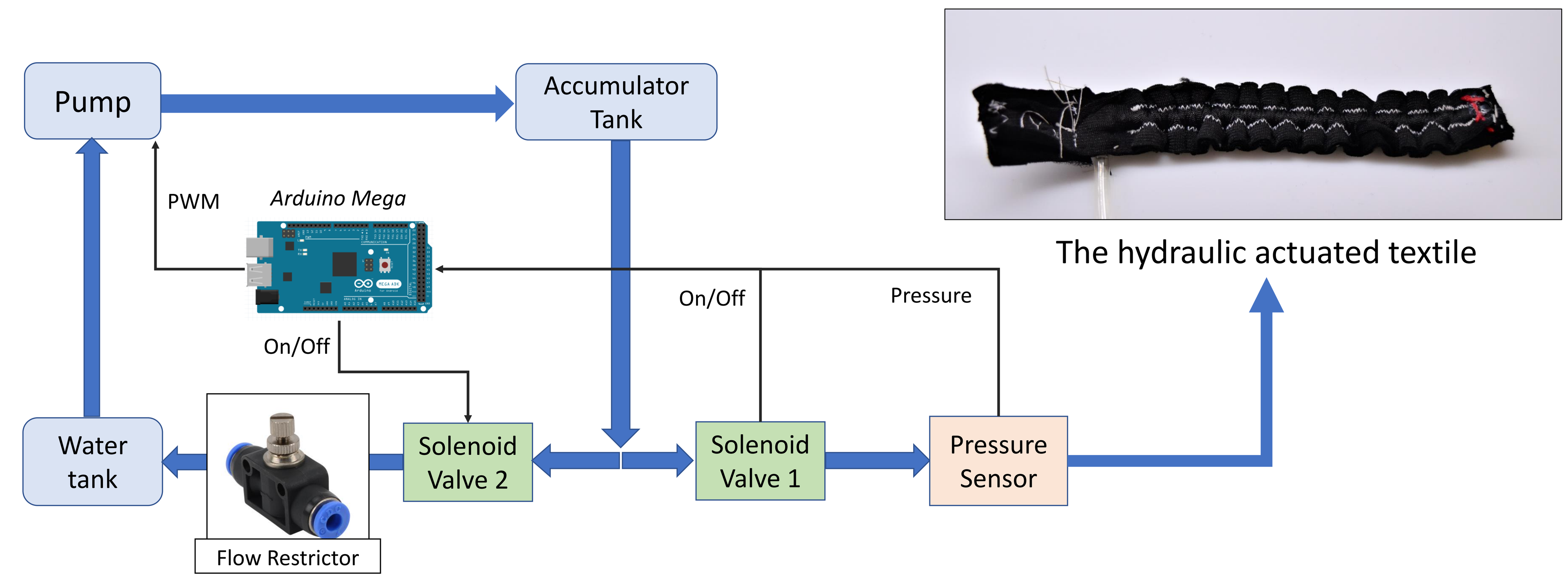
Fluid Control System for Hydraulic Actuated Textile

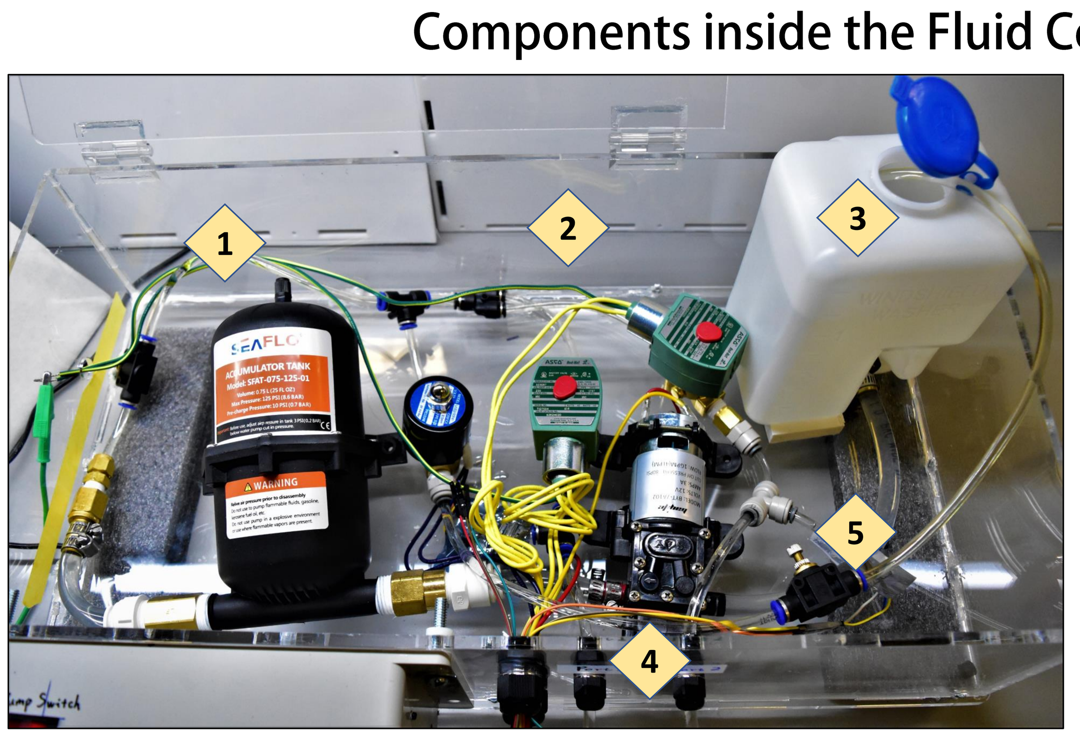
Xinghao Huang, Mengjia Zhu, Yon Visell
RE Touch Lab, Media Arts and Technology (2018)

Introduction

The flexible actuators are mainly used in soft robotics and medical compression device. These actuators, powered by fluid pressure, are able to contract, elongate, and bend while carrying a heavy load. My lab was researching a hydraulic actuated textile and currently seeking a hydraulic control system that is specifically designed for these textiles. Thus, I present the design, test, and implementation of a fluid control system, along with a physical prototype product: The Fluid Control Box. The system can provide a precise and stable water pressure for one or multiple hydraulic actuated textiles and accurately achieve motion control in a short time.



Components inside the Fluid Control Box



1. Accumulator
2. Solenoid Valve
3. Water Tank
4. Diaphragm Pump
5. Flow Restrictor
6. Control Panel

System Design

The fluid control system consists of both electrical and mechanical components. The Arduino Mega microcontroller is the “brain” that controls every mechanical movement in the system. A diaphragm pump is used to provide up to 100 psi water pressure, and the accumulator tank is able to reduce the pressure disturbance by preloading a pressure in water. By controlling two solenoid valves, the Arduino can let water either flow back to the tank or pump into the actuator. A pressure sensor will be installed before the actuator to detect the pressure change. The Arduino will then control the pumping speed through PWM value according to the pressure change.

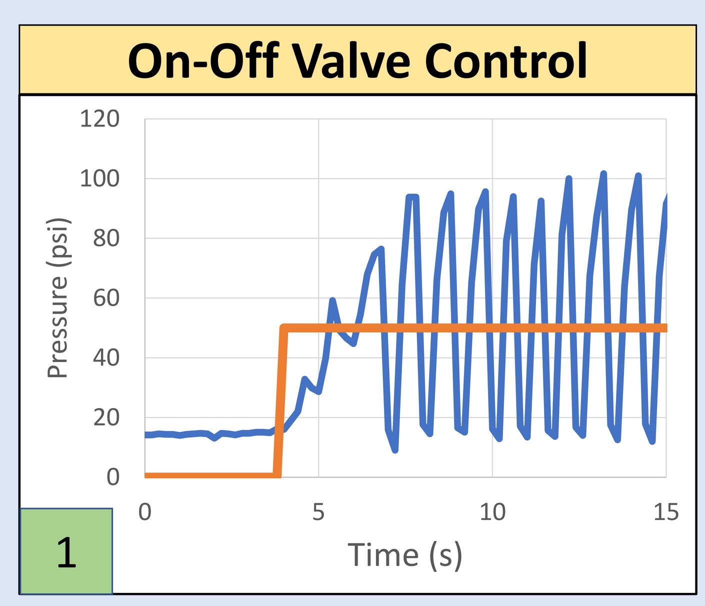
Methods

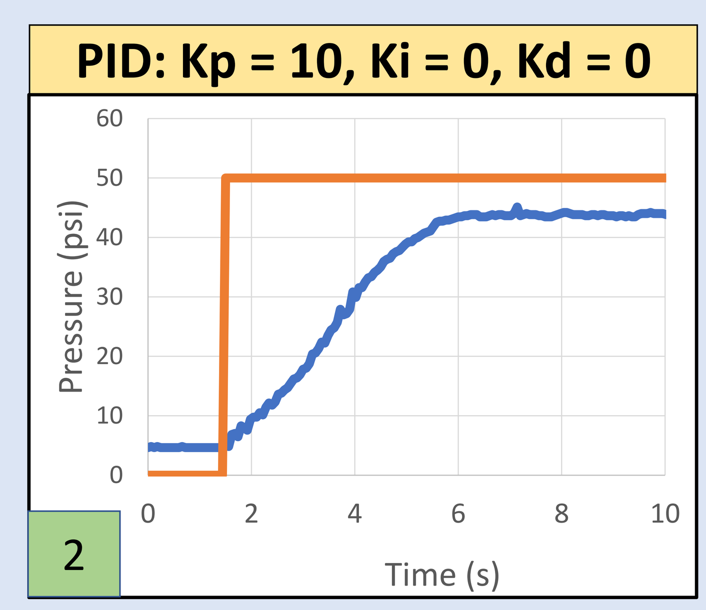
The initial method was a simple On-Off control by valves, in which the valve opens when pressure is high and closes when pressure is low. However, this caused pressure to oscillate (Figure 1) because water is incompressible, and the sampling period was too long.

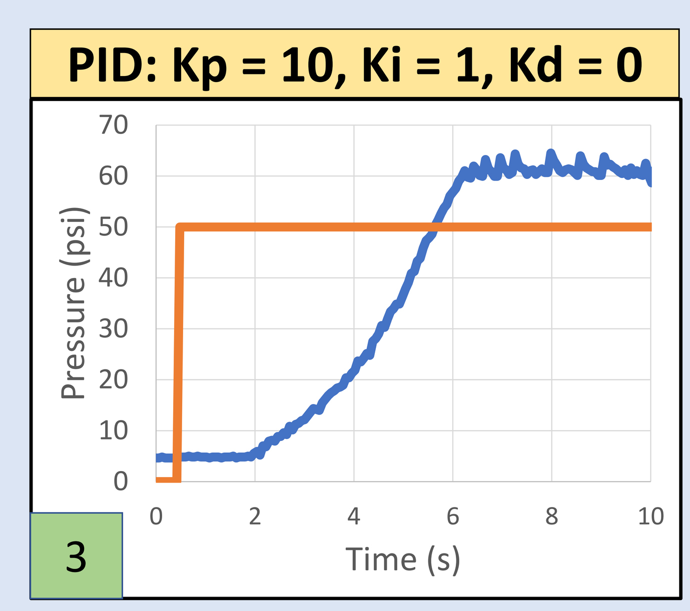
The second method added PID (proportional, integral, and derivative) control to let pump's speed regulate pressure. The concept of the PID algorithm is presented below:

$$u(t) = Kp * e(t) + Ki * \int_0^t e(\tau)d\tau + Kd * \frac{de(t)}{dt}$$

where *e* is the error and *u* is the command action. With a shorter sampling period, PID is able to reduce the error from the set point while suppressing system's oscillation. In the fluid control system, the PWM value is the command action varied by error, which is the difference between set pressure and actual pressure. Tuning the PID gains to a proper range can stabilize the system. Figure 3 and 4 show how I modify the Kp, Ki, and Kd gains. I raised pressure from 0 to 50 psi, and the system was more stable than valve control but didn't reach the set pressure because the gains were too big.



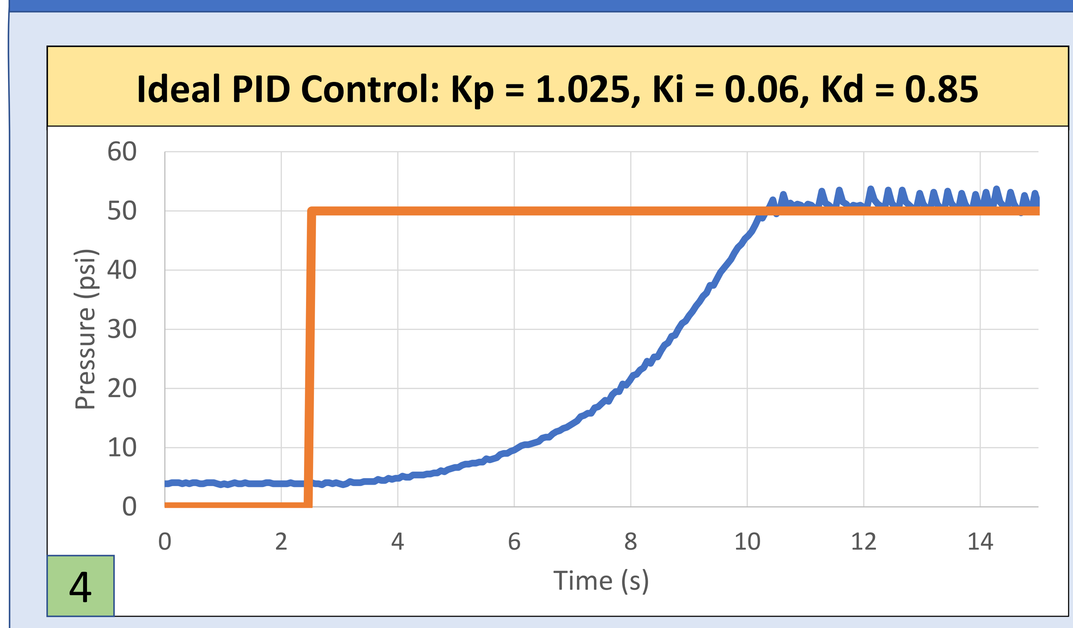




Set Pressure — Actual Pressure

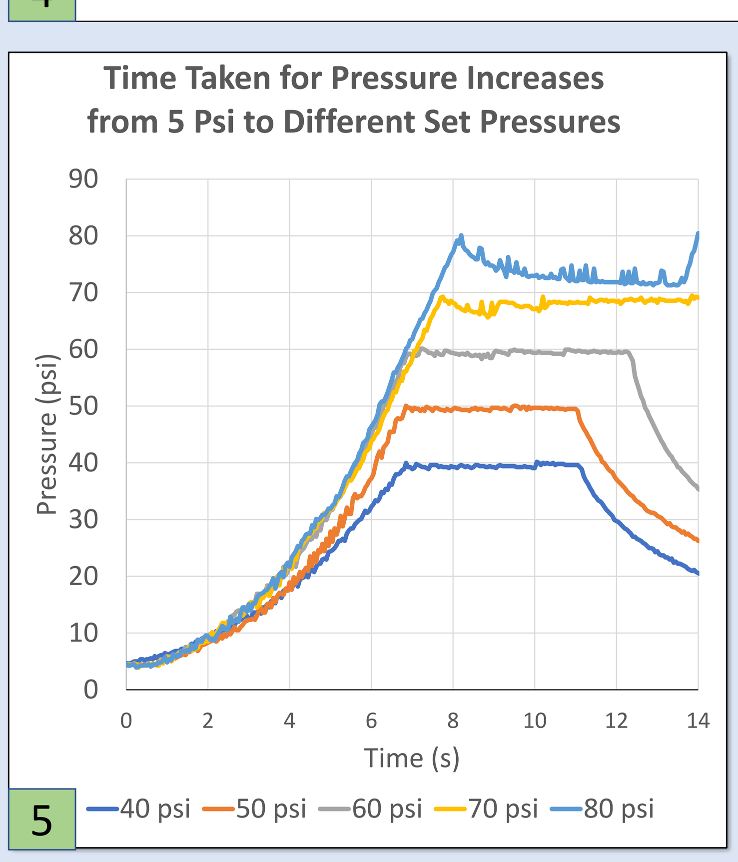
Sampling Period: 200 ms (Valve Control)
60 ms (PID Control)

Test Result



Tunning the ideal PID gains can be done by either calculation or manual testing. In this research, I manually tested each gain. As I added integral and derivative part and lower their gains, the system gradually became stable (Figure 4). However, these values are directly related to pump's performance and should be tuned again when the system uses a new pump.

After stabilizing the pressure, five system response tests were conducted to test its performance. In each test, a 36 mm, single-channel actuated textile was connected to the port, and its pressure was increased from 5 psi to five different set point. Figuer 5 shows each pressure growth, and the left the time taken from start to stable pressure state:



System's Response Time

(50 ms sampling period) :

5 to 40 psi -- 6.95 s
5 to 50 psi -- 7.05 s
5 to 60 psi – 7.45 s
5 to 70 psi -- 11.15 s
5 to 80 psi -- 13.75 s

According to Figure 5, the pressure growth is nonlinear and more like a sinuous curve. The system took much longer time and oscillated more frequently when raising pressure to 80 psi because the pump's maximum operating pressure is at 90 psi. High pressure also made my system unstable because the pump is close to its stall current and stall torque.

Conclusion

This system is actually a complicated hybrid system that consists of both continuous (PWM) and discrete variables (Valve's On-Off). PID is an efficient control method to regulate hydraulic pressure. Shortening the Arduino's sampling period and equipment's response time can also greatly improve the system's performance. However, it's difficult to use PID to control multiple actuators at different pressures since one pump cannot use different PWM values.

Next Step

For the next step, the system should react faster. We hope the pressure can increase to any set point within 3 seconds. Installing a fast-actuating valve and high-pressure pump can further shorten the system's response time. Replacing Arduino with a faster microcontroller can also shorten the processing speed. Controlling multiple actuator can be achieved by using proportional valve instead of PID, but this will also cause more expense on the component.