Final Project: Sensor Verification

Timothy J Boger, Amrita Sahu, Firdous Saleheen

12/1/11

Abstract

A temperature and a force sensor system were designed, calibrated and tested to determine the feasibility of using these sensors for the tactile imaging system. A temperature sensing system capable of detecting the applied temperature with accuracy was designed, tested and implemented with an Arduino board for data acquisition and verification. Data was compared to digital thermometer readings. A force sensing system capable of detecting the applied force and quantifying it was designed, calibrated, tested and implemented. A circuit for calculation of force applied on force sensing resistor has been designed, simulated and implemented on breadboard. The circuit setup is then integrated with Arduino board for data acquisition. Manufacturer-provided voltage vs force curve and Matlab curve fitting tool was used to find the relation between output voltage of the circuit and applied force. A new curve fitting was created through test data and verified by through comparison to force gauge readings.

Introduction

The purpose of these laboratory tests were to verify the accuracy of a temperature and force sensor and determined whether it would be applicable for the tactile imaging system. Both the temperature and force sensor data need to be compared to a standard measuring tool like a thermometer or force gauge. Manufacturer-provided curves can also be used for verification or a new curve can be generated from test data to calibrate the sensor. In this project an analog temperature, LM35 Precision Centigrade Temperature Sensor, was used. An analog temperature sensor is a chip that tells us what the ambient temperature and as temperature increases, the voltage across a diode increases at a known rate. The temperature sensor is shown in Figure 1.

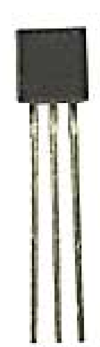

Figure 1. Temperature Sensor

A force sensor measures applied force on some arbitrary target. For this purpose, a force sensing resistor is used. A force-sensing resistor (FSR) is a material whose resistance changes when a force or pressure is applied (Force-Sensing Resistor, 2011). It is of small size, low cost and easy to use. The FSR being used is shown in Figure 2.

Figure 2. Force Sensor

The final task is to analyze whether this sensor is applicable for use on the tactile imaging system.

Method

Temperature Sensor : To test the temperature sensor it, was connected to the Arduino microcontroller and integrated with Matlab. The setup is shown in Figure 3. where the temperate sensor is being tested using a regulated hot plate.

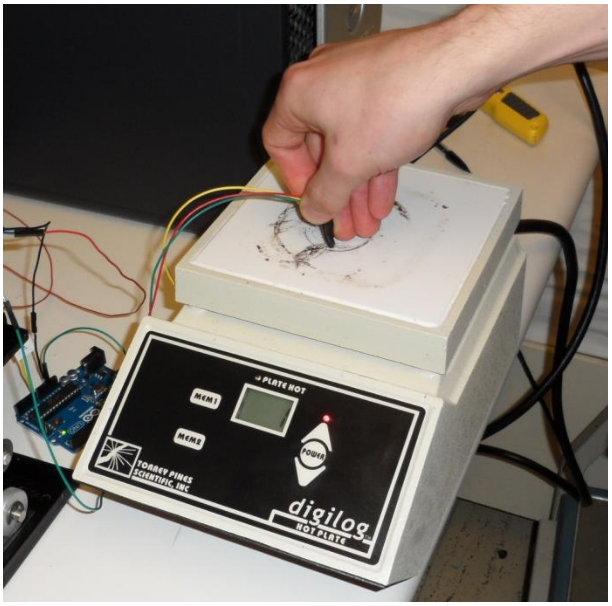

Figure 3. Test Setup for Temperature Sensor

To connect the temperature sensor to the Arduino board, the diagram in Figure 4. was used.

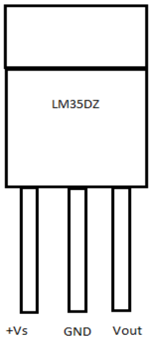

Figure 4. Temperature Sensor Pinout

The +Vs was connect to the 5 volts of the board, the GND pin to ground, and Vout to the analog pin of the Arduino board. The voltage output of the sensor can be plotted in Matlab and converted to temperature. To verify the temperature of each test, a digital thermometer was used.

Force Sensor :

Hardware and Software:

1. Force Gauge Mcmesin (0-50N)
2. Force sensor Interlink model 402
3. Arduino microcontroller
4. Stand
5. Scissor lift plate
6. Natural gum rubber ball
7. Matlab 2010b
8. Power supply +5V, +15V, -15V

Steps:

1. The circuit was built as Figure 5. In Figure 5, the circuit for force measurement is shown where a pull-down resistor R= 10k ohms, Vin = 5V is used. R_{FSR} indicates the force sensing resistor. A voltage op-amp (uA741) follower is used to match the impedance requirement of the upstream measuring circuit. V_{out} is the output voltage measured at pin 6 of the op-amp. With this configuration, the output voltage increases with increasing force.

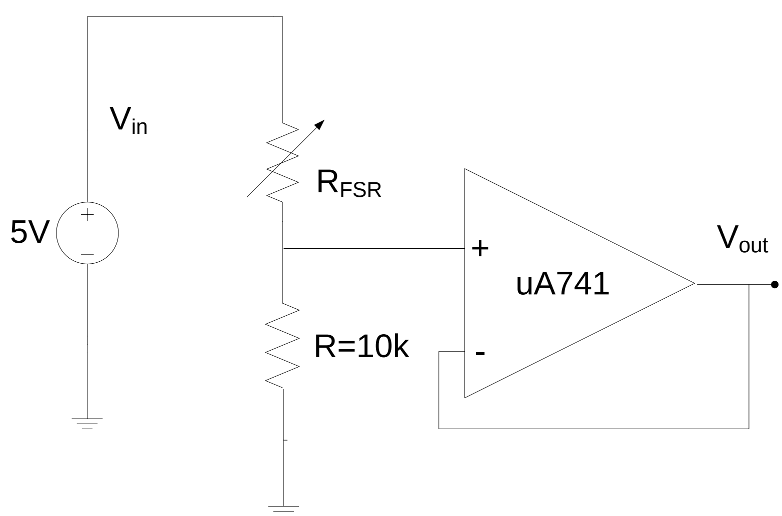

Figure 5: Voltage Divider Circuit for Force Sensing

For a simple force-to-voltage conversion, the FSR device is tied to a measuring resistor in a voltage divider configuration. The output is described by the equation:

$$V_{out} = \frac{R}{R + R_{FSR}} \times V_{in}$$

2. The force sensor needs to be calibrated. For that the setup was arranged as Figure 6. The force sensor with measuring was connected to the Arduino microcontroller and integrated with Matlab. The setup is shown in Figure 6 where the force sensor is being tested using a scissor lift plate and stand that is holding a force gauge and the force sensor attached to the tip of the gauge.

Figure 6. Force Sensor Test Setup

3. The natural gum rubber ball was pressed with the force sensor attached at the front of the force gauge tip.
4. 100 datapoints for output voltage was recorded and averaged on Matlab which is sensor voltage (V) in Table 1
5. At the same time, the reading from the digital force gauge is taken of the sensor was recorded which is force gauge force reading in ‘g’ in Table 1.

6. The calibration data to be curve fitted is shown in Table 1. This data can be used to create a new curve fitting line for accurate force sensor measurements. The matlab curve fitting codes are given in Appendix A.

Trial	Force Gauge (g)	Sensor Voltage (V)
1	33	1.0254
2	40	1.3379
3	64	1.4111
4	77	1.5283
5	82	1.8457
6	115	2.3193
7	164	2.6172
8	248	3.0225
9	537	3.7207
10	604	3.9063
11	372	3.6035
12	329	3.4619
13	266	3.1641
14	102	2.1338
15	698	3.9600
16	774	4.0381
17	839	4.1064
18	898	4.1895
19	993	4.2773
20	1028	4.3018
21	1186	4.3555
22	1268	4.3750
23	1102	4.3262
24	1285	4.3262
25	1474	4.3848

Table 1. Force Test Results

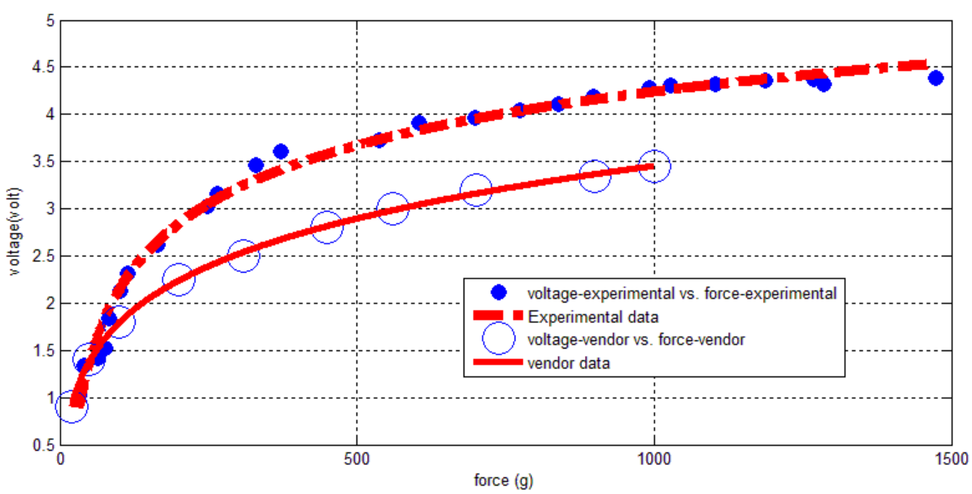

Figure 7. Experimental and Vendor Curve Fitting

Using Table 1 a relationship between applied force and output voltage of the sensor has been found. Note that the force is in 'g' force unit and voltage is in volt unit. The solid line in Figure shows the vendor fitted curve and the dotted line shows the experimental fitted curve. Different test conditions and equipment were used for calibration which resulted into different curve from experiment than that of vendor.

The relationship found from fitted curve is as follows:

$$v = af^{b} + c\,,$$

where v is sensor output voltage and f is force and the coefficients (with 95% confidence bounds) are as follows:

a = -14.82, b = -0.1281 and c = 10.36

Goodness of fit: Coefficient of determination R-sqaure means provides a measure of how well future outcomes are likely to be predicted by the model ranging from 0 to 1. RMSE means the root mean square error used measure of the differences between values predicted by a model or an estimator and the values actually observed from the thing being modeled or estimated. In this fitting, R-square: 0.9885 which close to 1 and RMSE: 0.1281 which is small.

7. After calibration was done, the same test was performed. 13 trials were done. Each instance average force value was recorded from the force gauge. Also at the same time, average value of force was recorded from force sensor. The relationship used was derived as follows:

$$v = af^{b} + c,$$

$$f^{b} = \frac{v-c}{a},$$

$$\log f^{b} = \log\frac{v-c}{a},$$

$$\log f = \frac{1}{b}\log\frac{v-c}{a},$$

$$f = \exp[\frac{1}{b}\log\frac{v-c}{a}]$$

a, b, c values are same as above. In case of force sensor, 100 data points were used for calculating average. These are recorded in Table 3.

Results

Temperature Sensor : The temperature sensor was tested using Body, the hot plate, room, and an avocado's to test the sensor. The results are shown in Table 2.

Material whose temperature is being tested	Temperature detected by thermometer (in degree centigrade)	Temperature detected by sensor (in degree centigrade)
Room temp	25.4	24.6
Body temperature	34.8	34.6
Hot plate#1	34	33.8
Hot plate#2	38.5	38.2
Hot plate#3	44	44.5
Hot plate#4	50	48.5
Cold Avocado	20	20.3

Table 2. Temperature Test Results

Force Sensor:

The experiment objective was to calibration of force sensor with force gauge and verify the calibration by collecting force values randomly. The experiment results for verification of force sensor are given in Table 3. Both the forces are in 'g' unit. Also the absolute difference and percentage error between two readings are shown.

Trial	Average force from force gauge, G (g)	Average force from force sensor, S (g)	Absolute difference	%error, (S-G)*100/G
1	86	84.7139	1.2861	1.495465116
2	139	158.85	19.85	14.28057554
3	151.5	197.8847	46.3847	30.6169637
4	179	250.6723	71.6723	40.04039106
5	252	335.8315	83.8315	33.26646825
6	277.4	420.7274	143.3274	51.66813266
7	387	575.3128	188.3128	48.65963824
8	458	620.6383	162.6383	35.51054585
9	598.5714286	857.3837	258.8122714	43.23832697
10	724.5	852	127.5	17.59834369
11	854	1113	259	30.32786885
12	1144.2	1288	143.8	12.56773291
13	1173.5	1305	131.5	11.20579463

Table 3. Force Sensor Test Results

In figure 9, the x-axis represents number of trials and y-axis represents average force in 'g' unit. It shows that force sensor reading was higher than the force gauge at each instance. Below 300g force sensor closely follows the force gauge value with maximum 143g absolute difference. After that difference becomes higher, although the percentage error shows not a great variation.

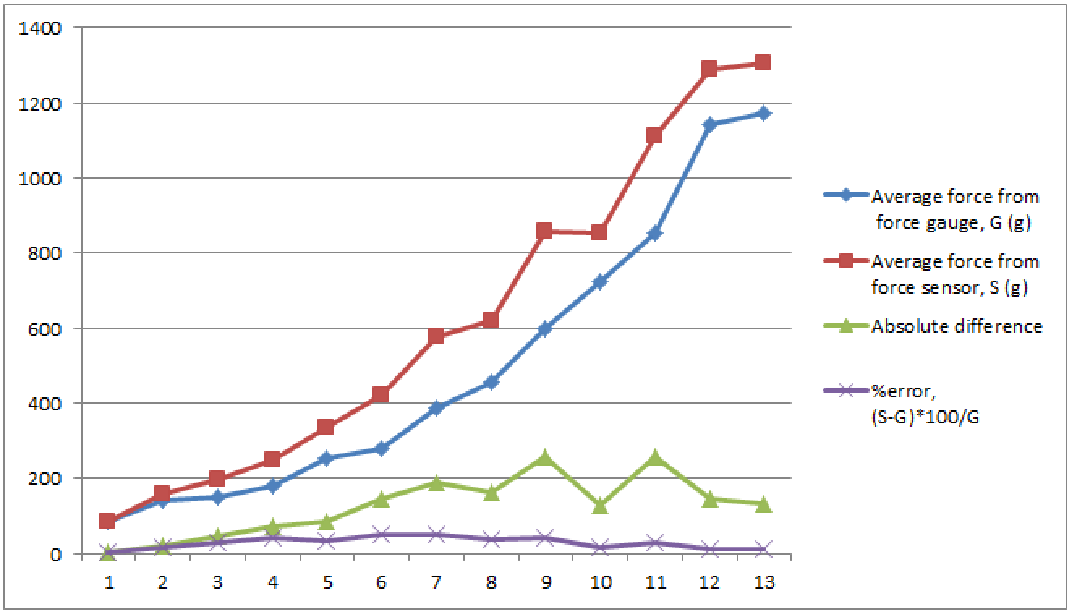

Figure 9. Force Gauge vs Force Sensor data with difference and percentage error

Discussion

Temperature Sensor : The results of the temperature sensor and thermometer were compared in Figure 8

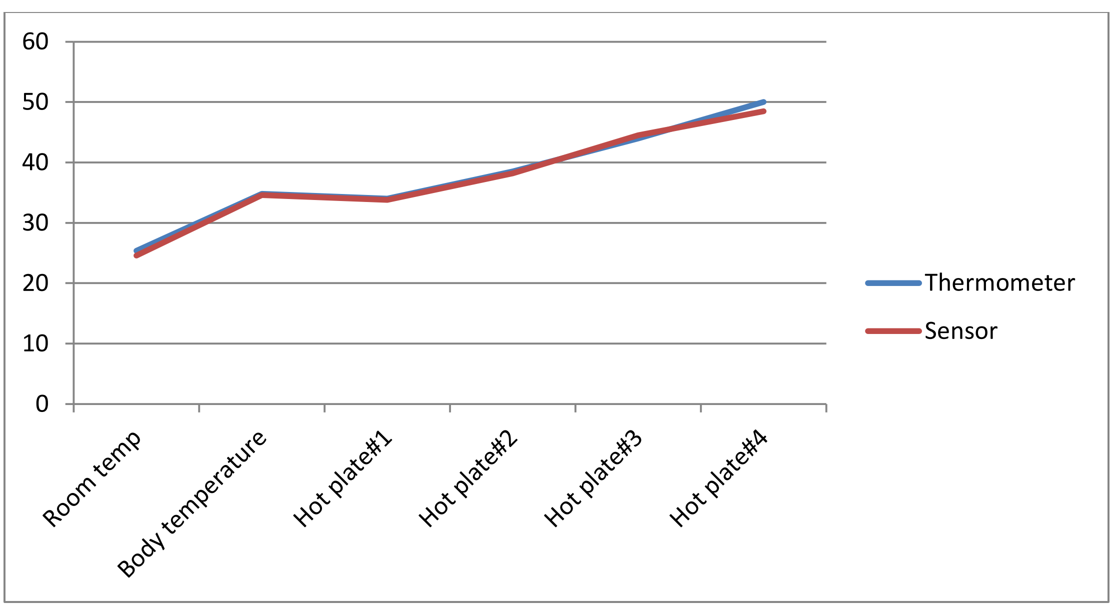

Figure 8. Temperature Test Results Comparison

From the graph, the temperature sensor provides an accurate measurement of temperature. Based on the results for body temperature, this sensor could be used in the tactile imaging system.

Force Sensor:

In this experiment, a force sensor was calibrated with a digital force gauge. The calibration was verified taking force sensor values randomly and comparing with force gauge value. The minimum difference found was 1.2861g and the maximum difference is 259g. The percentage error varies from ~1.49 to 50%. The reason for variation may be the contact area for two force meter was different. Force sensor was in direct contact with the surface of the sample whereas the force gauge indent tip was 30mm away from the sample. It might happen that the attaching part between force gauge tip and force sensor mounting area has effect on force values. The force sensor being to the nearest of the sample should register higher force reading than force gauge. However, the methodology taken for this is subject to human error and fitting error. The alternative way is to use fixed weight and calibrate the force sensor. The results show that this sensor is not very accurate. It suits best for the application that needs to detect force change. For the tactile imaging system, this sensor could be a better solution than the force gauge.

Acknowledgement

This project is sponsored by Dr. Chang-Hee Won. The author would also like to thank Dr. Won for his valuable advice and Matt Bosack for suggestions.

Appendices

Appendix A – Matlab Code for Curve Fitting

```
%forcevoltageexp1.m
%author Firdous Saleheen
%date 12/5/2011
close all

%% Fit: 'forcefittrial'.
load force_experimental.mat
load voltage_experimental.mat
%%
[xData, yData] = prepareCurveData( force_experimental, voltage_experimental
);

% Set up fittype and options.
```

```
ft = fittype( 'power2' );
opts = fitoptions( ft );
opts.Display = 'Off';
opts.Lower = [-Inf -Inf -Inf];
opts.StartPoint = [0.147531206404833 0.519627809074792 -0.36515838706883];
opts.Upper = [Inf Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );
%%
% Plot fit with data.
figure( 'Name', 'Force-Voltage Fitting Experiment vs Vendor' );
h = plot( fitresult, xData, yData );
legend( h, 'voltage-experimental vs. force-experimental', 'Experimental
data', 'Location', 'NorthEast' );
%Label axes



hold on
%%
load f_fsr_10k.mat
load v_fsr_10k.mat
[xData1, yData1] = prepareCurveData( f_fsr_10k, v_fsr_10k );

% Set up fittype and options.
ft = fittype( 'power2' );
opts = fitoptions( ft );
opts.Display = 'Off';
opts.Lower = [-Inf -Inf -Inf];
opts.StartPoint = [0.275556015376618 0.383884327551394 -0.0740229780672868];
opts.Upper = [Inf Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData1, yData1, ft, opts );

% Plot data
%figure( 'Name', 'Force-Voltage Fitting Experiment vs Vendor' );
h1 = plot( fitresult, xData1, yData1,'o');
legend( h1, 'voltage-vendor vs. force-vendor', 'vendor data', 'Location',
'SouthEast' );
grid on
xlabel( 'force (g)' );
ylabel( 'voltage(volt)' );
```