Int. j. adv. multidisc. res. stud. 2024; 4(6):398-406

Received: 02-10-2024
Accepted: 12-11-2024

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

Implementation of Embedded PID Controller to Water Level Control System

[1] Aboajela Kajaman, [2] Zeiad Saleh, [3] Khalifa Sahal

[1, 3] Department of Control Engineering, College of Electronic Technology, Bani walid, Libya

[2] Control System Engineering, Raslanuf Oil and Gas Processing Company (RASCO), Libya

DOI: https://doi.org/10.62225/2583049X.2024.4.6.3452 Corresponding Author: **Aboajela Kajaman**

Abstract

In chemical and industrial processes, the level control of liquids in tanks is significant and widely used, especially in power stations and petroleum industries. The liquid level can be controlled by the most common method, PID control due to its efficiency and performance in achieving the desired set point. In this paper, the water level in tank control using a level sensor and PID controller system will be designed and implemented based on the PID Arduino microcontroller algorithm that controls the change in flow rate by controlling a DC water pump with the help of a MOSFETs interface circuit. Tuning the specific PID level tank controller will be implemented to increase the system accuracy with no error and overshoot when there is a disturbance (drawn or excess water).

Keywords: PID, Level Control, Tank, MOSFETs

1. Introduction

Control liquid levels is an essential process in the industry nowadays, which is not vital only for control purposes as it is controlling the level of liquids by designing automatic electronic control systems. On the other hand, in industrial processes such as the treatment of water in desalination plants, the level of water in a container like a tank needs to be kept at a certain set of the desired point. The level of the water needs to be maintained at the desired set point for the proper functioning of the process and to achieve the desired target or product [1].

The controller plays a significant role in maintaining the accuracy level by implementing the system in a feedback control system. PID controllers can be used in most practical control systems ranging from consumer electronics such as cameras to industrial chemical processes [2].

Laith in 2015 used a proportional–integral–derivative (PID) controller to remove the steady state error by the PID controllers which are designed to get the desired output for many processes such as velocity in moving objects, the temperature in a particular environment, position at a certain location and liquid level in a container where the aim is to achieve a certain set point in a short possible time with minimal overshoot and with small steady state error. The design of the PID controllers is based on determining the specification needs like overshoots, rising time, settling time and steady state error by tuning the PID parameters to meet the objectives of control [3].

In 2015, Abdul Latheef and others implemented virtual instrumentation controllers for cylindrical tank systems. This Virtual Instrumentation PID controller implementation is possible on software called LabVIEW developed by National Instruments [4].

DC motors are used extensively in industries due to their variable speed that suits the process to be used for different applications, their most demanding speed-torque characteristics, and their reliability and simplicity in controlling aspects. Therefore, this work assumes a DC motor water pump filling a certain water container or tank and the motor speed determines the water flow rate to the tank. In comparison to AC drives, DC motor drives are simple and less expensive [5].

Concerning the interface, the PID controller could be implemented as an algorithm in the Arduino chip. In this situation, the interface circuit should be designed to control the pump via Arduino which is based on MOSFET. This circuit is an electronic voltage-current-controlled device, used to control the flow of current into a DC motor pump depending on the value of PWM that comes from Arduino. The real high of liquids in a tank is provided by the float sensor [6].

The PID control algorithm theory and feedback modeling system are applied to design the overall system. As a result of maintaining the required water set level, the assists in saving and conserving water by monitoring the level and eliminating overflow in the storage system.

2. Overview

Many researchers design, implement, and fabricate PID controllers and the level control process. In [3], the aims are to use an economical technology to build the liquid tank level controls the liquid level using Arduino UNO microcontroller as data acquisition with LabVIEW software interfacing with an ultrasonic sensor. This system with ultrasonic can only work with limited set points, or in other words there would be unlimited possibilities in the program.

In 2014, Theopaga and others used a PID algorithm with Arduino to control the temperature and humidity of a baby incubator. The temperature is measured and displayed using a programmed Arduino and controlled using Pulse Width Modulation (PWM) through the PID loop. The author emphasizes the use of the PID controller over the simple on/off controller for the following two reasons: Fast response, which is important for controlling a sensitive system, and for lower power consumption. The on/off controller consumes higher power as it has to regularly feed the power supply to the heater and fan to switch the system on and off [7].

The control of DC motors by using an Arduino-based PID controller was discussed by Vikhe and others. The Arduino is interfaced with LabVIEW. Although, DC motors are very important tools for industrial, scientific research, and experimental applications, the control of the motors to a precise accuracy is important for position control of sensors and in many other scientific applications. The authors are using a very low-cost technique to control the speed of the DC motors. An Arduino is used as a very low-cost data acquisition device. It is interfaced with a Graphic User Interface (GUI) of LabVIEW called LabVIEW Interface for Arduino (LIFA) so that the users can set their target speed for the motor on the screen of LabVIEW. The comparison between the open loop control and closed loop feedback control (PID control) performance is also made. The tachometer was used as a sensor to measure the speed of DC motors [8].

3. Problem Statement

The traditional water tank has many disadvantages such as the water level must draw the water manually to the tank when there is no water in the tank, but the problem of manual control is sometimes people forget to turn off or turn on the valve. Moreover, the automatic water level control is to avoid the overfilling of the open container in the industrial processes, however, if the float sensor is damaged, all the systems cannot function properly. Even though the PID controller is widely used in industrial processes, the tuning of the PID parameter is a crucial issue, particularly for the system's characteristic which has a large time delay and high order system.

For the model to succeed and to be implemented, the following objectives have to be achieved:

- Implementing the required tank model with the proper equipment (actuators and sensors).
- Determining the specific PID level tank controller to increase the system accuracy.
- Achieving low error and no overshoot specification when there will be a disturbance (drawn or excess water).

4. System structure

The whole project system is divided into two main stages which are the hardware and software. The hardware consists of (Tanks, DC pumps, Relays, MOSFETs, solenoid valve sensors, and a controller board). Each part is tested and implemented separately, and then integrated to construct the system. The software part was done by using C language with an ARDUINO interface board. In this chapter, input devices, output devices, and mechanical structures will be discussed and introduced.

The combination of the Arduino Board and Interface circuit are both formating the system's controller. Pumps, relays, and solenoid valves are included within the system actuators. This model has been implemented with two pumps, one for fill-up and the other for drain, relays, and one solenoid valve. Any physical variable that could be controlled via a designed control system is known as the process of the system, so controlling the level in tanks is the process of the proposed system. For the system to have less error in Fig 1, it should be closed loop by feedback with the sensor used for a process such as a resistive level float sensor.

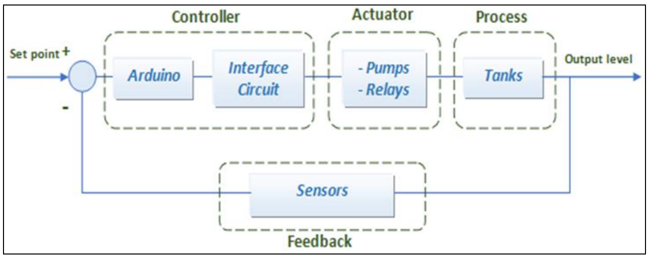


Fig 1: The block diagram of the system

Fig 2 shows the complete system architecture diagram and its controller board. The ARDUINO controller was used as the main brain of the whole system; it receives the data from the input devices and updates the output devices. The target point for the system is to keep the output level identical to the set point; so that the level float sensor is used as feedback to the controller on the tank level at any moment. Furthermore, the DC pump is used to correct the wrong values of levels.

The input devices are used to start the operation and gather information about the system and the outdoor environment, They consist of a float level sensor, to feed the controller (Microcontroller) information about the surrounding environment to make the decision**.** The ARDUINO board is a small programmable integrated circuit that contains a processor, memory, and programmable input and output pins. The controller is the brain of the system. The controller stores and sends programmed tasks to the actuators and other parts of the main system. Output devices are the actuators that convert an electrical signal into mechanical movement; the types of actuators are two DC motor pumps.

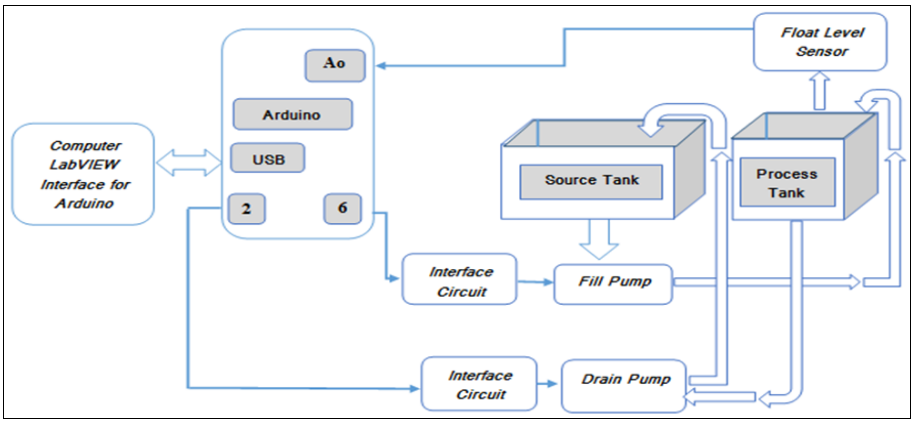


Fig 2: System architecture diagram

5. System Components

5.1 Tanks design

The design of the tanks is to choose the appropriate dimension for the mode. The dimension of the process tank is restricted by the mechanical movement distance of the float sensor, especially with height where it should not be less than the maximum high position in the float level sensor. Fig 3 shows the required dimension needed for the model and the source tank, however, it does not depend on any factor, the only consideration for source tank design is just to provide the required water quantity for the model.

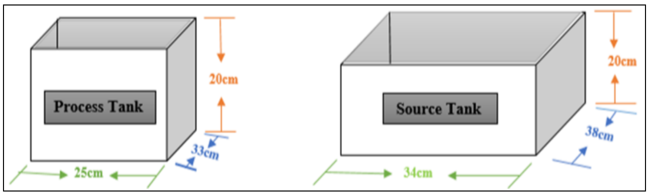


Fig 3: Process and source tank dimension

5.2 Resistive float level sensor

It is the sensor that is used in this project and will be discussed briefly. Fig 4 shows a float-level sensor based on the resistive concept. When the level rises the resistance of the element decreases, when the level decreases the resistance increases. Fig 5 shows the internal construction of the resistance level float sensor.

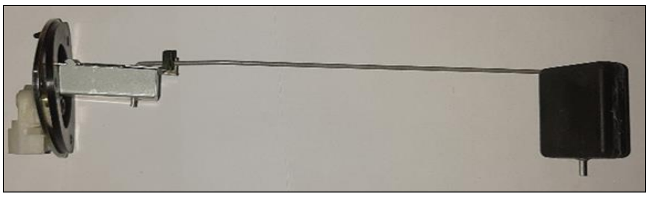


Fig 4: Resistive float level sensor

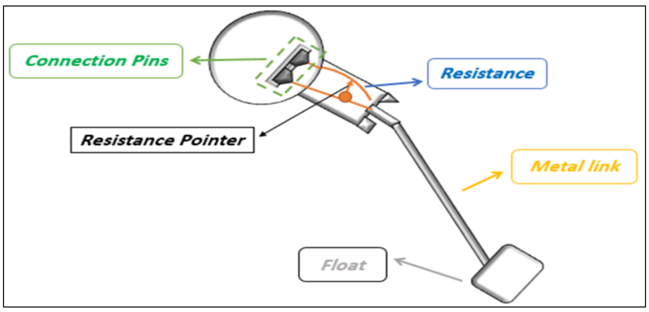


Fig 5: Float level sensor internal diagram

This type of level sensor has the required linearity for the process. The level process is a kind of process that needs repeatability measurements, so this sensor contains the required repeatability and it needs a modifying circuit to construct with Arduino.

Table 1: Specification of float sensor

Length	20cm
Weigh	25 g
Max. Resistance	(130-137) Ω
Min. Resistance	(6-10) Ω

5.3 Brushless Water Pump

This Brushless Water Pump's name is shaft pump DC 12V. Mini dc 12v 3m micro quiet motor submersible or outside working brushless water pump is used in many fields, it is used in gardening, water-air cooler, and aquarium cleaning the drain pump. It is a 12V dc pump and works on solenoid valves. It is a low-noise device, reliable, and has high efficiency. Its maximum flow is 10 L/M, and operating voltages are DC 12V.

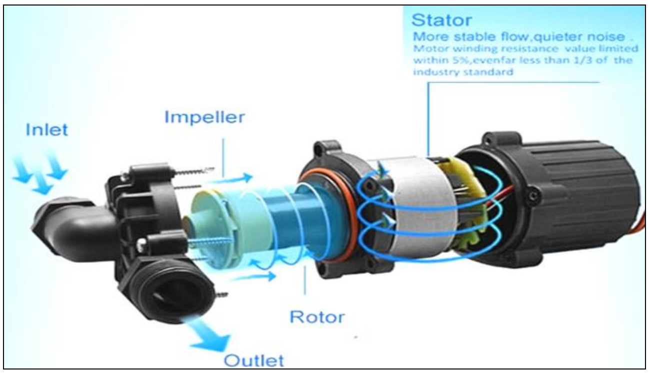


Fig 6: The Pump Water Brushless DC 12V

5.4 Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 microprocessor. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

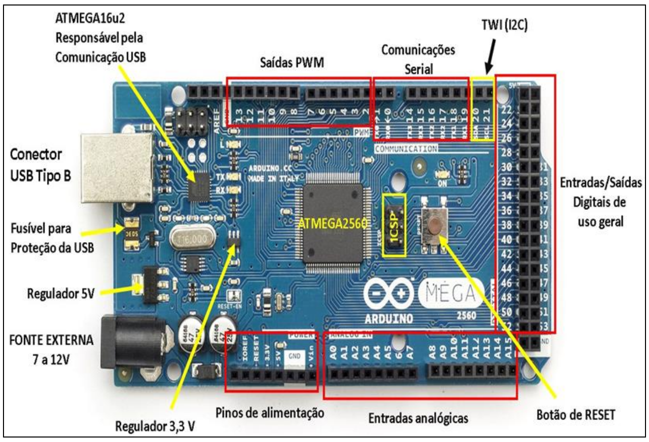


Fig 7: Arduino Mega 2560 Board

5.5 IRF520 MOSFET module

This little module (HCMODU0083) is a breakout board for the IFR520 MOSFET transistor. The module is designed to switch heavy DC loads from a single digital or PWM pin microcontroller. Its main purpose is to provide a low-cost way to drive a DC motor for robotics applications, but the module can be used to control the highest current DC loads. Screw terminals are provided to interface to your load and external power source. An LED indicator provides a visual indication of when your load is being switched [9].

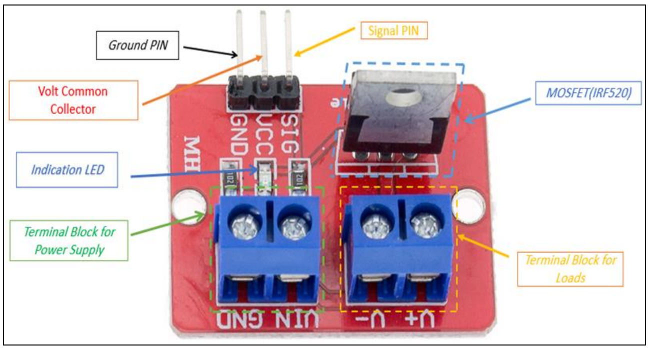


Fig 8: HCMODU0083 IRF520 MOSFET module

Terminal block for loads: This is the part of the module that is used to connect thick mains cable with the part, better than solder directly. It is used to connect loads (motors, pumps, fans, etc.) with the part module.

Terminal block for loads: This is the part of the module that is used to connect thick mains cable with the part, better than solder directly. It is used to connect power supply cables with the module part.

Indication LED: It is an LED fixed with the module, that varies its brightness according to the amount of voltage exerted on the module's pins, this helps to give indication about the channel gate of the module, where more brightness mor e opening gate and vice versa.

MOSFET (IRF520): It's the main element of the module, it is an electronic switch used to control current flow rate through electric circuits depending on the amount of voltage exerted on its pins.

Input jumper: Is used to supply power to the relay coil and LEDs. The jumper also has the signal pin, which receives the input voltage that comes from Arduino, in addition to the pin there is ground pin and VCC pin which is voltage common collector.

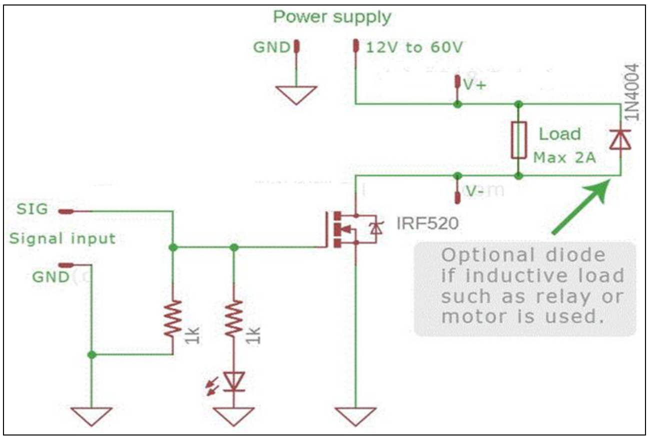


Fig 9: Internal circuit diagram of MOSFET module

5.6 Relay module

The relay uses an electric current to open or close the contacts of a switch. This is usually done using the help of a coil that attracts the contacts of a switch and pulls them together when activated, and a spring pushes them apart when the coil is not energized. There are two advantages of this system – First, the current required to activate the relay is much smaller than the current that relay contacts are capable of switching, and second, the coil and the contacts are galvanic ally isolated, meaning there is no electrical connection between them. This means that the relay can be used to switch mains current through an isolated low voltage digital system like a microcontroller [10].

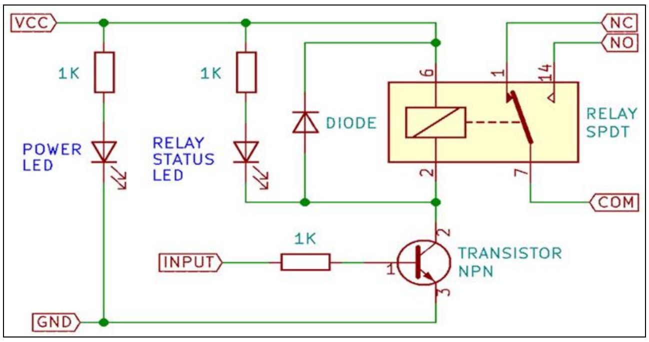


Fig 10: Relay internal diagram circuit

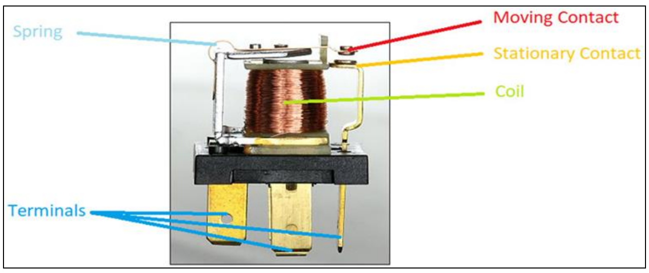


Fig 11: Mechanical construction

The single-channel relay module is much more than just a plain relay, it comprises components that make switching and connection easier and act as indicators to show if the module is powered and if the relay is active or not [10].

Table 2: Single-Channel Relay Module pins

Pin number	Pin name	Description
1	Relay Trigger	Input to activate the relay
2	Ground	0V reference
3	VCC	Supply input for powering relay coil
4	Normally Open	Normally open terminal of the relay
5	Common	Common terminal of the relay
6	Normally Closed	Normally closed contact of the relay

5.7 Normally closed contact solenoid valve

Solenoid valves are electromechanically operated valves that convert electric energy into mechanical energy. Their main purpose is to regulate the movement of gas or liquid and eradicate the need for an engineer to manually control the valve, saving time and money.

Solenoid valves consist of two basic parts: A solenoid (or electromagnet) and a valve. The valve body is made up of two or more orifices/openings. Whereas, the solenoid is home to several important parts, including a coil, sleeve assembly, and plunger. Solenoid valves work by employing the electromagnetic coil to either open or close the valve orifice. When the coil within the solenoid is energized, the plunger is lifted or lowered to open or close the orifice. This is what in turn controls flow, regulating the movement of gas or liquid.

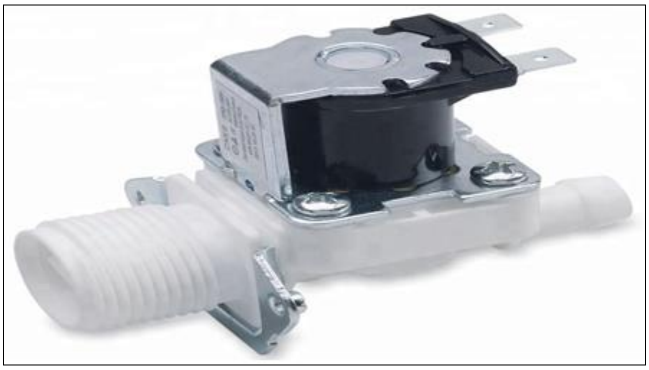


Fig 12: Normally closed contact solenoid valve

6. Design and implementation

This step explains the function of the components used in the project. Testing and calibrating the components of the system is done by wiring each component with Arduino and testing it with a simple code, to identify the properties of components.

6.1 Calibrating the components individually

Fig 2 shows the final block configuration of the system, so this section presents how to construct project components together, in addition, the components of the system will be individually tested with its software and hardware, to check its performance and response, this is necessary for system implementation. The procedures of implementation that will be presented give a clear overview of how the proposed system works successfully with its parts.

6.2 IRF520 MOSFET driver

This is an electronic element switch, used to control the current that flows through the pump. It opens its channel gate according to the voltage that comes from Arduino exerted on its gate. Fig 13 shows the diagram connection and the result is presented in Table 3.

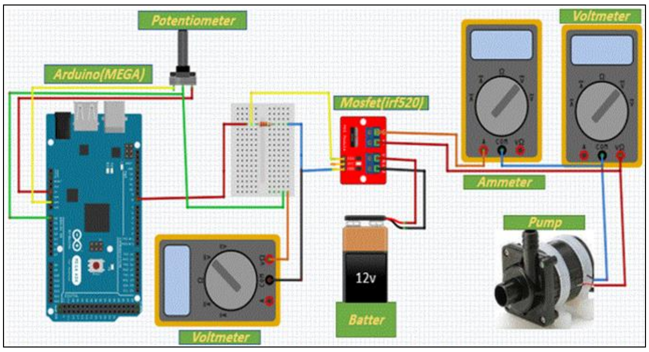


Fig 13: Wiring test of MOSFET Module

Table 3: Electric specification of the MOSFET

Potentiometer value (V)	Voltmeter (V)	Ammeter (A)
0	0	0
0.5	0.2	33.5m
0.6	0.4	47.7m
0.7	0.6	60m
0.8	0.75	81m
0.9	0.856	96m
1	0.906	100m
1.2	2.1	120m
1.5	2.6	165m
2	3	0.2
2.5	3.5	0.24
3	4.7	0.24
4	5	0.24

6.3 Construction of MOSFET with Arduino.

After identifying the electric characteristic of the MOSFET, the MOSFET is constructed with Arduino for the final configuration used in the project as shown in Table 4 which shows the wiring pins of components to Arduino.

Table 4: Wiring MOSFET(IRF520) with Arduino pins

Wire Color	Arduino (MEGA)	MOSFET (IRF520)	Pump	Batter
Red	PIN 6	SIG VIN	/	V+
Blue	/	V-	V-	/
Green	/	V+	V+	/
Black	GND	GND	/	V-

6.4 Test solenoid valve

A solenoid valve is an electromechanically operated valve that works with 220v AC. It is used in the project in the draining process, which opens when the drain pump is running and closes the pipeline stream when the drain pump stops, preventing any pipeline stream leakage. To test a device, the connection circuit in Table 5, should be wired, and after running the solenoid should produce a contact sound if it is done.

Table 5: Wiring pins of Relay with solenoid valve

Wire Color	Arduino (MEGA)	Relay	Solenoid valve	AC power
Red	5V	COM & VCC	/	~V+
Blue	GND	GND	/	/
Yellow	Pin 37	5V	/	/
Orange	/	ON	V+	/
Gray	/	/	V-	~V-

6.5 Test float level sensor

It is a sensor device used to send information about the level of the water tank. Table 6 shows the wiring pins.

Table 6: Wiring pins for level sensor test

Wire Color	Voltmeter	Arduino (MEGA)	Level Sensor	Battery	Resistor
Red	V	5V	V-	V+	V+
Black	COM	2GND	/	/	V-
Orange	/	Pin 6	/	/	/
Blue	/	/	V+	V-	/

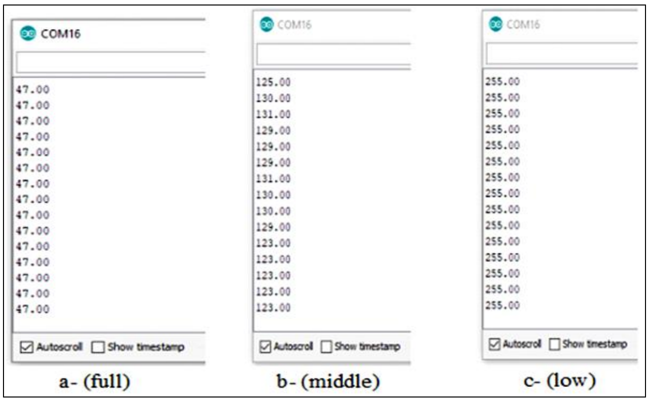


Fig 14: Float position

6.6 Hardware system construction

After testing the components of the model and identifying their principle working and response, the parts of the system are combined forming the final proposed system for the PID controller to control the level position inside a tank. Fig 14 shows the final main wiring of the model to be followed as a guide, and Fig 15 shows the real picture of the model after construction. The final step is to construct the appropriate software code.

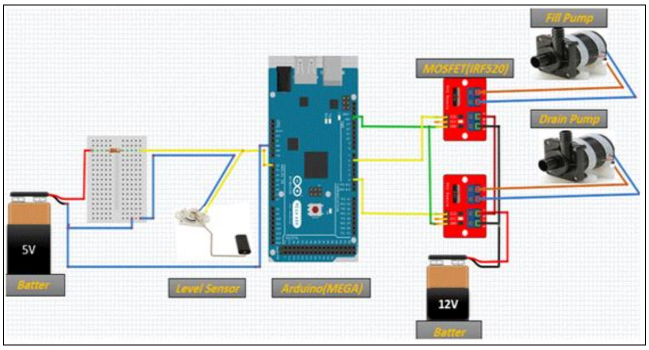


Fig 14: Overall system wiring

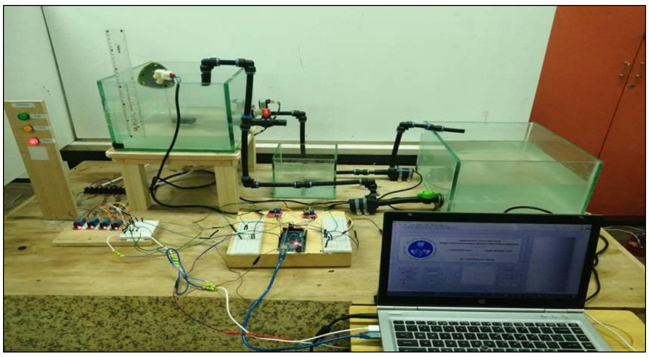


Fig 15: System hardware configuration

7. Results And Discussion

In this part, the designed system will be tested and applied to the suggested process, which contains two tanks, two pipeline streams, a manual valve, a solenoid valve, two water pumps, and a float level sensor as shown in the following piping diagram in Fig 14, 15. The two water pumps that used in the system, one to drive water from the source tank to the process tank (filling up process), and the other pump derives water from the process tank back to the supply tank (draining process). The float is used to inform the controller of the system about the height of level measurements and to make decisions, and the pumps are driven according to the measurements.

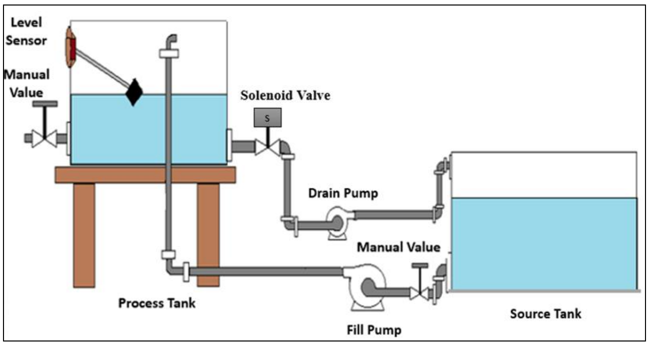


Fig 16: System piping diagram

7.1 System operation

Firstly, the desired level (setpoin) is selected from the computer LabVIEW interface panel, after that the controller compares the setpoint level value with the immediate height level quantity of water in the process tank, if the setpoint value is greater, then the controller gives the order to start the fill-up pump, that drives water from source tank to the process tank, accordingly the float will move upwards due to the incoming water, as soon as the desired level is being reached the flow rate of the pump will be reduced analog, avoiding any overshoot depended on float feedback measurements until the desired level is reached, then the controller stops the pump. If the result of the comparison was the contrary, which the setpoint value is less than the height level quantity in the tank, in this case, the controller decides to start the drain pump ON, where the water is derived back from the process tank to the supply tank analog too, to maintain the selected level value.

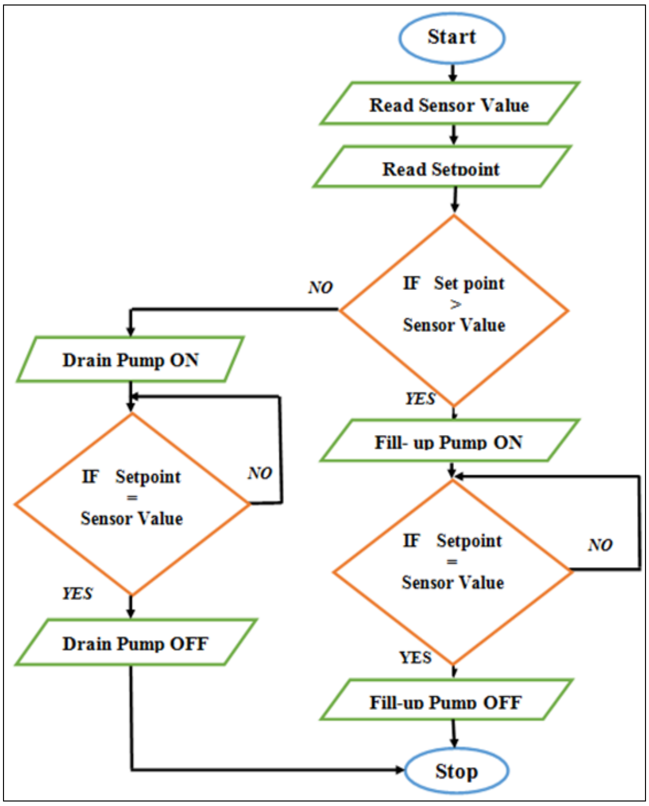


Fig 17: System operation flowchart

7.2 LabVIEW command window interface

The designed level control system needs an interface between the model and the user or operator, LabVIEW is a better solution, it is a virtual instrumentation software that allows the user to deal with or monitor the process without direct contact with the system. It shows the designed layout window for the system, this window provides an empty field to enter the desired setpoint into, the below field shows the actual level value measured inside the process tank, and the last field shows the error value, in addition to this layout contains three indication lamp for the tank state, moreover the ability of layout to show a virtual tank simulated with the real hardware process tank, additionally to a virtual graph to present the measured level value and the desired setpoint.

7.3 Tuning process of PID parameters

PID parameters are responsible for the specification of closed-loop control systems, overshoot, settling time, rise time, etc., based on this, the parameters should be precisely selected, for the best performance of the system. There are many methods to obtain the required PID parameters for an optimal response in closed-loop control systems. A method known as Ziegler–Nichol's method, which is based on the transfer function modeling of all-control system components and performing simple mathematical evaluation techniques will enable us to obtain the required parameter values. In addition to another method that is commonly used, the Trial-and-Error method, this method is based on random tuning of the values of the parameters and then checking the error or observing the results, this process continues until the minimum error is achieved, moreover it is the process used in the project. Table 7 shows the Effect of increasing Kp, Ki, and Kd on a closed-loop control system.

Table 7: Effect of increasing Kp, Ki, Kd on the closed-loop control

Controller Parameter	Rise Time	Overshoot	Settling Time	Steady State Error
KP	Decrease	Increase	Small Change	Decrease
KI	Decrease	Increase	Increase	Eliminate
KD	Small Change	Decrease	Decrease	None

7.4 Trial-and-Error method

It is the method used, the process in the system does not consume time for a steady state, so it will be the appropriate and reliable method to obtain the best PID parameter values for an optimal response. This method follows certain consequences for tuning PID parameters, and is summarized in a few simple steps as follows:

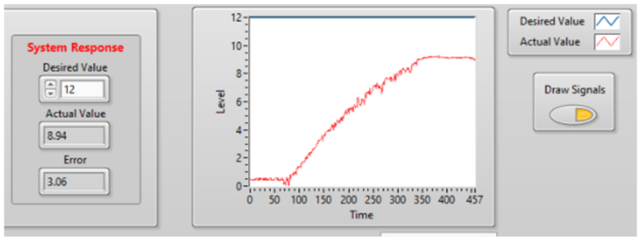
At the beginning of tuning let the parameters equal to zero, then raise the Kp parameter, until earning an acceptable response with some oscillation.

To reduce oscillation Kd parameter should be raised slowly, and observing the result, this process continuous until earning a response with minimum oscillation and error.

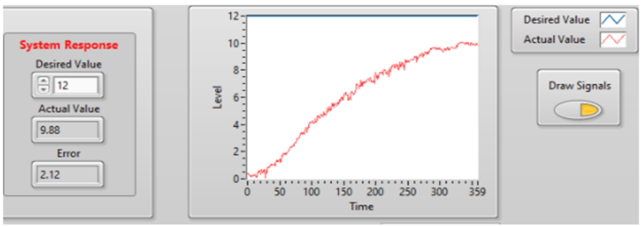
Reduce the error then the Ki parameter should be varied slowly, but with very minimum values.

7.5 Varying Kp only and ki=0 and Kd=0:

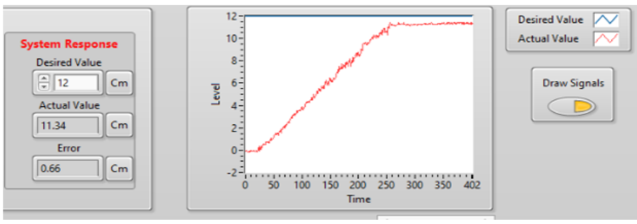
Fig 18 (a) shows the signal of a measured level with a setpoint of 12cm and kp=1.2 as a starting point, the error was too large. In (b) the error is reduced due to the increment of the kp from (1.2 to 1.5). the error value continues to decrease as soon as the kp value is raised, as in (c). Note that the level in the graph is in centimeters (cm) and time in seconds (s).



A: (kp=1.2, ki=0, kd=0)



B: (kp=1.5, ki=0, kd=1)

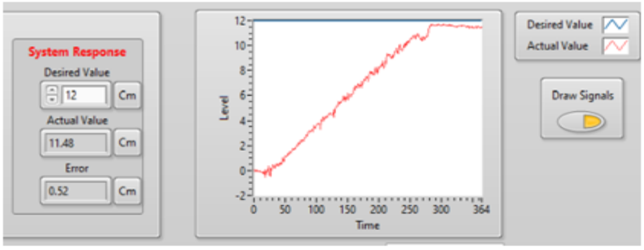


C: (kp=4.5, ki=0, kd=1)

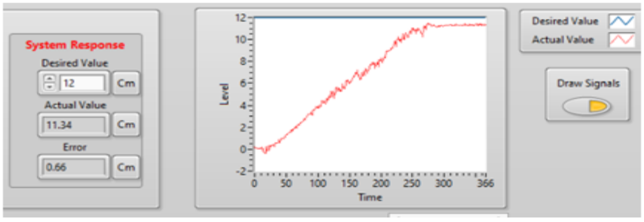
Fig 18: The response of the system

7.6 Tuning for kp fixed, ki =0 and kd is varied

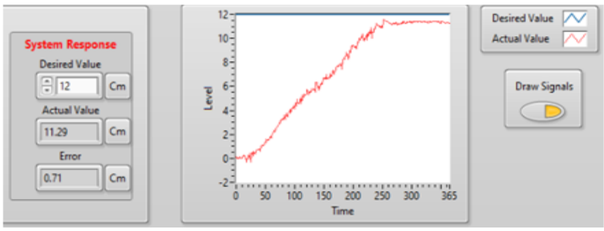
After obtaining an acceptable gain response from kp tuning, then the kd parameter value is adjusted, and here is tuned just to manipulate the speed of the response, not for damping, because the existence of electronic MOSFET eliminates overshoot and normal oscillation, as shown in Fig 19 (a) the transient response with kd=1 as a starting point, settling time was at approximately (300s), as the kd increases the response of the system speed increases as shown in (b) where Ts=(290s) and (c) where Ts=(260s), the kd parameter was set to 13, and the response seemed slightly slower compared to (c), due to the over-damped which negatively affected the transient response when the process sensor contains too much noise or distortion.



A: (kp=4.5, ki=0, kd=1)



B: (kp=4.5, ki=0, kd=5)

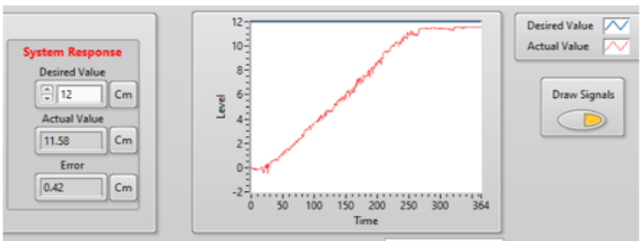


C: (kp=4.5, ki=0, kd=13)

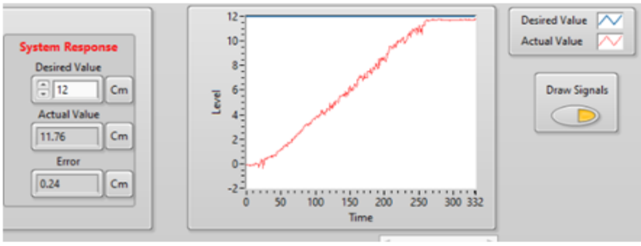
Fig 19: The response of the system

7.7 Tuning for ki and let kp and kd constants

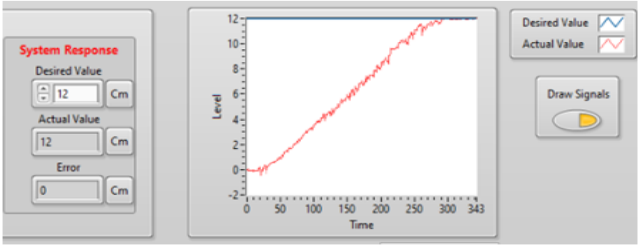
After tuning the kp and kd, and obtaining the best response with some error, in this turn the ki parameter should be adjusted, but with small values due to its large effeteness. Fig 20 (a) shows a response with ki=0.000001 with simple error, and as the ki increases in value, the error decreases compared to previous tuning results of the other parameters, as shown in next (b), and (c).



A: (kp=4.5, ki=0.000001, kd=5)



B: (kp=4.5, ki=0.000002, kd=5)



C: (kp=4.5, ki=0.0000035, kd=13)

Fig 20: The response of the system

7.8 Model results for step tracking

After tuning the controller parameters that satisfy the required specifications. The system was introduced to multiple setpoints and then examined in disturbance situations to check model performance. Fig 21 shows the response of the system with the tuned PID parameters, for multi-setpoints commanded at the same time starting from 10cm and then draining back to 6cm after filling up to 8cm, then the model tries to follow the desired setpoint with minimum error, note that the sensor exhibits large noise especially in drain process.

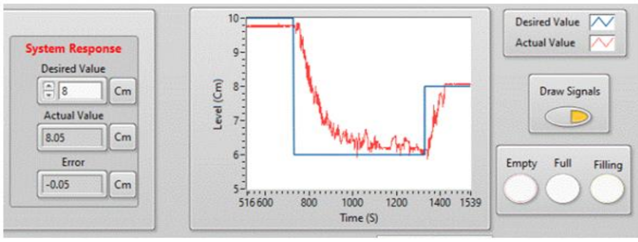


Fig 21: System response for step tracking

7.9 Model robustness

Robust control is the ability of a closed-loop control system to overcome external influences (disturbance). Fig 22 shows the system response when a cup of water is poured inside the process tank casing disturbance, note that the controller is maintaining the desired level setpoint by running the drain pump to eliminate over level, after that a manual drain valve is opened to make drain disturbance, then controller run filling pump to compensate the lost water.

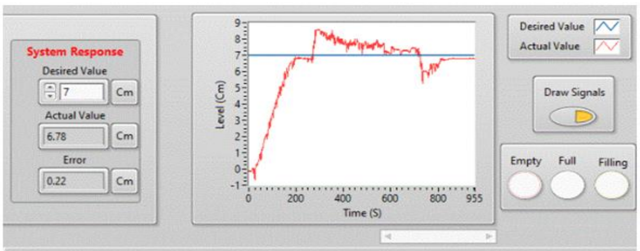


Fig 22: System response to disturbance

7.10 Sensor noise
In signal processing, noise is a general term for unwanted (and, in general, unknown) modifications that a signal may suffer during capture, storage, transmission, processing, or conversion [11]. Meanwhile, sensor noise is a common problem that faces engineers and affects response, as shown in Fig 21, the system response exhibited noise that may affect the stability, so it will be better for the response to contain such a small error to prevent that noise from reaching the desired setpoint, this for the system to distinguish between sensor noise and oscillation.

The PID algorithm code decryption

Arduino which is the main controller of the system, is programmed with full code that works to maintain the controlled variable within a desired level setpoint. There is a part in the code specialized with the PID algorithm, the part is discussed and described in a block diagram.

The IDE code for the PID algorithm is as follows:

```
timeChange = (NOW-lastTime);
 error = Setpoint-input;
  errSum += (error*timeChange);
   dErr = (error-lastErr)/timeChange;
    A = kp*error;
    B = ki*errSum;
    C = kd*dErr;
    Output = A+B+C;
    Output = constrain (Output,0,255);
   analogWrite (fillpump, Output);
```

8. Conclusion
PID control system for controlling the level inside tanks or containers has been presented, in addition to the design of the interface circuit which consists of MOSFETs that work in controlling high current loads with low voltages that come out from microcontrollers, based on the electric characteristic of MOSFETs. The determination of the dimension of the process tank only depends on the distance movement range of the level float sensor, whereas the source tank dimension is selected to be just enough to supply the process tank with the needed water.

In tuning the parameters of PID the selected method was the Trial-and-Error method, the reason behind choosing such a method among too many heuristic methods, is that this model does not consume too much time to reach a steady state, where this method of tuning is used only with process that does not take too long time to reach steady-state response.

During the tune of parameters that fit this model, the kd parameter should be in between (5 to 13), for better speed of the response, where too large values cases damped and too low values produce slow response, also to obtain amplified response without overshoot the parameter kp should be in between (1.8 to 2.1), final tuning steps is tuning ki parameter to reduces the error and should be small values due its sensitivity in this model and should be in between (0.000001 to 0.0000035) where any value of ki greater costs the response to be unstable.

To avoid or reduce the effect of the float level sensor, should the water be in a calm situation during the pump working, the signal should have some error to prevent the noise from making the system unstable.

Finally, the ability of the model to achieve low error and accuracy is needed, additionally to the ability to track the step change, moreover the ability of the system to maintain the desired level when external disturbances occur.

9. References

1. Getu BN, Hamad NA, Attia HA. Remote controlling of an agricultural pump system based on the dual tone multi-frequency (DTMF) technique. Journal of Engineering science and Technology. 2015; 10:1261-1274.
2. Bennett S. Development of the PID controller. IEEE Control Systems Magazine. 1993; 13:58-62.
3. Bansal HO, Sharma R, Shreeraman P. PID controller tuning techniques: A review. J. Control Eng. Technol. 2012; 2:168-176.
4. Abdul Latheef M, Shinu M, Clint Augustine RA. Labview Based Pid Controller For Cylindrical Tank Level Controlling. International Advanced Research Journal in Science, Engineering and Technology. 2015; 2.
5. Khan MR, Khan AA, Ghazali U. Speed control of DC motor under varying load using PID controller. International Journal of Engineering (IJE). 2015; 9:38-48.
6. Sen PC. Principles of electric machines and power electronics: John Wiley & Sons, 2021.
7. Theopaga AK, Rizal A, Susanto E. Design and Implementation of PID Control Based Baby Incubator. Journal of Theoretical & Applied Information Technology. 2014; 70.
8. Vikhe P, Punjabi N, Kadu C. Real time DC motor speed control using PID controller in LabVIEW. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering. 2014; 3:12162-12167.
9. Admin. IRF520 MOSFET Driver Module (HCMODU0083), 10 May, 2015. Available: https://forum.hobbycomponents.com/viewtopic.php?t=1872.
10. 5V Single-Channel Relay Module, 15 May, 2020. Available: https://components101.com/switches/5v-single-channel-relay-module-pinout-features-applications-working-datasheet
11. Tuzlukov V. Signal processing noise: CRC Press, 2018.