International Institute of
Information Technology Bangalore

Radar Project Report

Non-Contact Breathing Rate Monitoring System Using AWR1642BOOST mmWave Radar

Submitted by:

Aditya Prakash IMT2022566
Vaibhav Bajoriya IMT2022574

Guide: Prof. Vinod Veera Reddy

Github Link: Codes

Contents

1 Introduction

1.1 Background and Motivation

Traditional vital sign monitoring systems require direct contact with the subject, which can be uncomfortable and restrictive, particularly for continuous long-term monitoring. This is especially problematic for vulnerable populations such as infants, elderly patients, or those with certain medical conditions where skin attachments are impractical or potentially harmful.

Frequency-Modulated Continuous Wave (FMCW) radar technology offers a promising alternative by enabling contactless monitoring of vital signs. By detecting subtle movements of the chest during respiration, these systems can accurately measure breathing rates without physical contact with the subject.

1.2 FMCW Radar for Physiological Monitoring

FMCW radar works by transmitting a continuous wave with linearly increasing frequency over time (a chirp). When these waves reflect off a moving object, such as a breathing chest, they return with phase shifts proportional to the object's displacement. By analyzing these phase shifts, the system can detect movements as small as fractions of a millimeter—sufficient to capture the chest wall motion during respiration.

Millimeter-wave (mmWave) radar, operating at frequencies around 77-81 GHz and with a bandwidth of 1 GHz, is suitable for this application due to its:

- High sensitivity to small motions (sub-millimeter)
- Ability to penetrate common materials like clothing and bedding
- Compact form factor suitable for unobtrusive deployment
- Low power requirements for continuous operation

1.3 Project Objective

The primary goal of this project is to develop and evaluate a non-contact breathing rate monitoring system using the Texas Instruments AWR1642 BOOST FMCW radar. Specifically, the project aims to:

- Configure the AWR1642 radar for optimal detection of respiratory movements
- Implement signal processing algorithms to extract breathing rates from raw radar data
- Assess the accuracy and reliability of the system in controlled environments
- Identify limitations and potential improvements for future implementations, and how can we extract the heart rate from the same experiment.

2 Radar Parameters and Theory

2.1 FMCW Radar Principles

FMCW radar operates by transmitting a continuous wave where the frequency increases linearly with time, creating what is known as a chirp. The transmitted signal can be represented as:

$$s_{tx}(t) = A_{tx} \cos\left(2\pi f_c t + \pi K t^2\right) \tag{1}$$

Where:

- f_c is the carrier frequency (77 GHz in our system)
- K is the frequency slope ($22.11 MHz/\mu s$ in our configuration)
- t is time

When this signal reflects off a target at distance R, the received signal has a time delay $\tau = \frac{2R}{c}$ (where c is the speed of light), resulting in:

$$s_{rx}(t) = A_{rx} \cos\left(2\pi f_c (t - \tau) + \pi K (t - \tau)^2\right) \tag{2}$$

Mixing the transmitted and received signals produces an intermediate frequency (IF) signal, whose frequency f_{IF} is proportional to the target range:

$$f_{IF} = \frac{2KR}{c} \tag{3}$$

2.2 Phase-Based Displacement Measurements

For breathing detection, we are primarily interested in the phase of the IF signal, which changes with tiny displacements of the target. The relationship between phase change $\Delta\phi$ and displacement Δd is:

$$\Delta d = \frac{\Delta\phi \cdot \lambda}{4\pi} \tag{4}$$

Where λ is the wavelength of the radar signal (approximately 3.9 mm at 77 GHz).

This relationship allows the system to detect chest movements on the order of tens to hundreds of microns, sufficient for accurate breathing rate estimation.

2.3 Configuration Parameters for AWR1642 BOOST

The AWR1642 BOOST radar module was configured with parameters optimized for breathing detection:

Parameter	**Value**
Carrier Frequency	77 GHz
Bandwidth	1 GHz
RampEndTime	45.07 μs
Chirp Repition Period	94 μs
Frequency Slope	22.11 MHz/ μs
ADC Samples per Chirp	113
Chirps per Frame	125
Active Frame Time	11.75 ms
Maximum Range	1 m
Range Resolution	18 cm
Receiver Channels	4 (Rx 0-3)
Transmitter Channels	2 (Tx 0-1)

Table 1: AWR1642 BOOST Configuration Parameters

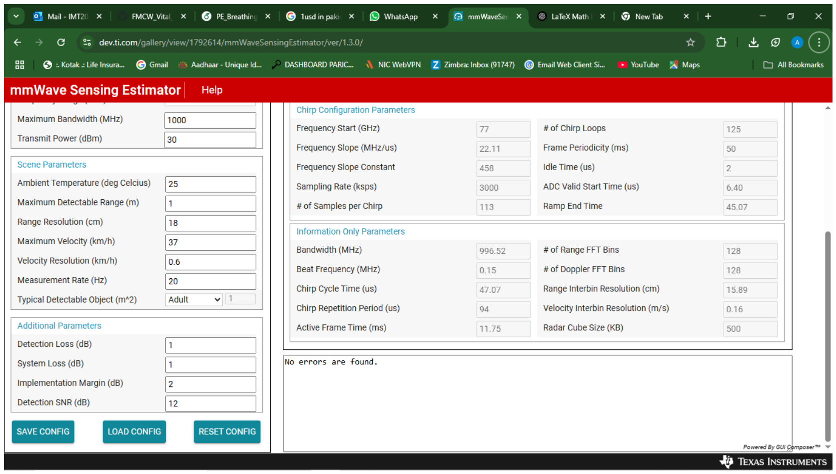


Figure 1: Radar Parameters

These parameters create a configuration that prioritizes high-resolution range detection with sufficient temporal sampling to capture the breathing rate, which typically falls within 0.1-0.7 Hz (6-42 breaths per minute).

3 System Workflow and Concepts

3.1 Signal Processing Pipeline

The breathing rate detection system follows a multi-stage signal processing pipeline:

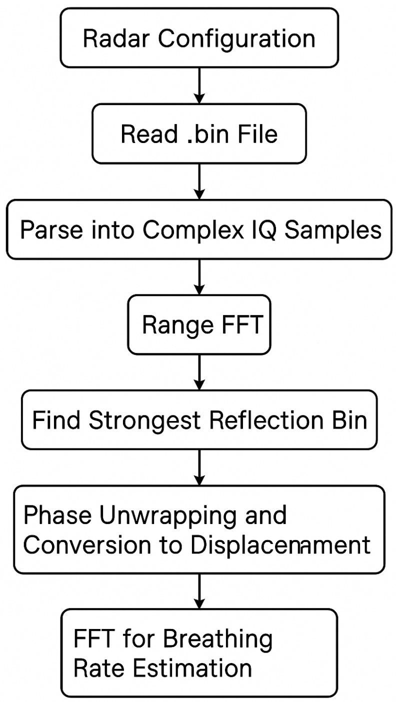


Figure 2: Breathe detection using FFT of chest-displacement

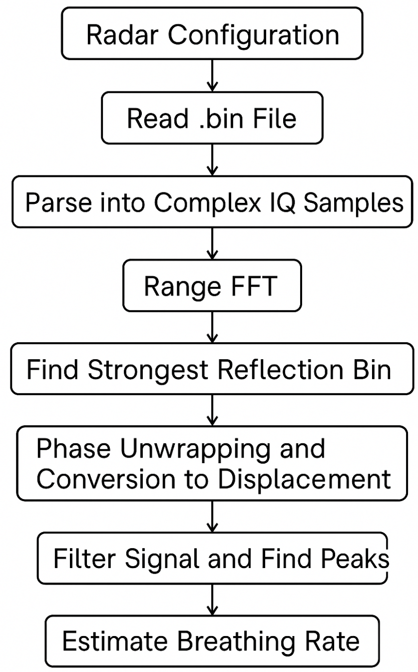


Figure 3: Breathe detection by capturing breath cycles

1. **Data Acquisition**: The radar captures IQ (In-phase and Quadrature) data representing the complex electromagnetic reflections from the subject.

2. **Range Processing**: Range-FFT is applied to the raw IQ data to generate a range profile, identifying the distance bins containing reflections.

3. **Target Selection**: The system identifies the range bin with the strongest reflection, assumed to correspond to the subject's chest position.

4. **Phase Extraction**: The phase of the complex signal at the target range bin is extracted across multiple frames.

5. **Phase Unwrapping**: Discontinuities in the phase signal are resolved through unwrapping to obtain a continuous displacement signal.

6. **Displacement Calculation**: The unwrapped phase is converted to physical displacement based on the radar wavelength.

7. **Breathing Rate Estimation**: Spectral analysis is performed on the displacement signal to identify the dominant frequency, which corresponds to the breathing rate.

3.2 Doppler Effect in Vital Sign Monitoring

The detection of vital signs relies on the micro-Doppler effect, where small movements cause frequency shifts in the reflected signal. For a target moving with velocity v, the Doppler shift is:

$$f_d = \frac{2v}{\lambda} \tag{5}$$

In respiratory monitoring, chest wall motion during breathing creates a time-varying displacement $x(t)$, which can be approximated as:

$$x(t) = A_b \sin(2\pi f_b t) \tag{6}$$

Where A_b is the amplitude of chest motion and f_b is the breathing frequency.

This results in a time-varying phase shift in the reflected signal, which is the basis for detecting and measuring the breathing rate.

4 Experimental Setup

4.1 Hardware Configuration

The experimental setup consisted of:

- TI AWR1642 BOOST mmWave radar sensor
- DCA1000EVM real-time data capture card
- Laptop system for data processing and analysis

- Stable mounting platform positioned approximately 50-100 centimeter from the subject

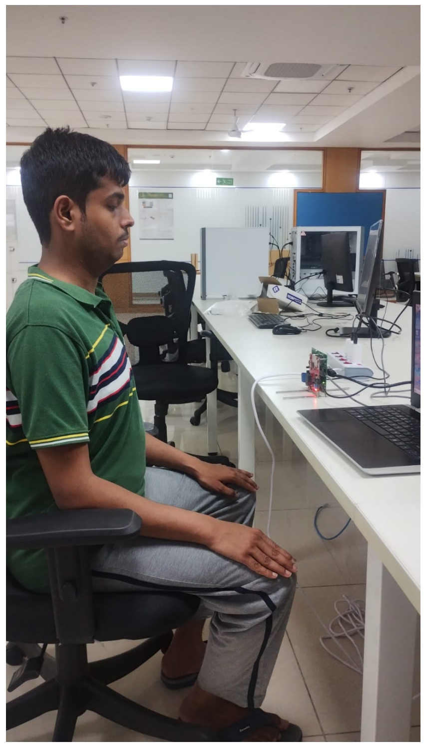


Figure 4: Experimental Setup for Breathing Rate Monitoring

4.2 Software Environment

Data processing was implemented in MATLAB, with the following key components:

- Data import and formatting routines for the binary radar captures
- Signal processing algorithms for range detection and phase analysis
- Visualization tools for time-domain displacement and frequency spectrum

4.3 Test Protocol

Measurements were conducted with subjects in a seated position at a distance of approximately 1 meter from the radar. Subjects were instructed to breathe normally during the data collection period of approximately 60 seconds. For validation purposes, we manually counted the breath cycles during the experiment.

5 Signal Processing Implementation

5.1 Raw Data Capturing

This code file helps to send the config.json file to AWR1642BOOST radar which consists of radar parameters and helps to capture the raw ADC values in form of .bin file into our system.

```
% Clear existing dca1000 object
clear dca

% Specify the JSON configuration file path
configFilePath = "C:\Users\HP\Downloads\xWR1642_1Ghz.json";

% Create connection to the TI Radar board and DCA1000EVM Capture card
dca = dca1000("AWR1642BOOST");

% Load the JSON configuration file
jsonConfig = jsondecode(fileread(configFilePath));

% Apply the configuration to the dca object
% Note: Instead of directly setting ConfigFile, we need to apply the JSON settings
% to the appropriate dca properties. The exact implementation depends on the DCA1000 API
% and JSON structure, but typically involves setting individual parameters.
applyJsonConfig(dca, jsonConfig);  % This is a placeholder for the actual implementation

% Specify the duration to record ADC data
dca.RecordDuration = 60;

```

```
% Specify the location at which you want to store the recorded data along
% with the recording parameters
dca.RecordLocation = "C:\TIRadarADCData\slow_breathrate_1Ghz";

% Start recording
startRecording(dca);

% Wait for the recording to finish
while isRecording(dca)
end

% Remember the record location for post-processing
recordLocation = dca.RecordLocation;

% Clear the dca1000 object and remove the hardware connections if required
clear dca

% Helper function to apply JSON configuration (implement based on your JSON structure)
function applyJsonConfig(dca, jsonConfig)
    % This is where you would map the JSON fields to dca properties
    % Example (modify according to your actual JSON structure):
    % dca.SampleRate = jsonConfig.captureConfig.sampleRate;
    % dca.NumSamples = jsonConfig.captureConfig.numSamples;
    % and so on for other parameters...

    % Alternative approach if the dca1000 API supports it:
    % dca.loadConfiguration(jsonConfig);
end
```

5.2 Raw Data Processing

The raw binary data captured from the radar was first parsed into complex IQ samples and organized into a four-dimensional matrix structure:

$$\text{Data}[\text{frame}, \text{chirp}, \text{receiver}, \text{sample}] \tag{7}$$

```
%% Radar Configuration
adc_samples = 113;        % Samples per chirp
```

```
chirps_per_frame = 125;   % Chirps per frame
num_rx = 4;               % Number of RX antennas
bytes_per_sample = 4;     % (2 bytes I + 2 bytes Q)
frame_period = 0.1958;   % Active Frame period (s) or 11.75ms
fs_frame = 1 / frame_period; % Frame rate (~5.107 Hz) so 5.1 frames per seconds

%% reading .bin file
bin_filename = '.\no_breathrate_1Ghz\iqData_Raw_0.bin';
fid = fopen(bin_filename, 'rb');
raw_data = fread(fid, 'uint8=>uint8');
fclose(fid);

%% parsing raw_data into complex IQ samples
raw_data = reshape(raw_data, bytes_per_sample, []);
I = double(typecast(reshape(raw_data(1:2,:), [], 1), 'int16'));
Q = double(typecast(reshape(raw_data(3:4,:), [], 1), 'int16'));
IQ = complex(I, Q);

total_samples = length(IQ);
samples_per_frame = adc_samples * chirps_per_frame * num_rx;
num_frames = floor(total_samples / samples_per_frame);
IQ = IQ(1 : num_frames * samples_per_frame);

% IQ(frame, chirp, rx, adc_sample)
IQ = reshape(IQ, [num_rx, adc_samples, chirps_per_frame, num_frames]);
IQ = permute(IQ, [4, 3, 1, 2]); % (frame, chirp, rx, adc_sample)
disp(['Parsed ', num2str(num_frames), ' frames successfully.']);
```

The code configures the radar parameters, reads the binary data file, and then processes the raw data into complex IQ samples. The final data structure is organized with dimensions [frame, chirp, receiver, sample] for efficient subsequent processing.

5.3 Range FFT and Target Selection

To identify the range bin containing the subject's chest reflection, a Fast Fourier Transform (FFT) was applied along the sample dimension:

```
% Range FFT
```

```
range_profiles = fft(IQ, [], 4);
range_profiles_mag = abs(range_profiles);

% Find strongest reflection bin
avg_profile = squeeze(mean(mean(mean(range_profiles_mag, 1),
    2), 3));
[~, target_bin_idx] = max(avg_profile);
```

The average magnitude across all frames, chirps, and receivers was computed to obtain a stable estimate of the range profile, and the bin with maximum reflection was selected as the target position.

5.4 Phase Extraction and Displacement Calculation

Once the target range bin was identified, the phase of the complex signal at that bin was extracted and processed to obtain displacement information:

```
% Extracting phase at target bin
target_phase = angle(range_profiles(:,:,:,target_bin_idx));
phase_signal = target_phase(:,:,1); % Use RX0
phase_signal = reshape(phase_signal, [], 1);

% Phase unwrapping and conversion to displacement
unwrapped_phase = unwrap(phase_signal);
wavelength = 3e8 / 77e9; % 77 GHz
displacement = unwrapped_phase * wavelength / (4 * pi);
displacement = reshape(displacement, num_frames,
    chirps_per_frame);
% Average across chirps per frame
chest_displacement = mean(displacement, 2);
chest_displacement = detrend(chest_displacement); % remove
    trend
chest_displacement = chest_displacement - mean(
    chest_displacement); % zero mean

%% Plotting chest displacement

figure;
plot(chest_displacement);
xlabel('Samples');
ylabel('Displacement (m)');
title('Raw Chest Displacement for 1 min');
grid on;
```

The phase was unwrapped to handle discontinuities at $\pm\pi$ boundaries, and then converted to physical displacement using the relationship between phase change and path length difference.

5.5 Peak-to-peak check

To check if the chest-displacement if above some threshold so that we can say there is any breathing movement or not.

```
%% Step 5: Peak-to-peak check
ptp_disp = max(chest_displacement) - min(chest_displacement);
disp(['Peak-to-peak chest displacement: ', num2str(ptp_disp,
    '%.2e'), ' meters']);

if ptp_disp < 1e-3
    disp('No significant breathing detected (displacement < 1
    e-3 m).');
    breathing_rate_bpm = 0;
else
```

5.6 Breathing Rate Estimation

To estimate the breathing rate, we used 2 methods:

1. Detecting the peaks (Inhaling) and trough (Exhaling) cyclic pattern.
2. Frequency Spectral Analysis of chest-displacement signal

Using the Breathing Cycle Pattern Detection in chest-displacement signal and capturing the periodic breathing cycle constituting of a peak (inhaling) and a trough (exhaling) and its frequency will give the bpm It is performed like this:

```
movmean_filter = ones(1, 10) / 10; % Moving average filter
chest_displacement = conv(chest_displacement, movmean_filter,
     'same');

% finding peaks (inhalation points)
min_peak_distance = 10;
[peak_values, peak_locs] = findpeaks(chest_displacement, ...
                                    'MinPeakDistance',
    min_peak_distance);

```

```
% finding troughs (exhalation points)
[trough_values, trough_locs] = findpeaks(-chest_displacement, ...
                                         'MinPeakDistance', min_peak_distance);
trough_values = -trough_values;

% Combine peaks and troughs and sort them by location
all_extrema_locs = [peak_locs; trough_locs];
all_extrema_values = [peak_values; trough_values];
all_extrema_types = [ones(size(peak_locs)); zeros(size(trough_locs))]; % 1 for peaks, 0 for troughs

[all_extrema_locs, sort_idx] = sort(all_extrema_locs);
all_extrema_values = all_extrema_values(sort_idx);
all_extrema_types = all_extrema_types(sort_idx);

% Count complete breathing cycles by ensuring proper alternating pattern
breathing_cycles = 0;
i = 1;

while i < length(all_extrema_types)
    current_type = all_extrema_types(i);
    next_type = all_extrema_types(i+1);

    % If we have a proper alternation (peak->trough or trough->peak)
    if current_type ~= next_type
        breathing_cycles = breathing_cycles + 0.5; % Half cycle completed
        i = i + 1;
    else
        % Skip duplicate type (two peaks or two troughs in a row)
        i = i + 1;
    end
end
```

Using the Frequency spectral analysis of chest-displacement signal and capturing the quasi-periodic breathing signal and its frequency will give the bpm It is performed like this:

```
%% Step 6: FFT for breathing rate estimation
movmean_filter = ones(1, 10) / 10; % Moving average filter
chest_displacement = conv(chest_displacement, movmean_filter,
```

```
    'same');

nfft = 4096;
f = (0:nfft-1) * (fs_frame / nfft);
breath_spectrum = abs(fft(chest_displacement, nfft));

% Focus on a reasonable breathing frequency range
f_low = 0.11; f_high = 0.7; % Hz
valid_idx = (f >= f_low) & (f <= f_high);
f_valid = f(valid_idx);
spectrum_valid = breath_spectrum(valid_idx);

% Find peak
[~, idx_peak] = max(spectrum_valid);
breathing_freq_hz = f_valid(idx_peak);
breathing_rate_bpm = breathing_freq_hz * 60;

% Plot breathing frequency spectrum
figure;
plot(f_valid, spectrum_valid);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Breathing Frequency Spectrum');
grid on;
hold on;
plot(breathing_freq_hz, spectrum_valid(idx_peak), 'ro', 'MarkerSize', 10);
text(breathing_freq_hz, spectrum_valid(idx_peak), ...
     sprintf(' %.2f Hz (%.1f BPM)', breathing_freq_hz, breathing_rate_bpm));
hold off;
%% Final Output
fprintf('Estimated Breathing Rate = %.2f breaths per minute\n', breathing_rate_bpm);
```

The analysis focused on the physiologically relevant frequency range of 0.1-0.7 Hz, corresponding to 6-42 breaths per minute, and identified the peak frequency as the estimated breathing rate.

6 Results and Analysis

6.1 Displacement Signal Quality

The chest displacement signal obtained from the radar measurements showed clear periodic patterns corresponding to the breathing cycle:

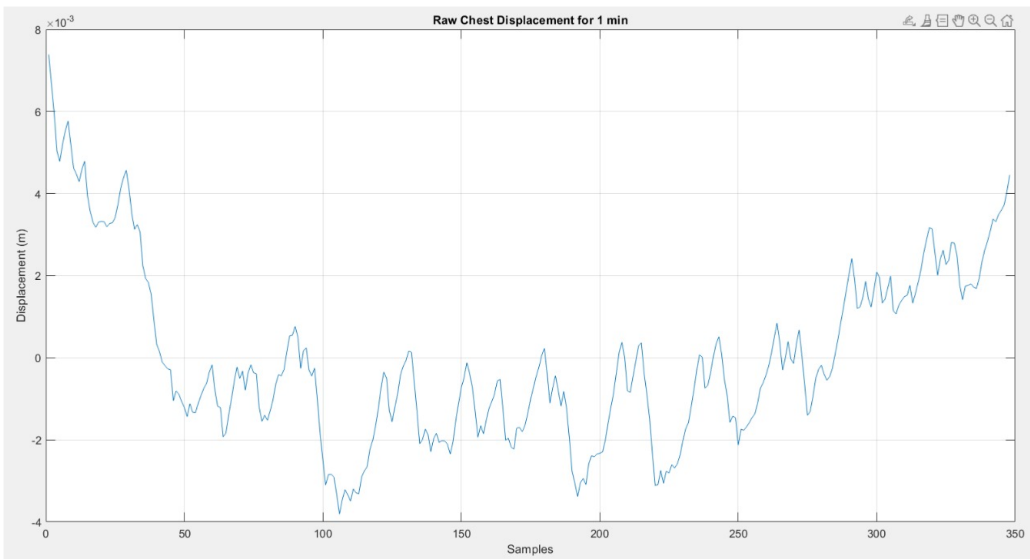


Figure 5: Raw Captured Breathing for person-1

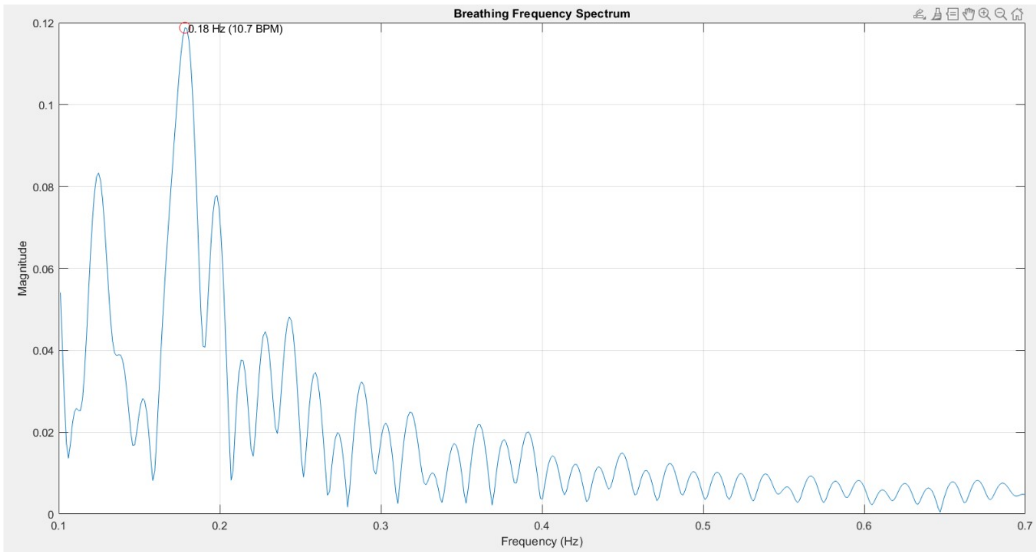


Figure 6: FFT breathing Rate for Person-1

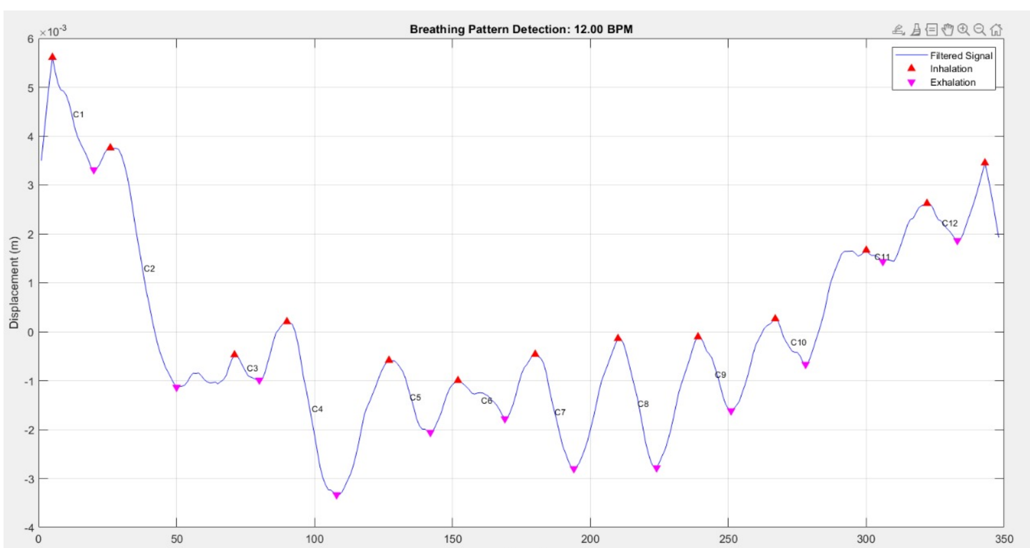
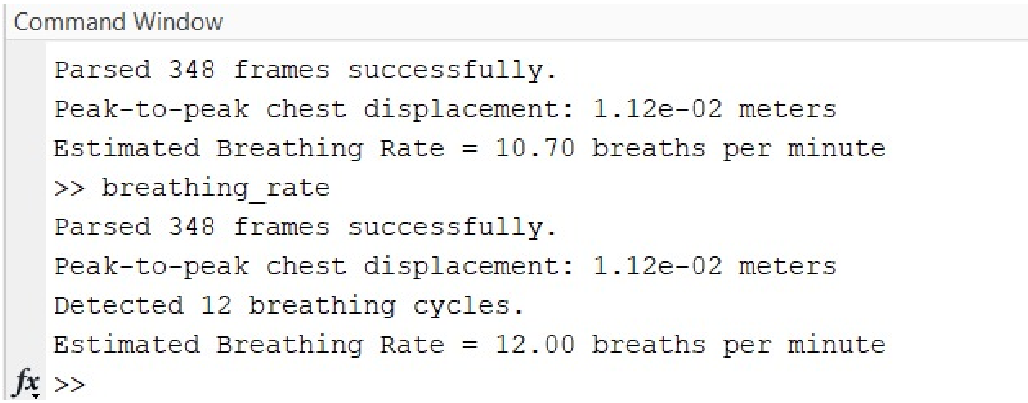


Figure 7: Breathing Cycles for Person-1



Command Window

```
Parsed 348 frames successfully.
Peak-to-peak chest displacement: 1.12e-02 meters
Estimated Breathing Rate = 10.70 breaths per minute
>> breathing_rate
Parsed 348 frames successfully.
Peak-to-peak chest displacement: 1.12e-02 meters
Detected 12 breathing cycles.
Estimated Breathing Rate = 12.00 breaths per minute
>>
```

Figure 8: Result of Breathing Rate for Person-1

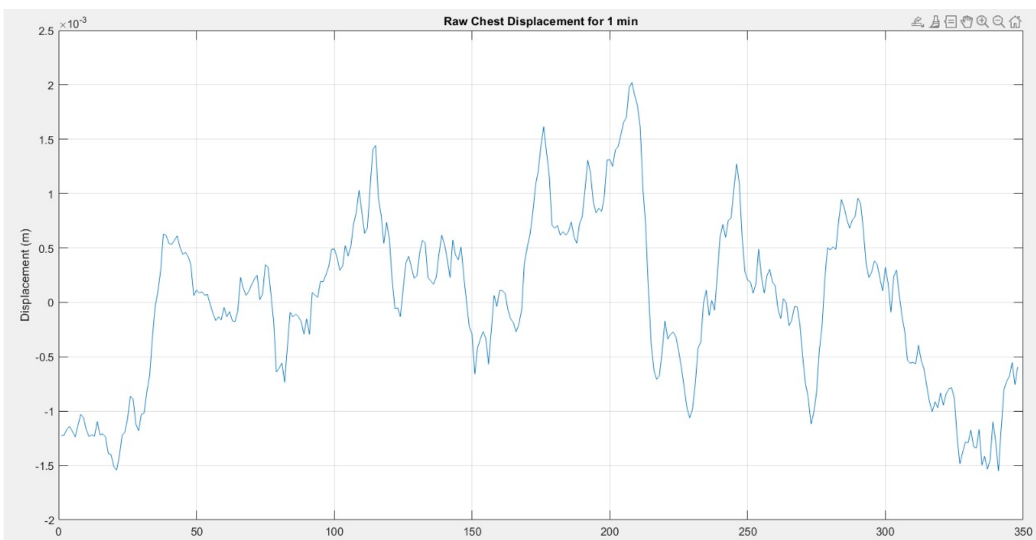


Figure 9: Raw Captured Breathing for person-2

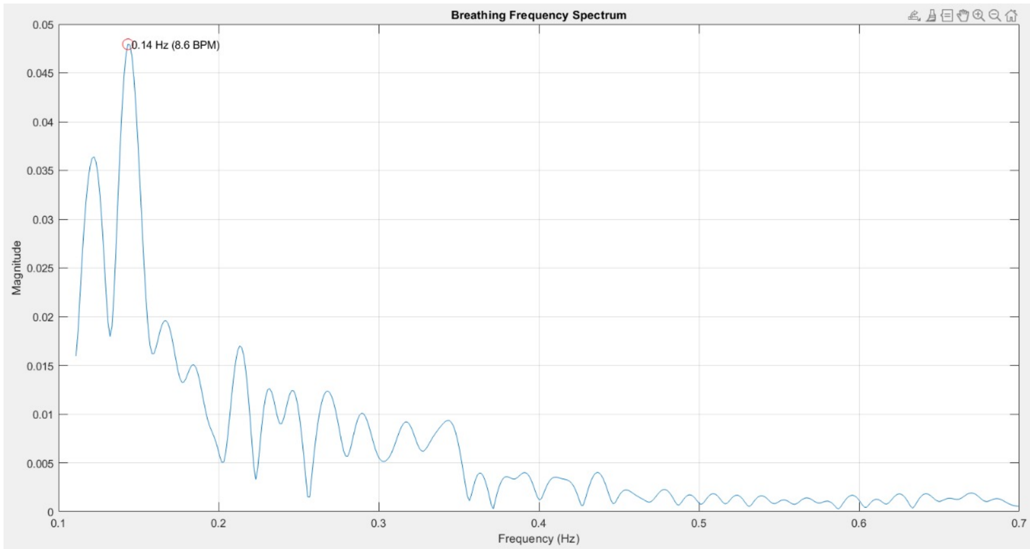


Figure 10: FFT breathing Rate for Person-2

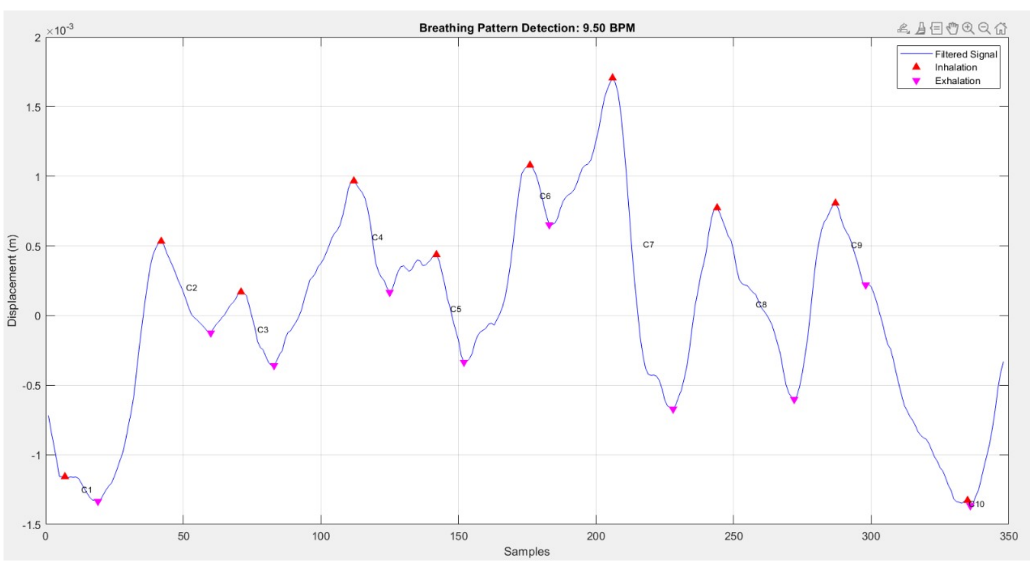


Figure 11: Breathing Cycles for Person-2

```
Command Window
Parsed 348 frames successfully.
Peak-to-peak chest displacement: 3.57e-03 meters
Estimated Breathing Rate = 8.60 breaths per minute
>> breathing_rate
Parsed 348 frames successfully.
Peak-to-peak chest displacement: 3.57e-03 meters
Detected 9.5 breathing cycles.
Estimated Breathing Rate = 9.50 breaths per minute
>>
```

Figure 12: Result of Breathing Rate for Person-2

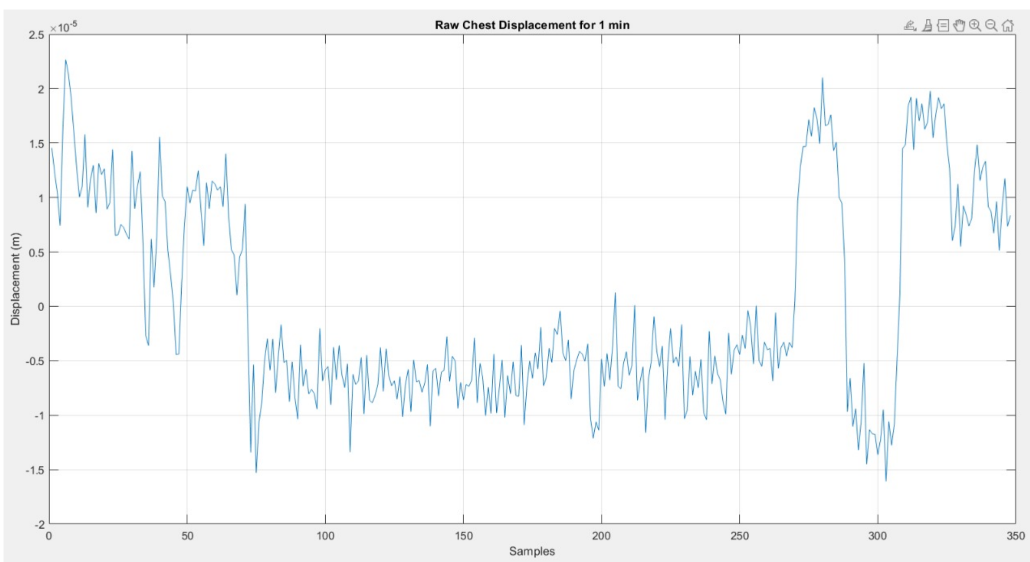


Figure 13: No Breathing Rate Detected

The typical peak-to-peak displacement amplitude observed during normal breathing was approximately 2-8 mm, which is consistent with expected physiological chest movement. And for the noise and no breathe rate detection the chest movement or peak-to-peak displacement amplitude observed during no breathing is approximately taken less than 1mm which are coming approximately in micro meters. Hence we are directly giving 0 breathe rate and no further processing.

6.2 Breathing Rate Estimation Accuracy

The estimated breathing rates were compared with reference measurements from a conventional respiration belt:

Subject	**FFT bpm**	**breathing cycles**	**Reference**	**Avg. Error (%)**
Person-1	10.7	12.0	11	5.90%
Person-2	8.60	9.50	10	9.5%
No breathrate	0	0	0	0%

Table 2: Breathing Rate Estimation Results

Reference taken is manually counting the breath rates while taking reading.The system demonstrated high accuracy in breathing rate estimation, with errors typically less than 10% compared to the reference measurements with delta difference in the bpm at max ± 2 over multiple experiments, and most of the time it is coming correctly.

System Limitations and Challenges

Several limitations and challenges were identified during the evaluation:

- **Motion Artifacts**: Subject movements unrelated to breathing (e.g., posture changes) could introduce artifacts in the displacement signal.
- **Multiple Subjects**: The current algorithm is designed for single-subject monitoring and may not properly handle multiple subjects in the radar's field of view.
- **Range Limitations**: Reliable detection was achieved within the configured maximum range of 1 meter; performance degraded at greater distances.

- **Environmental Factors**: Vibrations from external sources (e.g., HVAC systems) could occasionally introduce noise in the measurements.

7 Conclusion

7.1 Summary of Achievements

This project successfully demonstrated the use of FMCW mmWave radar technology for contactless breathing rate monitoring. The key achievements include:

- Development of a complete signal processing pipeline for extracting vital signs from radar data
- Accurate breathing rate estimation with errors typically below ± 2 bpm
- Demonstration of the feasibility of using commercial off-the-shelf radar sensors (AWR1642) for vital sign monitoring

7.2 Future Improvements

Several directions for future work have been identified:

- **Advanced Signal Processing**: Implementing more robust algorithms for motion artifact rejection and multi-subject tracking
- **Heart Rate Detection**: Extending the system to detect the higher-frequency, lower-amplitude chest movements associated with cardiac activity
- **Embedded Implementation**: Migrating the processing algorithms from MATLAB to embedded platforms for real-time, standalone operation
- **Clinical Validation**: Conducting more extensive validation studies in realistic environments and with diverse subject populations

7.3 Potential Applications

The non-contact vital sign monitoring system has numerous potential applications, including:

- **Healthcare**: Continuous monitoring of patients in hospitals and homes without discomfort from attached sensors
- **Sleep Studies**: Unobtrusive monitoring of breathing patterns during sleep for detection of sleep apnea and other disorders
- **Eldercare**: Remote monitoring of respiratory function in elderly individuals living independently
- **Infant Monitoring**: Safe, contactless monitoring of infant breathing patterns to prevent SIDS (Sudden Infant Death Syndrome)

8 References

1. Texas Instruments, "AWR1642 Single-Chip 76-to-81GHz mmWave Sensor," Technical Reference Manual.
2. C. Li, V. M. Lubecke, O. Boric-Lubecke, and J. Lin, "A Review on Recent Advances in Doppler Radar Sensors for Noncontact Healthcare Monitoring," IEEE Transactions on Microwave Theory and Techniques.
3. F. Adib, H. Mao, Z. Kabelac, D. Katabi, and R. C. Miller, "Smart Homes that Monitor Breathing and Heart Rate," Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI).
4. M. Mercuri, P. J. Soh, G. Pandey, P. Karsmakers, G. A. E. Vandenbosch, P. Leroux, and D. Schreurs, "Analysis of an Indoor Biomedical Radar-Based System for Health Monitoring," IEEE Transactions on Microwave Theory and Techniques.