Low Cost Global Health Simulator Report

Project Number: micro13ZM306
Team: Medical Robotics and Devices Laboratory
Team Members: Robin Walz, Zachary Meier, Michael Winek, Tim Kowalewski

The low cost global health simulator project was started as an undergraduate research project under the leadership of Zach Meier and Robin Waltz at the University of Minnesota. The aim of the project was to develop an ultra-low cost health simulator (<=$50) that could be used in training physicians on simple procedures in developing countries. There are many procedural simulators on the market today; however, almost all of these products are cost prohibitive for the markets we wish to assist. The constraints of the project were that the design be easy to assemble, cheap, and created from materials readily available in a developing country. With the above constraints in mind and a general aim at developing a simulator, there were three major pieces to this project. The first was to create an artificial tissue that could be used to mimic human skin tissue in its mechanical properties. The second was to create tracking system for surgical tools. Finally, a full assembly of the two smaller pieces with a specific function in mind was to be developed.

Importance

Mortality rates are unnecessarily high in developing countries due to lack of medical training or available procedures. For example, maternal deaths range from 277,000 to 817,000 per year, of which 99% happen in developing countries (United Nations: Department of Economic and Social Affairs, Statistics Division "Progress towards the Millennium Development Goals, 1990-2005" Retrieved March 4, 2012). Many of these deaths could be avoided if the physicians and nurses had a way to train and practice life-saving medical procedures like Cesarean sections or tracheotomies without risking harm to patients. A way to do this would be to have surgical simulators that they could practice on. While the medical profession in the developed world has shifted to simulation-based training (Kunkler, Kevin. "The Role of Medical Simulation: An Overview." *The International Journal of Medical Robotics and Computer Assisted Surgery* 2.3 (2006): 203-10. Print.), current simulators range from $2,500 to $300,000, which is not an option for most hospitals, or care providers in developing countries.

Project Scope

We have attempted to design an open-source ultra-low-cost medical procedure simulator platform that would be made of materials available in developing countries so that they could be locally made yet provide a means to accurately train and assess skill acquisition in medical procedures. We have identified two enabling technologies that may satisfy these requirements: bio-plastics to simulate tissue and two-dimensional surface potentiometers to electronically track surgical tools on tissue. The goal of this research is to assess the feasibility of such simulators by evaluating these technologies and constructing a prototype simulator that employs them.

Artificial Tissue Development

It was decided that the artificial tissue was to be some form of bioplastic, which is both easy to make and is made from very abundant materials. Dozens of recipes were attempted ranging from a simple gelatin bioplastic to a tapioca starch recipe. Initial tests were done with bioplastics created by hand and cast into petri dishes for evaluation as seen in figure 1.

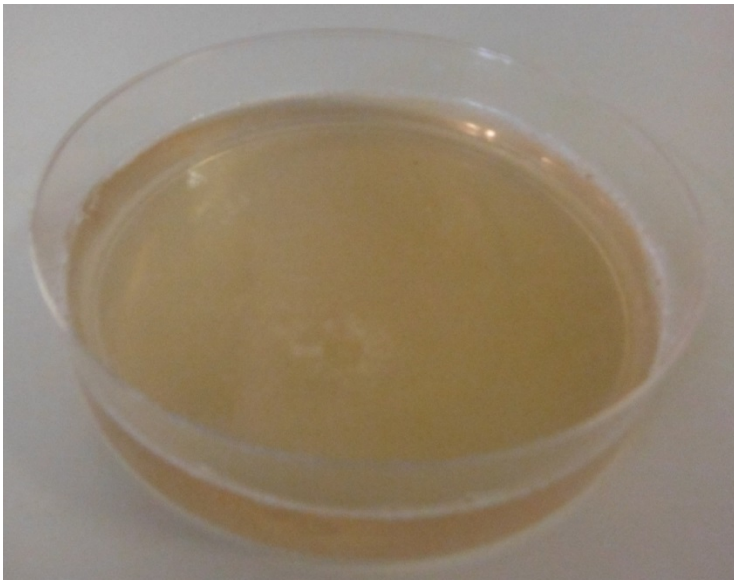


Figure 1: Bioplastic sample cast in petri dish

From the findings, we found the gelatin bioplastic was the quickest to dry and the most life-like right after it was made; however, it became rigid to the point of uselessness after a day or more of air exposure. This was deemed a potential for trainings that were to be completed within a short time frame after the preparation.

For a storable artificial tissue, a combination of the gelatin and tapioca starch was used to create a more durable bioplastic that lasts for weeks without significant change in the material's properties. Both recipes are included in Appendix B.

A useful feature of these bioplastics is that in addition to creating these bioplastics from scratch, they can be melted down and recast once the water weight that they lost when drying is replaced by fresh water. This further reduces the cost of the simulator.

As the impedance of the bioplastic is an important design consideration for this project, special care has been taken to characterize the electrical properties of the artificial tissues. Attempts have been made to increase the conductivity of the plastic through the addition of graphite powder. While it has been advantageous in helping decrease the impedance, it has not been the missing component we hoped it would be.

To create an appropriate platform to design a tracking system on a variety of mold shapes were attempted. These designs evolved into a rectangle well with space to mold an electrode into each corner to provide landmarks to measure resistance from. The electrodes we chose were American pennies as they were readily available and provided a large, conductive surface. Figure 2 shows the mold and figures 3 & 4 show samples created from the mold.

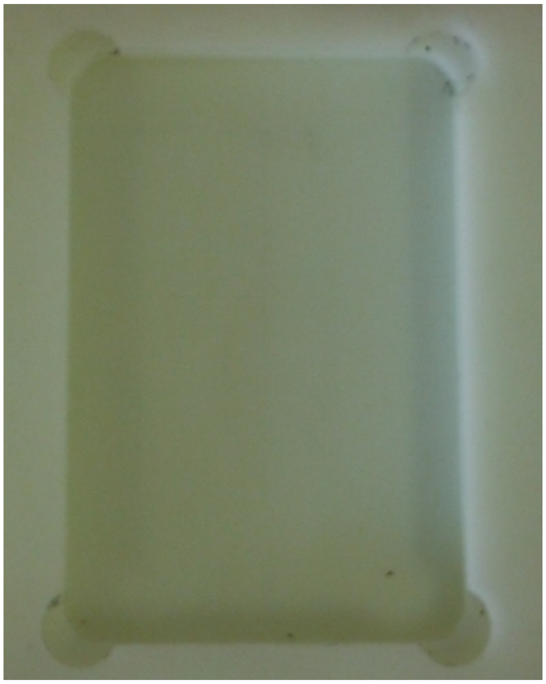


Figure 2: Final mold design for 2D tracking system

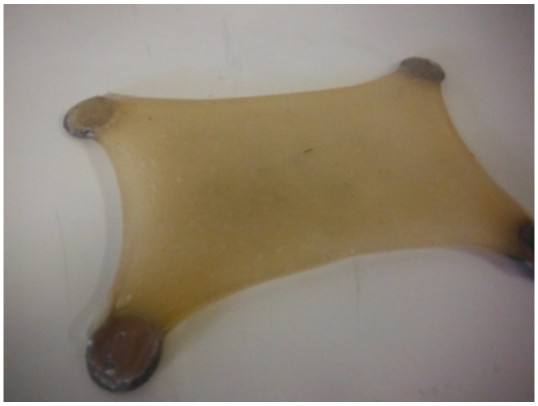


Figure 3: Fautex cast from mold

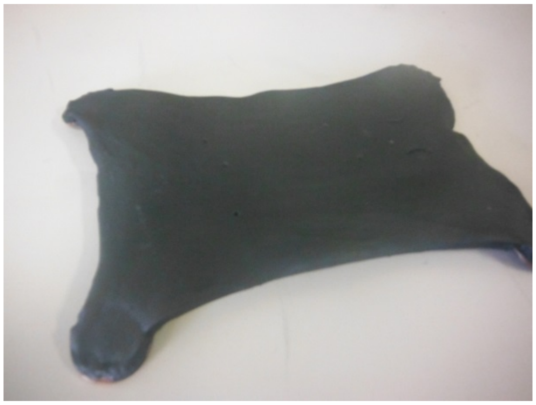


Figure 4: Graphite infused Fautex cast from mold

The development of the bioplastic was much more time consuming than anticipated, and the researchers were unable to quantify the mechanical properties through biaxial tensile testing as originally desired; however, the mechanical properties were qualitatively judged based on the overall feel and elasticity one would expect with skin. This was determined to be sufficient at this stage of development.

2D Tracking

The development of the surgical tool tracker is currently a work in progress, and is unfortunately not finalized in time for the MicroMedic competition submission deadline due to complications with the bioplastics; however, great progress has been made to create a tracking tool.

The method for tracking is to use analog voltages on an Arduino to read the voltage between the probing location and the 5V input at each corner, essentially creating a ratio between four voltages at different locations of the bioplastic. The bioplastic provides variable resistance depending on where the ground is probed. The wiring for this approach can be seen in figure 5.

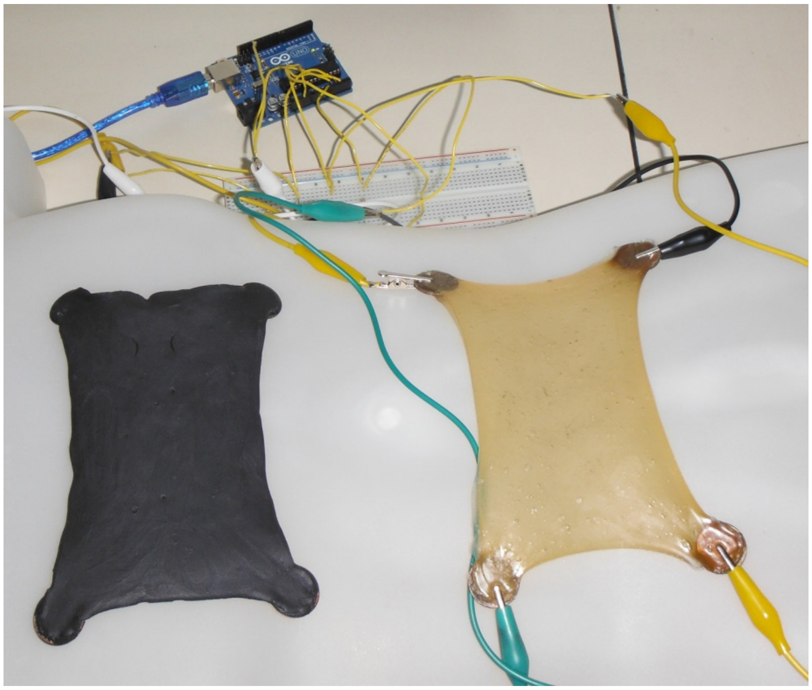
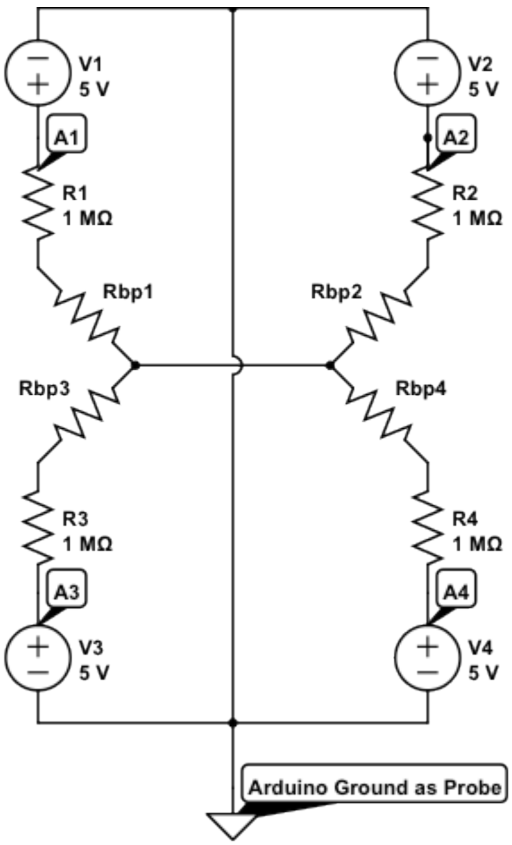


Figure 5: Circuitry for tracking

Approximations for the location were accounted for using trigonometry initial calibration. A schematic of the electrical setup is provided in the Appendix A, and the code written in Matlab interfacing with the Arduino has been included as well in Appendix C. Results from this part of the project have been successful on very small scales; however, due to the high impedance of the bioplastic and lack of homogeneity the results have not been as successful with creating the desired simulator. Significant work still needs to be done to create a more homogeneous bioplastic and better create an Arduino/Matlab program to track a scalpel's position. This requires more than simple trigonometry for determining probe location, and potentially a more effective wiring schematic. The end goal is to have the code written within the Arduino environment; however, utilizing Matlab is easier for debugging purposes.

Although there is still work to be done in creating the finalized low cost simulator, significant headway has been made since the beginning of this project. The final prototype is to be developed as a cesarean section simulator utilizing the bioplastic as the first layer of skin tissue and the Arduino as the processor behind tracking the incision made. It is believed that there is a lot of potential with this project and that it could be very useful for low cost surgical training.

APPENDIX A: Circuit Diagram



2-D Potentiometer Schematic
V1-4: Output voltage from Arduino
A1-4: Analog input of Arduino
Rbp: resistance of bioplastic between probe and voltage source

Appendix B: Bioplastic Recipes

To create the bioplastic bring liquid ingredients to heat in a pan and slowly add the dry ingredients while mixing.

Gelatin Bioplastic Recipe

60mL water
12g gelatin
3g glycerol

Gelatin-Starch (Fautex) Recipe

250mL water
7g gelatin
22.8g tapioca starch
60g glycerin
40g vinegar

Appendix C: Code

```
%-- connect to the board
 clc; close all; clear all;
 a = arduino('COM3')
 o_voltage1= 2;
 o_voltage2= 3;
 o_voltage3= 4;
 o_voltage4= 5;
 i_voltage1 = 0;
 i_voltage2 = 1;
 i_voltage3 = 2;
 i_voltage4 = 3;
 a.pinMode(o_voltage1, 'output');
 a.pinMode(o_voltage2, 'output');
 a.pinMode(o_voltage3, 'output');
 a.pinMode(o_voltage4, 'output');

 calR1_2=0;
 calR1_4=0;
 calR3_4=0;
 calR3_2=0;

 %Calibration Values of resistance
 input('Press Enter When Set For Calibration of R1-2\n4**3\n*
*\n1**2');
 a.digitalWrite(o_voltage1, 1);
 a.digitalWrite(o_voltage2, 1);
 a.digitalWrite(o_voltage3, 1);
 a.digitalWrite(o_voltage4, 1);
 v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage1)*(5.0 / 1024.0);
 end
 calR1_2= ((v/j)*10^3)/(5 - (v/j));

 input('Press Enter When Set For Calibration of R2-1\n4**3\n*
*\n1**2');
 a.digitalWrite(o_voltage2, 1);
 pause(1);
  v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage2)*(5.0 / 1024.0);
 end
 calR1_2= (((v/j)*10^3)/(5 - (v/j))+calR1_2)/2;

 input('Press Enter When Set For Calibration of R1-4\n4**3\n*
*\n1**2');
  a.digitalWrite(o_voltage1, 1);
 pause(1);
  v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage1)*(5.0 / 1024.0);
 end
 calR1_4= ((v/j)*10^3)/(5 - (v/j));
 input('Press Enter When Set For Calibration of R4-1\n4**3\n*
*\n1**2');
 a.digitalWrite(o_voltage4, 1);
 pause(1);
  v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage4)*(5.0 / 1024.0);
 end
 calR1_4= (((v/j)*10^3)/(5 - (v/j))+calR1_4)/2;

input('Press Enter When Set For Calibration of R4-3\n4**3\n*
*\n1**2');
 a.digitalWrite(o_voltage4, 1);
 pause(1);
  v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage4)*(5.0 / 1024.0);
 end
 calR3_4= ((v/j)*10^3)/(5 - (v/j));

input('Press Enter When Set For Calibration of R3-4\n4**3\n*
*\n1**2');
 a.digitalWrite(o_voltage3, 1);
 pause(1);
  v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage3)*(5.0 / 1024.0);
 end
 calR3_4= (((v/j)*10^3)/(5 - (v/j))+calR3_4)/2;

 input('Press Enter When Set For Calibration of R3-2\n4**3\n*
*\n1**2');
  a.digitalWrite(o_voltage3, 1);
 pause(1);
  v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage3)*(5.0 / 1024.0);
 end
 calR3_2= ((v/j)*10^3)/(5 - (v/j));

input('Press Enter When Set For Calibration of R2-3\n4**3\n*
*\n1**2');
 a.digitalWrite(o_voltage2, 1);
 pause(1);
  v=0;
 for j=1:100
     v= v + a.analogRead(i_voltage2)*(5.0 / 1024.0);
 end
 calR3_2= (((v/j)*10^3)/(5 - (v/j))+calR3_2)/2;

 %Determining the position
 running = true;
 while running == true
 input('Press Enter When Ready To Test Location');
 a.digitalWrite(o_voltage1, 1);
 a.digitalWrite(o_voltage2, 1);
```

```
a.digitalWrite(o_voltage3, 1);
a.digitalWrite(o_voltage4, 1);
pause(1);
v1=0;
v2=0;
v3=0;
v4=0;
for i=1:100
    v1= v1 + a.analogRead(i_voltage1)*(5.0 / 1024.0);
    v2= v2 + a.analogRead(i_voltage2)*(5.0 / 1024.0);
    v3= v3 + a.analogRead(i_voltage3)*(5.0 / 1024.0);
    v4= v4 + a.analogRead(i_voltage4)*(5.0 / 1024.0);

end
a.digitalWrite(o_voltage1, 0);
a.digitalWrite(o_voltage2, 0);
a.digitalWrite(o_voltage3, 0);
a.digitalWrite(o_voltage4, 0);

calR1_2
calR1_4
calR3_4
calR3_2

R1= ((v1/i)*10^3)/(5 - (v1/i))
R2= ((v2/i)*10^3)/(5 - (v2/i))
R3= ((v3/i)*10^3)/(5 - (v3/i))
R4= ((v4/i)*10^3)/(5 - (v4/i))

x1= R1*(R1^2+calR1_2^2-R2^2)/(2*R1*calR1_2);
y1= R1*sin(acos((R1^2+calR1_2^2-R2^2)/(2*R1*calR1_2)));
y2= R1*(R1^2+calR1_4^2-R4^2)/(2*R1*calR1_4);
x2= R1*sin(acos((R1^2+calR1_4^2-R4^2)/(2*R1*calR1_4)));
x3= R4*(R4^2+calR3_4^2-R2^2)/(2*R4*calR3_4);
y3= calR1_4-R4*sin(acos((R4^2+calR3_4^2-R2^2)/(2*R4*calR3_4)));
x4= calR3_4-R3*sin(acos((R3^2+calR3_2^2-R2^2)/(2*R3*calR3_2)));
y4= calR3_2-R3*(R3^2+calR3_2^2-R2^2)/(2*R3*calR3_2);

x= (x1+x2+x3+x4)/4;
y= (y1+y2+y3+y4)/4;

figure(1);
axis([0 1 0 1]);
hold on;
plot(x/calR1_2,y/calR1_4,'b+');
answer = input('Another point enter 1?\n');

if answer == 1
    running=true;
else
    running =false;
end
end
hold off;
delete(instrfind({'Port'},{'COM3'}));
```