Vital Signs Monitoring Using a mmWave Radar

Team Members:
Nolan Kosnic
Grant Hincher

Final Report
EECS430/SPACE431 Winter 2025

1. Motivation

Continuous monitoring of vital signs, like breathing rate and heart rate, is one of the many ways that patient health can be assessed. Monitoring these vitals, along with the purpose of basic monitoring of patients at hospitals, can be used to detect and possibly prevent fatal conditions, like obstructive sleep apnea, heart arrhythmia, or SIDS. At hospitals, these vitals are traditionally monitored continuously with the use of contact sensors, like an EKG machine. However, these sensors or straps on the chest area can cause skin irritation and limit the movement of the patient. This motivates the development of a FMCW (Frequency Modulated Continuous Wave) radar that can remotely monitor vital signs. The development and commercialization of such a product would serve as a non-invasive alternative for vitals sensing.

2. Background/Concepts

In order to successfully read vital signs with our radar, we first must understand the existing methodologies, and understand the advantages of mmWave frequencies over others for this application. Then, we must learn the operating principles of traditional FMCW radar, by understanding what data is actually being outputted, how range can be estimated from this data, and how the magnitude and phase of this signal can be interpreted. This information can be used to help us select the appropriate radar from Texas Instruments for this application.

Notable RF Concepts

While doing literature review for this project, we encountered various radar choices for vital signs monitoring. First, we found literature discussing the attempts at using a microwave radar for this task[1]; however we recognized this as an outdated paper, with technology having since progressed to mmWave precision. There was discussion in additional papers comparing the various signals transmitted and their benefits and drawbacks[2]. It discussed type continuous-wave (CW), pulsed, frequency-modulated continuous-wave (FMCW) and stepped-frequency continuous-wave (SFCW) transmissions. However, in the rest of the literature FMCW radars are the most commonly used, due to their ability to provide both range and doppler information, being low cost and compact as compared to other radars[3], and also being able to distinguish reflections from different ranges due to high propagation attenuation, allowing for detecting multiple targets[4]. It is for these reasons that we elected to purchase a FMCW mmwWave radar for the project.

Ideally, wide bandwidth would also be present in the radar because it provides great range resolution (wider chirps provide more frequency discrepancy), and several receive antennas would be desired to achieve high angular resolution. These desired characteristics are what will guide our radar selection.

To understand the operating principles behind FMCW radar, a valuable resource for this is a presentation[5] put together by TI, and we will elaborate on our findings from this presentation below.

IF Frequency

What most FMCW radars do is send out a chirp, which is defined as a radio wave with a sweeping frequency (in our radar, a chirp would sweep from 60 GHz to 63.6 GHz). The synthesizer within the radar produces a chirp, which is transmitted by the TX antenna (only one is utilized on our radar). This chirp reflects off of an object, and is received by the RX antenna. Lastly, at the receive end, the receive signal is subtracted by the transmitted signal by the mixer. Because the RX-signal is just a delayed version of the TX-signal, and this signal is linearly sweeping in frequency, the frequency spectrum at the output of the mixer (the difference in frequency between TX and RX) can be used to determine the various distances the signal traveled before returning. In other words, we can use this output at the mixer (called the IF signal) to determine the range of the object it reflected off of.

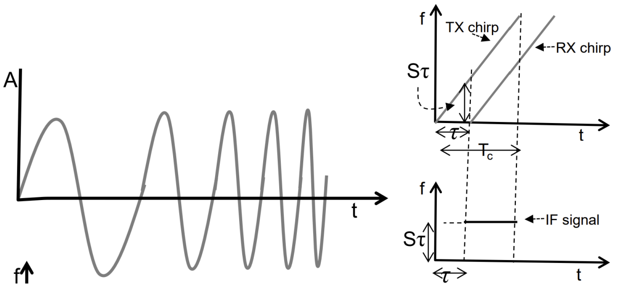


Figure 1: A time representation of a chirp (left)[5] and a frequency representation of the TX and RX signal (right)[5]

We obtain the frequency spectrum of the IF signal by doing a Fast Fourier Transform (FFT) across each chirp we receive, called a range FFT. We can correlate a sample bin with a beat frequency using the following equation involving ADC sampling rate, and number of ADC samples per chirp:

$$f_b = sample\ bin\ *\ \frac{ADC\ sampling\ rate}{ADC\ samples\ per\ chirp}$$

This beat frequency can then be used to solve for the range of the target with the following equation which incorporates the slope of the frequency sweep:

$$range\ =\ \frac{c * f_b}{2 * slope}$$

Phase of IF Signal

Once we know which range corresponds to the location of the chest, we can look at this particular range bin across each chirp and use their differences to obtain vibrational frequencies of the chest. At the mmWave level, the subtle millimeter movements of the chest (0.1-1.2 cm for breathing, 0.01 - 0.2 mm for heartbeat[6]) are not discernable in the frequency spectrum. However, the phase of the IF signal is very sensitive to small changes in object range. For example, if the chest were to move forward by a distance of 1 mm (for a 60 GHz radar this corresponds to $\lambda/5$), the frequency spectrum would not change, but the phase of the IF signal would change by $\Delta\Phi = \frac{4\pi\Delta d}{\lambda} = \frac{4\pi}{5} = 144°$. Therefore, the time evolution of this phase can be used to estimate both the amplitude and periodicity of the vibration at the desired range. This time, it would require another FFT being taken of the phase data at this desired range, called a vibration FFT.

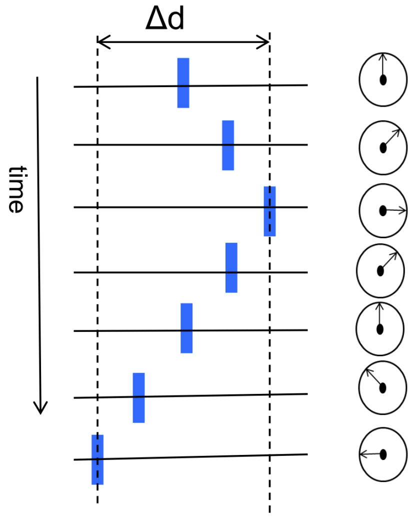


Figure 2: A representation of the time evolution of the phase of an IF signal[5]

4. System Architecture

Physical Testing Fixture

In order to test and demonstrate our system, we needed a sturdy and portable mounting fixture that would hold the radar and DCA upright and stable for the prolonged measuring period. It also needed to be somewhat elevated above whatever surface we were using to prevent unwanted readings in the data. We printed a simple solution that sits atop a table or flat surface. The DCA screws into it, with the radar fixed on top of it, secured just via the connector to the DCA. This was sufficient for all of our testing and demonstrating needs.

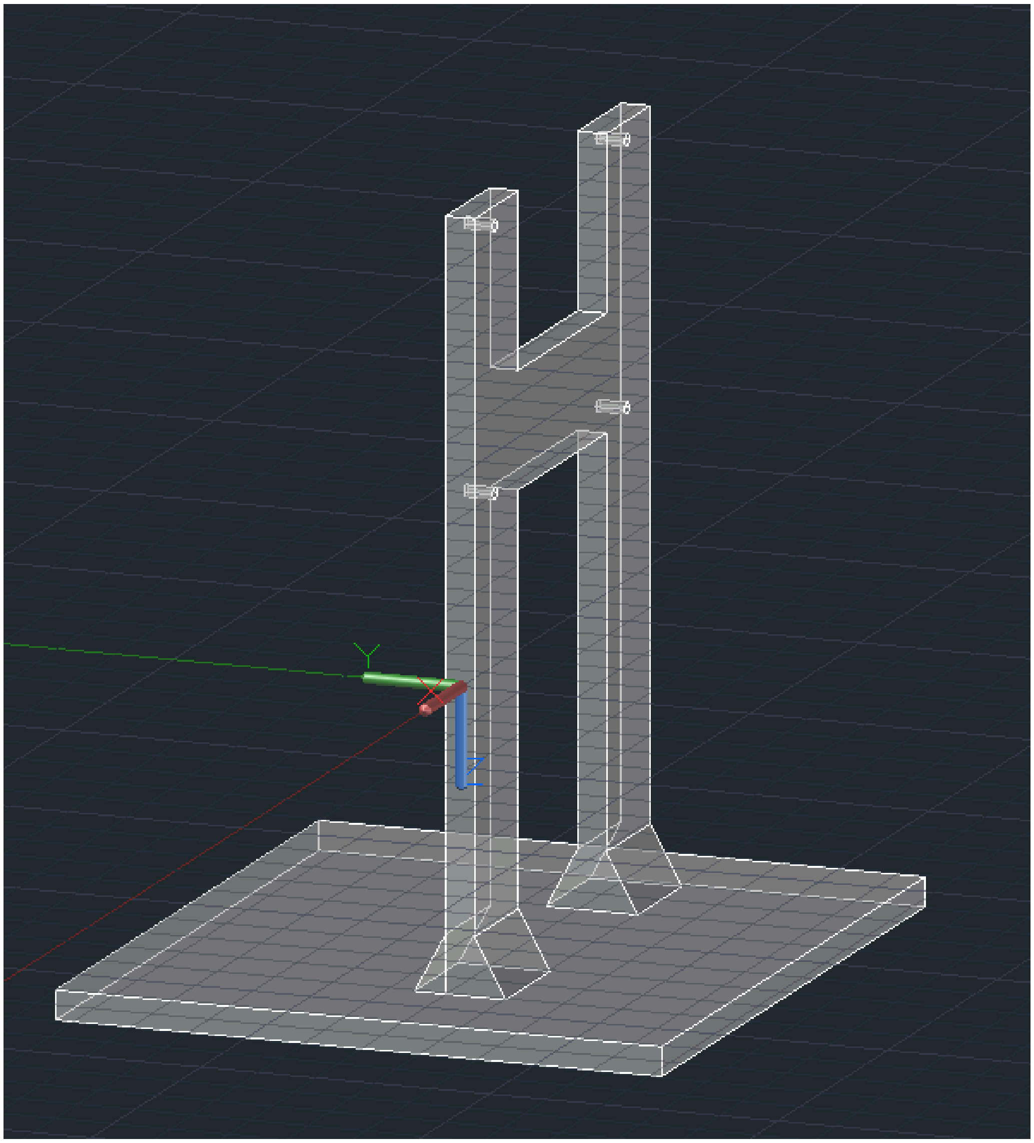


Figure 3: CAD drawing for mounting fixture

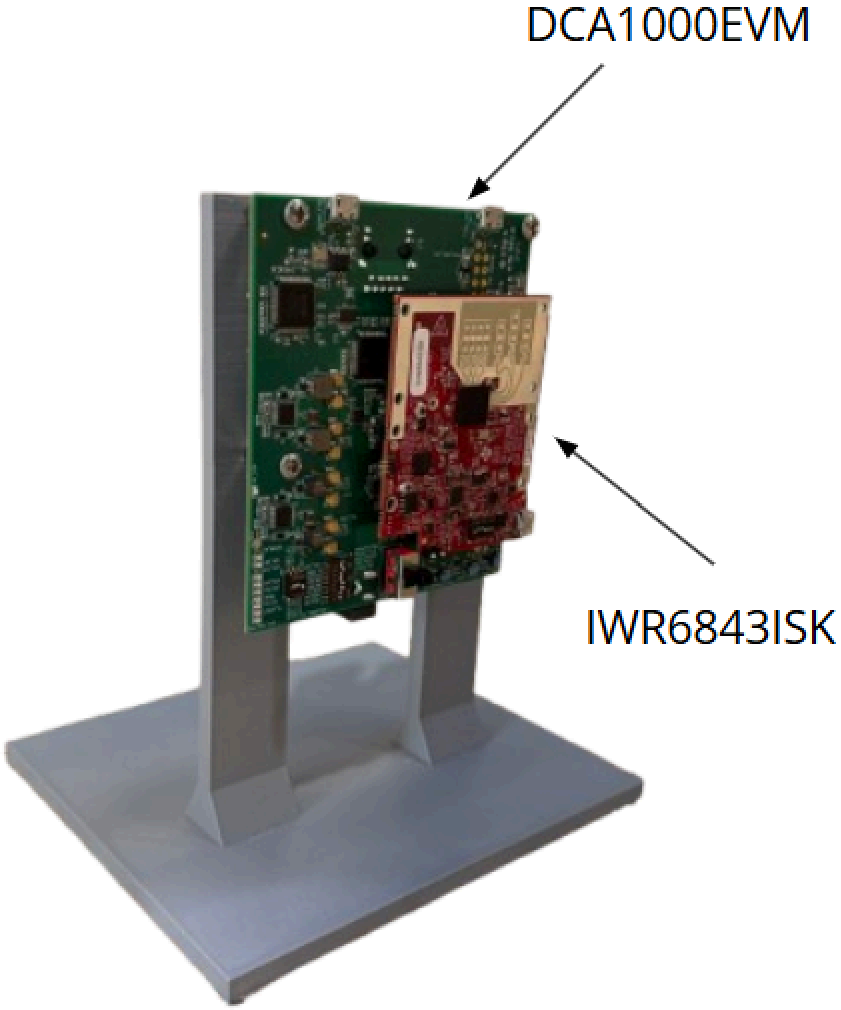


Figure 4: Printed mount with radar & DCA attached

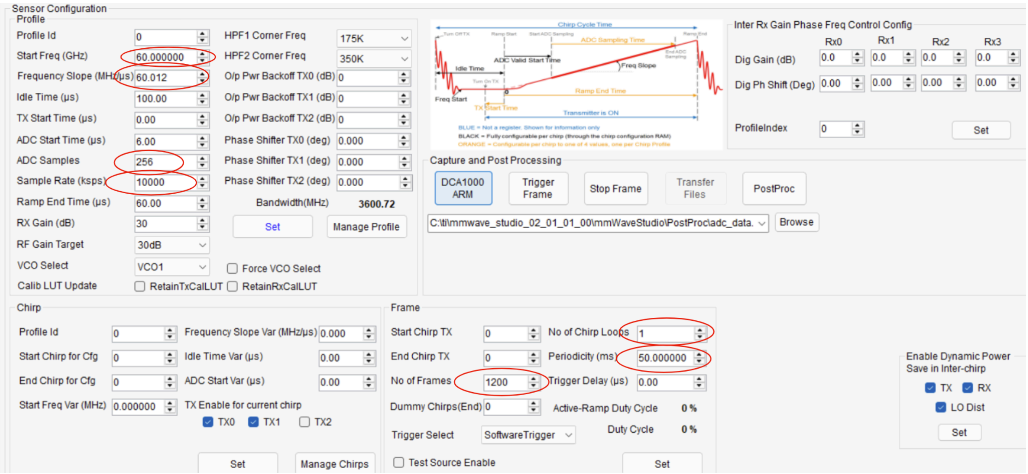
Texas Instruments Hardware

The first step in the design process was to configure the radar to get it to successfully read data. We elected for our system to consist of a IWR6843 60GHz long-range antenna mmWave sensor, and DCA1000 evaluation module (EVM)[7] for real-time data capture, visualization, and streaming. This radar was chosen because its frequency provides the necessary precision to detect millimeter movements, as well as the fact that it comes with development kits and a large number of channels, which aid in maximizing angular resolution.

To configure the radar, we first had to install the appropriate USB to UART drivers to communicate with the DCA and the radar. We struggled with this a little bit given the discrepancies in the documentation between similar models, but ultimately found the right ones, the Silicon labs and FTDI drivers. We also had to dig through documentation and forums to find out what the dip switches on the radar do and how they correspond to operating modes, eventually finding the correct “development mode” and its associated switch pattern to interface with the DCA. From there we just followed the instructions in the TI MMWAVE-STUDIO + DCA1000 setup tutorial[8], successfully connecting over UART and SPI.

MMWAVE-STUDIO

We installed and set up MMWAVE-STUDIO to begin collecting data from our radar. MMWAVE-STUDIO has loads of configuration options, of which we changed seven parameters to fully optimize the radar for our purposes. After triggering the radar to capture data, then selecting “PostProc,” the data is stored as a .bin file in the target directory. This .bin file can be read by MATLAB and subjected to our own processing from there.



Sensor Configuration

Profile

Profile Id 0
Start Freq (GHz) 60.000000
Frequency Slope (MHz/μs) 60.012
Idle Time (μs) 100.00
TX Start Time (μs) 0.00
ADC Start Time (μs) 6.00
ADC Samples 256
Sample Rate (ksps) 10000
Ramp End Time (μs) 60.00
RX Gain (dB) 30
RF Gain Target 30dB
VCO Select VCO1
Calib LUT Update □ RetainTxCalLUT □ RetainRxCalLUT
HPF1 Corner Freq 175K
HPF2 Corner Freq 350K
O/p Pwr Backoff TX0 (dB) 0
O/p Pwr Backoff TX1 (dB) 0
O/p Pwr Backoff TX2 (dB) 0
Phase Shifter TX0 (deg) 0.000
Phase Shifter TX1 (deg) 0.000
Phase Shifter TX2 (deg) 0.000
Bandwidth(MHz) 3600.72
□ Force VCO Select

Inter Rx Gain Phase Freq Control Config

	Rx0	Rx1	Rx2	Rx3
Dig Gain (dB)	0.0	0.0	0.0	0.0
Dig Ph Shift (Deg)	0.00	0.00	0.00	0.00

ProfileIndex 0

Capture and Post Processing

C:\ti\mmwave_studio_02_01_01_00\mmWaveStudio\PostProc\adc_data.

Chirp

Profile Id 0
Start Chirp for Cfg 0
End Chirp for Cfg 0
Start Freq Var (MHz) 0.000000
Frequency Slope Var (MHz/μs) 0.000
Idle Time Var (μs) 0.00
ADC Start Var (μs) 0.00
TX Enable for current chirp ☑ TX0 ☑ TX1 □ TX2

Frame

Start Chirp TX 0
End Chirp TX 0
No of Frames 1200
Dummy Chirps(End) 0
Trigger Select SoftwareTrigger
No of Chirp Loops 1
Periodicity (ms) 50.000000
Trigger Delay (μs) 0.00
Active-Ramp Duty Cycle 0 %
Duty Cycle 0 %
□ Test Source Enable

Enable Dynamic Power Save in Inter-chirp
☑ TX ☑ RX
☑ LO Dist

Figure 5: Sensor configuration settings on mmWave Studio

First, we set our start frequency to be 60 GHz, and the slope of our chirp to a level that would correspond to a 3.6 GHz bandwidth. Our motivation for increasing the bandwidth as much as the radar specifications recommended we could was simple: more bandwidth leads to greater range resolution, which will help us more accurately determine the location of the chest surface, which will help give us more accurate phase data to work with when we do our vibration FFTs.

Next, we set the number of ADC samples per chirp to 256, and our sampling rate to 10,000 kilo-samples per second. These were the default values and were more than we needed, but we kept them as is because the hardware could handle it. More resolution than we needed was hardly a problem.

Next, we chose only to send out one chirp per frame. We originally attempted to cram as many chirps as we could into a single frame to maintain phase continuity, but we found that there was a limit as to how many chirps you could fit into one frame before the radar needed to reset, and that number would not have provided a fine enough resolution in the frequency domain of the phase data at the chest to properly determine vital sign information. We found a github repository[6] of a similar project that they had only used one frame per chirp, since there is not a limit on frames. This method also provided phase continuity, and therefore decided to go with this method.

Lastly, we set periodicity (chirp repetition interval) to 50 ms, and number of chirps to 1200. Correlating the resulting FFT sample bins to a frequency, we can see that these parameters provide a vibration resolution of 1/60 Hz, or 1 bpm. We believe this to be an appropriate tradeoff between accuracy of measurement and data acquisition time, as to provide any greater resolution, the time would need to be well over the current acquisition time of 60 seconds.

Radar data, once processed by MMWAVE-STUDIO, is converted into a .bin file, which can be passed to an application of our choice for additional data processing and algorithm development. For this project, we used MATLAB for our data processing, as that is what we have the most prior DSP experience in. As a bonus, TI additionally supplies a MATLAB script[9] for converting this binary data into a complex IF signal, which gives us a launching off point for the rest of our signal processing code.

5. Implementation

As previously discussed, once we have the .bin file converted to complex data, we must use the frequency spectrum of that IF signal to calculate for range. This complex data comes in the shape [numRX,numChirps*numADCSamples], where numRX is the number of receive channels (4), numChirps is the number of chirps received (1200), and numADCSamples is the number of samples per chirp (256). In other words, it is all of the complex returns from each channel through the entire transmission period.

Range FFT

Since we only want to work in one range bin for the second half of the signal processing, it makes sense for us to reshape the matrix into something that more closely resembles a fast time vs slow time plot, with fast time being the dimension of the matrix corresponding to the number of samples per chirp, while the slow time is the dimension corresponding to the number of chirps.

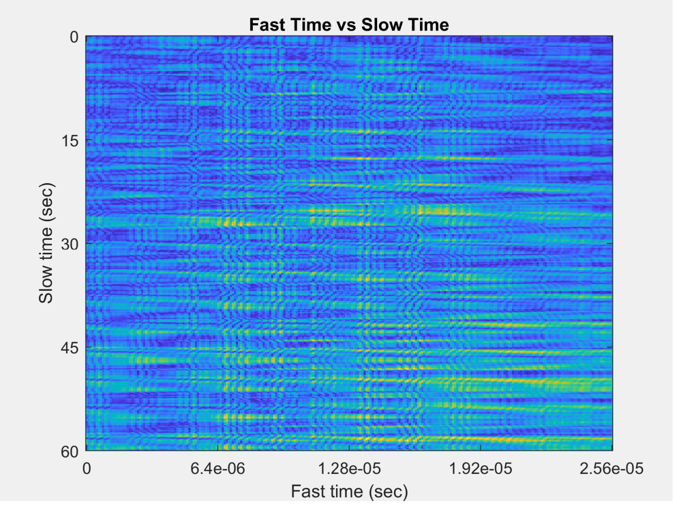


Figure 6: An example of a fast time vs slow time plot

This sample data doesn’t indicate much to us until we take the FFT of each individual chirp in this data to find the frequency spectrum.

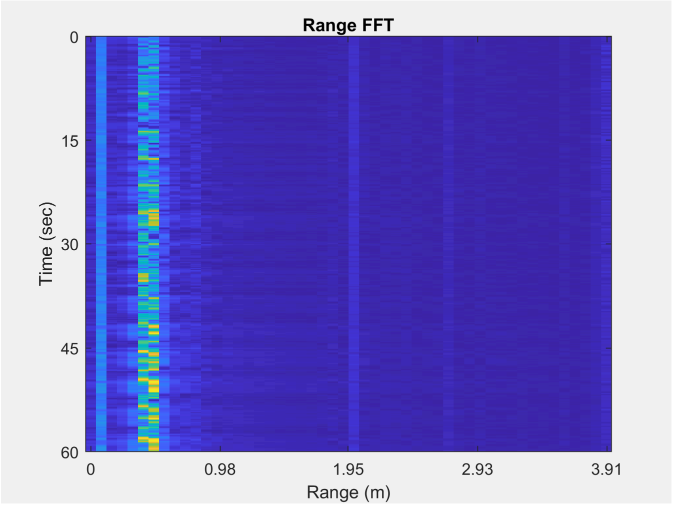


Figure 7: An example of a range FFT

Through this, we can see a very clear, defined line at bin 7 that indicates that there is a very strong beat frequency that corresponds to a return from a particular range. Using this information and plugging into the equations in section 3, we can determine that this range is 68 cm. This range is consistent with the distance we were from the radar, and so we can validate that this range FFT produces acceptable results.

Angle of Arrival (AoA)

Since our radar has four receivers staggered horizontally, this provides us with the ability to filter not only for range, but also for azimuth angle. There are two separate steps we can take to accomplish this. First, since we took our radar measurements when sitting normal to the antennas, we can map out 2D space to validate that the strong return at the 7th range bin is getting the majority of its return at a normal incidence, and not stray angles. Next, once we investigate this range bin across chirps, we can determine the angle spectrum across all chirps and filter out data points that correspond to undesired angles.

To map out 2D space (range vs angle), for a random chirp, we take the FFT across that chirp for each receiver, and store those values in a separate matrix with the size [numRX, numADCSamples], or [4,256]. With this resulting matrix, we can take a 1024 point FFT across the different receive channels to get the strength of return at different angles. Because we are doing this FFT to one chirp, which consists of bins corresponding to different ranges, the result of this FFT is a matrix that maps out an angular spectrum at each range, hence mapping out 2D space. In the below figure, we can see that the majority of our return is at this same range bin, mostly between -30 degrees and 30 degrees, and so our results are validated.

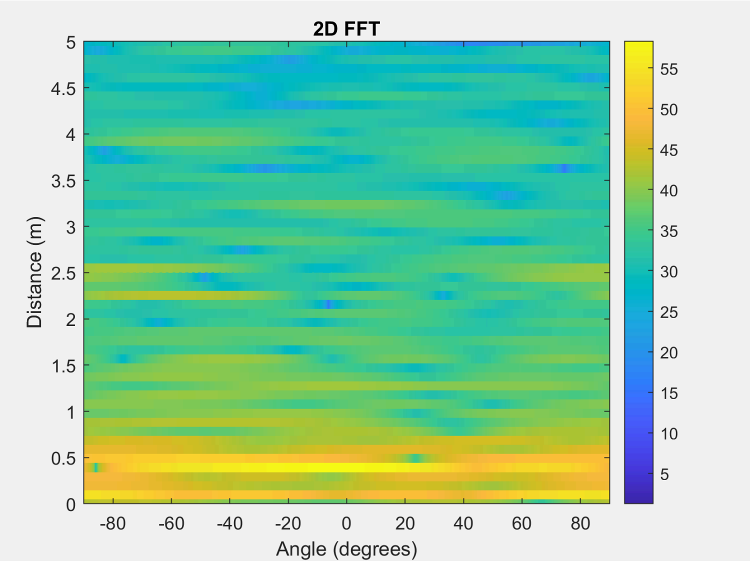


Figure 8: A contour showing 2D space (range vs angle)

We can further validate that our 2D FFT is accurately representing 2D space with the addition of a corner reflector at a non-normal angle to the radar. Running this data shows a desired result of a strong return a half a meter behind the subject at an angle of -60 degrees to the radar. Taking a range FFT of a random chirp shows a similar peak at this range, and so our AoA algorithm is further validated.

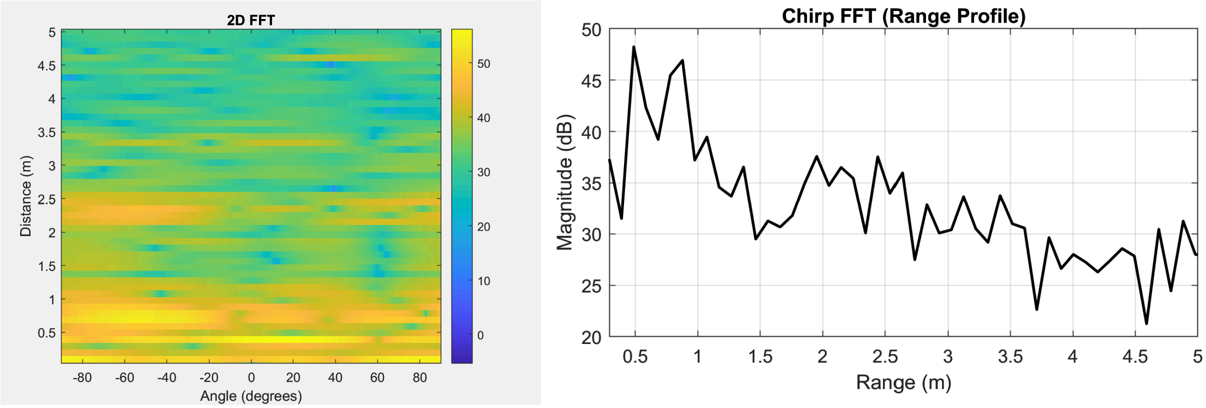


Figure 9: A contour showing 2D space with a subject plus a corner reflector (left), and a range FFT of this same data (right)

Next, if we are going to filter the desired range bin for angle, we must do the same process, but instead of taking the fast time data of one chirp, we must take the slow time data across one range. This new matrix will be of size [numRX, numChirps], or [4,1200]. If we take another 1024 point FFT across the different receive channels, we will get the strength of return at different angles for each chirp.

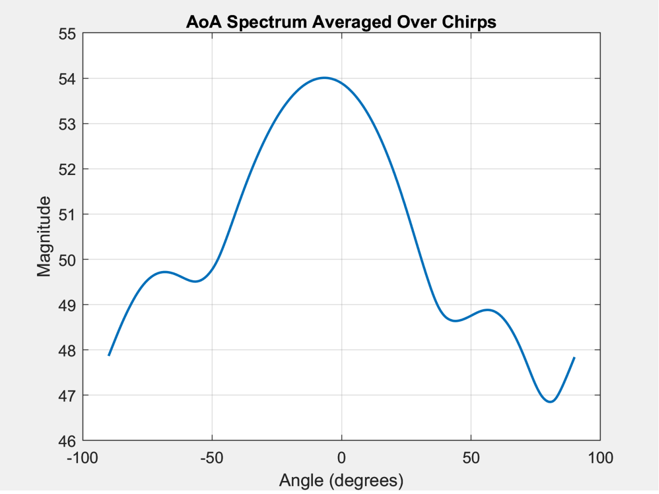


Figure 10: Average angular spectrum at one range bin. In this plot, we see that the peak angle of arrival is at -6.6 degrees, which confirms that we are receiving at normal incidence.

After this, all we have to do is to create a mask from -30 to 30 degrees, zero out angles outside of this mask, and take an inverse FFT to get back into the spatial domain.

Vibration FFT

Once we have filtered complex data at a particular range bin, we must investigate the phase at this range bin and how it changes in order to get a vibration FFT. The steps to getting this data are pretty similar to getting the frequency information of the first FFT, however first we must unwrap the phase data (adding appropriate multiples of 2π to each phase input to restore original phase values). Then, we introduce various filtering that we discuss in the next paragraph, take the FFT, and plot out the data with the bins changed to its corresponding bpm (frequency times 60).

The vibration FFT contained all the data we need to determine heart rate and breathing rate, but extracting and denoising that data proved to be a challenge. Breathing rate was significantly easier, due to the fact that the breathing motion is much larger and more purely periodic (sinusoidal) in nature. For this signal, we applied a FIR bandpass between roughly 10 and 25 BPM and this worked most of the time. However, there was some low frequency noise on the lower end of the band pass that would cause erroneous readings on occasion. To circumvent this, we applied another filter, a high pass, just a bit past 10 bpm to aid in attenuating this low frequency noise. This worked, and the breath rate monitoring was very successful, measuring correctly the majority of the time.

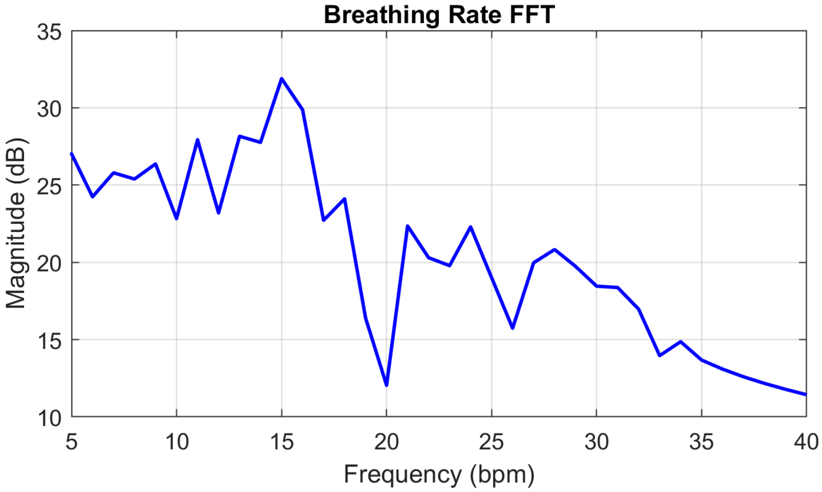


Figure 11: An example of the breath rate vibration FFT

Extracting heart rate from this data followed an identical process, but with a high pass filter set to 60 bpm and a band pass from about 60 bpm to 150. There was significantly more low frequency noise in this band, likely due to the heart rate's noisier nature. We passed the filtered heart data through a hann window before analyzing, and this helped some but we still had issues. Heart rate was accurate roughly half of the time, and we would need to try more sophisticated filtering as discussed later to try and do better.

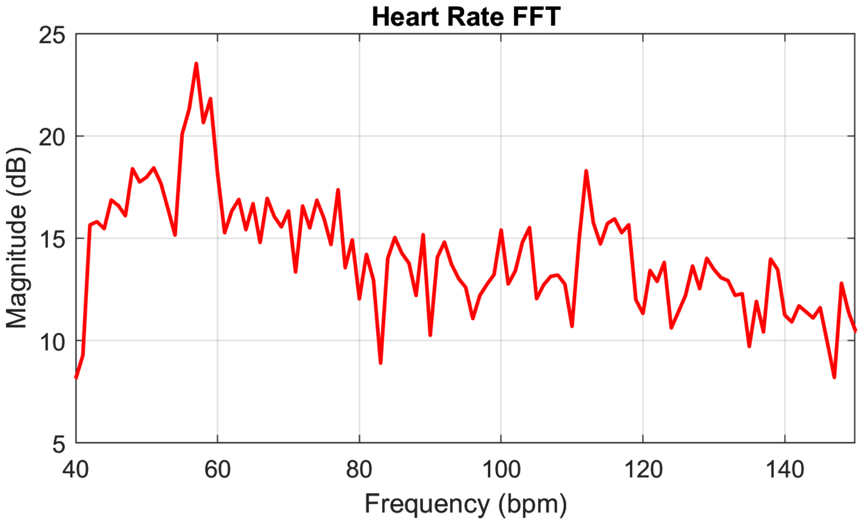
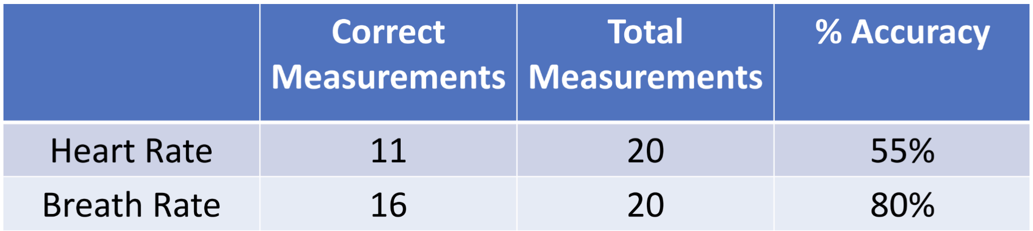


Figure 12: An example of an accurate heart rate vibration FFT

6. Results

Our system performed well for breathing rate detection, and did alright with heart rate. Cardiac waveforms are very noisy in and of themselves, and with added noise from our radar measurement, extracting the fundamental frequency proved to be a challenge.



	Correct Measurements	Total Measurements	% Accuracy
Heart Rate	11	20	55%
Breath Rate	16	20	80%

Figure 13: Accuracy results

More sophisticated filtering than the typical frequency discrimination methods we applied are needed to reliably measure the heart rate. These are discussed in the following section.

7. Future Work

In our time working on this project, we were unable to consistently pull out an accurate heart beat. We were not able to achieve an accuracy above around 50%, and in order to improve our chances, we would need to implement more advanced signal processing methods. One in particular that we have identified is a method called matched filtering. In general terms, this method consists of convolving the return signal with the conjugate of the theoretical desired return to identify the location of that return. To apply this method to our application, we would have to simulate the phase data of a typical heartbeat sampled at the same sampling frequency as our chirp repetition frequency, and convolve that with our phase data before we take the FFT. We would hope that this would result in the frequency of the heartbeat being more pronounced and more easily identifiable. We are not sure if it would introduce more difficulty if we are searching for a range of heartbeat frequencies, but this is something that would be investigated with additional time.

Additionally, if we wanted to create a 2D space plot where “targets”, AKA the chest, are identified, rather than simply looking for strength of return at certain areas, we would have to compare the return signal to its surrounding noise through Constant False Alarm Rate (CFAR) mapping. Through this algorithm, we would use a cell averaging “noise estimate” matrix to filter through the 2D FFT map. To summarize, this matrix involves a cell under test (CUT), guard cells, and training cells. Guard cells create a border of zeros around the CUT to prevent other signal components from leaking into the training cell. If any given CUT, after filtering, is more than a defined threshold above the noise estimate of the surrounding area, it counts as a detection and shows up in the CFAR map.

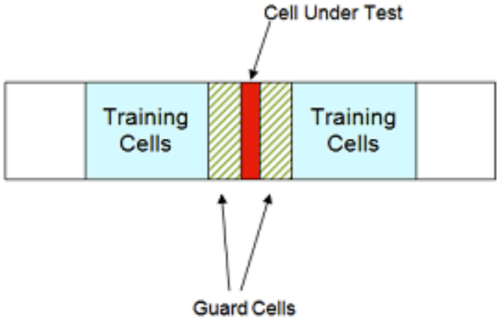


Figure 14: An example of the array to filter through the 2D FFT plot, involving a CUT, training cells, and guard cells

8. References

1. C. Li, J. Cummings, J. Lam, E. Graves and W. Wu, "Radar remote monitoring of vital signs," in IEEE Microwave Magazine, vol. 10, no. 1, pp. 47-56, February 2009, doi: 10.1109/MMM.2008.930675.keywords: {Remote monitoring;Radar detection;Doppler radar;Heart beat;Biomedical monitoring;Pediatrics;Radio frequency;Security;Humans;RAKE receivers},

2. Kebe, M., Gadhafi, R., Mohammad, B., Sanduleanu, M., Saleh, H., & Al-Qutayri, M. (2020). Human Vital Signs Detection Methods and Potential Using Radars: A Review. *Sensors*, *20*(5), 1454. https://doi.org/10.3390/s20051454

3. A. Santra, R. V. Ulaganathan and T. Finke, "Short-Range Millimetric-Wave Radar System for Occupancy Sensing Application," in *IEEE Sensors Letters*, vol. 2, no. 3, pp. 1-4, Sept. 2018, Art no. 7000704, doi: 10.1109/LSENS.2018.2852263. keywords: {Sensors;Doppler effect;Chirp;Legged locomotion;Doppler radar;Radar antennas;Sensor signals processing, sensor applications;human–machine interface;occupancy;sensing;smart homes;60 GHz-mm wave radar},

4. M. Alizadeh, G. Shaker, J. C. M. D. Almeida, P. P. Morita and S. Safavi-Naeini, "Remote Monitoring of Human Vital Signs Using mm-Wave FMCW Radar," in *IEEE Access*, vol. 7, pp. 54958-54968, 2019, doi: 10.1109/ACCESS.2019.2912956. keywords: {Heart rate;Phase noise;Monitoring;Radar remote sensing;Radar detection;Receivers;Breathing rate monitoring;FMCW radar;heart rate monitoring;Hexoskin;mm-wave;non-contact monitoring;phase analysis;remote sensing;vital signs;TI},

5. Basics on FMCW radar: https://www.ti.com/content/dam/videos/external-videos/en-us/2/3816841626001/5415203482001.mp4/subassets/mmwaveSensing-FMCW-offlineviewing_0.pdf

6. KylinC, “mmVital-Signs,” https://github.com/KylinC/mmVital-Signs

7. Radar used for project: https://www.ti.com/tool/IWR6843ISK

8. Texas Instruments, “DCA1000 Training Video,” https://www.ti.com/video/5827389052001

9. Interfacing .bin file with MATLAB:

https://www.ti.com/lit/an/swra581b/swra581b.pdf?ts=1740680092484