Digital Signal Processing for Frequency-Modulated Continuous Wave RADARs

Author

Dmitry Utyansky
Senior Staff Engineer,
Synopsys

Introduction

RADAR is an established technology, but interest has been stimulated recently by demands of driver assistance systems and emerging self-driving cars for applications including proximity warning, blind spot detection, adaptive cruise control, and emergency braking.

Advanced driver-assistance systems (ADAS) and autonomous driving (AD) systems typically combine several types of sensors, such as cameras, RADARs, and LiDARs. Different types of sensors have their strengths, and effectively complement each other. Cameras and the appropriate machine vision algorithms can "see" lane marking and recognize traffic signals and signs. LiDAR can offer high (cm-level) resolution and high density of collected data points. RADAR technology is indispensable in ADAS applications because of its robustness to a variety of environmental conditions, like rain, fog, snow, and its ability to directly and precisely measure range and velocity. Basic use cases can rely on RADAR sensors only, enabling cost-effective solutions. RADAR can be used to augment, cross-check or 'fuse' situational models derived using advanced computer vision algorithms.

Automotive applications add specific requirements, such as:

- Safety and security requirements.
- Life-cycle management: cars can have a 20+ year life cycle, but the domain evolves, both in terms of regulations and algorithms. Therefore, a system implemented today must allow the ability to upgrade and, ideally, some reserves for future growth of computational complexity, especially in higher-end applications.
- Electrical power available to the vehicle is usually limited: this affects the design of the car, costs, and mileage. Lower power consumption also means lower system temperatures and decreased probability of failures.
- As the number of RADAR and other sensors in a car grows, system costs become important. Higher integration and efficient system-on-chip (SoC) designs enable cost reduction, hence the growing need for highly integrated silicon.

To meet these requirements car manufacturers are looking for a more programmable solution that will increase flexibility and enable a future upgrade path.

RADAR implementation complexity and the respective algorithm computational requirements vary depending on application and system requirements. This paper provides an overview of digital signal processing algorithms typically used in frequency-modulated continuous wave (FMCW) RADARs, reviews system tradeoffs, offers a way to approximately estimate computational complexity with a few numerical examples, and finally presents implementation options and solutions available from Synopsys.

Note that the interference from the transmitted signal is easily filtered out with low-pass filtering. The frequency of this difference signal, Δf, is proportional to the round-trip time, Δt:

$$\frac{\Delta f}{\Delta t} = \frac{B}{T} \quad (2)$$

where T is transmission time and B is chirp bandwidth. And so, given (1), the range is ultimately proportional to Δf

$$R = \frac{cT\Delta f}{2B} \quad (3)$$

One observation from (3) is that since the resolution of frequency identification is inversely proportional to the time of observation, T, the range resolution is inversely proportional to the bandwidth used (Table 1):

$$\Delta R = \frac{c}{2B} \quad (4)$$

Bandwidth, MHz	Range resolution, m
10 MHz	15
250 MHz	0.6
500 MHz	0.3
4 GHz	0.04

Table 1: Range resolution vs. bandwidth

High-resolution requirements of automotive applications have been one of the driving forces of migration to 76-81 GHz bands from the legacy 24 GHz band. The accuracy also depends on linearity of RX and TX channels and signal-to-noise ratio (SNR). The following are a few examples of tradeoffs involved in waveform design:

- Wider bandwidth and faster frequency ramp improve range resolution, increasing Δf
- Maximum unambiguous range is limited by the required overlap of the TX and RX signals: T should be large enough
- T cannot grow too much; if the range substantially changes during the chirp, so does the “beat frequency”, introducing ambiguity. There is also ambiguity between range and velocity detection.
- Processing requirements depend primarily on the data rate of the sampled beat signal and the overall system time resolution requirements—how many estimations per second must be made (also referred to as frame rate).

Current systems typically utilize a sweep bandwidth of a few hundred MHz but can go above 1 GHz for ultra-short range. The typical beat frequency is around 10-20 MHz. Figure 2 shows a simplified diagram of a typical one-channel FMCW radar.

The system includes an analog frontend, with a waveform generator for generating chirp signals, controlled by a processor, transmit and receive amplifiers, mixer, low-pass filter, and analog-to-digital converter. There are multiple ways to implement a waveform generator—with a frequency synthesizer using programmable PLL, typically fractional PLL, or with explicit digital-to-analog converter. The digital processing section can be implemented on one or more commercial DSPs or application-specific instruction set processors (ASIPs).

Signal Processing Flow and Computational Requirements

A nice feature of RADARs and, in particular FMCW RADAR, is that both the range and velocity of the target can be obtained directly. Figure 3 shows the typical signal processing for each packet. The first step is identification of tones in the "beats" signal, which then correspond to the ranges of the targets. The straightforward approach is to use Fast Fourier Transform (FFT) algorithms. The tones identified correspond to target candidates. The next step is to find velocities, assuming they are needed. The distance to the target changes slightly from chirp to chirp within the packet, and this distance change results in change of phases in the tones identified in the "range" step. With known phase changes, the velocities of the targets can be computed. This can be considered as the second set of tone identification problems, slicing across the frequency lines of the results of the first layer of FFTs. This second stage of FFTs is often referred to as velocity or Doppler FFTs.

After these two stages of FFTs, there is a 2D map of velocity/range points, with higher amplitude values corresponding to target candidates. Then further processing should be applied to identify real targets. Usually this involves some version of thresholding operations, like Constant False Alarm Rate (CFAR) schemes. CFAR algorithms consider noise variance in the neighborhood of each value being tested, and adjust thresholds accordingly. The approach was originally proposed in the 1960s by Finn and Johnson [11], but now there is a fairly broad family of algorithms, with different variations of threshold computations, using averages, min/max, ordered statistics, etc. Reference [12] for a review.

Subsequently clustering and tracking/filtering algorithms can be applied, depending on how the results are to be used.

details of analog frontend design, but there are tradeoffs there as well, and often the way to decrease analog part complexity and power dissipation is to move more processing to the digital part, essentially doing more filtering as DSP. This has to be factored into computational requirements. Another factor affecting design of the analog frontend and processing of the sampled signal is whether the mixing functionality is implemented using real or complex — I/Q quadrature, also known as the Zero-IF approach. Complex sampling allows a sampling rate that is two times lower, but at the expense of complex symbols, each containing twice as much information.

Suppose for simplicity that after this pre-filtering the signal is down-sampled to the critical rate of 2*16 MSPS, or 16 MSPS with complex samples. That yields 512 complex samples per chirp, 256 chirps per packet, and packet duration of about 10 ms at this rate.

θ

and so

$$\Delta\theta = \frac{\Delta\omega\,\lambda}{2\pi d\cos\theta} \quad (8)$$

Near the point θ=0, where cosine is close to 1, we get:

$$\Delta\omega = \frac{2\pi d\Delta\theta}{\lambda} \quad (9)$$

$$\Delta\theta = \frac{\Delta\omega\,\lambda}{2\pi d} \quad (10)$$

So the wider antennas are placed (the larger d is), the better angular resolution is.

On the other hand, there is a limit to d. If the RADAR must cover the entire ±90 degrees range, then the sampled frequency ω is maximum when the signal comes from the side, $\theta=\pi/2$, and the critical sampling is attained when $d_{max} = \frac{\lambda}{2}$.

For

Increasing Resolution[9]

As was briefly discussed earlier, FFT-based estimation of angle of arrival has limited resolution, especially with few antennas. Estimation based on a small sized FFT suffers from biasing due to FFT “binning” effects. In the example of eight antennas and 8-point FFT, resolution is inherently limited. There is wide main lobe or high leakage (high side lobes), or both. Interpolation or zero padding with larger FFT can be to avoid binning bias. Windowing before FFT can help tuning tradeoffs between main lobe width and side lobes. But the overall estimation accuracy is still limited.

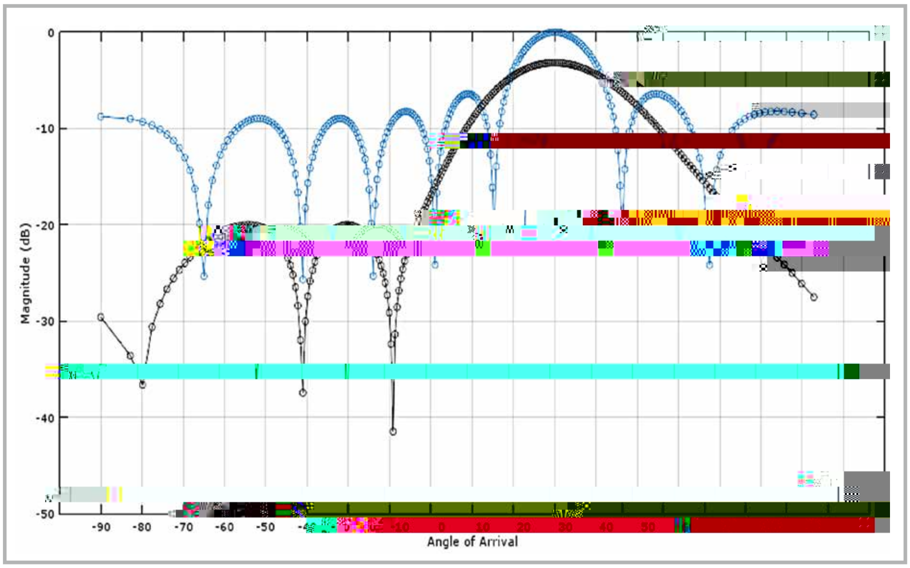


Figure 5: FFT estimation of a single signal with different windowing

Figure 5 shows a typical simulation result for a single signal coming from +20 degree direction, using eight antennas, under ideal noise conditions. Antennas are separated by half-wavelength, enabling +/-90 degrees field of view. Here 256-point FFT is used instead of 8-point one to have more points interpolated. The blue graph uses no windowing, the black graph uses the Hamming window before FFT. This illustrates how leakage to side lobes can be decreased at the expense of increasing the main lobe width (thus decreasing resolution further). Multiple target candidates will ultimately interfere, and close signals cannot be resolved. Figure 6 shows a simulation for the two signals separated by 15 degrees as another example.

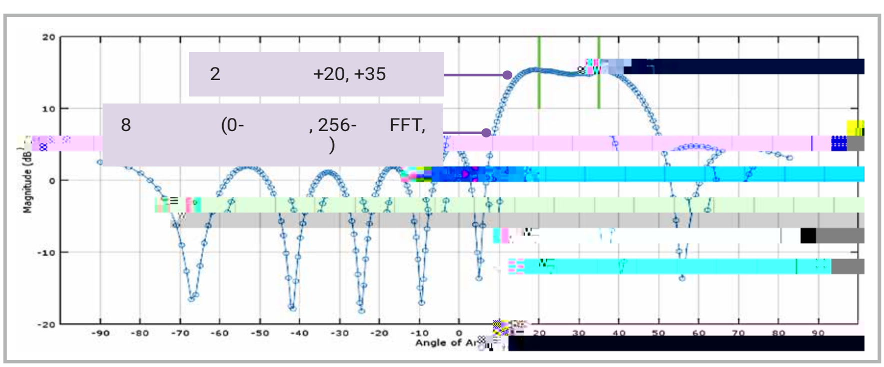


Figure 6: FFT estimation of two signals separated by 15 degrees

The problem can be addressed with larger antenna arrays, increasing system costs. An alternative approach could be to use high-resolution methods like MUSIC (Multiple Signals Classification)[8] and ESPRIT (Estimation of Signal Parameters via Rotational Invariance Techniques)[9]. Figure 7 shows simulated FFT and MUSIC spectrums for the case of two close signals, using FFT-based estimation with eight antennas (blue graph), FFT with 128 antennas (green graph), and using MUSIC algorithm using data from eight antennas (red graph).

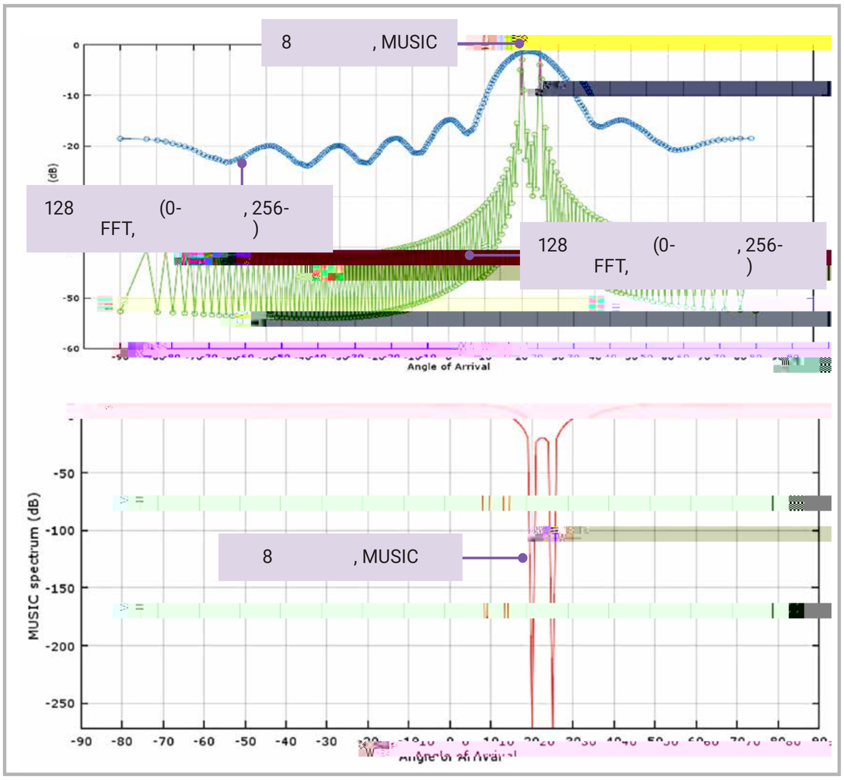


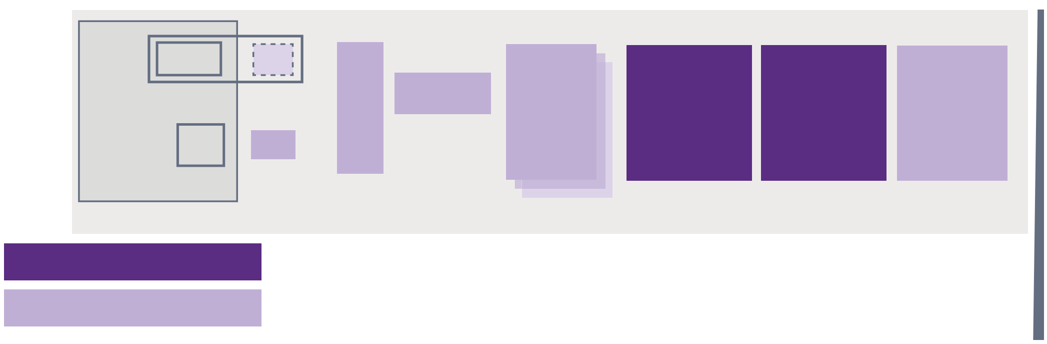
Figure 7: Identification of two close signals, using FFT and MUSIC

The figure shows FFT-based estimation based on eight antennas cannot resolve the two signals, while MUSIC "spectrum" clearly shows them. The reason is that the algorithm can efficiently utilize signal model assumptions, combining several data snapshots to increase accuracy. The underlying mathematics is beyond the scope of this whitepaper, but common features of high-resolution methods, also referred to as "super-resolution methods" (or "subspace methods", of which MUSIC, in particular, is representative) are:

- High resolution, even independent of number of antennas, given enough uncorrelated observations and high enough SNR.
- Usually the number of targets shall be below the number of antennas, or a similar condition exists depending on algorithm details.
- The algorithms ha4e46 rc -0.012 Tw 2.216 0 Td(d,)Tj-0.009 Tc 0 Tw 0.773 0 Td(25)Tj0.04 Tc -0.04 Tw 1.373 0 Td(6-)TjeinDC ‹100.01gt1.9

subsystem, needs to be taken into account. Additional preprocessing, such as windowing, must be factored in as well. Note also that data transfers can create bottlenecks comparable to or worse than the actual computations, therefore this must be considered from the early design stages. If the system relies on any resource sharing, the combined requirements for the supported use-cases must be estimated for processor cycles, memory buses, memory sizes, etc. ADAS applications usually have strict real-time requirements, therefore worst case conditions are of most interest.

For example, consider the case of 8-channel RADAR processing, with each channel sampled at 16 MSPS, and range and velocity FFTs sizes like those used earlier in the Signal Processing Flow and Computational Requirements section. The combined sample rate of 16*8=128 MSPS fits nicely into the range that can be covered with pipelined one sample per cycle FFT accelerators with 128 MHz clock rate, offloading constant heavy-duty FFT processing from the processor core. It might make sense to do FFT processing "on-the fly" getting data from the ADC without writing to memory. There is a fairly wide tradeoff space with such accelerator design; since the eight channels are independent of each other, and FFTs within the packet are independent of each other, the amount of parallelism within the accelerator and its clock rate can be traded off (e.g., going down to 16 MHz rate for each of eight parallel units). The choices also affect buffering requirements and system latency. Direction of arrival computation for an 8-channel case, if done using FFT, can be handled using a scalar DSP core such as Synopsys' DesignWare® ARC® HS47D Processor. The same core can take care of higher-level lower-bandwidth processing tasks, like CFAR, thresholding, higher-level filtering operations and further decision-making. Figure 8 shows such a system, highlighting components available from Synopsys.



1. DesignWare ARCv2 ISA Programmers Reference Manual for ARC HS Processors. Synopsys.
2. MetaWare DSP Programming Guide. Synopsys.
3. DSP Library Performance Databook for ARC HS Processors. Synopsys.
4. DSP Library Performance Databook for ARC EM Processors. Synopsys.
5. MetaWare Fixed-Point Reference for ARC EM and ARC HS. Synopsys.
6. ASIP Designer. Overview of the Manuals.
7. Example processor models. ASIP Designer. Synopsys.
8. R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” IEEE Trans. Antennas Propagation, vol. AP-34, pp. 276–280, Mar. 1986.
9. R. Roy and T. Kailath, “ESPRIT—Estimation of signal parameters via rotational invariance techniques,” IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-37, pp. 984–995, 1989.
10. J. Wenger, “Automotive radar—Status and perspectives,” in Proc. IEEE Compound Semicond. Integr. Circuit Symp., Oct. 2005, pp.21–25.
11. Finn, H. M. and R. S. Johnson, "Adaptive detection mode with threshold control as a function of spatially sampled clutter level estimates," RCA Review, Vol. 29, 414-464, 1968.
12. H. Rohling: Radar CFAR Thresholding in Clutter and Multiple Target Situations, IEEE Trans. Aerosp. Electron. Syst. 19 No. 4, 1983, pp. 608-621.

Pub: November 2018