Capstone Course in ECET Program: Design and Implementation of PID Controller Using FPAA

Sri Kolla and Brandon Bostater

Electronics and Computer Engineering Technology
Bowling Green State University
Bowling Green, OH 43403
Email: skolla@bgsu.edu

Abstract

Capstone course has become an essential part of our Electronics and Computer Engineering Technology (ECET) program at Bowling Green State University (BGSU) since opting for ETAC/ABET accreditation. Design and implementation of a Proportional-Integral-Derivative (PID) controller for a liquid level system using Field Programmable Analog Arrays (FPAA), developed as part of a capstone course project, is described in this paper. Using AnadigmDesigner® 2 software, the PID controller is created and downloaded to the FPAA. The controller accepts a set point voltage and an input voltage corresponding to the level of the liquid to produce an output voltage which controls the speed of a DC motor to pump the liquid into the tank from a reservoir. The implemented system is tested and its performance to control the level of liquid in the tank is evaluated for set point and other changes. The paper also presents these test results of the capstone course project.

I. Introduction

Capstone course has recently become an essential part of our Electronics and Computer Engineering Technology (ECET) program at Bowling Green State University (BGSU) since opting for ETAC/ABET accreditation. Capstone course projects allow students to exhibit their technical knowledge, project management, critical thinking and communication skills[1]. These capstone projects are viewed as an instrument for outcomes assessment for accreditation. Control systems and renewable energy systems topics that used new technological developments are a part of several capstone projects in our ECET program. One such technology is Field Programmable Analog Arrays (FPAA)[2].

Field programmable analog arrays can be used as analog equivalents to the Field Programmable Gate Arrays (FPGA) to implement complex analog signal processing functions. In one technology, Configurable Analog Blocks (CAB) made of switched capacitor arrays are used in FPAA[2], there are other technologies such as System-on-Chip (SoC) are recently developed for FPAA[3]. FPAA were proposed for use as controllers in the implementation of control systems recently[4]. One of the advantages of using analog circuits for controllers over their digital counterparts is that they do not require analog-to-digital converters and digital-to-analog converters thus reducing the cost and quantization errors. FPAA were proposed by different

Proceedings of the 2018 ASEE North Central Section Conference

researchers for use in a DC servo position control system[5], scanning probe microscope[6], chaotic hyper jerk system[7], robotics[8,9], acoustic signal processing[10], and power systems[11]. FPAA were also proposed to develop remote labs for online learning[12,13] and other education uses[14]. In this paper, design, implementation and test results of a low cost Proportional-Integral-Derivative (PID) controller using FPAA for a liquid level system are described.

The PID controller is implemented using commercially available AN221E04 FPAA[2] to control the level of liquid in a tank. The controller accepts a set point voltage and an input voltage corresponding to the level of the liquid to produce an output voltage which controls the speed of a DC motor to pump the liquid into the tank from a reservoir. Using AnadigmDesigner® 2 software[2], the PID controller is first created and simulated with different values for proportional, integral and derivative gains. Once the proper values are found, the program is downloaded to the FPAA. In a control circuit, the FPAA voltage output can control the pump motor to change the liquid level. The implemented system is tested and its performance to control the level of liquid in the tank is evaluated for set point and other changes. The paper presents these test results in addition to control system design and implementation details of this capstone course project.

II. Field Programmable Analog Arrays

Field programmable analog arrays can be used as analog equivalents to the field programmable gate arrays to implement complex analog signal processing functions. One of the advantages of using analog circuits over their digital counterparts is that the FPAA do not require an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC), since all of the computations are done with analog circuitry, thus reducing the cost and quantization errors.

Located inside the commercially available AN221E04 FPAA are four Configurable Analog Blocks (CAB) shown in Figure 1[2]. These blocks consist of switches, capacitors, and operational amplifiers shown in Figure 2[2]. These switched capacitor arrays can be used to simulate many different electrical components such as resistors and other more complex circuits. AnadigmDesigner® 2 software[2] is used to design complex circuits using pre-existing Configurable Analog Modules (CAM)[2]. Users can view a list of these CAM, choose the ones that they want to use, and place them directly onto the FPAA user interface block. For example, Figure 3 shows the user interface block of several CAM inserted, with the input connected to the input side of the CAM, and the output of the CAM connected to the output pins of the FPAA. In this project, AN221E04 FPAA is used to implement a PID controller in a closed-loop control system, and Figure 3 shows that CAM for PID control algorithm[2].

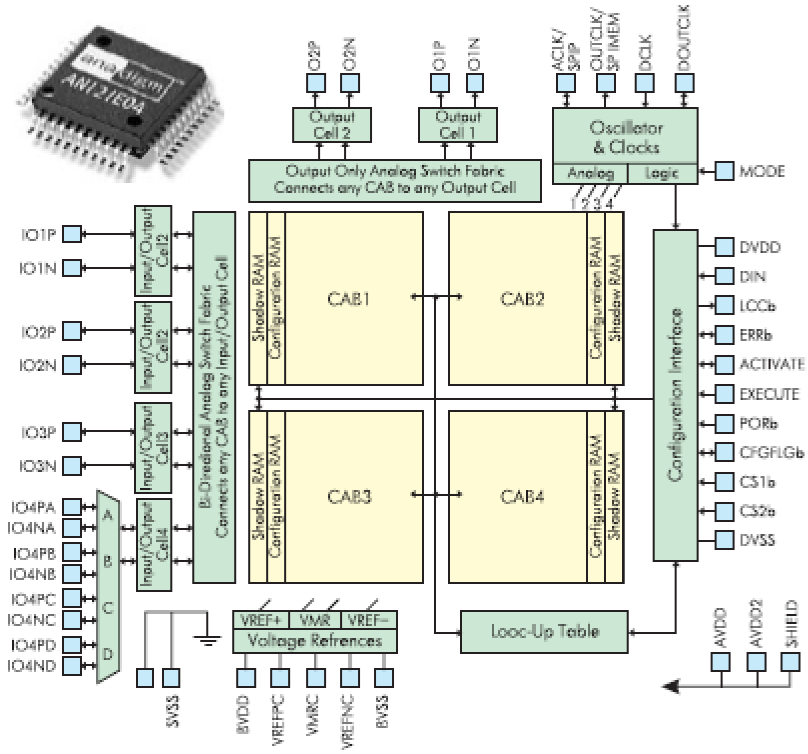

Figure 1. FPAA architecture (Anadigm, 2018)

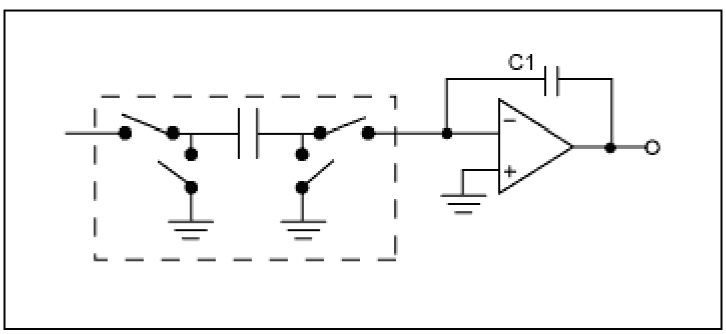

Figure 2. Switched capacitor schematic (Anadigm, 2018)

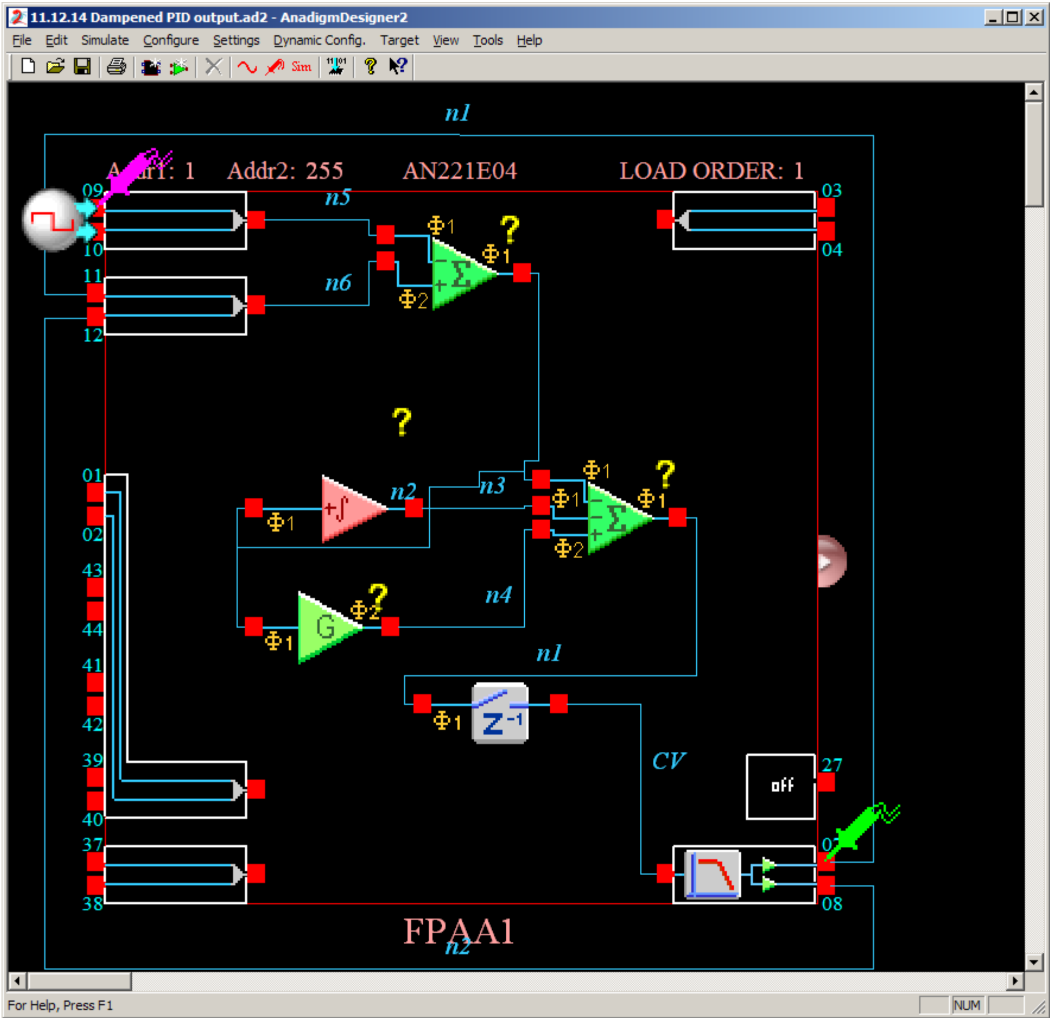

Figure 3. AnadigmDesigner® 2 PID controller.

III. PID Control

The type of controller used in this project is the Proportional-Integral-Derivative[15], which is an industry standard control method. This three mode controller can be broken down into how each mode is used in unison to create a controller that produces good system output. The proportional mode changes the output signal directly proportionally to the controller input. This mode introduces an offset (residual) error which does not allow for the output to return to zero-error. Though the proportional controller does not eliminate the error, it reduces the error. By adding the integral mode, the controller can correct a residual error that occurs during the operation. By introducing the derivative mode, the controller can anticipate future values by determining the rate of change of the variable. The PID controller can be mathematically represented by

$$p(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{d(t)} + p_0,$$

where p(t) is the controller output, e(t) is the controller input, p_0 is the initial controller output, K_P is the proportional gain, K_I is the integral gain and K_D is the derivative gain. By properly tuning the three gain parameters of the PID controller, it is possible to get different responses for the system output[15].

AnadigmDesigner® 2 software[2] allows selecting proper PID gain values by simulating the system response as shown in Figure 4. Table 1 shows one such set of gains. Given these gains, the liquid-level PID controller achieves a nearly damped response water level while reaching the set point, rather than producing a large overshoot response.

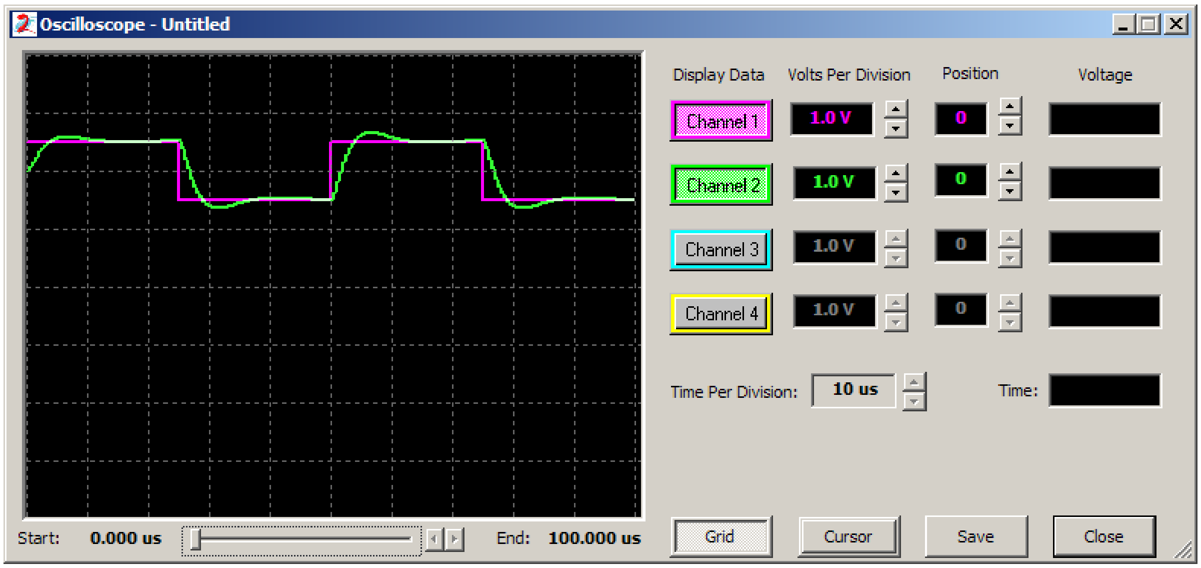

Figure 4. PID controller response simulation for selected gains in AnadigmDesigner® 2.

Table 1. PID gain values.

PID Gains	Gain Values
K_P	1.5
K_I	0.5
K_D	0.1

IV. Liquid Level Control System

In the designed system for this capstone course project, shown in Figure 5, a control loop is used to control the level of the liquid (water) in the tank. The overall cost to build the system is low, see Figure 6. The system consists of a water reservoir from which the pump gets the water, a holding tank, and a float which measures the water level in the tank. A DC motor operates the pump, and the motor voltage is controlled using an operational amplifier and transistor circuit. Attached to the float is a potentiometer to give a voltage corresponding to the water level.

Another potentiometer adjusts the set point voltage. The FPAA shown in Figure 7, with the PID controller programmed into it, is the main computational unit in the process. Figure 8 shows the block diagram of this closed loop control system.

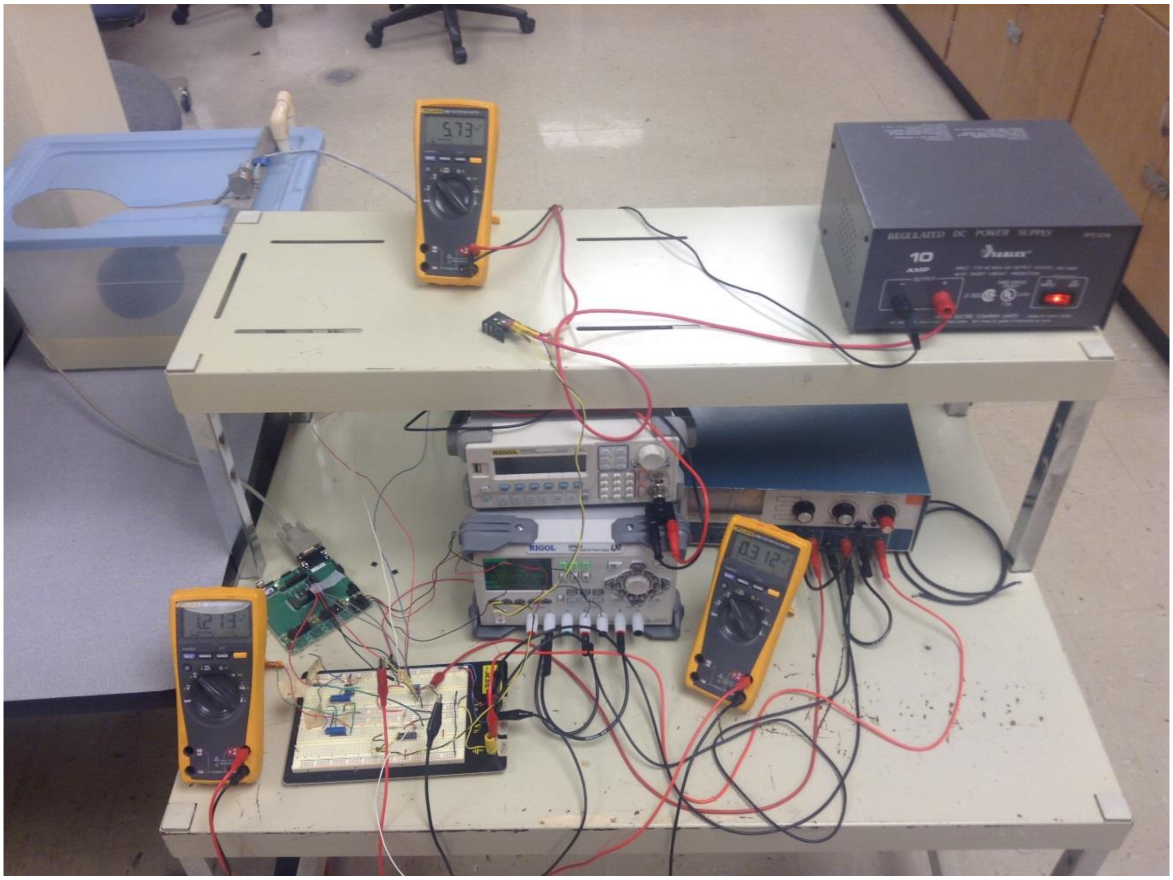

Figure 5. Overall liquid level control system.

Figure 6. Water level tank and reservoir.

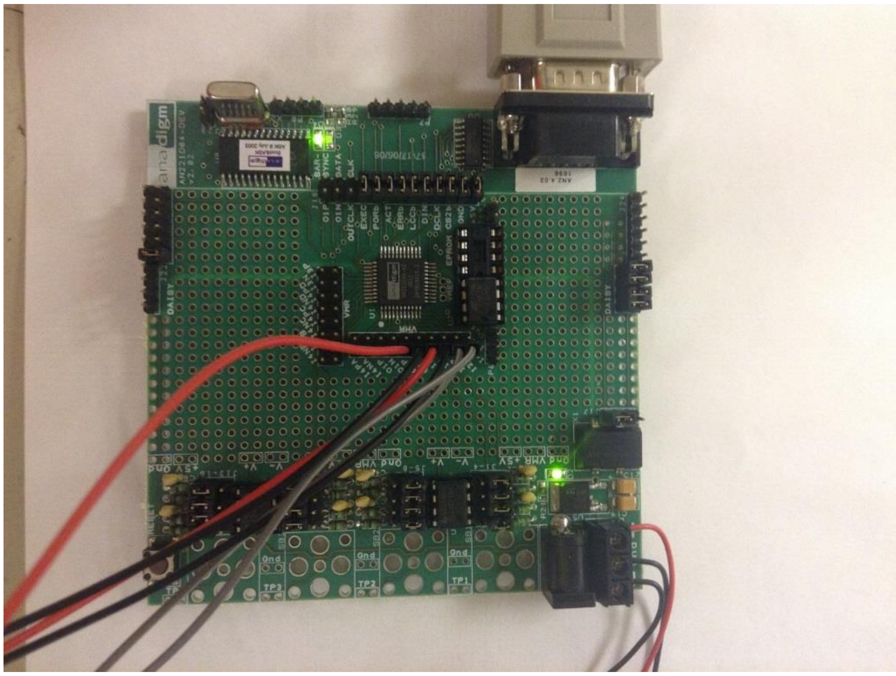

Figure 7. FPAA input and output connections.

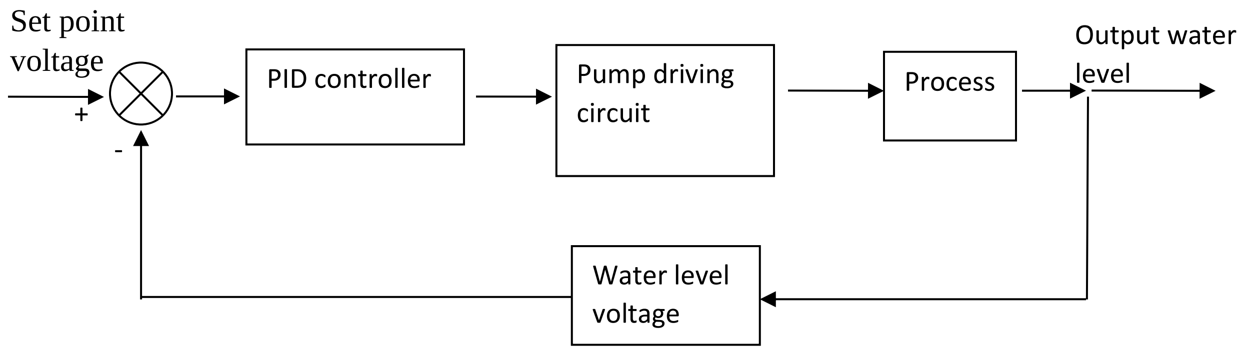

Figure 8. The closed loop system block diagram for the liquid level control system.

IV.1. Control System Operation

The electronic schematic diagram of the liquid level control system is shown in Figure 9. The set point is adjusted by a voltage that is varied by a 10k potentiometer, which is fed directly into the input pins of the FPAA. The set point voltage is compared with the feedback voltage, and the PID controller makes the proper adjustments to reduce the error between them. The FPAA then outputs the correcting voltage to the input of an operational amplifier circuit. This amplifier circuit is used to increase the voltage to a more useable range as the FPAA is limited to outputting a voltage between 1.5 and 4 volts. The output from the amplifier circuit is connected directly to the base of a transistor which is used to vary the voltage driving the DC motor. The collector of the transistor is connected to a 12V power supply, and the emitter connects to the DC motor shown in Figure 9. The DC motor is used to pump water into the holding tank from the reservoir. The speed at which the motor pumps water into the tank varies based on the voltage applied to the transistor, and determines the level of the water in the holding tank. Since the tank has a constant flow of water exiting the tank, the pump must be continuously pumping water into the tank to maintain the correct water level as determined by the set point voltage. The flow rate of the water exiting the tank can be adjusted manually via a valve located at the bottom of the tank. The float in the water tank is connected to another potentiometer. This potentiometer is connected to a 12V power source, which gives a voltage based on the level of the water. As the water level rises, so does the float, and the arm of the float changes the potentiometer, which in turn increases the voltage. This voltage is fed back into the FPAA, which compares it to the set point voltage, and repeats the closed loop system cycle.

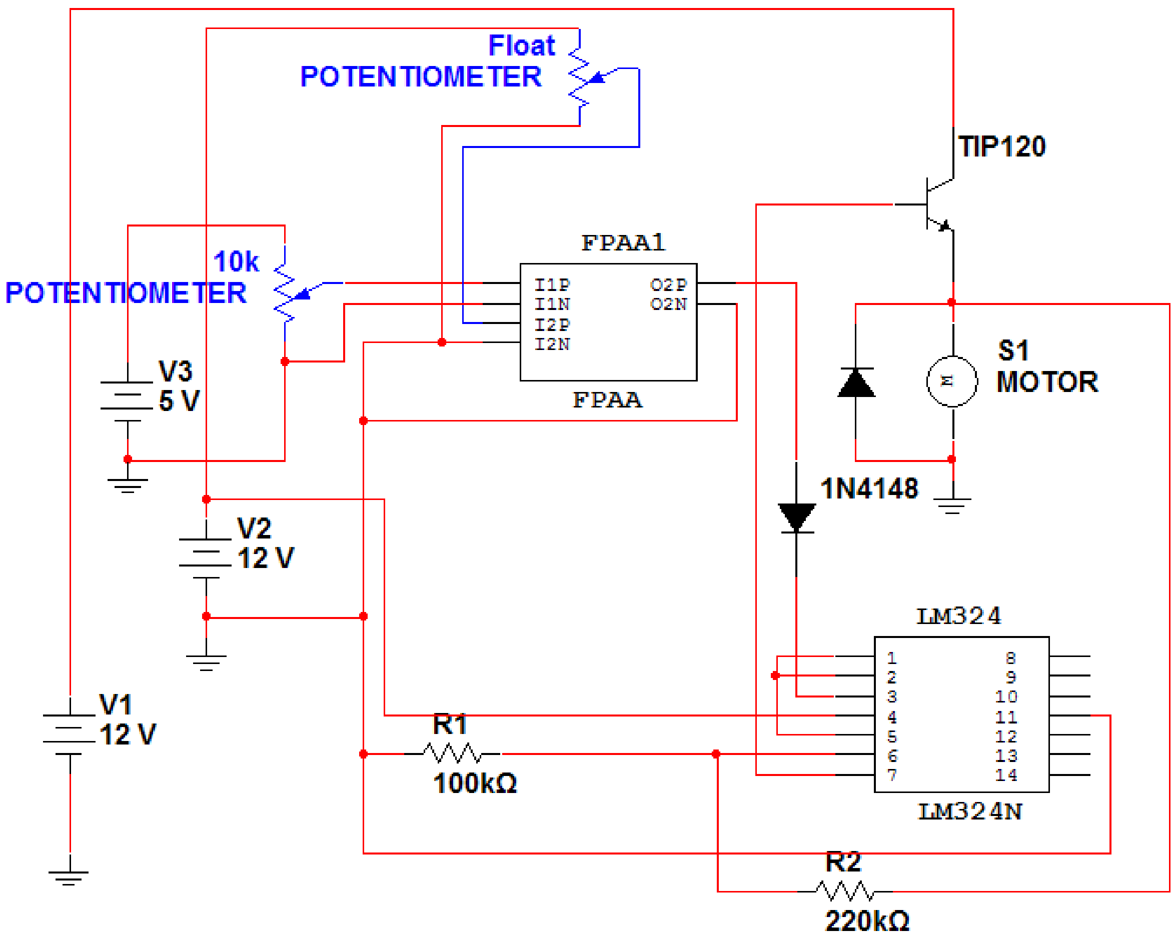

Figure 9. Overall liquid level system electrical schematic diagram.

V. Test Results

After all the components of the system are assembled, the overall system is tested, and the system worked as intended. A change in the set point voltage causes the amount of water being pumped into the tank to vary as expected, and the water level adjusts to the corresponding set point level. Creating a disturbance in the process by changing the outlet valve position results in a drop in the tank water level which is adjusted back by the PID controller action by changing the water inlet flow through pump motor voltage.

The response from the PID controller when the set point is changed from one value to another is shown in Figure 10. It shows the average level response of the float in the water tank as the set point is quickly changed from 0.332V to 0.621V shown in a blue dotted line. As can be seen, the pump rapidly begins pumping water into the holding tank, causing the float to go up as the water level is increased. This effect also increased the voltage from the level potentiometer which is plotted in the graph by a red line. The system response exhibited a nearly damped response without producing a large overshoot which can be seen in Figure 10 for the PID controller gains selected.

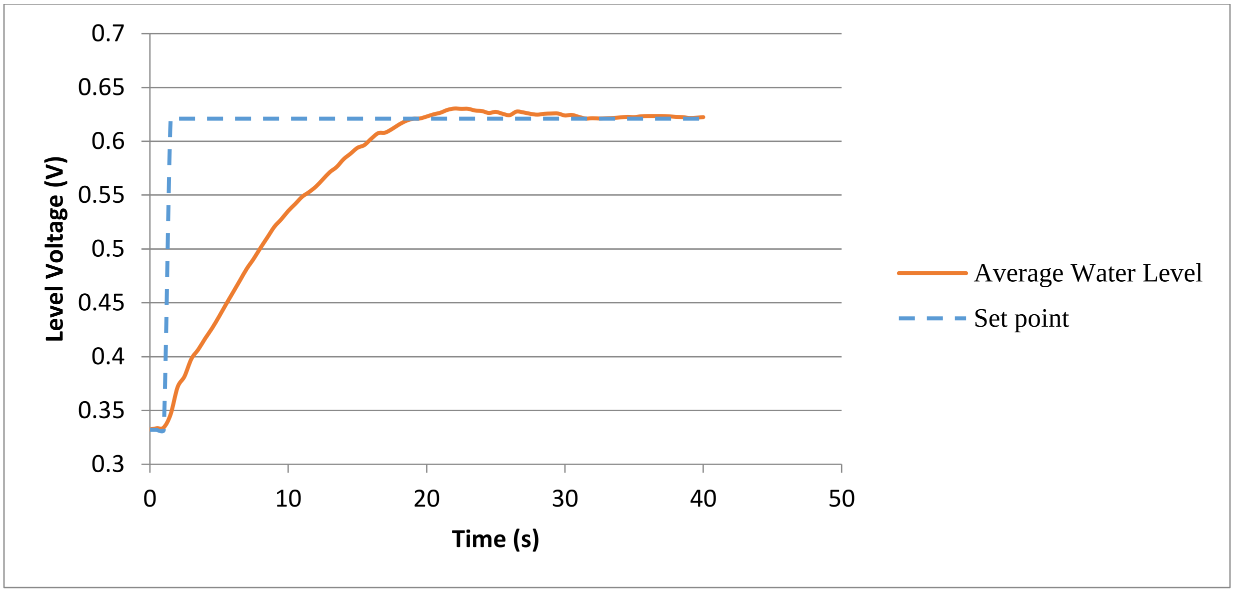

Figure 10. Liquid level response for set point change.

VI. Conclusions

A control system is successfully implemented in this capstone course project to control the level of water in a tank using FPAA. The use of the field programmable analog arrays offers a useful replacement of the field programmable gate arrays because they can complete the same tasks, and require no ADC and DAC since they operate using analog signals. The PID controller programmed using AnadigmDesigner® 2 software offers a stable output capable of reducing errors. Using the PID controller in this project showed how sensitive the tuning process is, and gives a real life application with actual responses. This capstone project allowed the student to integrate the technical knowledge gained in ECET program courses at BGSU with the latest FPAA technology to design and build a practical system. It also allowed the student to apply project management, technical report writing and presentation skills learned in various other courses taken as a part of ECET program at BGSU. This information is useful for the outcomes assessment part of ETAC/ABET accreditation process.

Bibliography

1. J. Staniszewski and A. Sergeyev, "Design and development of a low-voltage, low-current 3-phase power enabled lab bench, Proceedings of 2017 ASEE Zone II Conference, San Juan, PR, March 2017.
2. Anadigm, http://www.anadigm.com/, retrieved on January 2018.
3. S. George, et al., "A programmable and configurable mixed-mode FPAA SoC," IEEE Transactions on VLSI Systems, vol. 24 (6), June 2016, pp. 2253-2261.

4. J. M. Galliere, "A control-systems FPAA based tutorial," Proceedings of 2nd WAEAS/IASME International Conference on Educational Technologies, Bucharest, Romania, October 2006.
5. S. Majhi, V. Kotwal and U. Mehta, "FPAA-based PI Controller for DC servo position control system," Proceedings of 2nd IFAC Conference on Advances in PID Control, Brescia, Italy, March 2012.
6. M. Baranwal, "Application of field programmable analog arrays (FPAAS) to fast scanning probe microscopy, M.S. Thesis (Prof. S. Salapake, Advisor), University of Illinois, Urbana, IL, 2014.
7. F. Y. Dalkiran, and J. C. Sprott, "Simple chaotic hyper jerk system," International Journal of Bifurcation and Chaos, vol. 26, no. 11, 2016, pp. 1650189-1 to 1650189-6.
8. S. Koziol, et al., "Robot path planning using field programmable analog arrays," Proceedings of IEEE International Conference on Robotics and Automation, Saint Paul, MN, May 2012.
9. D. Berenson, et al., "Hardware evolution of analog circuits for in-situ robotic fault recovery," Proceedings of NASA/DoD Conference on Evolvable Hardware, Washington, DC, July 2005.
10. P. Falkowski and A. Malcher, "Dynamically programmable analog arrays in acoustic frequency range signal processing," Metrology and Measurement Systems, vol. XVIII, no. 1, 2011, pp. 77-90.
11. A. S. Deese and C. O. Nwankpa, "Design and testing of custom FPAA hardware with improved scalability for emulation of smart grids," IEEE Transactions on Smart Grid, vol. 5, no. 3, May 2014, pp. 1369-1378.
12. M. Carlos, et al., "Using remote lab for enhancing e-learning on FPAAs," International Journal of Online Engineering, vol. 12, no. 4, 2016, pp. 58-60.
13. J. Hasler, et al., "Remote system setup using large-scale FPAA to enabling wide accessibility of configurable devices," Journal of Low Power Electronics and Applications, vol. 6, no. 14, 2016, pp 1-17.
14. A. Macho, et al., "Dynamic reconfiguration in FPAA and its use in Education," Proceedings of Frontiers in Education Conference (sponsored by ASEE and IEEE), Indianapolis, IN, October 2017.
15. R. Bateson, Introduction to Control System Technology, 7th Edition, Prentice Hall, 2001.