Variable Controlled Solenoid Valve

Daniel Yoon - Maitreya Mujumdar - Ted Lietz
ECE 445 Project Proposal - Fall 2020
TA: AJ Schroeder

1 Introduction

1.1 Objective

Current control valves are primarily made using a servo or solenoid. A valve controlled by a servo offers high precision control but has a large size and a slower response time[1]. Solenoid valves are faster, but they usually do not have precision control and just offer binary output: an "open" or "closed"[1]. The valves in circulation are also large, and if a valve relies on current or voltage control, additional hardware is needed in between the microcontroller and the valve which can make the device bulky to operate.

These factors make it difficult for new robotic developments to reap the benefits of utilizing hydraulics because newer robots require both the precision control of servos and the fast response time of a solenoid valve. Additionally, a lot of these robots require waterproof valves; no solenoid valves today have that feature. More than that, the fact that a separate device needs to be attached to make the valve operate with current or voltage control makes it bulky, and difficult to waterproof.

Our goal is to integrate the current control functionality with a solenoid valve in the same device. This would allow us to have the precision control that a servo valve has, with the speed that using a solenoid provides. Moreover, this would allow easier waterproofing of the entire device, and the device would be smaller than adding the additional device to control current.

Our project will have a central control board that directs multiple solenoid valves via I2C protocol. For precision control, we use a PWM current signal to modulate how open the valve is.

1.2 Background

Efforts by solenoid valve manufacturers such as Tameson have succeeded in creating a variable controlled solenoid valve through the use of a PWM current control system [2]. However, there are some drawbacks that come with using their product.

Firstly, the solenoid valves are not waterproof. This project will be used to implement hydraulic movement of robots underwater. Therefore, our product has to be both light and waterproof. Secondly, when a solenoid valve breaks, it is almost impossible to open and fix the valve if the PWM generator circuit is damaged. We are implementing simpler design to the valves themselves and creating a control unit that sits outside the valves so that debugging is much easier. Thirdly, the average price of a Tameson variable controlled solenoid valve is about $200 [2]. Our goal is to make the valve as affordable as possible by changing the materials from brass or stainless steel to something cheaper since our applications will not need to withstand pressure up to 145 PSI like most Tameson valves do [3]. Fourthly, the method of controlling the current of Tameson proportional solenoid valves is by supplying a PWM signal from external hardware. This increases the size of the entire proportional valve system. Our design will utilize an I2C connection so that the circuitry involved with determining the PWM current signal can be separate to the actual valve instead of relying on the use of external hardware attached to the actual valve. This will make our design smaller overall and allow easier interfacing with robotic applications.

1.3 High-Level Requirements

- Valve is waterproofed to withstand pressure up to 1 foot underwater.

- Actual flow rate is within $\pm 5\%$ of the predicted flow rate value based on the 8-bit data signal being sent to the solenoid valve circuit via I2C.

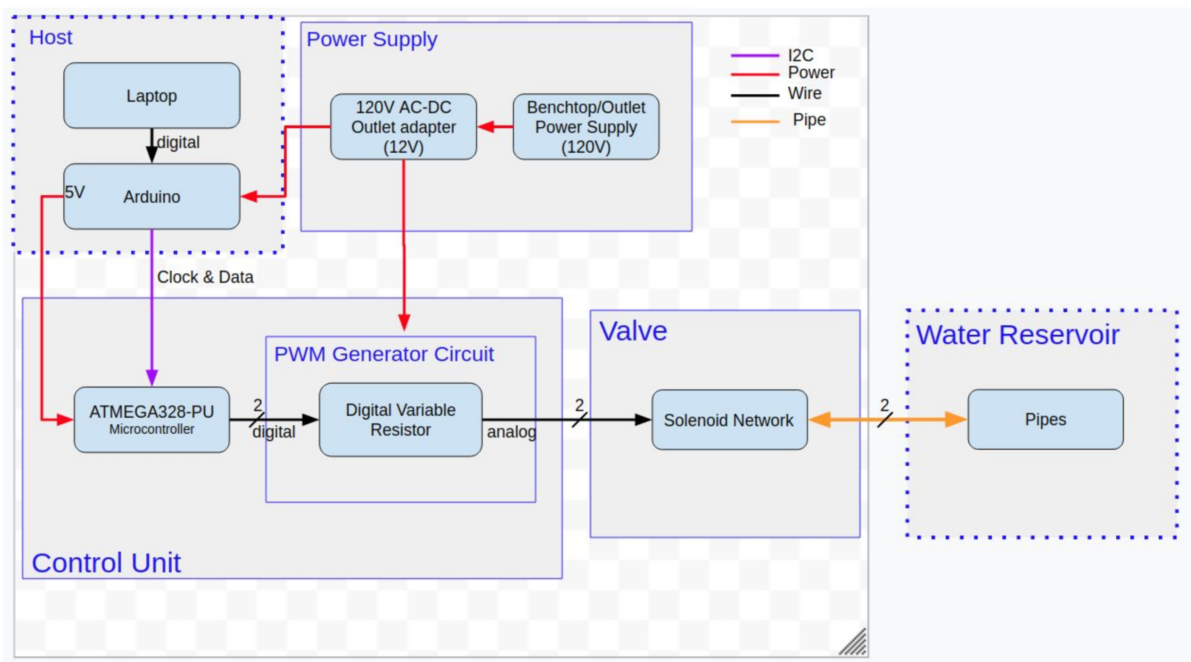
- Up to two valves able to be controlled from the same I2C connection.

2 Design

Our variable controlled solenoid valve will require four components for successful operation: a power supply, a user interface, a control unit, and the solenoid valve itself. Inside the control unit is the PWM generator circuit, and a microcontroller. The host will consist of a laptop that programs an Arduino to send I2C signals to the control unit. This will mimic the signals that will be sent to our solenoid valve system when used in actual robotic applications. The power supply for the PWM generator circuit will be produced from a wall outlet AC-DC adapter to 12V. The Arduino will supply the required 5V DC to power the microcontroller. In the control unit, the microcontroller will interpret the I2C signals and change the openness of the solenoids correspondingly by sending control signals to the PWM generator circuit. This PWM will then be output to the solenoid valves to create the variable openness of the valves. The entire valve will be waterproofed by being housed inside a casing.

Our system is scalable because if more valves are needed, the same I2C wire can be connected to more variable controlled valve modules up to the slave addressing limit of the I2C protocol.

To test our system, there will be a pipe connected to each individual valve from a water reservoir. We expect the pipes to be ¼ inch in diameter. The size of these pipes are subject to change as we progress further.



2.1 Power Supply

The power supply in this project will supply enough power to keep the two solenoid valves in certain open positions. The power supply will have to generate constant 12V DC. If there is a fluctuation in the voltage, the solenoid will not have a predictable open and closeness.

2.1.1 Outlet Power Supply

For the purpose of this project, we have decided to rely on a benchtop/outlet power supply connected to an adapter.

Requirement 1: Must provide 120V AC consistently to the adapter

2.1.2 AC-DC Adapter

A step down from the 120V wall outlet is needed to power the PWM generator circuit safely. Additionally, the circuit needs to have a DC power source to be able to operate correctly.

Requirement 1: Must convert 120V AC to 12V DC consistently
Requirement 2: Must provide 12V $\pm .1$ *DC consistently to the PWM Generator Circuit and the Arduino.*

2.2 Host

The host generates the I2C signals that will determine the openness of the solenoid valves. These signals are pre-programmed into the Arduino by a laptop

2.2.1 Laptop

Programs I2C signals onto the Arduino to send to the control unit.

Requirement 1: Must be able to connect to the Arduino and program I2C signals on it.

2.2.2 Arduino

A laptop will be used to program what I2C signals will be sent from the Arduino to the control unit. The Arduino will be powered through the wall outlet adapter. A preset clock will be implemented and connected to the microcontroller to allow the I2C protocol to operate.

Requirement 1: Must provide 5V $\pm .1$ *DC consistently to the microcontroller.*
Requirement 2: Must be able to produce and send coherent 8-bit I2C signals.
Requirement 3: Must sustain a clock speed of 100 kHz.

2.3 Control Unit

A control unit interprets the 8-bit values sent via I2C by the user interface. The control unit responds to the I2C signal sent from the host by regulating the duty cycle of the PWM signal sent to the solenoid in the valve. According to the duty cycle of the PWM signal, the solenoid will generate the force on the plunger to open or close the valve.

2.3.1 Microcontroller

We are using ATMEGA328-PU as our microcontroller. The microcontroller will interpret the 8-bit I2C values sent by the host robot and send out digital signals to the PWM generator to modulate the duty cycle of the PWM signals. When interpreting the I2C signal, the four MSBs will determine the openness of the first valve, while the four LSBs will determine the openness of the second valve. In this way, the synchronous operation of the valves can be achieved.

Requirement 1: Must be able to interpret 8-bit I2C signals coming from the host at a clock speed of 100kHz.
Requirement 2: Must be able to regulate the voltage of the output signal
Requirement 3: Must be able to control all voltage outputs simultaneously

2.3.2 PWM Generator Circuit

We are using a series of OpAmps to generate a PWM signal from the power that is received from the power supply. To change the duty cycle of the PWM signal, a digital variable resistor will be used. The digital variable resistors will receive instruction from the microcontroller to change its resistance to generate a specific duty cycle

Requirement 1: Must be able to generate and sustain 480mA output total (up to 240mA for each valve)
Requirement 2: Must be able to interpret output signals from the microcontroller

2.4 Valve

The valve regulates the flow of material by using a solenoid to operate the plunger. The PWM signal received from the control unit will determine the force acted on the plunger by the solenoid. On the opposite side will be a spring generating an opposite force so that the plunger can be specified to a position with one direction of force by the solenoid.

2.4.1 Solenoid

In order to make a solenoid, a wire will be spooled around a plastic tube. Sending a PWM signal with a duty cycle will create a magnetic flux. The magnetic flux will generate a force against the plunger made of diamagnetic material.

Requirement 1: Must produce enough magnetic field to life the plunger against the spring.
Requirement 2: Must be part of the waterproof design.
Requirement 3: Must be able to handle a constant current of 240mA.

2.5 Water Reservoir

A water source will be needed to flow through our solenoid valves so that flow rate tests can be performed.

2.5.1 Pipes

In order to test our solenoid valve system, pipes will be connected to each of the valves from a water reservoir.

Requirement 1: Must provide a constant stream of water at a constant pressure
Requirement 2: Must fit into the valve without leaking

2.6 Risk Analysis

A risk to the completion of the project is the pulse width modulation controller. This will be assembled from op-amps and variable resistors on the PCB. It is important that this PWM is incredibly precise, as precise as a digitally produced PWM. If this cannot be completed, the current modulation method would have to be changed to a different method, for example using MOSFETs.

The valve design is able to withstand water and a foot of water pressure is a risk to the completion of the project. This involves encasing the solenoid and the valve and using gaskets to create a watertight seal. Any water that gets into the wiring is a serious risk to both the circuitry of the control unit and safety risk.

The biggest risk to the successful completion of the project is establishing the I2C connection in the control unit. Because the connection would have to control 3 valves simultaneously. A single master and 3 slaves (the valves) will all need to be controlled synchronously, with a communication bus. The control unit will need to interpret the messages between the host and send the correct output to the valves. It will also have to regulate and control the voltages of the output for all valves. The microcontroller translating and understanding the I2C message and controlling the solenoids correctly.

3 Safety and Ethics

There are a couple of safety hazards involved in the project. The first and most obvious being anytime both water and electricity are involved, there is a possibility of being shocked if proper precautions are not taken. This is especially important because there is a possibility of a high power draw in this project, so any shock has a high possibility for damage. The easiest way to avoid this would be to make sure the power is off when making modifications, and always to make sure the circuitry is not damaged before operating the device. The device will have to be checked between every use to ensure safety, due to the high power draw. There is also a remote possibility of the pin in the solenoid becoming a projectile, if the current supplied to the valve is too high. The way to mitigate any safety concerns would be to make sure there is only one opening on the solenoid so the pin can only leave the solenoid in one direction. Another way to mitigate the concern is to make sure the opening is pointed away from anyone when the power is on.

The biggest ethics concern is the one surrounding IEEE Code of Ethics #1. "to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment". Since there are so many potential safety concerns, having ethical and safe design in mind. Because of this, all precautions have to be taken by every member of the team every time. And it is important that each team member make sure the rest of the team is taking the proper precautions every time. Every potential safety hazard has been thought of and accounted for to ensure safety and that the IEEE Code of Ethics #1 is followed.

References

[1] Hoyea, ‘Difference Between Solenoid & Servo Valves’, 2016. [Online]. Available: http://www.hoyea.com/news/Difference-Between-Solenoid-_-Servo-Valves.htm#:~:text=So%20the%20solenoid%20is%20an,cycle%20at%20some%20intermediate%20stage.&text=Servos%20start%20out%20as%20an,alternator%20but%20works%20in%20reverse. [Accessed: 16-September-2020]

[2] Tameson, ‘Proportional Solenoid Valve - How They Work’. [Online]. Available: https://tameson.com/proportional-solenoid-control-valve.html. [Accessed: 13-September-2020]

[3] Tameson, ‘Proportional Solenoid Valves’. [Online]. Available: https://tameson.com/valves/solenoid-valve/proportional/solenoid-valve/. [Accessed: 13-September-2020]

[4] TLXTechnologies, ‘PWM Solenoid Theory’. [Online]. Available: https://www.tlxtech.com/understanding-solenoids/theory-operation/pwm-solenoid-theory. [Accessed 15-September-2020]