Project Specification for Millimeter-wave People Detection Radar

Wireless Embedded Systems Capstone
by Steven Daniels and Matthew Hatch

April 9, 2024

UC San Diego

Contents

1 Project Charter

1.1 Overview

This project aims to implement a system that uses a high-frequency, short-range radar to detect humans present in front of the radar. This detection will be in the form of a two-dimensional plot that shows the distance to the target and the target's motion relative to the radar, and the strength of the detection. This plot is called a "range-Doppler map". Additionally, if time permits, it also aims to produce a three-dimensional point cloud that allows visualization of the human targets in 3D space.

1.2 Approach

We will implement this system using a pre-built radar system called an Antenna-on-Platform (AOP) offered by Texas Instruments. The IWR6843AOPEVM combines a 60 GHz radar platform with a highly capable digital signal processing subsystem in a single system-on-chip. The AOP provides the analog frontend necessary to transmit and receive a Frequency-modulated Continuous Wave (FMCW) radar waveform; it handles generation of the frequency ramp, duty cycle management, mixing the transmit and receive signals, and I/Q sampling of the intermediate frequency signal.

Since the AOP handles all aspects of waveform generation and reception, the project will focus on the digital baseband signal processing for the FMCW waveform. FMCW operates by transmitting a sequence of "chirps", which are short transmissions consisting of a carrier wave that linearly ramps in frequency over the duration of the transmission. The transmitted signal then enters the channel and reflects off of targets. The reflected signal then arrives at the receiving antenna; this signal is then mixed with the signal being transmitted to produce an "intermediate frequency" (IF) signal. The difference in frequency between the received signal and transmitted signal manifests as a tone in the IF signal; the frequency of the tone is used to estimate range to the target. If there are multiple targets reflecting energy at different ranges, then multiple tones will appear in the IF signal.

Upon the reception of each subsequent chirp, a Discrete Fourier Transform (typically by Fast Fourier Transform) is performed to identify the tones present therein; this step of the baseband processing is called the Range or "fast-time" FFT.

In order to estimate the velocity of the target, the radar transmits a fixed number of chirps; this collection of chirps is referred to as a "frame". As the radar transmits and receives chirps, it performs the fast-time FFTs and buffers the received signals. Once a full frame of chirps has been transmitted and received, the baseband processor performs the next step, which is a two-dimensional Discrete Fourier Transform over all of the IF signals of the chirps in the frame. This is referred to as the Doppler or "slow-time" FFT.

The slow-time FFT observes the minute changes in the phase of the received signal over the course of the frame and uses them to estimate the Doppler effect as a result of the relative motion between the radar platform and the target. While the fast-time FFT utilizes the difference in frequency between the transmitted and received signal to estimate range, the slow-time FFT relies on changes in phase to estimate frequency.

Once the Doppler frequency of the target is known, it is a simple algebraic problem to convert Doppler frequency to relative velocity.

The IWR6843AOPEVM features a baseband digital signal processor (the TMS320C674x) and an onboard hardware accelerator that includes cores for common FMCW processing blocks, including Fast Fourier Transforms. Additionally, Texas Instruments provides a software SDK that includes libraries for interfacing with the board hardware and communicating between the analog frontend and digital signal processing subsystem. Therefore, this project will primarily consist of developing an implementation of the range and Doppler processing steps and then utilizing the SDK to implement and verify the processing in hardware. We will then use Matlab or a similar tool to plot and visualize the output of the processing in real time to show that the radar can successfully image a human target.

1.3 Minimum Viable Product

The minimum viable product for this project is a demonstration that the IWR6843AOPEVM can be attached to a Windows PC and, using the software we have written, can successfully produce a range-Doppler map and display a plot of it in real time on the host PC. This range-Doppler map should show clean, strong returns for targets in front of the radar. We intend to demonstrate the range-Doppler map generation using corner reflectors (stationary and moving) and human targets. We also want to demonstrate that it can show returns for multiple targets at different ranges and velocities in its field of view.

Once the minimum viable product is achieved, the next step would be to implement radar tracking and clutter-removal algorithms. These algorithms process the range-Doppler data and perform two important functions that would allow the radar to be used in a real industrial application; they eliminate false returns from static objects in the environment that obscure the detection of actual targets of interest (i.e. humans), and then summarize the range-doppler map into a collection of identified targets by applying a detection threshold. The canonical algorithm for this purpose is called Constant False Alarm Rate (CFAR). The IWR6843AOPEVM hardware accelerator has built-in support for the processing involved in the CFAR algorithm.

Following that, we plan to leverage the fact that the IWR6843 radar is a MIMO system; it features 4 receive and 3 transmit antennas. The algorithms thus far have depended on and only use a single baseband I/Q stream from one of the receive antennas. However, the use of only a single transmit and receive antenna means that the radar cannot resolve targets in 3D space; it is only capable of estimating range and relative motion. The addition of multiple receive antennas allows observing the difference in phases between the incident received signals, which in turn can be used to estimate the angle of arrival (AoA, or azimuth) of the signal (in a plane around the radar with two antennas, or in three-dimensional space with three antennas). Since the IWR6843 possesses three receive antennas, it is possible to estimate the position of an object in 3D space relative to the radar by combining angle-of-arrival estimation with range estimation.

By combining CFAR target detection with AoA processing, the long-term goal for the project is to produce a three-dimensional point cloud visualization of the radar's field of view.

1.4 Constraints, Risk, and Feasibility

Given the progress we have already made, the main roadblock to achieving the MVP will be understanding and successfully utilizing Texas Instruments' Millimeter Wave Radar SDK to create a radar processing pipeline. We have already obtained all of the required hardware, written and tested Matlab simulations of the range and Doppler processing, and performed checkouts to ensure that the radar platform is fully functional. The software SDK, however, is extremely powerful and fully-featured, but this means that it is also rather complicated and there is a vast amount of documentation to comb through to understand it. We have given ourselves plenty of time by completing all the other steps early, though, and so we anticipate we should be able to write and debug a custom radar processing pipeline within the quarter.

Another potential pitfall is issues with the Texas Instruments evaluation boards; while we have already ordered, received, and checked out all the required hardware, there have been some issues with them working consistently; however, we have two copies of each of the boards, and the DCA1000EVM used for raw ADC capture is not strictly required for the project, so we do not consider this risk to be particularly worrisome.

2 Group Management

Thus far, we have mostly divided up work along a theory/implementation boundary; since Steven has more experience with radar systems and the theory behind them, and Matthew has experience primarily in communications systems and modems design, Steven has been the primary developer for the Matlab simulations, while Matthew has worked more on the hardware integration, including board checkout and testing and creation of Simulink models.

We communicate via Discord, and since there are only two of us, haven't needed to establish a formal decision-making process. We have been managing schedule by following a sprint-based model of establishing a set of obtainable goals for two-week periods and aiming to complete at least those goals in the time frame. Thus it is easy to see when the schedule slips, and we are able to rearrange the estimated work for later sprints to account for any such issues. We have given ourselves some extra time in the planned schedule at the end of the quarter to account for any unanticipated schedule changes.

3 Project Development

The project will use an IWR6843AOPEVM radar evaluation platform, MMWAVEICBOOST carrier card which provides an integrated JTAG probe/debugger for loading code onto the AOPEVM's ARM processor and digital signal processor and then debugging the code running on the board. These two boards are minimum requirements for the project, as both are needed to develop for the AOPEVM. Additionally, we are using a DCA1000EVM capture card which allows real-time streaming of the raw analog-to-digital converter output of the IWR6843 analog frontend to a host PC for capture and analysis to aid in debugging and algorithm design. We have ordered two sets of this hardware and have already received and verified all the units work.

We have a corner reflector that can be used as an easy-to-detect baseline target for the radar; combined with the fact that Texas Instruments supplies a reference implementation of the FMCW radar that generates a range-Doppler map, we are able to test our system by comparing the output of our custom-written processing pipeline against Texas Instruments'. We can also use the corner reflector in a controlled environment to determine what the 'ground truth' is that we should be seeing in the system output.

We have documented the algorithms development via heavily-annotated Matlab scripts and READMEs with plots of the output of the processing steps; we also plan to create Matlab live scripts which demonstrate the processing and produce visualizations of the algorithms inline with the code. Documentation for the embedded C++ code will be done by cross-referencing the C++ implementation against the corresponding Matlab implementation in a standard C++ documentation standard such as Doxygen-style doc comments.

4 Project Milestones and Schedule

The high-level concrete deliverables for our project are:

- Millimeter Wave Radar SDK application suite (DSP Subsystem and Mission Subsystem firmware) that outputs a range-Doppler map of the radar's field of view in real time, which can demonstrate the detection of human targets.
- Windows host PC software (e.g. Matlab script) that can start and stop the radar application without the need for Texas Instruments' Millimeter Wave Studio and visualize the output of the radar application.
- (Optional, time permitting) DSS/MSS firmware and visualizer for 3D point cloud of targets detected by the radar

We have organized these into biweekly sprints as follows:

1. April 5th to April 19th:
   (a) Using raw I/Q data captured from the IWR6843AOP ADC, validate that the Matlab scripts we have written produce results that are consistent with the range-Doppler map produced by Texas Instruments' own postprocessing. This will be handled by Steven.

   (b) Redo the preceding item with a more realistic scenario; i.e. instead of pointing the radar at a small, empty room, put real targets in front of it (a corner reflector and a person) and verify that the results are consistent with the real target picture. We should be able to observe returns in the range-Doppler map for the target at the correct range and velocity. This will be handled by Steven.

(c) Perform an initial checkout of the MMWAVEICBOOST carrier card. This includes that it powers on, the USB connection to Windows is functional, and that the integrated JTAG debugger is usable through Code Composer Studio. Additionally, survey the documentation for the carrier card to determine the steps necessary to compile and flash MSS and DSS firmware on the IWR6843AOPEVM using the JTAG debugger. This will be handled by Matthew.

2. April 20th to May 3rd:

   (a) Attempt to intercept the raw ADC data output by the DCA1000EVM over Ethernet and see if it can be streamed in real-time instead of merely being saved to a capture file. If so, consume the stream in Matlab or Simulink and run the range-Doppler processing on it to produce a real-time plot. This will be handled by Steven.

   (b) Create a custom radar processing pipeline using the Millimeter Wave SDK that runs on the IWR6843AOPEVM; verify that the output can be read from the UART over USB on the host Windows PC, e.g. in Matlab. The output may initially just be a dummy pattern; the goal here is to gain an understanding of how to write and flash an application on the radar board. This will be handled by Matthew.

   (c) Write a program (in Matlab or possibly a traditional programming language) to send commands to the Millimeter Wave application's command line interface over the UART, in order to set up the radar parameters and start and stop the radar. This will be handled by Matthew.

3. May 4th to May 17th:

   (a) Convert the dummy custom pipeline to one that produces the desired range-Doppler map by implementing the range and Doppler FFTs in the pipeline, and any other requisite filtering necessary to clean up the signal. This will be handled by Matthew.

   (b) Write a Matlab or Simulink script that reads the range-Doppler data off of the UART and plots it for viewing. This will be handled by Steven.

4. The remaining time is allocated as slip time to account for unexpected delays in the planned schedule; if we finish the range-Doppler map deliverables with time to spare, the remaining time will go to attempting to build a 3D point cloud visualization.