ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

A mm-Wave Breath Monitoring System for Smart Vehicle Applications

---

Team #20

KEYU LU (keyulu2@illinois.edu)
KANGNING LI (kl32@illinois.edu)
HE CHEN (hechen4@illinois.edu)
BOWEN SONG
(bowen15@illinois.edu)

Sponsor: Shurun Tan
TA: Xuyang Bai

May 12, 2023

Abstract

Ensuring the safety of children left in hot cars has been a persistent issue throughout the 21st century. To address this problem, we propose the use of a radar system capable of detecting the breath of children in vehicles. Our objective is to create a radar system specifically designed to detect the breath of children, providing an effective solution. Upon detection, the system will promptly send an alarm to alert parents or guardians. To achieve this goal, we use a mm-wave radar and have implemented radar signal processing techniques for breath detection. The radar system utilizes Frequency-Modulated Continuous Wave (FMCW) to detect breathing patterns. For the hardware and data collection aspects of the project, our team has opted to utilize the TI-60GHz mm-wave radar development board IWR6843ISK-ODS. Moving forward, we will employ Matlab software for essential data processing tasks such as distance detection, phase unwrapping, and signal denoising.

Keywords: Frequency-Modulated Continuous Wave (FMCW), Breath detection.

Contents

1 Introduction

In order to complete the hardware link and data collecting, our team ultimately decided to employ the TI-60GHz mm-wave radar development board IWR6843ISKODS and CP210 driver. They allow us to collect raw breathing data as the output of the radar system.

The major change we made during the semester is that we consider all the above hardware part as one subsystem. This means now the Radio Frequency (RF) system and Simulation system in our Design Document are now radar system. We divide the Control system into two smaller subsystems–distance detection and signal denoising, because the two are the key factor influencing our project's performance.

After data collection, Matlab software is used to apply micro-doppler detection and millimeter wave radar range detection. This is the distance detection system. The phase time domain graph is evaluated to identify the breathing rate once the FFT graph has been generated in Matlab using a predetermined possible distance range to assess the potential objects. Aim of the signal denoising system is to have better result. Better means the differences between breathing point and non-breathing point are larger and clearer. With the breathing rate we calculate from Matlab, we use the infinite impulse response filter (IIR filter) and Savitzky-Golay filter (SG filter) to denoise. These two filters can remove some background interference and high frequency noise from our results.

Finally, we design a GUI from Matlab application designer to allow users to know the breathing status in the car. User interface system allows users to operate our system easier and it can give warning through email, text, and phone call.

For power system, a power bank can provide the 5V power supply for the radar board.

2 Design

2.1 Design Procedure

Our design includes several blocks: data collection and power system, distance detection and phase unwrapping system, graphics user interface system, noise reduction system, reminder system and automatic breathing detection system.

Data collection and power system includes a radar board and a power supply. The board we chose is a TI-60GHz mm-wave radar development board IWR6843ISKODS and CP210 driver. Otherwise we can choose camera detection or infrared detection, but they are more expensive and will be influenced more by clothing obscuration. And radar is of better privacy and is more anonymous. Also, the radar can give a fine respiratory waveform that can be used for further feature detection. So finally we choose the radar as our detection tool. For power supply, we use 5V power supply as TI(Texas Instruments) recommended.

For distance detection and phase unwrapping system, we choose to find the distance

using phase and unwrapping the phase using DACM algorithm. Initially we choose to use FFT to find the distance, but it will be influenced more by the surroundings so we finally choose to not use it.

For graphics user interface system, we choose to use the GUI designer in MATLAB. Initially we want to design the user interface using python, however, as our signal analyzing processes are in MATLAB, it will be more convenient using the GUI designer inside the MATLAB. So we finally choose to use the GUI designer in MATLAB.

For noise reduction system, we choose to use the infinite impulse response(IIR) filter and Savitzky-Golay (SG) filter. Alternatively we can choose to avoid the background interference and use the generative adversarial network(GAN) to generate clean image. But when avoiding the background interference, it will destroy the phase information. And the GAN will lose some significant amplitude information and the training datasets is not large and accurate enough, so we finally choose not to use them.

For reminder system, we have worked out all we want includes email reminder, SMS reminder and voice reminder. They are enough for our product.

For automatic breathing detection system, we figure out the breathing wave by forcing amplitude, frequency and FFT peak value. Otherwise we can choose to train a neural network to work on pattern recognition. However, as the datasets and the training resource are limited, we finally didn't use it.

2.2 Design Detail

2.2.1 Using generative adversarial model(GAN) to Denoise

Our project requires converting a noise-filled waveform map into a clean breath map, and this can be done using the image transformation of the generative adversarial model(GAN)[1]. We choose to train a pixel-to-pixel GAN model proposed in [2] with pairs of noisy and clean breathing images, and then use the generator to implement the transformation between noisy and clean breathing images.

The loss function we use includes two part, the conditional GAN loss function to learn the translation relationship and the L1 loss function to encourage less blurring.

The conditional GAN loss function $\mathcal{L}_{GAN}$ can be expressed as:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,y}[\log(1 - D(x, G(x)))] \tag{1}$$

where x is our noisy waveform images and y is our real clean breathing images. The generator(G) tries to minimize the conditional GAN loss function against an adversarial discriminator(D) that tries to maximum it.

The L1 loss function $\mathcal{L}_{L1}$ can be expressed as:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y}[||y - G(x)||_1] \tag{2}$$

Our final objective can be:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{3}$$

Finally, we want to train a generator which can fool the discriminator, which means it can generator fake images which is similar to the clean breathing images from the noisy images. And in this way, we can use our generator to perform as a filter to denoise.

Because of the lack of real data, we generated our own datasets. We randomly generate two sine functions, and combining them to simulate real breathing. To simulate noisy image we have added random Gaussian noise as environment noise.

Before feeding the image into the model, we preprocess the image by resizing it to 256*256 and then cropping it. This speeds up the model convergence and makes the model more robust.

However, because of cropping, the data after denosing will loss significant information about the axis. And if we don't use the cropping, our computer couldn't successfully train it. So we finally decide to not use GAN as our denoising method.

2.2.2 Differentiate and Cross Multiply(DACM) Algorithm

The radar is usually obtained as IQ two-branch signal, and the mismatch between the two branches is inevitable, mentioning a DACM algorithm to recover the phase information of the signal.

$$\omega(t) = \frac{d}{dt}\left[\arctan\frac{Q(t)}{I(t)}\right] = \frac{I(t)\dot{Q}(t) - \dot{I}(t)Q(t)}{I(t)^2 + Q(t)^2} \tag{4}$$

$$\varphi(m) = \sum_{i=2}^{m}\{\omega[k]\cdot\Delta t\} = \sum_{i=2}^{m}\frac{I[i](Q[i]-Q[i-1]) - Q[i](I[i]-I[i-1])}{I^2[i]+Q^2[i]} \tag{5}$$

The principle is an approximate implementation of a discrete form of integrating the signal after the inverse tangent differentiation. Its real purpose is that in the integration operation can be some high-frequency noise suppressed.

2.2.3 Using Raspberry Pi as User Interface

We use a Texas Instruments(TI) official software called mmWave Studio on a computer to control the parameter of radar to collect desired data and analyze the data collected by radar using Matlab. However, our product is expected to be used in a limited space in a car, so we want the device to be as tiny as possible. Also, as the software we use are based on Windows system, we have to install Windows system on Raspberry Pi. We successfully use a software Windows on Raspberry[3] to install the image to the SD card and then install the system on the Raspberry. One significant issues for Windows is that due to the incompatibility of the driver, the Windows on Raspberry Pi can not use WIFI to get connected with the internet. To solve this problem, we use a network cable to

connect the Raspberry Pi to the network port. However, due to that ARM architecture is not compatible with the x86 and win32 file, so the driver installation file cannot be run, so we can't use Raspberry Pi to control radar as we expected before and decide to not use Raspberry Pi.

2.2.4 Distance Detection

We employ the DCA1000 Board to collect data from the radar board. Chirps in the Intermediate Frequency (IF) signal have its frequency depending on how far away the target object is.

$$f_{IF} = \frac{S2d}{c} \tag{6}$$

The c is the speed of light and d is the distance we want. From the slope (s) of the Frequency-Modulated Continuous Wave (FMCW) and the frequency of the IF signal, we can calculate the distances of objects.

From the slope of the Frequency-Modulated Continuous Wave (FMCW) and the frequency of the IF signal, we can calculate the distances of objects.

The control system in our design is responsible for this portion of the work. Peaks in the frequency spectrum are in precise relation to the variety of items.

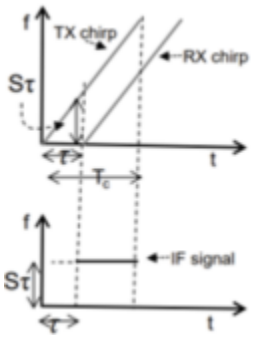

Figure 1: IF Signal[4]

2.2.5 Graphics User Interface(GUI)

To make our project easier to use, we create a GUI for it. Without the GUI, a user must run our code, which is not very user-friendly, to use our project. We use MATLAB App Designer to construct the GUI because MATLAB is where most of our codes are implemented.

2.2.6 Sending Reminder via Email

Implement this function using Matlab. In order to realize the function, a server email address is required. Multiple choices could serve as the central server. We choose the 163 Email as the tool. The 163 Email exploits the security code mechanism to avoid the case

that anyone holding the password could sign in to the email.[5] Considering the security of the central server email, POP3/SMTP should be opened to get the security code to replace the password. After setting the central server email account, SMTP server should be also set. The user account should be given as the receiver's email. The email content is set as a reminder that whether there is a child left in the car.

Then we could run this function in the terminal to check the performance.

2.2.7 Sending Reminder via Phone message

Sending messages to the user's phone requires a server phone. And the cost of sending messages should also be considered. So we apply for the free phone number on Twilio.com. Twilio allows developers to receive phone calls, send and receive text messages, and more by programming with its API[6]. This website provides us with a free phone account with 15 dollars free spending limit. To register for the free account, I need to provide my own phone number and activate the SMS service on Twilio.com. After applying for the free phone number, I implement the function using Python. Set the applied free phone number as the central server. Set my phone number as the user phone number. Then test whether I can receive the reminder email.

We also test the cost of one message to measure the total usage limit of this free phone number. If we want to extend the function to be more widely used, a paid service is required. And this free SMS service also limits the target of messages to only the registration phone account. If we want to use it to send messages to any other users, a paid service is required.

2.2.8 Voice Reminder

To Remind the user of the presence of children left in the car, we implement the voice broadcast function. There are two cases. Firstly, when there is a child in the car, the function will automatically broadcast "Attention! There is a child in the car!". Secondly, when there is no person in the car, the function will broadcast "Safe. There is no person in the car."

We implement the function using Matlab. And we use the computer voice recorder to generate two versions of the broadcasting file.

2.2.9 Infinite Impulse Response(IIR) Filter

The frequency of breathing is range from 0.1Hz to 0.5Hz. And this breathing signal has a simple signal model which is similar to Sin/Cos wave. Because there might be noise in the breathing signal, we decide to design the filter using the IIR method as our denoising method.

Matlab already provides users with many implementations of different filters. Here we choose to exploit the Ellipse filter. Eq.7 is the amplitude of the frequency response of a

low-pass elliptic filter[7]:

$$G_n(\omega) = |H_n(j\omega)| = \frac{1}{\sqrt{1+\epsilon^2 R_n^2(\omega)}} \tag{7}$$

where R_n is n^{th} order Chebyshev rational functions.

To design the elisp filter, some parameters should be set according to the constraints[8]. Elisp filter should be set with the order and normalized cutoff frequency. Thankfully, Matlab has already implemented a helper function, $[n, W_n] = ellipord(W_p, W_s, R_p, R_s)$, where W_p means Passband frequency, W_s means Stopband frequency, R_p means Passband corrugation, R_s means Resistance band attenuation. After we get n and W_n, use function $[b, a] = ellip(n, R_p, R_s, W_n)$ to design the Elisp filter. And our sampling frequency Fs is 100Hz.

Considering our background of breathing, a Lowpass Elisp filter should be designed. But the helper function exists one problem all classical IIR low-pass filters are not suitable for very low cutoff frequencies. So Passband frequency should be seriously chosen and starts from 0Hz.

2.2.10 Savitzky-Golay(SG) Filter

To smooth out the noisy data, we use Savitzky-Golay filter to eliminate data points with large. Savitzky-Golay filter fits the signal data by shifting the window using least-squares polynomial method and can directly handle data smoothing problem in the time domain. Therefore, with its simplicity and speed, we decide to adopt it for filtering our resultant waveforms.

The working principle of least-squares polynomial is fitting our sample data x[n] into the polynomial:

$$p(n) = \sum_{k=0}^{N} a_k n^k \tag{8}$$

Assume that we have $2M+1$ samples centered at $n = 0$, then the mean-squared approximation error could be represented as:

$$\varepsilon_N = \sum_{n=-M}^{M} (p(n) - x[n])^2 = \sum_{n=-M}^{M} (\sum_{k=0}^{N} a_k n^k - x[n])^2 \tag{9}$$

We hope to obtain the value of coefficients and decide to minimize the mean-squared approximation error as shown above by transforming the sample points into matrix form. We could build the $(N+1) * (2M+1)$ matrix X^T as shown below:

$$\begin{bmatrix} (-M)^0 & \cdots & (-1)^0 & 1 & 1^0 & \cdots & M^0 \\ (-M)^1 & \cdots & (-1)^1 & 0 & 1^1 & \cdots & M^1 \\ (-M)^2 & \cdots & (-1)^2 & 0 & 1^2 & \cdots & M^2 \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ (-M)^N & \cdots & (-1)^N & 0 & 1^N & \cdots & M^N \end{bmatrix} \tag{10}$$

Also, the coefficients could be represented as $(N+1)*1$ matrix A: $\begin{bmatrix} a_0 \\ \vdots \\ a_N \end{bmatrix}$ and the original sample points could be represented as $(2M+1)*1$ matrix Y: $\begin{bmatrix} y_0 \\ \vdots \\ y_{2M+1} \end{bmatrix}$ Then, according to the least-squares polynomial method, we could get the solution $\hat{A}$:

$$\hat{A} = (X^T X)^{-1} X^T Y \tag{11}$$

Finally, the fitted values of the data points are $\hat{Y}$:

$$\hat{Y} = XA = X(X^T X)^{-1} X^T Y \tag{12}$$

2.2.11 Breathing Frequency Estimation

After we obtain the detection result of breathing waveform, we hope to derive the breathing frequency from it. With the breathing frequency, we could determine if the frequency is within our expected range and thus know if the breathing detection waveform is valid or if the tester is breathing normally.

The simplest way is to find out the peaks and calculate the time interval between two peaks to get the breathing frequency:

$$f_{breathing} = \frac{1}{\frac{1}{n-1}\sum_{i=1}^{n-1}(t_{i+1} - t_i)} \tag{13}$$

where t_i refers to the time of peak i.

However, this method may be inaccurate when the signal contains high frequency noise. Thus, we plan to directly extract the breathing frequency from the analysis of frequency domain of breathing waveform. First of all, to avoid the interference of DC component, we eliminate the DC component by subtracting the average from the original signal:

$$x'_n = x_n - average(x_n) \tag{14}$$

Then, we adopt the Fast Fourier Transform method to obtain the result in frequency domain. Finally, the breathing frequency should be the horizontal coordinate of the highest peak.

2.2.12 Automatic Breathing Detection

As we mentioned in the previous section, the range resolution of our system is approximately 5cm. In our original algorithm, it will produce waveform result for every 5cm. Therefore, range detection method is adopted to predict where the correct result waveform should occur. However, we find out that the accuracy of the range detection method is not that precise, and thus we always need to manually find and identify the waveform results which costs us a lot of time.

Since we have calculated the breathing frequency by doing FFT transform to the signal, I recognize that I could make fully use of the breathing frequency to judge if a waveform result satisfies the requirements of humans' breathing waveform. First of all, I let the code traverse over a large range. For example, I will output all testing results among 10 to 40 (which refers to the distance 0.5m - 2m in reality). Then, I will send these graphs to do the FFT transform just like the way we do to obtain the breathing frequency. Finally, three following standards will be applied to distinguish the humans' breathing waveform:

1. The desired frequency (the horizontal coordinate of the highest peak) obtained from the FFT result should be not smaller than 0.2 Hz and no bigger than 0.5 Hz.
2. The difference between the maximum value and minimum value in original waveform should not be smaller than 2 mm.
3. The maximum FFT value of the signal should not be smaller than 0.1.

Generally, a normal person's breathing is maintained at 12 - 30 times per minute. Therefore, its breathing frequency should be in the range 0.2 - 0.5 Hz. Also, as we observe in daily life, the chest undulation of adults while breathing should be at least 2 mm. So, it helps us to determine the lower limit of waveform. What's more, I find out that the smaller the value of FFT, the lower the probability that this waveform is a human respiratory waveform. After the judgement, my improved algorithm will output the correct breathing waveform or give us warning message to show that no humans' breathing waveform is detected.

3 Verification

3.1 Distance Detection

The y axis in Figure represents the signal amplitude at a given distance, while the x axis represents the distance from the radar board. The distance detection module functions effectively, according to the testing. The resolution is around 0.05 meters. Below shows that a person is 1.10 meters in front of the radar board comes from one of the test scenarios. The program predicts an amplitude of 1.07 meters, which is reasonably close to the actual value.

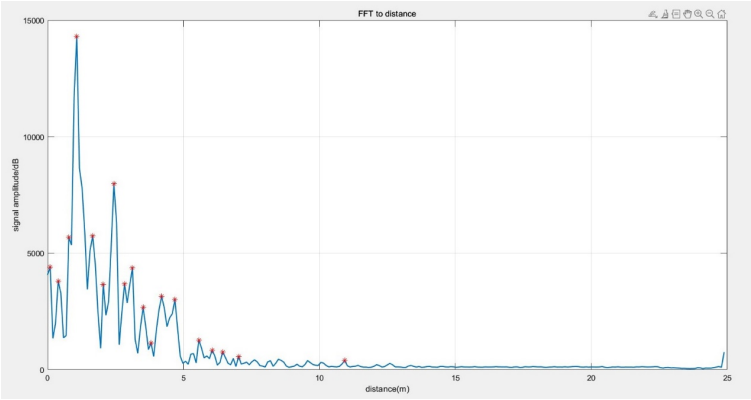

Figure 2: FFT for distance.

3.2 DACM algorithm

And we can see the result of phase in Fig.4 and Fig.3. This algorithm gives a good performance.

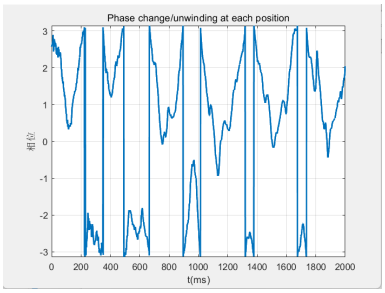

Figure 3: Phase change/original

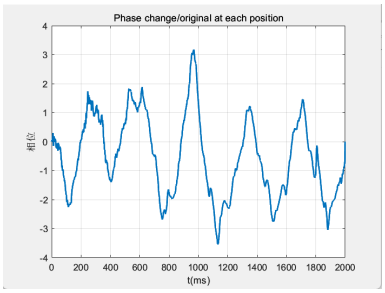

Figure 4: Phase change/unwrapping

3.3 GUI

By pressing the run button, the system's basic module will launch and provide a graph showing the movement of the chest as a result. The SG filter button and the n filter button

Figure 5: The GUI of our product.

are used to give the result after the corresponding filters. The GUI can be set to present the result figure separately or show the figure at the same page.

3.4 Breathing Frequency Estimation

We take a standard breathing waveform ($f_{breathing} \approx 0.3$) as input.

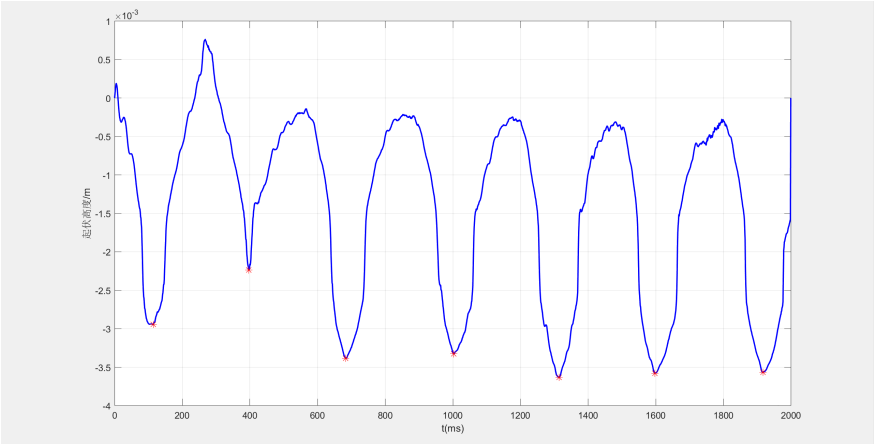

Figure 6: The breathing wave.

After we do the FFT transform, we get the result as shown below:

We could find out the breathing frequency is 0.35 which is the in the interval [0.1,0.5] and is close to our expectation value 0.3. The breathing frequency estimation is functional.

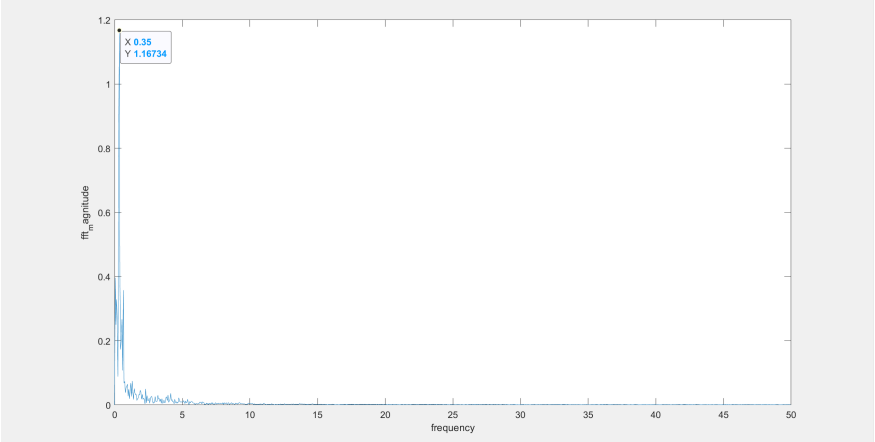

Figure 7: The figure after FFT transform.

3.5 Denoising

3.5.1 SG Filter

To test the filtering effect, we take as input a signal containing high frequency noise. Since the total number of sample points is 2000, we decide to add a shifting window that allows us to handle only 500 sample points at one time. Then, we set the highest order of our polynomial $p(n)$ to 9. Therefore, we build the matrix X^T as:

$$\begin{bmatrix} (-249)^0 & \cdots & (-1)^0 & 1 & 1^0 & \cdots & 250^0 \\ (-249)^1 & \cdots & (-1)^1 & 0 & 1^1 & \cdots & 250^1 \\ (-249)^2 & \cdots & (-1)^2 & 0 & 1^2 & \cdots & 250^2 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ (-249)^9 & \cdots & (-1)^9 & 0 & 1^9 & \cdots & 250^9 \end{bmatrix} \tag{15}$$

Also, to avoid the discontinuity condition between two adjacent windows, we design "patches" at the points: 500, 1000 and 1500. The length of patch is 100 and we filter the signal again at these points. The filtering result is shown as below and it's obvious to conclude that the Savitzky-Golay filter could eliminate the high frequency noise from original signal points without the analysis of characteristics in frequency domain.

3.5.2 Infinite Impulse Response(IIR) Filter

1) Set $R_p = 1dB$, $R_s = 60dB$, $W_p = 0.6Hz$, $W_s = 0.8Hz$. The reason why the passband frequency starts from 0 is already mentioned above in the design details.
2) Use Fast Fourier Transform(FFT) to get the frequency domain of the original signal. Exploit the designed IIR Elisp filter to denoise.

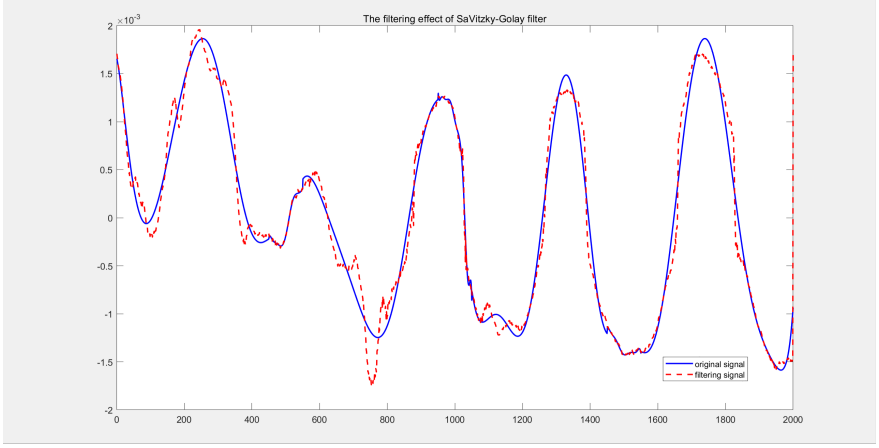

Figure 8: The figure after SG filter. The red dot line is the original breathing wave and the white line is the wave after filtering.

3) Compare the graph of the filtered signal and the original signal to measure the performance of the IIR filter.

4) Repeat the step 1-3 and test the performance of different parameters: $W_p = 0.7Hz$, $W_p = 0.4Hz$,

4) Analyze the performance of IIR filter with different parameters in Fig.9 and Fig.10.

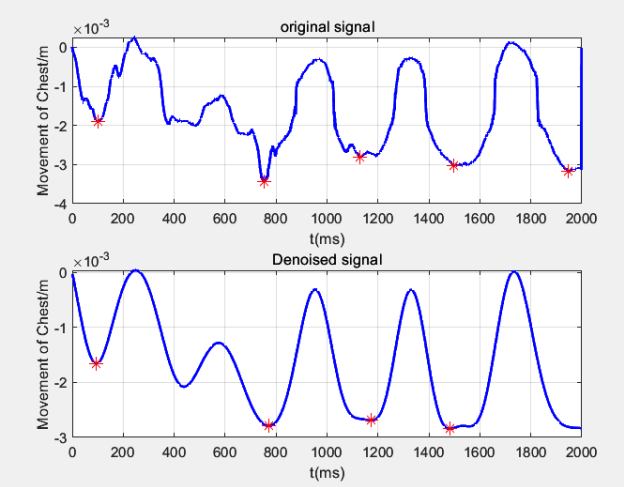

Figure 9: This the result of Denoised signal with $W_p = 0.6Hz$ and original data

3.5.3 GAN Denosing Method

The system should be able to transform the image between the source domain and the target domain. When we input a noisy waveform image, the GAN should be able to output a clean waveform image. The following are the result.

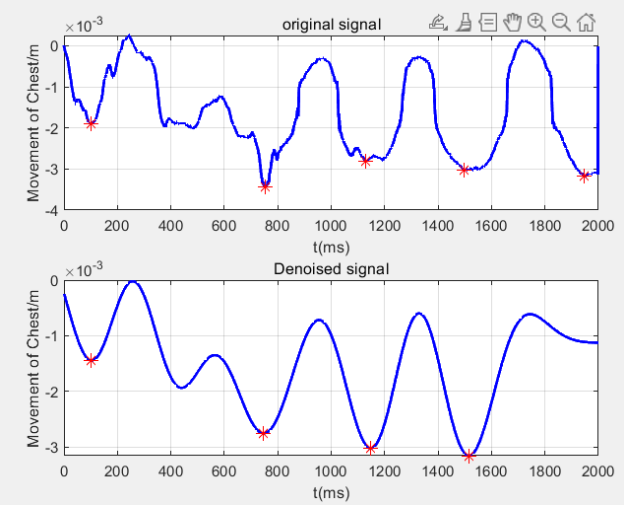

Figure 10: This the result of Denoised signal with $W_p = 0.4Hz$ and original data

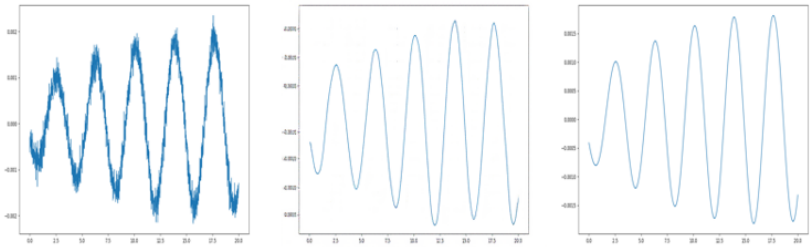

Figure 11: The first image is our simulated noisy waveform. The second image is our simulated clean waveform. The third image is our generated clean waveform.

We can see though it can learn shape well, the performance of preserving axis information is terrible because they are mostly number with meaning rather than the shape. It can be solved by cropping the axis before input the image and later fixing it. But my partner has already worked out other filter which can work better, so we decided not to continue working on GAN, and it can be done in the future if continuing.

3.6 Reminder

3.6.1 Sending Reminder via Email

1) Apply for one 163 Email account as the central server email.
2) Set one participant's email as the receiver email. And set the email content: "There is a child left in the car!!! Please be Careful!!!".
3) Run function: **sendmail(receiver, mailtitle, mailcontent)**
4) Check the receiver email whether it receives the reminder or not.

3.6.2 Sending Reminder via Phone message

1) Apply for the free phone number using my phone number as a user at Twilio.com.
2) Set the Account ID and Account-token gotten from Twilio.com to be the central server.
3) Install the twilio to local personal computer. , to send the message.
4) Run the function:
client.messages.create(to=mynumber, from=twilioNumber, body=message)
5) Check the receiver phone whether receives the reminder or not.

3.6.3 Voice Reminder

1) Use the computer voice recorder to generate two versions of .wav broadcasting file.
2) Run the function **soundsc(song,fs)** to play these two files.
2) Check whether voice files can be normally broadcast.

3.6.4 Automatic Breathing Detection

To test our automatic breathing detection algorithm, we should collect the bin file from experiment first. During the experiment, we let the tester to breathe in a varying rate. Then, I take the whole bin file as input and my improved algorithm should output the proper breathing waveform.

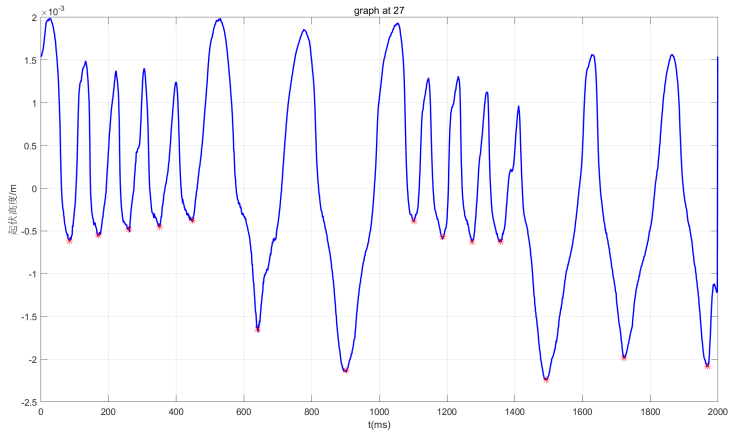

Figure 12: The output waveform from improved algorithm

From Fig.12, we could clearly tell that this is a human's breathing waveform and his breathing rate goes through a process of alternating fast and slow. The automatic breathing detection truly save us a lot of time to find out the desired waveform.

4 Cost

Each team member's labor is estimated to be $\$7$/hour, 10 hours/week for three people. We consider the project would last for 9 weeks, so the cost of member's labor would be:

$$4 * \frac{\$7}{hr} * \frac{10hr}{wk} * 9wk * 2.5 = \$6300$$

Our estimated costs for hardware are listed as the tabular below:

Part	Item(manufacturer)	Price per unit	Amount	Total Price
RF System	IWR6843ISK-ODS mmWave ODS(Texas Instruments)	$175	1	$175
	Internet Cable(Amazon)	$5.75	1	$5.75
	USB Type C Cable(Amazon)	$6.49	2	$12.98
	5V 1A Barrel Connector(PERFEIDY)	$13.99	1	$13.99
Control System	2021 HP Stream 14" HD SVA Laptop Computer(HP)	$299	1	$299
Power System	Portable Power Station(MARBERO)	$99.99	1	$99.99
Grand Total				$606.71

Table 1: The estimated hardware cost for our project

Another part of the cost would be the bonuses and reception expense for volunteers of our data collection. Estimated cost for one volunteer is $\$15$, and there will be approximately 2 people each for 10 different age periods. So the total expected cost would be:

$$\frac{\$15}{person} * \frac{2persons}{group} * 10groups = \$300$$

In a nutshell, the approximate grand total cost for all our project would be $\$7206.71$

5 Result

Below are experiment results that meet the High-level requirements. The y-axis of the graphs is the movement of the chest of the test participants in meters. The x-axis is the time in seconds.

5.1 Test person with Varying Breathing Frequency

In this experiment, the participant's breathing frequency is from fast to slow and then repeat the process. Our system gives the result that match the truth well as shown in

Fig.13.

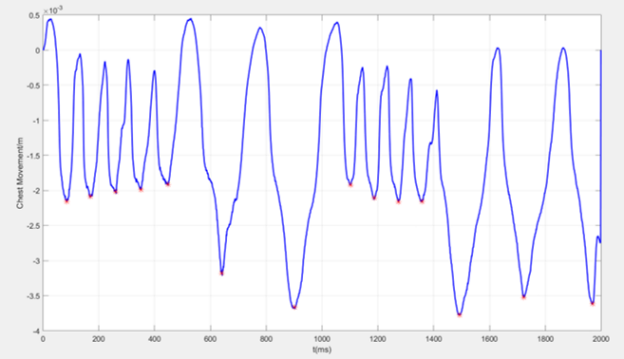

Figure 13: Test person with Varying Breathing Frequency

5.2 Test Multiple People

In this experiment, two subjects were seated 1m in Fig.15and 1.2m in Fig.14away from the radar. Our project succeeded in finding two breathing positions in one round of detection.

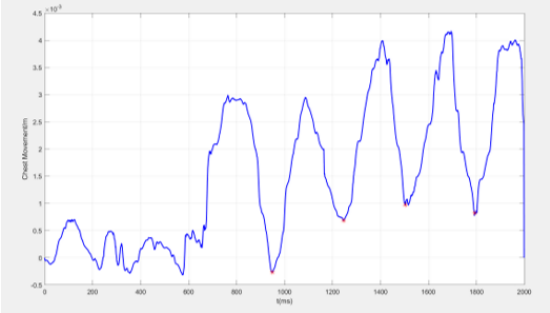

Figure 14: Test person with distance 1.2m

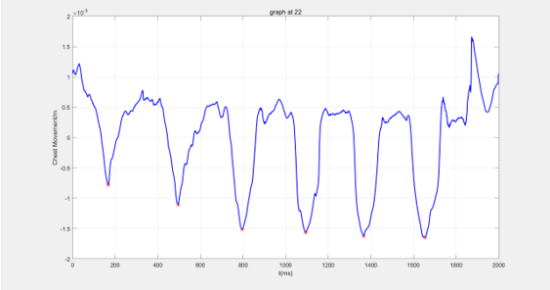

Figure 15: Test person with distance 1m

5.3 Test children participants

Additionally, we have conducted the lab with children participants as shown in Fig.**??**. And we successfully detect the presence of children. The waveform shown in Fig.17 also gives a good performance. After analyzing the waveform, we can see that the chest movement is less than adults in Fig.15.

Figure 16: Children participants

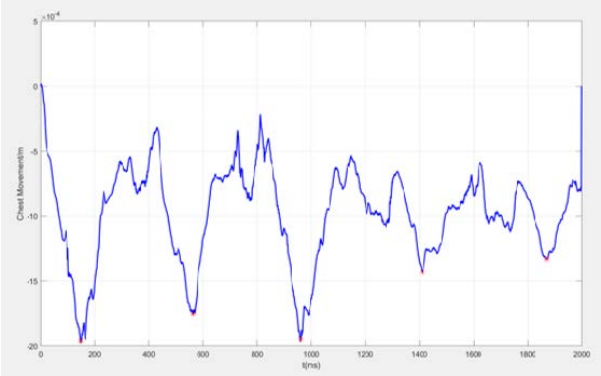

Figure 17: Test children

5.4 Test in the car

We have done the lab in the car to simulate the real application scene. And we got the breathing result and then use our filter to filter the noise and get a real breathing wave as shown in Fig.18

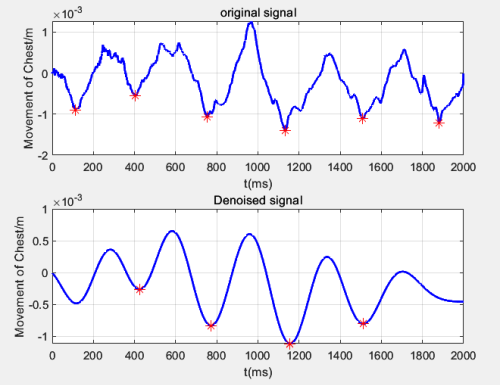

Figure 18: The breathing result in the car

5.5 Reminder functions

We also test the function, sending emails and sending phone messages. Here are the successful result as shown in Fig.19 and Fig.20.

Figure 19: Result of receiving email

Figure 20: Result of receiving message

6 Conclusion

6.1 Accomplishments

We have done a good job in manipulating the radar board and studying the algorithm of FFT. By exploiting IIR filter, DACM algorithm, and other tools, we can get relatively good data on breathing.

We successfully test our system under different conditions:

1) Test the system with nothing in front of the radar board. And we get the safe result without warning.

2) Test the system with one person in front of the radar board. And the system successfully detects the person and sends reminders.

3) Test the system with one person in front of the radar board but with varying breathing frequencies. And the system successfully detects the person and his changing breathing frequency and sends reminders. Its result is shown in Fig.13.

6.2 Uncertainties

Our radar board is easy to accumulate heat which decreases the accuracy of the board detection. So we need to control the working time of the radar board to be within 10 minutes for one lab.

And the connection of radar board and settings of mmwave sofeware offen crashes, so we need to check the connection carefully each time.

And our system requires an environment with no interruption. Otherwise, the radar board cannot give an accurate result.

6.3 Plans for remaining work

We should focus more on the older people. Because out system can detect the breathing frequency from 0.2Hz to 0.5Hz. But We cannot deal with the large movement of people. If them move their body, our system cannot give a good breathing wave. So we can use our system to detect older people whether they are asleep or not. And we could also work on tackling the problem of large movements of people.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[2] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[3] "Windows on r." (), [Online]. Available: https://worproject.com/downloads (visited on 04/09/2023).

[4] TI. ""IWR6843 datasheet"." (2021), [Online]. Available: https://www.ti.com/document-viewer/iwr6843/datasheet (visited on 03/15/2023).

[5] chrisda. ""Enable or disable POP3 or IMAP4 access to mailboxes in Exchange Server"." (), [Online]. Available: https://learn.microsoft.com/en-us/exchange/clients/pop3-and-imap4/configure-mailbox-access?view=exchserver-2019 (visited on 05/12/2023).

[6] Twilio. ""Communication APIs for SMS, Voice, Video Authentication"." (), [Online]. Available: https://www.twilio.com (visited on 05/12/2023).

[7] Wiki. ""Elliptic filter"." (2023), [Online]. Available: https://en.wikipedia.org/wiki/Elliptic_filter (visited on 05/12/2023).

[8] Mathworks. ""IIR Filter Design"." (), [Online]. Available: https://ww2.mathworks.cn/help/signal/ug/iir-filter-design.html (visited on 05/12/2023).