16-899: Sensing and Actuation Mechanisms
Lab #2 Report

Drew Marschner
11/02/2015

Introduction

This lab involved the testing and analysis of McKibben muscle open-loop control using pulsed-width modulation to control pneumatic valves.

Results

1. See Figure 1 and Figure 2 for force vs. time plots for all muscles at all duty cycles. Figure 1 displays data gathered from student-made pneumatic muscles, and Figure 2 displays data gathered from TA-made pneumatic muscles.

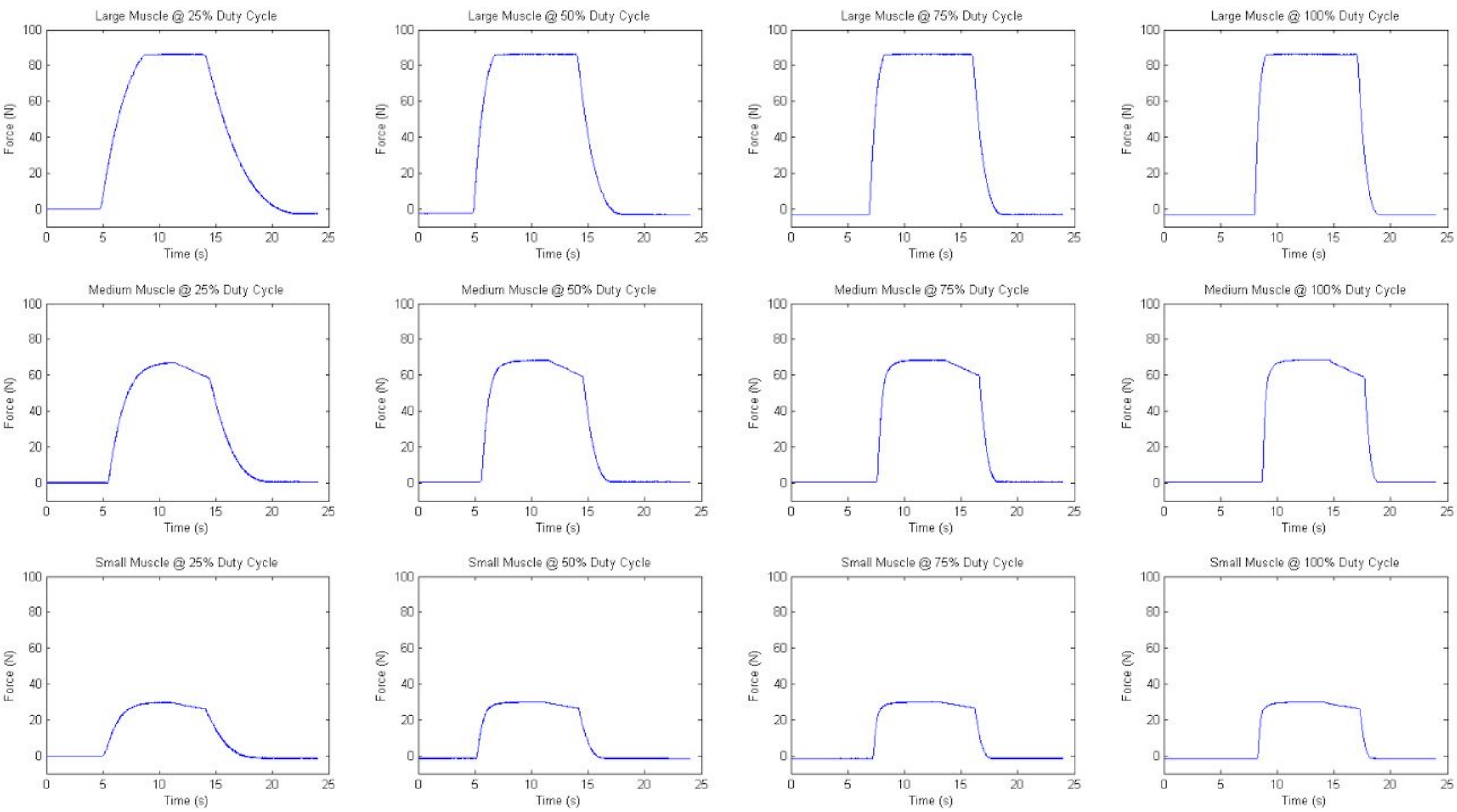

Figure 1: Force response of large, medium, and small student-made pneumatic muscles at 25%, 50%, 75%, and 100% duty cycles. Note the declining holding force of the medium and small muscles, indicating an air leak.

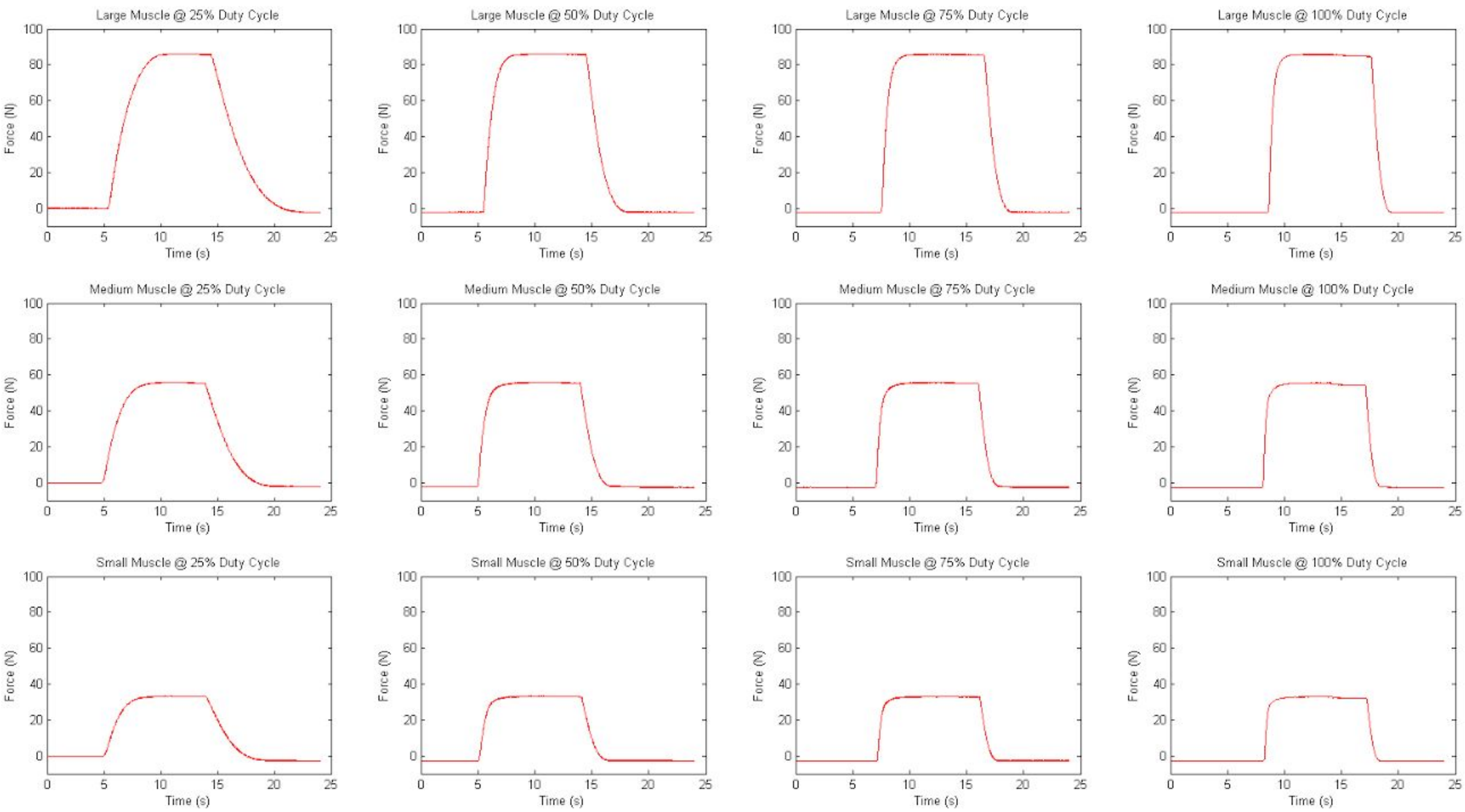

Figure 2: Force response of large, medium, and small TA-made pneumatic muscles at 25%, 50%, 75%, and 100% duty cycles. Note the steady holding force as compared to student-made pneumatic muscles.

2. The PWM frequency used was 30.64 Hz.
3. Rise times using the student-made muscles were inconsistent due to air leakage. This can be seen in Figure 1, where the “Holding” state of medium and small muscles displays a marked downward slope. In order to better assess rise and fall times the TA-made muscles were used.

Table 1: Rise/Fall times for all each muscle at each duty cycle.

	25% Duty Cycle	50% Duty Cycle	75% Duty Cycle	100% Duty Cycle
Large Muscle	4.5s / 8s	3s / 4s	2.5s / 2s	2s / 2s
Medium Muscle	4.5s / 6s	3s / 3s	2.5s / 2s	2s / 2s
Small Muscle	4s / 6s	2.5s / 2s	2s / 2s	2s / 2s

4. See Figure 3 for setup photos. An Arduino microcontroller was used to provide a PWM signal to solenoid valves which allowed the muscles to relax or contract. The valves were powered by a 12V power supply, and the forces exerted by the muscles were measured by using a strain gage.

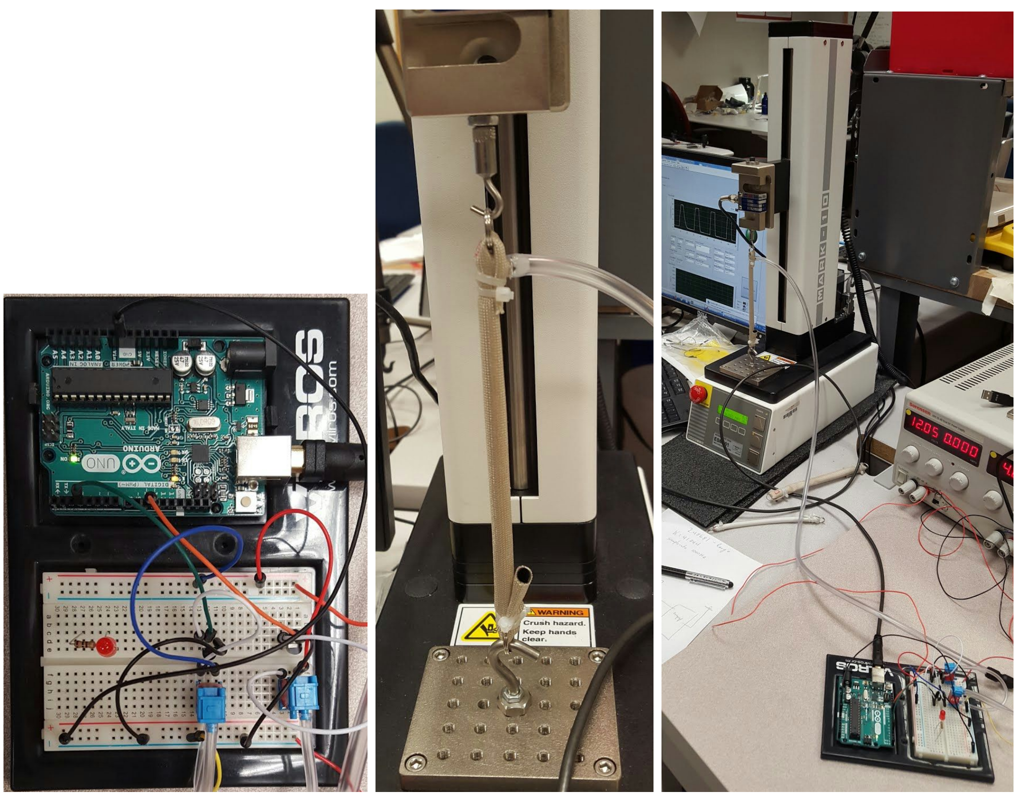

Figure 3: Setup pictures (from right): Arduino and solenoid valve wiring, Pneumatic muscle during testing, complete setup.

Discussion

1. A PWM signal to a solenoid valve at an appropriate frequency (<50Hz for this application) causes the valve to open and close rapidly, allowing air into the muscle at

some rate below the maximum rate of contraction. The maximum rate of contraction occurs very quickly, when a valve is completely opened with no PWM cycling.

2. PWM allows the rate of muscle contraction to be controlled (to a point), which makes it easier to control the level of contraction. See Figure 1 and Figure 2, where the force increase at the lower duty cycles is noticeably slower, and thus more controllable. The disadvantages of using PWM is the added code complexity, frequency limitations, and potential for valve damage (constant cycling will result in faster wear).

3. In electromechanical actuators, gearing can be used in place of PWM to decrease motor speed. Using different levels of voltage can also create similar behavior, by providing different levels of power to an actuator such as the pneumatic actuators used in this lab.

4. The PWM frequency needed to be adjusted to below 50Hz, which was stated to be the maximum frequency the solenoid could operate at. What this means is that the valve could open/close 50 times/sec at maximum speed. Any signals sent to the valve at a higher frequency would result in non-functional behavior.

5. The PWM frequency used was 30.64 Hz. This frequency was used because it was below 50Hz, which was the maximum frequency of the solenoid valve as stated by its data sheet, and was the lowest possible frequency to generate a PWM signal from an Arduino's PWM pins using the `analogWrite()` library function.

Appendix

Collected data can be found in `lab2.m` and plotted using the MATLAB script `plot_data.m`. The Arduino code used to control the solenoid valves can be found in the folder /`valve_pwm`.