2018

# Diginol: A Joule-Heated Binary-Partitioned Nitinol Actuator

Martin Dawson Holland
*University of Alabama at Birmingham*

# DIGINOL: A JOULE-HEATED BINARY-PARTITIONED NITINOL ACTUATOR

by

MARTIN DAWSON HOLLAND, I

DOUGLAS H. ROSS, COMMITTEE CHAIR
DALTON S. NELSON
DANEESH SIMIEN

A THESIS

Submitted to the graduate faculty of The University of Alabama at Birmingham,
in partial fulfillment of the requirements for the degree of
Master of Science

BIRMINGHAM, ALABAMA

2018

DIGINOL: A JOULE-HEATED BINARY-PARTITIONED NITINOL ACTUATOR

MARTIN DAWSON HOLLAND, I

MECHANICAL ENGINEERING

ABSTRACT

New and innovative robotic designs are always needed in both research and commercial fields. To that end, new and innovative actuators are needed as well. In this paper, a digitally controlled Nitinol actuator, Diginol, is proposed. Diginol is a Nitinol actuator that uses Joule heating to individually activate partitions of a length of Nitinol. The lengths of the partitions follow a base-2 pattern, minimizing the number of partitions required for a given number of actuation lengths. Activating separate partitions instead of the entire length of Nitinol circumvents the need to account for the hysteresis and nonlinearities inherent in the transformation of Nitinol.

KEY WORDS: Robotics, Actuator, Nitinol, Shape Memory, Joule Heating, Binary

TABLE OF CONTENTS

*Page*

LIST OF TABLES

*Table* *Page*

LIST OF FIGURES

*Figure* *Page*

# BACKGROUND

In this section, the general usefulness of robots to society will be discussed. Next, a brief explanation of the most common types of actuators will be given. Then, shape memory alloys (SMAs) and the SMA Nitinol and its use as an actuator will be discussed in more detail.

Robots have become an increasingly integral facet of modern society. With applications in areas such as surgery, aerospace, maintenance, inspections in hazardous spaces, manufacturing, nuclear decontamination, materials handling, agricultural harvesting, robotic refueling, service and painting, defense and rescue, and robotic prosthetics, robots exist everywhere in our day to day lives and are the next step in the technological evolution of mankind [1-9]. Three ways of categorizing robots: (1) the method of control either "waldo" (directly controlled by humans) or autonomous, (2) the type of material they are constructed from either hard (the robots components are rigid and do not deform when under normal use and loading) or soft (the robot's components are inherently compliant and undergo elastic deformation under normal use and loading), and (3) how flexible they are in reaching a location which would be discrete, hyper-redundant, or continuum [6, 10]. In this work, only the differences between discrete, hyper-redundant, and continuum will be discussed.

Discrete robots are what most people think of when they hear "robot." A discrete robot is any robot that is composed of a series of rigid links connected by single degree of freedom (DOF) joints [10]. Discrete robots are hard robots and are generally fully-

actuated. This means that the robot has the same number of actuators (motors, solenoids, pistons, etc.) as it has DOF. The design and control of discrete robots tends to be straight-forward, and discrete robots excel in applications where speed and accuracy are desired [10]. When a position feedback sensor is attached at each of the robot's joints, and the dimensions of all the robot's links are known, the relative positions of each of the robot's links is calculable [6]. This ability of a robot to know its own configuration is called proprioception and is very useful when creating a control scheme for a robot that needs to perform a task autonomously. However, the rigid-link design of discrete robots means that they cannot conform to their environment, they cannot navigate non-predefined paths, and are generally only suited to the task for which they were originally designed. In short, discrete robots are ill-suited for unknown, shifting, and/or organic environments.

Hyper-redundant robots are similar to discrete robots in that they are comprised of rigid links connected by discrete joints, however, hyper-redundant robots utilize a high density of repetitive links and joints to produce robots that are capable of smoother, more organic configurations and motion [10]. Hyper-redundant robots are much more maneuverable than discrete robots and are therefore better suited to navigating complex environments and adopting complex configurations. This also makes hyper-redundant robots a more adaptable design solution, that can be utilized in many different applications and environments.

Hyper-redundant robots can also be equipped for proprioception since they are comprised of rigid links and discrete joints. So long as each joint has a sensor and the dimensions of the robot's parts are known, its configuration can be determined. The major drawback of hyper-redundant robots is the increased complexity of control. The high

number of links, joint, and actuators makes modeling and controlling hyper-redundant robots much more intensive than their more straight-forward discrete counterparts.

Continuum robots are the most flexible of the three categories. Continuum robots do not contain rigid links or discrete joints and therefore have the potential for extreme flexibility and compliance [10]. Continuum robots are most commonly comprised of a flexible spine that can bend continuously along its length [5]. The most common actuation method for continuum robots involves running cables (sometimes called tendons) up the length of the robot and attaching them to the distal end [2, 3]. The cables are then controlled by actuators at the base, which apply tension to the cables causing the robot to bend in the direction of the applied tension [2, 3]. Continuum robots are the only robots that can be comprised solely of soft materials [6]. As such, continuum robots that are almost entirely comprised of soft actuators have been developed, such as Colobot, the OctArm VI, and a robot made of a trio of SPAMs (series pneumatic artificial muscles) surrounding a pneumatic backbone [6, 11, 12]. Continuum robots can also be made of a series of superelastic concentric tubes with predefined bends in them [3, 13]. The tubes can be rotated inside one another and can slide in and out of one another, telescopically extending or contracting the robot [3, 13]. The precurved bends of each tube either cancel each other out, producing a net curvature of zero or add together, producing curvature in the robot [3, 13]. These tubes can be rotated and extended in order to achieve a variety of positions and configurations based on the dimensions of the precurves [3, 13]. A continuum robot's high level of flexibility makes it an attractive choice for many applications where flexibility is key, such as minimally invasive surgeries, multi-purpose grasping by way of coiling around an object, and feeding lines, cables or other equipment

through narrow, winding pipes and tunnels [2-7, 14]. The high flexibility of and lack of joints in continuum robots, especially soft robots, means that they are generally under-actuated, having more DOF than actuators [6]. This means that, aside from robots that use soft actuators such as electroactive polymers (EAPs) as a combined actuator and structure, modeling, tracking, and proprioception of continuum robots can be difficult [5, 7, 10, 13, 15, 16]. Another drawback of many continuum robots is the need for a base at one end to contain the robot's actuators. This actuator base, whether it contains motors, air compressors, or simply handles, lowers the portability, maneuverability, and autonomy of the robot.

Regardless of the design or function, every robot has to move. To that end, a myriad of actuators have been utilized to facilitate movement for all shapes and sizes of robot in all kinds of environments.

The most common type of actuator is a motor. Motors are ubiquitous, powerful, and efficient rotary actuators, with efficiencies as high as 98%. However, motors are comparatively bulky and can be loud when operating. Additionally, as motors are scaled down, their efficiency drops off quickly. A 2 gram motor typically has an efficiency of no more than 50%, and the efficiency gets worse as scale decreases [17].

PAMs, or Pneumatic Artificial Muscles, are linear actuators that utilize air pressure to provide motion and power [6]. PAMs typically take the form of a rubber tube reinforced with a layer of woven fabric [6]. When the rubber tube inside is pressurized, the fabric layer expands, creating the actuating motion [6]. A PAM can either extend or contract depending on the angle of the fibers in the woven layer [6]. A series of extending and contracting PAMs can be used to control a system just like muscles do in the human

body. PAMs are excellent for reproducing muscle-like motion, but they are difficult to scale down, have a relatively short fatigue life (10,000 cycles), and always require a source of compressed fluid for actuation [6]. These limitations make PAMs unfit for many small, and highly mobile applications.

Piezoelectrics are unique actuators that convert electric potential directly into mechanical strain. The most common piezoelectric actuators are made of quartz, though tourmaline, topaz, cane sugar, and Rochelle salt also produce this effect, and some piezoelectric polymers do exist [7, 18]. The atomic structure of these materials gives them their unique properties wherein a voltage applied across the material induces a net dipole in the material's molecules causing them to deform [18]. Piezoelectric quartz crystals are quiet, efficient, Magnetic Resonance Imaging (MRI) compatible, and can be actuated longitudinally or transversely depending on how the voltage is applied [4, 18]. Their biggest drawback, however, is that the strain that is produced by these crystals is minuscule [18]. To get any reasonable amount of actuating motion out of them, piezoelectric crystals have to be stacked up tens or hundreds of times to magnify the effect, arranged in a way that utilizes their actuation in a reciprocating fashion to produce larger movements, or actuated in conjunction with a lever arm in order to magnify the actuation stroke. Linear and rotary actuators have been created using piezoelectrics, and while they produce little noise and heat and can be scaled down to the micron level without losing efficiency, piezoelectric actuators on their own produce very small actuation strokes are more expensive than other actuation methods [17, 18].

Shape Memory Alloys, or SMA's, are another set of materials that convert electricity into strain. SMA's are a special group of metals that display the shape memory

effect (SME), wherein a material that has been deformed, when heated, returns to its original shape [19-21]. SMAs have four important temperatures that control the SME: the martensite start temperature, $M_s$, the martensite finish temperature, $M_f$, the austenite start temperature, $A_s$, and the austenite finish temperature, $A_f$ [19-21]. These temperatures are different for each type of SMA and are affected by the constituents of the SMA. When held at a temperature below $M_f$, an SMA is in the twinned martensite phase (Figure 1) and is the most pliable, having a yield strength of 10,000 to 20,000 psi [19-21].

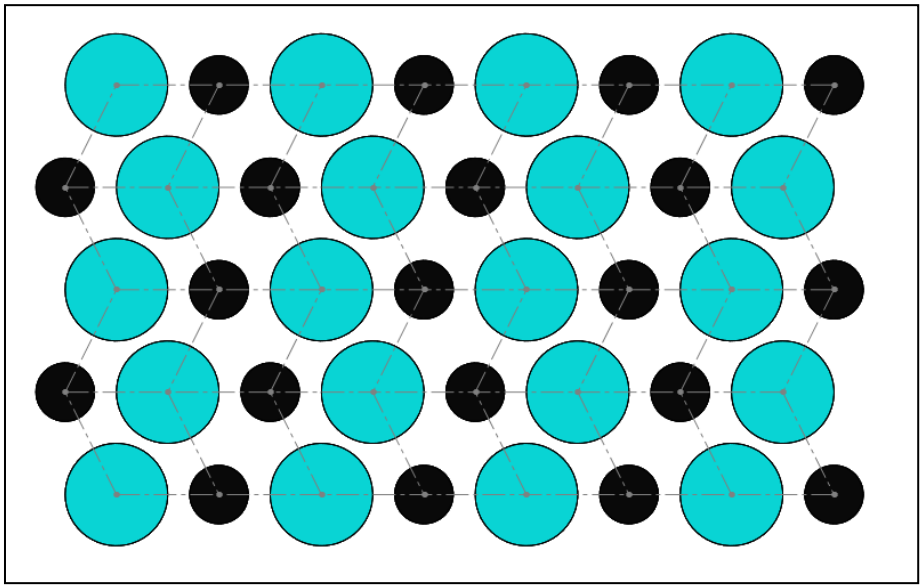

Figure 1. Twinned Martensite Atomic Structure (Titanium in Blue, Nickel in Black).

When an SMA held below $M_f$ is deformed, the region of deformation transforms into the detwinned martensite phase (Figure 2) [19-21].

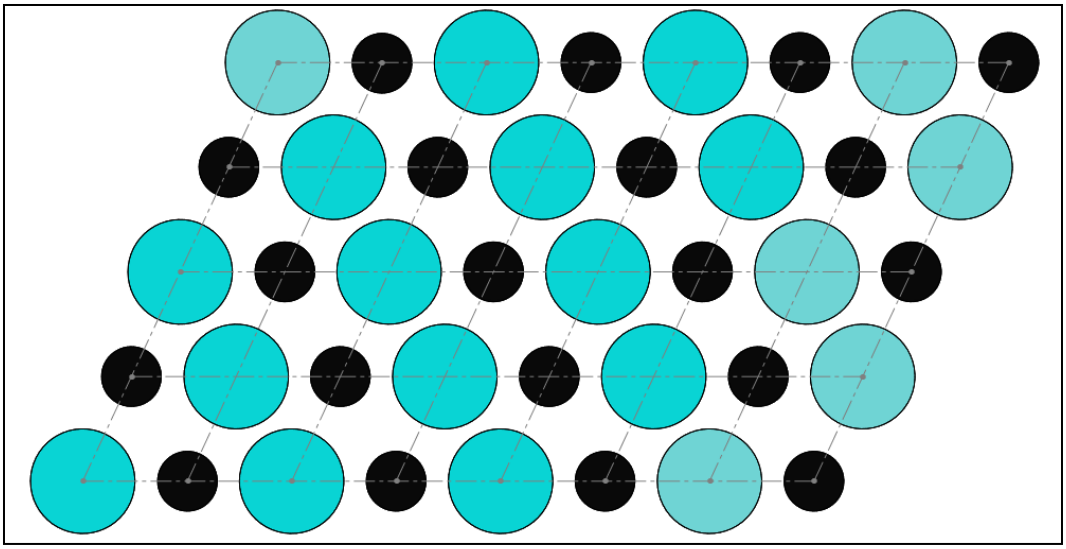

Figure 2. Detwinned Martensite Atomic Structure (Titanium in Blue, Nickel in Black).

When SMA in either martensitic phase is heated to $A_s$, it begins to transform into the much more rigid austenite phase (Figure 3) and, if it has been strained, begins to recover up to 10% of the strain that the material received below $A_s$ [20-22].

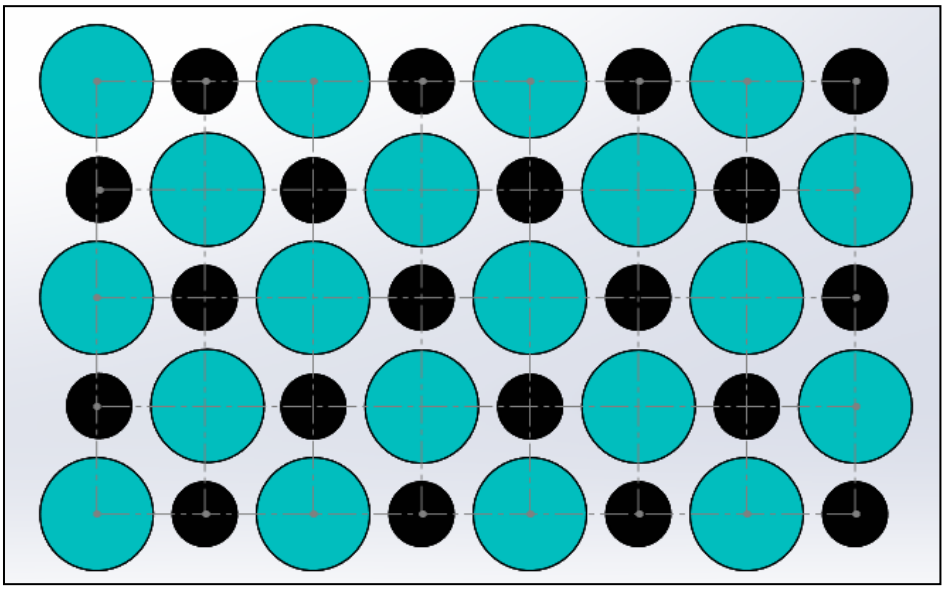

Figure 3. Austenite Atomic Structure (Titanium in Blue, Nickel in Black).

When the material reaches $A_f$, it has completely changed to austenite, which has a yield strength of 35,000 to 100,000 psi, and has fully recovered the deformation caused

below $A_s$ [19-21]. When an SMA at or above $A_f$ is cooled to $M_s$, it begins to transform back into martensite. If the load on the material has been removed when it reaches $M_f$, it will have fully transformed into the twinned martensite phase once more [20, 21]. Once in the twinned martensite phase again, the material will not return to its deformed shape until loading is applied to it [22]. The shape memory effect is possible because deformation of the twinned martensite phase causes reversible changes to the crystal structure of the SMA [23]. The crystals, when deformed, stretch and take on an acicular, or needle-like, shape[23]. The entire SME process is shown in Figures 3 and 4 on page 4 of "Shape Memory Alloys: A State of the Art Review" [19]. The SME is used, most commonly with the use of joule heating and some form of restorative force, to create linear or rotary actuators [19-22, 24, 25]. SMAs have roughly 100 times the energy density of piezoelectrics, putting out more than 100 Joules per cubic centimeter and, at maximum, produce over 500 times the stress of human muscle fibers, producing over 200MPa [26]. SMA actuators are suited to many applications due to their scalability, biocompatibility, corrosion resistance, wear resistance, high power to mass ratio, maintainability, reliability, clean and silent actuation, and lower cost [1, 19]. SMA's are already used as actuators in aerial vehicle fin positioning, automotive lighting systems, fuel management, climate control, autofocus in smart phones, variable area fan nozzle in jet engines, controllers for hot water valves in showers, micro valves, pumps, miniature grippers, and robotics [19-21, 24]. SMA actuators, being controlled by temperature change, can fail if exposed to extreme temperatures. In environments where the ambient temperature is far above or far below the normal operating temperatures of the SMA, the

actuator may be completely inoperable due to the actuator's inability to heat or cool the Nitinol to the temperatures required for the phase change that creates the actuation.

One SMA of particular note is Nitinol. Nitinol is a near equiatomic alloy of Nickel and Titanium first discovered in 1963 at the US Naval Ordinance Laboratory [21, 23]. Nitinol achieves its SME by way of a specific type of reversible martensitic phase transformation of the material's crystal structure [27]. When Nitinol is strained, martensitic variants that are oriented in the direction of the strain grow, and those oriented in other directions shrink [27]. Phase changing in this way causes very little permanent damage to the structure [27]. When the load is removed and the material is heated above $A_f$ then cooled back to $M_f$, the structure returns to a state of equally distributed martensitic variants [27]. This thermally-activated shape memory is possible because the path that the atoms take during the transformation from Martensite to Austenite is the path of least free energy [28]. This least-free-energy path takes the atoms to exactly the orientation they were in before the deformation occurred [28]. During the strain-recovery process in Nitinol, the crystals revert to their original orientation in reverse order [29]. In Nitinol, the austenitic phase takes on a B2, or body centered cubic (BCC). structure, and the martensitic phase takes on a B19', or monoclinic, phase [30]. Monoclinic refers to a geometric solid that resembles a skewed rectangular prism. All three of the axes of a monoclinic polyhedron are unequal in length, and two of the axes are orthogonal while the third is not. Additionally, the strengths of the different phases are directly tied to the size of their microstructures, with smaller microstructures being stronger than larger ones [28]. This in turn means that austenite, having a closer-packed structure, has higher strength than Martensite. Nitinol is one of the most commonly used

SMA's for actuation due to its low cycle fatigue, high corrosion resistance, stability, biocompatibility, thermo-mechanical performance, quiet actuation, and scalability [21, 24]. Nitinol's biocompatibility is due to the strong bond between the Nickel and Titanium that make it up, as well as a Titanium Oxide ($TiO_2$) layer that is applied to the majority of commercially produced Nitinol. The $TiO_2$ acts as a physical and chemical barrier to Nickel leakage [20]. Nitinol's biocompatibility makes it safe for medical applications such as orthodontic hardware and surgical robots and for everyday applications like cameras without the worry of heavy metal poisoning [19, 20]. Nitinol, on average can restore up to 8% strain upon heating [23, 31]. Most applications for pure Nitinol are limited to under 100°C, however, Nitinol can be alloyed with Palladium and Platinum to produce what are known as high temperature shape memory alloys (HTSMAs) that can operate anywhere from 100°C to 300°C [1, 21]. The lower-temperature formulations are more common in applications where normal human body temperature is sufficient to fully transform the nitinol, such as for use as stents in blood vessels, artery joints, surgical staples, and medical drug delivery systems [20, 21]. High Temperature formulations are more common in mechanical or industrial applications where the working temperatures are much higher [20, 21].

The goal of this work is to create an actuator that uses the SME of Nitinol in a digital fashion, the idea being that discrete partitions of a length of nitinol can be fully transformed independently of one another using Joule heating. This will be accomplished by partitioning a length of Nitinol into lengths following a base 2 pattern. This means if the smallest partition of the Nitinol has a length of one unit, the next partition will have a length of two, the next will have a length of four, the next eight, and so on. At the

intersection of any two partitions, electrical leads will be added to provide power for joule heating. By heating different combinations of partitions (as laid out in Table 1), different amounts of actuation are possible without the need to keep any part of the Nitinol only partially transformed. This means that, effectively, every part of the Nitinol will be either 100% martensite or 100% austenite, removing the need for the constant slight adjustments that are necessary to keep analog actuators at a certain position. In fact, the partitions, when activated, could each be over-heated to ensure that the actuation length remains constant regardless of changes in ambient temperature or load conditions. This type of actuator could be used in any application where small, linear actuators with high power-to-weight ratios are needed.

Table 1

*Activation Groups for Actuation Testing*

| Partition 1<br>1 unit long | Partition 2<br>2 units long | Partition 3<br>4 units long | Partition 4<br>8 units long | Closed Switches | Total Actuation Length (units) |
|---|---|---|---|---|---|
| Inactive | Inactive | Inactive | Inactive | NONE | 0 |
| Active | Inactive | Inactive | Inactive | V1;G2 | 1 |
| Inactive | Active | Inactive | Inactive | V2;G3 | 2 |
| Active | Active | Inactive | Inactive | V1;G3 | 3 |
| Inactive | Inactive | Active | Inactive | V3;G4 | 4 |
| Active | Inactive | Active | Inactive | V1;G2;G3;V4 | 5 |
| Inactive | Active | Active | Inactive | V2;G4 | 6 |
| Active | Active | Active | Inactive | V1;G4 | 7 |
| Inactive | Inactive | Inactive | Active | V4;G5 | 8 |
| Active | Inactive | Inactive | Active | V1;G2;G4;V5 | 9 |
| Inactive | Active | Inactive | Active | V2;G3;G4;V5 | 10 |
| Active | Active | Inactive | Active | V1;G3;G4;V5 | 11 |
| Inactive | Inactive | Active | Active | V3;G5 | 12 |
| Active | Inactive | Active | Active | V1;G2;G3;V5 | 13 |
| Inactive | Active | Active | Active | V2;G5 | 14 |
| Active | Active | Active | Active | V1;G5 | 15 |

As a proof of concept that two parts of the same length of Nitinol can be activated independently in order to produce two distinct and reproducible actuations, an experiment was devised. In this experiment, experimental setup 1, a length of 0.25mm diameter body temp Nitinol was connected between a TAL220 10kg load cell and a short length of Spider Wire (Figure 4) [32-35].

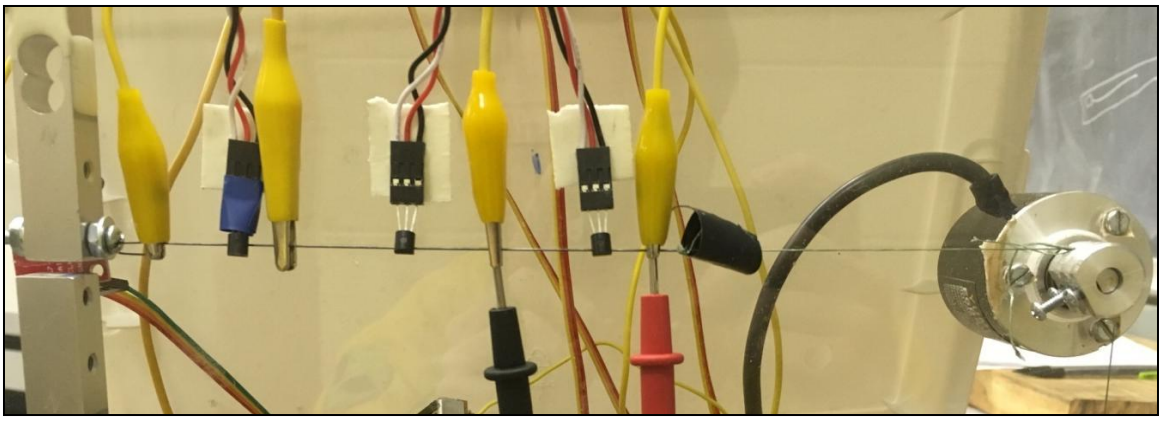

Figure 4. Picture of Experimental Setup 1.

The spider wire was then run over a pulley and tied to a five pound weight which acted as a restoring force for the Nitinol. The pulley was attached to a 1024 P/R (Quadrature) rotary encoder which fed count data, through an Arduino Uno, back to a computer [36, 37]. Temperature sensors and electrical leads (yellow alligator clips) were also attached to the Nitinol (Figure 4) [38]. The electrical leads were attached at roughly 25.4mm intervals. The exact length of the middle interval was not as critical as no current was expected to flow in that region (Figure 4). The electrical leads were attached such that the ground leads were next to each other near the middle of the Nitinol (Figure 4). Putting the ground leads next to each other in the middle ensures that the electricity flowing through one partition will not flow through the other. The leads were connected to a 12 volt power source through an array of N and P-channel MOSFETs. The

temperature sensors were connected to a computer through an Arduino Uno. The Arduino Uno used pulse width modulation to control the outgoing voltage through the MOSFETs. The Arduino Uno was also used to interpret incoming temperature sensor data. The array of N and P-channel MOSFETs was used to supply power to the Nitinol at a higher voltage and amperage than the Arduino was capable of on its own. Voltage was fed to each partition one at a time, and data from the encoder was recorded. Both partitions produced similar contractions. Then both partitions were powered simultaneously which, as predicted, produced double the contraction that each partition had produced on its own. Then one partition was allowed to cool to room temperature, whereupon the Nitinol returned to its length for only a single active partition. Experimental Setup 1 showed that it is possible to actuate discrete partitions of a Nitinol wire using Joule heating of partitions and produce predictable and consistent actuation. A wiring diagram for experimental setup 1 can be found in Figure 19 in the Appendix.

# LITERATURE REVIEW

Nitinol is already being used as an analog actuator in many applications such as plane rudder control, robotic arms, and small locomoting robots [6, 20]. The control schemes for these actuators are generally similar to the control scheme described by Lawson 2016 [21]. Typically, the Nitinol is controlled using Joule heating, though some have used thermoelectric units to allow for active cooling [21, 25, 26, 39, 40]. The advantages of controlling Nitinol in an analog fashion instead of a digital fashion are that analog actuators tend to involve fewer components and control loops and that analog Nitinol actuators have almost infinite resolution. The drawback of analog control is that the single control loop has to be sufficiently complex to account for the nonlinearities and hysteresis present in the phase transformation which is the source of Nitinol's actuation. Additionally, an analog Nitinol actuator, being dependent on the precise temperature of the Nitinol, is susceptible to changes in ambient conditions such as temperature, humidity, and loading [21].

Analog control of Nitinol has many drawbacks, and there are examples of digitally controlled Nitinol actuators designed specifically to overcome or circumvent the nonlinearities and instabilities that come from analog Nitinol control. One such non-analog Nitinol actuator was developed by a group from the Massachusetts Institute of Technology. The development of the actuator was split across four papers published by the group. The groundwork and basic initial design are presented in Selden, Cho, and Asada (2004), which discusses using a binary system for actuating Nitinol [25]. Binary,

in this case, refers to the Nitinol being in one of two states at any time during actuation, either 100% austenite or 100% martensite.

Additions to the design are presented in Cho and Asada (2004) [39], in which it is stated that the actuator is meant to ultimately control a robotic hand and as such is inspired by biological muscles. The design takes cues from the overlapping structure of human muscle fibers and the coupled nature of human hand tendons. The design uses lengths of Nitinol each activated in sections by a series of thermoelectric modules. The Nitinol and thermoelectric modules overlap in a way that mimics the coupling of tendons in the hand.

Further additions to the design are discussed in Selden, Cho, and Asada (2005) in which the actuation response time of the Nitinol actuator is improved by minimizing the effect of the hysteresis inherent to Nitinol actuation [26]. By preheating or precooling sections of Nitinol with thermoelectric modules to just below the austenite start temperature or just above the martensite start temperature, the system is able to respond much more quickly. Nitinol will not actuate at all until a critical temperature is met. Therefore, by keeping the sections of Nitinol near these critical temperatures ahead of time, the system only needs a small amount of time to add or subtract the remaining degrees necessary to activate or deactivate the Nitinol. The downside of this method is that while the Nitinol is in this liminal state, slight changes to the ambient temperature or load can cause unwanted actuation or a lack thereof. For example, when the Nitinol is heated above $A_f$ then cooled to just above $M_s$, if the ambient temperature drops, the Nitinol will cool to below $M_s$ and begin to transition to Martensite, causing a loss of actuation, and when the Nitinol is cooled below $M_f$ then heated to just below $A_s$, if the

ambient temperature increases, the Nitinol will warm to above $A_s$ and begin to transform to Austenite, causing unwanted actuation. Cho and Asada (2005) [40] introduces a multi-axis segmentation theory which reduces the number of Nitinol segments required to achieve the same amount of fidelity and resolution of the actuator. All of these designs incorporate a binary control style, where the Nitinol is either 100% austenite or 100% martensite.

The design most similar to the Diginol System is proposed in Selden, Cho, and Asada (2004) [25]. The similarities between the two include the use of Nitinol wire as an actuator, segmenting the Nitinol to achieve different actuation lengths, controlling the Nitinol in a binary fashion in which it is either 100% Martensite or 100% Austenite, and using MOSFETs as switches to control which partitions of the Nitinol are active. The two differ in that the design presented in Selden, Cho, and Asada (2004) uses thermoelectric modules to heat the Nitinol instead of the Joule heating used in the Diginol System, and the partitions of the Nitinol in Selden, Cho, and Asada (2004) are all equal-length, whereas in the Diginol System, the partition lengths follow a base 2 pattern (Equation 6). The use of Joule heating over thermoelectric modules means the Diginol System will be less bulky and will have faster heat flow to the Nitinol than the design proposed in Selden, Cho, and Asada (2004) [25], however, it also means Diginol will not be able to cool as quickly. In the design presented in Selden, Cho, and Asada (2004), the heat from the thermoelectric modules is transferred to the Nitinol by thermal gel [25]. While thermal gels have high heat transfer coefficients, using the Nitinol itself as a heat source reduces the amount of time it takes to heat the Nitinol, which lowers the response time of the system. Having each partition be twice the length of the previous one reduces the total

number of partitions while maintaining the same number of possible actuation lengths. Additionally, while it is not mentioned in Selden, Cho, and Asada (2004), it would be possible to use the cooling capabilities of the thermoelectric modules to help in eliminating any thermal bleed between sections, as sections that need to remain un-actuated could be forcibly cooled by their corresponding module. Thermal bleed occurs when heat created in one partition of the Nitinol wire is conducted through the wire into neighboring partitions. Thermal bleed will cause partial transformation and unintended actuation of the Nitinol. The use of Joule heating alone leaves the system more susceptible to thermal bleed as there is no method for cooling sections into which the excess heat escapes.

What sets the Diginol System apart from analog shape memory alloy actuators is its robustness in the face of changing ambient temperature, humidity, and load conditions. When the ambient temperature around a Nitinol actuator is increased, the heat transfer from the Nitinol due to natural convection is lessened because of the decrease in temperature gradient between the Nitinol and its surroundings. This means that the Nitinol will not return to the Martensite phase as easily, reducing the actuation potential of the system. If the ambient temperature around the actuator is decreased, the heat transfer from the Nitinol due to natural convection is increased. This means that more power will have to be used to heat the Nitinol, which reduces the efficiency of the actuator. A similar effect is created when the humidity around a Nitinol actuator is changed. If the humidity is increased, the rate of heat transfer by natural convection increases and if the humidity is lowered, the heat transfer is lowered [41]. These changes in humidity, like changes in ambient temperature affect the efficiency and efficacy of the

actuator. The phase of the Nitinol is dependent upon not only temperature but also stress applied to the material. If the Nitinol is activated and then additional load is applied that exceeds the strength of the austenite phase of Nitinol, the Nitinol can be forced out of the austenite phase and back into the detwinned Martensite phase. This constitutes a loss of actuation and a failure of the actuator. Loss of actuation by overloading and decreased response time due to changes in ambient temperature and humidity can be counteracted by increasing the temperature to which the Nitinol is heated during actuation. Therefore, changes in loading and ambient conditions during actuation can be compensated for by heating the Nitinol an amount above $A_f$ that would compensate for the extra loading and/or heat loss that the actuator experiences. Compensation by overheating in this way is impossible for analog Nitinol actuators which depend on an exact Nitinol temperature and load to maintain a certain amount of actuation. Such fluctuations in ambient conditions and load are compensated for by a feedback loop in the analog control system. These corrections happen after the change takes place, they take time to make, and they can cause inconsistent actuation. The Diginol System, however, is always 100% Austenite when activated and can, therefore, be overheated to compensate for changes in loading and ambient temperature and humidity. The Diginol System's ability to provide the same actuation every time it is used, no matter where it is used, makes it useful in any application where changes in ambient conditions are common or where consistent precision is paramount. The fact that Diginol keeps the Nitinol at either 100% Austenite or 100% Martensite also means that Diginol's control system will not need to compensate for the nonlinearities associated with Nitinol's transformation between martensite and austenite. Since the control system only needs to keep the active partitions of Nitinol

above the Austenite finish temperature, the Diginol System can be used without the need for any feedback other than temperature. The Diginol System will, however, require additional hardware compared to analog actuators in order to control the many partitions individually, which will make the Diginol system heavier and more costly than similarly-sized analog Nitinol actuators. Additionally, there will be a limit on how small the smallest partition of the system can be and still function properly.

## RESEARCH APPROACH

The goals of this research are to show that the Diginol System is capable of repeatedly producing precise actuation numerous times without the need to compensate for hysteresis and to compare the Diginol System to a similar analog actuator. To achieve this, first an analytical model of the system is made and used to show the theoretical capabilities and limitations of the system. Next, the Diginol System is physically constructed, wired to sensors and a computer and a series of tests is run to determine the efficacy of Diginol as an actuator.

The following equations comprise the analytical model used to characterize the Diginol system. This model tracks how length, stress, strain, and resistance change with temperature and microstructure phase. This model is used under the assumption that at all times the Nitinol in the system is either 100% martensite or 100% austenite and that there is no thermal bleed between partitions. Thermal bleed in the system and the affects thereof are covered in more detail in "Preliminary Thermal and Stress Analysis of an Electrical Connection Acting as a Thermal Dam on a Nitinol Wire" by Balfour [42].

$$L_{total} = \frac{\Psi \cdot \pi \cdot r \cdot 2^{B+1}}{\chi \cdot m_G \cdot E} \quad (1)$$

$L_{total}$ - Total length of nitinol actuator wire [m]

$\Psi$ - Number of counts desired per bit of actuation []

r - Radius of pulley or shaft nitinol is attached to [m]

B - Number of bits in the actuator []

$\chi$ - Number of counts per revolution of encoder []

$m_G$ - Gear ratio between Nitinol and encoder []

E - Percent of strain that is recoverable due to transformation []

Equation 1 is based on experimental setup 2 (Figure 20 in the Appendix), which is described in the next section. Equation 1 is used to calculate how long the full length of the nitinol in the Diginol system being tested should be. It is based on how accurate the results of the experiment need to be, specifically, how many counts one unit or bit of actuation will be equivalent to.

$$R = \frac{\rho \cdot L}{A} \tag{2}$$

R - Resistance of a length of Nitinol wire [Ω]

ρ - Resistivity of Nitinol [Ω · m]

L - Length of Nitinol wire [m]

A - Cross-sectional area of Nitinol wire [$m^2$]

Equation 2 is used to calculate the resistance of a length of Nitinol. The resistance is a direct indicator of length change and, as such, is a way to double check the actuation length of the system.

$$I = \sqrt{\frac{T - T_{amb}}{4 \cdot \rho} \cdot \pi^2 \cdot d^3 \cdot h} \tag{3}$$

I - Current required to maintain steady-state temperature [amps]

T - Temperature of the Nitinol wire [K]

$T_{amb}$ - Temperature of the surroundings [K]

ρ - Resistivity of Nitinol [$\Omega \cdot m$]

h - Heat Transfer Coefficient [$W/m^2K$]

d - Wire Diameter [m]

The resistance value is used in equation 3 to determine how much current is required to maintain a constant internal temperature in a cylindrical Nitinol wire based on the ambient temperature and the heat lost due to convection.

$$\Delta L = \frac{(P/A)}{D(\xi)} \cdot L \tag{4}$$

ΔL - Change in Length of Nitinol Wire [m]

P - Load Applied to Wire [kN]

A - Cross-sectional Area of Wire [$m^2$]

$D(\xi)$ - Young's Modulus of Nitinol based on the volume fraction of Martensite [kPa]

L - Original Total Length of Nitinol Wire [m]

Equation 4 is used to determine the total length change that can be expected in a given length of Nitinol wire under a given load. This is used to determine how much preload a Nitinol wire should be put under to optimize its actuation stroke. For these calculations, the Nitinol is assumed to always either be 100% Austenite or 100% Martensite, therefore, in this work, $D(\xi)$ refers to either the modulus of Martensitic Nitinol or Austenitic Nitinol. It will be stated which when calculations are shown. In this

work, calibrations and initial testing will be done with a preload of 2.268kg on a101.6mm long, 0.127mm-diameter Nitinol wire.

$$\Delta L = \frac{\left({}^{P}/_{A}\right)}{D(\xi)} \cdot L = \frac{\left({}^{22.2411N}/_{\pi \cdot (6.35x10^{-5}m)^2}\right)}{34.5x10^{9}Pa} \cdot 0.1016m = 0.00517m$$

$$= 5.17mm$$

Assuming the wire is 100% Martensite during loading, this creates a length change of 5.17mm in the Nitinol, well below the 8% maximum strain recovery for Nitinol.

$$\Delta L = \frac{2 \cdot \pi \cdot r \cdot c}{\chi \cdot m_G} \quad (5)$$

$\Delta L$ - Change in Length of Nitinol Wire [m]

r - Radius of pulley or shaft nitinol is attached to [m]

c - Number of counts measured from the encoder []

$\chi$ - Number of counts per revolution of encoder []

$m_G$ - Gear ratio between Nitinol and encoder []

Equation 5 is used to convert the number of counts recorded by the encoder for any experimental setup into the change in length of the actuator during that test.

$$l_n = \left(\frac{L}{2^m - 1}\right) \cdot 2^{n-1} \quad (6)$$

$l_n$ - length of partition n

L - total length of wire

m - number of partitions

Equation 6 is used to calculate the lengths of the partitions in a Diginol actuator given the total length of Nitinol wire being used and the number of partitions desired. The results of the analytical model are laid out in tables 5-8 in the Appendix.

The Diginol system is set up as shown in Figure 5. Five 1.4mm wide electrical contacts are attached to a 157mm length of 0.127mm Nitinol [43]. The contacts are crimp connectors that have been bent at a 90° angle at one end and crimped directly to the Nitinol wire. A jumper wire is then simply inserted into the other end and wired into the rest of the circuit with clip leads. These connectors were chosen for their ease of application, low cost per connector, and solid connection to the Nitinol. The second contact is 7mm from the first, the third is 20mm from the first, the fourth is 47mm from the first, and the fifth is 102mm from the first contact.

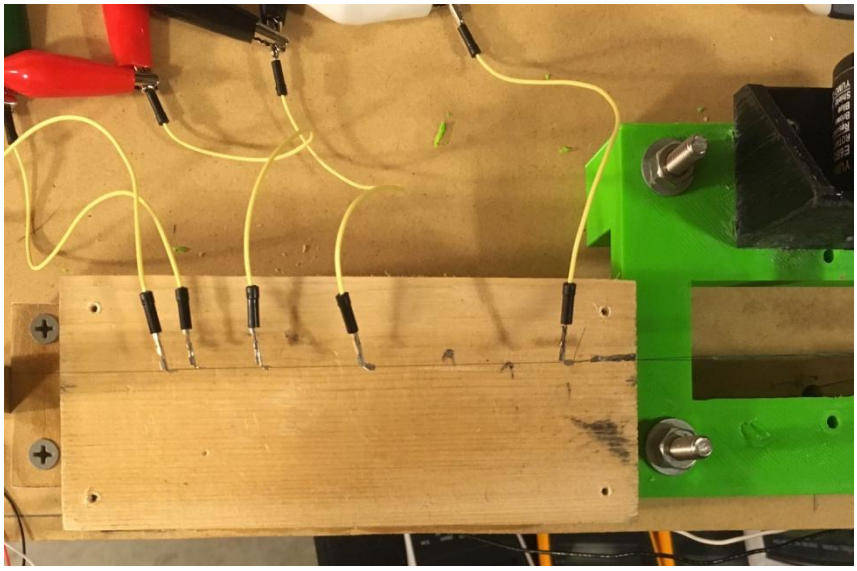

Figure 5. Overhead View of Diginol Wiring.

The 102mm length is chosen because the actuator needs to be kept as small as possible while maintaining a large enough displacement of the shortest partition that it can be accurately recorded during testing. The length of each partition is based on Equation 6. This method of partitioning maximizes the number of discrete actuation lengths per the number of control loops. Using this partitioning method, every actuation length that is a whole-number multiple of the shortest partition's actuation length can be attained via a specific combination of activated partitions. These combinations and their respective activation lengths are laid out in Table 1. If the Nitinol had partitions all of equal length, it would require $N = 2^m - 1$ individual partitions, where N is the number of equal-length partitions and m is the minimum number of partitions that can achieve the same motion as N partitions. This is to say that a system with 15 equal-length partitions is equivalent to a system with only 4 partitions assuming base 2 length partitions ($2^4 - 1 = 15$). By the same token, an actuator with equal-length partitions would need $2^8 - 1 = 255$ partitions to achieve the same number of actuation lengths as a Diginol actuator with 8 partitions. Each equal-length partition would require its own wiring and control loop to create the same number of discrete actuation lengths as the optimized partition method. Each contact is wired to an array of SLA5060 P-channel MOSFET's as shown in Figure 21 in the Appendix. The P-channel MOSFET's in the arrays act as switches controlling which partitions receive power and are controlled by an Arduino Mega using pulse-width modulation (PWM) [44]. The original design of for this experiment used an Arduino Uno, however, to control all of the MOSFET's and relays, the greater number of pins provided by the Arduino Mega was required. The MOSFET's, and by extension, the Diginol, receive power from a variable-voltage desktop power source set to supply 5V. P-

channel MOSFET's require the voltage at the gate be the threshold voltage ($V_{th}$) below the source to allow current flow from the source to the drain. This means that the signal coming to the gate from the Arduino has to be below a certain voltage for the transistor to allow current flow between the desk power supply and the system. To that end, the PWM signals sent to the MOSFET from the Arduino (which range between 0 and 255) are set to 255 to turn the MOSFET off, set to 0 to turn the MOSFET fully on, and are set anywhere in between if a voltage between 0 and 5 volts is required.. That is, if there are 5 volts at the source and the voltage at the gate is also 5 volts (255 for Arduino analogWrite command), no current will pass through the MOSFET, but if the voltage at the gate is more than 2 volts below the Source or for example 0 volts(analogWrite( 0)), the resistance in the MOSFET will be small and current can flow from the source to the drain. The MOSFET's required careful and repeated rewiring to ensure they were all activating properly. The resisters wired into the gates have to have a specific ratio of Ohms (about 10:1) for the MOSFET's to fully activate. Additionally, the first four contacts are each wired to a Tongling 5 volt relay [45]. The ground connections are made with relays instead of the N-channel MOSFET's present in the MOSFET arrays because the N-channel MOSFET's did not fully ground the connections when activated. The N-channel MOSFET's, being variable resistors, still retain some resistance even when fully activated. This not only added extra resistance to the system, but also, each N-channel had different current running through it, which created different voltage drops across them. The relays are controlled digitally by the Arduino. The relays control which contacts are grounded, as shown in Figure 21 in the Appendix. It is important that the ground leads have these switches so that the flow of electricity can be properly controlled

and contained within the desired sections. If the ground connections were all wired together without switches, the electricity would simply flow towards the nearest ground which would not be the desired path. The MOSFET arrays and relays can then be switched on and off in various combinations in order to power different sets of Nitinol partitions, allowing the Diginol system to achieve 16 different actuation lengths (including zero) with only 4 partitions. A simplified version of the wiring of the Diginol, with only the Nitinol wire, power and ground leads, and simple switches, is shown in Figure 6. For example, with 4 partitions, the shortest partition would have a unit length of 1, and the other partitions would have lengths of 2, 4, and 8. If the first partition is activated by itself, that creates an actuating motion of length 1 (Figure 7). If the second partition is activated by itself, that creates motion of length 2 (Figure 8). If the first and second partitions are activated together, that creates motion of length 3 (Figure 9). If the third partition is activated by itself, that creates motion of length 4 (Figure 10). If the first and third partitions are activated, that creates a motion of length 5 (Figure 11). If the second and third partitions are activated, that creates a motion of length 6 (Figure 12). If the first, second and third partitions are activated, that creates a motion of length 7 (Figure 13). If the first, second, and fourth partitions are activated, that creates a motion of length 11 (Figure 14). (All 16 possible activation combinations are laid out in Table 1.)

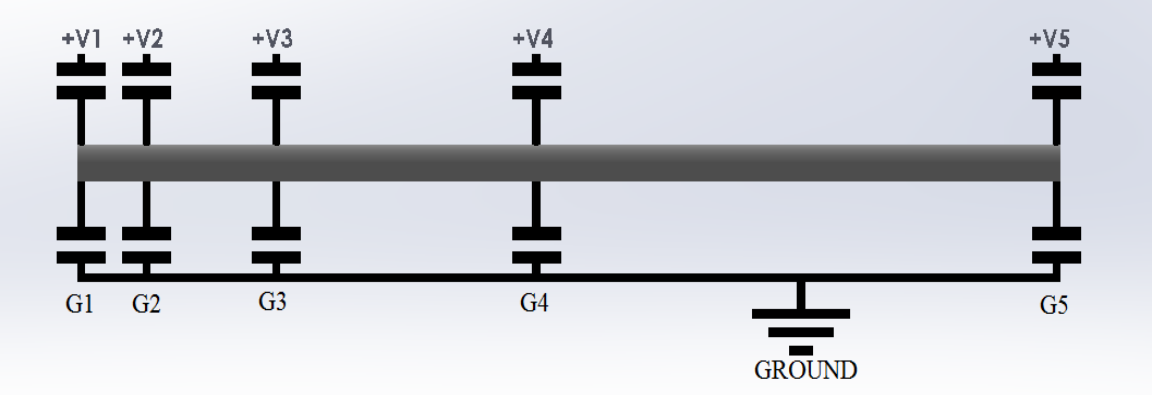


Figure 6. Schematic of Diginol Wiring Connections

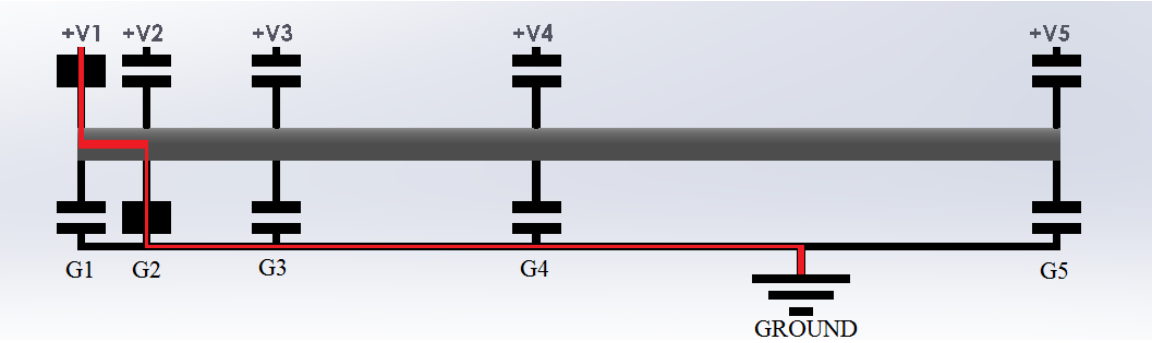


Figure 7. Schematic of electrical pathway for partition 1 being activated which creates 1 unit of motion.

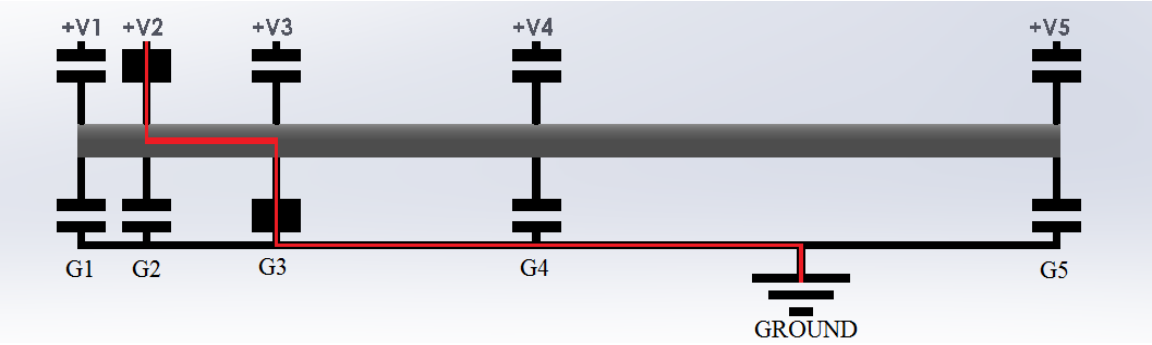


Figure 8. Schematic of electrical pathway for partition 2 being activated which creates 2 units of motion.

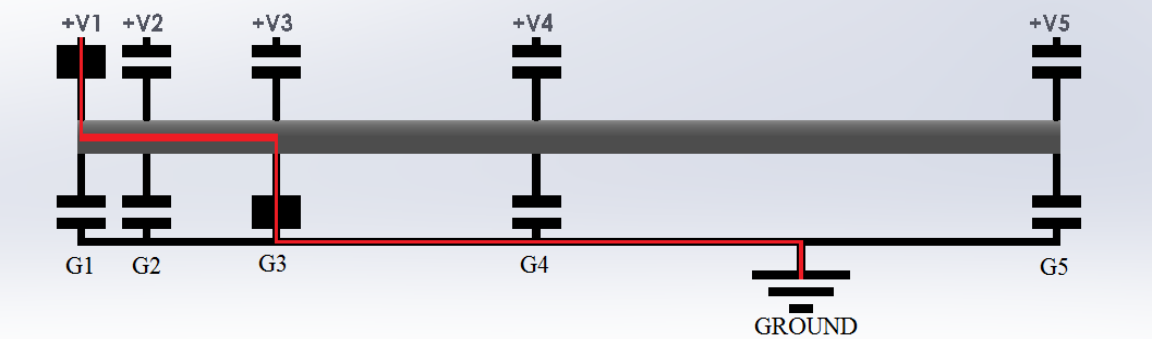


Figure 9. Schematic of electrical pathway for partitions 1 and 2 being activated which creates 3 units of motion.

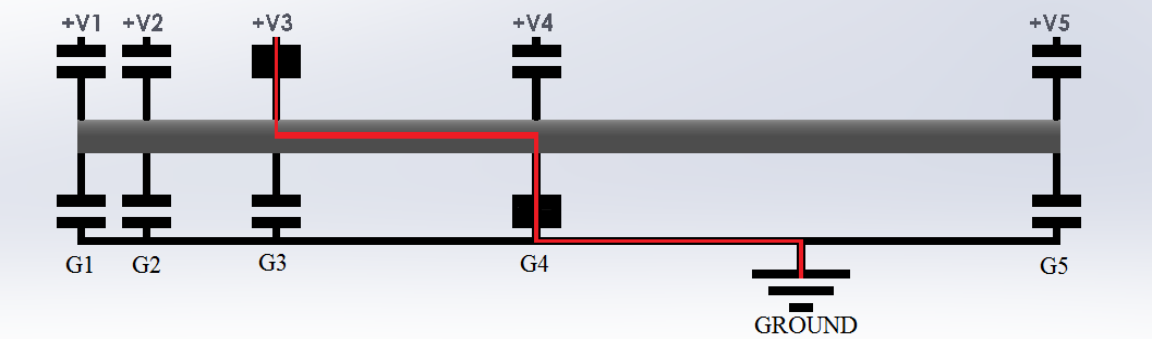


Figure 10. Schematic of electrical pathway for partition 3 being activated which creates 4 units of motion.

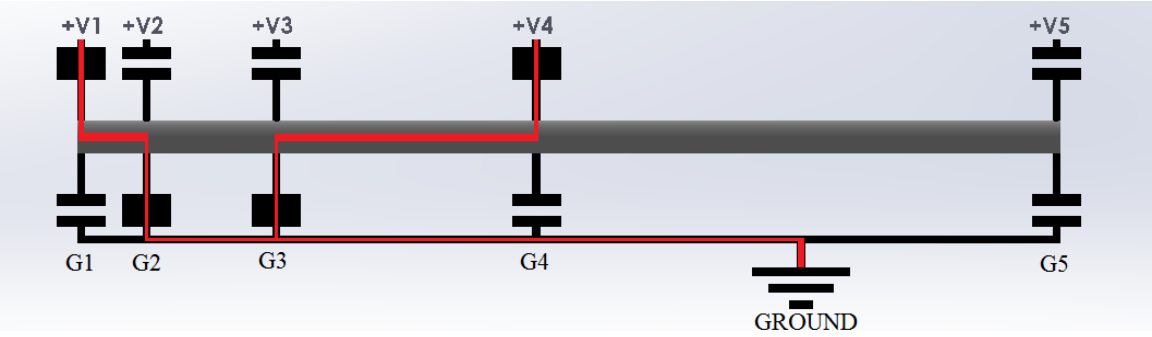


Figure 11. Schematic of electrical pathway for partitions 1 and 3 being activated which creates 5 units of motion.

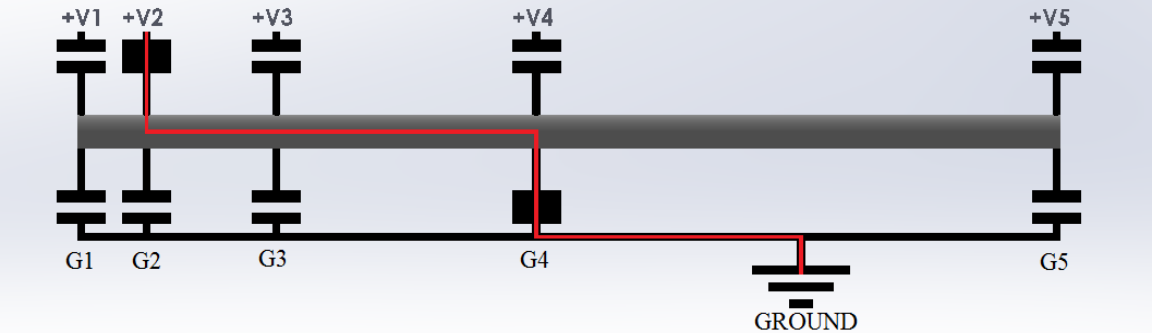


Figure 12. Schematic of electrical pathway for partitions 2 and 3 being activated which creates 6 units of motion.

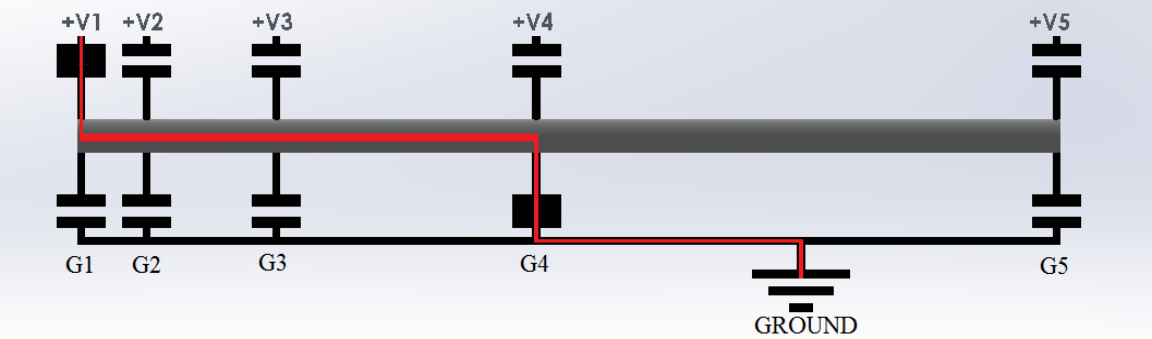


Figure 13. Schematic of electrical pathway for partitions 1, 2, and 3 being activated which creates 7 units of motion.

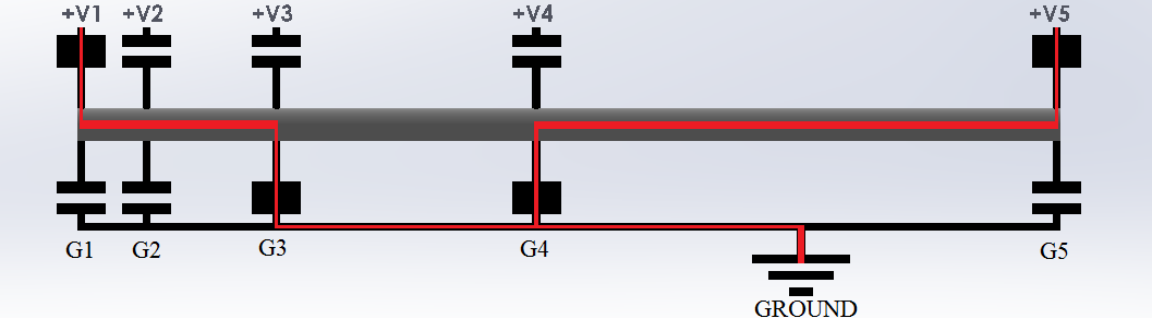


Figure 14. Schematic of electrical pathway for partition 1, 2, and 4 being activated which creates 11 units of motion.

An experiment is set up as shown in the model in Figure 15, and is used to determine how well the Diginol is able to actuate different lengths. For experimentation, a 157 mm piece of 0.127mm diameter Nitinol (transformation temperature: ~100°C) [46]

is used with an initial load of 0.8kg. The Nitinol has four partitions, the lengths of which were calculated using equation 6. These lengths can be found in Table 5 in the Appendix. The $A_f$ of the Nitinol is 100°C according to its data page on SparkFun [43]. The Diginol system is rigidly fixed to a load cell at one end and attached to a length of flexible, low deflection (Modulus ~ 1.9GPa) cord (Spider Wire) at the other end. To affix the Nitinol to the load cell, a small hole was drilled through the top of a machine screw. The Nitinol was fed through the drilled hole, then held in place with a washer and nut on the screw.

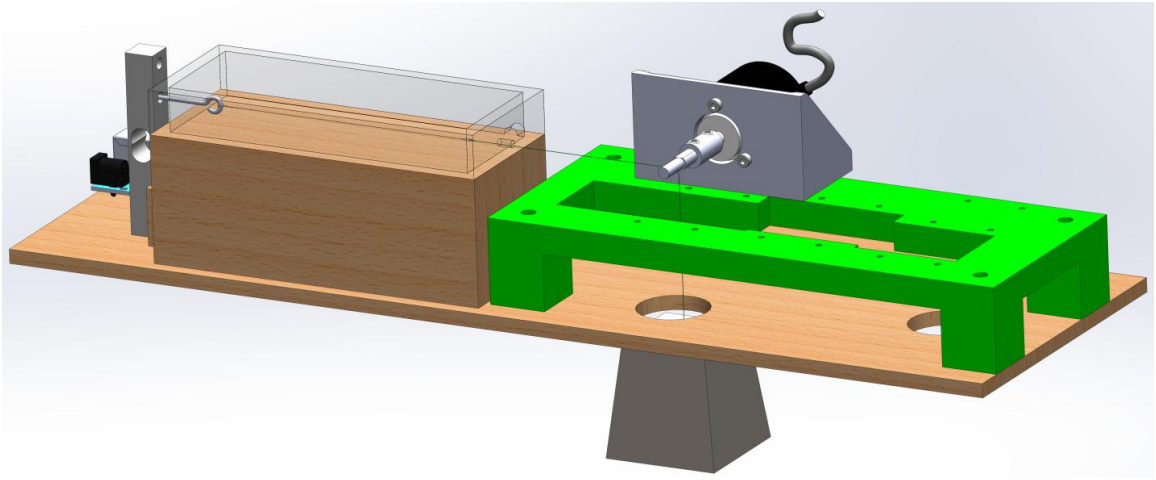

Figure 15. Isometric view of a Model of Experimental Setup 3

Then the screw with attached Nitinol was fed through one of the mounting holes in the load cell and held in place by another nut. The load cell is used to verify the load placed on the Nitinol. Control code was written for the load cell which not only allowed the load cell to pass data to the Arduino, but also was necessary to calibrate the load cell. This code can be found in the Appendix. To attach the Nitinol to the Spider Wire, first the Nitinol and Spider Wire were clamped between two washers on a short machine screw with a nut. The weight of the screw, nut, and washers, however, was found to interfere with the data gathered during testing, so a new method was devised. The new method involved tying a loop in the end of the Spider Wire, feeding the Nitinol through

the loop back onto itself, then crimping the Nitinol to itself using a crimp connector and a crimping tool. The connection is then soldered to ensure the Nitinol would not slip out of the crimped connector. The Spider Wire is looped around a 5mm diameter shaft which is attached directly to a rotary encoder with a coupler and set screws.  A small basket is attached to the end of the Spider Wire with a carabiner clip (combined weight of 110g). Then, 48 weights which are each 14.2g (0.5oz) are added to the basket for a total load of 790g. Experimental setup 3is shown in Figure 16

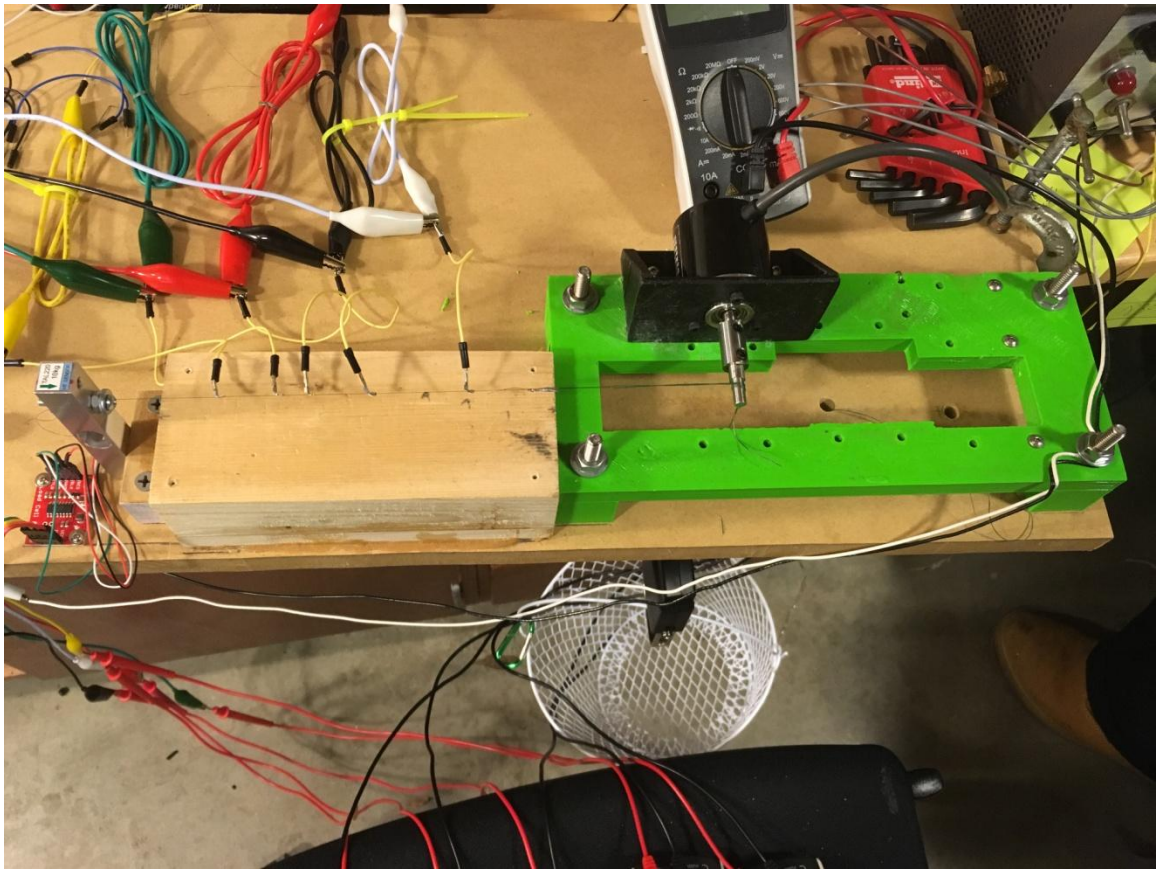

Figure 16. Experimental Setup 3.

The rotary encoder attached to the shaft is wired to the Arduino for data gathering. Control code for the encoder was written to interpret the signals from the encoder into counts. Using the known resolution of the encoder (1024 counts per

revolution) the amount of rotation is calculated. The Arduino will record the rotations of the shaft, which will be used to calculate the length change of the actuator using equation 5. An array of multi-meters are used to monitor the voltage across each partition. The control code for all of the electronics can be found in the Appendix.

Testing of the Diginol starts by zeroing the encoder, turning on the desk-top power supply, and setting the supply to 5 volts. Next, partition 1 is powered to optimal voltage based on calculations by activating the MOSFET at contact 1 and the relay at contact 2 (Figure 7). Optimal Voltage is the minimum voltage required to fully activate a given Partition of Nitinol. While activated the voltage across partition 1 and the number of counts produced are recorded. This is repeated for each of the subsequent activation combinations. Then, the power supply was increased to 6 volts, and all of the readings for each partition combination were taken again. Originally, the Experimental Setup included a gear train between the Spider Wire and the encoder that would increase the resolution of the encoder by a factor of 64. This was needed because the encoder used at the time had a resolution of 200 counts per revolution which was not enough on its own to gather any useful data. After some preliminary testing, it was discovered that the friction and backlash present in the gear train introduced irregularities and anomalies into the data, such as sudden, random jumps in the encoder count value, and each time the load was altered, the load value would change to what it was expected to be, but if left alone it would slowly drift to lower and lower values. For example, a hard stop was placed on the first shaft so that it would always return to the exact same position. The shaft was positioned at the hard stop, and the encoder was zeroed. Then the shaft was rotated through about one half of a revolution and returned to the hard stop. When the shaft

returned to the hard stop, the encoder was not at zero. This test was repeated, and every time the shaft was moved without resetting the encoder, the encoder read further and further from zero at the hard stop. Instead, a higher resolution encoder was attached to a single shaft with a coupler and set screws. In addition to the inherently higher resolution of the new encoder (1024 instead of 200 counts per revolution), the control code for the encoder was altered such that the resolution was doubled to 2048 counts per revolution, more than enough sensitivity to gather useful data. Additionally, the Experimental Setup originally included, an acrylic enclosure around the Nitinol with Peltier modules and temperature sensors inside. The Peltier modules would be used to change the ambient temperature within the enclosure to study the effects changing ambient temperature had on the system, and the temperature sensors would measure both the temperature of each section of Nitinol and the ambient temperature inside the enclosure. It was found, however that these temperature sensors were not accurate enough to provide reasonable data and the controls for the Peltier modules were never completed, as they required their own separate set of MOSFET's.

Using the same experimental setup as was described above, an experiment to characterize the stress-strain curve of the material was devised. The material was placed in the setup as above, but was left deactivated for the duration. The load cell and encoder were both zeroed, then the material was loaded, first with a small basket (110g), then with 14.2g (0.5oz) weights. After each increase in weight, encoder and load cell data was recorded. After all the data was recorded, it was converted into stresses and strains and graphed as a curve. This experiment was begun, but never completed due to time constraints. The data collected from it is laid out in Table 2

Table 2

*Preliminary Nitinol Stress-Strain Data*

| Load (lb) | Counts | Stress (MPa) | Strain (%) |
|---|---|---|---|
| 0.00 | 0.00 | 0.00 | 0.00 |
| 0.04 | 565 | 1.43 | 1.30 |
| 0.09 | 816 | 3.22 | 1.88 |
| 0.13 | 1001 | 4.65 | 2.31 |
| 0.17 | 1153 | 6.09 | 2.66 |
| 0.20 | 1259 | 7.16 | 2.90 |
| 0.24 | 1396 | 8.59 | 3.22 |
| 0.26 | 1489 | 9.31 | 3.43 |
| 0.28 | 1534 | 10.0 | 3.54 |
| 0.35 | 1788 | 12.5 | 4.12 |
| 0.39 | 1938 | 14.0 | 4.47 |
| 0.43 | 2088 | 15.4 | 4.82 |
| 0.45 | 2165 | 16.1 | 4.99 |
| 0.48 | 2263 | 17.2 | 5.22 |
| 0.59 | 2653 | 21.1 | 6.12 |
| 0.61 | 2747 | 21.8 | 6.34 |
| 0.72 | 3167 | 25.8 | 7.31 |
| 0.75 | 3309 | 26.9 | 7.63 |
| 0.77 | 3355 | 27.6 | 7.74 |
| 0.80 | 3521 | 28.6 | 8.12 |
| 0.84 | 3688 | 30.1 | 8.51 |
| 0.87 | 3805 | 31.2 | 8.78 |
| 0.89 | 3930 | 31.9 | 9.07 |
| 0.92 | 4042 | 32.9 | 9.32 |
| 0.94 | 4173 | 33.7 | 9.63 |
| 0.96 | 4255 | 34.4 | 9.82 |
| 0.98 | 4354 | 35.1 | 10.0 |
| 1.00 | 4518 | 35.8 | 10.4 |
| 1.03 | 4688 | 36.9 | 10.8 |
| 1.05 | 4781 | 37.6 | 11.0 |
| 1.08 | 4696 | 38.7 | 10.8 |
| 1.10 | 5058 | 39.4 | 11.7 |
| 1.13 | 5223 | 40.5 | 12.0 |
| 1.16 | 5404 | 41.5 | 12.5 |
| 1.18 | 5550 | 42.3 | 12.8 |
| 1.21 | 5726 | 43.3 | 13.2 |
| 1.23 | 5883 | 44.0 | 13.6 |
| 1.26 | 6049 | 45.1 | 14.0 |
| 1.28 | 6215 | 45.8 | 14.3 |
| 1.29 | 6329 | 46.2 | 14.6 |

| | | | |
|---|---|---|---|
| 1.35 | 6802 | 48.3 | 15.7 |
| 1.38 | 7011 | 49.4 | 16.2 |
| 1.43 | 7276 | 51.2 | 16.8 |
| 1.46 | 7405 | 52.3 | 17.1 |
| 1.49 | 7547 | 53.4 | 17.4 |
| 1.52 | 7702 | 54.4 | 17.8 |
| 1.54 | 7854 | 55.1 | 18.1 |
| 1.56 | 7945 | 55.9 | 18.3 |
| 1.59 | 8116 | 56.9 | 18.7 |
| 1.61 | 8257 | 57.6 | 19.0 |
| 1.71 | 8779 | 61.2 | 20.3 |
| 1.79 | 9195 | 64.1 | 21.2 |
| 1.86 | 9624 | 66.6 | 22.2 |
| 1.94 | 10052 | 69.5 | 23.2 |
| 1.99 | 10391 | 71.3 | 24.0 |
| 2.06 | 10807 | 73.8 | 24.9 |
| 2.13 | 11180 | 76.3 | 25.8 |
| 2.18 | 11463 | 78.1 | 26.4 |
| 2.61 | 13974 | 93.5 | 32.2 |
| 2.75 | 16100 | 98.5 | 37.1 |

## RESULTS

The initial load of 0.79kg caused an elongation of 6.35mm (848 counts) which is equal to 4.07% strain (Equation 4). 4.07% is less than the maximum recoverable strain (8%), which means that the deformations during testing are expected to be recoverable by the system. The voltages across each partition, the expected and experimental counts from the encoder that were produced, the expected and experimental equivalent displacement, and the % difference between the expected and the experimental displacement are laid out in Table 3.

Table 3

*Voltage, Count, and Displacement values from test 1*

| Partition Combination | Voltage (V) | Counts from Encoder | Equivalent Displacement (mm) | Expected Counts | Expected Displacement (mm) | % Difference of Displacement |
|---|---|---|---|---|---|---|
| 0 | 0.00 | -2 | 0.0E+00 | 0 | 0.000 | 0.00 |
| 1 | 0.11 | -2 | 0.0E+00 | 36 | 0.275 | 100 |
| 2 | 0.21 | -2 | 0.0E+00 | 72 | 0.550 | 100 |
| 3 | 0.34 | -1 | 7.64E-03 | 108 | 0.825 | 99.1 |
| 4 | 0.42 | -1 | 7.64E-03 | 144 | 1.10 | 99.3 |
| 5 | 0.15/0.42 | -1 | 7.64E-03 | 180 | 1.38 | 99.4 |
| 6 | 0.64 | -1 | 7.64E-03 | 216 | 1.65 | 99.5 |
| 7 | 0.75 | -1 | 7.64E-03 | 252 | 1.92 | 99.6 |
| 8 | 0.81 | -1 | 7.64E-03 | 288 | 2.20 | 99.7 |
| 9 | 0.14/0.79 | -1 | 7.64E-03 | 324 | 2.47 | 99.7 |
| 10 | 0.25/0.82 | -1 | 7.64E-03 | 360 | 2.75 | 99.7 |
| 11 | 0.35/0.79 | -1 | 7.64E-03 | 396 | 3.02 | 99.8 |
| 12 | 1.21 | -1 | 7.64E-03 | 432 | 3.30 | 99.8 |
| 13 | 0.17/1.19 | -1 | 7.64E-03 | 468 | 3.57 | 99.8 |
| 14 | 1.41 | -1 | 7.64E-03 | 504 | 3.85 | 99.8 |
| 15 | 1.50 | -1 | 7.64E-03 | 540 | 4.12 | 99.8 |

The voltages produced across the partitions are the predicted values, however, according to the calculations, given the 4.07% strain, the first partition alone should produce 36 counts of rotation from the encoder as opposed to the zero counts seen in testing. This also contradicts the calculation of 4.1% strain based on a 5lb load, and contradicts the calculation of a maximum strain of 1.2% based on maximum tensile strength. These calculations can be found in the Appendix. During experimentation, it was found that the Nitinol used in this work could not hold the initially estimated 5lb, and the maximum load was much closer to 2.5lb. The true maximum load value was never determined due to a shortage of Nitinol samples.

Due to the exceptionally low count change across all activation combinations when they were run at the expected minimum required voltage, the voltage from the power supply was increased to 6 volts and the test was run again. The source voltage was increased to 6 volts for two reasons. First, because the count data from the first run had such a high percent difference from the expected values. It logically follows that if the voltage supplied to the Nitinol is increased, the amount of Joule heating in the Nitinol will increase, and the length change of the Nitinol will increase, thereby closing the gap between the experimental and expected count results. The second reason the source voltage was increased was that the control code for the system was already set up to give each partition combination the correct percentage of the source voltage in order to achieve the expected voltages across each partition. Therefore, it was simpler and faster to increase the source voltage for the entire system instead of manually altering the control code for each combination.

The results of the second test are recorded in Table 4. The slashes in the voltage values in the second column of Table 4 denote two non-contiguous partitions, and the two values on either side of the slashes are the voltages used in the two partitions that are active for that combination. The voltages from test 2 are, on average, 19.59% different from the calculated voltage values, and the count values are an average of 89.25% different from the calculated values. Additionally, the count values for partition combinations 8 and 9 in the table of values for test 2 are outliers. Combination 8 activates partition 4 alone combination 9 activates partition 4 and 1. Partition 4 is, however, active when used in other combinations such as 10, 11, 12, 13, 14, and 15. Partition 1 is active in combinations 11, 13 and 15, It is worth noting also that for both tests 1 and 2, there is a positive correlation between increasing partition combination and increasing equivalent displacement as shown in Figures 17 and 18.

Table 4

*Voltage, Count, and Displacement values from test 2*

| Partition Combination | Voltage (V) | Counts from Encoder | Equivalent Displacement (mm) | Expected Counts | Expected Displacement (mm) | % Difference of Displacement |
|---|---|---|---|---|---|---|
| 0 | 0.00 | 0 | 0.0E+00 | 0 | 0.00 | 0.00 |
| 1 | 0.13 | 0 | 0.0E+00 | 36 | 0.275 | 100 |
| 2 | 0.26 | 1 | 7.64E-03 | 72 | 0.550 | 98.6 |
| 3 | 0.40 | 1 | 7.64E-03 | 108 | 0.825 | 99.1 |
| 4 | 0.52 | 5 | 3.82E-02 | 144 | 1.10 | 96.5 |
| 5 | 0.17/0.51 | 13 | 9.93E-02 | 180 | 1.38 | 92.8 |
| 6 | 0.77 | 34 | 2.60E-01 | 216 | 1.65 | 84.3 |
| 7 | 0.87 | 34 | 2.60E-01 | 252 | 1.92 | 86.5 |
| 8 | 0.98 | 20 | 1.53E-01 | 288 | 2.20 | 93.1 |
| 9 | 0.16/0.97 | 21 | 1.60E-01 | 324 | 2.47 | 93.5 |
| 10 | 0.31/1.00 | 53 | 4.05E-01 | 360 | 2.75 | 85.3 |
| 11 | 0.41/0.96 | 57 | 4.35E-01 | 396 | 3.02 | 85.6 |
| 12 | 1.49 | 78 | 5.96E-01 | 432 | 3.30 | 81.9 |
| 13 | 0.20/1.47 | 83 | 6.34E-01 | 468 | 3.57 | 82.3 |
| 14 | 1.74 | 106 | 8.10E-01 | 504 | 3.85 | 79.0 |
| 15 | 1.86 | 106 | 8.10E-01 | 540 | 4.12 | 80.4 |

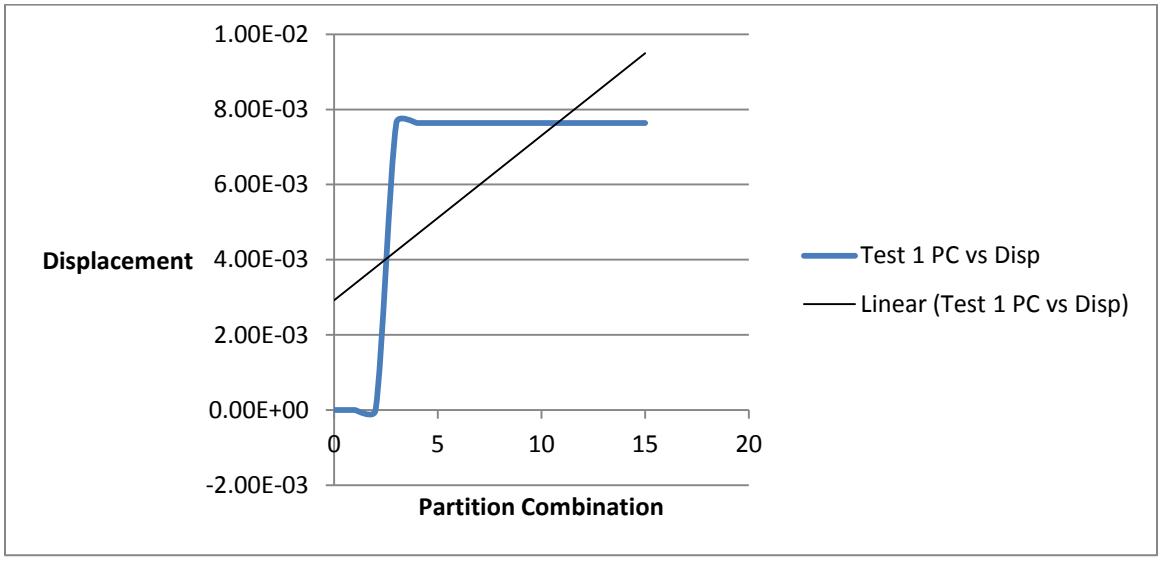


Figure 17. Graph of Partition Combination vs. Displacement for Test 1.

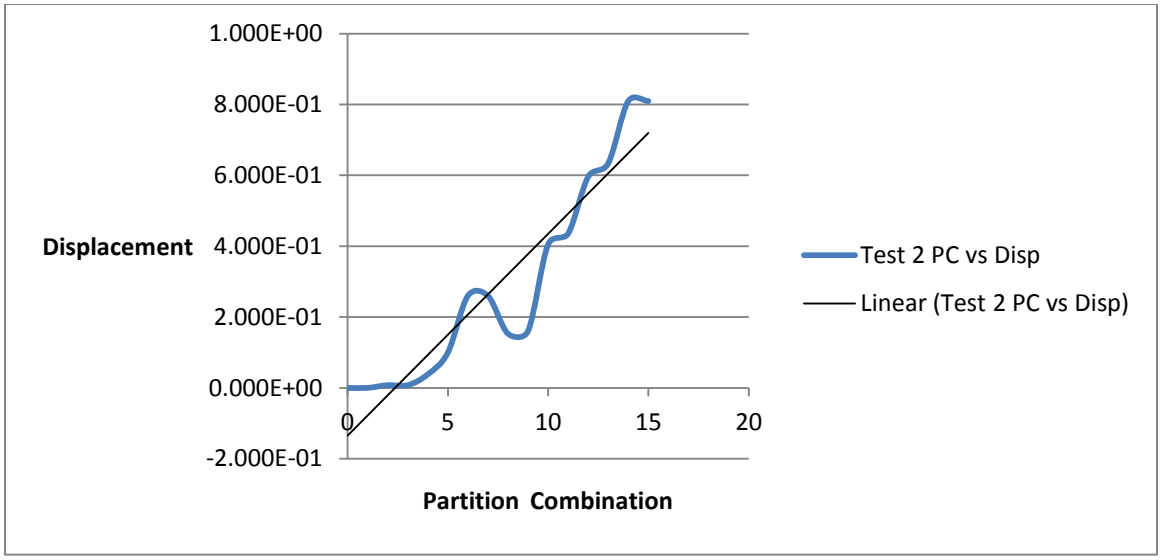


Figure 18. Graph of Partition Combination vs. Displacement for Test 2.

The most likely cause for these discrepancies is either a programming or wiring error. The Nitinol used in this work is sourced from SparkFun and, as such, did not come with a thorough description of the materials properties. Due to this, it is possible that the material used in this work does not fully conform to the properties assumed in the

calculations. All values used in calculations, except for $A_f$=100°C, were averages of values taken from various literature.

The purpose of this work was to develop and test a preliminary small-scale 4-partition digitally-controlled linear actuator which uses Joule-heated Nitinol as an actuation source. Over the course of this work, a prototype (experimental setup 1) was developed and used as a proof of concept for the final design. The prototype's design was then expanded to include a greater number of partitions, more sophisticated controls, a greater amount of total actuation, a greater number of possible actuation lengths, and a higher degree of resolution for the data gathered. The prototype was developed and was used to prove that electrically partitioning Nitinol wire and using it as a digitally-controlled actuator is feasible. The prototype was then refined, adding a gear train, more electrical connections, a desk-top power source, a more sensitive encoder, and an array of MOSFET's for control. After some testing, the gear train was removed due to friction losses and backlash. The final design, experimental setup 3 (Figure 16) was tested for length change based on which partition combinations are active.

In conclusion, while the data gathered from experimentation does not closely match the expected values, the general trend of the data, that being that more partitions activated correlate to more actuation, lends itself to the notion that an actuator of this type is feasible, though it requires further investigation.

# FUTURE WORK

Future work for this line of research would include a more thorough inspection and development of the electrical control system. Many of the difficulties in this work arose from the MOSFET array and related secondary systems. Additionally, the system would undergo steady-state and transient response testing under a number of different loading and temperature conditions. As well, the system would be miniaturized and optimized for use in a small-scale, continuum robot, and the material used in the experiment would be fully characterized by a stress strain curve created by those running the experiments.

# REFERENCES


[1] O. A. Grigory Adamovsky, Masood Andani, et. al., Y. Bar-Cohen, Ed. *High Temperature Materials and Mechanisms*. Boca Raton, FL: Taylor & Francis Group, LLC, 2008, p. 380.

[2] G. Immega and K. Antonelli, "The KSI tentacle manipulator," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, 1995, vol. 3, pp. 3149-3154 vol.3.

[3] E. Amanov, J. Granna, and J. Burgner-Kahrs, "Toward improving path following motion: Hybrid continuum robot design," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4666-4672.

[4] Y. Kim, S. S. Cheng, M. Diakite, R. P. Gullapalli, J. M. Simard, and J. P. Desai, "Toward the Development of a Flexible Mesoscale MRI-Compatible Neurosurgical Continuum Robot," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1-12, 2017.

[5] R. J. Roesthuis and S. Misra, "Steering of Multisegment Continuum Manipulators Using Rigid-Link Modeling and FBG-Based Shape Sensing," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 372-382, 2016.

[6] C. D. R. Deepak Trivedi, William M. Kier, and Ian D. Walker, "Soft Robotics Biological Inspiration, State of the Art, and Future Research," *Applied Bionics and Biomechanics*, Open access vol. Volume 5, no. Issue 3, pp. Pages 99-117, September 2008 2008.

[7] H. Liu *et al.*, "Shape Tracking of a Dexterous Continuum Manipulator Utilizing Two Large Deflection Shape Sensors," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5494-5503, 2015.

[8] J. D. B. Matthew S. Johannes, James M. Burck,, M. V. K. Stuart D. Harshbarger, and a. T. V. Doren, "An Overview of the Developmental Process for the Modular Prosthetic Limb," (in English), *JOHNS HOPKINS APL TECHNICAL DIGEST*, vol. Volume 30, no. Number 3, pp. 207-216, 2011.

[9] X. Zhe and E. Todorov, "Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3485-3492.

[10] G. Robinson and J. B. C. Davies, "Continuum robots - a state of the art," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 1999, vol. 4, pp. 2849-2854 vol.4.

[11] G. Chen, L. Fu, M. T. Pham, and T. Redarce, "Characterization and modeling of a pneumatic actuator for a soft continuum robot," in *2013 IEEE International Conference on Mechatronics and Automation*, 2013, pp. 243-248.

[12] J. D. Greer, T. K. Morimoto, A. M. Okamura, and E. W. Hawkes, "Series pneumatic artificial muscles (sPAMs) and application to a soft continuum robot," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5503-5510.

[13] L. Wu, R. Crawford, and J. Roberts, "Dexterity Analysis of Three 6-DOF Continuum Robots Combining Concentric Tube Mechanisms and Cable-Driven Mechanisms," *IEEE Robotics and Automation Letters,* vol. 2, no. 2, pp. 514-521, 2017.

[14] L. H. B. Elliot W. Hawkes, Joseph D. Greer, and Allison M. Okamura, "A soft robot that navigates its environment through growth," (in English), *Science Robotics,* Science, Robotics, Engineering, Scholarly Article vol. Vol 2, no. Issue 8, July 26, 2017 2017.

[15] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, "Mechanics Modeling of Tendon-Driven Continuum Manipulators," *IEEE Transactions on Robotics,* vol. 24, no. 6, pp. 1262-1273, 2008.

[16] G. Chen, M. T. Pham, and T. Redarce, "Development and kinematic analysis of a silicone-rubber bending tip for colonoscopy," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2006, pp. 168-173.

[17] J. Kedzierski, K. Meng, T. Thorsen, R. Cabrera, and S. Berry, "Microhydraulic Electrowetting Actuators," *Journal of Microelectromechanical Systems,* vol. 25, no. 2, pp. 394-400, 2016.

[18] P. S. Dineva, D. Gross, R. Müller, and T. Rangelov, *Dynamic Fracture of Piezoelectric Materials: Solution of Time-Harmonic Problems via BIEM* (Solid Mechanics and Its Applications, no. 212). Switzerland: Springer International Publishing, 2014, pp. XIV, 249.

[19] C. Naresh, P. S. C. Bose, and C. S. P. Rao, "Shape memory alloys: a state of art review," *IOP Conference Series: Materials Science and Engineering,* vol. 149, no. 1, p. 012054, 2016.

[20] A. R. S. Ashwin Rao, J.N. Reddy, *Design of Shape Memory Alloy (SMA) Actuators* Springer International Publishing AG Switzerland, 2015, p. 130.

[21] B. C. Lawson, *Shape Memory Alloys, Muscle Wires, and Robotics*. United States of America: Capitol Publishers, LLC, 2016, p. 88.

[22] S. P. Kennedy, "Material Characterization of Nitinol Wires for the Design of Actuation Systems," MS in Mechanical Engineering Master's, Mechanical Engineering, California Polytechnic State University, Master's Theses and Project Reports, 2013.

[23] W. B. Cross, A. H. Kariotis, and F. J.Stimler, "Nitinol characterization study," NASA, NASA, WASHINGTON, United States1969.

[24] J. M. Jani, M. Leary, and A. Subic, "Designing shape memory alloy linear actuators: A review," *Journal of Intelligent Material Systems and Structures,* vol. 28, no. 13, pp. 1699-1718, 2017.

[25] B. Selden, C. Kyu-Jin, and H. H. Asada, "Segmented binary control of shape memory alloy actuator systems using the Peltier effect," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 2004, vol. 5, pp. 4931-4936 Vol.5.

[26] S. Brian, C. Kyujin, and H. H. Asada, "Segmented shape memory alloy actuators using hysteresis loop control," *Smart Materials and Structures,* vol. 15, no. 2, p. 642, 2006.

[27] K. C. Mare Meyers, *Mechanical Behavior of Materials Second Edition,* Second ed. New York, United States of America: Cambridge University Press, 2009, p. 882.

[28] M. Van der Wijst, "Shape memory alloys featuring Nitinol," *Stageverslag, WFW-rapport,* pp. 92-085, 1992.

[29] F. E. Wang, W. J. Buehler, and S. J. Pickart, "Crystal Structure and a Unique ``Martensitic" Transition of TiNi," *Journal of Applied Physics,* vol. 36, no. 10, pp. 3232-3239, 1965.

[30] K. Otsuka and X. Ren, "Physical metallurgy of Ti–Ni-based shape memory alloys," *Progress in Materials Science,* vol. 50, no. 5, pp. 511-678, 2005/07/01/ 2005.

[31] L. Case, Z. Kreiner, J. Redmond, and B. Trease, "Shape Memory Alloy Shape Training Tutorial," 2004.

[32] SparkFun.com. (November 19, 2018). *Load Cell - 10kg, Straight Bar (TAL220)* [Webpage]. Available: https://www.sparkfun.com/products/13329

[33] SpiderWire.com. (2018, 3/2/2018). *SPIDERWIRE: BORN FROM THE WORLD'S STRONGEST FIBER™* [Short Article]. Available: http://www.spiderwire.com/SpiderWire-pro-tips-spiderwire-born-from-the-worlds-strongest-fiber.html

[34] SpiderWire.com. (2018, 4/3/2018). *SPIDERWIRE® EZ BRAID™* [Webpage]. Available: http://www.spiderwire.com/spiderwire-line-superline-spiderwire-ez/spiderwire-ez-braid/1285825.html

[35] kelloggsresearchlabs.com. (2015, 4/5/2018). *NITINOL ROUND WIRE (5 FEET)* [Web Page]. Available: https://www.kelloggsresearchlabs.com/Nitinol-Round-Wire

[36] SparkFun.com. (11/19/2018). *Rotary Encoder - 1024 P/R (Quadrature)* [webpage]. Available: https://www.sparkfun.com/products/11102

[37] SparkFun.com. (11/19/2018). *Arduino Uno*. Available: https://www.sparkfun.com/products/11021

[38] SparkFun.com. (11/19/2018). *Temperature Sensor - TMP36*. Available: https://www.sparkfun.com/products/10988

[39] C. Kyu-Jin and H. H. Asada, "Segmentation architecture of multi-axis SMA array actuators inspired by biological muscles," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, 2004, vol. 1, pp. 254-259 vol.1.

[40] C. Kyu-Jin and H. Asada, "Multi-Axis SMA Actuator Array for Driving Anthropomorphic Robot Hand," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 1356-1361.

[41] J. Zhang, A. Gupta, and J. Baker, "Effect of Relative Humidity on the Prediction of Natural Convection Heat Transfer Coefficients," *Heat Transfer Engineering,* vol. 28, no. 4, pp. 335-342, 2007/04/01 2007.

[42] L. D. Balfour, Jr., "Preliminary Thermal and Stress Analysis of an Electrical Connection Acting as a Thermal Dam on a Nitinol Wire," M. S. Thesis, Mechanical Engineering, University of Alabama at Birmingham, AL, 2018.

[43] SparkFun.com. (11/19/2018). *Flexinol - 0.005" Diameter (1 foot)*. Available: https://www.sparkfun.com/products/12095

[44] SparkFun.com. (11/19/2018). *Arduino Mega 2560 R3*. Available: https://www.sparkfun.com/products/11061

[45] Amazon.com. (11/18/2018). *1 Channel, 5V DC 10A low level trigger relay module, 5 volt Coil Voltage Relay Module (2) STK kit for Ardunio, and other development environments, RBTMKR; TONGLING JQC-3FF-S-Z 5VDC Relay*. Available: https://www.amazon.de/Channel-trigger-

Spulenspannung-development-environments/dp/B06XY31XBK/ref=pd_lpo_sbs_328_img_2/257-0084736-7372603?_encoding=UTF8&psc=1&refRID=VKQHBMWFVH15Q5X7SVA0#detail_bullets_id

[46] SparkFun.com. *Technical Characteristics of Flexinol Actuator Wires* [Technical Characteristic Document]. Available: https://www.sparkfun.com/products/12095

[47] en.Wikipedia.org. (2018, 3 March 2018 01:14 UTC). *Carbon steel* [Electronic Encyclopedia Article]. Available: https://en.wikipedia.org/w/index.php?title=Carbon_steel&oldid=828506706

[48] JMMedical.com. (2018, 3/18/2018). *Nitinol Technical Properties* [Technical data website]. Available: http://jmmedical.com/resources/221/Nitinol-Technical-Properties.html

# APPENDIX A

Calculation for Spider wire strain under experimental load:

Assuming:

Highest load seen in testing: 4.08kg

Diameter of spider wire = 0.33mm/0.00033m

X-sec area of spider wire = $8.53x10^{-8}$ $m^2$

Length of Spider wire used = 0.102m

Modulus = ~ 200 GPa [33, 47]

$$\varepsilon = \frac{P}{\left(\frac{E \cdot A}{L}\right)} = \frac{40.0N}{\left(\frac{200x10^{9}Pa \cdot 8.5x10^{-8}m^{2}}{0.102m}\right)} = 2.38x10^{-6}m$$

This amount of strain is negligible for the tests performed in this work.

Calculation of granularity of measurements during testing:

The Nitinol is 1012mm long and has 4 partitions, so the partitions will, in order of shortest to longest, be 6.77 mm, 13.5 mm, 27.093 mm, and 54.2mm long (Equation 6). The Nitinol will have a counterweight of 0.79kg, and so it will theoretically stretch 18.0 mm (Equation 4). The shortest partition is 1/15 the total length of the actuator which means its stroke will be 1/15 the total strain. This means the change in length of the shortest partition during actuation will be $\frac{18.0\text{ mm}}{15} =$ 1.20mm. The axle that the Nitinol will be connected to, via the Spider Wire, has a diameter of 4.29 mm, therefore, the actuation of the shortest partition (or one unit of actuation in this case) will cause the pulley to rotate 32.1°. The encoder collects data at a rate of 2048 counts per rotation or

5.69counts per degree. This means that one unit of actuation will produce ~182 counts from the encoder. This is sufficient resolution to track and quantify the subtle changes in length that will occur during actuation and under changing load and temperature conditions.

Table 5

*Lengths of Partitions*

| Partition | Length (mm) |
|---|---|
| **1** | 6.77 |
| **2** | 13.5 |
| **3** | 27.1 |
| **4** | 54.2 |

Calculation of the resistance of a 102mm length of Nitinol:

$$R = \frac{\rho \cdot L}{A} = \frac{(10^{-6}\Omega m) \cdot (0.102m)}{\pi \cdot (6.35x10^{-5}m)^2} = 7.22\Omega$$

R - Resistance of a length of Nitinol wire [Ω]

ρ - Resistivity of Nitinol (Austenite Phase) [$\Omega \cdot m$]

L - Length of Nitinol wire [m]

A - Cross-sectional area of Nitinol wire [$m^2$]

Using the same equation, the resistances for the individual partitions and combinations of consecutive partitions are calculated and tabulated below

Table 6

*Resistance of Partition Sets*

| Partition Set | Resistance (Ω) |
|---|---|
| **1** | 0.535 |
| **2** | 1.07 |
| **3** | 2.14 |
| **4** | 4.28 |
| **1,2** | 1.60 |
| **2,3** | 3.21 |
| **3,4** | 6.42 |
| **1,2,3** | 3.74 |
| **2,3,4** | 7.49 |

Calculation for the current required to maintain transformation temperature [42, 48]:

$$I = \sqrt{\frac{T - T_{amb}}{4 \cdot \rho} \cdot \pi^2 \cdot d^3 \cdot h} = \sqrt{\frac{373K - 298K}{4 \cdot (10^{-6}\Omega m)} \cdot \pi^2 \cdot (1.27x10^{-4}m)^3 \cdot 9.86\,{}^{W}/_{m^2K}}$$

$$= 0.061amps$$

I - Current required to maintain steady-state temperature [amps]

T - Temperature of the Nitinol wire [K]

$T_{amb}$ - Temperature of the surroundings [K]

ρ - Resistivity of Nitinol (Austenite Phase) [$\Omega \cdot m$]

h - Heat Transfer Coefficient [$W/m^2K$]

d - Wire Diameter [m]

Voltage required to heat each combination of consecutive partitions:

Table 7

*Voltage, Current, and Power Required for Each Partition Set*

| Partition Set | Resistance (Ω) | Voltage (V) | Current (A) | Power Consumed (W) |
|---|---|---|---|---|
| 1 | 0.534 | 0.106 | 0.199 | 0.021 |
| 2 | 1.07 | 0.213 | 0.199 | 0.042 |
| 3 | 2.14 | 0.319 | 0.149 | 0.048 |
| 4 | 4.28 | 0.425 | 0.099 | 0.042 |
| 1,2 | 1.60 | 0.531 | 0.331 | 0.176 |
| 2,3 | 3.21 | 0.638 | 0.199 | 0.127 |
| 3,4 | 6.42 | 0.744 | 0.116 | 0.086 |
| 1,2,3 | 3.74 | 0.850 | 0.227 | 0.193 |
| 2,3,4 | 7.49 | 0.957 | 0.128 | 0.122 |

Table 8

*Voltages and Corresponding Analog Signals for Activation Combinations*

| Partition Combination | Voltage (V) | Analog Write Signal (PWM) |
|---|---|---|
| 0 | 0.000 | 255 |
| 1 | 0.106 | 250 |
| 2 | 0.213 | 244 |
| 3 | 0.319 | 239 |
| 4 | 0.425 | 233 |
| 5 | 0.531 | 250/233 |
| 6 | 0.638 | 222 |
| 7 | 0.744 | 217 |
| 8 | 0.850 | 212 |
| 9 | 0.957 | 250/212 |
| 10 | 1.063 | 244/212 |
| 11 | 1.169 | 239/212 |
| 12 | 1.275 | 190 |
| 13 | 1.382 | 250/190 |
| 14 | 1.488 | 179 |
| 15 | 1.594 | 174 |

Calculation for expected change in length due to preload in Martensite phase [48]:

Note: Modulus values are an average of the range given in the source.

$$\Delta L = \frac{(P/A)}{D(\xi)} \cdot L = \frac{\left(22.2411N/\pi \cdot (6.35x10^{-5}m)^2\right)}{34.5x10^9 Pa} \cdot 0.1016m = 0.00517m = 5.17mm = 5.1\%$$

$\Delta L$ - Change in Length of Nitinol Wire [m]

P - Load Applied to Wire [N]

A - Cross-sectional Area of Wire [$m^2$]

$D(\xi)$ - Young's Modulus of Nitinol [Pa]

L - Original Total Length of Nitinol Wire [m]

Calculation for expected change in length due to preload in Austenite phase [48]:

Modulus values are an average of the range given in the source.

$$\Delta L = \frac{(P/A)}{D(\xi)} \cdot L = \frac{\left(22.2411N/\pi \cdot (6.35x10^{-5}m)^2\right)}{442.5x10^9 Pa} \cdot 0.1016m = 0.00215m$$

$$= 2.15mm$$

$\Delta L$ - Change in Length of Nitinol Wire [m]

P - Load Applied to Wire [N]

A - Cross-sectional Area of Wire [$m^2$]

$D(\xi)$ - Young's Modulus of Nitinol [Pa]

L - Original Total Length of Nitinol Wire [m]

Calculation for actuation length of first partition:

$$L_{actuation} = \frac{\Delta L_{Martensite} - \Delta L_{Austenite}}{15} = \frac{5.18mm - 2.15mm}{15} = 0.202mm$$

$L_{actuation}$ - Change in Length of Nitinol Wire After Actuation (First Partition) [mm]

$\Delta L_{Martensite}$ - Change in Length of Nitinol Wire Under Preload (Martensite) [mm]

$\Delta L_{Austenite}$ - Change in Length of Nitinol Wire Under Preload (Austenite) [mm]

Calculation for Maximum Load on 5mil Nitinol based on Ultimate Tensile Strength [43]

$$\varepsilon = \frac{\sigma}{E} = \frac{8.95x10^{8}}{7.5x10^{10}} = 0.012 \approx 1.2\% \text{ strain}$$

$\varepsilon$ - Strain []

$\sigma$ - Stress equal to Ultimate Tensile Strength [Pa]

E - Modulus of Elasticity [Pa]

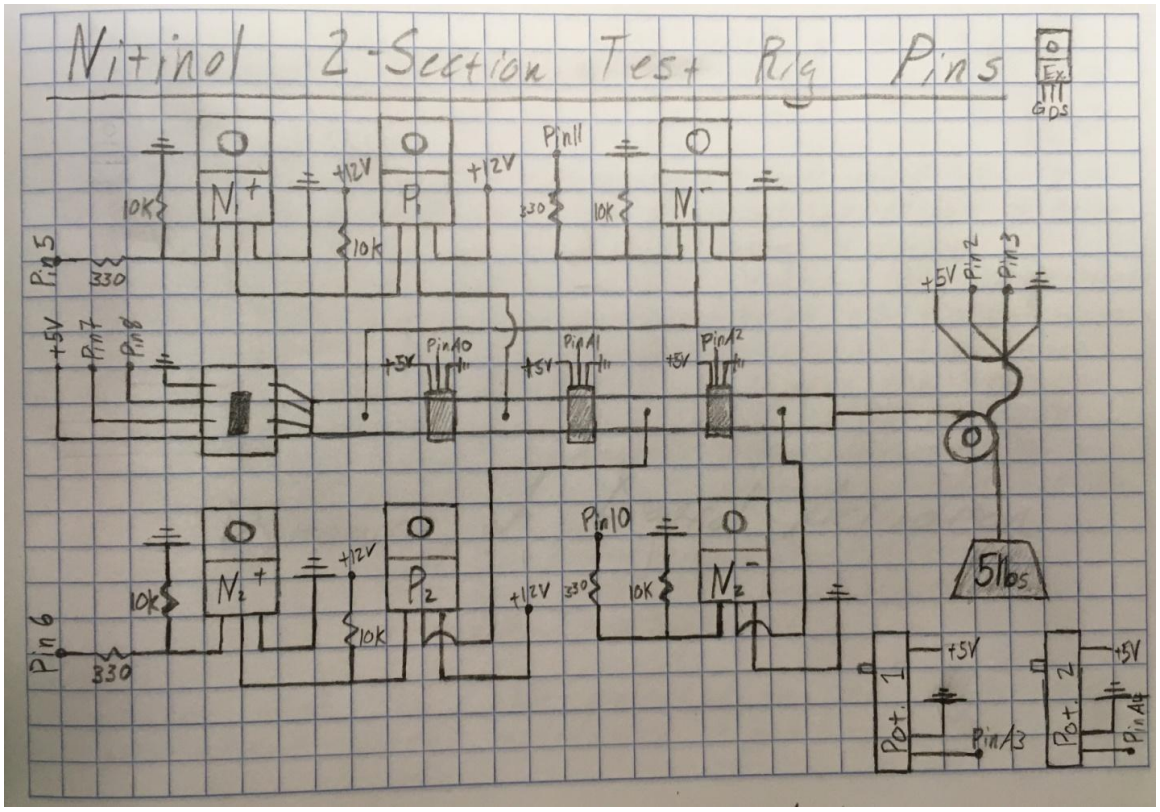


Figure 19. Wiring Diagram of Experimental Setup 1.

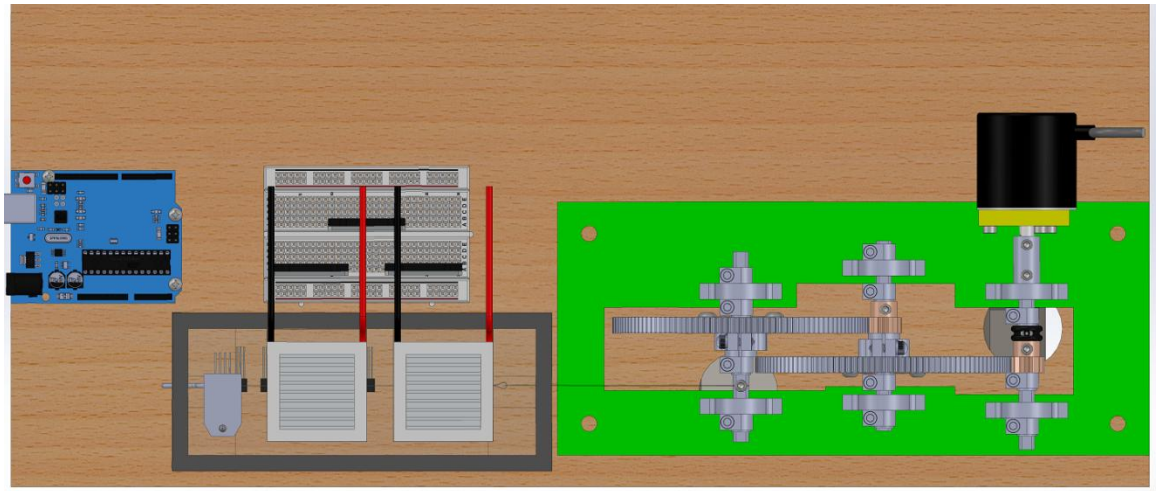

Figure 20. Model of Experimental Setup 2.

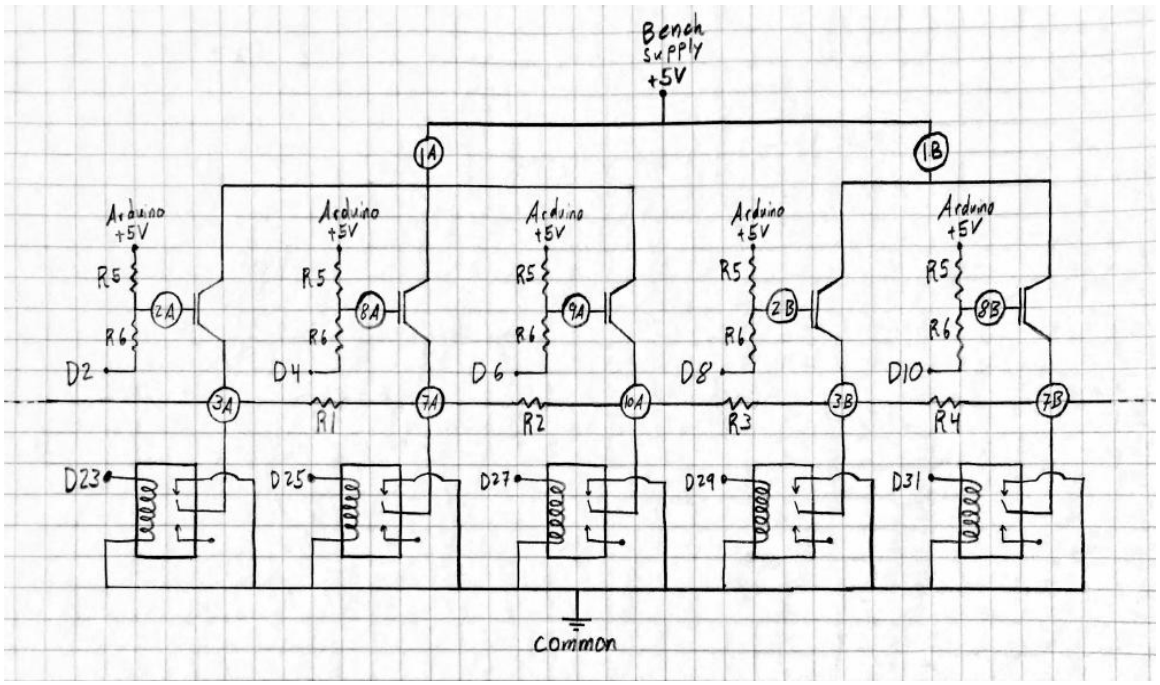


Figure 21. Wiring Diagram of Experimental Setup 3.

In Figure 21, R1-R4 represent the partitions of Nitinol, while R5 and R6 are just resistors.

R1 = 1.5 Ω; R2 = 2.8 Ω; R3 = 3.5 Ω; R4 = 5.3 Ω; R5 = 100 Ω; R6 = 10Ω

Arduino code used in experimental setup 1:

```
/*
  Martin Holland I
  Lloyd Balfour Jr.
  Douglas Ross Esq.
*/
int PositiveNType1 = 5;
int NegativeNType1 = 11;
int PositiveNType2 = 6;
int NegativeNType2 = 10;
int E1;
int E2;
volatile long count = 0;
volatile int bValue;
int encoder1 = 2;
int encoder2 = 3;
float Temperature = 0;
#include <HX711.h>
#define DOUT 7
#define CLK 8
```

```
HX711 scale(DOUT, CLK);
float calibration_factor = (-7050);
int temperpin = 0;
double VoltageCF = 5.0 / 1024.0;
double m = 104.167; // slope of the T(V) curve;
double b = 54.167; // slope intercept of the T(V) curve;
double gettemp(int sensorPin) {
        int VoltageReading = analogRead(sensorPin);
        double voltage = VoltageReading * VoltageCF;
        double temperC = m * voltage - b;
        double temperF = (temperC * 9.0 / 5.0) + 32.0;
        return (temperF);
}
void setup() {
        Serial.begin(9600);
        Serial.println("hello");
        pinMode(PositiveNType1, OUTPUT);
        pinMode(NegativeNType1, OUTPUT);
        pinMode(PositiveNType2, OUTPUT);
        pinMode(NegativeNType2, OUTPUT);
        Serial.begin(9600);
        pinMode(encoder1, INPUT_PULLUP);
        pinMode(encoder2, INPUT_PULLUP);
        attachInterrupt(digitalPinToInterrupt(encoder1), encoderPulse, CHANGE);
        Serial.begin (9600);
        Serial.println("HX711 calibration sketch");
        Serial.println("Remove all weight from scale");
        Serial.println("After readings begin, place known weight on scale");
        Serial.println("Press + or a to increase calibration factor");
        Serial.println("Press - or z to decrease calibration factor");
        scale.set_scale();
        scale.tare();
        long zero_factor = scale.read_average();
        Serial.print("Zero Factor:");
        Serial.println(zero_factor);
}
void loop() {
        int potentiometer1 = analogRead(A3);
        int potentiometer2 = analogRead(A4);
        int ScalePotentiometer1 = potentiometer1 / 32;
        int ScalePotentiometer2 = potentiometer2 / 32;
        analogWrite(NegativeNType1, ScalePotentiometer1);
        if (potentiometer1 > 2) {
                digitalWrite(PositiveNType1, HIGH);
        }
        else {
```

```
                digitalWrite(PositiveNType1, LOW);
        }
        analogWrite(NegativeNType2, ScalePotentiometer2);
        if (potentiometer2 > 2) {
                digitalWrite(PositiveNType2, HIGH);
        }
        else {
                digitalWrite(PositiveNType2, LOW);
        }
        Serial.print("Slider1: ");
        Serial.print(ScalePotentiometer1);
        Serial.print(" ");
        Serial.print("Slider2: ");
        Serial.print(ScalePotentiometer2);
        Serial.print(" ");
        Serial.print(" ");
        Serial.print("Encoder:");
        Serial.print(count);
        Serial.print(" ");
        Serial.print(" ");
        scale.set_scale(calibration_factor);
        float getunits = scale.get_units();
        float pounds = getunits * 5.6 / 73;
        Serial.print("Reading:");
        Serial.print(pounds, 2);
        Serial.print(" lbs");
        Serial.print(" ");
        Serial.print(" Temp1:");
        Serial.print(gettemp(0));
        Serial.print(" Temp2:");
        Serial.print(gettemp(1));
        Serial.print(" Temp3:");
        Serial.print(gettemp(2));
        Serial.println();
        if (Serial.available()){
                char temp = Serial.read();
                if (temp == '+' || temp == 'a')
                calibration_factor += 10;
                else if (temp == '-' || temp == 'z')
                calibration_factor -= 10;
        }
        delay(1);
}
void encoderPulse(){
        E1 = digitalRead(encoder1);
        E2 = digitalRead(encoder2);
```

```
        if (E1 == E2){
                count = count - 1;
        }
        else{
                count = count + 1;
        }
}
```

Arduino Code used in Experimental Setup 2:

```
/*Martin Holland
Douglas Ross
11/15/18*/
int State;
int Pin2 = 2;
int Pin8 = 4;
int Pin9 = 6;
int Pin2b = 8;
int Pin8b = 10;
int Pin4 = 23;
int Pin6 = 25;
int Pin11 = 27;
int Pin4b = 29;
int Pin6b = 31;
int E1;
int E2;
int encoder1 = 18;
int encoder2 = 19;
volatile long count = 0;
volatile int bValue;
float Temperature = 0;
float calibration_factor = (-7050);
int temperpin = 0;
double VoltageCF = 5.0 / 1024.0;
double m = 104.167; // slope of the T(V) curve;
double b = 54.167; // slope intercept of the T(V) curve;
#include <HX711.h>
#define DOUT 50
#define CLK 52
HX711 scale(DOUT, CLK);
double gettemp(int sensorPin) {
        int VoltageReading = analogRead(sensorPin);
        double voltage = VoltageReading * VoltageCF;
        double temperC = m * voltage - b;
        double temperF = (temperC * 9.0 / 5.0) + 32.0;
        return (temperF);
```

```
}
void setup() {
	Serial.begin(9600);
	Serial.println("hello");
	pinMode(Pin2, OUTPUT);
	pinMode(Pin4, OUTPUT);
	pinMode(Pin8, OUTPUT);
	pinMode(Pin6, OUTPUT);
	pinMode(Pin9, OUTPUT);
	pinMode(Pin11, OUTPUT);
	pinMode(Pin2b, OUTPUT);
	pinMode(Pin4b, OUTPUT);
	pinMode(Pin8b, OUTPUT);
	pinMode(Pin6b, OUTPUT);
	pinMode(encoder1, INPUT_PULLUP);
	pinMode(encoder2, INPUT_PULLUP);
	attachInterrupt(digitalPinToInterrupt(encoder1), encoderPulse, CHANGE);
	Serial.println(scale.read());
	Serial.println("HX711 calibration sketch");
	Serial.println("Remove all weight from scale");
	Serial.println("After readings begin, place known weight on scale");
	Serial.println("Press + or a to increase calibration factor");
	Serial.println("Press - or z to decrease calibration factor");
	scale.set_scale(calibration_factor);
	scale.tare();
	long zero_factor = scale.read_average();
	Serial.print("Zero Factor:");
	Serial.println(zero_factor);
}
void loop() {
	int slider1 = analogRead(7);
	State = map(slider1, 0, 1000, 0, 15);
	if (State == 0){
		analogWrite(Pin2, 255);
		digitalWrite(Pin4, LOW);
		analogWrite(Pin8, 255);
		digitalWrite(Pin6, LOW);
		analogWrite(Pin9, 255);
		digitalWrite(Pin11, LOW);
		analogWrite(Pin2b, 255);
		digitalWrite(Pin4b, LOW);
		analogWrite(Pin8b, 255);
		digitalWrite(Pin6b, LOW);
		Serial.print("0 ");
	}
	else if (State == 1){
```

```
        analogWrite(Pin2, 250);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, HIGH);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, LOW);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, LOW);
        analogWrite(Pin8b, 255);
        digitalWrite(Pin6b, LOW);
        Serial.print("1 ");
}
else if (State == 2){
        analogWrite(Pin2, 255);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 237);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, HIGH);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, LOW);
        analogWrite(Pin8b, 255);
        digitalWrite(Pin6b, LOW);
        Serial.print("2 ");
}
else if (State == 3){
        analogWrite(Pin2, 227);
        digitalWrite(Pin4, LOW);
        digitalWrite(Pin8, HIGH);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, HIGH);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, LOW);
        analogWrite(Pin8b, 255);
        digitalWrite(Pin6b, LOW);
        Serial.print("3 ");
}
else if (State == 4){
        analogWrite(Pin2, 255);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 222);
        digitalWrite(Pin11, LOW);
        analogWrite(Pin2b, 255);
```

```
        digitalWrite(Pin4b, HIGH);
        analogWrite(Pin8b, 255);
        digitalWrite(Pin6b, LOW);
        Serial.print("4 ");
}
else if (State == 5){
        analogWrite(Pin2, 244);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, HIGH);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, HIGH);
        analogWrite(Pin2b, 222);
        digitalWrite(Pin4b, LOW);
        analogWrite(Pin8b, 255);
        digitalWrite(Pin6b, LOW);
        Serial.print("5 ");
}
else if (State == 6){
        analogWrite(Pin2, 255);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 209);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, LOW);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, HIGH);
        analogWrite(Pin8b, 255);
        digitalWrite(Pin6b, LOW);
        Serial.print("6 ");
}
else if (State == 7){
        analogWrite(Pin2, 202);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, LOW);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, HIGH);
        analogWrite(Pin8b, 255);
        digitalWrite(Pin6b, LOW);
        Serial.print("7 ");
}
else if (State == 8){
        analogWrite(Pin2, 255);
```

```
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, LOW);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, HIGH);
        analogWrite(Pin8b, 199);
        digitalWrite(Pin6b, LOW);
        Serial.print("8 ");
}
else if (State == 9){
        analogWrite(Pin2, 244);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, HIGH);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, LOW);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, HIGH);
        analogWrite(Pin8b, 199);
        digitalWrite(Pin6b, LOW);
        Serial.print("9 ");
}
else if (State == 10){
        analogWrite(Pin2, 255);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 237);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, HIGH);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, HIGH);
        analogWrite(Pin8b, 199);
        digitalWrite(Pin6b, LOW);
        Serial.print("10 ");
}
else if (State == 11){
        analogWrite(Pin2, 227);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, HIGH);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, HIGH);
```

```
        analogWrite(Pin8b, 199);
        digitalWrite(Pin6b, LOW);
        Serial.print("11 ");
}
else if (State == 12){
        analogWrite(Pin2, 255);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, LOW);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, HIGH);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, LOW);
        analogWrite(Pin8b, 177);
        digitalWrite(Pin6b, LOW);
        Serial.print("12 ");
}
else if (State == 13){
        analogWrite(Pin2, 244);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, HIGH);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, HIGH);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, LOW);
        analogWrite(Pin8b, 177);
        digitalWrite(Pin6b, LOW);
        Serial.print("13 ");
}
else if (State == 14){
        analogWrite(Pin2, 255);
        digitalWrite(Pin4, LOW);
        analogWrite(Pin8, 255);
        digitalWrite(Pin6, HIGH);
        analogWrite(Pin9, 255);
        digitalWrite(Pin11, LOW);
        analogWrite(Pin2b, 255);
        digitalWrite(Pin4b, LOW);
        analogWrite(Pin8b, 166);
        digitalWrite(Pin6b, LOW);
        Serial.print("14 ");
}
else if (State == 15){
        analogWrite(Pin2, 255);
        digitalWrite(Pin4, HIGH);
```

```
                analogWrite(Pin8, 255);
                digitalWrite(Pin6, LOW);
                analogWrite(Pin9, 255);
                digitalWrite(Pin11, LOW);
                analogWrite(Pin2b, 255);
                digitalWrite(Pin4b, LOW);
                analogWrite(Pin8b, 161);
                digitalWrite(Pin6b, LOW);
                Serial.print("15 ");
        }
        else {
                analogWrite(Pin2, 255);
                digitalWrite(Pin4, LOW);
                analogWrite(Pin8, 255);
                digitalWrite(Pin6, LOW);
                analogWrite(Pin9, 255);
                digitalWrite(Pin11, LOW);
                analogWrite(Pin2b, 255);
                digitalWrite(Pin4b, LOW);
                analogWrite(Pin8b, 255);
                digitalWrite(Pin6b, LOW);
                Serial.print("else ");
        }
        scale.set_scale(calibration_factor);
        float getunits = scale.get_units(1);
        float pounds = -getunits * 5.6 / 73;
        Serial.print("Load Cell:");
        Serial.print(pounds, 2);
        Serial.print(" lbs");
        Serial.print("    ");
        Serial.print("slider1: ");
        Serial.print(slider1);
        Serial.print("    ");
        Serial.print("State: ");
        Serial.print(State);
        Serial.print("    ");
        Serial.print("Encoder:");
        Serial.print(count);
        Serial.print("    ");
        Serial.print(E1);
        Serial.print("    ");
        Serial.print(E2);
        Serial.println();
        delay(1);
}
void encoderPulse(){
```

```
	E1 = digitalRead(encoder1);
	E2 = digitalRead(encoder2);
	if (E1 == E2){
		count = count - 1;
	}
	else{
		count = count + 1;
	}
}
void encoderPulse2(){
	E1 = digitalRead(encoder1);
	E2 = digitalRead(encoder2);
	if (E1 == E2){
		count = count + 1;
	}
	else{
		count = count - 1;
	}
}
```