Eng.J.CMU.[2021] 28 [1], 153-168

วารสารวิศวกรรมศาสตร์
มหาวิทยาลัยเชียงใหม่
Engineering Journal
Chiang Mai University

Comparison of Ziegler-Nichols and Cohen-Coon Tuning Methods: Implementation to Water Level Control Based MATLAB and Arduino

Thanita Suksawat and Pornsiri Kaewpradit*

Department of Chemical Engineering, Faculty of Engineering,
Prince of Songkla University, Songkhla 90112, Thailand
*E-mail: pornsiri.k@psu.ac.th, Tel.: +6686-888-4640

ABSTRACT

The purpose of this study is to design a proportional (P), proportional-integral (PI), and integral derivative (PID) controller for the water level control system. The system uses Arduino as a data acquisition running through MATLAB/Simulink. Tuning methods, Zeigler-Nichols (ZN) and Cohen-Coon (CC), are based on a first order plus dead time (FOPDT) model and open-loop tuning, and the results were compared. Due to the fast development of the process industry, the higher accuracy of the system is required. Kalman filter was also applied in this study to compensate for the errors of both water level measurement and the process model. Experimental results are shown for comparison of those tuning methods without Kalman filter and the best controllers of ZN and CC tuning methods is PI controller with Kalman filter. The rise time and settling time of the ZN-PI controller with Kalman filter are 40.3 s and 170 s, respectively. The rise time and settling time of the CC-PI controller are 39.3 s and 43.0 s, respectively. The CC-PI controller with Kalman filter has a better performance with a smaller rise time and settling time. After several tests with different tuning methods, this proves the useful application and the efficiency of Kalman filter.

Keywords: Level control; PID; Arduino; MATLAB/Simulink; FOPDT (First Order Plus Dead Time); Kalman filter

1. INTRODUCTION

In industrial processes, the control system is widely used to make more precise of their products, which leads to a demand for a successful control system in both practical and theory [1]. Level control is an important part of industry such as chemical engineer, nuclear power generation. An inefficient control system could failure plant output specification [2]. The water level control system is very complex because of its nonlinearity and uncertainty [3]. More than 90% of control loops in process industries are implemented by using the Proportional Integral Derivative (PID) controller due to simplicity, effectiveness, and easy to understand control performance. The three parameters in the PID controller are adjusted to handle the desire values of the industrial processes

Received 30 November 2020
Revised 19 March 2021
Accepted 25 March 2021

[4]. Due to the fast development of process industry, the higher accuracy of system is required. The control method based on a process model. Telepatil et al. (2017) [5] showed a system that interfacing between the Arduino board and MATLAB to control and monitoring household appliances. The Arduino board was used as the brain of the system. The commanding signals were given through MATLAB to the Arduino board via serial communication. The continuous monitoring and control of home appliances were done by the cooperation of Arduino hardware and MATLAB software. Anarase et al. (2016) [6] designed a closed-loop water level system and implemented the model in a simulation environment based on MATLAB. The controlled variable of the process was level, and the manipulated variable was the flow rate. PID controller was designed for the water level system based FOPDT model. This level loop was configured with Supervisory Control and Data Acquisition System (SCADA), which has high cost. However, a low-cost microcontroller is another interesting option. Arduino has been widely used. Due to the fact that it is an open-source platform, cheap, easily programmable, and easily communicated with MATLAB [7]. The simulation results for PID controller tuning by the open-loop tuning methods such as Ziegler Nichols (ZN) & Astrom Hagglund shown that more accurate results came using Astrom Hagglund PID Controller over Ziegler-Nichols PID controller. Babu et al. (2020) [8] focused on maintaining the water level in the storage tank and determined by ZN methods with P, PI, and PID controllers. They compared the response characteristics of the controllers, the results showed PID controller minimized the steady-state error, but the PI controller had the smallest rise time. The tuning method plays a very vital role. The values of the parameters in the controller can affect the performance of the system. To ensure the efficiency of one of the tuning methods, the comparison of tuning methods is important. Cohen-Coon (CC) tuning method is the second popular after the ZN tuning method because it is more flexible than the ZN tuning method in a wider variety of processes. The CC tuning method is reasonable for the process that the dead time is less than two times of the time constant, but ZN tuning method works well only on the process that the dead time is less than half of the time response [9]. Kapale et al. (2016) [10] proposed a liquid control system and estimated the actual level of the tank from noisy measurements by using the Kalman filter algorithm to reduce noise in liquid level measurement system due to dynamic environment, such as sloshing. The results showed that the Kalman filter can reduce the noise from sloshing and get a smoother output value. The Kalman filter optimized for level measurement. Yumurtaci et al. (2020) [11] controlled liquid level by using MATLAB/Simulink and Arduino Due board. The manipulated variable is the power of the pump operating with the PWM technique. Liquid level is carried out with on-off controller, PID, ANN-PID and Fuzzy-PID controller. The result showed that the pump is driven at full power or disabled–resulting in the oscillation of liquid level, and Fuzzy-PID controller gave the fastest response. In literatures, there are rare studies demonstrated the implementation of the Arduino with Kalman filter technique and the comparison of ZN and CC tuning

methods for water level control system via MATLAB/Simulink.

In this study P, PI, and PID controllers, simple method purposed by ZN and CC were implemented for real-time measurement of water level control system, and control system achieved by using Arduino UNO board as a data acquisition running through a computer by using "Arduino IO library" in MATLAB/Simulink, the software of the control system was created without code need. The real-time result monitoring consisted of the desired level, actual level, error signals, and control signals via MATLAB/Simulink. The performances of P, PI, and PID controllers were compared with different tuning methods and examined the best controller for this water level control system. This study manipulated the PWM of a solenoid valve, the valve is fully open if PWM is 255 and disabled if PWM is less than 255. However, the level control process has a dynamic environment from the spattering of water, which takes high time consumption for the calculation of the controller to track the desired value, as well as the model mismatches like parameter changes, system non-linearities, and saturation effects [12]. These problems were solved in this work by using the Kalman filter.

2. DESCRIPTION OF THE PROCESS

Figure.1 showed the process diagram of the water level control in the lab scale. Water in the storage tank was pumped into the system through a flow indicator and solenoid valve. An ultrasonic sensor was installed at the top of the water column, its signal was sent to the Arduino board to calculate the level of water. The controller sent a pulse-width modulation (PWM) signal to a digital solenoid valve. Here, the normally open solenoid valve was open when the PWM signal ≥ 255, and it was closed when the PWM signal < 255. Table.1 shows the specification of the instruments were used in the process.

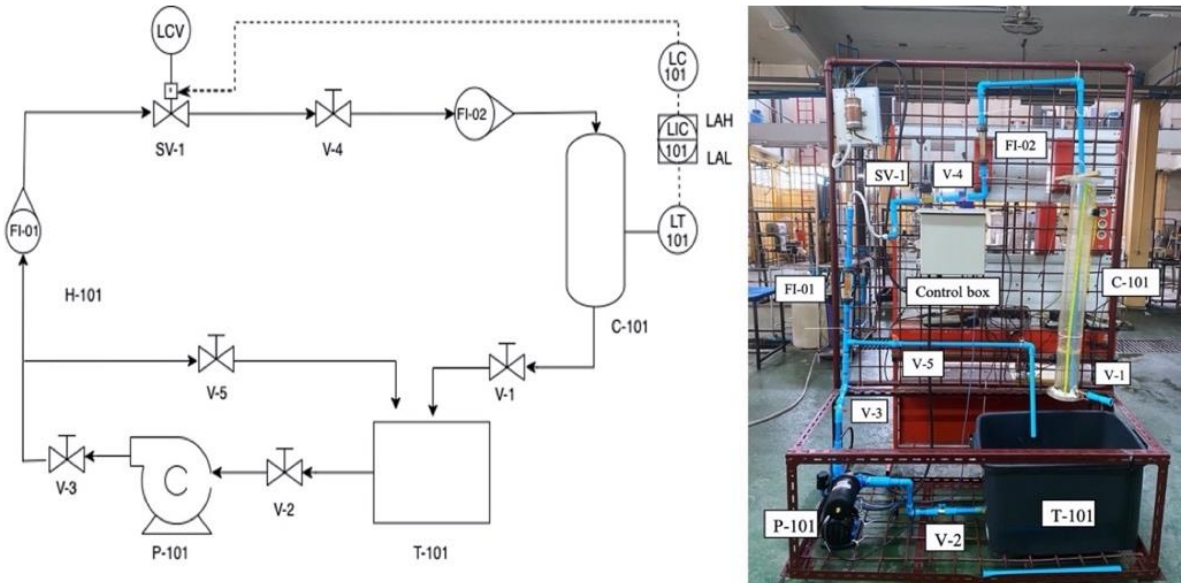

Figure 1 Diagram of level control (Left) and lab-scale system (Right).

Table 1 Specification of the process

Equipment code	Equipment type	Specification
C-101	Water column	Acrylic: Height 101 cm, Diameter: 10 cm
V-1, V-2, V-3, V-4, V-5	Manual valve	Ball valve, size 1/2"
FI-01, FI-02	Rotameter	Range flow rate: 1-10 LPM
P-101	Pump	Maximum flow rate: 38 L/min, Maximum head: 35 m
SV-1	Solenoid valve	Brass, size ½'', 12VDC
T-101	Water tank	Plastic, contained 100 L
L-101	Ultrasonic sensor	US-016 model, detection distance 2 cm- 300 cm

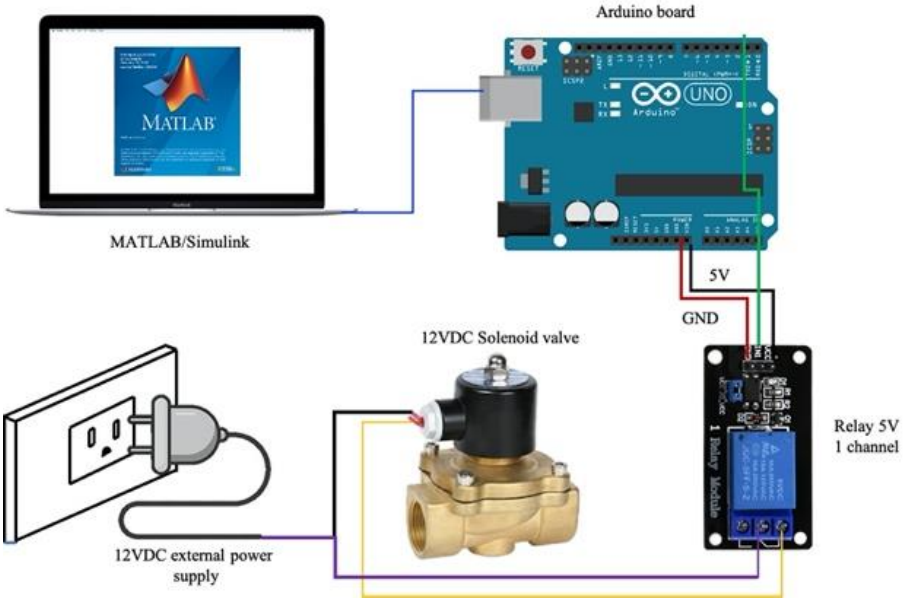

Figure 2 Diagram of the Solenoid valve with Arduino board

Figure 2 showed the diagram of the final element of the process. Arduino board connected with MATLAB/Simulink through the USB port. Relay is a switch that is used to close and open circuits electronically. It will normally open (NO) when voltage is applied to the relay/contactor terminals, this contact closes. Normally closed (NC) when the relay is not energized, when voltage is applied, this contact opens and interrupts the current. Relays are generally used to switch currents in a control circuit such as small motors and low amps solenoids. In this study uses the NC relay 5V 1 channel to switch the currents. A relay can prevent the damage of equipment by detecting abnormalities in electrical circuits such as overloads, overcurrent, undercurrent, and reverse currents [13]. The 12VDC solenoid valve is connected with a 12VDC external supply power supply. The external power supply converts ac power into lower voltage dc or ac power to be used directly by electronic circuits [14]. The solenoid valve is connected with a relay to adjust the opening from MATLAB/Simulink demands.

The process is tested by the Arduino board through MATLAB/ Simulink. The real time water levels are collected in the workspace in MATLAB program. All data is plotted to the graph to find the transfer function of the process and the parameters of the transfer function which indicate the behavior of the process are used in tuning methods.

3. CONTROLLER DESIGN

3.1 P controller

P controller is one of the control systems. For an open loop, the proportional gain can change as controller gain (K_c) and the closed loop dynamics will occur. If controller gain is large. It will result in an unstable response [15].

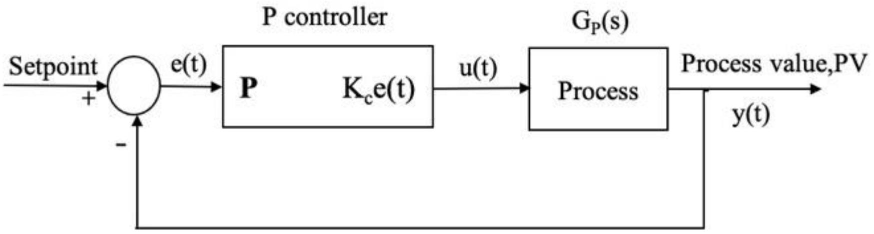

Figure 3 The general structure of the P controller.

The proportional term is expressed in equation (1).

$$MV(t)=K_c e(t) \quad (1)$$

While $MV(t)$ is the process input or manipulated variable, K_c is controller gain and $e(t)$ is the difference between the desired output and the measured signal.

3.2 PI controller

PI controller will minimize the steady-state error. However, the integral action has a disadvantage that affect the speed of the system. Thus, proportional action was added to increase the response of the process and eliminate the steady state error from a pure proportional controller, but integral term can cause overshoot [15].

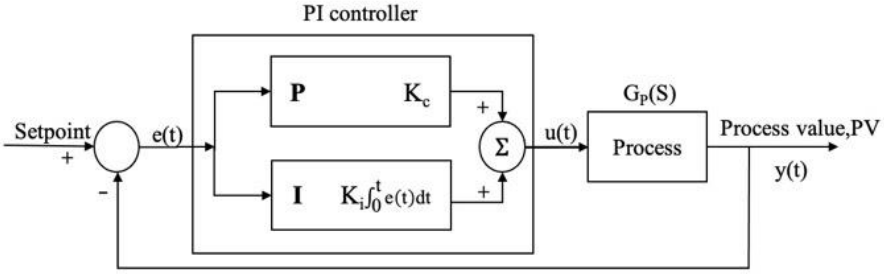

Figure 4 The general structure of the PI controller.

The proportional-integral controller is given by

$$\mathrm{MV(t)}=\mathrm{K_c e(t)} + \mathrm{K_i}\int_0^t \mathrm{e(t)dt} \quad (2)$$

3.3 PID controller

PID controller is popularly used in industries. The controller algorithm combines the actions of three parameters based on the error signal, which is the difference between the desired output and the measured signal (as shown in equation (3)). The controller attempts to bring the present output to the desired value by minimizing the errors, and adjusting the process input or manipulated variable, $\mathrm{MV(t)}$.

$$\mathrm{MV(t)}=\mathrm{K_c e(t)} + \mathrm{K_i}\int_0^t \mathrm{e(t)dt} + \mathrm{K_d}\frac{\mathrm{d}}{\mathrm{dt}}\mathrm{e(t)} \quad (3)$$

where $\mathrm{K_c}$, $\mathrm{K_{i,}}$ $\mathrm{K_d}$ are controller gain, integral gain, and derivative gain, respectively. To calculate the controller, these three parameters are summed, denoted by P, I, and D, respectively (as shown in Figure 5). While P depends on the present error and I is the accumulation of past errors and D predicts future errors [1]. These parameters affect the process if $\mathrm{K_c}$ and $\mathrm{K_i}$ are too high resulting in high offset and high overshoot, respectively, while $\mathrm{K_d}$ can reduce the overshoot caused by $\mathrm{K_i}$ [1].

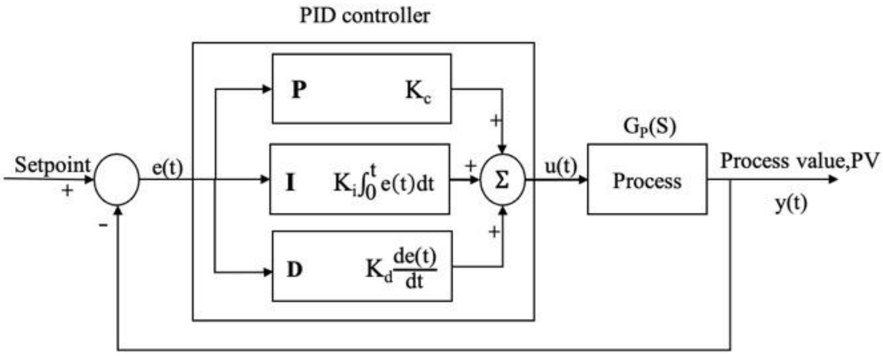

Figure 5 The general structure of the PID controller.

3.4 Tuning methods

Tuning method is the determination of the parameters of PID controller values for getting the optimum performance or the acceptable performance from the process [16]. Table 2 shows various tuning methods for FOPDT model in this study. In the process control system, better performance is accomplished by adjusting the control parameters to provide the desired process responses [12]. In this work, those parameters were designed by ZN, and CC tuning methods based FOPDT model. The PID tuning method is the determination of the PID parameters for getting the desired, acceptable, and fast process performances. The tuning steps involved the dynamic personalities of the control loop and the evaluation of the tuning parameters. In this study, open-loop tuning was used. A unit step response of the experimental process appears an S-shaped curve as shown in Figure 6 Delay time (L), a time constant (T), and process gain ($\mathrm{K_p}$) were obtained, and the control parameters ($\mathrm{K_c}$, $\mathrm{T_i}$, $\mathrm{T_d}$) were calculated based on two different tuning methods as shown in Table 2.

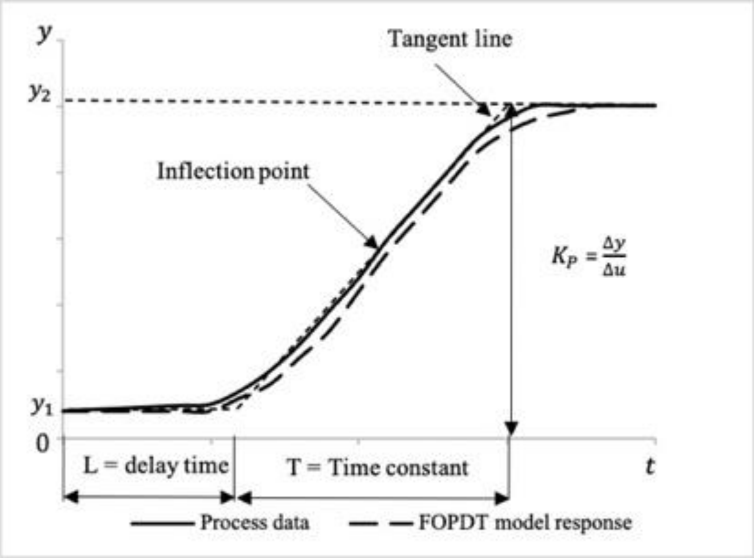

Figure 6 Illustration of the FOPDT model determination.

The feedback control system is shown in Figure 5 The process $G_P(s)$ is a FOPDT model shown by the following transfer function in equation (4).

$$G_P(s) = \frac{K_P}{1+Ts} e^{-Ls} \tag{4}$$

Equation (5) and Equation (6) shown the controller is the PID type.

$$G_c(s) = K_c\left(1+\frac{1}{T_i s}+T_d s\right) \tag{5}$$

$$G_c(s) = K_c + \frac{K_i}{s} + K_d s \tag{6}$$

where $K_i = K_c/T_i$ while T_i is the integral time constant, and $K_d = K_c \cdot T_d$ while T_d is the derivative time constant. Three tuning methods have been considered in this work to estimate the three parameters by performing a simple experimental process. They are based either on a closed-loop feedback system or an open-loop step response [17]. In this study, ZN, and CC tuning method for the FOPDT model are considered.

Table 2 Tuning methods for FOPDT model in this study.

Methods	Type of controllers	Parameters		
		Proportional gain (K_c)	Integral time (T_i)	Derivative time (T_d)
Ziegler-Nichols [18]	P	$\frac{T}{L}$	∞	0
	PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
	PID	$1.2\frac{T}{L}$	2L	0.5L
Cohen-Coon [19]	P	$\frac{T}{K_P L}\left[1+\frac{L}{3T}\right]$	∞	0
	PI	$\frac{T}{K_P L}\left[\frac{9}{10}+\frac{L}{12T}\right]$	$\frac{L\left(30+\frac{3L}{T}\right)}{9+\frac{20L}{T}}$	0
	PID	$\frac{T}{K_P L}\left[\frac{4}{3}+\frac{L}{4T}\right]$	$\frac{L\left(32+\frac{6L}{T}\right)}{13+\frac{8L}{T}}$	$\frac{4L}{11+\frac{2L}{T}}$

4. KALMAN FILTER

Kalman filter is widely known as an effective method that gives good estimation results under noisy surroundings [20]. It estimates the unobserved variables based on imprecision and uncertainty of measurements and the model parameters through a series of predictions and corrections, even when the modeled system has the unfamiliar precise nature [21]. Also, the Kalman filter predicts the future system state based on the early estimations. The Kalman filter has been applied to many industries such as aerospace systems, vehicle systems, robots, power prediction, weather forecasts, etc. [20]. The schematic diagram of the Kalman filter with the PID controller is shown in Figure. 7 where u_k is called a vector of inputs, $\tilde{y}_k$ is a vector of the measured process outputs, and $\hat{y}_k$ is the estimated measured outputs.

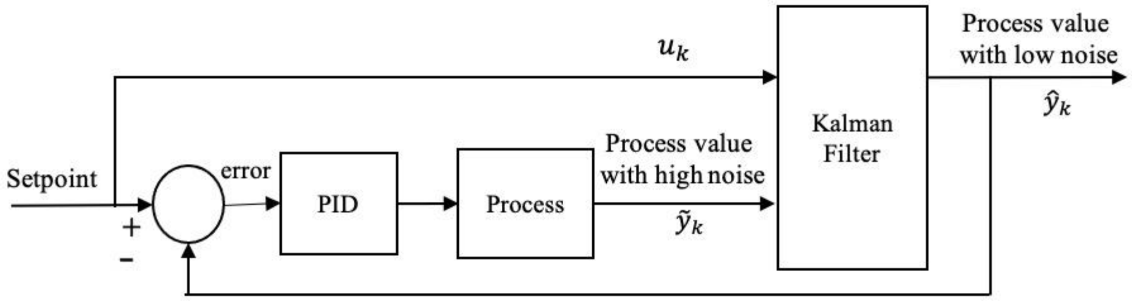

Figure 7 The schematic diagram of the Kalman filter.

A state-space model is used in the Kalman filter estimation algorithm, which represents the association of input value and the output value of the process. The process has uncertainty, which is "Measurement noise", errors from measurement, and "Process noise", errors from the process model. Because of measurement noise and process noise, provide the process values miss the desired values. Kalman filter tries to reduce the estimation error by adjusting parameters. The standard state-space form expressed in equation (7) to (9).

$$x_k = A_k x_{k-1} + B_k u_k + w_k \quad (7)$$

$$y_k = C_k x_k \quad (8)$$

$$\tilde{y}_k = y_k + v_k \quad (9)$$

where x_k is a vector of the present states at step time k. y_k is a vector of the present process outputs at step time k. C_k is the matrix which is the relationship of the actual state and the measurement Kalman filter. A_k and B_k are the state matrix and the control matrix, respectively. w_k and v_k are process and output noise with covariance matrices Q and R.

The Kalman filter consists of a 2- step algorithm; the predictor step and the correction step.

The predictor step involves the current state estimation and the error covariance estimation from the current time forwards in time to calculate a predicted estimation (or *a-priori*) of the states at the current time. The predictor step is shown by equation (10) to equation (11),

$$\hat{x}_k^- = A_k\hat{x}_{k-1} + B_k u_k \quad (10)$$

$$P^-_k = A_k\hat{P}_{k-1}A_k^T + Q_k \quad (11)$$

And the corrector step as shown in equation (12) to (14), corrects the predicted estimated calculated state in the first step by consolidating the recent process measurement to generate an updated state (or *a-posteriori*) estimation. The corrector step is given by,

$$K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1} \quad (12)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(\tilde{y}_k - C_k\hat{x}_k^-) \quad (13)$$

$$P_k = (I - K_k C_k)P_k^- \quad (14)$$

The equations as above, K_k is the Kalman gain, P_k is the covariance of the measurement error estimation, P_k^- is the error covariance matrix, $\hat{x}_k$ is the estimation of the current state after the prediction and correction algorithm has been performed. The superscript - denote predicted estimates. Both $\hat{x}_k$ and P_k are collected and used in the predictor step of the next period [22]. The process covariance (Q) estimates the ability in observing the process. The measurement noise covariance (R) is used to compensate for the variance of the measurement noise. In this study, Q and R is defined as 5 and 0.001, respectively.

5. RESULTS AND DISCUSSION

5.1 Process modelling

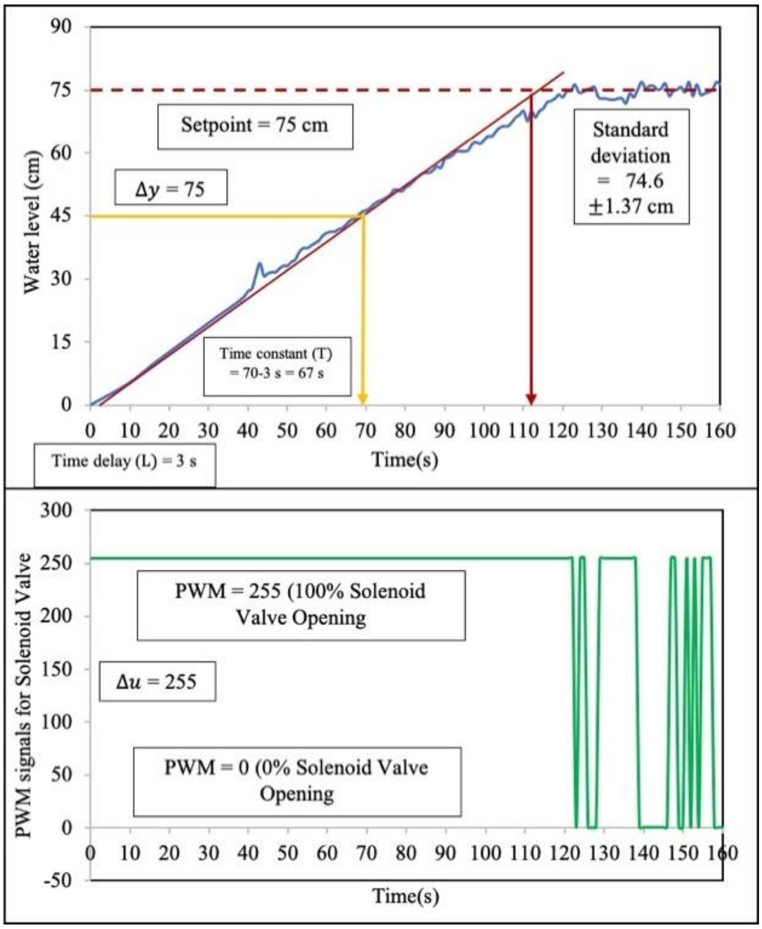

Figure 8 Process response for FOPDT model determination.

The transfer function of the process that indicates the process behavior is a FOPDT model which is expressed in equation (15). The lower graph is the behavior of PWM for the solenoid valve as shown in Figure.8. When the water level reaches the setpoint, solenoid valve will close immediately (PWM = 0) and when the water level is less than the setpoint, the solenoid valve will open with PWM = 255, and maintain the level of the water by switching the opening of the solenoid valve.

$$G_P(s) = \frac{0.29e^{-3s}}{67s+1} \tag{15}$$

The equation (15) showed that the process had a bit of time delay (L) = 3 s and time constant (T) = 67 s. These parameters were used in the tuning methods. The values of each tuning method are shown in Table. 3. Also, this transfer function will convert to the state space model in order to use in Kalman filter algorithm. Each parameter was used is A = -0.0149, B = 0.0043, C = 1 and D = 0.

Table 3 The values of parameters for the different tuning methods.

Methods	Type of controllers	Parameters				
		K_c	T_i	T_d	K_i	K_d
ZN	P	22.3	∞	0	0	0
	PI	20.1	10	0	2.01	0
	PID	26.8	6.00	1.50	4.47	40.2
CC	P	78.2	∞	0	0	0
	PI	69.6	9.10	0	7.62	0
	PID	104	7.25	1.10	14.3	114

5.2 P control performances

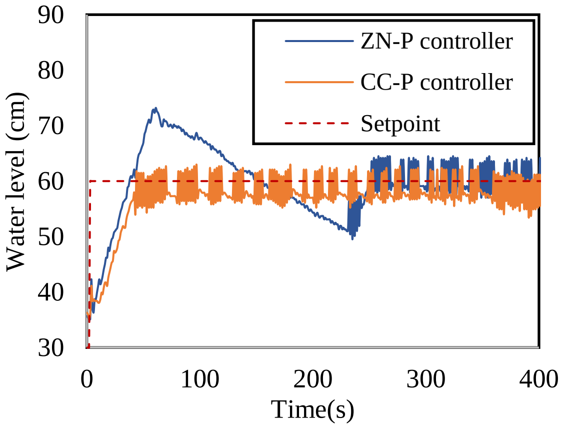

Figure 9 Control performances of P controller without Kalman filter

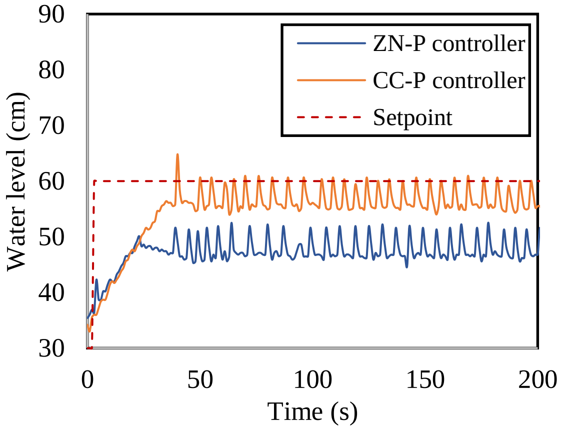

Figure 10 Control performances of P controller with Kalman filter

Figure 9 showed the response of the process to the P controller without Kalman filter. The controller is purposed by ZN and CC tuning methods. The results showed ZN-P controller without Kalman filter had an overshoot but CC-P controller without Kalman filter had no overshoot and seem like can track the setpoint, yet the result had high noise. Figure. 10 showed the results of ZN and CC tuning methods with Kalman filter, As seen the Kalman filter could reduce the noise from measurement but the CC-P controller with Kalman filter had steady state error but better than in case of ZN-P controller with Kalman filter and lower noise than CC-P controller without Kalman filter.

5.3 PI control performances

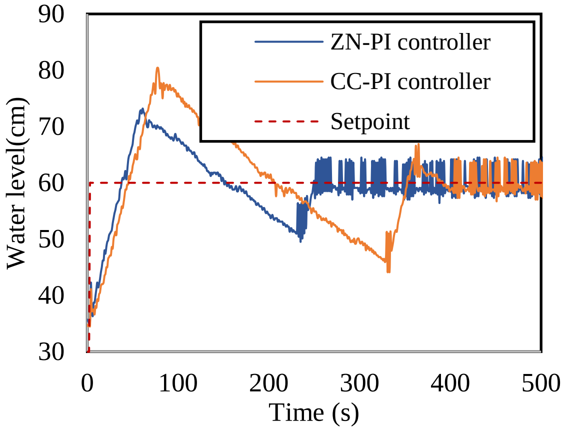

Figure 11 Control performances of PI controller without Kalman filter

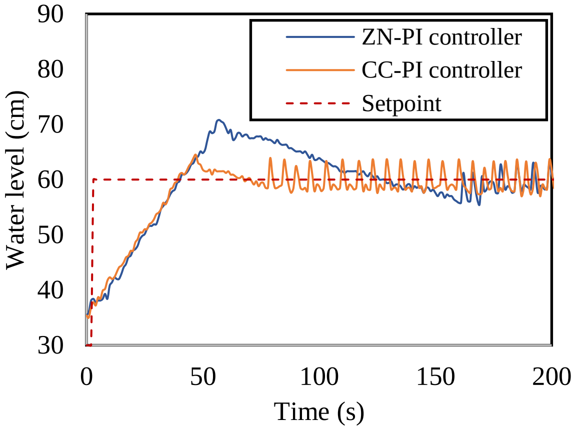

Figure 12 Control performances of PI controller with Kalman filter.

In this section, the results were from ZN and CC tuning method in PI controller type in both of without Kalman filter and with Kalman filter. The performance of ZN-PI controller without Kalman filter is shown in Figure 11. And the response of ZN-PI controller without Kalman filter had a 22% overshoot of the desired values and settling time at 269 s. The response of CC-PI controller without Kalman filter is shown in Figure 11. The response had a 34% overshoot of the desired value and settling time at 429 s.

Figure 12 showed the performances of the PI controller with Kalman filter. The response of the process of ZN-PI controller with Kalman filter is faster than ZN-PI controller without Kalman filter. The rise time of ZN-PI controller with Kalman filter is 40.3 s and the settling time is 170 s. And the process had less percentage overshoot than in the case of ZN-PI controller without Kalman filter. This ZN-PI controller with Kalman filter had an 18% overshoot of the desired value as shown in Figure 12. The CC-PI controller with Kalman filter had a performance better than ZN-PI controller. The CC-PI controller with Kalman filter had only a 7.50% overshoot. Hence, The CC-PI controller with Kalman filter is better than other PI controllers.

5.4 PID control performances

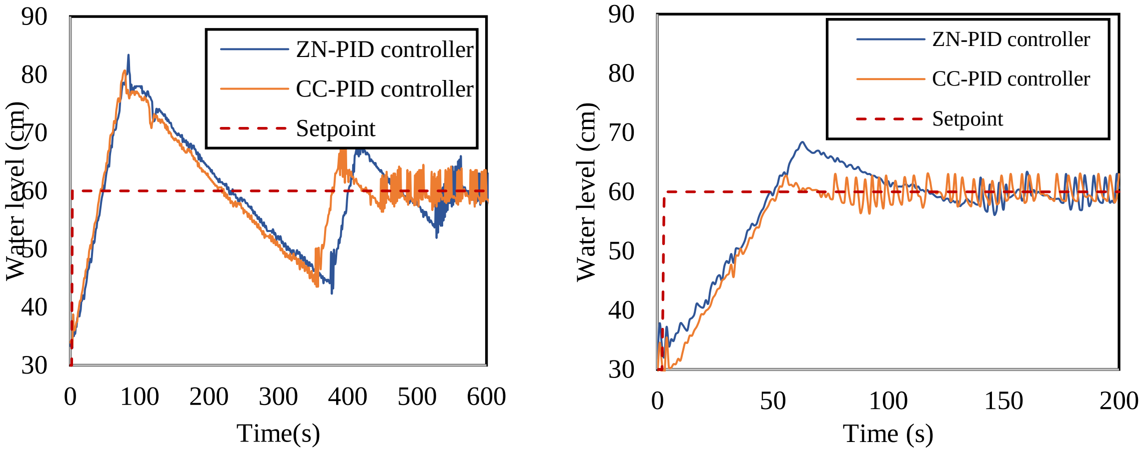

Figure 13 Control performances of PID controller without Kalman filter.

Figure13 and Figure14 showed the control performance of PID based on different tuning methods. It can be seen that the process responds to both controllers. The response of the process to CC-PID controller without Kalman filter. The rise time is 43.5 s, and the settling time is 453 s. and ZN-PID controller without Kalman filter had the rise time at 47.3 s and the settling time at 564. Hence, CC-PID controller without Kalman filter is better than ZN-PID controller in terms of the rise time and settling time.

The control performances for the ZN-PID and CC-PID controller with Kalman filter are shown in Figure 14. In the case of ZN-PID controller with Kalman filter, the rise time was 49.0 seconds and the settling time was 147 seconds. In the case of CC-PID, the rise time was 52.3 seconds and the settling time was 47 seconds. Although the rise time of the ZN tuning method was a bit lower than the CC tuning method, Besides the overshoot percentage in the case of ZN-PID controller with Kalman filter (14%) was higher than the CC-PID controller. While the CC-PID controller had no overshoot of the desired values. As seen in the experimental results, the CC tuning method could bring the present output to the desired value and had a better performance than the ZN-PID controller. Although those two tuning methods are widely used in chemical industries, but precise process transfer function or personal experiences of the engineer are required to gain acceptable control performances [23].

The comparison of various parameters of different methods is shown in Table 4. The results showed P controller is not applicable to this system due to the responses had the offset and high noises. CC-P controller without Kalman filter cannot considered settling time because the responses were not stay within a range of 5% error band. ZN-PI controller and CC-PI controller with Kalman had the results better than other controllers in terms of settling time, maximum overshoot, IAE, and standard deviation.

Table 4 Comparison of various parameters for different methods.

Experiments	Tuning methods	Type of controllers	Parameters					
			Rise time (s)	Settling time (s)	Maximum overshoot (%)	IAE	Standard deviation	Coefficient of variation (%)
Process without Kalman filter	Ziegler-Nichols	P	N/A	N/A	N/A	18000	N/A	N/A
		PI	37.8	269	22.0	5200	2.57	2.28
		PID	47.3	564	39.0	7100	2.28	3.81
	Cohen-Coon	P	45.8	N/A	0	5500	N/A	N/A
		PI	45.2	429	34.0	6300	2.24	4.19
		PID	43.5	453	34.5	6300	2.24	3.74
Process with Kalman filter	Ziegler-Nichols	P	N/A	N/A	N/A	2600	N/A	N/A
		PI	40.3	170	18.0	1100	1.71	2.91
		PID	49.0	147	14.0	1200	2.95	5.16
	Cohen-Coon	P	N/A	N/A	N/A	1100	N/A	N/A
		PI	39.3	48.0	7.50	780	1.92	3.34
		PID	52.3	47.0	0	1100	1.71	3.08

Table 5 Comparison of Ziegler-Nichols and Cohen-Coon tuning methods with Kalman filter.

Time-domains	Ziegler-Nichols- PI controller	Cohen-Coon -PI controller
Rise time (s)	40.3	39.3
Settling time (s)	170	48.0
Maximum overshoot (%)	18.0	7.50
IAE	1100	780
Standard deviation	1.71	1.92
Coefficient of variation (%)	2.91	3.34

The subsequence discussion is to compare the performances of ZN-PI controller and CC-PI controller with Kalman filter as shown in Table.5. It can be seen that CC-PI controller with Kalman filter had a smaller rise time compare with ZN-PI controller with Kalman filter. Also, the settling time of CC-PI controller with Kalman filter was smaller than ZN-PI controller with Kalman filter.

5.5 Setpoint tracking

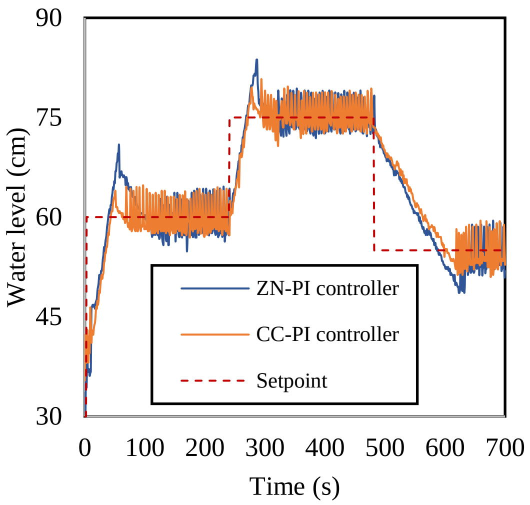

Figure 15 Multiple setpoint tracking responses for several tuning methods with Kalman filter.

Figure 15 showed the performances of the setpoint tracking. The PI controller with Kalman filter was tested for the different water levels in the column. The results were obtained by using the ZN tuning method, and CC tuning method with Kalman filter in PI controller type. The process behavior showed the controllers tracked the setpoint well. Although the performances of the system had an overshoot when the setpoint is increased. Also, when the setpoint has decreased the controller was tracked to the setpoint yet their settling time was longer than the increasing setpoint because the outlet flow is smaller than the inlet flow, so it took time to reach the setpoint.

6. CONCLUSIONS

From the proposed performances. The controllers with Kalman filter present satisfactory results. The process value can track the desired values well. P controller had offset which means the P controller does not suit this study. But the process with PI and PID controllers is acceptable. If the processes cannot accept the overshoot, the CC-PID controller with Kalman filter can be used. If the processes can accept the overshoots, the ZN-PI, ZN-PID, and CC-PI controller with Kalman filter can be used, depend on the proper process. Moreover, the Kalman filter can provide a more accurate and precise estimation of the unobserved variables in presence of uncertainty. Also, the Kalman filter is useful in a practical way.

7. ACKNOWLEDGMENT

I would like to thank the technicians “Mr.Somkid Geenapong” and “Mr.Narong Apayanugool” for providing me with all the facilities and fixing my equipment. I express my sincere thanks to department of chemical engineering, Faculty of Engineering, Prince of Songkla University for providing me the area for my laboratory. I also like to acknowledge with gratitude to Graduate school, Prince of Songkla University for providing me the scholarship for this study and Graduate school of engineering, Prince of Songkla University for the scholarship for 1-year tuition. Finally, my family is an important inspiration for me. I express my gratitude to them.

REFERENCES

[1] Anarase, B.V. and Parvat, B.J. Design of Model Predictive Controller for Level Process. *IOSR Journal of Computer Engineering*, 2016; 10-16.

[2] Abirami, S., Hussain, Z., Saravana Muthu, A.S. and Aravind Kumar, C.B. Performance Comparison of Different Controllers for a Level Process. *International Journal of Engineering Research and Applications*, 2014; 4(3): 341.

[3] Dinakaran, C. Temperature and Water Level Control in Boiler by using Fuzzy Logic Controller. *International Journal of Electrical Power System and Technology*, 2015; 1(1), 27-37.

[4] Chew, I.M., Wong, F., Bono, A. and Wong, K.I. Simulation of Process Identification and Controller Tuning for Flow Control System. 29th Symposium of Malaysian Chemical Engineers (SOMChE) 2016, December 1-3, Miri, Malaysia, 2016.

[5] Telepatil, A.R., Hambar, A.N. and Terwadkar, P.S. Home Automation with MATLAB and ARDUINO Interface. *International Journal of Innovative Research in Computer and Communication Engineering,* 2017; 5(3): 5501-5507.

[6] Anarase, B.V., Parvat, B.J. and Kadu, C.B. Design of Model Predictive Control for Non Linear Process. *International Journal of Emerging Technology and Advanced Engineering,* 2016; 6(1): 304.

[7] Osman U., Ishak E. and Nihat A. Embedded System Based Real Time Position Control of a Dc Motor Using Matlab. *International Journal of Research in Engineering & Advanced Technology*, 2016; 4(5): 37-38.

[8] Babu, A.R., Kibreab, S., Mehari, S. Experimental Studies on Step Response of Water Level Control System with P, PI and PID Control Mechanisms. *International Research Journal of Engineering and Technology (IRJET)*, 2020; 7(10): 1504-1509.

[9] Jaafar, H.I., Hussien, S.Y.S., Selamat, N.A. and Rashid, M.Z.A. Development of PID Controller for Controlling Desired Level of Coupled Tank System. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2014; 3(9): 32-36.

[10] Kapale, S. and Bharakad, S. Liquid Level Estimation in Dynamic Condition using Kalman Filter. *International Journal of Engineering and Application*, 2016; 6(8): 12-16.

[11] Mehmet, Y. and Özgür, V. Liquid Level Control with Different Control Methods Based on Matlab/Simulink and Arduino for the Control Systems Lesson. *International Advanced Researches and Engineering Journal*, 2020; 4(3): 249-254.

[12] Walz, S., Lazar, R., Buticchi, G. and Liserre, M. Dahlin-Based Fast and Robust Current Control of a PMSM in Case of Low Carrier Ratio. *Institute of Electrical and Electronics Engineers,* 2019; 7: 102199-102208.

[13] How Relays Work. GALCO. [Online] Available: https://www.galco.com

[14] External Power Supplies. ENERY STAR, 2010. [Online] Available: https://www.energystar.gov/index.cfm?c=archives.power_supplies.

[15] Swain, S., Mandavi, D.S. and Behera, A.A. Comparison of Performance Analysis of Different Control Structures. Department of Electronics & Communication Engineering National Institute of Technology, Rourkela, 2012.

[16] PID Controller- Working and Tuning Methods. Electronic Hubs, 2015. [Online] Available: https://www.electronicshub. org.

[17] Barber, R., Horra, M. and Crespo, J. Control Practices using Simulink with Arduino as Low Cost Hardware. The 10th IFAC Symposium Advances in Control Education. August 28-30, Sheffield, UK, 2013**.**

[18] Korsane D.T., Yadav V. and Raut K.H. PID Tuning Rules for First Order plus Time Delay System. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering*, 2014; 2(1): 582-586.

[19] Kumar, A. and Garg K.K. Comparison of Ziegler-Nichols, Cohen-Coon and Fuzzy Logic Controllers for Heat Exchanger Model. *International Journal of Science, Engineering and Technology Research*, 2015; 4(6): 1918.

[20] Wakitani, S., Nakanishi, H., Ashida, Y. and Yamamoto, H. Study on a Kalman Filter based PID Controller. The 3rd IFAC Conference on Advances in Proportional- Integral-Derivative Control, May 9-11, Ghent, Belgium, 2018.

[21] Jovanovic, B. Kalman Filter Estimation of the Unrecorded Economy in Macedonia. *National Bank of the Republic of Macedonia*, 2015.

[22] Goddard, P. An Introduction to the Kalman Filter. 2003-2020. [Online] Available https://www.goddardconsulting.ca/kalman-filter.html./comp/prod/relay.htm

[23] Joseph, E. and Olaiya, O.O. Cohen-Coon PID Tuning Method: A Better Option to Ziegler Nichols-Pid Tuning Method. *Computer Engineering and Intelligent Systems*, 2018; 9(5): 33-37.