International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 01, January 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

FINGER MOVEMENT DETECTION USING INFRARED SIGNALS

Dr. Jillella Venkateswara Rao.

Professor, Department of ECE, Vignan Institute of Technology and Science, Hyderabad, (India)

ABSTRACT

It has been created an armband that used infrared (IR) diodes to capture finger movements from the extensor digitorum communis muscle. From here, it has been used a machine-learning model to classify these signals, in order to create a real-time predictor. The idea was to shine infrared light on the user's upper forearm. As the user moved different fingers, muscles on the upper forearm would move, causing the IR light to diffract in differing directions. This reaction was detected using photodiodes, digitized through an ADC, and then processed via machine learning in order to predict exactly which finger the user was moving. Using two channels, it has been able to achieve almost perfect prediction accuracy when distinguishing between three fingers in real time. While a third channel was built, due to time constraints it has been not able to acquire three channels of data. It would be able to accurately predict four different finger movements, and possibly the thumb.

Keywords: Analog to Digital Converter, Armbands, Data Acquisition, Infrared, Machine Learning, Real-time prediction.

I. INTRODUCTION

Figure 1 shows a schematic of the circuit used. The circuit was powered by the 5V and GND rails of an Atmega1284p microcontroller. An infrared light was biased with a 75Ω resistor and shined on the user's arm. The reflection of the infrared light was then picked up by a LTR4206 photodiode. The output of this was high pass filtered with a cutoff frequency of:

$$fc = 1/2\pi(30k\Omega)(10\mu F) = 0.53Hz$$

This was to remove any DC offset present. The signal was then sent through an LM358 for amplification with a gain of:

$$G = 1 + 51k\Omega/1k\Omega = 52$$

To reduce noise in the system, the signal was low passed filtered with a cutoff frequency of:

$$fc = 1/2\pi\ (51k\ \Omega)\ (0.47\mu F) = 6.6Hz$$

The output was fed through a 10kΩ trim-pot before the second amplifier stage. This served as a voltage divider so that only a fraction of the first stage would receive further amplification, providing a variable gain control. Because

International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 01, January 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

discrete amplifiers introduce DC offset, our signal was again high pass filtered at 0.53Hz and then amplified further. The second stage had a gain of:

$$G = 1 + 51k\Omega/330\Omega = 155$$

The output of this circuitry was then fed into the ADC ports of an Atmega1284p to be digitized and processed through software. Three channels were built and soldered as shown in Figure 2. A picture of the armband is also shown in Figure 3. The wires from the IR light and photo sensor were twisted in order to reduce noise artifacts from moving wires. Foam was placed in between each sensor and light pair in order to isolate each sensor so that it only picked up IR light reflecting of the user's arm.

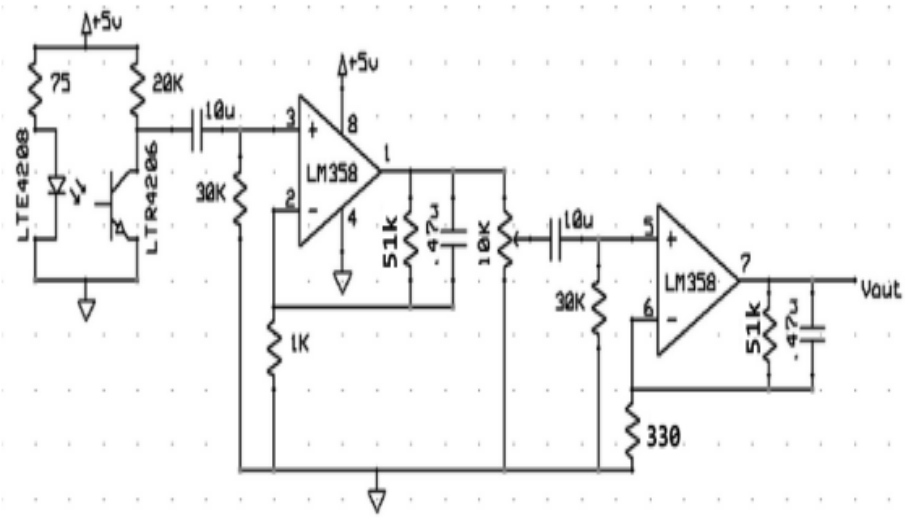


Fig. 1. Schematic of IR circuit.

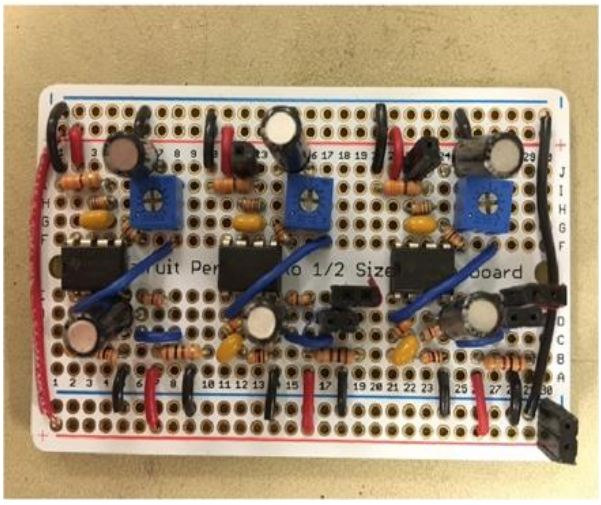


Fig. 2. Picture of IR circuit.

ISSN (O) 2454-9665
ISSN (P) 2454-0663

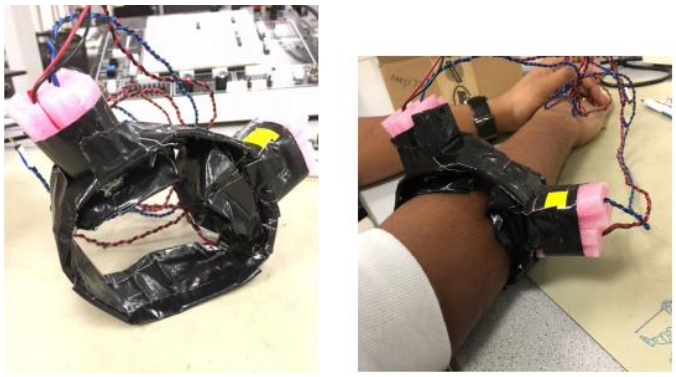


Fig. 3. Picture of IR armband.

II. SOFTWARE USED

There were three components to our software: data acquisition, machine learning, and real- time prediction.

a) Data Acquisition

To acquire the data from the signals, we fed the signals from the IR circuits into the ADC of an Atmega1284p microcontroller. This digitized our voltages within an 8-bit range (0-255). From here, we used a serial interface to send this data to a computer in real-time.

b) Machine Learning

In order to determine which signals corresponded to what, we needed to write some kind of program that was able to gather data from the microcontroller, format it appropriately, and feed it into a machine learning algorithm. We chose to use Python to accomplish all of these tasks. Python is a high-level language and has many modules readily available for various tasks. Specifically, we made use of the open-source scikit-learn machine learning module to handle our learning.

There were three modes to the program: see, learn, and predict [1, 2]. In see mode, the raw data from the microcontroller is simply shown on the screen. The data was acquired through serial communication with the Atmega1284p. See mode was mainly used for debugging purposes, so that we could know whether or not we were getting meaningful data or if the ADC was working properly.

In learn mode, the program would gather finger movement data from the user. For a given set of fingers (index, middle, ring, pinky, thumb, or any combination of the five), the program would ask the user to move the appropriate finger within a given time window and record data from each channel. The program would also keep track of which finger was being moved and would store it along with the ADC data. This process was repeated for a specified number of readings for each finger.

ISSN (O) 2454-9665
ISSN (P) 2454-0663

To verify the accuracy of our program, we used a technique called cross-validation. Cross-validation takes a percentage of the input data as the training data, and uses the remainder of the data as test data. The program learns the training data, and uses the model it builds to predict the test data. As the selection is random, different seeds can be chosen to ensure consistent results [3,4].

c) Prediction

The final component of our software and mode of our program, was real-time prediction. It has been created a real-time predictor. The predictor would use the SVM created after the user had already gone through the learning mode of the program once. Our real-time predictor would monitor the inputs coming in from the microcontroller. If the sample was above a given threshold, it would classify it as a finger movement [5, 6]. In this case, it would start recording the data, process it in the same way the learn mode does and input it into the predictor model [7, 8]. The predictor model would then tell us its prediction.

III. RESULTS

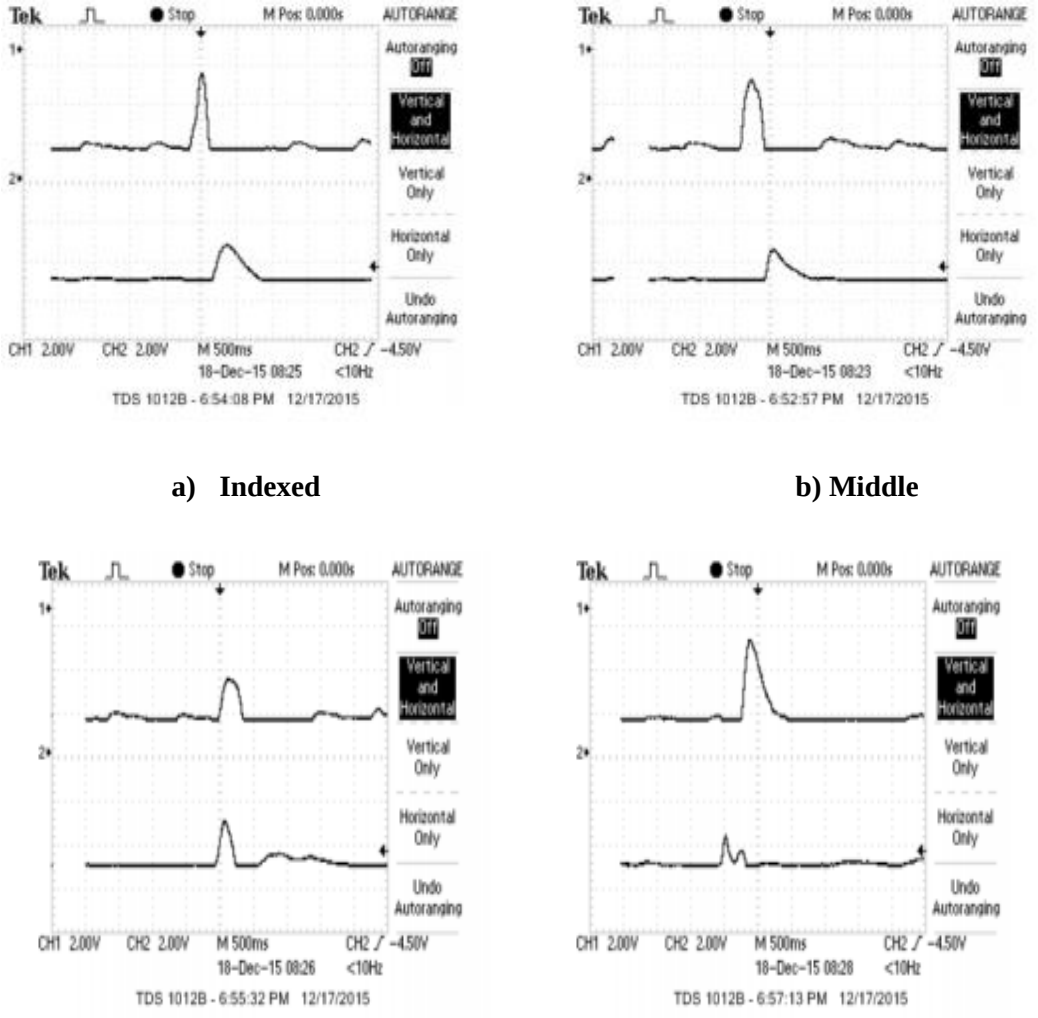
As seen in Figure 4, strong EMG signals were captured for movement of the index, middle, ring and pinky fingers. The frequency of these signals seemed to range from about 2-4Hz. Even from the oscilloscope waveforms, each finger movement is noticeably different. One-hundred readings were taken for the index, middle, and ring fingers. Using the default SVC kernel, our average prediction accuracy was 96.2%. This number was calculated by cross-validating our data over 100 different random seeds, and averaging the accuracy of each result. The max accuracy obtained was 100%, while the minimum accuracy observed was 80%. Table 1 shows a summary of these results, and other kernels for convenience. We tried 3 different kernels and compared the results. We also applied Principal Component Analysis (PCA) to our data and repeated the learning comparisons. PCA attempts to take a data set and rotate/stretch it along its principal component axes, thus increasing the variance of the data set.

Table 1. Machine learning prediction accuracy.

Statistic (%)			
Kernel	**Average**	**Maximum**	**Minimum**
Default	96.2	100	80
Linear	94.8	100	83.3

International Journal of Innovative Research in Science and Engineering
Vol. No.3, Issue 01, January 2017
www.ijirse.com
ISSN (O) 2454-9665
ISSN (P) 2454-0663

RBF	43	60	30
Default w/ PCA	67.5	90	50
Linear w/ PCA	57.7	76.6	33.3
RBF w/ PCA	66.6	86.7	46.7

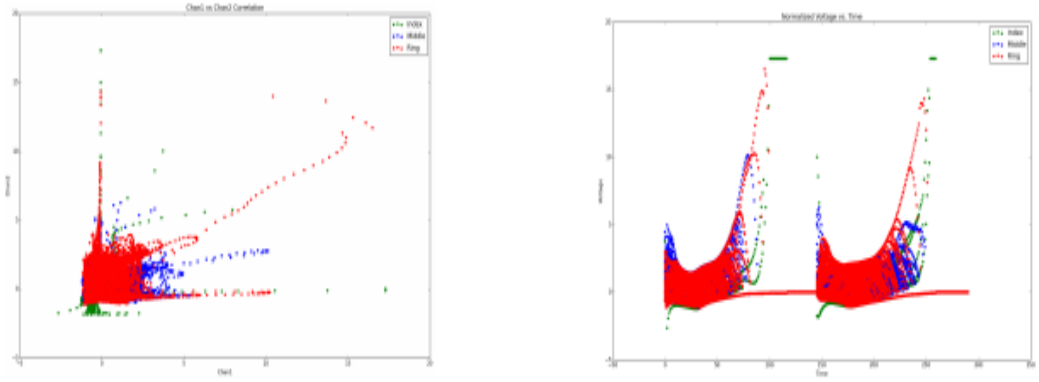


a) Indexed **b) Middle**

c) Ring **d) Pinky**

Fig. 4. EMG signals acquired from the circuit. Channel 1 is for middle (ED3), channel 2 is for ring (ED4), and channel 3 is for index (ED2).

ISSN (O) 2454-9665
ISSN (P) 2454-0663



(a) Channel correlation plot. **(b) Normalized voltage Vs. time.**

Fig. 5. Visualization of the machine learning data.

Figure 5 shows us a visualization of the data. Figure 5(a) shows the channel correlation while Figure 5(b) shows the voltage Vs. time plots. From both of these plots, it has been observed that there is much overlap between each of the fingers, but there are still distinct properties between them (ex. middle finger voltage on channel 2 takes longer to decay than the ring finger). Because there is so much overlap between the different data sets, we need a lot of data in order to be able to differentiate between them; otherwise the accuracy suffers.

IV. CONCLUSION

We successfully built a 2-channel circuit to detect the IR signals from different finger movements. Each finger tested showed distinct voltage waveforms upon moving. Through machine learning, we were able to accurately map three finger movements with 96% accuracy.

REFERENCES

[1]. Qiang Li, Bo Li Online Finger Gesture: Recognition Using Surface Electromyography Signals. Journal of Signal and Information Processing. Vol. 4, pp. 101-105 (2013).

[2]. Muhammad Zahak Jamal: Signal Acquisition Using Surface EMG and Circuit Design Considerations for Robotic Prosthesis. ISBN 978-953- 51-0805-4, (2012).

[3]. J. Rafiee, M.A. Rafiee, F. Yavari, and M.P. Schoen: Feature extraction of forearm EMG signals for prosthetics. Expert Systems with Application, (2011).

[4]. J.N.A.L. Leijnse, N. H. Campbell-Kyureghyan, D. Spektor, and P. M. Quesada: Assessment of Individual Finger Muscle Activity in the Extensor Digitorum Communis by Surface EMG. J Neurophysiol (2008).

ISSN (O) 2454-9665
ISSN (P) 2454-0663

[5]. O. Fukuda: A Human-Assisting Manipulator Teleoperated by EMG Signals and Arm Motions. Robotics and Automation, IEEE Transactions on, vol.19, no.2, pp.210-222, (2003).

[6]. Sebastian Maier and Patrick van der Smagt: Surface EMG surfaces to classify the motion of each finger independently. Conference: 9th International Conference on Motion and Vibration Control (2008).

[7]. The Rehabilitation Process: Prosthetic Devices. Ottobock. Retrieved from: https://ottobockus.com/zb2b4ob/us01/en/USD/Upper-Limb-Amputees.

[8]. Targeted Muscle Reinnervation. Advanced Arm Dynamics. Retrieved from: http://armdynamics.com/pages/tmr.