general, any other type of D/A converter can be used with similar specifications. The complete system is as shown in the fig.

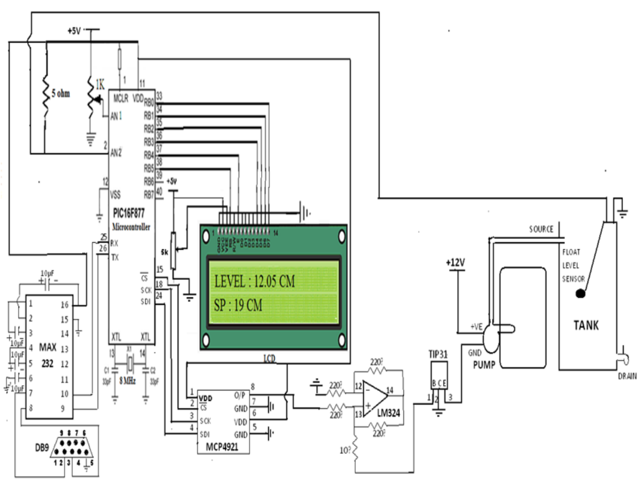

Figure. 6: Complete System

2) Model Identification

First of all for the identification of the system, we interface the system to pc by RS232 to serial communication port. Then we use the Instrument Control Toolbox of the MATLAB SIMULINK for data acquisition and plotting the response of the system. We plot graph of LEVEL IN THE TANK VS TIME. Then we use this data for the system identification. We use IDENT command of the MATLAB for the identification of the system.

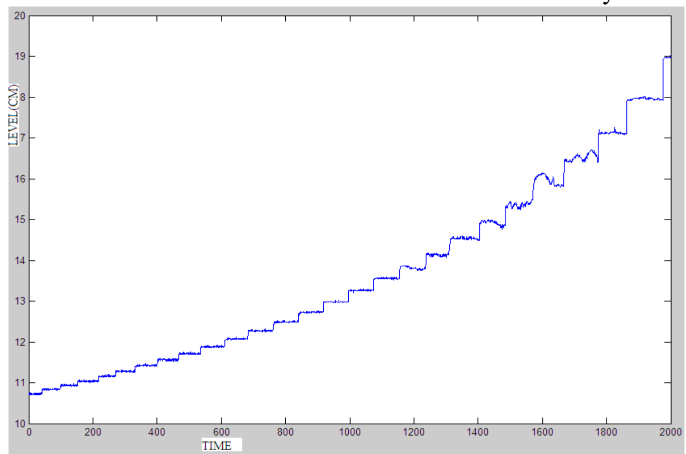

Figure 7: Step resonose of the open loop system

Following figure(7) shows the open loop response of the system,.

Then we import that data in identification tool of the MATLAB following figure shows the process of data importing and process identification. The response of the LTI viewer is as shown in the figure 8.

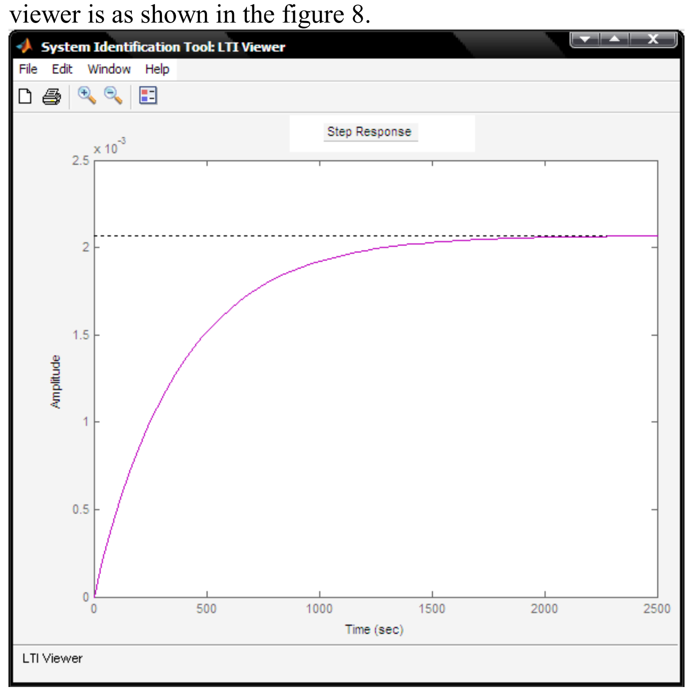

Figure 8 : Response of the systme on LTI viewer

We get the transfer function of the system as follows:

$$G(s) = 1.88 / (410s+1) \qquad (4)$$

3) PID Implementation.:

A MATLAB based script is used to obtain the PID parameters with the help of Table(2). The obtained PID parameter values are :

Kp = 1.06

Ki = 0.05

Kd = 0.4875

Most PID controllers nowadays are digital. For digital implementation, discrete time domain is considered. Here, the value of sampling time is of utmost importance.
The velocity algorithm is based on splitting the calculation of the control value into two steps:
1. First the incremental control value Δu(tk) is calculated.
2. Then the total or absolute control value is calculated with

u(tk) = u(tk−1) + Δu(tk−1)
Thus, the final discrete PID velocity algorithm is:

$$\Delta u(t_k) = K_p\,[\,e(t_k) - e(t_{k-1})] + \frac{K_p\,h}{T_i}e(t_k) + \frac{K_p\,T_d}{h}\,[e(t_k) - 2e(t_{k-1}) + e(t_{k-2})]$$

$$u(t_k) = \Delta u(t_{k-1}) + \Delta u(t_k)$$

(5-6)
As opposed to the fixed control reference used in the positional algorithm, here, the calculation of current control uses the previous control value as reference. In essence, the control is calculated as a change, hence the term ‘velocity form’.

III. Complete IMC Tuned PID Result

A. Set Point Tracking

A MATLAB based GUI is developed for interfacing the practical system with the software. The set point tracking is done through the MATLAB based GUI and load variation given manually to the system. As shown in the Figure 9. the MATLAB based GUI is used to interface with the system. Here the range is 0 to 300 mm has been set for plotting the response of the system.

B. Disturbance Rejection

Here as shown in the Figure 10 when the load variation is applied system is settle down quickly to the set point value of 14.3 cm . Disturbance is applied to the system manually by opening another drain valve of the system for 1 minute. The second valve is closed then after however the main drain valve is still remaining in open condition. As shown in the figure 10 system is able to reject the disturbance also without any overshot condition.

C. Load variation

In first part of the figure 10 as shown, the main drain valve is slightly closed during the normal operation. The process value is suddenly reached to the 150mm. but again it came back to the normal set point value that is 143mm..

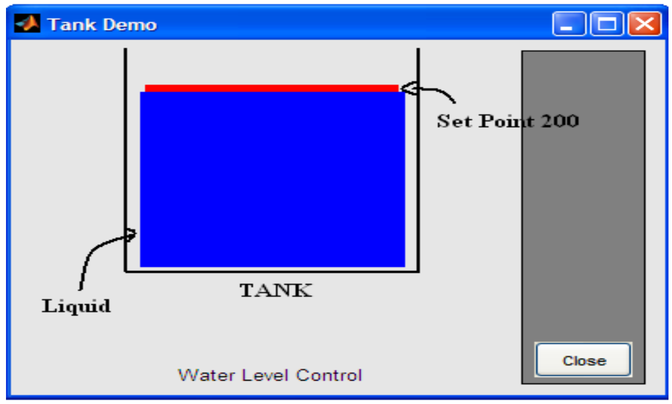

Figure 9 : MATLAB Based GUI for setpoint adjustment

.

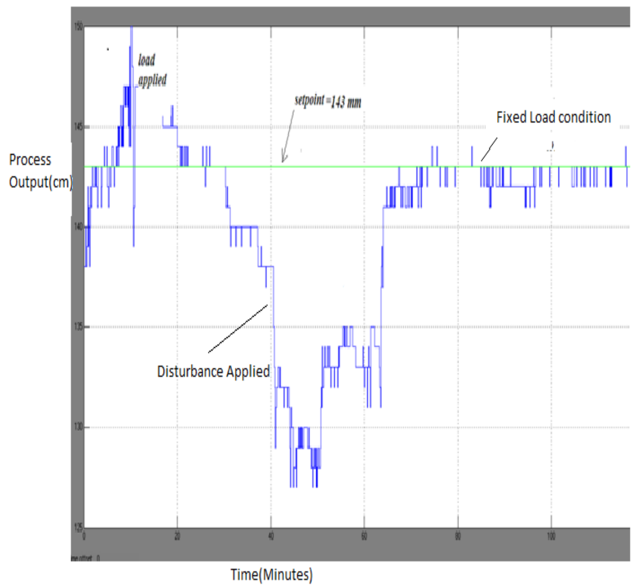

Figure 10 : Process output showing load variation and disturbance rejection

IV. Conclusion

We have successfully implemented IMC based level control system starting from identification of system to be implementation suitable control algorithms in microcontroller. We have successfully tuned the system using matlab simulation. The comparison of system responses shows that a system using a IMC based digital PID controller with tuning parameters derived from auto-tuning shows a much better system response than conventional PID controller. The system response has fewer oscillations, less settling time and rise time and there is no steady state offset as compared to open loop system and system with proportional controller which shows that the tuning parameters derived were accurate. Due to DC pump is used in the system this system does not required any kind of pneumatic supply and I/P converter for operating pneumatic pump which make the system least bulkiest and cost also reduces.

Starting from the development of DC pump based liquid level control system, from our best of knowledge this type of work is not carried out anywhere to verify the digital PID parameters obtained from IMC tuned system on a practical liquid level system.

V. References

[1] Coughanour, D. *Process Systems Analysis & Control* NY, McGraw-Hill c1991 Chemical Engineering Series, 1991

[2] Bequette, B. W., *Process Control: Modelling, Design and Simulation*. Prentice Hall, Upper Saddle River, NJ, 2003

[3] Morari, M., & Zafiriou, F., *Robust Process Control.* Prentice Hall, Englewood Cliffs NJ, 1989

[4] K. Anil Naik et al. Stability Enhancement of DC Motor using IMC Tuned PID Controller (IJAEST) NTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES Vol No. 4, Issue No. 1, 092 - 096, 2011

[5] S. Umamaheswari,et al. simple method of tuning PID controllers for interval plant of cold rolling mill, IEEE International Conference on Control, Automation, Communication and Energy Conservation, 2009. INCACEC 2009.

[6] Kokate, R.D.et.al; Review of Tuning Methods of DMC and performance evaluation with PID Algorithms on a FOPDT Model, IEEE International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), 2010.