PROJECT REPORT (2016-2017)

Working of PPG sensors in fitness tracking watches

A STUDY OF UNDERLYING PPG ALGORITHMS AND IMPLEMENTING FAST FOURIER TRANSFORM

Under the guidance of

DR. RAJENDRA PAMULA

Assistant Professor

CSE Department

IIT-ISM, Dhanbad

Siddharth Shukla (B. Tech CSE)

3rd Semester

15JEOO1462

Acknowledgement

I take this opportunity to express my deep sense of gratitude and respect towards my project guide, Dr. Rajendra Pamula, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology (I.S.M), Dhanbad. I am very much indebted to him for the generosity, expertise and guidance that I have received from him while working on this project.

I wish to express profound gratitude to Prof. Chiranjeev Kumar, Head of Department, Computer Science and Engineering, Indian Institute of Technology (I.S.M), Dhanbad for his inspiration and guidance.

Date: -

Siddharth Shukla

15JE001462

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY (I.S.M)

DHANBAD – 826004

<u>CERTIFICATE</u>

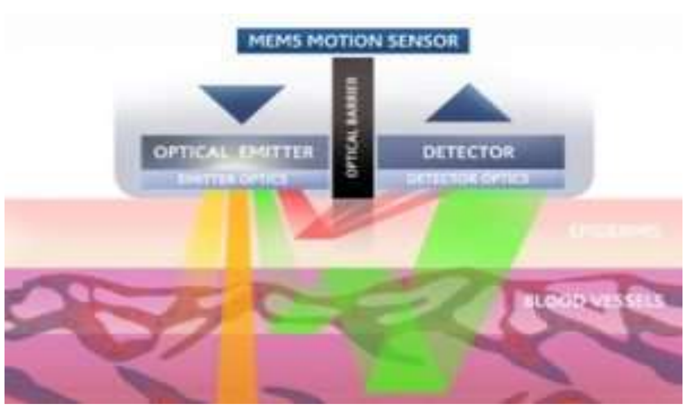
This is to certify that the report entitled "**Working of PPG sensors in fitness tracking gears**" submitted by **Siddharth Shukla**, Department of Computer Science and Engineering, Indian Institute of Technology, Dhanbad has successfully completed a project in 3rd Semester B. Tech of Academic year 2016-2017.

Prof. Chiranjeev Kumar	**Dr. Rajendra Pamula**
HOD	Assistant Professor (Project Guide)
Department of CSE	Department of CSE
IIT (ISM), Dhanbad	IIT (ISM), Dhanbad

INTRODUCTION

Most wearable with heart rate monitor today use a method called **photoplethysmography** or simply **PPG**. PPG is a technical term for shining light into the skin and measuring the amount of light that is scattered by blood flow. PPG is based on the fact that light entering the body will scatter in a predictable manner as the blood flow dynamics change, such as with changes in blood pulse rates (heart rate) or with changes in blood volume (cardiac output).

How does the technology work?



PPG uses four primary technical components to measure heart rate:

1. **Optical emitter**: - It is generally made up of at least 2 LED's that send green light waves into the skin. Because of the wide differences in skin tone, thickness, and

morphology associated with a diversity of consumers, most state-of-the-art OHRM's use multiple light wavelengths that interact differently with different levels of skin and tissue.

2. <u>Digital Signal Processor (DSP):-</u> The DSP captures the light refracted from the user of the device and translates those analog signals into ones and zeros that can be calculated into meaningful heart rate data.

3. <u>Accelerometer:-</u>The accelerometer measures simultaneous motion in three dimensions and the data is used in combination with the DSP signal as inputs into motion-tolerant PPG algorithms.

4. <u>Algorithms</u>:-The algorithms process the signals from the DSP and the accelerometer into motion-tolerant heart rate data, but can also calculate additional biometrics such as VO_2, calories burned, R-R interval, heart rate variability, blood metabolite concentrations, blood oxygen levels, and even blood pressure.

PPG is actually almost 150 years old, but it has been revolutionized in the 21st century for new uses. Real-time optical blood flow monitoring was first used in the late 1800's by having people hold their hand up to a candle in a dark room to see the vascular structure and blood flow. More recently in the early 1980's, the first pulse oximeters were launched for hospital use, measuring pulse rate and blood oxygen using two alternating LED's. These are very similar to the finger or ear clip devices still used in healthcare facilities today.

PPG developments in the last 5-10 years have focused on consumer applications of the technology to wearable devices. This required a radical development known as motion-tolerant PPG, because using PPG sensors during motion and activity massively increases the amount of motion noise that must be removed to find the blood flow signals.

Primary challenges with OHRM wearable

PPG sounds relatively simple, but it's actually very difficult to implement accurately for wearables. Measuring PPG during a resting state (sleeping, sitting, and standing still) is relatively straightforward, but measuring PPG during physical activity is incredibly complex. In fact, there are five fundamental challenges you will face in building wearable devices with OHRM:

1. **Optical noise:** – The biggest technical hurdle in processing PPG signals is separating the biometric signal from the noise, especially motion noise. Unfortunately, when you shine light into a person's skin only a small fraction of the light returns to the sensor, and of the total light collected, only ~1/1000th of it may actually indicate heart-pumped blood flow. The rest of the signals are simply scattered by other material, such as skin, muscle, tendons, etc.

2. **Skin tone:** – Humans have a diverse range of skin tones and different skin tones absorb light differently. For example, darker skin absorbs more green light, which presents a problem because most OHRM's use green LED's as light emitters, limiting their ability to accurately measure heart rate through dark skin. This also presents a problem for measuring heart rate through tattooed skin.

3. **Crossover problem:** – One of the most challenging aspects of optical noise for OHRMs that is created by motion and activity happens during what is known as periodic activity, which is activity that involves continuous repetition of similar motion. This is most often seen in the step rates measured during jogging and running, because step rates typically fall into the same general range as that of heartbeats (140-180 beats/steps per minute). The problem that many OHRMs face is that it becomes easy for the algorithms interpreting incoming optical sensor data to mistake step rate ("cadence") for heart rate. This is known as the "**crossover problem**", because if you look at the measurements on a graph, when the heart rate and step rate crossover each other, many OHRMs tend to lock on to step rate and present that number as the heart rate, even though the heart rate may be changing drastically after the crossover.

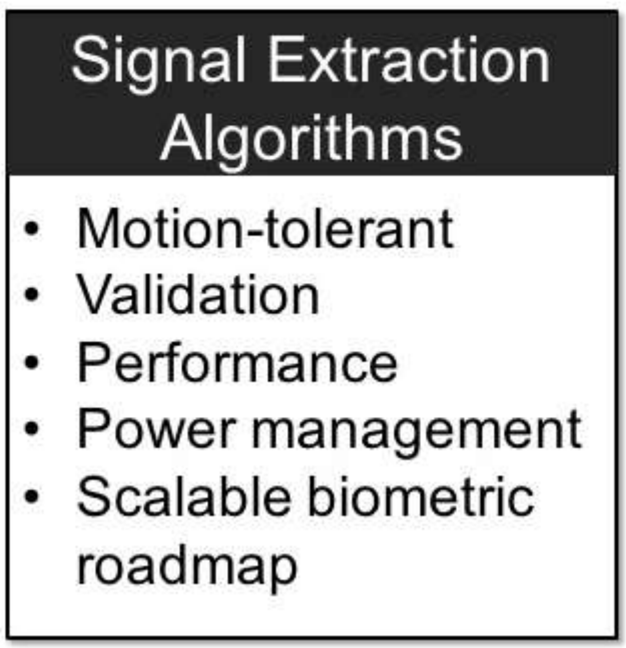
4. **Sensor location:** – The location of the OHRM on the body presents unique challenges that vary significantly by location. It turns out that the wrist is one of the worst places for accurate PPG monitoring of heart rate because of the much higher optical noise created in that region (muscle, tendon, bone, etc.) and because of the high degree of variability in vascular structure and blood perfusion across the human populations. The forearm is considerably better because of the higher density of blood vessels near the surface of the skin. However, the ear is by far the best location on the body for OHRM because it is essentially just cartilage and blood vessels, which don't move much even when the body is in vigorous motion, and

because of an ideal arteriole bank between the anti-tragus and concha of the ear, thereby drastically reducing the optical noise that must be filtered.

5. **<u>Low perfusion:</u>** – Perfusion is the process of a body delivering blood to capillary beds. As with skin tone, the level of perfusion is highly variable across populations, with issues such as obesity, diabetes, heart conditions, and arterial diseases each lowering blood perfusion. Low perfusion, especially in the body's extremities where most wearable devices are located, can present challenges for OHRMs because the signal-to-noise ratio may be drastically reduced, as lower perfusion correlates with lower blood flow signals. The head region (including the ear, temple, and forehead) supports much higher perfusion and better quality photoplethysmograms than the wrists or feet.

Motion tolerant Signal Extracting PPG Algorithms

OHRM is obviously very difficult to do accurately, but it's certainly possible and here's how. At a high level, you need to have very good signal extraction algorithms.



The Proposed Algorithm

Photoplethysmography (PPG) is a non-invasive optical technique that can be used to quantify the arterial blood pulse rate. Signal corruption by motion artifacts limits the practical accuracy and applicability of instruments for monitoring pulse rate during intense physical exercise.

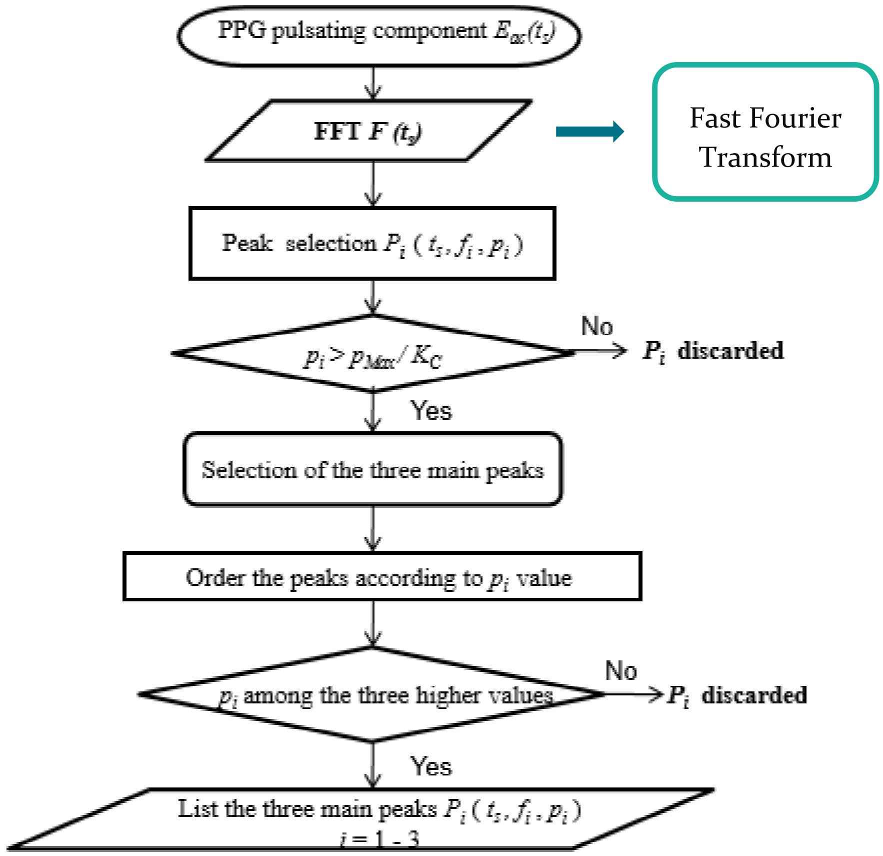
This algorithm is based on linear filtering and frequency-domain and heuristic analyses, for extracting the heart rate from a PPG signal in the presence of severe motion artifacts. The basis of the hearth beat frequency selection is the observed high harmonic content of movement artifact signals with respect to the PPG-derived heartbeat. The **Bland-Altman method** is used to compare and evaluate PPG signals. The accuracy of the heartbeat measurement is better than ± 6.5 beats per minute (bpm) (≤ 4.2 %) even under maximal exercise conditions.

In most cases, the noise falls within the frequency band of the physiological signal of interest, rendering linear filtering ineffective. Algorithms have been developed to reduce the sensitivity of PPG signals to artifacts commonly encountered in clinical environments and during controlled or moderate motion. A non-contact, laser-based remote PPG system has been proposed, with positive results obtained in clinical applications. PPG has been also applied to the evaluation of severe exercise during the incremental maximal exercise test (IMET) on a cycle ergometer, where several parameters were monitored, among them oxygen saturation and heart rate (HR). Active research efforts are beginning to demonstrate that PPG has utility beyond oxygen saturation and HR determination. For instance, the conditions required for a correct utilization of PPG HR variability (PPGV) have been recently studied. Future trends are being heavily influenced by modern digital signal processing. New commercial developments of the PPG-based prototype suggest that a new interest is growing about PPG techniques for HR tracking.

Under severe exercise conditions, motion artifacts present a challenge for PPG analysis. Sensor and packaging designs can help reduce the impact of motion disturbance, but they are rarely sufficient for noise removal. Advanced signal processing techniques are often required to remove motion artifacts under vigorous activity. Several techniques have been developed to deal with such artifacts, such as processing context information from additional on-body sensors and light sources and adaptive noise cancellation using accelerometers as a noise reference. Very recently, heart rate measurements have been taken using a PPG sensor integrated with an adaptive noise cancellation device during common physical activity, from walking to running at up to 8 km/h.

Spectral analysis, such as the traditional fast Fourier transform (FFT), is a simple and inexpensive tool for separating motion artifacts and cardiac physiologic spectra.

The algorithm can be represented by the following flowchart: -



IMPLEMENTING FAST FOURIER TRANSFORM USING C/C++

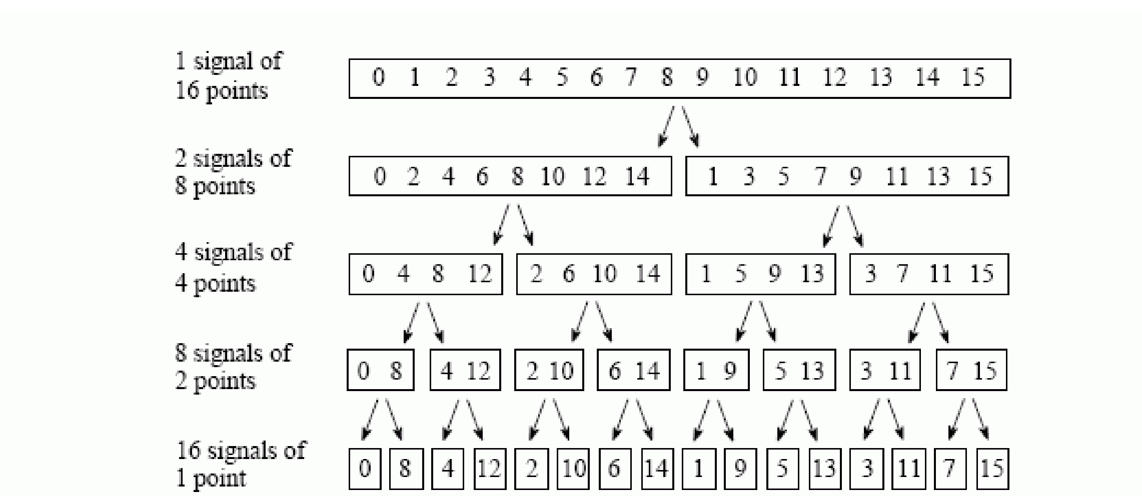
With the actual algorithm beyond the scope of graduation level, I have tried to implement **Fast Fourier Transform** using C/C++ programming language that is used in the first step of the proposed algorithm to separate motion artifacts from the heart data. FFT is used for separating motion artifacts and cardiac physiologic spectra in terms of complex time – frequency signals.

How FFT works? [MUST READ BEFORE UNDERSTANDING THE CODE]

The FFT is a complicated algorithm, and its details are usually left to those that specialize in such things. This section describes the general operation of the FFT, but skirts a key issue: the use of complex numbers. Few scientists and engineers that use the FFT could write the program from scratch.

In complex notation, the time and frequency domains each contain **one signal** made up of **N complex points**. **The input that I have considered is one signal made up of 16 complex points. Each of these complex points is composed of two numbers, the real part and the imaginary part. For example, when we talk about complex sample X[42], it refers to the combination of ReX[42] and ImX[42].** In other words, each complex variable holds two numbers. When two complex variables are multiplied, the four individual components must be combined to form the two components of the product. The singular terms: signal, point, sample, and value, refer to the combination of the real part and the imaginary part.

The FFT operates by decomposing an N (here a 16 point time domain signal) point time domain signal into N time domain signals each composed of a single point. The second step is to calculate the N frequency spectra corresponding to these N time domain signals. Lastly, the N spectra are synthesized into a single frequency spectrum (which is also a 16 point frequency domain signal output, in this case).



The above diagram shows the decomposition of an N (here a 16 point time domain signal) point time domain signal into N time domain signals each composed of a single point. There are *Log_2N* stages required in this decomposition, i.e., a 16 point signal (2^4) requires 4 stages.

C Code

```
/* INPUT: float input[16], float output[16] */
/* Re{F[0]}= out0 */
/* Im{F[0]}= 0 */
/* Re{F[1]}= out8 */
/* Im{F[1]}= out12 */
/* Re{F[2]}= out4 */
/* Im{F[2]}= -out6 */
/* Re{F[3]}= out11 */
/* Im{F[3]}= -out15 */
/* Re{F[4]}= out2 */
/* Im{F[4]}= -out3 */
/* Re{F[5]}= out10 */
/* Im{F[5]}= out14 */
/* Re{F[6]}= out5 */
/* Im{F[6]}= -out7 */
/* Re{F[7]}= out9 */
/* Im{F[7]}= -out13 */
/* Re{F[8]}= out1 */
/* Im{F[8]}=0 */
/* F[9] through F[15] can be found by using the formula */
/* Re{F[n]}=Re{F[(16-n)mod16]} and Im{F[n]}= -Im{F[(16-n)mod16]} */
/* The algorithm behind this program is to find F[2k] and F[4k+1] */
/* separately.  To find F[2k] we take the 8 point Real FFT of x[n]+x[n+8] */
```

/* for n from 0 to 7. To find F[4k+1] we take the 4 point Complex FFT of */

/* exp(-2*pi*j*n/16)*{x[n] - x[n+8] + j(x[n+12]-x[n+4])} for n from 0 to 3.*/

```
#include <stdio.h>
#define SIN_2PI_16 0.3826
#define SIN_4PI_16 0.7071
#define SIN_6PI_16 0.9238
#define C_P_S_2PI_16 1.3065
#define C_M_S_2PI_16 0.5411
#define C_P_S_6PI_16 1.3065
#define C_M_S_6PI_16 -0.5411

void FFT(float input[16],float output[16] ) {
  float temp, out0, out1, out2, out3, out4, out5, out6, out7, out8;
  float out9,out10,out11,out12,out13,out14,out15;
  out0=input[0]+input[8]; /* output[0 through 7] is the data that we take the 8
                              point real FFT of. */
out1=input[1]+input[9];
out2=input[2]+input[10];
  out3=input[3]+input[11];
  out4=input[4]+input[12];
  out5=input[5]+input[13];
  out6=input[6]+input[14];
  out7=input[7]+input[15];
```

out8=input[0]-input[8]; /* **inputs 8,9,10,11 are the Real part of the 4 point Complex FFT inputs. Outputs 12,13,14,15 are the Imaginary pars of the 4 point Complex FFT inputs.***/

out9=input[1]-input[9];

out10=input[2]-input[10];

out11=input[3]-input[11];

out12=input[12]-input[4];

out13=input[13]-input[5];

out14=input[14]-input[6];

out15=input[15]-input[7]; /* **C_M_S_2PI/16=cos(2pi/16)-sin(2pi/16) when replaced by macro expansion and C_P_S_2PI/16=cos(2pi/16)+sin(2pi/16) when replaced by macro expansion and (SIN_2PI_16)=sin(2pi/16) when replaced by macro expansion** */

temp=(out13-out9)*(SIN_2PI_16);

out9=out9*(C_P_S_2PI_16)+temp;

out13=out13*(C_M_S_2PI_16)+temp;

out14*=(SIN_4PI_16);

out10*=(SIN_4PI_16);

out14=out14-out10;

out10=out14+out10+out10;

temp=(out15-out11)*(SIN_6PI_16);

out11=out11*(C_P_S_6PI_16)+temp;

out15=out15*(C_M_S_6PI_16)+temp;

```
out8+=out10;

out10=out8-out10-out10;

out12+=out14;

out14=out12-out14-out14;

out9+=out11;

out11=out9-out11-out11;

out13+=out15;

out15=out13-out15-out15;

output[1]=out8+out9;

output[7]=out8-out9;

output[9]=out12+out13;

output[15]=out13-out12;

output[5]=out10+out15;

output[13]=out14-out11;

output[3]=out10-out15;

output[11]=-out14-out11;
```

/* **What follows is the 8-point FFT of points output[0-7] .This 8-point FFT is basically a Decimation in Frequency FFT where we take advantage of the fact that the initial data is real***/

```
out0=out0+out4;

out4=out0-out4-out4;

out1=out1+out5;

out5=out1-out5-out5;

out2+=out6;

out6=out2-out6-out6;

out3+=out7;
```

```
  out7=out3-out7-out7; /* Computations to find X[0], X[4], X[6] */
  output[0]=out0+out2;
  output[4]=out0-out2;
  out1+=out3;
  output[12]=out3+out3-out1;

  output[0]+=out1;/* Computations to find X[0], X[4], X[6] */
output[8]=output[0]-out1-out1; /*Real Part of X[4] */
/* out2 = Real Part of X[6] */
/* out3 = Imag Part of X[6] */
/* Computations to find X[5], X[7] */
  out5*=SIN_4PI_16;
  out7*=SIN_4PI_16;
  out5=out5-out7;
  out7=out5+out7+out7;

  output[14]=out6-out7; /* Imaginary Part of X[5] */
  output[2]=out5+out4; /* Real Part of X[7] */
  output[6]=out4-out5; /*Real Part of X[5] */
  output[10]=-out7-out6; /* Imaginary Part of X[7] */
 }

void main() {
  float data[16];
```

```
float output[16];

float zero=0;

printf("\ntype 16 point input vector\n");

scanf("%f   %f   %f   %f   %f   %f   %f   %f   %f   %f   %f   %f   %f   %f   %f
%f",&data[0],&data[1],&data[2],&data[3],&data[4],&data[5],&data[6],&data[7],&dat
a[8],&data[9],&data[10],&data[11],&data[12],&data[13],&data[14],&data[15]);

FFT(data , output);

printf("\nresult is:\n");

printf("k,\t\tReal Part\t\tImaginary Part\n");

printf("0\t\t%.9f\t\t%.9f\n",output[0],zero);

printf("1\t\t%.9f\t\t%.9f\n",output[1],output[9]);

printf("2\t\t%.9f\t\t%.9f\n",output[2],output[10]);

printf("3\t\t%.9f\t\t%.9f\n",output[3],output[11]);

printf("4\t\t%.9f\t\t%.9f\n",output[4],output[12]);

printf("5\t\t%.9f\t\t%.9f\n",output[5],output[13]);

printf("6\t\t%.9f\t\t%.9f\n",output[6],output[14]);

printf("7\t\t%.9f\t\t%.9f\n",output[7],output[15]);

printf("8\t\t%.9f\t\t%.9f\n",output[8],zero);

printf("9\t\t%.9f\t\t%.9f\n",output[7],-output[15]);

printf("10\t\t%.9f\t\t%.9f\n",output[6],-output[14]);

printf("11\t\t%.9f\t\t%.9f\n",output[5],-output[13]);

printf("12\t\t%.9f\t\t%.9f\n",output[4],-output[12]);

printf("13\t\t%.9f\t\t%.9f\n",output[3],-output[11]);

printf("14\t\t%.9f\t\t%.9f\n",output[2],-output[9]);

printf("15\t\t%.9f\t\t%.9f\n",output[1],-output[8]);
```

}

PROGRAM WINDOW:

```
type 16 point input vector
12.3 12.4 12.5 12.5 12.6 12.7 12.8 12.9 13.0 13.1 13.2 13.3 13.4 13.5 13.6 13.7

result is:
k,              Real Part               Imaginary Part
0               207.500000000           0.000000000
1               -0.536901474            3.912892580
2               -0.629291177            1.760661006
3               -0.732442617            1.034187436
4               -0.800003052            0.699996948
5               -0.808978796            0.512865663
6               -0.770710349            0.360659480
7               -0.721676350            0.191573858
8               -0.699996948            0.000000000
9               -0.721676350            -0.191573858
10              -0.770710349            -0.360659480
11              -0.808978796            -0.512865663
12              -0.800003052            -0.699996948
13              -0.732442617            -1.034187436
14              -0.629291177            -3.912892580
15              -0.536901474            0.699996948
```

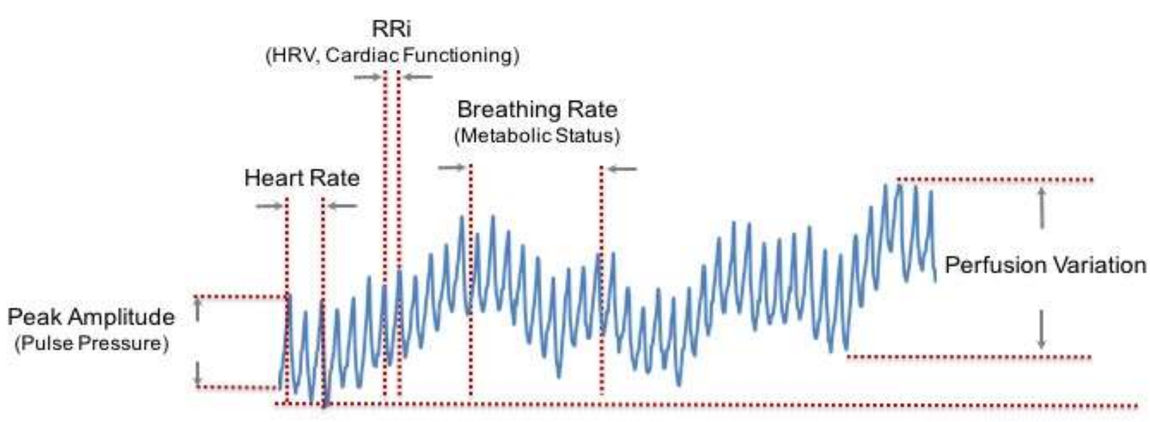
INPUT: - It is a 16 point time domain signal.

OUTPUT: - It is a single 16 point frequency spectrum with both real as well as imaginary parts from which peak selection is done on the basis of the proposed algorithm in Page 8.

What metrics can be derived from PPG?

While PPG is really hard to get right, but when done, it can be very powerful. A high-quality PPG signal is foundational to a wealth of biometrics that the marketplace is demanding today. For example, here's a list of some (but not all) of the biometrics that can be derived from highly accurate, motion-tolerant PPG:

- **Breathing Rate**: - Breathing rate is the number of breaths taken in a period of time (typically 60 seconds) and lower resting breathing rates are generally correlated with higher levels of fitness.
- **VO_2 max**: - VO_2 measures the maximum volume of oxygen someone can use and VO_2max is widely considered to be an indicator of aerobic endurance.
- **Blood oxygen levels (SpO_2, oxygen saturation)**: - Blood oxygen levels indicate the concentration of oxygen in the blood.
- **R-R Interval (heart rate variability)**: - In layman's terms, R-R interval is the time between blood pulses (or ECG beats), and generally the more varied the time between beats, the better. R-R interval analysis can be used as an indicator of stress levels and various cardiac issues, among other things.
- **Blood pressure**: – Most people are very familiar with blood pressure as an indicator of cardiovascular health, but most people don't know that some of the most advanced technologies today can assess blood pressure using PPG signals.
- **Cardiac Efficiency**: – This is another indicator of fitness that typically measures how efficiently your heart works to take one step. This serves as a proxy for how hard your heart would have to work to do more challenging exercises like running or cycling.



SUMMARY

It's that time of year, when everyone recommits to a healthy lifestyle. Increasingly, people are turning to activity trackers—electronic devices that track everything from caloric expenditure to quality of sleep—to help them stay on course and meet their health and fitness goals.

An estimated 19 million devices were in use in 2014, and that number is expected to grow exponentially over the next few years. In fact, a recent report by Juniper Research predicts that the use of activity trackers—also called fitness wearables will triple by 2018. While all this new technology is really cool and some of it is really fun to use, very little published research exists demonstrating the accuracy or validity of these devices.

Most devices are pretty good for measuring steps taken during traditional activities. Once you start getting outside of that—like elliptical or sports-related movements—it becomes harder to detect actual data .

This project discussed an algorithm for eliminating signals caused by motion artifacts that limit the use of PPG in quantifying key cardiovascular variables such as the HR (Heart Rate) during exercise. The algorithm converts PPG signals collected into an HR value. It identifies three frequencies of interest from the PPG signal and uses decision-making logic to identify the two harmonic frequencies that most likely represent motion artifacts, leaving one frequency to represent the PPG-derived HR. The algorithm allows for significant advancement in the area of biosensor development as PPG technology is ideally suited to be incorporated into wearable devices.