Zewail City of science and technology
Applied Digital Control and Drives NANENG 512
FALL 2020

Coupled tank system:

Water tank level controller. Project phase 1.

Rami Wail - Ahmed Alaa - Mohamed Elshafey
IDs: 201600112 - 201600352 - 201700907

DATE
Dec 17th, 2020

I. INTRODUCTION:

Our Water tank level controller consists of a dynamic model of coupled tank system [1]. Recently, the most common control problem for the process of an industrial site is to control various variables such as the temperature, water level, and internal pressure of a tank and a chemical reactor [2]. Ultimately, controlling the liquid level of a tank is the primary element for oil and chemical liquid processes. The purpose of level control is to make the water level follow a given set value or to quickly restore a constant water level in case of disturbance [2]. Generally, a PID controller is widely used for level control; and the gains of the controller should be tuned so that the required performance of a controlled system can be satisfied. This tuned PID controller can satisfy the required performance of the controlled system when the operating environment of the controlled system is unchanged. Unfortunately, this is mostly not the case as parameters of the controlled system change due to the change in the operating environment of the controlled system. Thus, satisfactory control cannot be guaranteed unless the gains of the controller are artificially changed again. Accordingly, this paper aims to provide an extensive overview of the control system and its parameters while considering the software (MATLAB) and hardware (components) perspectives, alternatively.

II. Level Control System:

The schematic diagram of the coupled tank is shown in Figure 1. Two tanks are connected in an interactive valve. The inflow of tank 1 is qi and outflow of tank 2 is qo2. The control variable is level in tank 2. According to Figure 1 the input u is the input voltage which is taken to the pump, and the output h2 is the water level in tank 2 [1].

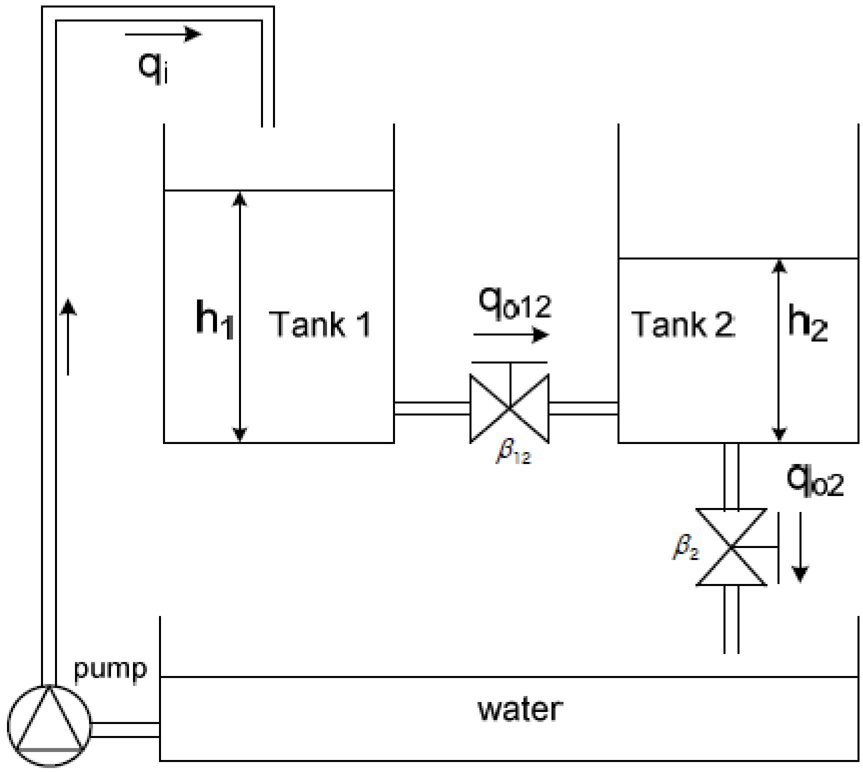


Figure 1: The coupled tank system.

The nonlinear equation can be obtained by mass equivalent equation and Bernury's law is given by.

$$\frac{dh_1(t)}{dt} = -\frac{\beta_{12}a_{12}}{A_1}\sqrt{2g\big(h_1(t) - h_2(t)\big)} + \frac{k}{A_1}u(t) \tag{1a}$$

$$\frac{dh_2(t)}{dt} = -\frac{\beta_2 a_2}{A_2}\sqrt{2gh_2(t)} + \frac{\beta_{12}a_{12}}{A_2}\sqrt{2g(h_1(t) - h_2(t))} \tag{1b}$$

where A1 and A2 are the cross section area (cm2) of tank 1 and tank 2, a2 is the cross section area (cm2) of outlet of tank 2, a12 is the cross section area (cm2) of jointed pipe between tank 1 and tank 2, β2 is the value ratio at the outlet of tank 2, β12 is the value ratio between tank 1 and tank 2, g is the gravity (cm/s2) and k is the gain of pump (cm3/V × s).
According to Equation (1), a linearized model is given by Equation (2).

$$G(s) = \frac{H_2(s)}{U(s)} = \frac{K}{T_{12}T_2 s^2 + (T_{12} + 2T_2)s + 1} \tag{2}$$

where T12 is the time constant between tank 1 and tank 2, and T2 is the time constant of tank 2. T12, T2 and K can be obtained via the Equation (3) as follows.

$$T_{12} = \frac{A_1}{\beta_{12} a_{12}} \sqrt{\frac{2(\bar{h}_1 - \bar{h}_2)}{g}}, \quad T_2 = \frac{A_2}{\beta_2 a_2} \sqrt{\frac{2\bar{h}_2}{g}}, \quad K = \frac{kT_2}{A_2} \tag{3}$$

Where h_1^{-} and h_2^{-} is the water level at the operating point of the coupled tank system.
For the coupled tank parameters are obtained by experimental results or data sheet and the parameters including the operating point of the process as shown in Table 1 [1].

Table 1: Parameters of coupled tank system

A_1, A_2	a_2, a_{12}	β_2	β_{12}	$\bar{h}_1$	$\bar{h}_2$	k
154	0.5	0.682	1.531	30	25	30

Using Table 1 and Equation (2), the transfer function in operating points is shown below and this model is simulated in MATLAB.

$$G(s) = \frac{19.8617}{2070.8939\, s^2 + 224.2254\, s + 1}$$

Similarly, hand analysis was performed to validate the form of the final transfer function. The steps were generally the ones taken in the mechanical modeling tutorials. Hand analysis is provided in the following section.

III. Hand analysis:

tank 1

$$q_{o12} = \frac{h_1 - h_2}{R_1}$$

$$(q_i - q_{o12})\,dt = A_1\,dh_1$$

$$\left(q_i - \left(\frac{h_1 - h_2}{R_1}\right)\right)dt = A_1\,dh_1$$

$$(R_1 q_i - (h_1 - h_2))\,dt = R_1 \frac{dh_1}{dt}$$

$$R_1 Q_i(s) = R_1 A_1 s H_1(s) + H_1(s) - H_2(s)$$

tank 2 $$q_{o2} = \frac{h_2}{R_2}$$

$$(q_{o12} - q_{o2})\,dt = A_2\,dh_2$$

$$\left(q_{o12} - \frac{h_2}{R_2}\right)dt = A_2 \frac{dh_2}{dt}$$

$$[R_2 A_2 s + 1]\,H_2(s) = R_2 Q_{o12}(s)$$

$$\frac{H_2(s)}{Q_{o12}(s)} = \frac{R_2}{R_2 A_2 s + 1}$$

K.M.S

$$\frac{Q_{o2}(s)}{Q_i(s)} = \frac{1}{(R_1 A_1 R_2 A_2)s^2 + (R_1 A_1 + R_2 C_2 + R_2 A_1)s + 1}$$

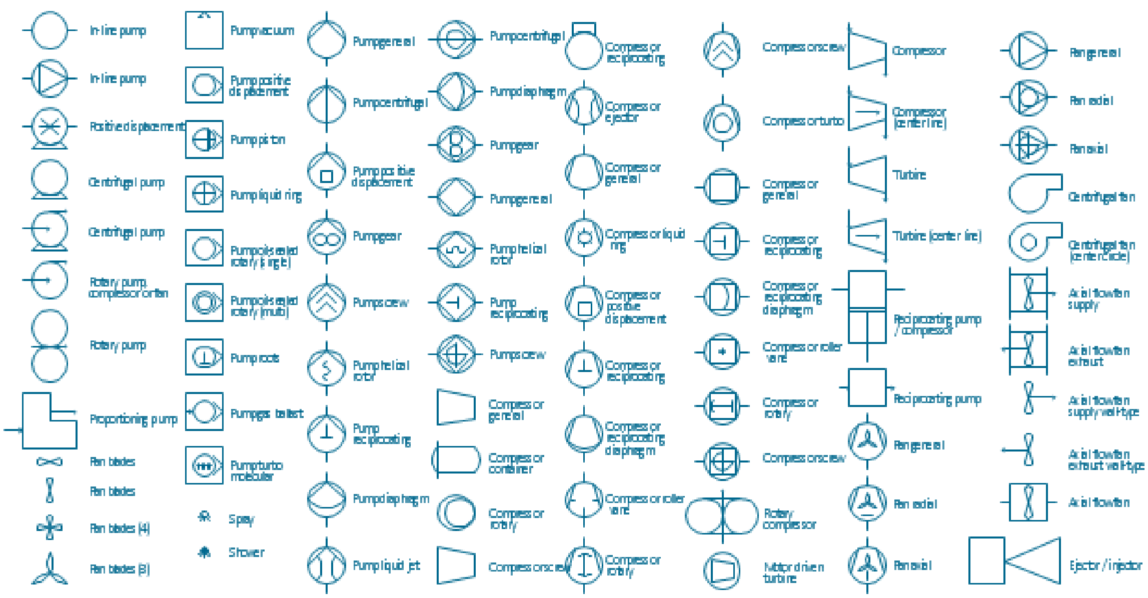
$$\frac{H_2(s)}{Q_{o2}(s)} = \frac{R_2}{(R_1 A_1 R_2 A_2)s^2 + (R_1 A_1 + R_2 A_2 + R_2 A_1)s + 1}$$

IV. Hardware components:

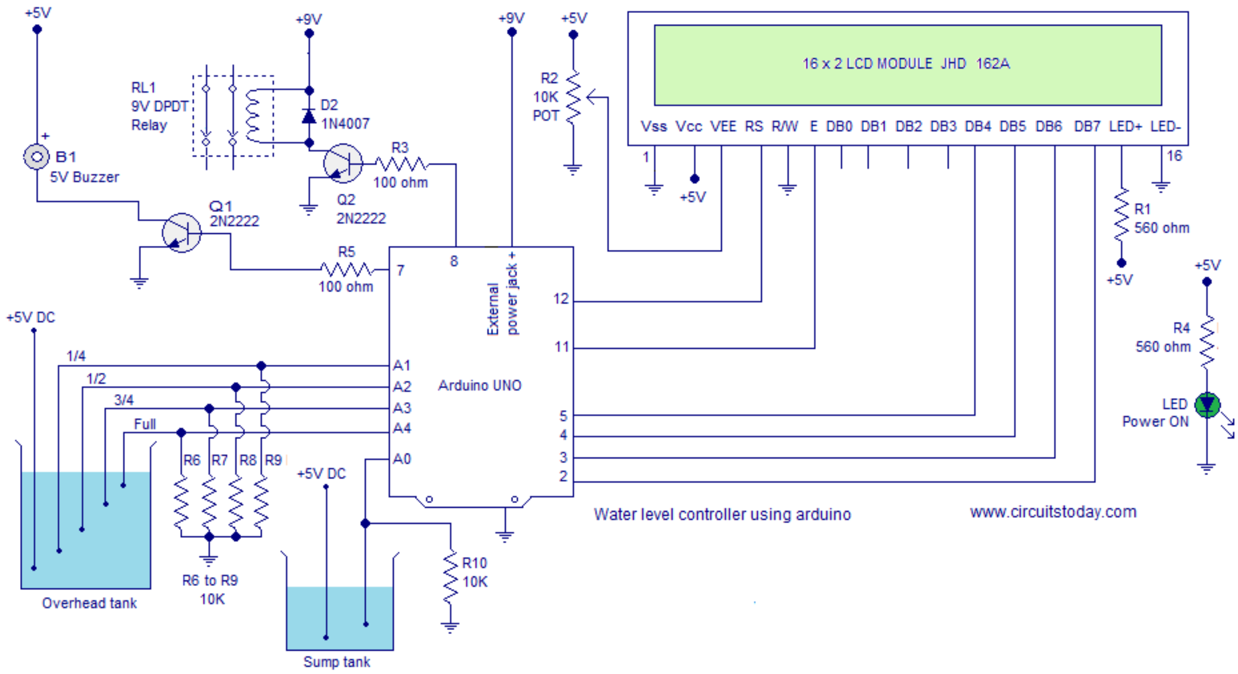
The hardware implementation of the system consists of the following basic components:

- pump “Motor”
- tanks “containers”
- sensor “ultrasonic most probably”
- water tap
- tubes
- hose
- water valve
- PID-Controller
- bread board ,wires, LEDs
- Arduino module for water level control

Various pump schematics are shown below:



Based on availability of these components on campus or in the market, the physical model will vary slightly but this should not affect the system itself. Also, an overview schematic of the arduino system is shown below [3]:



V. Software components (MATLAB):

In order to simulate the level controller and calculate its parameters and characteristics, the following code was implemented in MATLAB:

```
clear all; clc; close all;
s=tf('s');
t=0:0.01:5;
%define function
%open loop
G=(19.8617)/(2070.8939*s^2+224.2254*s+1)
%closed loop
T=feedback(G,1);
%obtain damping
damp(T)
%obtain overshoot, undershoot, tr, ts, peak, etc
stepinfo(T)
%obtain ess
%e=1/s*s*1/1+openloopgain
%ess=lim(e)as s->zero
%knowing that the open loop gain is G
E=1/(1+G);
ess=dcgain(E);
ess
%plot step response
step(T)

figure %initialize figure
t = 0:0.01:5; %small time for plot
plot(step(T, t)); %step
hold on
title('Step Response base')
legend('Step','Ramp','Parabolic')
xlabel('Time')
ylabel('Amplitude')
```

In this code, we simply initialize the transfer function 'tf' in s domain, followed by identifying the form of this TF as an open loop. Then employed feedback() built-in function to obtain the closed loop. The TF characteristics are extracted in addition to the steady state error with a unit step input. Built-in function damp(T) calculates the poles, damping, frequency, and time constants. Finally, the step response is plotted.

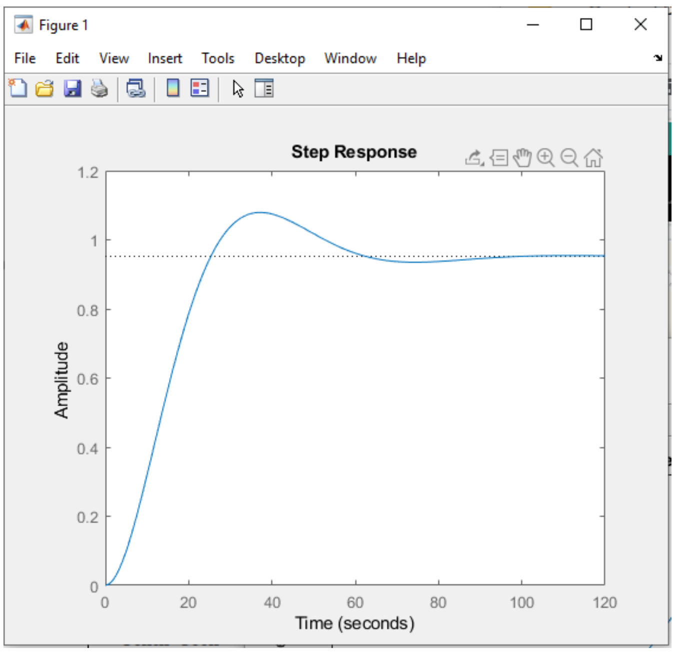
```
G =

          19.86
  ----------------------
  2071 s^2 + 224.2 s + 1

Continuous-time transfer function.
```

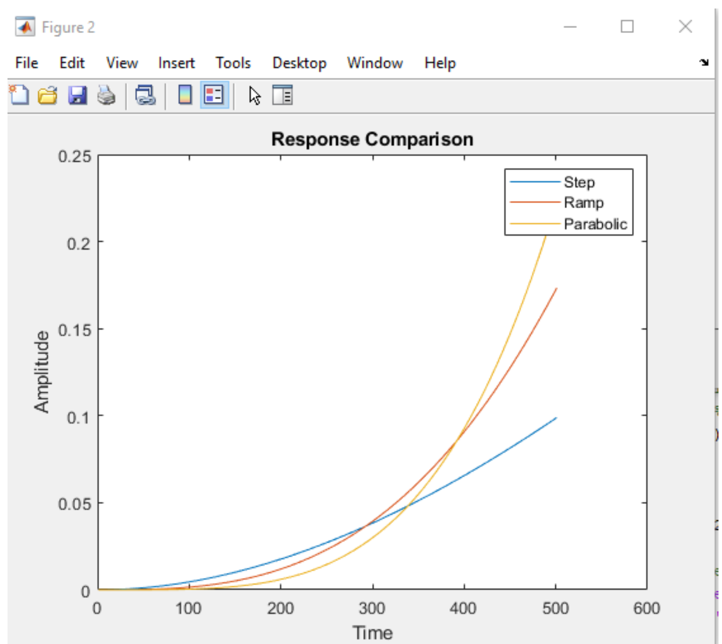
```
       Pole                 Damping       Frequency       Time Constant
                                        (rad/seconds)       (seconds)

-5.41e-02 + 8.45e-02i       5.39e-01       1.00e-01          1.85e+01
-5.41e-02 - 8.45e-02i       5.39e-01       1.00e-01          1.85e+01
```

```
        RiseTime: 17.1210
    SettlingTime: 57.8193
     SettlingMin: 0.8606
     SettlingMax: 1.0793
       Overshoot: 13.3627
      Undershoot: 0
            Peak: 1.0793
        PeakTime: 37.4284
```

```
ess =

    0.0479
```

VI. Simulation Results and comparison:





The simulation results we have obtained are similar to the ones found in [1] where we got settling time of 57.8 while they got 40 using the PSO method and even more (in the range from 75 to 110) using other methods. Rise time on the other hand was found to be 3.8 using PSO while we got 17 similar to the other models in [1]. We may use the PSO technique to gain better results and performance but we know that any way results would face some type of error and lack of performance when done in real time due to the inaccuracy of the equipment we will use.

In the coming phase, we hope to identify the effects of hardware and software (PID tuning in MATLAB) on the performance characteristics of the model. The values and plots obtained from Lee et al [1] are shown below.

Table 3: Comparison of time domain specifications for set-point tracking

Transient Characteristic	Z-N	C-C	IMC	PSO
Rise time[s]	11.07	11.33	22.22	3.80
Peak amplitude	32.43	32.13	30.25	31.96
Peak time[s]	29.02	29.42	46.15	9.35
Settling time[s]	111.19	101.58	75.58	40.01

Table 2: Comparison of PID controller parameters

Tuning rules	PID parameters		
	K_p	K_i	K_d
Zeigler-Nichols	1.4928	0.0855	6.5159
Cohen–Coon	1.6712	0.0791	5.2666
IMC	1.0153	0.0046	4.3440
PSO	12.144	0.030	9.349

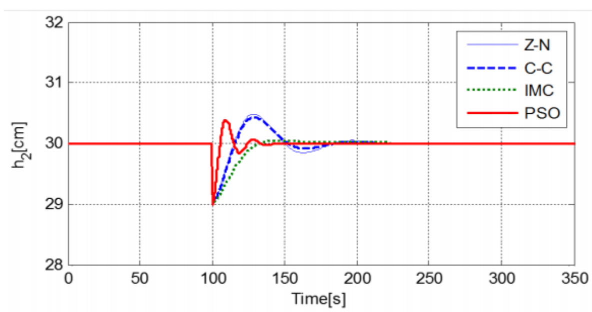


Figure 6: Closed-loop responses for disturbance rejection with PID controllers

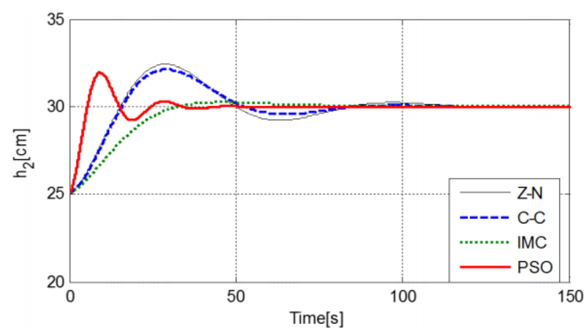


Figure 5: Closed-loop responses for set-point tracking with PID controllers

Also the schematic of the PSO based PID control system is shown for convenience from [1].

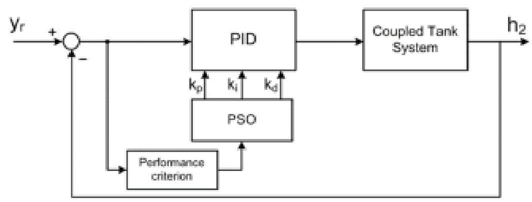


Figure 2: Structure of PSO based PID control system

We hope to further improve and compare on these values in the PID phase in MATLAB.

VII. References:

1. Lee, Yun-Hyung & Ryu, Ki-Tak & Hur, Jae-Jung & So, Myung-Ok. (2014). PSO based tuning of PID controller for coupled tank system. Journal of the Korean Society of Marine Engineering. 38. 1297-1302. 10.5916/jkosme.2014.38.10.1297.
2. Lee, Yun-Hyung & Jin, Gang-Gyoo & So, Myung-Ok. (2014). Level control of single water tank systems using Fuzzy-PID technique. Journal of the Korean Society of Marine Engineering. 38. 550-556. 10.5916/jkosme.2014.38.5.550.
3. Water level controller using arduino. Water level indicator using arduino

4. Jiffy Anna John, Dr. N. E. Jaffar and Prof. Riya Mary Francis, 2015. Modelling and Control of Coupled Tank Liquid Level System using Backstepping Method. International Journal of Engineering Research and, V4(06).