INSTRUMENTATION APPROACHES TO BIOLOGICAL SYSTEM ASSESSMENT, GUIDED MSC DIFFERENTIATION, AND FUNCTIONAL TISSUE ENGINEERING

Avery Ted Cashion IV

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biomedical Engineering.

Chapel Hill
2013

Approved By:

Robert Dennis, PhD

Mark Tommerdahl, PhD

Oleg Favorov, PhD

Paul Dayton, PhD

John van Aalst, MD

ABSTRACT

Avery Ted Cashion IV: Instrumentation Approaches to Biological System Assessment, Guided MSC Differentiation, and Functional Tissue Engineering
(Under the direction of Robert Dennis)

Instrumentation methods and tools are developed for electrical and mechanical stimulus/assessment of biological systems. In **Chapter 1**, a distributed control data acquisition system (DAQ) was developed and demonstrated in the application of measurement and analysis of dielectric relaxation sensing of bio-cultures. As a demonstrative example, a simple probe was constructed and used to measure absorption behavior of varying volumes of soybean oil. Extracted features of this calibration dataset were used to train a support vector machine (SVM). The SVM was then able to predict the known volumes of a separate specimen with approximately 98% accuracy. In **Chapter 2**, enabling bioreactor technologies were developed for the investigation of the effects of periodic strain on patches of self-organized rat myocardial tissue in-vitro using a novel voice coil actuator (VCA) system. The DAQ was purposed as a control board for the VCA mechanobioreactor network. The researcher can program strain protocols for individual bioreactors using a single user interface on a computer. Possible implementations of the described system include micro-newton twitch force measurement capability as well as the ability to generate stress-strain curves of engineered tissues. To examine the compatibility of the bioreactor with engineered tissue samples, a uniaxial 1Hz cyclic 10% tensile strain was applied to 2cm square tissue patches for 4 hours. A significant increase in contractile force was observed after the

strain period. In **Chapter 3**, the VCA system was modified to enable research on the effects of periodic vibratory stimulus on human and porcine mesenchymal stem cells (MSCs). Multiple bioreactors within the incubator can be independently addressed. Once programmed, the embedded microprocessor and sensor system on each bioreactor execute the specified protocol independent of the computer. Sinusoidal stimuli were applied to culture plates in 1 minute intervals with a 15 minute rest following each, for a total of 15 hours per day for 10 days. Frequencies of 1 and 100Hz were applied to cultures of both human and porcine umbilical cord (UC)-derived MSCs. Staining and mRNA quantification indicated that 1 Hz stimulation resulted in a cartilage phenotype for both human and porcine MSCs; 100Hz stimulation resulted in bone phenotype.

To my family, friends and mentors.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

Introduction

Following the initial expansion phase of the zygote into the blastocyst, gastrulation organizes the embryonic stem cells into three layers labeled ectoderm, mesoderm and endoderm. Masses of ESCs from each of these germ layers are formed into tissues and organs by intricate activations and repressions of specific genes. The ectoderm forms most of the central and peripheral nervous system as well as hair, nails and tooth enamel. From the mesoderm layer comes the majority of the musculoskeletal, cardiovascular and lymphatic systems as well as gonads and connective tissues. Finally, the endoderm produces most of the organs in the gastrointestinal tract as well as liver, pancreas and lungs. The human body continues to repair and replace damaged tissue throughout life based on the blueprint laid out during embryogenesis but is never again capable of regenerating fully functional organs from scratch.

When tissue damage is beyond repair by natural healing processes, replacement interventions often become necessary. Depending on the specific condition and on the organ or tissue type, successful treatments can be achieved by autograft, xenographt, or allograft of whole or parts of organs and tissues. The holy grail in tissue engineering is to create fully functional replacement parts for the human body from a person's own cells in order to eliminate recipient rejection issues. The research path to achieve this goal requires diverse approaches, skillsets, and tools as well as management of ethical and sociopolitical difficulties.

A recent query of 24 leaders in the field of tissue engineering was conducted on the topic of achieving large scale clinical significance by the year 2021.[1] The study is largely subjective but provides an interesting amalgamated expert perspective. Briefly, each person was asked to generate a list of the 10 most important concepts that will lead to the stated goal. Dominance of each listed research concept over the others was quantified by analysis of the interdependence of all of the provided concepts. Progress in each research direction was estimated by literature review. The authors of the study normalized the dominance scores by the progress scores to yield a sorted list of the most strategic research directions for tissue engineering to achieve large scale clinical significance. Topping the list of the most strategic research directions were angiogenesis research and stem cell science. These topics are central to the majority of the work presented in this dissertation.

Instrumentation

At the core of all reliable scientific observation and experimentation are the principles of measurement and instrumentation. The development of universal standards has made possible the successful cataloging of verifiable scientific data from research spanning over many centuries. Over the years, the development of science has demanded several increases in the precision of known standards. At the time of this writing, the International System of Units defines a second as the amount of time taken by 9192631770 oscillations of light emitted by a cesium-133 atom, a meter as the distance light travels in 1/299792458 seconds, and 1kg as the mass of a 3.9cm x 3.9cm platinum-iridium cylinder. Interrelationships between these standards, along with those of Temperature (1K), Current (1A), moles (1mol) and luminous intensity (1cd), form the units for all scientific measurement. In the present

context, instrumentation is the art of meeting the complex demands of a given experiment in order to generate data within the framework of existing standards. Basic physical concepts can be logically interconnected and compounded into precise, functional systems in much the same way fine music can be composed using a finite number of piano keys.

Biology and regenerative medicine is a broad field of research that creates an enormous need for sensor systems and bioreactor engineering. Highly specialized skillsets are in many cases making interdisciplinary collaboration a necessity for scientific advancement. Each part of the work presented here is a product of a diverse group of intelligent individuals noncompetitively working toward a common goal. Electrical and mechanical engineering principles are applied to biological research problems to create engineering solutions that provide enabling technologies to important areas of biological research.

Outline of Work

The focus of Chapter 1 is on the application of direct current dielectric absorption and relaxation phenomena for bulk material sensory functions. Measurement of oil content in algae was selected as the most impactful demonstration because of the increasing global need for renewable energy resources. Indeed, the majority of petroleum on Earth is believed to have been formed by and from algae-like life forms; as is logical when considering that they were the only life on the planet for billions of years. While there is an elegant and elaborate evolutionary connection between single cellular life forms and complex mammalian tissues, that topic is not approached herein. The intended association between Chapter 1 and the other

chapters lies in the versatile data acquisition system that was developed and further employed in Chapters 2 and 3, as well as the nondestructive assessment of a biological system.

Chapter 2 details the development of instrumentation controlled by the Chapter 1 DAQ to explore the effects of cyclic mechanical strain on patches of self-organized myocardial tissue. Engineered cardiac patches may have great potential for the treatment of damaged or scarred heart tissue in myocardial infarction patients. In order to achieve clinically relevant adult phenotype patches, a great deal of basic research into the factors that govern heart tissue organization and extracellular matrix production is necessary. In order to study pure myocardial tissue, cardiac patches were cultured from primary rat heart cells using a digestible "scaffold". The resultant patches are held together by cell-cell and cell-matrix interactions. To simulate the physically dynamic environment of the beating heart, a programmable voice coil actuator was developed and attached to the cardiac constructs. Viability of the instrumentation was examined by the application of a 1Hz cyclic 10% strain. While still an order of magnitude lower than adult phenotype tissue, patch twitch force showed a marked increase following the applied stimulus. This result would suggest that physical manipulations of cardiac patches may play a critical role in the development of clinically relevant heart tissues.

Mesenchymal stem cells (MSCs) are known to differentiate into a great number of (predominantly mesodermal) tissues. Their multipotency and ready availability from adult tissue (bone marrow, adipose tissue, etc.) make MSCs an attractive option for application in regenerative medicine and tissue engineering research. Cardiomyocytes are among the cell types that have been reported from MSCs[2,3] and their potential for improvement of cardiac function must be explored. On this line of reasoning, the instrumentation from Chapter 2

was modified and applied to the mechanical stimulation of MSC cultures in Chapter 3. The results from this experiment agree with the suggestion from the literature that in the absence of other stimuli, the gene expression regulators that are sensitive to mechanical stimulus will only trigger osteogenic or chondrogenic MSC lineages. However, it may be possible that other cell signaling pathways will increase the sensitivity to vibration of other gene expressions. For example, co-culture of MSCs with self-organized cardiac patches and applied cyclic strain to the knowledge of the author has not been explored. An initial study is detailed in Chapter 3, wherein the efficacy of the instrumentation was evaluated and shown to elicit a chondrogenic lineage at 1Hz stimulus and osteogenic at 100Hz.

Recently in the field of cardiac tissue engineering, decellularized rat heart has been demonstrated to be an extremely effective biomatrix for the production of functional myocardial tissue.[4] Perfusion decellularization methods remove the cellular and genetic material and leave behind the (mostly collagen) extracellular matrix. Along with preserved organ geometry and matrix molecules, the resultant constructs retain an intact vasculature that can be used for both nutrient delivery and recellularization. From these constructs, bioartificial hearts have been developed by recellularization with cardiac cells.[4] With electrical stimulation, the engineered hearts performed at about 2% adult rat heart pump function.

At the time of this writing, developments in whole organ decellularization methods are taking place at UNC using rat liver.[5] Appendix 1 summarizes a collaborative work wherein a functional assessment method is developed for decellularized biomatrix scaffolds using acoustic angiography.[6]

CHAPTER 1: INSTRUMENTATION AND ANALYSIS FOR SENSOR APPLICATIONS OF LOW FREQUENCY DIELECTRIC ABSORPTION[1]

1.1 Introduction

Dielectric Absorption

Dielectric absorption describes an intrinsic property of dielectric materials employed within electrical capacitors wherein the charge on a capacitor returns to a certain degree after briefly shorting the terminals. Dielectric theory suggests that dielectric absorption arises from a gradual polarization of the molecules of the dielectric material and/or orientation of existing dipoles within the material in the presence of an electric field.[7–10] Because all dielectric materials are by definition electrical insulators, charge redistribution within dielectric materials is much slower than in electrical conductors which results in a slower change in the resulting electric field condition[10–12]. Thus, for a capacitor to be completely discharged, the short circuit condition must remain for a length of time sufficient to allow the dielectric polarization to relax, permitting this form of stored energy within the dielectric to be fully equilibrated.

Dielectric absorption is significant in modern-day capacitors, leading to important non-linear behavior often referred to in electronics literature as "soakage." This non-linear effect on capacitance has been a thorn in the side of circuit designers since the invention of

---

[1]This chapter was submitted to IEEE Transactions on Instrumentation and Measurement. The citation is as follows:
Cashion, A. T., Hubbard, D. K., Donnelly, K., Favorov, O. & Dennis, R. G. Instrumentation and Analysis for Sensor Applications of Low Frequency Dielectric Absorption. (2013).

capacitors over 100 years ago.[7,8,13] Soakage must be considered and to the extent possible eliminated when working with systems that rely heavily on precision analog signals that are stored as a voltage in a capacitor because errors are often introduced over time as energy "soaks" into and is released from the dielectric material.[9,13,14] Such behavior can cause very large errors in many common analog signal conditioning and filtering circuits, including but not limited to the ubiquitous sample-and-hold components found in virtually all analog-to-digital conversion circuits. In addition to measurement accuracy and precision, safety is another major concern related to charge soakage in capacitors. In the case of high voltage/high power systems, capacitors must often be discharged for long periods of time before they can be safely handled. Therefore, minimization of this effect is an important criterion in the selection of dielectric materials by capacitor manufacturers. Some common modern dielectrics include mylar, silver mica, and ceramic but for better absorption performance higher-cost dielectric materials such as polypropylene or PTFE (Teflon) can be used.[8,14]

An absorption-compensating mathematical model of a capacitor has been described using a succession of RC circuits connected in parallel with the original capacitor.[15] The RC circuits, referred to as "compartments," mimic dielectric absorption behavior by representing delayed release energy storage elements. This compartmental model describes the total current *i(t)* involved in recharging a capacitor due to dielectric absorption as an infinite sum of decaying exponential currents according to:

$$i(t) = \sum_{i=1}^{m} I_{j0} exp\left[-\frac{t_{sc}}{\tau_j}\right] exp\left[-\frac{t}{\tau_j}\right] \quad (1)$$

Where the resulting voltage on the capacitor is described by:

$$V(t) = \frac{|V_C - V_s|}{C} \sum_{j=1}^{m} \frac{\tau_j}{R_j} exp\left[-\frac{t_{sc}}{\tau_j}\right]\left(1 - exp\left[-\frac{t}{\tau_j}\right]\right) \quad (2)$$

In these equations, V_C and V_s are the charge voltage and short voltage respectively. *Short voltage* is the voltage to which the capacitor has been clamped for discharging. C represents the overall capacitance, m is the number of parallel RC compartments, τ_j is the decay rate of each compartment, R_j is the resistance in each compartment, t_{sc} is the capacitor short duration, and t is the time elapsed since the short condition was removed. It should be noted that values specific to each compartment, namely R_j and τ_j are not, and cannot be explicitly measured—they are simply useful as a model for conceptualization.

Dielectric Sensors

Sensor applications based on other dielectric behaviors have been applied successfully in several ways. One common use for dielectric technology is in the design of gauges for monitoring the fluid level in a container, such as diesel fuel, hydraulic fluid, or motor oil. Two long parallel plates coated in a nonconductive material are placed vertically into the container with a small gap between them to form a variable capacitor. When empty, the air in the tank serves as the only dielectric material forming a capacitor between the two parallel plates. As the fluid (oil, fuel, other) fills the tank it also progressively occupies a larger percentage of the volume between the plates. It thus replaces air as the dielectric material in the liquid-level sensing capacitor in the tank. In general, air has a lower dielectric constant than the fluid being measured so the capacitance between the plates rises as fluid is

added to the container. The resulting change in capacitance is easily measured, and by this means a liquid-level sensor with no moving mechanical components can be realized. Differential capacitance probes have also shown promise in application to determining relative compositions of liquids.[16]

Another sensor implementation is dielectric spectroscopy, which generally refers to the observation of dielectric behavior as a function of applied frequency. One use of dielectric spectroscopy relevant to biological systems is a method of determining biomass concentration in suspended cultures of microorganisms using a capacitive probe.[17,18] The process works by selecting a capacitor charge frequency based on the dielectric dispersion principles of biological cells.[17–20] Conductivity and permittivity data, obtained by monitoring conductance of the sample and capacitance of the probe, can be used with a series of equations[17] to determine the volume fraction of the sample that is inhabited by cells.

While applications of dielectric-based capacitive sensors are widely varied, sensors based on direct current (DC) dielectric absorption characteristics, to the knowledge of the authors, have not been explored. The DC measurement of characteristic dielectric absorption behavior of materials has certain advantages, including the fact that the phenomenon can be observed and measured without the need for careful selection of capacitor charging frequencies and with otherwise relatively simple electronics. This absorption and relaxation information can be obtained in the frequency range of around 10Hz to an approximate DC dielectric response. Potential applications of a sensor based on dielectric absorption to provide rapid determinations of bulk sample characteristics are widely varied and are not explored fully herein. Preliminary experiments described in this manuscript suggest one

possible application of this method in the rapid assessment of oil content in a capacitive probe.

Algae derived biofuels

The potential contribution of algal oil to future economic growth goes well beyond its value as a replacement for current petrochemical liquid fuels. To fully replace petro-oil (a finite and non-renewable resource) in the global economy with any renewable resource, consideration must also be given to the production of chemical substrates into the global material commodities stream. The current petrochemical industry provides massive amounts of material that is employed in all industries, especially plastics and agricultural (fertilizer), and energy alternatives such as wind and solar simply do not produce concurrent streams of useful materials, thus they can never fully replace petrochemical resources. Algal oil, on the other hand, can be modified to replace most, perhaps all petrochemical-derived material substrates. For this to become a practical reality, it will be necessary to monitor and control not only the total mass of algal oil being produced in large algal fields, but also to monitor and control the quality and types of oil being produced. The richness of data obtainable from dielectric absorption may allow the necessary information to be gathered rapidly from industrial-scale algal oil producing installations.

Assessment of oil productivity, such as the mass of oil per unit volume of algal cultures on an industrial scale, is of critical importance to algae-derived bio-fuels research. The oil content of the cells is both strain-specific and highly dependent on environmental growth conditions.[21,22] Genetic and metabolic engineering are also likely to have a significant effect in the future on oil content in algal cultures. Metrics developed for this type of

measurement are mainly limited to theoretical yield calculations based on multiple efficiency assumptions[23] and experimentally determined weight percentage of algal lipids versus dried biomass. Though theoretical approaches provide a valuable means for estimation of algal oil yield possibilities, actual oil productivity measurements must be collected to facilitate the necessary incremental research in this field, and to allow large-scale industrial plants to accurately monitor and control the process of generating oil from large algae fields. Determining percent lipids by weight measurements are currently time and energy intensive and require relatively large samples of algae. A capacitive probe based on dielectric absorption may help fill this technological void in bio-fuels research.

1.2 Methods

System Overview

To enable low frequency and DC dielectric absorption measurements, we have selected five user defined parameters for data collection settings in this system: charge voltage, short time, collection time and number of data collection cycles (Fig. 1). One data collection cycle includes three modes of data collection. When the system is in charging mode, the positive terminal of the probe or subject capacitor is connected to the charge voltage. The charge voltage is disconnected and the

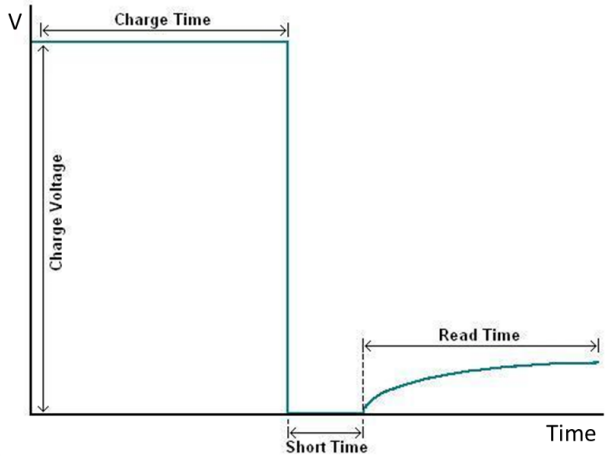


Fig. 1. Representation of a single data collection cycle and data collection parameters. The graph displays an example voltage on a capacitor over time in this system. All parameters are 16-bit values allowing user settings in the range of 0 to 65534. A charge voltage setting of 65534 will yield a 10v output in the circuit. Each time variable signifies the amount of time in milliseconds that the system will be in that data collection mode during each cycle.

capacitor is shorted during shorting mode. Finally, the system transitions into collection mode where the capacitor charge is building and is connected exclusively to the data digitization circuitry.

The data collection settings are specified in the system software where they are automatically logged at the top of the data file and sent to the system hardware (Fig. 2) for execution. A green LED on the circuit indicates that the system is ready to start a new dataset.

Electronic Data Collection System

The system described here employs a method for measurement of this property adapted from a design by Robert Pease wherein analog switches and a strip chart recorder were used to characterize some absorption behaviors of different dielectrics in commercial capacitors.[8] The present system acquires absorption signals using solid state electronics and digital methods (Figure 2).

This system is capable of generating DC dielectric soaking signals ranging from zero to ten volts. A 16-bit digital-to-analog converter (DAC8411) is set using SPI communication from a PIC18F4550 microcontroller. The output of the DAC is then doubled using a non-inverting amplifier operational amplifier circuit (LF444CM) to increase the voltage range and the available current of the charging signal.

A set of three MOSFETs (IRF7341, IRF7343) and a relay (HE3621A051) are used to set the circuit in charge, short, or collection modes. The amplified charge signal is connected to the source pin of a P-Ch MOSFET. The P-Ch is activated when the voltage on the gate pin is at least one volt less than that of the source. This means that at source voltages of six volts

or more, the P-Ch cannot be controlled by a microcontroller digital I/O pin directly. In light of this constraint, a N-Ch MOSFET serves as an open-collector interface between the microcontroller and the gate of the P-Ch. Thus, activation of the N-Ch grounds the gate of the P-Ch which connects the charge signal to the relay. To short the capacitor, the open-collector N-Ch is deactivated and a separate N-Ch is activated which grounds the relay. The relay is in place to allow for complete disconnection of the capacitor from the charging/shorting circuitry during collection mode to eliminate data aberrations from transistor switching noise. Data is digitized and collected during all modes of operation.

Following the momentary short of the capacitor terminals, the system is transitioned into collection mode. The signal is buffered with a voltage follower op-amp circuit to isolate the collection circuitry from the signal with a high input impedance to minimize distortions.

Three 16-bit ADCs (TLC4545) allow for three parallel options for signal conditioning. All signal conditioning circuits are electrically isolated from one another using voltage followers. These ADCs work on the range from zero to five volts which allows the first ADC to collect voltages five volts or lower over the full 16-bit resolution of the ADC. The second ADC has a one half voltage divider following the buffer for digitizing voltages up to ten volts. The third acquisition option is an analog differentiated signal. This is in place to offer a measurement of the signal current (first derivative of the voltage). In order to make the differentiated signal comply with the input range of the ADC, a combination of a summing amplifier and an inverting amplifier provides a +2.5V offset. The ADCs are interfaced via SPI communication from the microcontroller. All three clock lines connect to one I/O pin and the chip select lines connect to another. This allows for simultaneous

sampling of all three data collection options. Only the 10V collection method was used in data presented here.

Two PIC18F4550 microcontrollers, labeled I and II in the (Figure 2) schematic, were used in the hardware design of this system; referred to as PICI and PICII henceforth. All microcontroller code was written using the PICC compiler by Custom Computer Services (CCS). PICI acts as a USB intermediary between PICII and the graphical user interface (GUI) generated in VisualBasic 2010 (Figure 2). USB communication on these PICs is

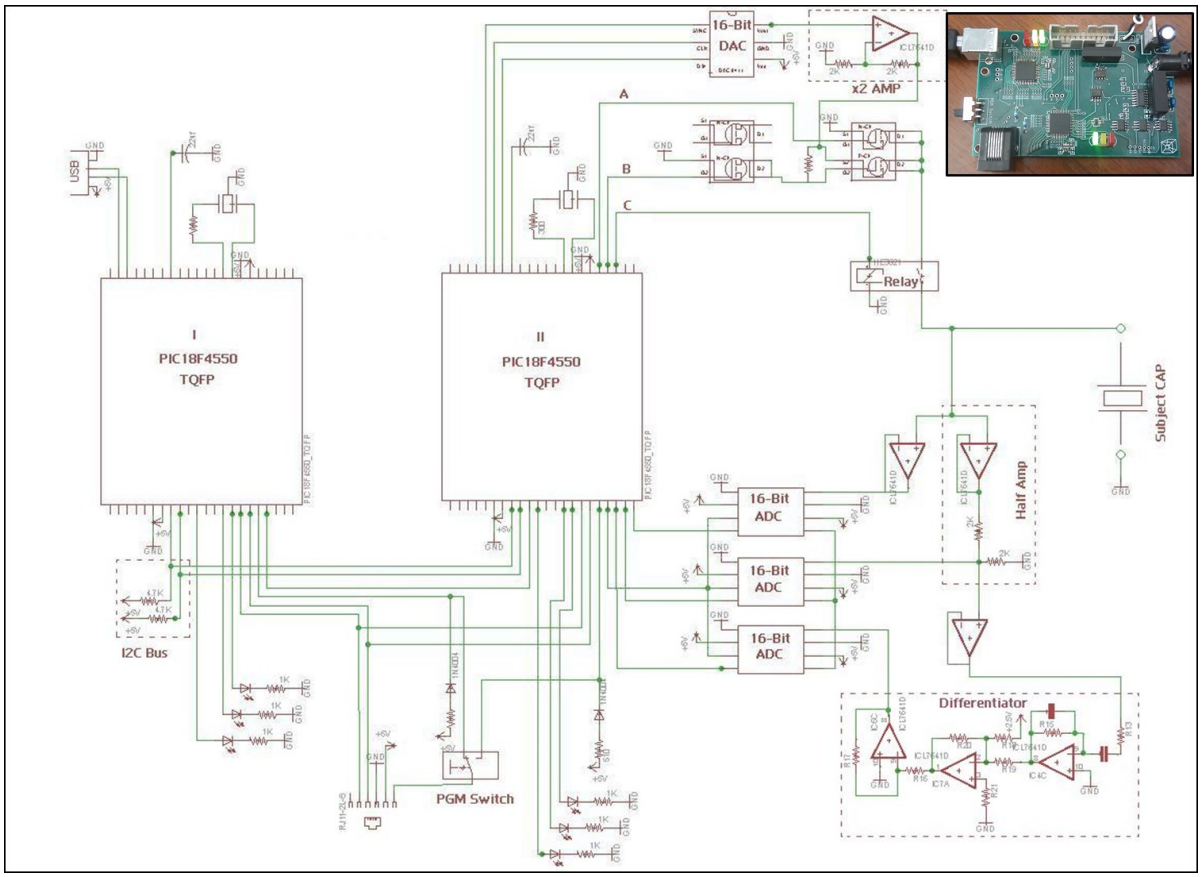


Fig. 2. Circuit schematic for generation of charge signals and acquisition of data. Created using EAGLE Layout Editor CAD software. Experimental parameters set in software control the behavior of the circuit. Three 16-bit ADCs are used to provide options of reading the raw signal, the signal halved (for values in excess of 5V), and/or a differentiated signal. (Inlay) System hardware. The printed circuit board (PCB) was created using ExpressPCB development software.

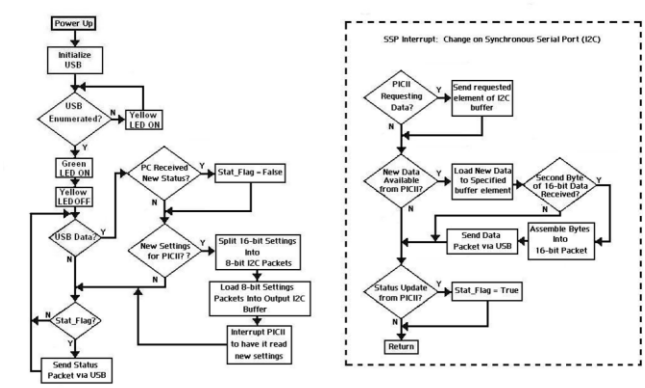


Fig. 3. Program flow diagram for the USB intermediary chip (PICI). PICI acts as the I2C slave. The SSP interrupt triggers when the master is sending data to the slave or requesting data from it. When data is received from the master device, it is sent to the computer through USB under the Human Interface Device (HID) protocol. Conversely, when USB data is received from the computer PICI triggers an external interrupt on the I2C master device (PICII) to tell it to read the new settings.

relatively demanding of microcontroller resources, resulting in undesirable design constraints when attempting to implement all the necessary system firmware on a single chip.

Inter-Integrated Circuit (I2C) was selected as the communication protocol between the PICs for its speed, reliability, and modularity in adding multiple slave devices to the same clock and data lines. Generally, in I2C communication the master device (PICII) interrupts the slave device (PICI) whenever it has data to send. For data transfers in the opposite direction however, the master device has to request the data from the slave and generate clock cycles to receive it. In order to achieve seamless bidirectional data flow in this system, an additional digital connection line was added between the PICs wherein the slave can trigger an external interrupt on the master device to notify it of newly

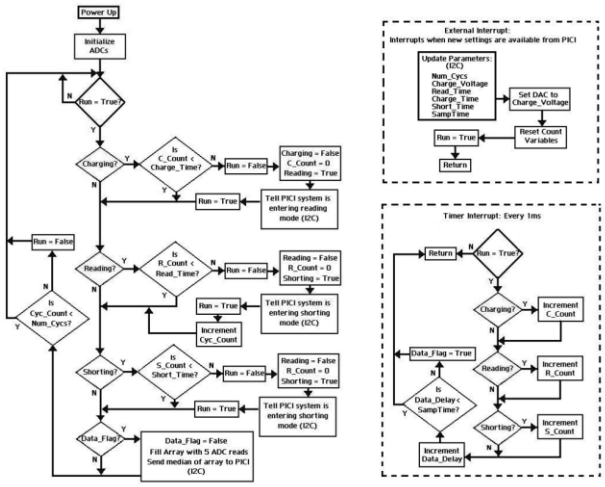


Fig. 4. Program flow diagram for PICII, the microcontroller that directly controls data collection parameters. This chip interfaces the ADCs, the DAC, and the set of transistors and relay that switch between charging, shorting, and reading modes. Acting as the I2C master device, this chip interrupts the slave device at each data interval for updates. The median of five rapid ADC reads constitutes a single data point to be recorded.

available settings from the GUI. PICI and PICII program flow diagrams can be seen in (Figure 3) and (Figure 4) respectively.

The design specifications require communication of 16-bit collected data and settings. One consideration of many data transfer protocols, including I2C, is an 8-bit packet size per transfer. Well established methods using binary rotations and firmware implementations of logic gates accomplish the splitting and reassembling of the 16-bit integers on each microcontroller.

Signal noise was significantly reduced with the introduction of a preconditioning median filter on PICII. Five rapid reads of the ADC into a data array comprises one data point to be collected. The value recorded is the middle element of the bubble-sorted array (median).

Probe Design

A simple parallel plate capacitive probe was designed and constructed for collecting data from varying volumes of soybean oil (Figure 5). The base plate is a 10.16 x 12.7 x 1.27cm (4 x 5 x 0.5in) block of aluminum and the top plate is a 7.62 cm (3in) diameter by 1.27cm (0.5in) thick cylinder. The circular plate shape was selected to even out edge effects due to electric field concentrations. The plate surfaces facing one another were machined flat using a 1.27cm (0.5 in) flat end mill. The sample was placed between two sheets of Teflon FEP, each 0.127mm thick (McMaster-Carr:

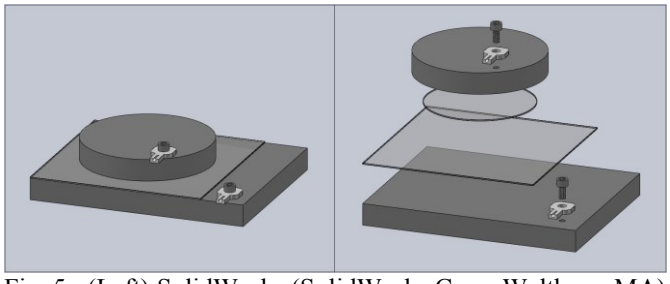


Fig. 5. (Left) SolidWorks (SolidWorks Corp: Waltham, MA) representation of probe and (Right) exploded view

85905K66). The bottom sheet is a 10.16 cm (4-in) square and the top sheet is a 6.35cm (2.5-in) diameter circle**.** The PTFE circle is smaller than the top plate so that surface tension can keep the liquid sample from leaking out of the probe. The plates were electrically coupled to the circuitry using #6 socket screws. The probe was housed in a grounded faraday cage during all data collection.

Signal Feature Selection for Sensor

In order to compare absorption responses in differing dielectric samples, specific features of the signal must be selected. Ideally for creating sensors based on this phenomenon, the selected features will contain the maximum amount of absorption information specific to the sample.

We have selected two features of the dielectric dipole relaxation response for comparison between samples: maximum recharge voltage V_{max}, and the length of time it takes to reach 63% of the maximum voltage. When discussing capacitors, the time constant of a capacitor can be represented as RC, a value whose product represents a time. A useful method for measuring the time constant of a capacitor can be produced by observing that at time t = RC, the voltage (V_{RC}) on a charging capacitor is:

$$V_{RC} = V_{max} * (1 - e^{-1}) \sim 0.63 * V_{max} \quad (3)$$

Where V_{max} represents the driving voltage on the capacitor. Thus, by knowing the length of time required to reach $0.63*V_{max}$, one can infer the value of RC. As the Iorga model[15] is based on an infinite sum of parallel capacitors, it is logical that a useful metric for gauging

the material properties of a dielectric would be the value of a "bulk time constant." In our case, we have named this bulk time constant $t_{63\%}$ and it is derived as follows:

An explicit equation for dielectric absorption V_{max} can be found by taking the limit of equation 2 as time goes to infinity.

$$V_{max} = V(t) = \lim_{t\to\infty} \frac{|V_C - V_s|}{C} \sum_{j=1}^{m} \frac{\tau_j}{R_j} exp\left[-\frac{t_{sc}}{\tau_j}\right]\left(1 - exp\left[-\frac{t}{\tau_j}\right]\right) \quad (4)$$

Which for shorting voltage, $V_s = 0$:

$$V_{max} = \frac{|V_C|}{C} \sum_{j=1}^{m} \frac{\tau_j}{R_j} exp\left[-\frac{t_{sc}}{\tau_j}\right] \quad (5)$$

The 63% point of the response curve can be defined as the time where:

$$V(t_{63\%}) = 0.63 * V_{max} \quad (6)$$

Substituting equation 5 into equation 2 and simplifying gives:

$$0.63 = \frac{\sum_{j=1}^{m} \frac{\tau_j}{R_j} exp\left[-\frac{t_{sc}}{\tau_j}\right]\left(1 - \exp\left[-\frac{t_{63\%}}{\tau_j}\right]\right)}{\sum_{j=1}^{m} \frac{\tau_j}{R_j} exp\left[-\frac{t_{sc}}{\tau_j}\right]} \quad (7)$$

Thus, V_{max} can be expressed as a function of the initial charge voltage and the overall capacitance of the probe. $T_{63\%}$ on the other hand, can be expressed independently of charge voltage and probe capacitance. This suggests that $T_{63\%}$ may be largely independent of area of

the capacitive plates, the distance between them, and the dielectric constant of the material. These observations supported the decision to use these two features for the sensor.

Data Description and Analysis

Software developed in Matlab (Mathworks, Inc: Natick, MA) first breaks down the dielectric relaxation signals received as a tab delimited file from the acquisition system. The software implements a rank 3 median filter to the raw voltage signal and extracts the selected features ($T_{63\%}$ and V_{max}) for analysis.

To demonstrate an application of dielectric absorption to sensor technologies, data was collected on precise volumes of soybean oil (Fischer Science: 8001-22-7) in order to generate an SVM calibration file. The 10.16 cm square PTFE sheet was placed on the base capacitive plate. Samples of soybean oil were pipetted onto the center of the sheet using an Eppendorf micropipette. The circular PTFE sheet was placed over the oil droplet and the oil was allowed to spread between the sheets. The cylindrical plate was placed concentrically over the circular sheet and the faraday cage was assembled around the probe. Data collection parameters were set in the GUI as follows: charge voltage (10v), charge time (65sec), short time (10ms), collection time (65sec). All data presented in this paper used the same parameter settings. The PTFE sheets were cleaned with soap and water and dabbed dry with a kim-wipe between samples.

The above steps were followed for 0µL,10µL, 20µL, 30µL and 40µL of soybean oil. Additionally, 25 cycles of a 45µL sample were collected to supplement the training file. The SVM calibration file created from this data included known sample volumes as well as extracted features from 60 cycles of each of 0µL, 10µL, 20µL, 30µL, 40µL. This data

collection procedure was repeated twice more; once for SVM parameter optimization and once for the SVM test.

Support Vector Machine

A support vector machine is a very powerful tool for pattern recognition and data classification in computers.[24,25] SVMs were selected for data analysis because they can learn relationships very quickly and accurately from few samples and because they have few parameters to optimize. In this study, the SVM^{light} implementation[26,27] of Vapnick's SVM[28] was employed using Matlab. This package allows the user to train an SVM and to test it using features from data it has not been trained on. The radial basis function kernel was employed in regression mode for this experiment with an empirically optimized hyperplane complexity parameter of 4.2.

The application of an SVM to this experiment involved regression of volumes of soybean oil based upon their respective $T_{63\%}$ and V_{max} values. After the SVM was trained on the training volume dataset, the parameters were optimized for a second dataset. The optimized parameters were then tested using features from a final third dataset. The SVM output is a vector of estimated volumes based upon the input features. The SVM output was plotted against the known sample volumes of the third dataset test file and the correlation coefficient was calculated (Figure 8).

1.3 Results

The processor of the PIC18F4550 provides high-speed transfer rates and mathematical functions in working with 16-bit data. Each microcontroller is operating at 24.00 MHz in this system. At this clock speed, the system hardware is capable of sampling at a rate of approximately 2.2 KHz. Each successive sample includes reading the ADC five times and loading it into an array, sending the median of the array via I2C, and sending that value to the computer via USB. The hardware sample rate can be increased to 11 KHz if the median filter is removed but the sample rate limitation in this system is not the hardware, but the USB data receiving software.

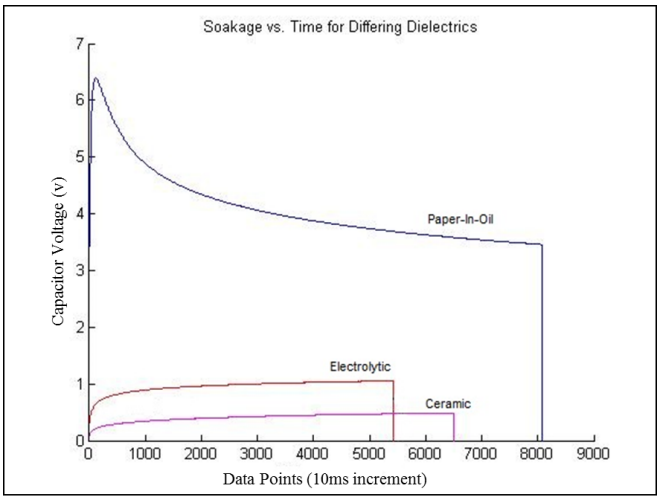


Fig. 6. Example data collected with this system on commercial capacitors with differing dielectric materials. In this figure, the ceramic capacitor data was collected for 65 seconds at a sample rate of 100Hz. The electrolytic and paper-in-oil (PIO) capacitors were sampled at rates higher than the GUI software can accommodate for read times of 45000 ms and 65534 ms respectively. Dividing the read time by the number of data points collected yields a sample rate of about 125 Hz in both cases.

Based upon experimental testing, the Human Interface Device USB template used in development of the GUI can only receive values from a peripheral device at a maximum rate of 125 Hz (Fig. 6). This limitation introduces some ambiguity in the time scale of the data being collected at high sample rates. Time between successive data points must be consistent

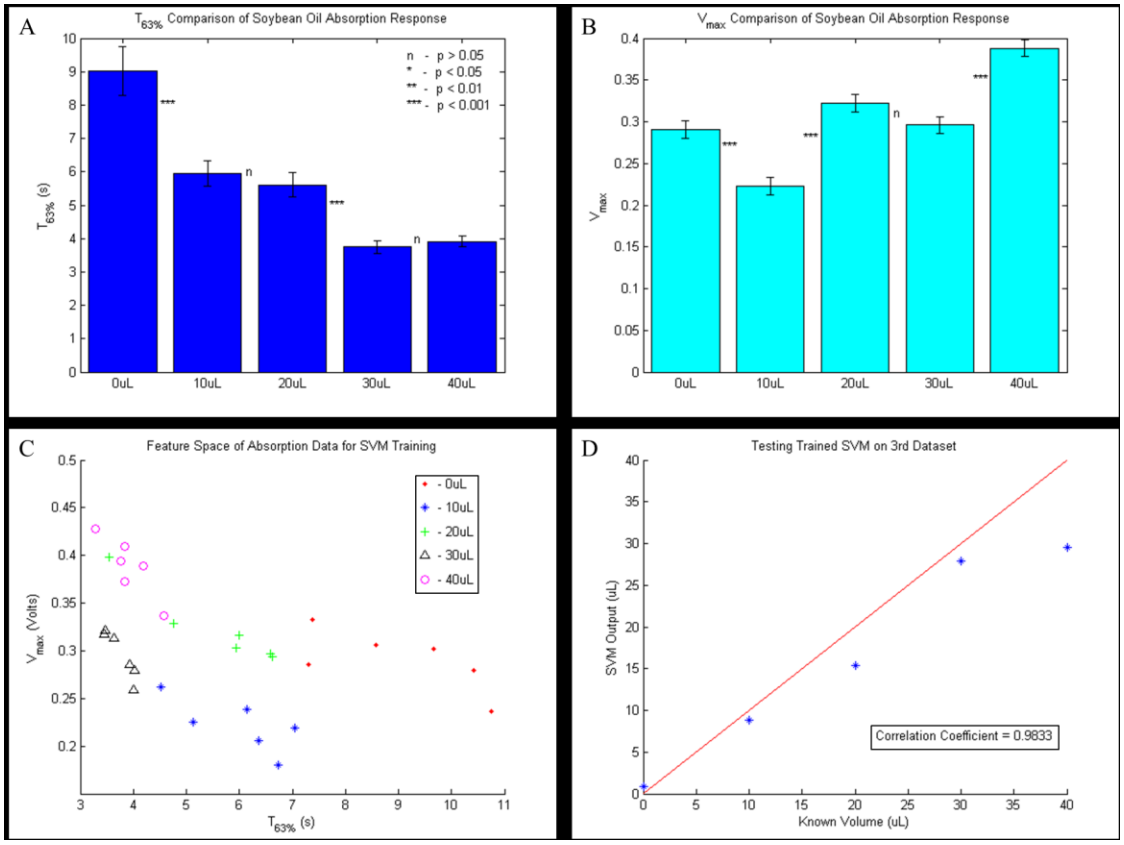


Fig. 7. (A&B) Summary of means of (A) $T_{63\%}$ and (B) V_{max} sample data collected for training the support vector machine. The error bars represent standard error. Results of two-tail t-tests are displayed for comparison of nearest volumes. (C) SVM training data presented in feature space (D) Results from testing the trained SVM using inputs: $T_{63\%}$ and V_{max} features from a new dataset. Each input feature was an average of 20 data collection cycles at each volume. The graph depicts SVM volume predictions versus known volumes. The line in the graph represents an ideal SVM result. The correlation coefficient between the known volumes and the SVM estimated volumes provides a measure of accuracy.

and accurate for analysis methods to work properly. In order to ensure operation in the range of reliability, values are sent to the computer at a rate of once every 10 ms (100 Hz) which provides manageable data file sizes and is still an order of magnitude greater than Nyquist rate for the signal being sampled.

The data collection system was tested successfully on a variety of commercial capacitors with differing dielectric materials. The absorption response of a ceramic, electrolytic and a paper-in-oil capacitor is depicted in (Fig. 6). The difference in responses of these capacitors can be observed without difficulty. The decaying voltage on the paper-in-oil

capacitor is attributed to a less than ideal series resistance or “leakage” of the dielectric material.

The dielectric absorption data collected for training the SVM was separated into $T_{63\%}$ and V_{max} features. A summary of the means of the features at each volume in a sample dataset can be seen in (Fig. 7 A&B). The results of two-tail t-tests performed between neighboring volumes are also displayed. In this dataset, the null hypothesis was rejected for most neighboring volume feature means but could not be rejected for three of them. For example, according to the t-test between the $T_{63\%}$ values of the 10µL and 20µL samples the difference between the mean $T_{63\%}$ values is not statistically significant. However, it should be noted that the result of the t-test between the V_{max} values of the 10µL and 20µL samples yields a p-value less than 0.001. Thus, in this dataset, where one feature fails to reliably discriminate between samples, the other can accomplish the task easily.

The SVM training dataset is represented in feature space in (Fig. 7C). Each point represents an average of 10 individual data collection cycles. The data is observed to generally cluster by volume with some cross-over between clusters. A support vector machine is ideally suited for separating the clusters for classification by features.

After training, the SVM was tested using features from a separate dataset. The results of this test can be seen in (Fig. 7D). The correlation coefficient between the SVM output vector of estimatedvolumes and the known volumes of the test data was calculated to be 0.9833. This means that the SVM was able to successfully classify volumes of soybean oil based on their $T_{63\%}$ and V_{max} input features even at low sample sizes (<50µL). This SVM was trained on 300 individual cycles of data where each point in the dataset was an average of 10 cycles.

1.4 Discussion

Through the use of a SVM, a soybean oil volume sensor was successfully demonstrated based on the concepts of dielectric absorption and relaxation. This result suggests that DC dielectric absorption signals provide enough information about bulk material characteristics to classify samples in sensor applications. One particular advantage that this DC dielectric relaxation response sensor has over sinusoidal systems is overall simplicity of signal acquisition and analysis.

By generating a SVM training dataset using samples of algae and their respective lipid contents, the system described in this manuscript could be purposed as a device for rapid field sample measurements of lipid concentration in algae farms. Algal oil content could also be cost-effectively tracked daily for selecting the maximal yield harvest time. The device could be calibrated to specific sample populations of algae to maximize accuracy.

A similar process can be implemented in the creation of sensors for many applications. Possibilities may include a tool for improvement and testing of industrial capacitors, detection of concealed materials within objects, or even a food diagnostic tool for relative determinations of fat content.

The I2C network on the system hardware allows for individual microcontrollers to operate in parallel which is an important feature at this stage for future design implementations. For the data presented in this paper, the signal feature extraction and SVM analysis was all performed using a computer. These methods could be implemented on microcontrollers embedded in the system hardware; thus eliminating the need for a computer and allowing for hand-held sensor applications. More advanced systems could be created for wireless monitoring and data logging on a base computer. Exploring other analysis options in

place of SVMs is another area for future research including but not limited to artificial neural networks (ANNs).

1.5 Conclusion

This paper describes hardware and software methods for measuring low frequency dielectric absorption properties of materials. A parallel plate probe design is detailed and demonstrated in the application of collecting data from varying small sample volumes of soybean oil. The V_{max} and $T_{63\%}$ features of the relaxation signal were shown to contain enough information about the sample for the system to be successfully applied as a soybean oil volume sensor. Finally, a support vector machine was demonstrated to be capable of "learning" non-linear relationships between data clusters created from these features.

CHAPTER 2: A CYCLIC STRAIN BIOREACTOR FOR MECHANICAL ENHANCEMENT AND ASSESSMENT OF SELF-ORGANIZED MYOCARDIAL PATCHES[2]

2.1 Introduction

In 2008, approximately 1 out of 6 deaths in the United States was caused by coronary heart disease.[29] The human heart is a complex organ that requires near perfect timing and operation of many intricate parts in order to function properly. The myocardium consists of three major types of cardiac muscle fibers: atrial muscle, ventricular muscle, and excitatory and conductive muscle. The fibers are arranged in two distinct latticeworks generally referred to in the literature as the atrial and ventricular syncytia.[30] Under normal conditions, the excitatory heart contraction signal is generated at the sinoatrial (SA) node and spreads diffusely through the atrial syncytium causing atrial contraction. The excitatory signal reconvenes at the atrioventricular (AV) node from which it spreads throughout the ventricular syncytium and causes the systolic stroke of the heart pump. Fibrillations in this stepwise rhythm are commonly caused by ischemic conditions of sections of the syncytia where coronary arteries are narrowed or blocked.

While heart has been shown to contain populations of stem cells[31–35], the average rate

---

[2]This chapter was recently submitted to IEEE Transactions on Biomedical Circuits and Systems. The citation is as follows:
Cashion, A. T., Salazar, B., Birla, R. & Dennis, R. G. A Cyclic Strain Bioreactor for Mechanical Enhancement and Assessment of Self-Organized Myocardial Patches. (2013).

of regeneration is estimated to be less than 1% turnover annually[36]. This inadequate regeneration means that the key to the most effective treatment of myocardial infarction remains in limiting tissue damage by minimizing the time of the arterial clog. Thrombolytics, blood thinners, vascular stents and bypass surgery are all effective treatments of the clogged arteries but the necrotic myocardial tissue is most often permanently damaged. Ventricular assist devices can be used to aid the heart pump but the gradual build-up of non-contractile scar tissue will eventually lead to heart failure.[37] Currently, heart transplantation is the only standard clinical practice in place to treat infarcted tissue.[32] Stem cell transplantation is a logical research endeavor[32] but clinical trials of non-scaffold cell injection have not yet led to significant recovery of cardiac function.[37] Other experimental approaches include heart geometry modifications with biomaterial restraints[38–42], in-situ injectable scaffolds with cardiac cells[43–47], and cardiac muscle patches cultured in-vitro and implanted on the epicardial surface of the heart[48–53].[54]

The gold standard in myocardial tissue engineering is to create clinically relevant autologous grafts for damaged tissue replacement.[37,55,56] In order for an engineered patch to successfully replace a section of the myocardium, it must be approximately 1cm thick, conduct electrical signals, and produce contractive forces similar to adult phenotype tissue.[37,57] Vascularization is necessary when developing tissues thicker than ~200μm;[58] so in order to stack thin myocardial patches, angiogenesis paradigms must be explored.[55,59–62] The road to engineered replacement myocardium includes a great deal of basic research into epicardial patch implants, scaffold manipulation, developmental interventions, and the organization of the tissues themselves.

Mechanical stimulus has been shown to enhance the function of myocardial patches[63–67] but has not been exhaustively explored. A review of the literature has led the authors to determine that the field has matured to a point where it would benefit from the development of more capable instrumentation.[68] The work presented here centers on the effects of cyclic mechanical strain on self-assembled patches of cardiac tissue developed in-vitro without the use of a scaffold.

2.2 Methods

Theory of Bioreactor Operation

Strain is applied to the cardiac muscle constructs using a very low friction and low hysteresis linear voice coil actuator (VCA) apparatus similar to one described elsewhere.[69] The principle of our VCA system is depicted in figure 1A. Lorentz forces are generated orthogonal to the direction of the static magnetic field and to the direction of charge flow through a suspended wire.[70] The force magnitude is proportional to the total length of wire (l), the magnetic field strength (B), and the electrical current (i) as is represented by the following cross product:

$$F_{Lorentz} = il \times B$$

Coiling the wire increases the total length within the magnetic field. The two sides of the coil are suspended in opposite static magnetic field directions to accomplish a unilateral net force direction. Thus, the magnetic field strength and the length of wire are set initially and maintained constant which makes current the only variable governing the Lorentz force magnitude and direction.

The coil of wire is suspended in the magnetic field by a flexible ABS armature which acts as a spring under elastic deformation. In static conditions the spring force exactly balances the

Lorentz force which allows for precise linear control of the coil position by modulating electrical current.

Figure 1B shows a SolidWorks representation of a frictionless sensor we modified from a previous design[71] to monitor position in this system. Two infrared light sources are directed at two detectors of the relevant wavelength. An interrupter attached to the VCA armature moves within the sensor space proportionally to the movement of the end effector. Position is accurately tracked by observing the differential activations of the two detectors.

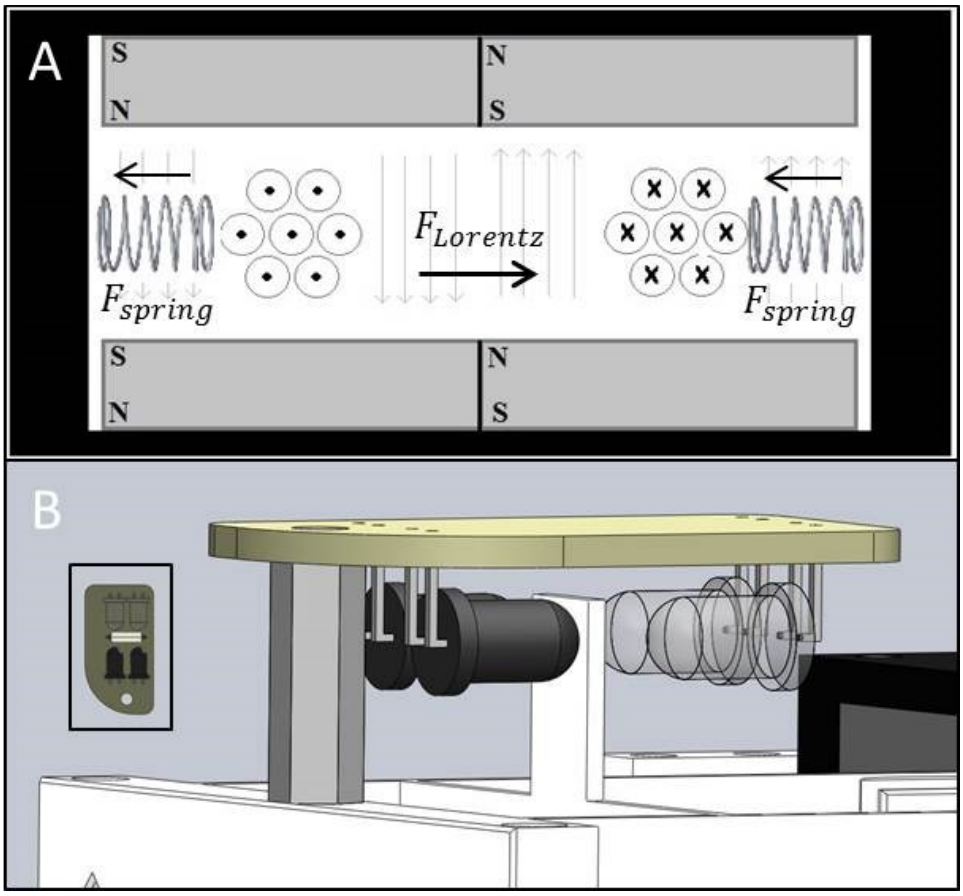


Figure 1. (A) Cross-sectional conceptual drawing of the voice coil actuator. The direction of current in the coil is depicted conventionally. The springs in this figure represent the effect of the flexible ABS frame. **(B)** Optoelectronic differential position sensor

Bioreactor Mechanical Development

The SolidWorks design and assembly as well as an image of the VCA bioreactor are presented in figure 2. The mechanical design was originally developed in SolidWorks in order to preserve time and materials by most effectively eliminating clearance errors prior to construction. After the software design phase, rapid manufacturing techniques were employed to generate the physical bioreactor.

The baseplate was laser cut from a 0.75cm thick piece of stock acrylic. During laser cut manufacturing, holes were placed in appropriate locations for mounting the electronics, the VCA armature, and the magnet housing. Additionally, a circle was cut near the end effector for snug placement of a standard 35mm culture plate.

The VCA armature was designed to elastically deform in such a way that allows very near uniaxial movement at the end effector. The action of the resulting mechanism is similar to a spring loaded four-bar linkage. The armature design includes an appropriately sized slot for the coil bobbin to snap in place. A SW displacement study was executed to test clearances and to set the proportional movement of the beam interrupter to the end effector (figure 2A). The bobbin and the armature were 3D printed out of ABS plastic and mounted to the acrylic baseplate. A 150 turn coil of 36-AWG wire was wound around the bobbin and then snapped into the armature slot.

The magnet housing was constructed by machining low carbon flat-stock steel. A set of four 1.9cm square neodymium N42 rare earth magnets were used to set up the static magnetic field within the housing. After assembly around the armature, the magnet housing was attached to the baseplate using steel 4-40 screws. The optoelectronic sensor is fit over the beam interrupter using a 1.3cm threaded nylon spacer.

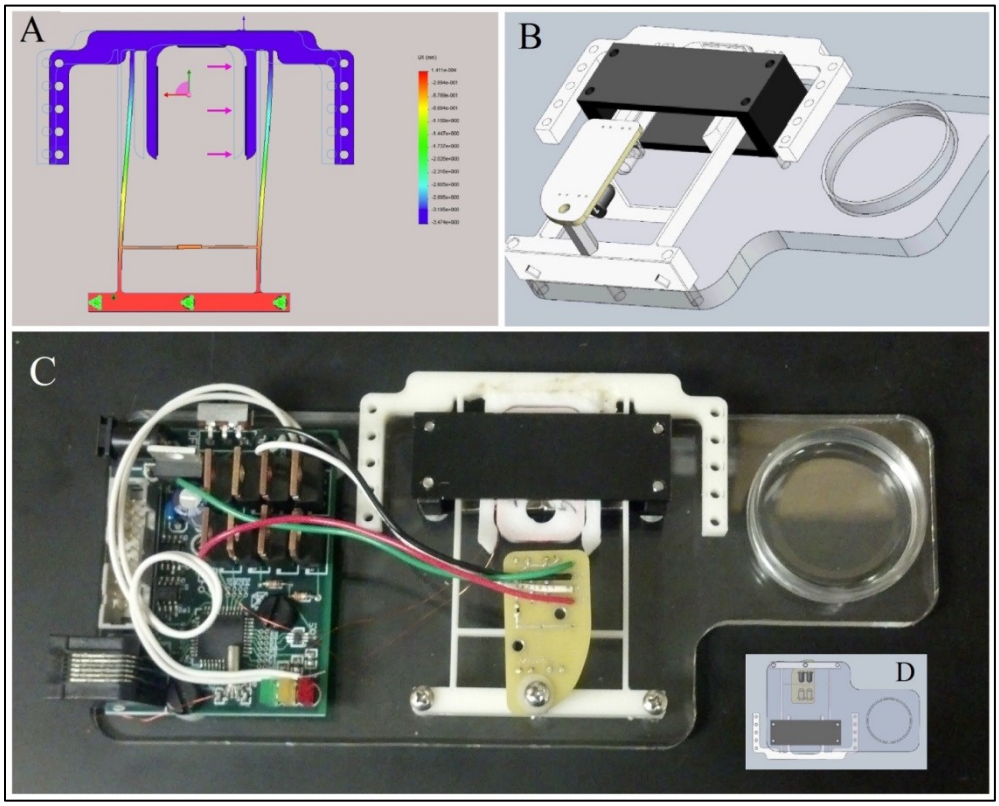


Figure 2. (A) SolidWorks motion study for the design of the flexible frame **(B)** Assembly of the VCA mechanical design **(C)** Image of complete bioreactor hardware **(D)** Inverted view of SolidWorks Assembly

Electronics

The schematic, printed circuit board design, and final image of the electronics developed for this system are shown in figure 3. The circuit is part of an inter-integrated circuit (I2C) digital network where it can be connected to other bioreactors, sensors, and a previously described custom data acquisition (DAQ) module.[72] The DAQ bidirectionally communicates via USB with a computer running a program written in VisualBasic 2010 for setting stimulation parameters. The DAQ acts as the I2C master device and can independently address each individual VCA bioreactor on the network for parallel experiments. The data presented in this manuscript used only one bioreactor. An embedded

PIC18F4550 microcontroller is used for digital communication, armature position monitoring, and stimulus generation.

The VCA is driven using a digital H-Bridge circuit composed of N-Ch (Digikey: IRLZ44) and P-Ch (IRF4905) MOSFETS. The microcontroller setting of the pins labeled A, B, C, and D in the figure 3 schematic controls the direction of current through the coil and therefore direction of armature movement. The magnitude of the coil current is controlled by a 16-bit digital to analog converter (Digikey: DAC8411) attached to the base pin of a NPN power transistor (Digikey: TIP3055).

The activations of each IR detector in the position sensor are represented by variance of a direct current voltage from 0 to 5 volts. In order to calculate the differential activation, two 16-bit analog to digital converters (Digikey: TLC4545) digitize each detector voltage. The difference between the voltages is calculated in the microcontroller firmware. For each position update, this process is repeated 5 times and the median difference is used.

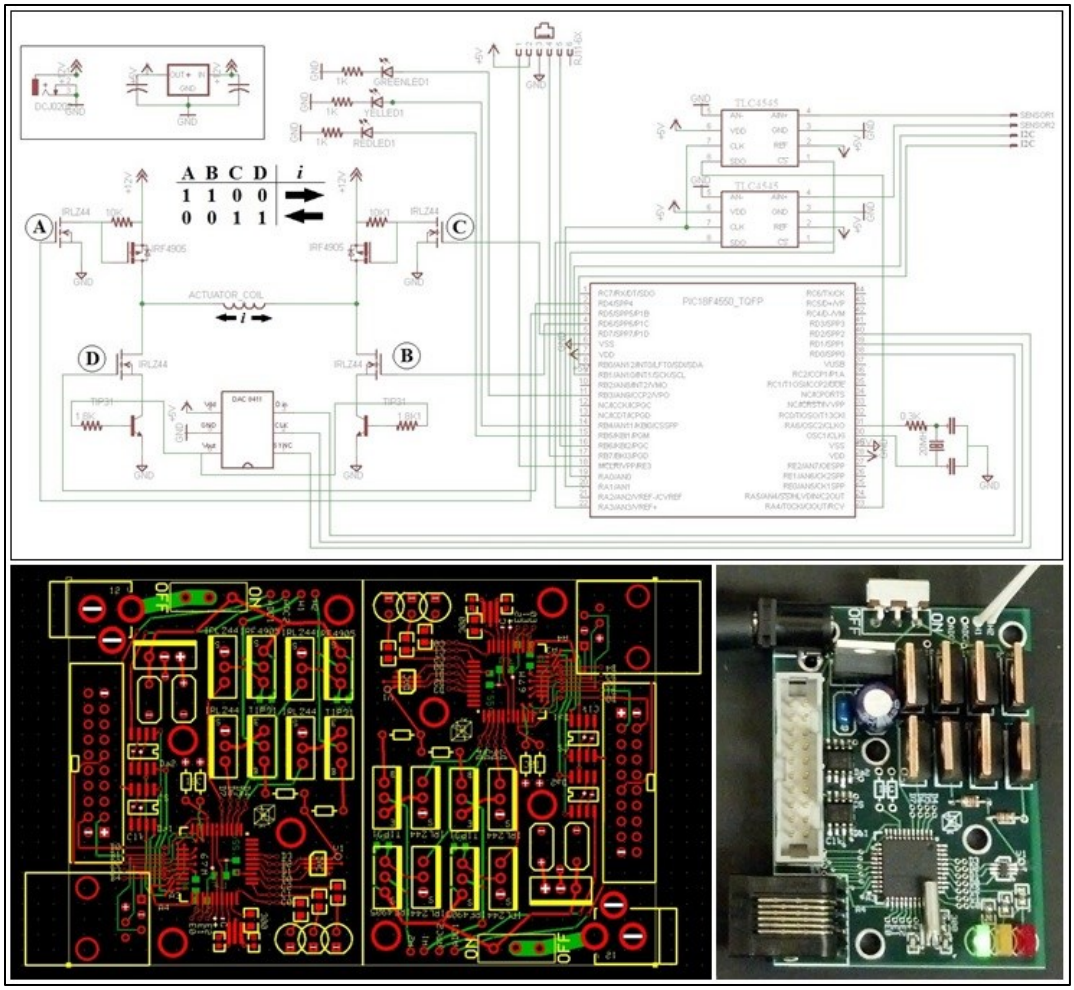


Figure 3. (Top) Circuit schematic of the VCA system **(Bottom Left)** PCB designed using ExpressPCB freeware **(Bottom Right)** Populated PCB

Calibrations

In order to relate the positional sensor signal with physical length, sensor increments were correlated with the displacements of a linearly variable differential transformer (Schaevitz Sensors: DC-SE1000). The resultant linear relationship from this calibration dataset yielded an R^2 value of 0.996 and a slope magnitude of 0.0097 which gives a sensor resolution of just under 1µm (figure 4A).

To calibrate the electrical current to Lorentz force relationship, the VCA armature was held firmly in the home position by a force transducer (Wagner model FDI: 5471885). Force measurements were recorded as coil current was gradually increased (figure 4B). The

r^2 value was 0.9994 and the slope was .0041 yielding a relationship of approximately 4mN per mA.

In each static condition, the spring force exactly balances the Lorentz force which means that information from the first two calibrations can be used to calculate the spring constant of the VCA armature (figure 4C). The VCA was allowed to move freely and the position sensor was monitored as coil current was gradually increased. The spring force to spring displacement relationship has an r^2 value of 0.9992 and slope of 72.49 which by Hooke's law is also the spring constant. Knowledge of the spring constant will be necessary when the system is used to measure cardiac patch twitch forces. Multiplying the spring constant by the positional sensor precision yields a force resolution in this system of approximately 70μN. This resolution could be optimized by minimizing the spring constant and also by increasing the proportional movement of the beam interrupter to the end effector.

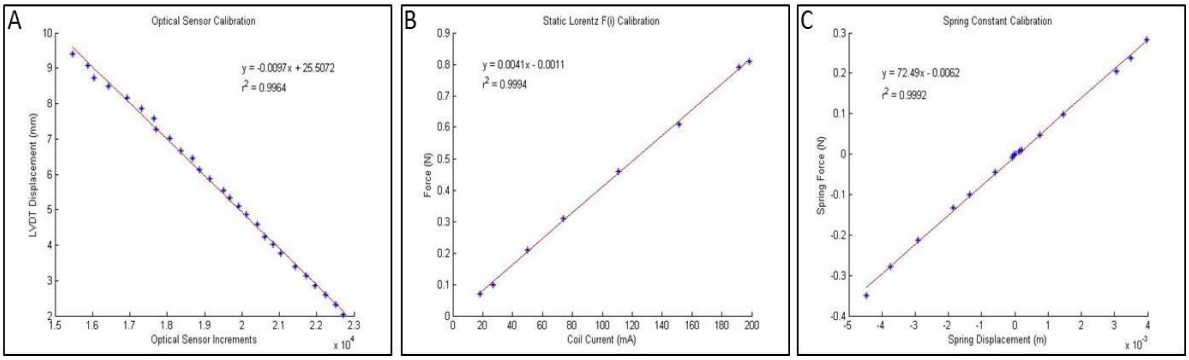


Figure 4. (A) Calibration of the position sensor against the linearly variable differential transformer (LVDT) **(B)** Calibration of the Lorentz force generated as coil current is varied. **(C)** Armature spring constant calculation using the calibrations of A and B.

Isolation of neonatal cardiac myocytes

Neonatal cardiac cells were isolated from the hearts of 2 to 3 day old Sprague-Dawley rats utilizing a pre-established method **REF**. Each heart was cut into 3 to 4 sections in ice-cold PBS phosphate buffer. All the pieces were gently rinsed to remove blood cells and thereupon

transferred to a secondary phosphate buffer solution for additional mincing. Tissues were minced into 1 mm^2 fragments and transferred to a dissociation solution (DS) consisting of 0.32 mg/mL collagenase type 2-filtered (Worthington Biochemical Corporation, Lakewood, NJ) and 0.6 mg/mL pancreatin in phosphate buffer. The minced tissues and 15mL of DS were placed in an orbital shaker for 30 minutes at 37°C and 60 rpm to initiate serial digestion. Once the 1st digestion was completed, the supernatant was collected in 3mL of horse serum in order to neutralize the enzyme and placed in a centrifuge at 1000rpm for 5 minutes at 4°C. The cell pellet was re-suspended in 5mL of horse serum and kept in an incubator at 37°C supplied with 5% CO_2. Fresh DS was added to the partially digested tissue and the procedure for the digestion was repeated 2 to 3 times. The cells acquired during the serial digestion were pooled, centrifuged and suspended in culture medium (CM). Cell viability was determined by Trypan blue (4%) staining according to the manufacturer's protocol.

Preparation of the plates

35mm tissue culture plates were coated with 2mL of SYLGARD (PDMS, type 184 silicone elastomer) (Dow Chemical Corporation, Midland, MI). The plates were air dried for 2 weeks to allow the silicone coating to form properly and sterilized with 80% ethanol before use. 4 minutien pins (Fine Science Tools, Foster City, CA) 0.1mm in diameter, were placed in the culture plate to form a 20mm X 20mm square.

Formation of the fibrin gel and cell plating

Fibrin gel was formed by adding 1mL of CM containing 10U/mL thrombin to the coated surface of each culture plate. Afterwards, 500µL of saline containing 20mg/mL fibrinogen was added. The culture plates were shaken well to ensure complete mixing of both solutions and complete plate coverage. Subsequently, the plates were placed in the incubator

to promote the formation of the gel within 45 minutes. Primary cardiac cells were diluted in CM at 2 million cells/mL and plated in 2mL of CM for each plate after the gel was fully formed.

The cells were cultured in an incubator at 37°C supplied with 5% CO_2 and underwent media changes every other day. The cardiac tissues were maintained for 7 to 10 days then loaded into the bioreactor for mechanical stimulation.

Attachment of cardiac tissue onto bioreactor

The fibrin gel tissue constructs were cultured in individual 35mm culture plates until the self-organization process was concluded. After the tissues constructs were fully formed, each specimen was loaded onto the bioreactor individually (figure 5B). First, the 35mm plate was placed into the bioreactor well. Second, two adjacent minutien pins were removed to allow stretching of the tissue construct. Third, the hooks attached to the bioreactor are placed into the tissue construct on the side where the pins were removed.

The hooks are composed of 20 gauge braided copper wire, soldered to 26 gauge enamel covered solid copper wire, which came into direct contact with the tissue at three equidistant locations. The attachment hooks were sterilized with 80% ethanol before and after use.

Mechanical stimulation protocol

For the purpose of demonstrating the functionality of the bioreactor, cardiac tissue constructs were loaded onto the bioreactor and stimulated once for a 4 hour period at a frequency of 1Hz with 10% stretch (figure 5A). The twitch force of each construct was evaluated before and after the stimulation protocol.

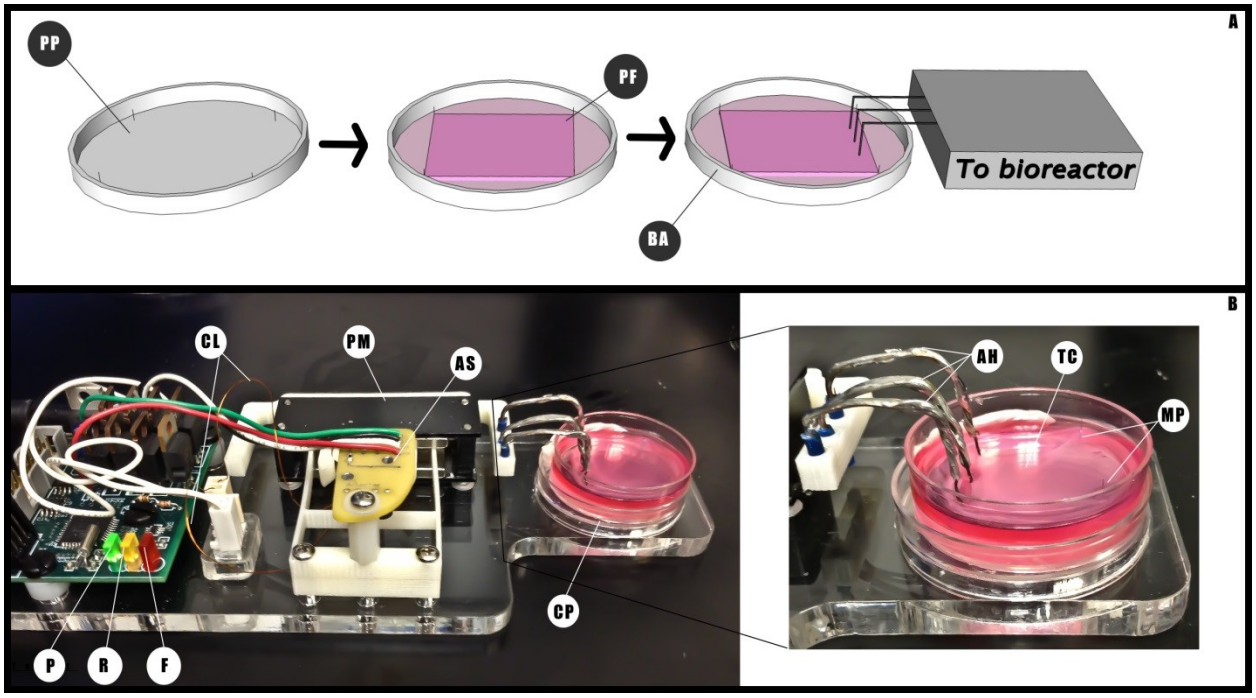


Figure 5. Patch formation process and setup of the bioreactor for the engineered tissues. (A) Schematic showing the overview of the method used to construct the cardiac tissues. PP: plate preparation, PF: patch formation, BA: bioreactor attachment to tissue construct. (B) Setup of the culture plate in the bioreactor and close picture showing the attachment between the device and the tissue construct. CL: coil leads, PM: permanent magnet, AS: armature sensor, P: power LED, F: forward movement LED, R: reverse movement LED, CP: culture plate, AH: attachment Hooks, TC: tissue construct, MP: minutien pins.

2.3 Results

The changes in contractile performance over time were evaluated in culture of tissue with a plating density of 2 million cells/construct (figure 6B). We noted that the constructs were able to maintain contractile performance for a period of 8 days after which, a significant decrease in performance was observed. Our studies were devised to measure the effect of mechanical stretching on engineered tissue constructs once they reached the decreasing phase. The data recorded from 3 separate constructs showed improvement of their contractile properties after the 4 hour stimulation interval (figure 6D). No significant change in construct size after the stretching period was noted (data not shown).

Because all of the data in this paper were collected using a single bioreactor, the culture time among the tissues varied between 2 to 3 days. In order to ensure that the tissue constructs suffered little to no damage, slight modifications had to be made to the attachment hooks. Initially, the hooks were made with 22 gauge solid tin wire; however, this resulted in tearing of the tissues. Once compatibility of the bioreactor with the tissue constructs was confirmed documentable results were obtained.

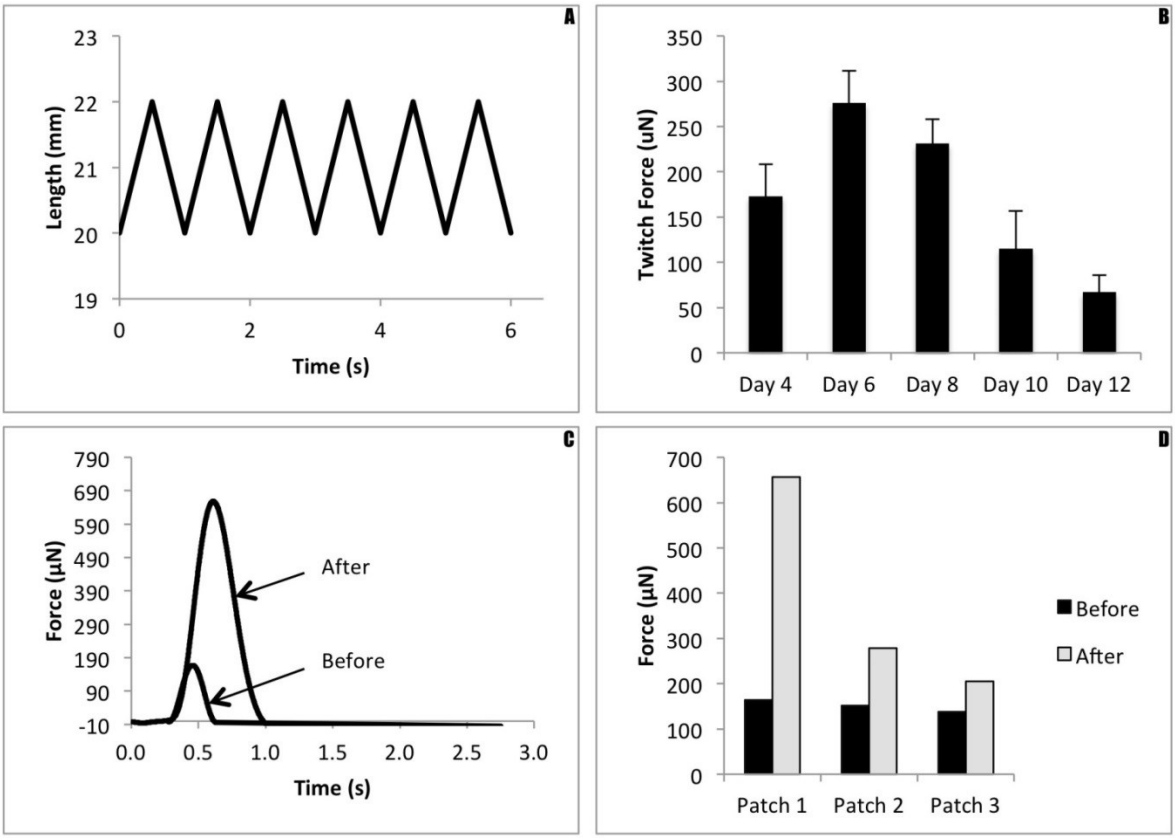


Figure 6. (**A**) Stimulation protocol applied to the constructs for 10% stretch and a 1 Hz frequency. (**B**) Changes in the contractile properties over time of cultured tissues, with a plating density of 2 million cells/construct. (**C**) Force tracing of one of the tissue constructs in response to loading when stretched for 4 hours. (**D**) Changes in the active force of constructs after undergoing mechanical stimulation.

2.4 Discussion

The experiments described in this manuscript were devised to evaluate the compatibility of the bioreactors with our engineered cardiac patch constructs. During the initial testing phase we focused on fabricating practical attachments to affix to the bioreactor and connect to the tissues. The main challenge we encountered was the loading and unloading of the tissues onto the bioreactor without incurring physical damage to the tissues. Once a functional attachment method was formulated, we focused on obtaining data that showed the effect of short term stretching on the cardiac constructs. In the future it may be pertinent to develop a more durable and serviceable attachment as well as develop a bioreactor into which multiple specimens can be loaded.

In addition to applying strain protocols, the elements in place in this bioreactor system are theoretically capable of mechanical assessment of engineered tissue samples. The ability to directly measure twitch force and tensile strength, as well as to generate stress-strain curves could make this device extremely versatile. As engineered tissues on this scale are extremely delicate, minimal sample manipulation and reattachments of instrumentation are desirable.

2.5 Conclusion

A novel programmable bioreactor was developed for mechanically stimulating and assessing self-organized cardiac muscle constructs. Compatibility with engineered tissue samples was experimentally confirmed. Preliminary results suggest that the application of a 1Hz cyclic tensile strain to scaffold free, self-organized cardiac patches elicits at marked increase in twitch force.

CHAPTER 3: PROGRAMMABLE MECHANOBIOREACTOR FOR EXPLORATION OF THE EFFECTS OF PERIODIC STIMULUS ON MSC DIFFERENTIATION[3]

3.1 Introduction/Background

Bone and cartilage defects, either individually or in combination, result from a wide variety of congenital anomalies, traumatic injury, and cancer extirpation. These defects are generally treated with autologous grafts that require a secondary surgical site, and can be associated with significant donor-site morbidity. The capacity of tissue engineering to provide solutions to these challenges is significant. The availability and versatility of mammalian MSCs for potential applications in tissue engineering and regenerative medicine has immense potential.[73–75]

The factors regulating cell fate of MSCs are widely varied and are not yet fully characterized or understood. Mature human cells generally produce a maximum of ~30% of the proteins the genome is capable of.[30] Cell differentiation is therefore commonly accepted in the literature as the selective repression of specific groups of genes, regardless of the mechanism of influence. Research suggests that active mechanical stimulation will play an important role in the differentiation and organization of MSCs into mature, functional tissues.[76–82] For example, tensile strain of substrates seeded with MSCs has been shown to

---

[3]This chapter was submitted to Tissue Engineering. The citation is as follows: A. T. Cashion, M. Caballero, A. Halevi, A. Pappa, R. G. Dennis, and J. A. van Aalst, "Computer controlled mechanobioreactor network: vibration frequency regulates MSC differentiation," 2013.

induce an osteogenic lineage as indicated by significant increases of BMP2 levels.[78] Comparatively, cyclic compressive strain has been associated with increased TGF-β1 synthesis in a study of rabbit MSCs which indicates chondrogenesis.[83]

Mechanotransduction of biological cells is a complex integration of signals including integrin binding, signaling molecules, stretch sensitive ion channels, and cytoskeletal deformation.[84] Choreographing specific mechanical influences to selectively guide cell fate will thus necessarily include a large amount of research into scaffold design, ECM composition, and active mechanical stimulus. The state of research into the mechanobiological influence on MSC fate has been recently reviewed in detail.[80] Pulsed ultrasound has been shown to promote healing of skeletal injuries possibly due to effects on cellular proliferation and differentiation.[85–92] Other mechanical stimuli such as forces from accelerations and vibrations have been shown to elicit effects on osteogenesis[93–99] and angiogenesis[100] in tissues, as well as changes in cellular metabolism[101] and ECM composition.[102–104] Mechanical strain protocols have been applied to 2D cell cultures primarily by stretching, compressing, or bending biocompatible elastic substrates.[105–109] Other methods have included hydrostatic or direct contact pressure and fluid shear.[108]

A review of the literature yields independent studies where specific MSC fates were achieved with cyclic mechanical stimulus. Hydrostatic pressure applied as a 1Hz sinusoid to hMSC cultures has been demonstrated to enhance chondrogenesis.[83,110,111] Other studies have shown that higher stimulation frequencies of MSCs have yielded osteogenesis.[98,112] To the knowledge of the authors, the MSC differentiation response from vibrational frequency and waveform power spectrum has been scarcely explored beyond individually focused experiments like those mentioned here. It is our belief that the research community would

benefit greatly from a series of controlled experiments into the vibrational frequency dependence of MSC differentiation in both in-vitro and in-situ cultures. Based on these findings in the literature, we have hypothesized that the combination of compressive, tensile and shear forces from an applied vibratory stimulus will generate bone formation at higher frequencies and a cartilage phenotype at lower frequencies.

3.2 Methods

Bioreactor Frame Construction

A mechanical drawing of the bioreactor system and a picture of it in use are shown in Figure 1. The frame was constructed primarily of laser cut 0.635cm black acrylic and was solvent welded using dichloromethane. The assembly consists of a stationary stage and a translatable stage spaced 2.5cm apart and mounted to a baseplate (33.0cm x 7.6cm).

The reciprocating stage of the mechanobioreactor consists of an acrylic platform (7.6cm x 12.7cm) suspended by four lengths of alloy 402 stainless steel wire (7.6cm long x .051cm diam). Linear translation is achieved similarly to a dual four bar linkage driven by an electromechanical voice coil actuator (VCA). The movable stage can be translated approximately within a plane for small displacements by applying horizontal forces, which in turn deform the long and thin support wires as the moveable stage is displaced from its neutral resting point. The support wires also provide a gentle restoring force which tends to keep the moveable stage in the center of its plane of motion. The VCA is configured to drive the culture plate in a linear trajectory within the allowable plane of motion of the moveable stage. The edges of the moveable stage are cut to facilitate securement of culture plates with straps or rubber bands. The VCA mechanism is secured to the stationary platform using

nylon thumbscrews. A thin strip of acrylic (not shown in figure) is used to attach the VCA to the movable stage to serve as a flexible drive linkage.

The horizontal motion of the culture plate driven by the VCA causes fluid motion back and forth across the cells in culture at the bottom of each well. In this configuration it is the fluid motion that transduces mechanical shear forces to the cells in culture. In other configurations[113] the VCA can directly apply mechanical strain to engineered tissues, or to the substrate material upon which the tissues are cultured. The degree of flexibility in how this device may be configured and attached to culture vessels, synthetic substrates, or directly to tissues themselves allows it to be employed more widely in tissue engineering research than mechanisms that are specifically designed to drive only fluid flow or mechanical strain of uniform substrate materials.

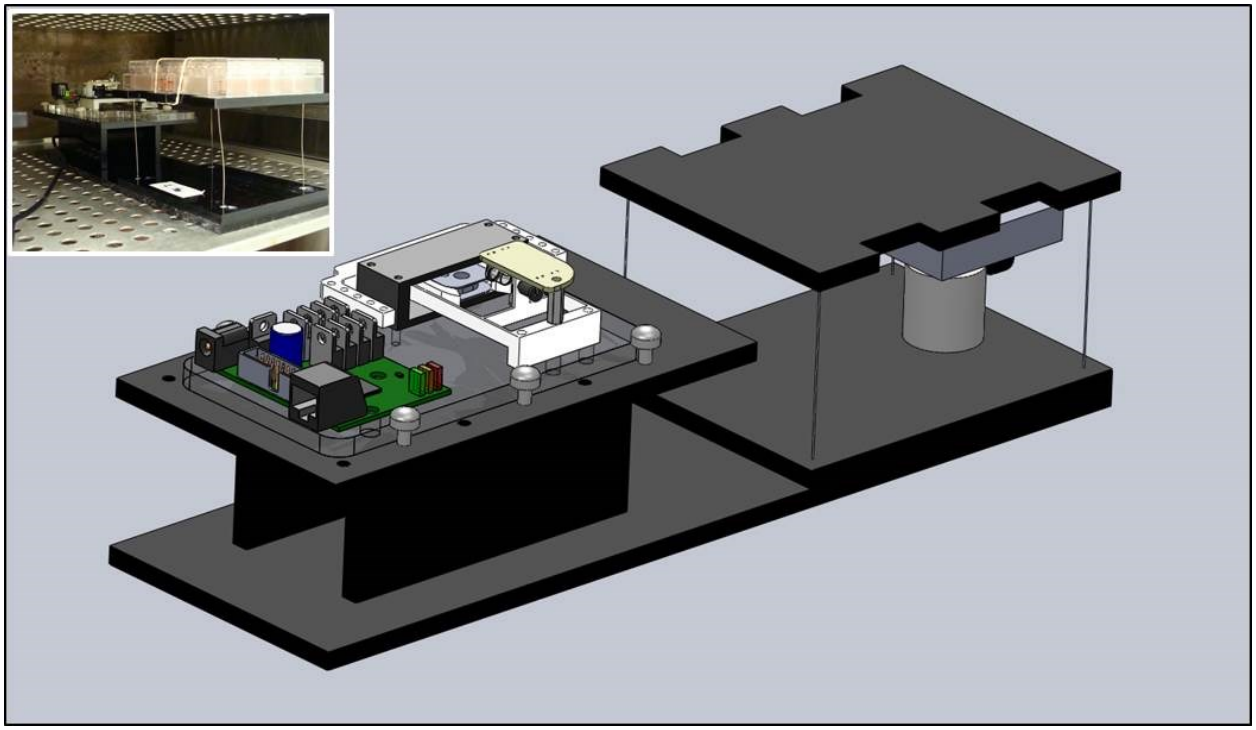


Figure 1. SolidWorks assembly of the programmable voice coil actuator vibration mechanobioreactor (inlay:Bioreactor system in use in the incubator with multi-well culture plate shown in foreground, strapped to the moveable stage)

Electronics and Voice Coil Actuator

The instrumentation of the voice coil linear actuator, the embedded electronics[113] and the computer intermediary board[72] are described elsewhere. Briefly, the researcher selects stimulation protocol parameters on a computer user interface written for this purpose in Visual Basic 2010. The parameters are sent via USB to an intermediary control board where they are distributed to the appropriate bioreactor within the incubator via I2C digital communication.

The voice coil actuator operates by Lorentz forces generated within an electric coil in a direction orthogonal to a static magnetic field.[70] The coil is mounted within a compliant mechanism spring with very low hysteresis and low spring constant such that by varying the direction and magnitude of the current in the coil, the linear position can be controlled over a displacement of ~ +/- 2 mm from the neutral position. The VCA position is tracked with sub-micron precision using an optoelectronic differential displacement sensor fitted with an optical beam interrupter.[71] The embedded microcontroller monitors the position of the actuator and controls the current through the coil in accordance with the programmed stimulus protocol.

Oscillatory Behavior

The dynamic mechanical behavior of this system can be modeled using the equations of motion describing an ordinary harmonic oscillator:

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0$$

where m is the mass, x is the linear position, c is the damping coefficient, k is the spring constant, and t represents time. The solution to this equation yields the following

relationships that allow it to be expressed in terms of the natural frequency, ω_n and the damping ratio, ζ:

$$\omega_n = \sqrt{k/m}$$

$$c_{critical} = 2\sqrt{km} = 2m\omega_n$$

$$\zeta = \frac{c}{c_{critical}}$$

which can be expressed as:

$$\frac{d^2x}{dt^2} + 2\zeta\omega_n\frac{dx}{dt} + {\omega_n}^2x = 0$$

As depicted in the above equations, the natural resonance of the system is a function of both the oscillating mass and the spring constant. The range of under-damped natural resonance frequencies in the experiments presented here was measured with the optical displacement sensor and an oscilloscope to be between 5 and 7 Hz.

Adjustable Damping Mechanism

At stimulation frequencies close to that of the resonance of the system or related harmonics, the amplitude of the resonance oscillations becomes significant. To reduce resonant movement which is superimposed on the desired oscillatory motion, the damping coefficient was adjusted using a mechanism that implements principles identical those of magnetic braking[70] (Figure 2). A 1.91cm square neodymium magnet was mounted to the underside of the translation platform. As the magnet passes over a nonmagnetic electrical conductor, the induced electrical currents in the conductor induce internal eddy currents that then give rise to a magnetic field that opposes the motion of the permanent magnet and

therefore the translation stage. The opposing magnetic field is proportional to the velocity of motion of the permanent magnet with respect to the non-magnetic conductor. This results in a zero-hysteresis and nearly ideal linear damper, and thus is often employed as a damper in high-precision mechanisms. For example, functionally identical damping mechanisms are readily visible at the end of "magnetic damping triple beam balances", usually in the form of a thin aluminum blade passing between two permanent magnets at the distal end of the triple beam assembly.

The nonmagnetic conductor is a machined block of aluminum 5.08cm x 5.08cm x 1.27cm with a press fit aluminum cylinder 2.54cm x 1.59cm diam. The aluminum cylinder slides vertically into an appropriately machined Delrin cylinder (3.81cm x 2.54cm diam) and is held in place with a nylon (10-32) screw. This arrangement allows the damping ratio to be adjusted manually to tune the dynamics of the system: the closer the aluminum block is to the magnet, the higher the damping ratio. The use of very powerful grade N42 rare earth element magnets, now widely available for purchase on the Internet, allows a wide range of damping ratios to be achieved.

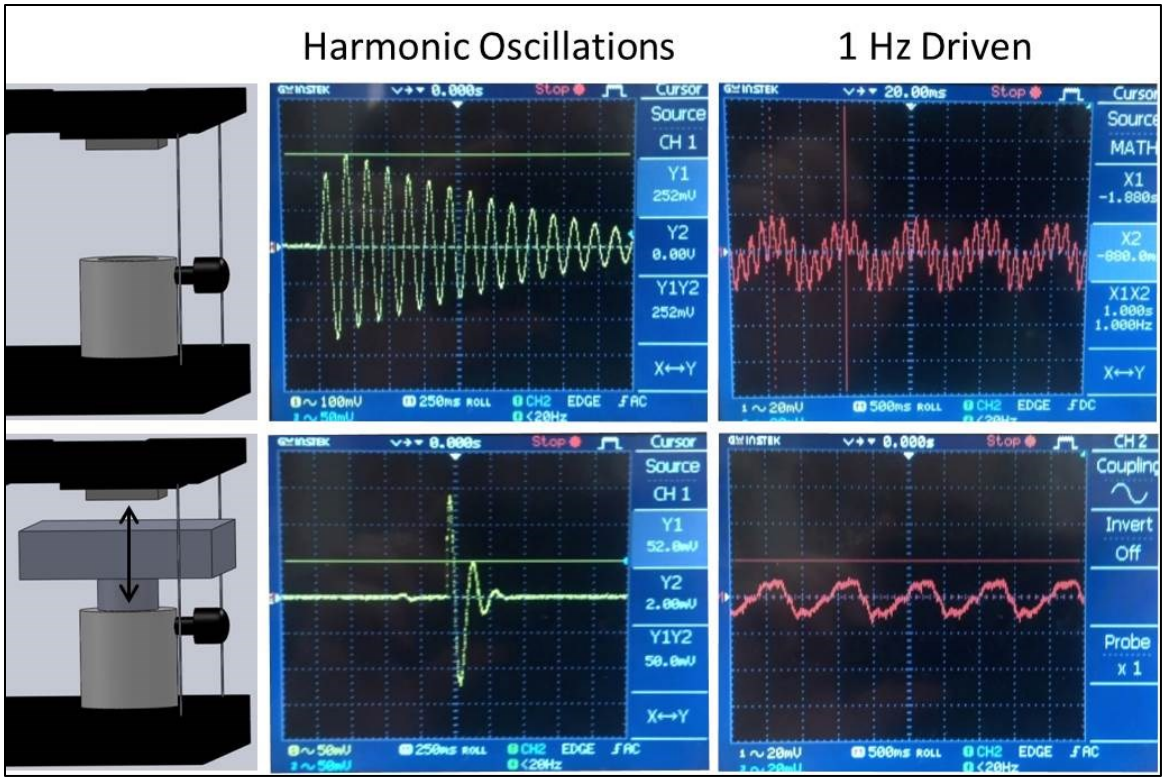


Figure 2. Demonstration of the damping mechanism. All signals are oscilloscope readings of the optoelectronic displacement sensor. **(Top)** Without damper **(Bottom)** With damper

The efficacy of the adjustable damper mechanism can be quantified in each position by calculating the damping ratio of the harmonic system. Since the system is underdamped, the damping ratio can be determined from the amplitudes of two successive peaks, A_0 and A_1using the logarithmic decrement method[114]:

$$\zeta = \frac{1}{\sqrt{1 + \left(\frac{2\pi}{\ln(A_0/A_1)}\right)^2}}$$

Because the system is designed to oscillate, the target damping ratio is still less than the critically damped ratio of one. Using the adjustable damping mechanism depicted in (Figure 2) the damping ratio can be adjusted from $\zeta = 0.01$ to $\zeta = 0.15$.

Waveform Generation

A graphical user interface was created in VisualBasic 2010 for selection of wave form parameters. A sinusoidal vector is created on the embedded PIC18F4550 microcontroller using the following equation:

$$y = A * 10 \sin(2\pi f t)$$

The linear displacement of the platform from the zero position is represented by (y) and is measured by optical displacement sensor increments (~1 micron). The Boolean selection of the sine waveform, the amplitude (A) in increments of 10 microns and the frequency in hertz (f) are sent to the bioreactor microcontroller via the electronics described above. Before beginning the protocol, the sinusoid is calculated on the microcontroller for one complete period using a time increment of: $\frac{Period}{1000}$ in order to yield exactly 1000 steps per sine wave. The sine wave is repeated successively for the duration of the stimulus. Frequency is then set by establishing the update period for the sinusoidal function such that the waveform amplitude is updated at 1000 times the desired frequency of the output sinusoid.

Stimulation Protocol

To avoid over-stressing the cells and thereby disrupting the cell cycle, the stimulation protocol was set to allow 15 minutes of rest between each 1 minute stimulus duration. Two bioreactors were set up in parallel, one with 1Hz stimulus and the other with 100Hz. Waveform amplitudes were adjusted to apply similar levels of energy at each frequency.

Umbilical Cord Harvest

Human umbilical cord (hUC)-derived mesenchymal stem cells (MSCs) were obtained from human UCs following appropriate consent and cells were isolated from tissue explants, as previously described.[115] Briefly, UCs were cut from the placenta and immersed in sterile

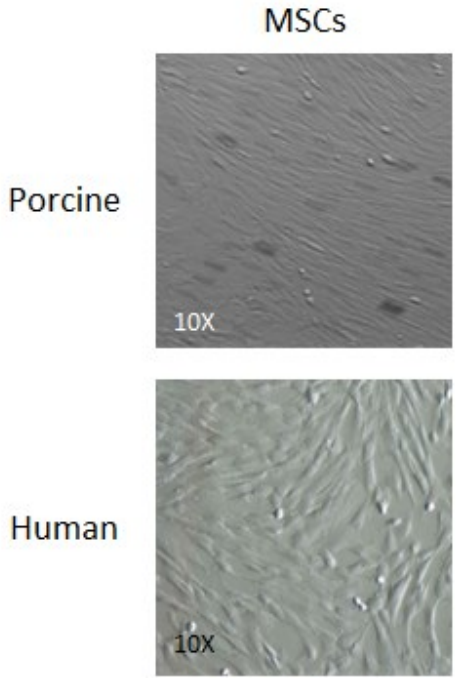
transport solution (PBS supplemented 300 U/mL penicillin, 300 µg/mL streptomycin) Approximately 6 cm sections were cut and washed to remove residual blood. The UC epithelium and vessels were removed and discarded, and the Wharton's Jelly was cut into $1mm^2$-sized pieces and placed in Dulbecco Modified Eagle Medium (DMEM/F12) supplemented with 10% MSC-FBS, 50 µg/mL gentamicin, 100 U/mL penicillin, 100 µg/mL streptomycin, 55 µM β-Mercaptoethanol, and 1 mM Sodium Pyruvate. Growth medium was replaced every 3 to 4 days.

Porcine umbilical cord (pUC)-derived MSCs were obtained from pigs at the NC State Swine Education Unit. Cords were processed as described for human cords with small variation. To prevent contamination, cords were dipped briefly in betadine followed by 70% ethanol before dissection as described for human cells. Isolated Wharton's Jelly was cut into $1mm^2$-sized pieces and placed in DMEM Supplemented with 15% FBS, 50 µg/mL gentamicin, 100 U/mL penicillin, 100 µg/mL streptomycin, and 0.25 µg/mL amphotericin B). Growth media was replaced every 2 or 3 days.

Flow Cytometry

Cultured cells were trypsinized and resuspended at a concentration of 10^6 cells/mL in blocking buffer (0.5% BSA, 0.01% Na-azide, 1X PBS). A total of 10^5 cells were treated with 10µL of diluted primary antibody ($CD90^+$, $CD73^+$, $CD105^+$, $CD34^+$, or $SSEA\text{-}4^+$) then incubated for 20 minutes on ice. After 3 washes with cold PBS, cells were incubated with 1.5 µg of secondary antibody in 100 µL of blocking buffer for 20 minutes on ice in the dark. Samples were then washed with PBS 3 times and fixed in 1% paraformaldehyde and stored

at 4°C in the dark until analysis. Cells were analyzed using a Beckman-Coulter (Dako) CyAn ADP and Summit 4.3 software. The results of this process can be seen in (Figure 3).



Surface Markers

Marker	Mean %
CD105	89
CD90	99
CD73	71
SSEA-4	<1
CD34	<1

Marker	Mean %
CD105	98
CD90	99
CD73	97
SSEA-4	65
CD34	<1

Figure 3. Characterization of human and porcine MSCs at confluence (left). Results of flow cytometry (right)

Osteogenesis

Either hUC or pUCMSCs were seeded at 2 x10^4 cell/cm^2 and incubated in-MEM supplemented with 10% FBS, 2 mM glutamine, 100 U/mL/100 µg/mL Penicillin/Streptomycin, 10 mM b-glycerophosphate, 0.1 µm dexamethasone, and 50 µM ascorbic acid. Medium was replaced every 2 days. Osteoblastogenic differentiation was assayed at 21 days by Alizarin Red staining.

Chondrogenesis

UCMSCs (human or porcine) were seeded at a density of 4x10^5cells/cm^2an incubated in (DMEM)–high glucose supplemented with ITS-1, 0.1 mM ascorbic acid, 10^{-7}M dexamethasone, 10 ng/mL transforming growth factor(TGF)–β1 , 10ng/mL TGF-β3 100

ng/mL insulin-like growth factor (IGF)D and 40 μg/mL L-proline. Chondrogenic differentiation was assayed at 21 days by alcian blue staining.[116]

Vibration

UC-MSC were seeded in regular growth media, at either 2 $x10^4$ cell/cm^2 (for 100 Hz) or $4x10^5$cells/cm^2 (for 1 Hz). Cells were allowed to attach for 24 hours and then transfer to vibratory bioreactor program for either 100 Hz or 1 Hz vibration. Cultures were subjected to vibration cycles, as described above, for 15 hours/day for a period of 10 days. Regular growth media was changed every 3 days.

Differentiation Analysis

At 21 days post induction, cells were fixed in 10% buffered formalin for 30 minutes and rinsed with distilled water. To detect calcium deposits in osteogenic induction, cells were stained with 2% alizarin red S solution, pH 4.2, for 10 minutes; in chondro-induced cells the presence of glycosaminoglycans and mucopolysaccharides was demonstrated by staining with 0.1 mg/ml Alcian Blue 8 GX for 20 minutes. Excess dye was removed by careful washes with distilled water. Staining cultures were analyzed under light microscopy, using a Nikon Eclipse Ti-S inverted microscope.

3.3 Results

The bioreactor developed for these experiments successfully generated the desired waveform parameters. The 100Hz and 1Hz stimuli were experimentally measured to be exactly 100.0Hz and 1.0 Hz respectively with zero frequency drift after 2 days of operation.

In this preliminary study, the frequency of stimulus had a pronounced effect on UC-MSC differentiation in both human (Figure 4) and porcine (Figure 5) models when compared

to controls. Relative to the respective controls, the results of the human and porcine studies were similar. The positive control wells were chemically induced to promote osteogenesis or chondrogenesis. Alizarin red staining of the osteogenesis positive control showed a high calcium content as expected. Correspondingly, Alcian blue staining of the chondrogenesis positive control showed high levels of GAGs as expected. The negative control wells tended toward osteogenesis in this study with very little cartilage present after 10 days.

The samples driven at 100Hz showed denser calcium deposits than negative controls in both human and porcine studies as well as very low levels of GAGs. The samples driven at 1Hz demonstrated substantially higher GAG content than the negative controls and lower calcium content than the 100Hz samples.

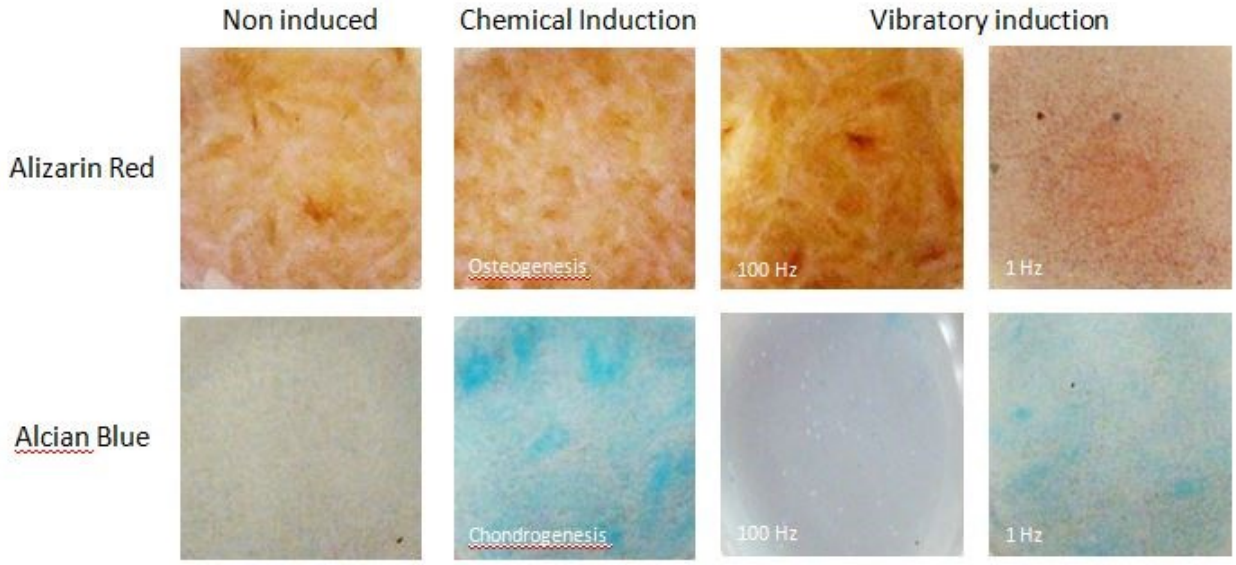


Figure 4. hUCMSC preliminary results **(Top)** Stained for calcium using Alazarin red **(Bottom)** Stained for GAGs using Alcian blue

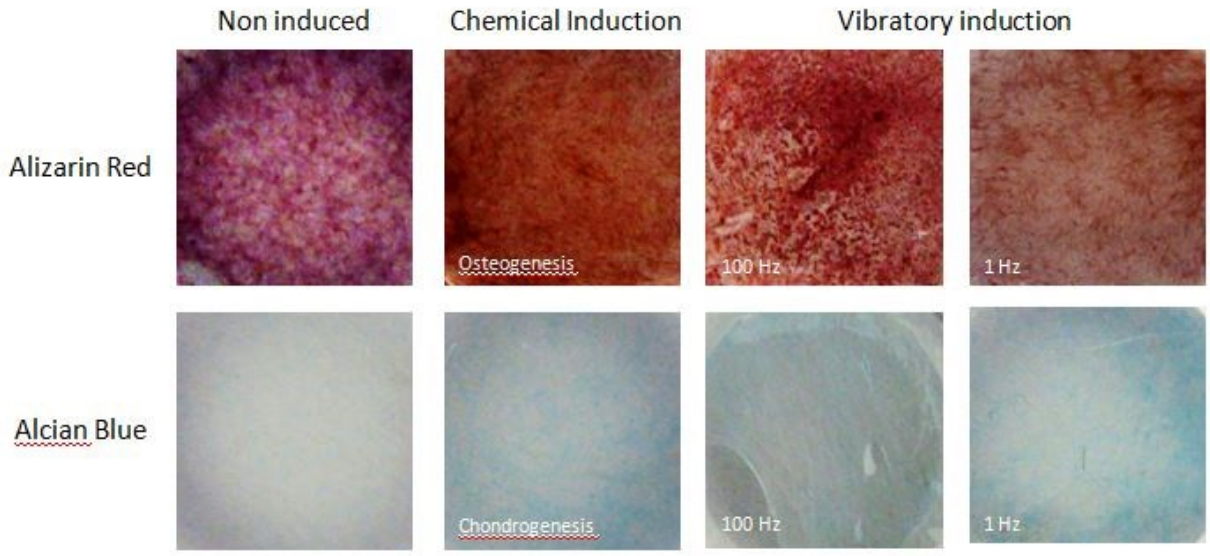


Figure 5. pUCMSC preliminary results **(Top)** Stained for calcium using Alazarin red **(Bottom)** Stained for GAGs using Alcian blue

The results of the mRNA quantification experiment are presented in figure 6. In both hUCMSCs and pUCMSCs, the 100Hz stimulus showed a significant increase in BMP2 levels when compared with the 1Hz samples and the controls. Additionally, both hUCMSCs and pUCMSCs had significantly higher levels of COL2/COL1 after the 1Hz stimulus than the 100Hz samples and the controls.

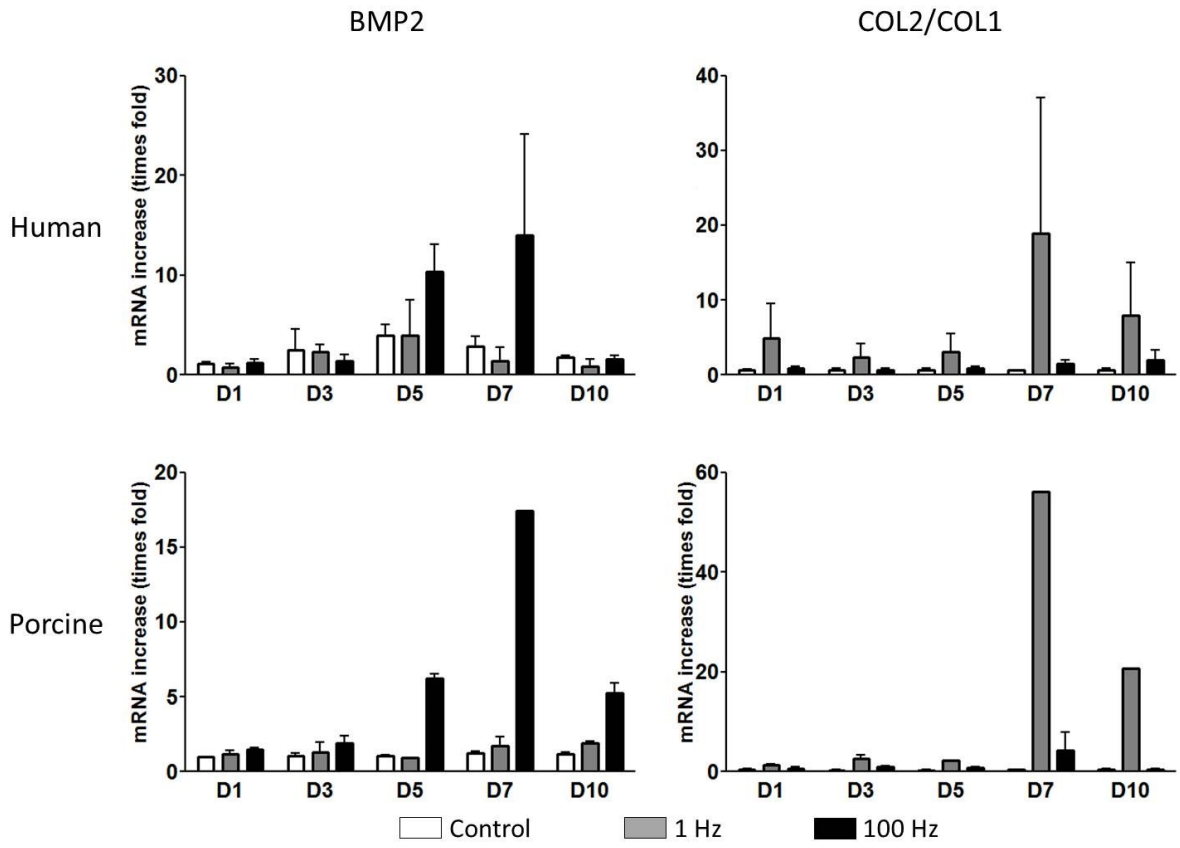


Figure 6. mRNA preliminary results **(Top)** hUCMSC **(Bottom)** pUCMSC

3.4 Discussion

The data generated in this study demonstrate that vibratory signals can be used to stimulate both human and porcine UC MSCs toward specific phenotypes. If these vibratory signals can eventually be delivered in vivo, and at specific sites, the clinical value would be significant. At lower frequencies, the phenotype appears to be cartilage and at higher frequencies, the phenotype appears to be bone. This preliminary data qualitatively agrees with the differentiation results of individualized studies in the literature in that 1Hz cyclic stimulus enhances chondrogenesis[83,110] and higher frequency stimuli enhance osteogenesis[98,112]. It should be noted however that the aforementioned studies each focused on relative amounts of their respective bone or cartilage differentiation markers rather than

tracking comprehensive gene expression profiles. Additionally, the studies that indicated chondrogenesis differentiation of MSCs applied only compressive loads to the samples. The assessment from the alcian blue and alizarin red staining was reiterated quantitatively by measuring BMP2 and COL2/COL1 mRNA in the samples following the stimulus. This result supplies compelling evidence that MSC differentiation is sensitive to vibrational frequency.

This manuscript has established bioreactor methods that will enable further investigation into the differentiation response of MSCs to vibrational frequency and to determine the thresholds for cartilage and bone formation. In the near future, we plan to develop an implantable bioreactor that will be tested in a subcutaneous porcine model. The stimulation paradigm of the implantable bioreactor will be informed by the results of in-vitro studies conducted with the system presented here.

3.5 Conclusion

A versatile bioreactor was constructed to enable research of the effects of vibrational stresses on MSCs in-vitro. As a demonstrative example, human and porcine umbilical cord-derived MSCs were stimulated with the present bioreactor system at frequencies of 1Hz and 100Hz. The lower frequency (1Hz) resulted in a cartilage-like phenotype for both human and porcine MSCs, whereas the higher frequency (100Hz) resulted in bone-like phenotype, as indicated by expression of BMP2 and COL2/COL1 and by Alizarin Red and Alcian Blue staining.

APPENDIX 1: FUNCTIONAL ULTRASOUND IMAGING FOR ASSESSMENT OF EXTRACELLULAR MATRIX SCAFFOLDS USED FOR LIVER ORGANOID FORMATION[4] (Summary)

A1.1 Introduction

Liver transplantation is the gold standard for end-stage liver disease. Currently, more than 16,000 adults and children are in need of liver transplants, but unfortunately the amount of livers available for transplantation is in short supply.[117] Therefore, alternatives, such as bioartificial organs, are being explored. Clinically relevant bioartificial livers typically use hepatocytes from xenographs, seeded in a semi-permeable membrane and based on hemodialysis of the patient's blood.[118] Unfortunately, these bioartificial livers only provide temporary relief until a transplantable organ is available. In order to provide a true replacement for live donor transplantations, a more sophisticated artificial organ is needed. The desire to produce an artificial liver to replace the dependency on living donors has led to a push to develop decellularized liver organs that, after recellularization, would house patient-specific human hepatic cells. The method of decellularization converts an *ex vivo* or cadaveric liver organ to an acellular, three-dimensional structure of naturally occurring scaffold materials. These scaffolds would then be reseeded with a patient's own cells and, in

---

[4]This appendix discusses work published in Biomaterials. The citation is as follows:

R. C. Gessner, A. D. Hanson, S. Feingold, A. T. Cashion, A. Corcimaru, B. T. Wu, C. R.. Mullins, S. R. Aylward, L. M. Reid, and P. a Dayton, "Functional ultrasound imaging for assessment of extracellular matrix scaffolds used for liver organoid formation.," *Biomaterials*, pp. 1–11, Sep. 2013.

theory, would act as a functional liver with minimal transplant rejection. The new organoid would provide the same, albeit improved, metabolic operations as their original liver.

In addition to transplantation research, generating decellularized liver biomatrix scaffolds from non-human species provides a useful model for research in the broader field of regenerative medicine. Scaffolds prepared using perfusion tissue decellularization of rat liver have been shown to have differentiation factors that are tissue-specific but not species-specific, meaning that human hepatic stem cells will differentiate into human liver cells when seeded onto rat liver biomatrix.[5,119] This result is significant in that it may enable the creation of functional human liver organoids for use in toxicology research and experimental liver treatment testing.

Protocols to decellularize organs vary among labs, and thus the resulting scaffolds contain varying amounts of matrix molecules. The matrix molecules that are retained after decellularization are important for cell engraftment, mitosis and differentiation.[120,121] Some labs have been able to achieve cellular engraftment in the decellularized scaffolds, though the reported functional data - such as albumin and urea secretion - from reseeded cells has been lower than normal human values. For reseeded scaffolds to provide either an adequate organ transplantation alternative, or *in vivo* model for drug discovery, cell function must meet that of normal human livers. It is thus one of our group's long term objectives to optimize the scaffold reseeding protocol, thereby maximizing the structural integrity and functionality of the organoid. Prior to this being possible, we first needed to develop a method to non-destructively assess perfusion through the scaffold matrix.

The purpose of the work presented here is to develop a protocol to enable rapid assessment of perfusion within a rodent *ex vivo* scaffold. Without adequate perfusion, the

reseeding process of populating a liver biomatrix scaffold with a new host's cells will necessarily fail, as this process relies on fluid transport for the delivery of these cells throughout the 3D volume of tissue. Perfusion in rodent matrix scaffolds is assessed in different ways, but generally this is achieved by qualitatively examining the color of the tissue sample, with regions of discoloration in the sample indicating a lack of adequate internal flow. This approach can be useful for identifying insufficient flow within large regions in the sample, such as throughout an entire lobe, but insensitive to detecting subtle variations in volumetric perfusion over time. Corrosion casting, colored die injection[122], and fluoroscopy[119] have all been used for 3D assessment of vasculature patency in rat liver biomatrix, but these techniques either destroy the sample or provide little structural information about the vascular network as a whole. Thus a non-destructive method to image both the anatomy and flow within the sample would be highly desirable.

A1.2 Figures

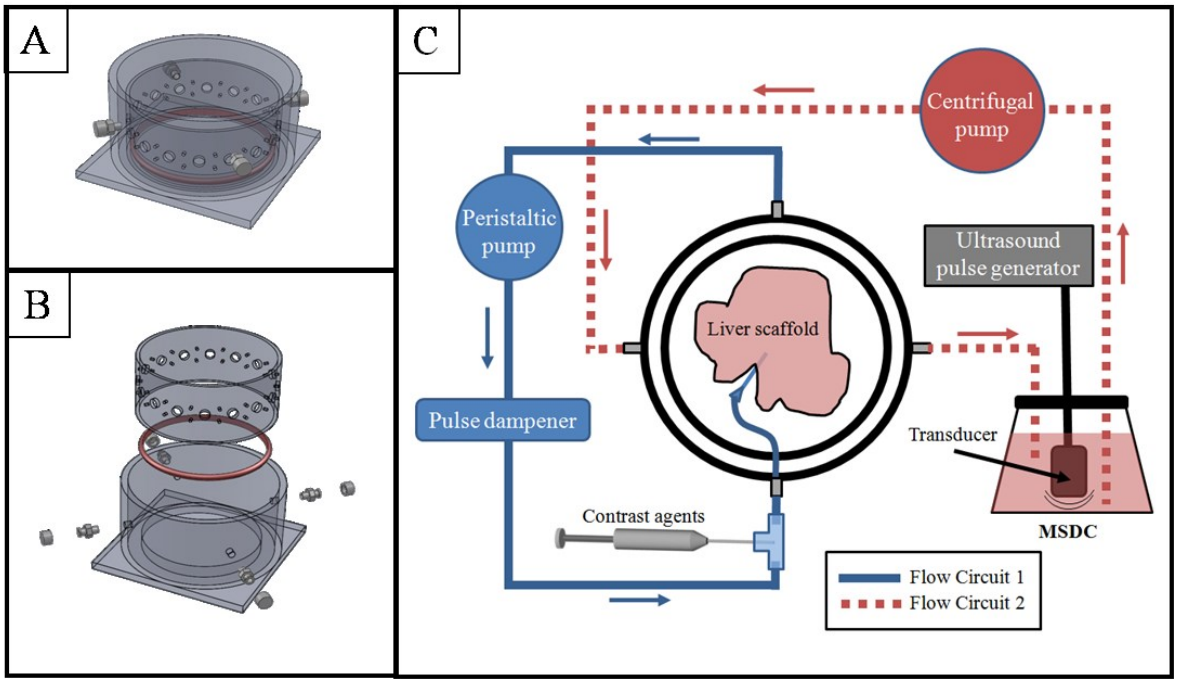


Figure 1. Schematics for the sample imaging chamber. A) Assembled sample imaging chamber and B) exploded view of the sample imaging chamber. C) A top-down cartoon schematic illustrating the two flow circuits in the setup. Flow Circuit 1 provided perfusion and microbubbles to the liver scaffold, while Flow Circuit 2 provided continuous circulation through the microbubble sequestration and destruction chamber (MSDC) to remove contrast excreted from the sample.

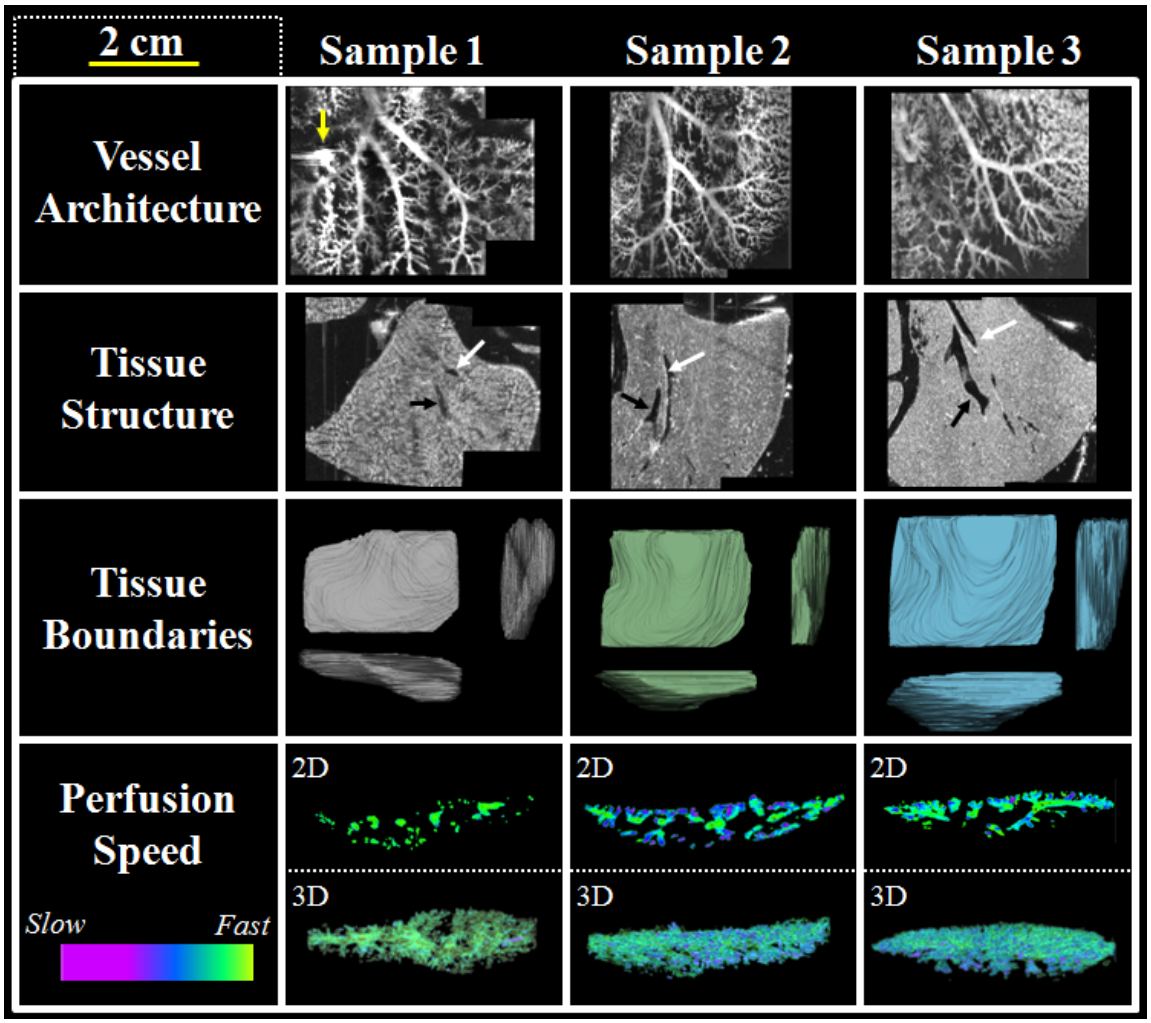


Figure 2. A compilation of all image data acquired of the three biomatrix scaffold samples. Yellow arrow indicates location of sample #1 which were perforated, and thus leaking microbubbles. Each row was acquired with the following imaging modes (from top to bottom): Acoustic angiography, b-mode, manually defined regions of interest, and flash replenishment. White and black arrows on the b-mode images indicate vessels identified as part of either the portal or hepatic circuits, respectively. In the flash replenishment images, the top images are 2D XY slices, and the bottom are Z-projection renderings. The flow rate colorbar indicates pixels mapped to between 10 and 1 second reperfusion times for slow to fast flow, respectively.

APPENDIX 2: Source Code for CHAPTER 1, PICI, PICII, and the GUI

Source code for PICI:

```
#include <18F4550.h>
#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, USBDIV, PLL5, CPUDIV3, VREGEN, PUT
#use delay (clock=24000000)
#use i2c(Slave,sda=PIN_B0,scl=PIN_B1,address=0xB2,force_hw,no_stretch)


//  PCB board pin defines
#define GREEN_LED PIN_C0
#define YELLOW_LED PIN_C1
#define RED_LED PIN_C2



/*
//  setup USB endpoints  //may not always be necessary
#define USB_EP1_TX_ENABLE USB_ENABLE_INTERRUPT
#define USB_EP1_TX_SIZE 32
#define USB_EP1_RX_ENABLE USB_ENABLE_INTERRUPT
#define USB_EP1_RX_SIZE 8
*/

//  included headers and source code
#include <pic18_usb.h>  //Select PIC for USB
#include <usb_desc_hid.h>  //USB Application (in this case HID_IO)
#include <usb.c>  //USB commands.  Look through this to understand
specifics of commands.


//Special Function Registers/////////
#byte TRISB = 0xF93
#bit TrisB0 = TRISB.0
#bit TrisB1 = TRISB.1
#byte SSPBUF = 0xFC9
#byte SSPSTAT = 0xFC7
#bit BF = SSPSTAT.0
#byte SSPCON1 = 0xFC6
#bit CKP  = SSPCON1.4
#bit SSPOV = SSPCON1.6
#bit WCOL = SSPCON1.7

//////////////////////////

//Define Globals//
int16 Charge_Time = 20000; //in ms
int16 Short_Time = 2000; //in ms
int16 Read_Time = 30000; //in ms
int16 Voltage = 65534;
int16 Cycles = 50;
```

```
int16 Status;
int8 in[3];
int16 out[2];
int8 buffer_to_16[2];
int1 packet_count = 0;
int which_Var = 0;

typedef enum {NOTHING, CONTROL_READ,
              ADDRESS_READ, READ_COMMAND_READ} I2C_STATE;

I2C_STATE fState;
BYTE adress, buffer[0x10];

int1 New_Data_Flag = 0;
int16 Data;

//The following subroutine Takes a low and high byte and assembles them
into a
//single 16-bit number.  It is used by first loading your 8-bit numbers
into an
//array called "buffer" and then call the command:
//Assemble_16(element number of Low_Byte, element number of High_Byte);
//The assembled 16-bit number is returned
int16 Assemble_16(int L_Byte_Add, int H_Byte_Add)
{
  int L_Byte_8;
  int H_Byte_8;
  int16 Assembled;
  int16 Shift;
  int16 L_Byte_16;
  int16 H_Byte_16;
  int16 H_Shifted;

  L_Byte_8 = buffer_to_16[L_Byte_Add];
  H_Byte_8 = buffer_to_16[H_Byte_Add];
  Shift = 256;  //2^8 = 256  (Multiplying by 2^8 shifts 8 bits left)



  //Turning the 2 8-bit numbers into 16-bit numbers for PIC math
operations
  L_Byte_16 = L_Byte_8;
  H_Byte_16 = H_Byte_8;


  H_Shifted = H_Byte_16*Shift;  //(Multiplying by 2^8 shifts 8 bits left)
  Assembled = H_Shifted + L_Byte_16;  //Adding the low byte to complete
operation

  return Assembled;


}


//The following subroutine splits a 16-bit number into its high and low
bytes
```

```
//It is used by calling "Split_16(16-bit number to split, 0 or 1)"
//0 will return Low_Byte and 1 will return High_Byte
int Split_16(int16 To_Split, int1 H_or_L){

int16 Shift;
int16 AND16bit;
int16 H_Byte_16;
int16 L_Byte_16;
int H_Byte;
int L_Byte;

Shift = 256;  //2^8 = 256  (Dividing by 2^8 shifts 8 bits right)
AND16bit = 255;  //0b0000000011111111

H_Byte_16 = To_Split / Shift;  //(Dividing by 2^8 shifts 8 bits right)
L_Byte_16 = To_Split & AND16bit; //ANDing with 255 gives you the low byte

//Turning the numbers into 8-bit pieces
H_Byte = H_Byte_16;
L_Byte = L_Byte_16;

if (H_or_L ==1){return H_Byte;}
if (H_or_L ==0){return L_Byte;}

}

//The following subroutine sends a 16-bit number via USB.  It actually is
sending
//2 individual bytes that you'll have to bring together in software.
void USB_Send(int16 Send)
{
   out[0]= Send;
   out[1] = 1007;
   usb_put_packet(1,out,2,USB_DTS_TOGGLE);
}


//This subroutine reads USB data if it is available.
void USB_Read()
{

if (usb_kbhit(1)) {        //usb_kbhit is a status bit for available data
in input buffer.
usb_get_packet(1,in,4);    //Reads the input buffer into array "in" and
clears the input buffer.

//2 received 16-bit packets (two updates of the "in" array represents one
transmission
//for updating a data setting
if(packet_count == 1) {
packet_count = 0;

if(which_Var == 2){ //New Voltage available
buffer_to_16[0] = in[0];
buffer_to_16[1] = in[1];
Voltage = Assemble_16(0,1);
```

```
}

if(which_Var == 3){ //New Charge Time available
buffer_to_16[0] = in[0];
buffer_to_16[1] = in[1];
Charge_Time = Assemble_16(0,1);

}

if(which_Var == 4){ //New Read Time available
buffer_to_16[0] = in[0];
buffer_to_16[1] = in[1];
Read_Time = Assemble_16(0,1);

}

if(which_Var == 5){ //New Short Time available
buffer_to_16[0] = in[0];
buffer_to_16[1] = in[1];
Short_Time = Assemble_16(0,1);

}

if(which_Var == 6){ //New # of Cycles available (Also, Cycles is the last
update so I2C should be activated
buffer_to_16[0] = in[0];
buffer_to_16[1] = in[1];
Cycles = Assemble_16(0,1);

//Set the new values to be ready to send
buffer[2] = Split_16(Charge_Time,0);
buffer[3] = Split_16(Charge_Time,1);
buffer[4] = Split_16(Read_Time,0);
buffer[5] = Split_16(Read_Time,1);
buffer[6] = Split_16(Short_Time,0);
buffer[7] = Split_16(Short_Time,1);
buffer[8] = Split_16(Voltage,0);
buffer[9] = Split_16(Voltage,1);
buffer[10] = Split_16(Cycles,0);
buffer[11] = Split_16(Cycles,1);

//interrupt master and tell it to read in new values
output_high(PIN_B2);
delay_ms(1);
output_low(PIN_B2);
}
which_Var = 0;
}

if (packet_count == 0){
if (in[0]>1) {
packet_count = 1;  //Set packet_counter flag to indicate that a setting is
being received.
which_Var = in[0];
//output_toggle(YELLOW_LED);

}
```

```
if (in[0]==1) {output_high(RED_LED);} //Turns red LED on
if (in[0]==0) {output_low(RED_LED);}  //Turns red LED off

}


}
}

#INT_SSP
void SSP_isr()
{

   BYTE incoming, state, clear;
   state = i2c_isr_state();

   if (state <= 0x80){
   incoming = i2c_read();

   if (state ==1){
   adress = incoming;
   //output_high(RED_LED);
   }

   if (state == 2){
   buffer[adress] = incoming;
   if (adress ==1){  //Second byte of 16-bit data point has been received
   New_Data_Flag = 1;
   buffer_to_16[0] = buffer[0];
   buffer_to_16[1] = buffer[1];
   Data = Assemble_16(0,1);
   USB_Send(Data);
   }
   if (adress ==2){ //Status byte has been updated
   buffer_to_16[0] = buffer[2];
   buffer_to_16[1] = 0;
   Status = Assemble_16(0,1);
   USB_Send(Status);

   output_toggle(YELLOW_LED);
   }
   }

   }

   if (state == 0x80){
   //TrisB0 = 1;
   //clear = i2c_read();
   i2c_write(buffer[adress]);


   //SSPBUF = 0;
   //i2c_write(0b00000111);
   //BF = 0;
  // CKP = 0;
  // SSPBUF = buffer[adress];
  // CKP = 1;
```

```
   //output_high(RED_LED);
   }

}



void main (void) {

   //  initialize variables
    int i;
    int Test_H;
    int Test_L;

   fState = NOTHING;
   adress = 0x00;
   for (i=0;i<0x10;i++){
      buffer[i] = 0x00;}

   enable_interrupts(INT_SSP);
   enable_interrupts(GLOBAL);

   //  initialize USB
   usb_init();
   //initialize ADC
  // adc_init();

   //output_high(Charge_Relay);



while (TRUE) {

if (usb_enumerated())    //USB must be enumerated (PC ack) for any USB
communications
{
output_high(GREEN_LED);
//output_low(YELLOW_LED);
USB_Read();

if(Voltage == 500 && Charge_Time == 1000 && Read_Time == 2000 &&
Short_Time == 4000 && Cycles == 8000){
output_toggle(YELLOW_LED);
delay_ms(250);
}

 if(New_Data_Flag ==1){
  New_Data_Flag = 0;
   //USB_Send(Data);
  // output_toggle(PIN_E0);
 }

}
else  //USB not enumerated
{
output_low(GREEN_LED);
//output_high(YELLOW_LED);  //yellow LED is on until USB is enumerated
```

```
}



}
}
```

Source code for PICII:

```
#include <18F4550.h>
#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, USBDIV, PLL5, CPUDIV3, VREGEN, PUT
#use delay (clock=24000000)

#use i2c(Master,sda=PIN_B0,scl=PIN_B1,fast)

//  included headers and source code
#include <pic18_usb.h>  //Select PIC for USB
#include <usb_desc_hid.h>  //USB Application (in this case HID_IO)
#include <usb.c>  //USB commands.  Look through this to understand
specifics of commands.



//Initialize Slave Addresses

//Initialize Variables
int8 Send_Byte = 8;
int8 Slave_One_Address_Write = 0xa0;
int8 Slave_Two_Address_Write = 0xb0;
int8 data;
int8 i2c_buffer[12];

int16 Charge_Time = 10000; //in ms
int16 Short_Time = 500; //in ms
int16 Read_Time = 10000; //in ms
int16 Voltage = 65000;
int16 Cycles = 10;
int16 USB_VARS[5]; //[charge_time, read_time, short_time, voltage, cycles]
int16 Cyc_Num = 0;
int16 analog_in[5];
int H_Byte;
int L_Byte;
int1 Go = 0;
////////////////////////////

//  PCB board pin defines
#define GREEN_LED PIN_C0
#define YELLOW_LED PIN_C1
#define RED_LED PIN_C2
#define ADC_CS PIN_A0
#define ADC_CLK PIN_A1
#define ADC_IN PIN_A4
#define Charge_Cap PIN_D3
#define Short_Cap PIN_D4
```

```
#define Cap_Relay PIN_D5
#define DAC_SYNC PIN_D2
#define DAC_CLK PIN_D1
#define DAC_Data PIN_D0



//  included headers and source code
#include <pic18_usb.h>  //Select PIC for USB
#include <usb_desc_hid.h>  //USB Application (in this case HID_IO)
#include <usb.c>  //USB commands.  Look through this to understand
specifics of commands.

//Special Function Registers
#byte T1CON = 0xFCD
#byte T3CON = 0xFB1


//Defining Globals
int8 in[2];
int16 out[2];
int16 ADC_Val;

 //  initialize variables

   int16 analog_center = 0;
   int16 data_delay = 0;
   int16 charge_delay = 0;
   int16 read_delay = 0;
   int16 short_delay = 0;
   int16 timer = 0;
   int1 charging = 0;
   int1 reading = 0;
   int1 shorting = 0;
   int16 val;
   int16 val1 = 0;

 ////////////////////////////



void Write_Slave(int Slave_Address, int Buffer_Address, Write_Byte)
{
   i2c_start();
   i2c_write(Slave_Address);     // Device address
   i2c_write(Buffer_Address);  // address of the buffer that you are
writing to 'buffer[address]'
   i2c_write(Write_Byte);  //data sent to the buffer(address) above
   i2c_stop();
   delay_us(5);
}
int Read_Slave(int Slave_Address, int Buffer_Address)
{
   i2c_start();
   i2c_write(Slave_Address);
   i2c_write(Buffer_Address);
   //i2c_stop();
```

```
   delay_us(1);
   i2c_start();
   i2c_write(Slave_Address+1);
   delay_us(2);
   data = i2c_read(0);

   i2c_stop();

   return data;


}

void adc_init()
{
int i;
      output_high(ADC_CS);    //  Chip Select (active LOW)
      output_high(ADC_CLK);    //  Serial Clock (active on FALLING Edge)
         delay_us(1);

      //  Select ADC Chip
      output_low(ADC_CS);    //  Chip Select (active LOW)
      output_high(ADC_CLK);    //  Serial Clock (active on FALLING Edge)
         delay_us(1);

      //  Cycle Serial Clock 4 times with CS = LOW to reset ADC:
      for(i = 1; i<=4; ++i)
      {
      output_low(ADC_CLK);    //  Serial Clock (active on FALLING Edge)
         delay_us(1);
      output_high(ADC_CLK);    //  Serial Clock (active on FALLING Edge)
         delay_us(1);
      }



}

void Read_16()
{
int16 volts = 0b0000000000000000;
int bits, c;

 volts = 0b0000000000000000;
         //  Start with ADC INACTIVE
      output_high(ADC_CS);    //  Chip Select (active LOW)
      output_high(ADC_CLK);    //  Serial Clock (active on FALLING Edge)
      //delay_us(1);
     // delay_cycles(2);

   //  Run 24 cycles of SClk to complete the next conversion
      for (c = 1; c <=25; ++c)
      {
      output_high(ADC_CLK);    //  Serial Clock (active on FALLING Edge)
   //      delay_cycles(2);
      output_low(ADC_CLK);    //  Serial Clock (active on FALLING Edge)
   //      delay_cycles(2);
```

```
      }


   //  Convert and load data.
      //  Conversion begins on 1st falling edge with MSB first
      //  ... then add 8 falling edges on SClk to complete the next
conversion


      //  Select ADC Chip: initiate conversion cycle
      output_low(ADC_CS);   //  Chip Select (active LOW)
      output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
    //     delay_cycles(2);

      //The following loop gets a single reading from the adc
       for ( bits=15; bits>0; bits = bits-1)
       {
       output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
    //     delay_cycles(2);
       output_low(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
   //     delay_cycles(2);
       if (INPUT(ADC_IN)) BIT_SET(volts, bits);
       }


      //  add 8 falling edges on SCLK to complete the conversion for the
next cycle
      for (c = 1; c <=9; c = c + 1)
      {
      output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
   //     delay_cycles(2);
      output_low(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
    //     delay_cycles(2);
      }


      //De-select the ADC chip
      OUTPUT_BIT(ADC_CS, 1);   //  Chip Select (active LOW)
      OUTPUT_BIT(ADC_CLK, 1);   //  Serial Clock (active on FALLING Edge)
    //     delay_cycles(2);


      ADC_Val = volts; // Set the analog_in value

}

void Get_Data()
{

int element;


   for ( element=0; element <=4; ++element)  //gets 5 values
      {
      Read_16();
      analog_in[element] = ADC_Val; // Set the analog_in value
      }
```

```
}

//The following Subroutine gets the median of the analog_in array
int16 Median()
{

//Bubble sort Data Array
   int r,k;
   int16 c,Med;

 for ( k=0; k <=4; ++k)
 {
   for ( r=0; r <=4; ++r)
      {
        if (analog_in[r] <  analog_in[r+1] )
          {
             c=analog_in[r+1];              // next 3 statements swap 2
values
             analog_in[r+1] = analog_in[r];
             analog_in[r] = c;

          }
      }
 }
     Med = analog_in[2];

     return Med;


}
void DAC_16(int16 Volts)
{

int bits, c;


     //  Start with DAC INACTIVE
      output_high(DAC_SYNC);   //  Sync (active LOW)
      output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_us(1);

     //  Select DAC Chip: initiate conversion cycle
      output_low(DAC_SYNC);   //  Sync (active LOW)
      delay_us(1);
      output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_us(1);


   //Set Operation mode Normal Operation: (0,0)
    output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
       OUTPUT_BIT(DAC_Data, 0);  //Sets or clears the output bit in
accordance with Voltage variable
         delay_us(1);
       output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
         output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
```

```
         delay_us(1);
       OUTPUT_BIT(DAC_Data, 0);  //Sets or clears the output bit in
accordance with Voltage variable
         delay_us(1);
       output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);


   //The following loop sends a single 16-bit voltage value to the DAC
      for ( bits=15; bits>0; bits = bits-1)
       {
       output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
       OUTPUT_BIT(DAC_Data, BIT_TEST(Voltage, bits));  //Sets or clears
the output bit in accordance with Voltage variable
         delay_us(1);
       output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);

       }



      //  add 6 falling edges on SCLK to complete the conversion for the
next cycle
      for (c = 1; c <=7; c = c + 1)
      {
      output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
      output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
      }


      //De-select the DAC chip
      OUTPUT_BIT(DAC_SYNC, 1);   //  Sync (active LOW)
      OUTPUT_BIT(DAC_CLK, 1);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);

}


//The following subroutine Takes a low and high byte and assembles them
into a
//single 16-bit number.  It is used by first loading your 8-bit numbers
into an
//array called "i2c_buffer" and then call the command:
//Assemble_16(element number of Low_Byte, element number of High_Byte);
//The assembled 16-bit number is returned
int16 Assemble_16(int L_Byte_Add, int H_Byte_Add)
{
  int L_Byte_8;
  int H_Byte_8;
  int16 Assembled;
  int16 Shift;
  int16 L_Byte_16;
```

```
  int16 H_Byte_16;
  int16 H_Shifted;

  L_Byte_8 = i2c_buffer[L_Byte_Add];
  H_Byte_8 = i2c_buffer[H_Byte_Add];
  Shift = 256;  //2^8 = 256  (Multiplying by 2^8 shifts 8 bits left)



  //Turning the 2 8-bit numbers into 16-bit numbers for PIC math
operations
  L_Byte_16 = L_Byte_8;
  H_Byte_16 = H_Byte_8;


  H_Shifted = H_Byte_16*Shift;  //(Multiplying by 2^8 shifts 8 bits left)
  Assembled = H_Shifted + L_Byte_16;  //Adding the low byte to complete
operation

  return Assembled;


}


//The following subroutine splits a 16-bit number into its high and low
bytes
//It is used by calling "Split_16(16-bit number to split, 0 or 1)"
//0 will return Low_Byte and 1 will return High_Byte
void Split_16(int16 To_Split){

int16 Shift;
int16 AND16bit;
int16 H_Byte_16;
int16 L_Byte_16;


Shift = 256;  //2^8 = 256  (Dividing by 2^8 shifts 8 bits right)
AND16bit = 255;  //0b0000000011111111

H_Byte_16 = To_Split / Shift;  //(Dividing by 2^8 shifts 8 bits right)
L_Byte_16 = To_Split & AND16bit; //ANDing with 255 gives you the low byte

//Turning the numbers into 8-bit pieces
H_Byte = H_Byte_16;
L_Byte = L_Byte_16;

}

//Interrupt subroutine will be triggered when new USB_Settings are
available
//External Interrupt on B2
#INT_EXT2
void ext2_isr() {
//i2c_buffer is filled
int c;
int i = 0;
```

```
int n;
int r = 0;
int j = 1;

for (c = 2; c <=11; c = c + 1){
i2c_buffer[i] = Read_Slave(0xB2, c);
i = i+1;
}

for (n = 0; n <= 5; n=n+1){
USB_VARS[n] = Assemble_16(r, j);
r = r+2;
j = j+2;
}

Charge_Time = USB_VARS[0];
Read_Time = USB_VARS[1];
Short_Time = USB_VARS[2];
Voltage = USB_VARS[3];
Cycles = USB_VARS[4];

DAC_16(Voltage);//Update DAC value
charge_delay = 0;
read_delay = 0;
short_delay = 0;
charging = 0;
shorting = 0;
reading = 0;
Go = 1;
Cyc_Num = 0;
output_high(RED_LED);
output_low(GREEN_LED);
output_low(YELLOW_LED);

if (Charge_Time == 5000 && Short_Time == 5001 && Read_Time == 5002 &&
Voltage == 65534 && Cycles == 4){output_high(GREEN_LED);}
}

 #int_CCP1  //interupts every 1ms
void  CCP1_isr(void)
{

 if(Go == 1){
 if(charging == 1){
 charge_delay = charge_delay+2;
 }

 if(reading == 1){
 read_delay = read_delay+2;
 }

 if(shorting == 1){
 short_delay = short_delay+2;
 }


 output_toggle(YELLOW_LED);
```

```
 //Read_16();  //reads ADC
 //Split_16(ADC_Val);
 Get_Data();  //Fills the Analog_in array to be median filtered
 //Split ADC value into separate bytes and send them via I2C

 val = Median();
 Split_16(val);

 Write_Slave(0xB2,0,L_Byte);
 Write_Slave(0xB2,1,H_Byte);
 }


}

#int_CCP2  //interupts every 100us
void  CCP2_isr(void)
{
 //enable_interrupts(INT_CCP1);
// disable_interrupts(INT_CCP2);
 //enable_interrupts(GLOBAL);

 //Take and send data
 //Read_16();  //reads ADC
// if(Go == 1){
// output_toggle(YELLOW_LED);

 //Read_16();  //reads ADC
 //Split_16(ADC_Val);
// Get_Data();  //Fills the Analog_in array to be median filtered
 //Split ADC value into separate bytes and send them via I2C
// Split_16(Median());

// Write_Slave(0xB2,0,L_Byte);
// Write_Slave(0xB2,1,H_Byte);
// }
 //enable_interrupts(INT_CCP2);
}


void main (void) {

ext_int_edge( 2, L_TO_H); // Set up PIC18 EXT2 (PIN_B2)
enable_interrupts(INT_EXT2);
setup_ccp1(CCP_COMPARE_INT|CCP_COMPARE_RESET_TIMER);
setup_ccp2(CCP_COMPARE_INT|CCP_COMPARE_RESET_TIMER);
enable_interrupts(INT_CCP1);
//enable_interrupts(INT_CCP2);
enable_interrupts(global);

T1CON = 0b11100101;  //Setup Timer1
T3CON = 0b10101101;  //Setup Timer3 (use cpu clk, div_by_4, CCP1 = Timer1,
CCP2 = Timer3)


//  1/((24000000/4)/4) = .666667us per timer increment
```

```
CCP_1 = 1499;//CCP2*.667us = 1ms  CCP2 = 1499 for a 1ms interrupt
//CCP_2 = 149; //CCP1*.667us = 100us
CCP_2 = 149; //CCP1*.667us = 100us
   //initialize ADC
   adc_init();

 delay_ms(250);
 //DAC_16(0b1111111111111111);

//USB_VARS[0] = 5000; //in ms
//USB_VARS[1] = 5000; //in ms
//USB_VARS[2] = 5; //in ms
//USB_VARS[3] = 65000;
//USB_VARS[4] = 5;
//Short_Time = USB_VARS[2];

output_high(GREEN_LED);

   while(TRUE){
//output_high(GREEN_LED);
//output_low(YELLOW_LED);

if(charging == 0 && reading == 0 && shorting == 0 && Go == 1){

   //Charge_Time = USB_VARS[0];
   //Read_Time = USB_VARS[1];
  // Short_Time = USB_VARS[2];
  // Voltage = USB_VARS[3];
 //  Cycles = USB_VARS[4];


  //DAC_16(Voltage);

   //disable_interrupts(GLOBAL);
   Go = 0;
   output_low(Charge_Cap);
   output_low(Short_Cap);
   output_high(Charge_Cap);
   output_high(Cap_Relay);


   charging =1;  //Sets charging flag
   reading = 0;
   shorting = 0;

   delay_ms(10);
   Write_Slave(0xB2,2,1);//Tell Slave to tell VB that Status is charging
   delay_ms(2);
   charge_delay = 0;
  // enable_interrupts(GLOBAL);
  Go = 1;

}

if(charging == 1 && Go == 1){
// output_high(GREEN_LED);
 //output_low(RED_LED);
```

```
// reading = 0;
 if(charge_delay >= Charge_Time){

// disable_interrupts(GLOBAL);
Go = 0;
 charging = 0;
 reading = 0;
 shorting = 0;
 charge_delay = 0;


 output_low(Charge_Cap); //


 delay_ms(10);
 Write_Slave(0xB2,2,2);//Tell Slave to tell VB that Status is shorted
 delay_ms(2);
 output_high(Short_Cap); //  Short Cap
// delay_ms(10);
 shorting = 1;
 short_delay = 0;
// enable_interrupts(GLOBAL);
Go = 1;


 }
 }

 if (shorting ==1 && Go == 1){
 if (short_delay >= Short_Time){

// disable_interrupts(GLOBAL);
Go = 0;
 shorting = 0;
 charging = 0;
 reading = 0;
 short_delay = 0;

  //delay_ms(Short_Time);   //  Short Time
 output_low(Cap_Relay);
 output_low(Short_Cap);  //


 delay_ms(10);
 Write_Slave(0xB2,2,3); //Tell Slave to tell VB that Status is reading
 delay_ms(2);
 reading = 1;
 read_delay = 0;
 //enable_interrupts(GLOBAL);
 Go = 1;
 }

 }

 if(reading ==1 && Go == 1){
 //output_high(RED_LED);
 //output_low(GREEN_LED);
```

```
 if(read_delay >= Read_Time){

 //   disable_interrupts(GLOBAL);
 Go = 0;
    reading = 0;
    shorting = 0;
    charging = 0;
    read_delay = 0;
    delay_ms(10);
    Write_Slave(0xB2,2,1);//Tell Slave to tell VB that Status is charging
    delay_ms(2);
  output_low(Short_Cap);
  output_high(Charge_Cap);
  output_high(Cap_Relay);
  charging = 1;
  //delay_ms(10);
  charge_delay = 0;
 // enable_interrupts(GLOBAL);
 Go = 1;


    Cyc_Num = Cyc_Num + 1;
   if(Cyc_Num >= Cycles){
   Go = 0;
   Cyc_Num = 0;
   output_high(GREEN_LED);
   output_low(RED_LED);
   output_low(YELLOW_LED);
   }
   else{output_high(RED_LED); output_low(GREEN_LED);}

 }
 }


   }

}
```

Source code for GUI:

```
Imports System
Imports System.IO
Imports System.Windows.Forms
Imports System.Security.Permissions


Public Class Data_Collect
    ' vendor and product IDs
    Private Const VendorID As Short = &H461    'Replace with your device's
    Private Const ProductID As Short = &H20      'product and vendor IDs

    ' read and write buffers
    Private Const BufferInSize As Short = 5 'Size of the data buffer coming IN to the
PC
```

```
    Private Const BufferOutSize As Short = 4    'Size of the data buffer going OUT
from the PC
    Dim BufferIn(BufferInSize) As Byte          'Received USB data will be stored here
- the first byte in the array is unused
    Dim BufferOut(BufferOutSize) As Byte    'Transmitted USB data is stored here - the
first item in the array must be 0

    ' Other Global Varables
    Dim Value_In As UInt16                  'Concatanated bytes to make a 16 bit value

    'Devin's House
    Dim DataFileName As String = "C:\Users\dhubbard\Desktop\Cap_Data\test.txt"

    'Phillips Computer
    'Dim DataFileName As String = "C:\Users\CASE\Desktop\Cap_Data\test.txt"

    Dim Take_Data As Boolean = False
    Dim Take_Data1 As Boolean = False
    Dim count As ULong = 0
    Dim count2 As ULong = 0
    Public maximumvalue As Integer = 0
    Public maximumvalue2 As Integer = 0
    Dim data_in(1) As UInt16
    Dim data_in2(1) As UInt16
    Dim datafile As New System.IO.StreamWriter(DataFileName, True)

    Dim Volts As UInt16 = 50000
    Dim Charge_tim As UInt16 = 5000
    Dim Read_tim As UInt16 = 5000
    Dim Short_tim As UInt16 = 500
    Dim Cycs As UInt16 = 3
    Dim Zer As UInt16 = 0
    Dim Status As UInt16 = 0
    Dim Next_read As UInt16 = 0
    Dim Modflag As Boolean = False
    Dim value As UInt16 = 0
    Dim value1 As UInt16 = 0
    Dim maxholder As UInt16 = 0
    Dim minholder As UInt16 = 0
    Dim holder As UInt16 = 0
    Dim plot_count As Integer = 0
    Dim plot_count2 As Integer = 0
    Dim stat_flag As Boolean = False

    ' ****************************************************************
    ' when the form loads, connect to the HID controller - pass
    ' the form window handle so that you can receive notification
    ' events...
    '****************************************************************
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        ' do not remove!
        ConnectToHID(Me)

        currentLogFile.Text = DataFileName
        Voltage_16.Text = CStr(Volts)
        Charge_Time_16.Text = CStr(Charge_tim)
        Short_Time_16.Text = CStr(Short_tim)
```

```
        Read_Time_16.Text = CStr(Read_tim)
        Cycles_16.Text = CStr(Cycs)

    End Sub


    '*****************************************************************
    ' disconnect from the HID controller...
    '*****************************************************************
    Private Sub Form1_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
        DisconnectFromHID()
    End Sub

    '*****************************************************************
    ' a HID device has been plugged in...
    '*****************************************************************
    Public Sub OnPlugged(ByVal pHandle As Integer)
        'TextBox1.Text = "Yay"
        If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID
Then
            ' ** YOUR CODE HERE **
            USB_Label.Text = "Connected"
        End If
    End Sub

    '*****************************************************************
    ' a HID device has been unplugged...
    '*****************************************************************
    Public Sub OnUnplugged(ByVal pHandle As Integer)
        If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID
Then
            hidSetReadNotify(hidGetHandle(VendorID, ProductID), False)
            ' ** YOUR CODE HERE **
            USB_Label.Text = "Disconnected"
        End If
    End Sub

    '*****************************************************************
    ' controller changed notification - called
    ' after ALL HID devices are plugged or unplugged
    '*****************************************************************
    Public Sub OnChanged()
        ' get the handle of the device we are interested in, then set
        ' its read notify flag to true - this ensures you get a read
        ' notification message when there is some data to read...
        Dim pHandle As Integer
        pHandle = hidGetHandle(VendorID, ProductID)
        hidSetReadNotify(hidGetHandle(VendorID, ProductID), True)
    End Sub

    '*****************************************************************
    ' on read event...
    '*****************************************************************
    Public Sub OnRead(ByVal pHandle As Integer)
        ' read the data (don't forget, pass the whole array)...
        If hidRead(pHandle, BufferIn(0)) Then
            ' ** YOUR CODE HERE **
```

```
            ' first byte is the report ID, e.g. BufferIn(0)
            ' the other bytes are the data from the microcontroller...

            'Bring two bytes into one 16-bit value, BufferIn(1) = HByte, BufferIn(2) =
LByte
            value = BitConverter.ToUInt16(BufferIn, 1)

            ADC_Val_textbox.Text = CStr(value)
            TextBox4.Text = CStr(Status)



            ' The following three if statements ensure help eliminate erroneous data
            If Status = 0 Then Take_Data = False 'Fixes false starts


            minholder = Math.Min(value, value1)
            maxholder = Math.Max(value, value1)
            holder = maxholder - minholder
            If holder > 500 Then Take_Data1 = False 'Helps with data blips
            If holder < 500 Then Take_Data1 = True 'Helps with data blips


            If value > 4 Then value1 = value




            'If stat_flag = False Then
            If value < 4 And value > 0 Then
                Take_Data = False
                Modflag = True
                plot_count = 10
                plot_count2 = 10
                stat_flag = True

            End If

            If Modflag = True Then
                Next_read = Next_read + 1
            End If



            If value = 1 Then
                BufferOut(1) = 16 'Tell the PIC that Charging indicator was received
                hidWriteEx(VendorID, ProductID, BufferOut(0))
                Status = 1
                count = 0
                count2 = 0
                maximumvalue = 0
                maximumvalue2 = 0
                ReDim data_in(count)
                ReDim data_in2(count2)
```

```
            End If
            If value = 2 Then
                BufferOut(1) = 17 'Tell the PIC that a new Voltage value is available
                hidWriteEx(VendorID, ProductID, BufferOut(0))
                Status = 2
            End If
            If value = 3 Then
                BufferOut(1) = 18 'Tell the PIC that a new Voltage value is available
                hidWriteEx(VendorID, ProductID, BufferOut(0))
                Status = 3
            End If

            'End If


            If Take_Data = True And Take_Data1 = True Then
                Dim datafile As New System.IO.StreamWriter(DataFileName, True)
                datafile.WriteLine(ControlChars.Tab & value & ControlChars.Tab &
Status)
                datafile.Close()

                If Status = 3 Then
                    plot_count = plot_count + 1

                    If plot_count > 10 Then
                        data_in(count) = value
                        If value > maximumvalue Then maximumvalue = value
                        picDataPlot.Refresh()
                        count = count + 1
                        ReDim Preserve data_in(count)
                        'ReDim data_in(count)
                        plot_count = 0
                    End If

                End If
                'End If

                plot_count2 = plot_count2 + 1
                If plot_count2 > 10 Then
                    data_in2(count2) = value
                    If value > maximumvalue2 Then maximumvalue2 = value
                    picDataPlot2.Refresh()
                    count2 = count2 + 1
                    ReDim Preserve data_in2(count2)
                    'ReDim data_in2(count2)
                    plot_count2 = 0
                End If


            End If




            If Next_read > 2 Then
                Take_Data = True
                Next_read = 0
```

```
                Modflag = False
            End If


        End If
    End Sub


    'Red Led Toggle
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click

        'BufferOut(1) = 1
        If BufferOut(1) >= 1 Then
            BufferOut(1) = 0
        Else : BufferOut(1) = 1
        End If

        hidWriteEx(VendorID, ProductID, BufferOut(0))

        'TextBox1.Text = "Hello World"
    End Sub



    'Data Filename textbox
    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles newFilename.TextChanged
        'DataFileName = newFilename.Text
    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles currentLogFile.TextChanged

    End Sub

    Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ADC_Val_textbox.TextChanged

    End Sub

    Private Sub TextBox4_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox4.TextChanged

    End Sub

    Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles USB_Label.Click

    End Sub

    Public Sub Graph_paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles picDataPlot.Paint
        Dim picturebox_height, picturebox_width As Double
        Dim gfx As System.Drawing.Graphics
        Dim points() As Point
        Dim Control() As Point
```

```
        Dim graph_1 As Pen
        Dim i As Integer = 0
        Dim x_centre As Double
        Dim y_centre As Double
        Dim x_offset As Double
        Dim y_offset As Double
        On Error Resume Next
        x_offset = 10
        On Error Resume Next
        y_offset = 10
        gfx = e.Graphics
        picturebox_width = picDataPlot.Width - x_offset
        picturebox_height = (picDataPlot.Height - y_offset)
        gfx.TranslateTransform(x_offset, picturebox_height)
        ReDim Preserve points(count)
        ReDim Preserve Control(count)
        For i = 0 To count
            Control(i).X = i * (picturebox_width / (count))
            Control(i).Y = (-1) * ((picturebox_height) / maximumvalue)
            points(i).X = i * (picturebox_width / (count))
            points(i).Y = (-1) * data_in(i) * ((picturebox_height) / maximumvalue)

        Next
        gfx.DrawLines(Pens.Red, points)
        gfx.DrawLines(Pens.Blue, Control)
    End Sub
    Public Sub Graph_paint2(ByVal sender2 As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles picDataPlot2.Paint
        Dim picturebox_height2, picturebox_width2 As Double
        Dim gfx2 As System.Drawing.Graphics
        Dim points2() As Point
        Dim Control2() As Point
        Dim graph_12 As Pen
        Dim i2 As Integer = 0
        Dim x_centre2 As Double
        Dim y_centre2 As Double
        Dim x_offset2 As Double
        Dim y_offset2 As Double
        On Error Resume Next
        x_offset2 = 10
        On Error Resume Next
        y_offset2 = 10
        gfx2 = e.Graphics
        picturebox_width2 = picDataPlot2.Width - x_offset2
        picturebox_height2 = (picDataPlot2.Height - y_offset2)
        gfx2.TranslateTransform(x_offset2, picturebox_height2)
        ReDim Preserve points2(count2)
        ReDim Preserve Control2(count2)
        For i2 = 0 To count2
            Control2(i2).X = i2 * (picturebox_width2 / (count2))
            'Control2(i2).Y = (-1) * 5
            Control2(i2).Y = (-1) * ((picturebox_height2) / maximumvalue2)
            points2(i2).X = i2 * (picturebox_width2 / (count2))
            points2(i2).Y = (-1) * data_in2(i2) * ((picturebox_height2) /
maximumvalue2)

        Next
        gfx2.DrawLines(Pens.Red, points2)
```

```
        gfx2.DrawLines(Pens.Blue, Control2)
    End Sub

    'Set File Name
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles createFile.Click

        Take_Data = False
        Volts = Val(Voltage_16.Text)
        Charge_tim = Val(Charge_Time_16.Text)
        Read_tim = Val(Read_Time_16.Text)
        Short_tim = Val(Short_Time_16.Text)
        Cycs = Val(Cycles_16.Text)
        'DataFileName = "D:\Zack\Collected_Data\" + newFilename.Text
        DataFileName = newFilename.Text
        Dim datafile As New System.IO.StreamWriter(DataFileName, True)
        datafile.WriteLine(ControlChars.Tab & Volts & ControlChars.Tab & Zer)
        datafile.WriteLine(ControlChars.Tab & Charge_tim & ControlChars.Tab & Zer)
        datafile.WriteLine(ControlChars.Tab & Read_tim & ControlChars.Tab & Zer)
        datafile.WriteLine(ControlChars.Tab & Short_tim & ControlChars.Tab & Zer)
        datafile.WriteLine(ControlChars.Tab & Cycs & ControlChars.Tab & Zer)
        datafile.Close()
        currentLogFile.Text = newFilename.Text
    End Sub


    Private Sub collectData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles collectData.Click
        Take_Data = True
    End Sub

    Private Sub pauseCollect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles pauseCollect.Click
        Take_Data = False
    End Sub

    Private Sub TextBox3_TextChanged_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ADC_Val_textbox.TextChanged

    End Sub

    Private Sub Load_Settings_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Load_Settings.Click

        count = 0
        count2 = 0
        ReDim data_in(1)
        ReDim data_in2(1)
        maximumvalue = 0
        maximumvalue2 = 0

        picDataPlot.Refresh()
        picDataPlot2.Refresh()

        value1 = 0

        currentLogFile.Text = DataFileName
```

```
        Status = 0
        Volts = Val(Voltage_16.Text)
        Charge_tim = Val(Charge_Time_16.Text)
        Read_tim = Val(Read_Time_16.Text)
        Short_tim = Val(Short_Time_16.Text)
        Cycs = Val(Cycles_16.Text)



        'Convert 16-bit numbers into separate bytes\
        Dim Volts_8 As Byte() = BitConverter.GetBytes(Volts)
        Dim C_tim_8 As Byte() = BitConverter.GetBytes(Charge_tim)
        Dim R_tim_8 As Byte() = BitConverter.GetBytes(Read_tim)
        Dim S_tim_8 As Byte() = BitConverter.GetBytes(Short_tim)
        Dim Cycs_8 As Byte() = BitConverter.GetBytes(Cycs)


        ''''''Load each individually into Bufferout and send via USB to the Slave
Pic'''''''''

        BufferOut(1) = 22 'Tell the PIC that a new Voltage value is available
        hidWriteEx(VendorID, ProductID, BufferOut(0))
        BufferOut(1) = Volts_8(0)
        BufferOut(2) = Volts_8(1)
        hidWriteEx(VendorID, ProductID, BufferOut(0))

        BufferOut(1) = 23 'Tell the PIC that a new Charge time is available
        hidWriteEx(VendorID, ProductID, BufferOut(0))
        BufferOut(1) = C_tim_8(0)
        BufferOut(2) = C_tim_8(1)
        hidWriteEx(VendorID, ProductID, BufferOut(0))

        BufferOut(1) = 24 'Tell the PIC that a new Read time is available
        hidWriteEx(VendorID, ProductID, BufferOut(0))
        BufferOut(1) = R_tim_8(0)
        BufferOut(2) = R_tim_8(1)
        hidWriteEx(VendorID, ProductID, BufferOut(0))

        BufferOut(1) = 26 'Tell the PIC that a new Short time value is available
        hidWriteEx(VendorID, ProductID, BufferOut(0))
        BufferOut(1) = S_tim_8(0)
        BufferOut(2) = S_tim_8(1)
        hidWriteEx(VendorID, ProductID, BufferOut(0))

        BufferOut(1) = 27 'Tell the PIC that a new Cycles value is available
        hidWriteEx(VendorID, ProductID, BufferOut(0))
        BufferOut(1) = Cycs_8(0)
        BufferOut(2) = Cycs_8(1)
        hidWriteEx(VendorID, ProductID, BufferOut(0))


        Take_Data = True

    End Sub

    Private Sub picDataPlot_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles picDataPlot.Click
```

```
            Refresh()
        End Sub


    End Class
```

APPENDIX 3: STRESS-STRAIN CURVES FROM ENGINEERED CARDIAC TISSUE USING THE INSTRUMENTATION DESCRIBED IN CHAPTER 2

We hypothesize that the relative amount of myofiber alignment can be tracked empirically by monitoring the elasticity modulus of the sample. We further hypothesize that the full stress-strain profile can be mapped out for samples in the appropriate size range for the described device. For the self-organized tissues we are researching, we expect the stress-strain curve to have a shape similar to that of a soft elastic band with an added element of viscoelastic creep. In order to assuage errors from the expected time and temperature dependent viscoelastic behavior, strain rate and temperature will be held constant. The predicted stress-strain behavior of the cardiac patches is shown in the figure for comparison with other types of materials. The instrumentation methods for this research and a hypothetical experiment are discussed.

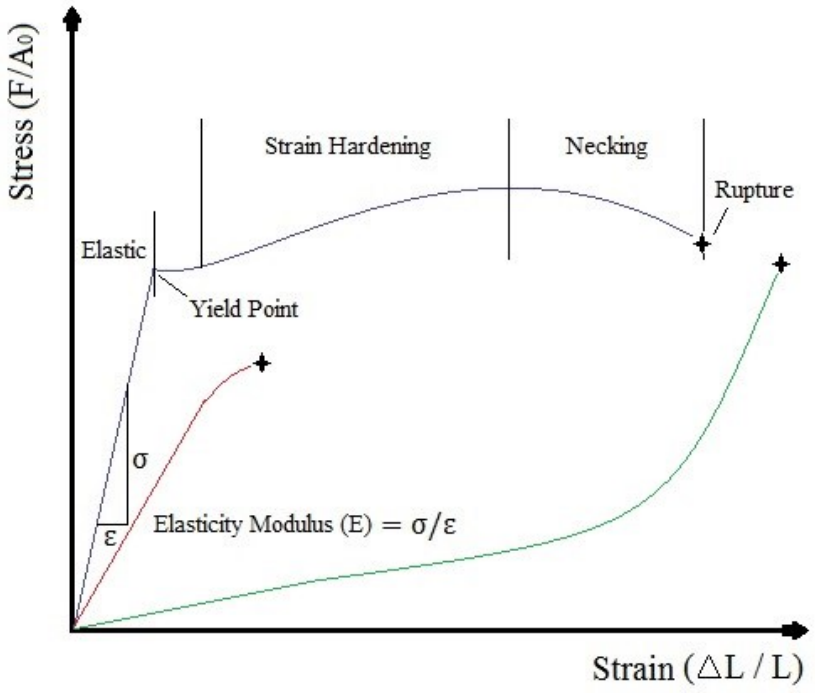


Figure 1. Stress-strain curves for different types of materials. Blue represents the generic shape of a ductile material. Red represents a brittle material. Green represents the expected result for engineered cardiac tissue.

Additional Calibrations

With knowledge of the armature spring constant and the current to Lorentz force relationship, stress-strain information can be measured from attached tissue samples. The position of the armature is tracked along with the current through the VCA coil as the actuator is moved. The electrical current necessary to reach each armature position will be affected by the force from the deforming sample. By comparison to the known unloaded condition, the force required to deform the sample can be dynamically tracked. For example, the difference between the force required to reach point A with the sample attached and the point A force in the unloaded case is the stress of the sample.

The 16-bit digital to analog converter allows 65535 current steps over the range set by a bias resistor on the current amplifier transistor circuit. In the configuration presented here, the current can be linearly adjusted from 0 mA up to 375 mA which yields a precision of approximately 5μA. Using this precision, the unloaded system can theoretically achieve a positional precision of < 300nm under ideal conditions.

Hypothetical Stress-Strain Experiment

Because the engineered patches are formed from cell monolayers, they will have some tension preload even prior to attachment of the VCA device. As long as the preload tension is in the elastic region, the critical points of the stress-strain curve will still be attainable. The sample is attached to the end effector of the calibrated VCA system using woven silk suture. The dimensions of the cardiac patch and initial displacement are recorded. The electrical current and therefore force is steadily increased and monitored along with position. The force on the sample at each position is isolated from the raw signal using the method described in the previous section. Normalizing the force to the initial cross-sectional area of the patch gives the apparent stress and normalizing the displacement to the initial length gives strain.

The tissue begins with elastic deformation wherein the stress-strain curve is linear with a shallow slope equal to the modulus of elasticity. A release of the stress in this region will return the sample to the initial length. When the sample is stretched past the yield point, plastic deformation begins as the myofibers begin to pull apart. Plastic deformation is irreversible. As the tissue resists rupture and myofiber bonds are pulled, the sample may

become more brittle and significantly higher stress may be required to continue strain. As the number of broken cell-cell and cell-matrix bonds increases, the sample will begin to tear at the weakest point and eventually rupture completely.

APPENDIX 4: SAMPLE MICROCONTROLLER SOURCE CODE FOR VOICE COIL ACTUATOR SYSTEM

Microcontroller code for generating a 1Hz stimulus at 2mm displacement on the VCA system:

```
#include <18F4550.h>
#fuses HSPLL, NOWDT, NOPROTECT, NOLVP, USBDIV, PLL5, CPUDIV3, VREGEN, PUT
#use delay (clock=24000000)


//  included headers and source code
#include <pic18_usb.h>  //Select PIC for USB
#include <usb_desc_hid.h>  //USB Application (in this case HID_IO)
#include <usb.c>  //USB commands.  Look through this to understand specifics of
commands.



//  PCB board pin defines
#define GREEN_LED PIN_C0
#define YELLOW_LED PIN_C1
#define RED_LED PIN_C2

#define ADC_CS PIN_A0
#define ADC_CLK PIN_A1
#define ADC1_IN PIN_A4
#define ADC2_IN PIN_A3

#define DAC_SYNC PIN_D2
#define DAC_CLK PIN_D1
#define DAC_Data PIN_D0

#define GND1 PIN_D6
#define PWR1 PIN_D7
#define GND2 PIN_D4
#define PWR2 PIN_D5

//Globals
int16 Med1;
int16 Med2;

int16 ADC_Val1;
int16 ADC_Val2;

int16 Full_Range_Direction1;
```

```
int16 Full_Range_Direction2;
int16 Baseline;

int16 Output = 1;
int16 Pos;
int test_flag = 1;


int16 Displacement_in_SensorSteps = 1334;
float32 Desired_Displacement = 2;   //mm


float32 Desired_Time = 500000;  //us
int16 Disp_INC;
int16 Step_Time;
int Direction = 0;
int1 H_Bridge = 0;
int step;
float32 Disp_Volts;
int16 Base_Volts;


void DAC_16(int16 Volts)
{

int bits, c;


     //  Start with DAC INACTIVE
      output_high(DAC_SYNC);   //  Sync (active LOW)
      output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_us(1);

     //  Select DAC Chip: initiate conversion cycle
      output_low(DAC_SYNC);   //  Sync (active LOW)
      delay_us(1);
      output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_us(1);


   //Set Operation mode Normal Operation: (0,0)
    output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
       OUTPUT_BIT(DAC_Data, 0);  //Sets or clears the output bit in accordance with
Voltage variable
         delay_us(1);
       output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
         output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
       OUTPUT_BIT(DAC_Data, 0);  //Sets or clears the output bit in accordance with
Voltage variable
         delay_us(1);
       output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);


   //The following loop sends a single 16-bit voltage value to the DAC
```

```
      for ( bits=15; bits>0; bits = bits-1)
       {
       output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
       OUTPUT_BIT(DAC_Data, BIT_TEST(Volts, bits));  //Sets or clears the output bit
in accordance with Voltage variable
         delay_us(1);
       output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);

       }



      //  add 6 falling edges on SCLK to complete the conversion for the next cycle
      for (c = 1; c <=7; c = c + 1)
      {
      output_high(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
      output_low(DAC_CLK);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);
      }


      //De-select the DAC chip
      OUTPUT_BIT(DAC_SYNC, 1);   //  Sync (active LOW)
      OUTPUT_BIT(DAC_CLK, 1);   //  Serial Clock (active on FALLING Edge)
         delay_us(1);

}


void Read_16()
{
int16 volts1 = 0b0000000000000000;
int16 volts2 = 0b0000000000000000;

int bits, c;

 //volts = 0b0000000000000000;
        //  Start with ADC INACTIVE
      output_high(ADC_CS);   //  Chip Select (active LOW)
      output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
      //delay_us(1);
      delay_cycles(2);

   //  Run 24 cycles of SClk to complete the next conversion
      for (c = 1; c <=25; ++c)
      {
      output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_cycles(2);
      output_low(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_cycles(2);
      }


   //  Convert and load data.
      //  Conversion begins on 1st falling edge with MSB first
```

```
      //  ... then add 8 falling edges on SClk to complete the next conversion


      //  Select ADC Chip: initiate conversion cycle
      output_low(ADC_CS);   //  Chip Select (active LOW)
      output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_cycles(2);

      //The following loop gets a single reading from the adcs
       for ( bits=15; bits>0; bits = bits-1)
       {
       output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
       delay_cycles(2);
       output_low(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
       delay_cycles(2);
       if (INPUT(ADC1_IN)) BIT_SET(volts1, bits);
       if (INPUT(ADC2_IN)) BIT_SET(volts2, bits);
       }


      //  add 8 falling edges on SCLK to complete the conversion for the next cycle
      for (c = 1; c <=9; c = c + 1)
      {
      output_high(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_cycles(2);
      output_low(ADC_CLK);   //  Serial Clock (active on FALLING Edge)
      delay_cycles(2);
      }


      //De-select the ADC chip
      OUTPUT_BIT(ADC_CS, 1);   //  Chip Select (active LOW)
      OUTPUT_BIT(ADC_CLK, 1);   //  Serial Clock (active on FALLING Edge)
      delay_cycles(2);


      ADC_Val1 = volts1;   // Set the analog_in1 value
      ADC_Val2 = volts2;  // Set the analog_in2 value

}

void Med_11_Read_16()
{

int element;
int16 analog_in1[11];
int16 analog_in2[11];
   int r = 0;
   int k = 0;
   int16 c;


   for ( element=0; element <=10; ++element)  //gets 11 values
      {
      Read_16();
      analog_in1[element] = ADC_Val1; // Set the analog_in1 value
      analog_in2[element] = ADC_Val2; // Set the analog_in2 value
      }
```

```
//Median filter the two analog_in arrays

//Bubble sort Data Array


 for ( k=0; k <=10; ++k)
 {
   for ( r=0; r <=10; ++r)
      {
        if (analog_in1[r] <  analog_in1[r+1] )
          {
             c=analog_in1[r+1];              // next 3 statements swap 2 values
             analog_in1[r+1] = analog_in1[r];
             analog_in1[r] = c;

          }

          if (analog_in2[r] <  analog_in2[r+1] )
          {
             c=analog_in2[r+1];              // next 3 statements swap 2 values
             analog_in2[r+1] = analog_in2[r];
             analog_in2[r] = c;

          }



      }
 }
     Med1 = analog_in1[5];
     Med2 = analog_in2[5];

}

void Med_5_Read_16()
{

int element;
int16 analog_in1[5];
int16 analog_in2[5];
   int r = 0;
   int k = 0;
   int16 c;


   for ( element=0; element <=4; ++element)  //gets 5 values
      {
      Read_16();
      analog_in1[element] = ADC_Val1; // Set the analog_in1 value
      analog_in2[element] = ADC_Val2; // Set the analog_in2 value
      }

//Median filter the two analog_in arrays

//Bubble sort Data Array
```

```
 for ( k=0; k <=4; ++k)
 {
   for ( r=0; r <=4; ++r)
      {
        if (analog_in1[r] <  analog_in1[r+1] )
          {
             c=analog_in1[r+1];              // next 3 statements swap 2 values
             analog_in1[r+1] = analog_in1[r];
             analog_in1[r] = c;

          }

          if (analog_in2[r] <  analog_in2[r+1] )
          {
             c=analog_in2[r+1];              // next 3 statements swap 2 values
             analog_in2[r+1] = analog_in2[r];
             analog_in2[r] = c;

          }



      }
 }
     Med1 = analog_in1[2];
     Med2 = analog_in2[2];

}

int16 Get_Position()
{
 int16 Position;

 //Med_11_Read_16();
 Med_5_Read_16();

 Position = (Med2 + 26000) - Med1;

 return Position;

}
int16 Get_Position_11Reads()
{
 int16 Position;

 Med_11_Read_16();
 //Med_5_Read_16();

 Position = (Med2 + 26000) - Med1;

 return Position;

}

void Calibrate_Sensor()
{

   DAC_16(0);
```

```
   //Read Baseline
   output_low(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_low(PWR2);
   delay_ms(8000);

   Baseline = Get_Position_11Reads();


}

void Calibrate_Forces()
{

  Baseline = Baseline - 100;

//Direction2: Away from the Culture dish
   output_low(GND1);
   output_high(PWR1);
   output_high(GND2);
   output_low(PWR2);
   delay_ms(1);
   Pos = Get_Position_11Reads();

   while(Pos > (Baseline - Displacement_in_SensorSteps))
   {
   Pos = Get_Position_11Reads();
   Output = Output + 200;
   DAC_16(Output);
   }

   output_low(RED_LED);
   output_high(YELLOW_LED);
   Disp_Volts = Output;

   while(Pos < Baseline)
   {
   Pos = Get_Position_11Reads();

   if(Pos >= (Baseline - 80)){break;}
   else {Output = Output - 100;}

   DAC_16(Output);
   }

   output_low(RED_LED);
   output_high(YELLOW_LED);
   Base_Volts = Output;


}

void main (void) {

ext_int_edge( 2, L_TO_H); // Set up PIC18 EXT2 (PIN_B2)
enable_interrupts(INT_EXT2);
```

```
setup_ccp1(CCP_COMPARE_INT|CCP_COMPARE_RESET_TIMER);
setup_ccp2(CCP_COMPARE_INT|CCP_COMPARE_RESET_TIMER);
enable_interrupts(INT_CCP1);
enable_interrupts(global);


 delay_ms(250);

output_high(GREEN_LED);
delay_ms(1000);
output_low(GREEN_LED);
delay_ms(1000);
output_high(GREEN_LED);

Calibrate_Sensor();
Calibrate_Forces();

  //Direction1: Toward the Culture Dish (State: H_Bridge = 1)
   //output_high(GND1);
  // output_low(PWR1);
   //output_low(GND2);
   //output_high(PWR2);
   //delay_ms(10);

  //Direction2: Away from the Culture Dish (State: H_Bridge = 0)
   output_low(GND1);
   output_high(PWR1);
   output_high(GND2);
   output_low(PWR2);
   delay_ms(10);

   Disp_INC = Disp_Volts / 200;
   Step_Time = (Desired_Time / 200)-1187;

while(true){


//Move out to  near Desired Displacement position in steps
if(Direction == 0)
{
Pos = Get_Position();

if(Pos >= (Baseline - 500))
{
for (step=1; step <=200; ++step)
{

if(H_Bridge == 0)
{
Output = Output + Disp_INC;
DAC_16(Output);
delay_us(Step_Time);
}
if(H_Bridge == 1)
{
Output = Output - Disp_INC;
step = step - 1;
if(Output == 0 || Output >= 60000)
```

```
{

   //if(Output >= 60000){Output = 65534 - Output;}
   //if(Output == 0){Output = 1;}
   Output = 1;
   H_Bridge = !H_Bridge;
   if(H_Bridge == 0)
{
   output_low(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_low(PWR2);
//Direction2: Away from the Culture dish
   output_low(GND1);
   output_high(PWR1);
   output_high(GND2);
   output_low(PWR2);
}
   if(H_Bridge == 1)
{
   output_low(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_low(PWR2);
  //Direction1: Toward the Culture dish
   output_high(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_high(PWR2);
}
}
DAC_16(Output);
}

Pos = Get_Position();
if(Pos <= (Baseline - Displacement_in_SensorSteps)){break;}

}
}

Output = Output + 50;
DAC_16(Output);




if(Pos <= (Baseline - Displacement_in_SensorSteps))
{
    Direction = 1;
    output_low(RED_LED);
    output_high(Yellow_LED);
    delay_ms(100);
}


}
```

```
//Move in to near the zero position in steps.
if(Direction == 1) //toward culture dish
{
Pos = Get_Position();

if(Pos <= (Baseline - (Displacement_in_SensorSteps-500)))
{
for (step=1; step <= 200; ++step)
{


if(H_Bridge == 0)
{
Output = Output - Disp_INC;
if(Output == 0 || Output >= 60000)
{

   //if(Output >= 60000){Output = 65534 - Output;}
   //if(Output == 0){Output = 1;}
   Output = 1;
   H_Bridge = !H_Bridge;
   if(H_Bridge == 0)
{
   output_low(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_low(PWR2);
//Direction2: Away from the Culture dish
   output_low(GND1);
   output_high(PWR1);
   output_high(GND2);
   output_low(PWR2);
}
   if(H_Bridge == 1)
{
   output_low(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_low(PWR2);
  //Direction1: Toward the Culture dish
   output_high(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_high(PWR2);
}
}
DAC_16(Output);
delay_us(Step_Time);
}
if(H_Bridge == 1)
{
Output = Output + Disp_INC;
DAC_16(Output);
delay_us(Step_Time);
}

Pos = Get_Position();
if(Pos >= (Baseline)){break;}
```

```
}
}

if(H_Bridge == 0){Output = Output - 50;}
if(H_Bridge == 1){Output = Output + 50;}
if(Output == 0 || Output >= 60000)
{

   //if(Output >= 60000){Output = 65534 - Output;}
   //if(Output == 0){Output = 1;}
   Output = 1;
   H_Bridge = !H_Bridge;
   if(H_Bridge == 0)
{
   output_low(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_low(PWR2);
//Direction2: Away from the Culture dish
   output_low(GND1);
   output_high(PWR1);
   output_high(GND2);
   output_low(PWR2);
}
   if(H_Bridge == 1)
{
   output_low(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_low(PWR2);
  //Direction1: Toward the Culture dish
   output_high(GND1);
   output_low(PWR1);
   output_low(GND2);
   output_high(PWR2);
}
}
DAC_16(Output);


if(Pos >= Baseline) //Returned to base
{
    Direction = 0;
    output_low(YELLOW_LED);
    output_high(RED_LED);
    delay_ms(100);
}
}
}
}
```

REFERENCES

1. Johnson, P. C., Mikos, A. G., Fisher, J. P. & Jansen, J. a. Strategic directions in tissue engineering. *Tissue engineering* **13,** 2827–37 (2007).

2. Wang, H.-S. *et al.* Mesenchymal stem cells in the Wharton's jelly of the human umbilical cord. *Stem cells (Dayton, Ohio)* **22,** 1330–7 (2004).

3. Martin-Rendon, E. *et al.* 5-Azacytidine-treated human mesenchymal stem/progenitor cells derived from umbilical cord, cord blood and bone marrow do not generate cardiomyocytes in vitro at high frequencies. *Vox sanguinis* **95,** 137–48 (2008).

4. Ott, H. C. *et al.* Perfusion-decellularized matrix: using nature's platform to engineer a bioartificial heart. *Nature medicine* **14,** 213–21 (2008).

5. Wang, Y. *et al.* Lineage restriction of human hepatic stem cells to mature fates is made efficient by tissue-specific biomatrix scaffolds. *Hepatology (Baltimore, Md.)* **53,** 293–305 (2011).

6. Gessner, R. C. *et al.* Functional ultrasound imaging for assessment of extracellular matrix scaffolds used for liver organoid formation. *Biomaterials* 1–11 (2013). doi:10.1016/j.biomaterials.2013.08.033

7. Maxwell, J. C. *A Treatise on Electricity and Magnetism, Volume 1*. 450–458 (London: Oxford University Press, 1892).

8. Pease, R. A. Understand capacitor soakage to optimize analog systems. *Electronics Design Magazine* 125–129 (1982).

9. Fattaruso, J. & Wit, M. De. The effect of dielectric relaxation on charge-redistribution A/D converters. *Solid-State Circuits, ...* **25,** 1550–1561 (1990).

10. Coelho, R. *Physics of Dielectrics for the Engineer*. 70–105 (Elsevier Scientific Publishing Company, 1979).

11. Cole, K. S. Dispersion and Absorption in Dielectrics II. Direct Current Characteristics. *The Journal of Chemical Physics* **10,** 98 (1942).

12. Jonscher, A. K. Dielectric relaxation in solids. **57,** (1999).

13. Pease, R. A. What's all this soakage stuff, anyhow? (1998).

14. Kuenen, J. C. & Meijer, G. C. M. Measurement of dielectric absorption of capacitors and analysis of its effects on VCOs. *IEEE Transactions on Instrumentation and Measurement* **45,** 89–97 (1996).

15. Iorga, C. Compartmental Analysis of Dielectric Absorption in Capacitors. *IEEE Transactions on Dielectrics and Electrical Insulation* **7,** 187–192 (2000).

16. DONG, X., MOHAN, R. S., MARRELLI, J. D., SHOUBO, W. & SHOHAM, O. A Modular Differential Dielectric Sensor for Use in Multiphase Separation, Process Measurement, and Control—Part II: Experimental Investigation. *Journal of energy resources technology* **133,**

17. Harris, C. M. *et al.* Dielectric permittivity of microbial suspensions at radio frequencies : a novel method for the real-time estimation of microbial biomass. **9,** 181–186 (1986).

18. Kell, D. B. The principles and potential of electrical admittance spectroscopy: an introduction. *Oxford University Press* 427–468 (1987).

19. Schwan, H. P. Electrical properties of tissues and cell suspensions: mechanisms and models. *Proceedings of 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* A70–A71 doi:10.1109/IEMBS.1994.412155

20. Fehrenbach, R., Comberbach, M. & Pêtre, J. O. On-line biomass monitoring by capacitance measurement. *Journal of biotechnology* **23,** 303–14 (1992).

21. Schenk, P. M. *et al.* Second Generation Biofuels: High-Efficiency Microalgae for Biodiesel Production. *BioEnergy Research* **1,** 20–43 (2008).

22. Sheehan, J. A Look Back at the U . S . Department of Energy ' s Aquatic Species Program — Biodiesel from Algae Office of Fuels Development. (1998).

23. Weyer, K. M., Bush, D. R., Darzins, A. & Willson, B. D. Theoretical Maximum Algal Oil Production. *BioEnergy Research* **3,** 204–213 (2009).

24. Mining, D., Discovery, K., Laboratories, B. & Technologies, L. A Tutorial on Support Vector Machines for Pattern Recognition. **167,** 121–167 (1998).

25. Moraes, R., Valiati, J. F. & Gavião Neto, W. P. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications* **40,** 621–633 (2013).

26. Joachims, T. in *Advances in Kernel Methods - Support Vector Learning* (Scholkopf, B., Burges, C. J. C. & Smola, A. J.) (MIT Press, 1999). at <https://eldorado.uni-dortmund.de/handle/2003/2596>

27. Joachims, T. Learning to Classify Text using Support Vector Machines. (2002). at <http://svmlight.joachims.org/>

28. Vapnik, V. N. *The Nature of Statistical Learning Theory*. (Springer, 1999).

29. Roger, V. L. *et al.* Heart disease and stroke statistics--2012 update: a report from the American Heart Association. *Circulation* **125,** e2–e220 (2012).

30. Guyton, A. C. & Edward, J. *Textbook of Medical Physiology*. 27–42 (Elsevier Inc., 2006).

31. Messina, E. *et al.* Isolation and expansion of adult cardiac stem cells from human and murine heart. *Circulation research* **95,** 911–21 (2004).

32. Segers, V. F. M. & Lee, R. T. Stem-cell therapy for cardiac disease. *Nature* **451,** 937–42 (2008).

33. Beltrami, A. P. *et al.* Adult cardiac stem cells are multipotent and support myocardial regeneration. *Cell* **114,** 763–76 (2003).

34. Laugwitz, K.-L. *et al.* Postnatal isl1+ cardioblasts enter fully differentiated cardiomyocyte lineages. *Nature* **433,** 647–53 (2005).

35. Oh, H. *et al.* Cardiac progenitor cells from adult myocardium: homing, differentiation, and fusion after infarction. *Proceedings of the National Academy of Sciences of the United States of America* **100,** 12313–8 (2003).

36. Bergmann, O. *et al.* Evidence for cardiomyocyte renewal in humans. *Science (New York, N.Y.)* **324,** 98–102 (2009).

37. Radisic, M. & Christman, K. L. Materials science and tissue engineering: repairing the heart. *Mayo Clinic proceedings. Mayo Clinic* **88,** 884–98 (2013).

38. Pilla, J. J. *et al.* Passive ventricular constraint to improve left ventricular function and mechanics in an ovine model of heart failure secondary to acute myocardial infarction. *The Journal of Thoracic and Cardiovascular Surgery* **126,** 1467–1475 (2003).

39. Enomoto, Y. *et al.* Early ventricular restraint after myocardial infarction: extent of the wrap determines the outcome of remodeling. *The Annals of thoracic surgery* **79,** 881–7; discussion 881–7 (2005).

40. Bowen, F. & Jones, S. Restraining acute infarct expansion decreases collagenase activity in borderzone myocardium. *The Annals of thoracic …* **4975,** (2001).

41. Moainie, S. L. *et al.* Infarct restraint attenuates remodeling and reduces chronic ischemic mitral regurgitation after postero-lateral infarction. *The Annals of thoracic surgery* **74,** 444–9; discussion 449 (2002).

42. Kelley, S. T. *et al.* Restraining Infarct Expansion Preserves Left Ventricular Geometry and Function After Acute Anteroapical Infarction. *Circulation* **99,** 135–142 (1999).

43. Huang, N. F., Yu, J., Sievers, R., Li, S. & Lee, R. J. Injectable biopolymers enhance angiogenesis after myocardial infarction. *Tissue engineering* **11,** 1860–6 (2005).

44. Ryu, J. H. *et al.* Implantation of bone marrow mononuclear cells using injectable fibrin matrix enhances neovascularization in infarcted myocardium. *Biomaterials* **26,** 319–26 (2005).

45. Christman, K. L. *et al.* Injectable fibrin scaffold improves cell transplant survival, reduces infarct expansion, and induces neovasculature formation in ischemic myocardium. *Journal of the American College of Cardiology* **44,** 654–60 (2004).

46. Christman, K. L., Fok, H. H., Sievers, R. E., Fang, Q. & Lee, R. J. Fibrin glue alone and skeletal myoblasts in a fibrin scaffold preserve cardiac function after myocardial infarction. *Tissue engineering* **10,** 403–9 (2004).

47. Kofidis, T. *et al.* Novel injectable bioartificial tissue facilitates targeted, less invasive, large-scale tissue restoration on the beating heart after myocardial injury. *Circulation* **112,** I173–7 (2005).

48. Zimmermann, W.-H. *et al.* Engineered heart tissue grafts improve systolic and diastolic function in infarcted rat hearts. *Nature medicine* **12,** 452–8 (2006).

49. Leor, J., Aboulafia-Etzion, S. & Dar, A. Bioengineered Cardiac Grafts A New Approach to Repair the Infarcted Myocardium? *Circulation* **102,** (2000).

50. Zimmermann, W., Didié, M. & Wasmeier, G. Cardiac grafting of engineered heart tissue in syngenic rats. *Circulation* **106,** (2002).

51. Zimmermann, W.-H. & Eschenhagen, T. Cardiac tissue engineering for replacement therapy. *Heart failure reviews* **8,** 259–69 (2003).

52. Shimizu, T. Fabrication of Pulsatile Cardiac Tissue Grafts Using a Novel 3-Dimensional Cell Sheet Manipulation Technique and Temperature-Responsive Cell Culture Surfaces. *Circulation Research* **90,** 40e–48 (2002).

53. Miyahara, Y. *et al.* Monolayered mesenchymal stem cells repair scarred myocardium after myocardial infarction. *Nature medicine* **12,** 459–65 (2006).

54. Christman, K. L. & Lee, R. J. Biomaterials for the treatment of myocardial infarction. *Journal of the American College of Cardiology* **48,** 907–13 (2006).

55. Birla, R. & Borschel, G. Myocardial engineering in vivo: formation and characterization of contractile, vascularized three-dimensional cardiac tissue. ... *engineering* **11,** (2005).

56. Hecker, L. & Birla, R. Engineering the heart piece by piece: state of the art in cardiac tissue engineering. *Regenerative Medicine* (2007). at <http://www.futuremedicine.com/doi/abs/10.2217/17460751.2.2.125>

57. Vunjak-Novakovic, G. *et al.* Challenges in cardiac tissue engineering. *Tissue engineering. Part B, Reviews* **16,** 169–87 (2010).

58. Colton, C. K. Implantable biohybrid artificial organs. *Cell Transplantation* **4,** 415–436 (1995).

59. Jr, W. H. *et al.* Increased vascularization and heterogeneity of vascular structures occurring in polyglycolide matrices containing aortic endothelial cells implanted in the rat. *Tissue ...* **3,** 149–160 (1997).

60. Mikos, a G. *et al.* Prevascularization of porous biodegradable polymers. *Biotechnology and bioengineering* **42,** 716–23 (1993).

61. Soker, S., Machado, M. & Atala, a. Systems for therapeutic angiogenesis in tissue engineering. *World journal of urology* **18,** 10–8 (2000).

62. Epstein, S. E., Fuchs, S., Zhou, Y. F., Baffour, R. & Kornowski, R. Therapeutic interventions for enhancing collateral development by administration of growth factors: basic principles, early results and potential hazards. *Cardiovascular research* **49,** 532–42 (2001).

63. Akhyari, P., Fedak, P., Weisel, R. & Lee, T. Mechanical stretch regimen enhances the formation of bioengineered autologous cardiac muscle grafts. *Circulation* (2002). doi:10.1161/01.cir.0000032893.55215.fc

64. Fink, C. *et al.* Chronic stretch of engineered heart tissue induces hypertrophy and functional improvement. *FASEB journal : official publication of the Federation of American Societies for Experimental Biology* **14,** 669–79 (2000).

65. Zimmermann, W.-H. Tissue Engineering of a Differentiated Cardiac Muscle Construct. *Circulation Research* **90,** 223–230 (2001).

66. Schaaf, S. *et al.* Human engineered heart tissue as a versatile tool in basic research and preclinical toxicology. *PloS one* **6,** e26397 (2011).

67. Tulloch, N. L. *et al.* Growth of engineered human myocardium with mechanical loading and vascular coculture. *Circulation research* **109,** 47–59 (2011).

68. Birla, R. K., Huang, Y. C. & Dennis, R. G. Development of a novel bioreactor for the mechanical loading of tissue-engineered heart muscle. *Tissue engineering* **13,** 2239–48 (2007).

69. Holden, J. K. *et al.* A novel device for the study of somatosensory information processing. *Journal of neuroscience methods* **204,** 215–20 (2012).

70. Halliday, D., Robert, R. & Walker, J. *Electromagnetism and Optics*. 764–816 (John Wiley & Sons, Inc., 2006).

71. Dennis, R. G. Measurement of Pulse Propagation in single permeabilized muscle fibers by optical diffraction. 137–138 (1996).

72. Cashion, A. T., Hubbard, D. K., Donnelly, K., Favorov, O. & Dennis, R. G. Instrumentation and Analysis for Sensor Applications of Low Frequency Dielectric Absorption. (2013).

73. Caplan, A. Adult mesenchymal stem cells for tissue engineering versus regenerative medicine. *Journal of cellular physiology* 341–347 (2007). doi:10.1002/JCP

74. Richardson, S. M. *et al.* Mesenchymal stem cells in regenerative medicine: opportunities and challenges for articular cartilage and intervertebral disc tissue engineering. *Journal of cellular physiology* **222,** 23–32 (2010).

75. Hilfiker, A., Kasper, C., Hass, R. & Haverich, A. Mesenchymal stem cells and progenitor cells in connective tissue engineering and regenerative medicine: is there a future for transplantation? *Langenbeck's archives of surgery / Deutsche Gesellschaft für Chirurgie* **396,** 489–97 (2011).

76. Huang, C.-H., Chen, M.-H., Young, T.-H., Jeng, J.-H. & Chen, Y.-J. Interactive effects of mechanical stretching and extracellular matrix proteins on initiating osteogenic differentiation of human mesenchymal stem cells. *Journal of cellular biochemistry* **108,** 1263–73 (2009).

77. Potier, E., Noailly, J. & Ito, K. Directing bone marrow-derived stromal cell function with mechanics. *Journal of biomechanics* **43,** 807–17 (2010).

78. Kearney, E. M., Farrell, E., Prendergast, P. J. & Campbell, V. a. Tensile strain as a regulator of mesenchymal stem cell osteogenesis. *Annals of biomedical engineering* **38,** 1767–79 (2010).

79. Ku, C.-H. *et al.* Collagen synthesis by mesenchymal stem cells and aortic valve interstitial cells in response to mechanical stretch. *Cardiovascular research* **71,** 548–56 (2006).

80. MacQueen, L., Sun, Y. & Simmons, C. a. Mesenchymal stem cell mechanobiology and emerging experimental platforms. *Journal of the Royal Society, Interface / the Royal Society* **10,** 20130179 (2013).

81. Vogel, V. & Sheetz, M. P. Cell fate regulation by coupling mechanical cycles to biochemical signaling pathways. *Current opinion in cell biology* **21,** 38–46 (2009).

82. Gaston, J., Quinchia Rios, B., Bartlett, R., Berchtold, C. & Thibeault, S. L. The response of vocal fold fibroblasts and mesenchymal stromal cells to vibration. *PloS one* **7,** e30965 (2012).

83. Huang, C., Charles, Y., Hagar, K. & Frost, L. Effects of Cyclic Compressive Loading on Chondrogenesis of Rabbit Bone- Marrow Derived Mesenchymal Stem Cells. *Stem ...* 313–323 (2004). at <http://onlinelibrary.wiley.com/doi/10.1634/stemcells.22-3-313/full>

84. Ingber, D. TENSEGRITY : THE ARCHITECTURAL BASIS OF CELLULAR MECHANOTRANSDUCTION. *Annual review of physiology* **59,** 575–99 (1997).

85. Cheung, W.-H., Chin, W.-C., Wei, F.-Y., Li, G. & Leung, K.-S. Applications of exogenous mesenchymal stem cells and low intensity pulsed ultrasound enhance fracture healing in rat model. *Ultrasound in medicine & biology* **39,** 117–25 (2013).

86. Cheung, W.-H., Chin, W.-C., Qin, L. & Leung, K.-S. Low intensity pulsed ultrasound enhances fracture healing in both ovariectomy-induced osteoporotic and age-matched normal bones. *Journal of orthopaedic research : official publication of the Orthopaedic Research Society* **30,** 129–36 (2012).

87. Bashardoust Tajali, S., Houghton, P., MacDermid, J. C. & Grewal, R. Effects of low-intensity pulsed ultrasound therapy on fracture healing: a systematic review and meta-analysis. *American journal of physical medicine & rehabilitation / Association of Academic Physiatrists* **91,** 349–67 (2012).

88. Rutten, S., Nolte, P. a, Korstjens, C. M., van Duin, M. a & Klein-Nulend, J. Low-intensity pulsed ultrasound increases bone volume, osteoid thickness and mineral apposition rate in the area of fracture healing in patients with a delayed union of the osteotomized fibula. *Bone* **43,** 348–54 (2008).

89. Lai, C.-H. *et al.* Effects of low-intensity pulsed ultrasound, dexamethasone/TGF-beta1 and/or BMP-2 on the transcriptional expression of genes in human mesenchymal stem cells: chondrogenic vs. osteogenic differentiation. *Ultrasound in medicine & biology* **36,** 1022–33 (2010).

90. Griffin, X. & Smith, N. Ultrasound and shockwave therapy for acute fractures in adults. *... Database Syst Rev* (2012). at <http://onlinelibrary.wiley.com/doi/10.1002/14651858.CD008579.pub2/pdf/standard>

91. Kaya, M. *et al.* Acoustic radiation force for vascular cell therapy: in vitro validation. *Ultrasound in medicine & biology* **38,** 1989–97 (2012).

92. Chung, S. L. *et al.* Fracture healing enhancement with low intensity pulsed ultrasound at a critical application angle. *Ultrasound in medicine & biology* **37,** 1120–33 (2011).

93. Alikhani, M. *et al.* Osteogenic effect of high-frequency acceleration on alveolar bone. *Journal of dental research* **91,** 413–9 (2012).

94. Wu, S.-H., Zhong, Z.-M. & Chen, J.-T. Low-magnitude high-frequency vibration inhibits RANKL-induced osteoclast differentiation of RAW264.7 cells. *International journal of medical sciences* **9,** 801–7 (2012).

95. Zhang, C. *et al.* Effects of mechanical vibration on proliferation and osteogenic differentiation of human periodontal ligament stem cells. *Archives of oral biology* **57,** 1395–407 (2012).

96. Shi, H.-F., Cheung, W.-H., Qin, L., Leung, A. H.-C. & Leung, K.-S. Low-magnitude high-frequency vibration treatment augments fracture healing in ovariectomy-induced osteoporotic bone. *Bone* **46,** 1299–305 (2010).

97. Tirkkonen, L. *et al.* The effects of vibration loading on adipose stem cell number, viability and differentiation towards bone-forming cells. *Journal of the Royal Society, Interface / the Royal Society* **8,** 1736–47 (2011).

98. Kim, I. S., Song, Y. M., Lee, B. & Hwang, S. J. Human mesenchymal stromal cells are mechanosensitive to vibration stimuli. *Journal of dental research* **91,** 1135–40 (2012).

99. Chow, D. H.-K., Leung, K.-S., Qin, L., Leung, A. H.-C. & Cheung, W.-H. Low-magnitude high-frequency vibration (LMHFV) enhances bone remodeling in osteoporotic rat femoral fracture healing. *Journal of orthopaedic research : official publication of the Orthopaedic Research Society* **29,** 746–52 (2011).

100. Cheung, W.-H. *et al.* Stimulated angiogenesis for fracture healing augmented by low-magnitude, high-frequency vibration in a rat model-evaluation of pulsed-wave doppler, 3-D power Doppler ultrasonography and micro-CT microangiography. *Ultrasound in medicine & biology* **38,** 2120–9 (2012).

101. Tsuang, Y.-H., Lin, Y.-S., Chen, L.-T., Cheng, C.-K. & Sun, J.-S. Effect of dynamic compression on in vitro chondrocyte metabolism. *The International journal of artificial organs* **31,** 439–49 (2008).

102. Bian, L., Zhai, D. Y., Zhang, E. C., Mauck, R. L. & Burdick, J. A. Matrix Synthesis and Distribution and Suppresses. **18,** (2012).

103. Wolchok, J. C., Brokopp, C., Underwood, C. J. & Tresco, P. a. The effect of bioreactor induced vibrational stimulation on extracellular matrix production from human derived fibroblasts. *Biomaterials* **30,** 327–35 (2009).

104. Haugh, M. G. *et al.* Temporal and spatial changes in cartilage-matrix-specific gene expression in mesenchymal stem cells in response to dynamic compression. *Tissue engineering. Part A* **17,** 3085–93 (2011).

105. Moraes, C., Chen, J.-H., Sun, Y. & Simmons, C. a. Microfabricated arrays for high-throughput screening of cellular response to cyclic substrate deformation. *Lab on a chip* **10,** 227–34 (2010).

106. Kamotani, Y. *et al.* Individually programmable cell stretching microwell arrays actuated by a Braille display. *Biomaterials* **29,** 2646–55 (2008).

107. Moraes, C., Wang, G., Sun, Y. & Simmons, C. a. A microfabricated platform for high-throughput unconfined compression of micropatterned biomaterial arrays. *Biomaterials* **31,** 577–84 (2010).

108. Brown, T. D. Techniques for mechanical stimulation of cells in vitro: a review. *Journal of biomechanics* **33,** 3–14 (2000).

109. MacQueen, L., Chebotarev, O., Simmons, C. a & Sun, Y. Miniaturized platform with on-chip strain sensors for compression testing of arrayed materials. *Lab on a chip* **12,** 4178–84 (2012).

110. Angele, P. *et al.* Cyclic hydrostatic pressure enhances the chondrogenic phenotype of human mesenchymal progenitor cells differentiated in vitro. *Journal of orthopaedic research : official publication of the Orthopaedic Research Society* **21,** 451–7 (2003).

111. Puetzer, J., Williams, J., Gillies, A., Bernacki, S. & Loboa, E. The effects of cyclic hydrostatic pressure on chondrogenesis and viability of human adipose- and bone marrow-derived mesenchymal stem cells in three-dimensional agarose constructs. *Tissue engineering. Part A* **19,** 299–306 (2013).

112. Uzer, G., Pongkitwitoon, S., Ete Chan, M. & Judex, S. Vibration induced osteogenic commitment of mesenchymal stem cells is enhanced by cytoskeletal remodeling but not fluid shear. *Journal of biomechanics* 1–7 (2013). doi:10.1016/j.jbiomech.2013.06.008

113. Cashion, A. T., Salazar, B., Birla, R. & Dennis, R. G. A Cyclic Strain Bioreactor for Mechanical Enhancement and Assessment of Self-Organized Myocardial Patches. (2013).

114. Inman, D. J. *Engineering Vibration*. (Pearson, 2008).

115. Caballero, M., Reed, C. R., Madan, G. & van Aalst, J. a. Osteoinduction in umbilical cord- and palate periosteum-derived mesenchymal stem cells. *Annals of plastic surgery* **64,** 605–9 (2010).

116. Dahl, J. P. *et al.* Analysis of human auricular cartilage to guide tissue-engineered nanofiber-based chondrogenesis: implications for microtia reconstruction. *Otolaryngology--head and neck surgery : official journal of American Academy of Otolaryngology-Head and Neck Surgery* **145,** 915–23 (2011).

117. Smith, J. M. *et al.* Kidney, pancreas and liver allocation and distribution in the United States. *American journal of transplantation : official journal of the American Society of Transplantation and the American Society of Transplant Surgeons* **12,** 3191–212 (2012).

118. Van de Kerkhove, M. P., Hoekstra, R., Chamuleau, R. a. F. M. & van Gulik, T. M. Clinical Application of Bioartificial Liver Support Systems. *Annals of Surgery* **240,** 216–230 (2004).

119. Baptista, P. M. *et al.* The use of whole organ decellularization for the generation of a vascularized liver organoid. *Hepatology (Baltimore, Md.)* **53,** 604–17 (2011).

120. Daley, W. P., Peters, S. B. & Larsen, M. Extracellular matrix dynamics in development and regenerative medicine. *Journal of cell science* **121,** 255–64 (2008).

121. Berzat, A. & Hall, A. Cellular responses to extracellular guidance cues. *The EMBO journal* **29,** 2734–45 (2010).

122. Uygun, B. E. *et al.* Organ reengineering through development of a transplantable recellularized liver graft using decellularized liver matrix. *Nature medicine* **16,** 814–20 (2010).