ISTITUTO
DI TECNOLOGIE DELLA
COMUNICAZIONE,
DELL'INFORMAZIONE
E DELLA
PERCEZIONE

ABSTRACT

The Motivation behind this project is to Test & Evaluate the performance of TI mmWave FMCW Radar system using the AWR1843 EVM development board. Accoringly, the objective is to detect specifically objects within close proximity the radar using USRR configuration and generally evaluate different ranges profiles short, medium and long for further application. Initially, applications related to obstacle detection and collision avoidance can be deployed easily by interfacing to our system. Further application such adaptive cruise control, automatic emergency brake, blind-spot detection, pedestrian or bicyclist protection, and area mapping, localization and exploration to name a few can be deployed by changing the sensing profile configuration module to SRR, MRR, or LRR to fit the application requirements. Using the TI AWR1843 EVM, algorithms run on-board the single-chip device to create Range-Azimuth and Range-Elevation heat-maps, then performs clustering using dbSCAN ML-Algorithm and further non-linear filtering algorithms. The USRR profile synthesizes 12-RX antennas (8 in Azimuth and 4 in Elevation) by utilizing all on-board transmitters (3-TX antennas) in TDM mode. Accordingly this is capable of of achieving a higher angular resolution of 15 and 7.4 degree in azimuth and elevation plan respectively. Further, we were able to detect objects within 20 m range with 4.0cm and velocity up to 30 Km/h with a resolution of 0.3 m/s

Version 1.0 | May 2019

Contents

List of Figures

1 System Introduction

1.1 Features of AWR1843

The AWR1843 features a complete single-chip radar by integrating analog and digital components, including multiple transmit and receive chains, PLL, A2D converters, ARM Cortex-R4F MCU and C674x DSP, memories, and various I/O interfaces. as shown in the table below.

continuous self monitoring and calibration of the RF and analog functionality to a dedicated, built-in ARM R4F-based radio process subsystem which is responsible for front-end configuration, control, and calibration.

AWR1843 also features two ARM Cortex FMCUs running at 200 MHz. One of these is locked and used for the calibration and continuous self monitoring engine of the RF and analog functionality to the the radio subsystem. This R4F is programmed through firmware provided by TI and is not available for user code. However, the second cortex software R4F is available for high-level application processing. The devices support various industry standard input-output interfaces, such as CAN, SPI, I2C, UART, and support high-speed raw ADC data output using CSI2.

The receiver chain starts with the RF front end receiving the reflected radar signal, which is mixed with the transmitter signal to generate a beat frequency signal (IF Signal), which is delivered to the ADC. The ADC converts the analog signal to digital samples which are pre-processed for digital processing. Successive FFTs are computed on the digitized samples for range, velocity, and angle of arrival calculation.

The radar hardware accelerator onboard the 14xx device can be used to offload the FFTs and detection processing to get a point cloud output. The 14xx device can be used to run advanced clustering, tracking, and object classification algorithms using the onboard C674x DSP.

Synchronization Multiple AWR1243 TI 1243 sensor front end provides the high-frequency clock synthesizer output on device bin boundary, which can be fed to other AWR1242 devices to make them all work in sync and act as a single sensor for higher angular resolution.

With cascading, the angular resolution increases exponentially as the number of devices goes up because the total number of angular bins is a product of the total number of physical transmit and receive channels. For instance, a single AWR1243 with three transmit and four receive antennas provides for four times three, or 12 angular bins. But with two AWR1242 devices in cascaded mode, the number of angular bins goes up to six times eight, that is, 48.

The radar sensors are connected over CSI2 to an external processor which performs the FMCW signal processing on the raw ADC data coming from the sensors.

AWR1843 Device Feature

- FMCW transceiver
  - Integrated PLL, transmitter, receiver, Baseband, and A2D
  - 76- to 81-GHz coverage with 4 GHz available bandwidth
  - Four receive channels
  - Three transmit channels
  - Ultra-accurate chirp engine based on fractional-N PLL
  - TX power: 12 dBm
  - RX noise figure:
    - 14 dB (76 to 77 GHz)
    - 15 dB (77 to 81 GHz)
  - Phase noise at 1 MHz:
    - –95 dBc/Hz (76 to 77 GHz)
    - –93 dBc/Hz (77 to 81 GHz)
- Built-in calibration and self-test (monitoring)
  - ARM® Cortex®-R4F-based radio control system
  - Built-in firmware (ROM)
  - Self-calibrating system across frequency and temperature
- C674x DSP for FMCW signal processing
- On-chip Memory: 2MB
- Cortex-R4F microcontroller for object tracking and classification, AUTOSAR, and interface control
  - Supports autonomous mode (loading user application from QSPI flash memory)
- Integrated peripherals
  - Internal memories With ECC
- Host interface
  - CAN and CAN-FD
- Other interfaces available to user application
  - Up to 6 ADC channels
  - Up to 2 SPI channels
  - Up to 2 UARTs
  - I^2C
  - GPIOs
  - 2-lane LVDS interface for raw ADC data and debug instrumentation
- ASIL B targeted
- AECQ100 qualified
- AWR1843 advanced features
  - Embedded self-monitoring with no host processor involvement
  - Complex baseband architecture
  - Embedded interference detection capability
  - Programmable phase rotators in transmit path to enable beam forming
- Power management
  - Built-in LDO network for enhanced PSRR
  - I/Os support dual voltage 3.3 V/1.8 V
- Clock source
  - Supports external oscillator at 40 MHz
  - Supports externally driven clock (square/sine) at 40 MHz
  - Supports 40 MHz crystal connection with load capacitors
- Easy hardware design
  - 0.65-mm pitch, 161-pin 10.4 mm × 10.4 mm flip chip BGA package for easy assembly and low-cost PCB design
  - Small solution size
- Supports automotive temperature operating range

1.2 Hardware Evaluation Target

The hardware tool-chain setup for capturing the raw ADC data consists of the AWR1843 BoosterPack evaluation module (EVM), the mmWave-DevPack provided by Texas Instruments (TI) as long as a PC for visualization (Not for processing) are required to perfrom our design, verification for our use cases. AWR1843BOOST contains everything required to start developing an application on a low-power Arm Cortex R4F processor. As well C67x DSP core for digital signal processing. The evaluation board includes onboard emulation for programming and debugging, onboard buttons, and LEDs for quick integration of a simple-user interface.

Figure 1.1 and Figure 1.2 show the front and rear view of the EVM, respectively.

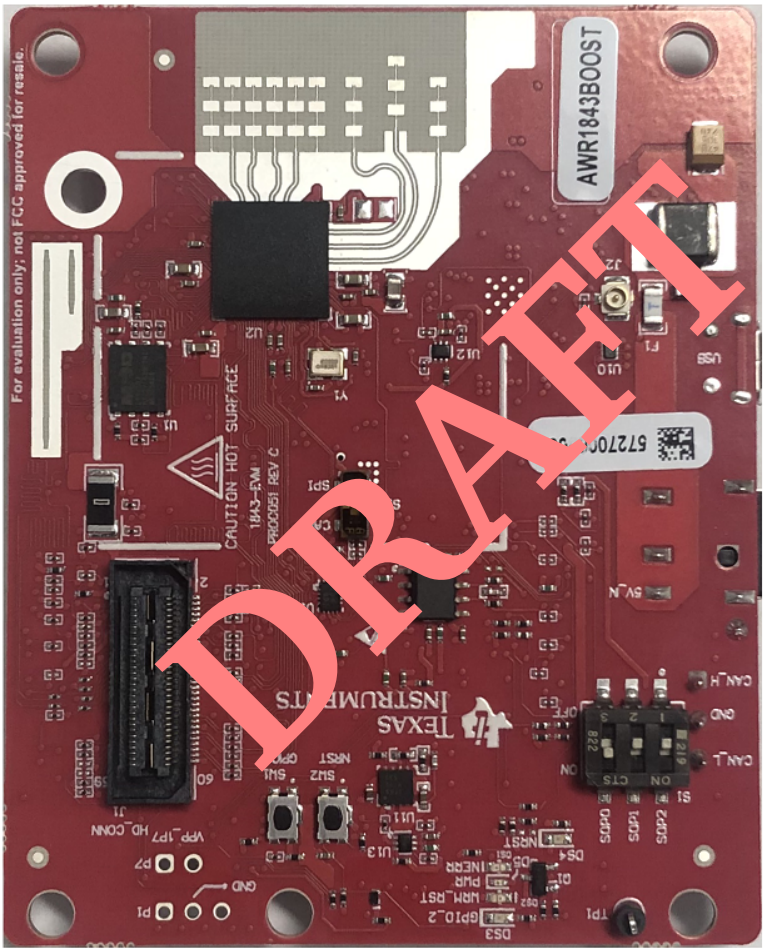


Figure 1.1. EVM AWR1833 Rear View

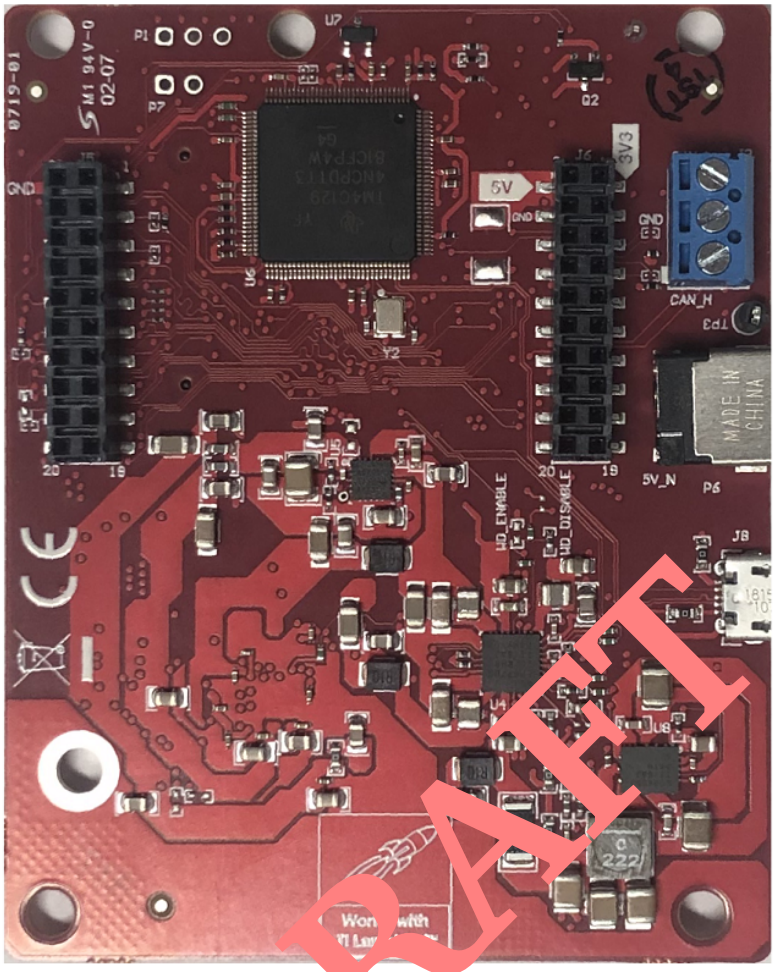


Figure 1.2 EVM AWR1833 Front View

The AWR1843 EVM includes onboard-etched antennas for the four receivers and three transmitters that enable tracking multiple objects with their distance and angle information. This antenna design enables estimation of distance and elevation angle that enables object detection in a three-dimensional plane. Figure 1.3 shows the PCB antennas

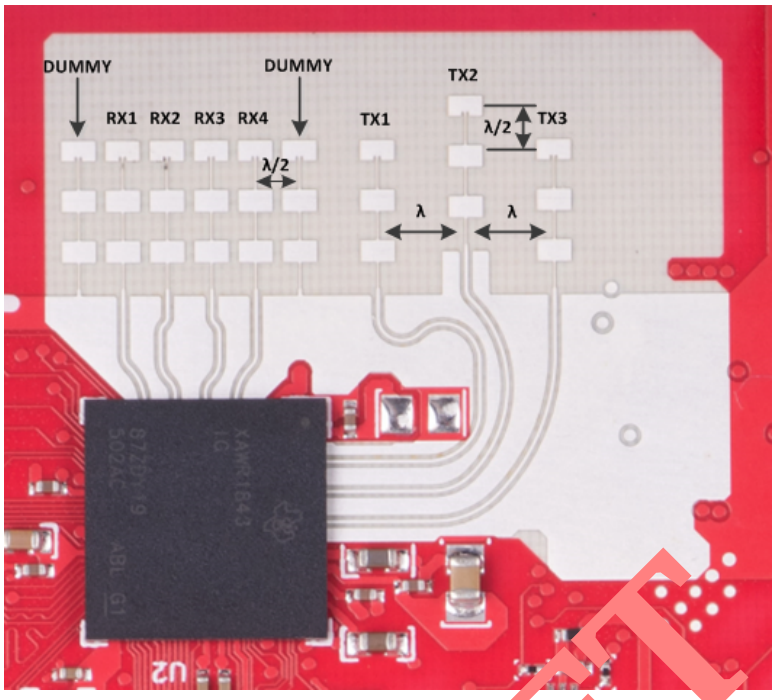


Figure 1.3. TRx Antenna

As seen in figure 1.4, the AWR1843 device can be set to operate in three different modes based on the state of the SOP (Sense-on-Power) jumpers/lines. These lines are sensed only during boot up of the AWR device.

Function/Mode	Reference	SOP Mode
Flash programming	SOP2	SOP mode 5: 101
Functional	SOP1	SOP mode 4: 001
Debug	SOP0	SOP mode 2: 011

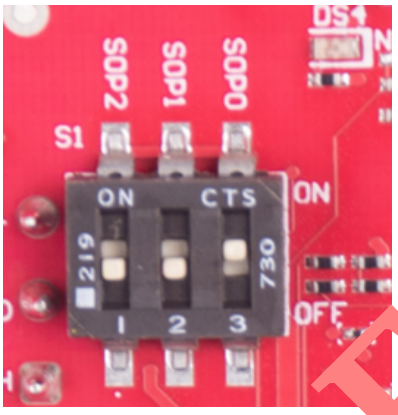


Figure 1.4. SOP on-b[illegible] Switches

1.3 System Functional Specifications

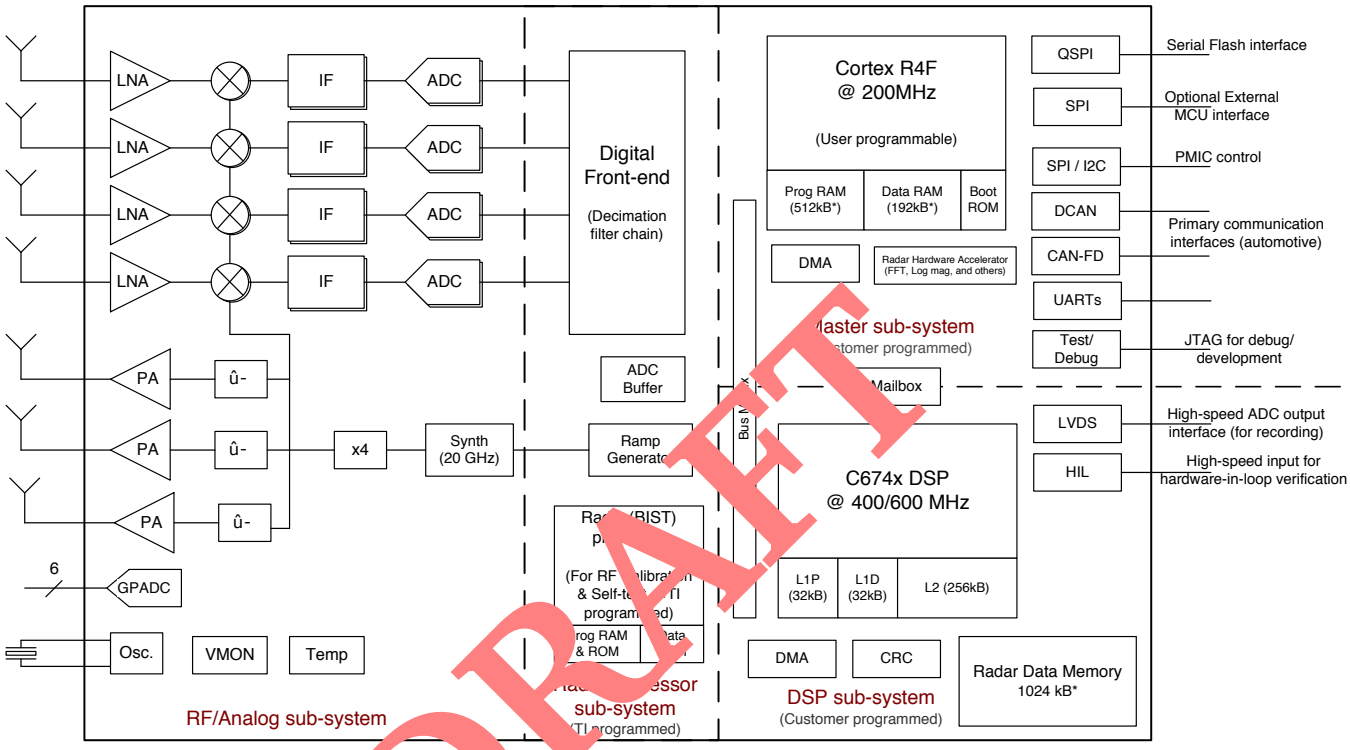


Figure 1 AWR18 3 Functional Block Diagram

The FMCW RADAR SOC System is divided into 3 subsystems as illustrated in 1.5

- **RADAR Processor Sub-system** Also known as BSS.

  BSS is responsible for the RF and analog functionality of the device. The sybsystem contains the FMCW transceiver and ARM Cortex R4F-based radio control system. Hence, the subsystem is equipped with a Built-in firmware with an API interface to the on-chip Cortex-R4F application MCU for configuration, monitoring, and calibration of the low-level RF/Analog components. Access to the Radar subsystem is provided through hardware mailboxes and a well defined API controlled by the MSS sus-system which will be discussed in this document.

  The RADAR subsystem consists of the RF/Analog subsystem and the radio processor subsystem. The RF/analog subsystem implements the FMCW transceiver system with RF and analog circuitry – namely, the synthesizer, PA, LNA, mixer, IF, and ADC. Additionally, it includes the crystal oscillator and temperature sensors. The three transmit channels as well as the four receive channels can all be operated simultaneously. On the other hand, the radio processor subsystem includes the digital front-end, the ramp generator, and an internal processor for controlling and configuring the low-level RF/analog and ramp generator registers, based on well-defined API messages from the master subsystem. This radio processor is programmed by TI, and addresses both RF calibration needs and some basic self-test and monitoring functions (BIST); this processor is not available directly for customer use. The digital front-end filters and decimates the raw sigma-delta ADC output, and provides the final ADC data samples at a programmable sampling rate.

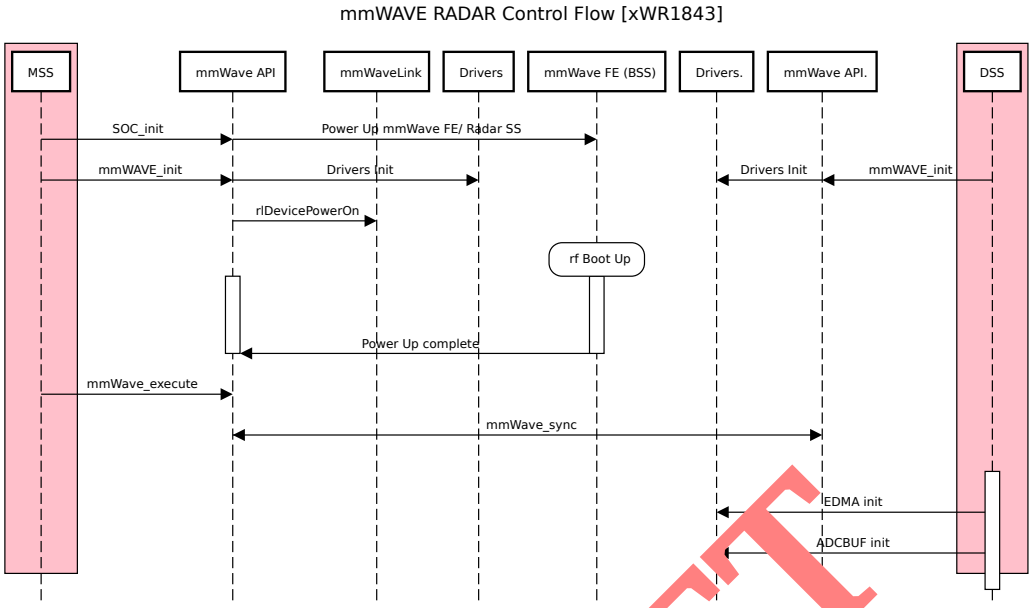
- **DSP Sub-system** Also known as DSS:

  DSS Includes TI's standard TMS320C674x (C674x) megamodule and several blocks of internal memory (L1P, L1D, and L2). The C674x DSP clocked at 600 MHz for advanced Radar signal processing. Core algorithms were implemented on the DSP sub-system such as the Range-FFT, Dopple-FFT, Angle-FFT, CFAR, dBSCAN and extended kalman filter. Details about the execution flow and the algorithms will be discussed in next sections after introducing the sensing dimensions and the enabling algorithms.

- **Master Sub-system** Also known as MSS

  MSS includes an ARM Cortex R4F processor, clocked using a MSS_VCLK clock with a maximum operating frequency of 200 MHz. User applications executing on this processor control the overall operation of the device, including radar control through well-defined API messages, radar signal processing (assisted by the radar hardware accelerator), and peripherals for external interfaces.

1.4 Application Control Sequence



The MSS application uses the *SOC_init* to initialize the device and powers up the mmWave Front End mmWave. It then uses the *MMWave_init* API to do the basic initialization of the mailbox and the other key drivers. Concurrently, DSP application on the other hand, do the same initialization procedures. After that, it waits for the (FE) front end boot-up completion. Then it uses the *MMWave_execute* API, which sets up the Inter Process Communication (IPC) to receive the data from the MSS. Both the MSS and the DSS uses synchronization to check the health of each other. DSS also initializes the EDMA and the ADC buffer for the data processing.

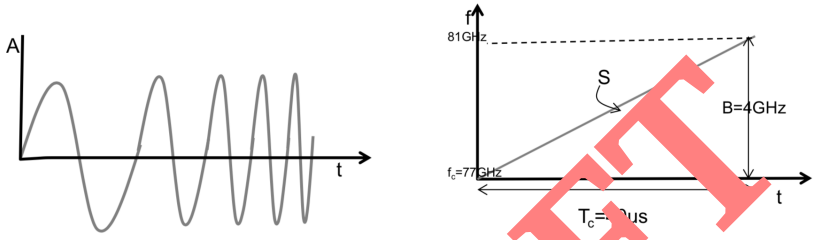
Once the initialization is complete and the MSS and the DSS are both synchronized, MSS application use the *MMWave_config* API to parse the configuration from the application to the mmWave Front End. mmWave API uses the mmWave Link API, which constructs the mailbox message and sends it to the mmWave Front End. The mmWave Front End, once it receives a message, checks the integrity of the message and sends an acknowledgement back to the mmWaveLink. In this way, all the messages are sent to the front end.

left intentionally blank

2 Sensing Dimensions

RADAR is essential for object profiling in terms of range, speed and phase. I'ts important to characterize the main component to characterize the radar performance namely chirp. In simple words, FMCW radar transmits a signal called a "chirp". A chirp is a sinusoidal whose frequency increases linearly with time, as shown in the Amplitude vs time (A-t) plot 2.1-a. Accordingly, It's very convenient to use frequency vs time (f-t) plot to represent a chirp. Hence, in 2.1-b we can observe that; a chirp is characterized by a **start frequency (fc)**, **Bandwidth (B)** and **duration (Tc)**.

The **Slope (S)** of the chirp defines the rate at which the chirp ramps up. In this example the chirp is sweeping a bandwidth of 4GHz in 40us which corresponds to a Slope of 100MHz/us. Hence, 4GHz B.W utilize the maximum device capabilities which is crucial to achieve the maximum range resolution of 3.75cm



(a) time-domain A-T plot 1 **(b)** f[illegible]omain F-T plot

Figure 2.1. A-T an[illegible]plot

In 2.2, we represent a very simple SISO tra[illegible]ceiver RADAR model to represent the execution flow. First, the synthesizer generates a ch[illegible] wh[illegible]is de[illegible]gned according to the specified requirements in terms of start frequency, bandwid[illegible]and chirp duration. The chirp is then transmitted by the TX antenna. After that, the [illegible]p[illegible]refle[illegible]d off an object and the reflected chirp is received by the RX antenna. Finally, th[illegible]X signal[illegible]d TX signal are mixed to produce the **IF signal** to be processed lated by the digital bac[illegible]nd cor[illegible]

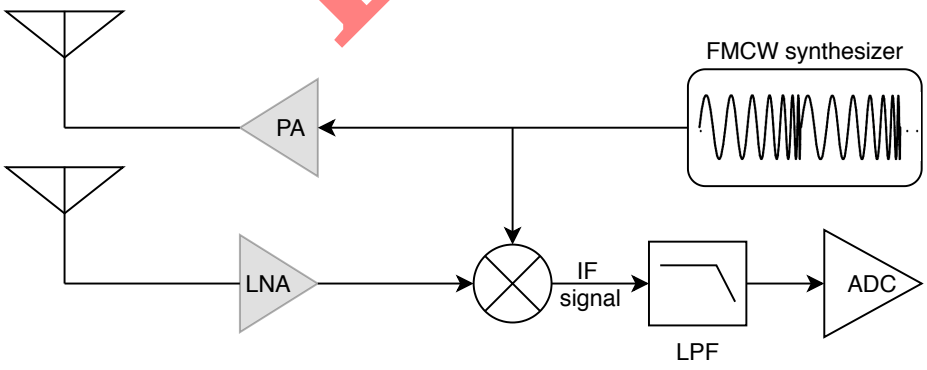


Figure 2.2. SISO RADAR Basic Block Diagram

For our purposes, a mixer is basically a simple demodulator which can be modelled as follows.

For two sinusoidal **x1** and **x2** input at the two input ports, the output is a sinusoidal with **IF (Instantaneous/Intermediate Frequency)** equal to the difference of the instantaneous frequencies of the two input sinusoids and similarly a **Phase** equal to the difference of the phase of the two input sinusoids.

$$x_1 = sin[\omega_1 t + \Phi] \otimes x_2 = sin[\omega_2 t + \Phi]$$

$$\blacktriangledown$$

$$x_o = sin[(\omega_1 - \omega_2)t + (\Phi_1 - \Phi_2)]$$

A closer look to IF, which is obtained after receiving the reflecting chirp by mixing it by the original signal

$\omega_1 - \omega_2$: IF Frequency

Hence, in 2.3 The top plot shows the TX-signal and the RX-signal that is reflected from an object. Note that the RX-signal is just a delayed version of the TX signal. τ denotes the round-trip time between the radar and the object. Also S denotes the slope of the chirp. Recall that the frequency of the signal at the mixers output is the difference of the instantaneous frequency of the TX-chirp and RX-chirp. As shown in the figure below, this is a straight line. Hence: A single object in front of the radar produces an IF signal that is a constant frequency tone. The frequency of this tone is $S\tau = S2d/c$ since $\tau = 2d/c$, where d is distance of the object and c is the speed of light.

A single object in front of the radar produces an IF signal with a constant frequency of S2d/c

Next, we are going to explore the sensing dimensions to detect range, velocity and angle of arrival on an object. And then we will design our system requirements and map it to the equivelent chirp parameters.

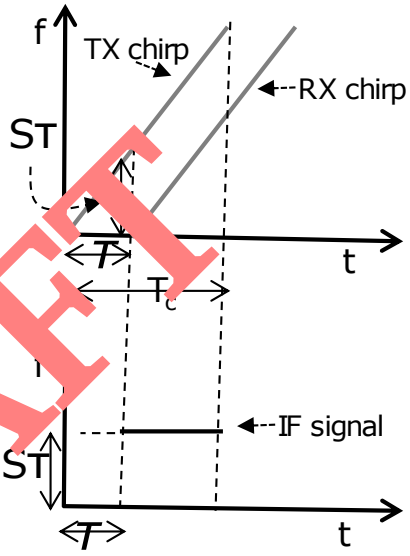


Figure 2.3. Reflected Chirp and the. IF Frequency

left intentionally blank

2.1 1D Range Estimation

Fast Fourier Transform (FFT) is the enabler algorithm for determining the range, velocity and angle of arrival (AoA) of objects in front of the RADAR respectively.

In this section we will focus on a multiple objective which the range and range resolution. Before designing our systems we need to know our objectives, hardware constraints and hence-after design our chirp accordingly to meet our objectives. The main constrains of the maximum measurable distance $dmax$ is constrained by ADC sampling frequency Fs but also it's related to the bandwidth of the IF signal, where;

$$f_{IF_{max}} = \frac{S2d_{max}}{c}$$

IF signal is typically digitized (LPF+ADC) for further processing on a DSP. Accordingly, IF Bandwidth is limited by the ADC sampling rate (Fs)

$$F_s > \frac{S2d_{max}}{c}$$

An ADC sampling rate of Fs limits the maximum range of the radar t

For **Multiple objects** scenario in front of the radar, we observe multiple reflected chirps at the RX antenna. As seen in 2.4 and 2.5. A frequency spectrum of the IF signal will reveal multiple tones, the frequency of each being proportional to the range of each object from the radar; Which means, first peak represents the lower frequency (closest object) and the last(third peak) represents the highest frequency (furthest object).

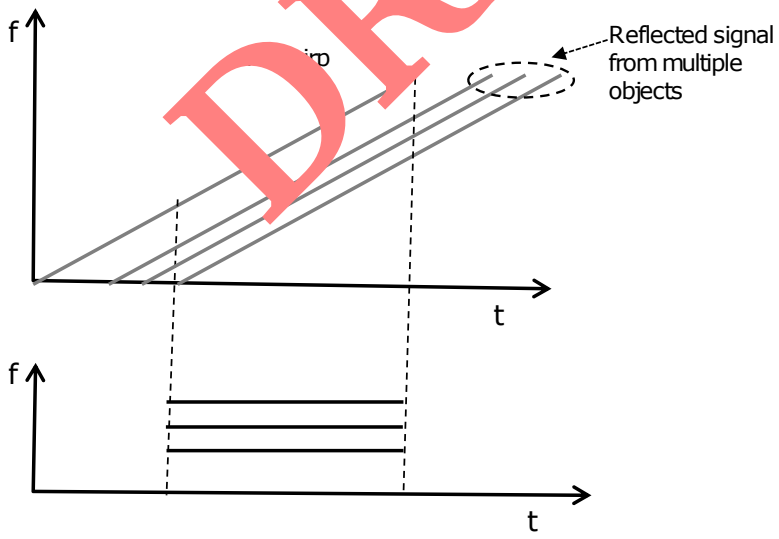


Figure 2.4. Multiple tones in the IF signal

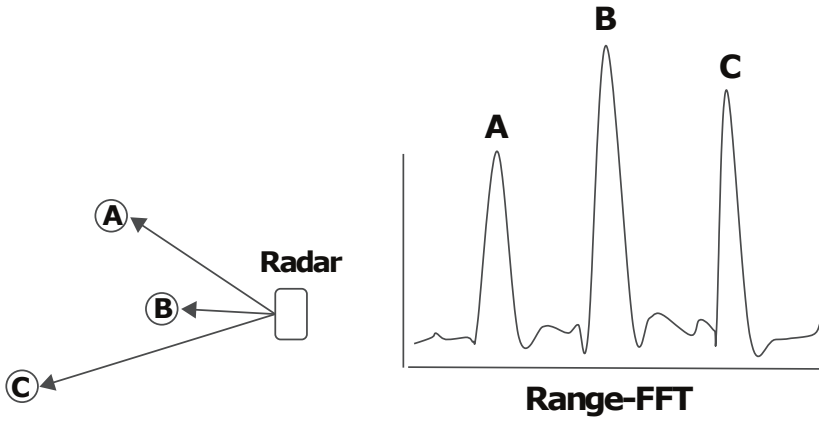
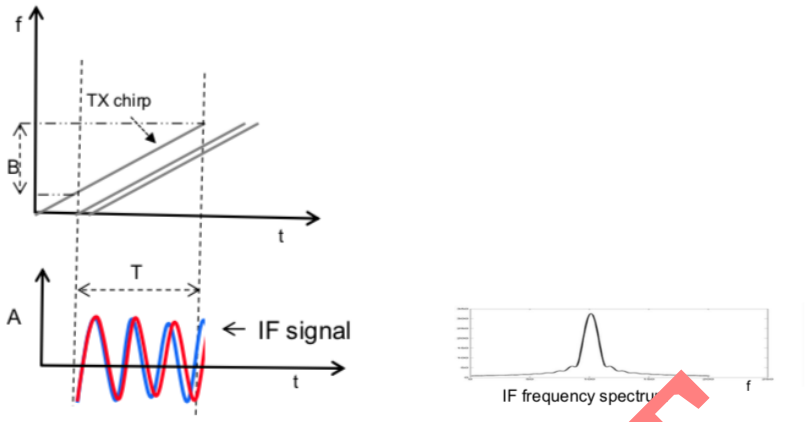


Figure 2.5. IF frequency spectrum; FFT processing translates peak frequencies from signals bounced back from an object into range measurements

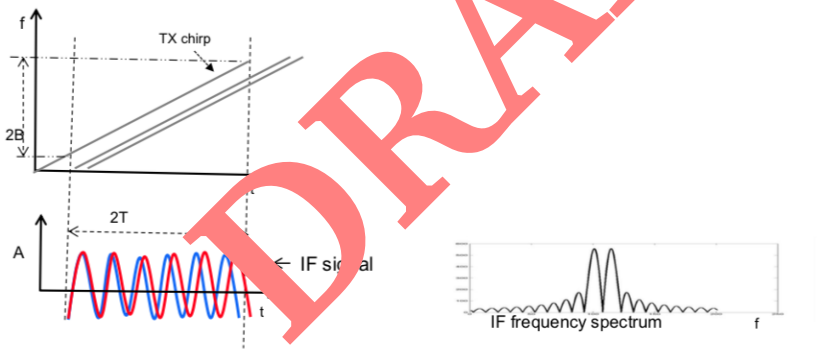
Now, the next natural question is **range resolution**. That is, how close can two of these objects be and still be resolved as two peaks in the IF spectrum. In 2.6 objects cannot be resolved because of the observation window is too narrow. But its very clear in 2.7 where we increased the observation window. aka the bandwidth we are able to resolve the objects.



(a) Multiple tones in the IF signal

(b) IF frequency spectrum (A Fourier transform: 1D (range) FFT)

Figure 2.6. Range Resolution refers to the ability to resolve two closely spaced objects.Here, the two objects are too close that they show up as a single peak in the frequency spectrum.)



(a) Multiple tones in the IF signal

(b) IF frequency spectrum (A Fourier transform: 1D (range) FFT)

Figure 2.7. The two objects can be resolved by increasing the length of the IF signal. Note that this also proportionally increases the bandwidth. Thus intuitively: Greater the Bandwidth, better the resolution)

Recall that; An object at a distance d results in an IF tone of frequency S2d/c. Two tones can be resolved in frequency as long as the frequency difference $\Delta f > \frac{1}{\Delta t}$ Now, to derive an equation for the range resolution of the radar, we need to specify on what parameters does the range resolution depend i.e. Chirp Duration, Bandwidth, Slope.

For two objects separated by a distance Δd, the difference in their IF frequencies is given by $\Delta f = s2\Delta c$ Since the observation interval is Tc, this means that

$$\Delta f > \frac{1}{T} \longrightarrow \frac{s2\Delta d}{c} > \frac{1}{T_c} \longrightarrow \Delta > \frac{c}{2ST_c} \longrightarrow \frac{c}{2B}$$

where $B = ST_c$

The Range Resolution d_{res} depends only on the bandwidth swept by the chirp $d_{res} = \frac{c}{2B}$

Bandwidth	Range Resolution
4 GHz	3.75cm
2 GHz	7.5cm
1 GHz	15cm
600 MHz	25cm

Longer the observation period => better the resolution. In general, an observation window of T can separate frequency components that are separated by more than 1/T Hz.

It's Important to realize the design parameters (the chirp bandwidth and the IF bandwidth). Hence, larger chirp bandwidth gives a better range resolution. On the other hand, larger IF Bandwidth means faster chirps, therefore better maximum distance

2.1: IF & Distance

An object at a distance d produces an IF frequency of:

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

2.2: dres & BW

Range resolution (dres) depends on the chirp bandwidth (B):

$$d_{res} = \frac{c}{2B}$$

2.3: Fs & Dmax

The ADC sampling rate Fs, limits the max range (dmax) t

$$d_{max} = \frac{F}{2S}$$

From 2.1, we can realize clearly how the sampling frequen affecting the maximum range. In fact, Fs is not the only parameter limiting th aximu range. However, transmitter power and target characteristic are main constraints Next page, in 1 we illustrated and defined the power equations and then we defined how the ma in equation $dmax$ is refactored according to the sampling frequency and power c traint

This is necessary in order to ave a tru measure of the of the received signal, therefore we introduce the SNR in relation with t antenn parameters, target distance and target characteristics (by evaluating the Radar Cross Secti RCS).

It's Important to note here, that in the next page, we are taking about ideal scenarios by considering the isotrobic antennas (which is a fiction/impractical. In any practical system, Effective radiated power (e.r.p.) (in a given direction) is considered)

Terms, Units	Equation	Definition
`Radiated Power Density` W/m^2	$\frac{P_t G_{tx}}{4\pi d^2}$	The product of the power supplied to the antenna Pt and the antenna gain Gt in a given direction relative to an isotropic antenna (absolute or isotropic gain). Hence the terms, d is the distance to the target. r is the range from antenna (i.e. radius of sphere)
`Power reflected by object` W	$\frac{P_t G_{tx} \sigma}{4\pi d^2}$	Same as above, but we include the scattering rays comming from the object by taking into considerationt he σ Radar Cross Section (RCS) area
`Effective aperture area of RX antenna` W/m^2	$\frac{G_{rx} \lambda^2}{4\pi}$	A_{RX}. Using effective area of aperture to measure the rx antenna performance (how effective an antenna is at receiving power). In other words we use it to know how much power is captured from the plane wave and delivered by the antenna. This area factors in the losses intrinsic to the antenna (ohmic losses, dielectric losses, etc.).
`Power density at RX antenna` W/m^2	$\frac{P_t G_{tx} \sigma}{(4\pi)^2 d^4}$	Unlike the transmitter, which it's power that is radiated from an isotropic antenna will have a uniform power density (power per unit area) in all directions. The reciver will have power loss/dissipation. Hence, we take into our consideration the Radar Corss Section (RCS).
`Power captured at RX antenna` W/m^2	$\frac{P_t G_{tx} \sigma A_{RX}}{(4\pi)^2 d^4}$ $\frac{P_t G_t X \sigma G_{RX} \lambda^2}{(4\pi)^3 d^4}$	The total power recieved by the reciever is a result of the Effective aperture area of the rx antenna multiplied by the power density)
`Noise Floor` W	$K_B T \Delta f$	where K_B is the Boltzmann constant, T is the antenna temperature and Δf is the measured bandwidth)

SNR is the ratio between the the received signal power to the level of the background noise (Noise Floor)

$$SNR = \frac{P_r}{P_{noise}}$$

Where the Pr represents the concentrated power from a single/multiple target and $Pnoise$ represents the unwanted power from the background noise. It is important to understand that the background noise can be generally understood as the combination of unwanted or disturbing energy from different type of sources. From all the noise sources, the two most relevant ones are the radar thermal noise and radar clutter. Radar thermal noise (also known as Johnson or Nyquist noise) can be interpreted as a white signal (AWGN) with uniform spectral power density across the entire frequency spectrum. On the other side, radar clutter can be divided into two categories: mainlobe and sidelobe [2]. Mainlobe clutter is the consequence of unwanted ground returns within the radar beam-width where sidelobe clutter is the combination of the undesired returns from any other direction outside the main-lobe. Ground return is the most common source of side-lobe clutter due to the high amount of reflective area at close range. The received power (Power captured at RX antenna) and the Noise Floor can be seen in 2.1. And by substituing we can get the SNR as seen below 2.1 Hence, the terms $Tmeas = \frac{1}{\Delta f}$ is the measurement period and F represents the antenna noise figure.

$$SNR_{out} = \frac{SNR_{in}}{F}$$

SNR_{in} is sufficient if we consider the antenna network and its elements such as attenuators, transmission lines, filters, among others as noise-free. For a complete analysis, consider the antenna as a "lossy" network. For this purpose, the definition of the noise figure (F) as the ratio between the input and output SNR is necessary. Considering.

2.4: SNR

$$\frac{\sigma P_t G_{TX} G_{RX} \lambda^2 T_{meas}}{(4\pi)^3 d^4 KTF} W$$

There is a minimum SNR (SNRmin) that is required for detecting an target.There is a minimum SNR (SNRmin) that is required for detecting an target. Choice of SNRmin is trade-off between probability of missed detections and probability of false alarms. Typical numbers are in the 15dB-20dB range. Therefore, given an SNRmin, the maximum distance that can be seen by the radar can be computed as shown in 2.1

2.5: dmax

$$\left(\frac{\sigma P_t G_{TX} G_{RX} \lambda^2 T_{meas}}{(4\pi)^3 SNR_{min} KTF}\right)^{\frac{1}{4}}$$

To Summarize the Synthesis and Aquasition process

- Synthesizer generates a chirp
- Chirp is transmitted. RX sees delayed versions of this chirp
- IF signal consists of multiple tones, the frequency (f) of each tone being proportional to the distance (d) of the corresponding object
- The IF signal is passed through LPF then it's digitized. The ADC must support an IF bandwidth of $\frac{S2d_{max}}{c}$
- An FFT is performed on the ADC data. The location of peaks in the frequency spectrum directly correspond to the range of objects (aka Range FFT or 1D FFT)

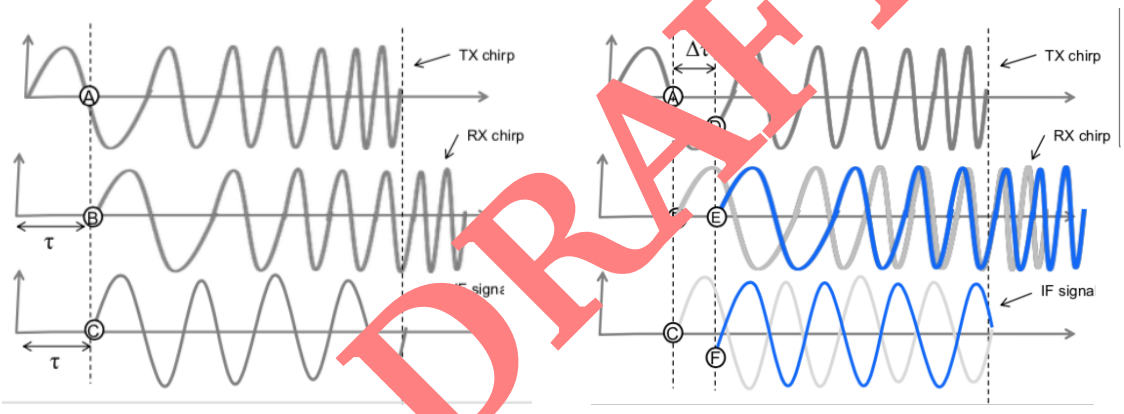
left intentionally blank

2.2 2D Velocity/Doppler Estimation

This section addresses the second dimension estimation of multiple objects in the RADAR field of view. Accordingly, we discuss system model of the Velocity/Doppler detection algorithm using a 2D FFT in FMCW radar systems.

In a certain use-case scenario where the RADAR FoV is consisting of two or more targets with different relative velocities which are equidistant from the radar, performing the 1D-FFT (Range-FFT) will give us miss-information regarding how many objects in our sight. Accordingly, miss-detection, miss-identification and miss-localization could cause disasters. This problem is because the fact, that targets will appear as a single peak in the frequency domain due to limited measurements information. Hence, after range-FFT processing, we get single IF frequency tone since all targets are equidistant. Therefore, further sensing information are needed to be extracted and to obtain the objects relative velocities. In order to detect the objects relative radial velocities, we need to extract the phase of the IF signal.

The answer for enabling the second sensing parameter relies on exploiting the Doppler effect and therefore, getting the radial relative velocity information. Hence, while processing returned signal, the radial velocity will be estimated based on change in phase. To get more intuition into the nature of the IF signal, lets look at the 'A-t' plot 2.8



(a) round τ delay is constant **(b)** round τ delay changes by a small amount $\Delta\tau$

Figure 2.8. A-T and F-T plot

For an object at a distance d from the radar, the IF signal is a sinusoid, where $f = \frac{S2d}{c}$ and ϕ_o is the phase of the IF signal in point c in 2.8.a where the round delay is constant. Although when the round delay changes by a small amount ϕ will equal then to phase of tx chirp signal - phase of rx signal.

$$Asin(2\pi ft + \Phi)$$

in 2.8.b we can see, what happens of the phase of the IF signal if the object moves by a small amount such that the round-trip delay, τ, changes by the delta tau. The new RX signal, represented in blue curve here, is going to shift by an amount delta tau. Also, the IF signal is correspondingly

going to change.

Now, the starting phase of the new IF signal– that is, the phase of the IF signal at point F– is going to be the difference of the phase at D and the phase at E.

Now, the phase of the RX chirp at E is going to be the same as the phase of the earlier RX signal at B. But the phase of the TX chirp at D is going to be the earlier phase at A with an additional phase offset of 2pi fc delta tau. And this is because the TX chirp would have traversed an additional phase of 2pi fc delta tau during this period delta tau. And this additional phase, which is the phase difference between the point A and D, is going to get directly reflected in the phase of the IF signal– that is, the phase at F.

Phase difference between A and D as well as between C and F is.

$$\Delta\phi = 2\pi f_c \Delta\tau = \frac{4\pi\Delta d}{\lambda}$$

Where $\Delta\tau = \frac{2\delta d}{c}$ and $\lambda = \frac{c}{f_c}$

So now, for a single object in front of the radar, the IF signal is a tone with a frequency that is proportional to the distance of the object and also has a starting phase which has the property that it changes linearly with small changes delta d in the distance of the object where $Asin(2\pi ft + \Phi)$.

Consider the chirp where S=50MHz/us and Tc=40us. What happens if an object in front of the radar changes its position by 1mm

(for 77GHz radar $1mm = \frac{\lambda}{4}$)

The phase of the IF signal changes by $\Delta\phi = \frac{4\pi\Delta d}{\lambda} = 18$

The frequency of the IF signal changes by $\Delta = \frac{S2\Delta}{c} = \frac{50*10^{12}*2*1*10^{-3}}{3*10^8} = 333Hz$

333Hz looks like a big number, but in one observation window this corresponds to only additional $\Delta f T_c = 333*40*10^{-6} = 0.013$ Cycles. This change would not be discernible in the frequency spectrum

The phase of the IF signal is very sensitive to small changes in object range

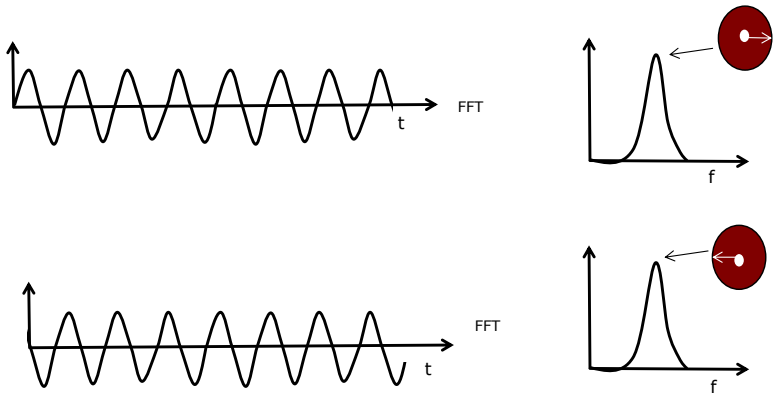


Figure 2.9. An object at certain distance produces an IF signal with a certain frequency and phase. Hence, small motion in the object changes the phase of the IF signal but not the frequency

Now we have all necessary information to knwo how to measure velocity.

Basically, we design our system to transmit N equi-spaced chirps. (This unit is typically called a FRAME). An FFT on the sequence of phasors corresponding to the range-FFT peaks resolves the two objects. This is called a doppler-FFT. In case scenario where there are two objects in front of the radar, we can generalize their velocity according to the following equations, where $\omega 1$ and $\omega 2$ correspond to the phase difference between consecutive chirps for the respective objects and v1, v2 are thier velocities.

$$v_1 = \frac{\lambda \omega 1}{4\pi T_c}$$

$$v_2 = \frac{\lambda \omega 2}{4\pi T_c}$$

Before going into details, in a visualization for the Doppler effect and the phase changing accordingly for a single moving target in front of the radar. Processing the recieved signal using the 2D-FFT will resolve the object velocity. Hence, The measured phase difference ω between the two chirps corresponds to a motion in the object of vTc (moved by a velocity v during time Tc. Hence vTc is the distance the object moved during time Tc with speed v thats why we substituded in w below where $vTc = \Delta d$

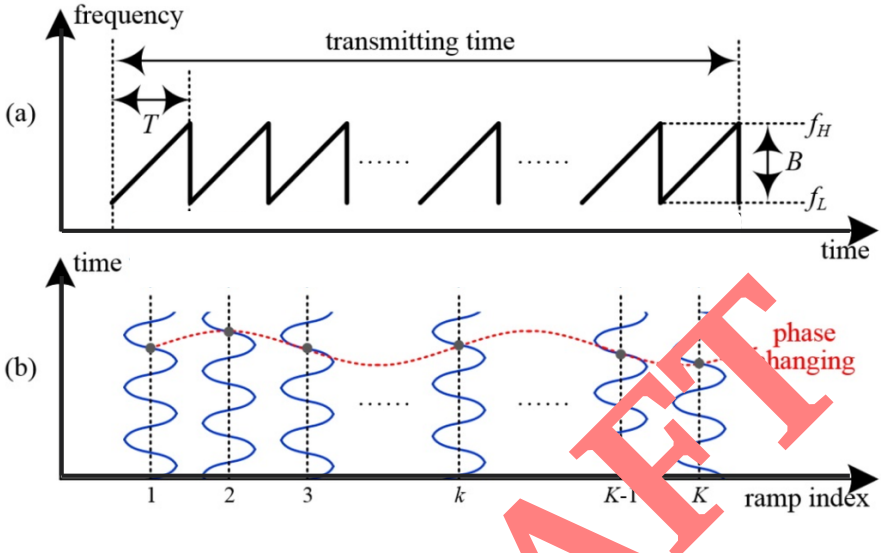


Figure 2.10. (a) transmitted signal in the frequency-time domain, and (b) reflected chirp signal for a single moving target in ramp/chirp index domain. Here, T is chirp duration, B is bandwidth, f_H is maximum instantaneous carrier frequency, f_L is the lowest carrier frequency, and K is the number of chirps. [3]

Velocity Resolution

The Velocity Resolution of the Radar is inversely proportional to the framce time T_f and it's given by

$$V_{res} = \frac{\lambda}{2Tf}$$

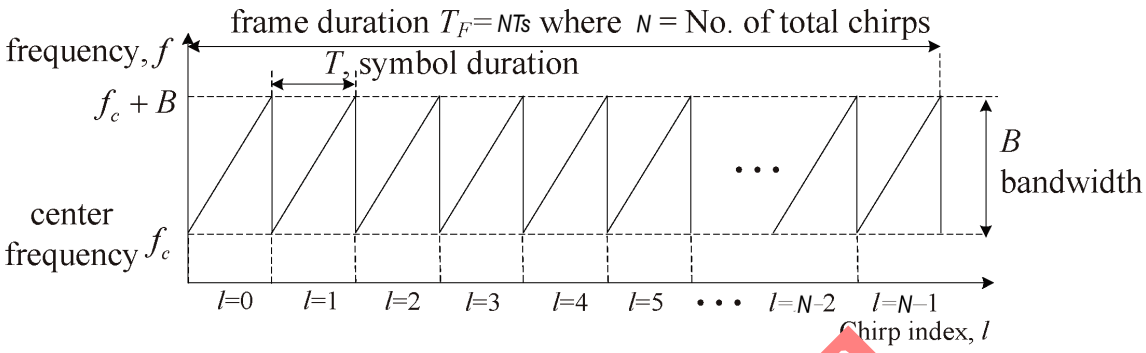


Figure 2.11. The Velocity Resolution of the Radar is inversely proportional to the framce time Tf

FFT on a sequence of length N, can separate two frequencies $\omega 1$ and $\omega 2$ as long $|\omega 1 - \omega 2| > \frac{2\pi}{N}$. This can be used to derive vres

$$\Delta\omega = \frac{4\pi\Delta v}{\lambda}$$

$$\Delta\omega > \frac{2\pi}{N}$$

$$\Rightarrow \Delta v > \frac{\lambda}{2NT_c}$$

DESIGN TRADE-OFF A frame in Radar A has a frame of length Tf consisting of 2N equispaced chirps. And a frame in Radar B has a frame of the same length Tf but half the number of chirps (N).

So Tf is constant which means vres is the same for both radars, but radar A can achieve maximum velocity measurments compared to B since Increasing the number of chirps is inversely proportional to the Tc therefore we have a lower duration if we increased the N. And Tc is inversely proportional to the Vmax therefore lower Tc gives a higher Vmax

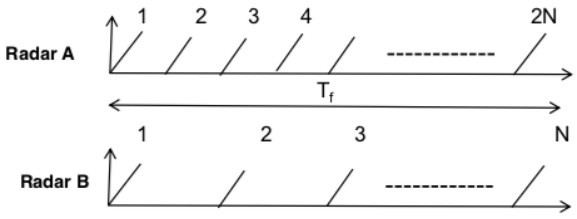


Figure 2.12. velocity resolution and maximum velocity design trade-off

left intentionally blank

2.3 3D Angle of Arrival Estimation

Angle of Arrival (AoA) estimation is a critical parameter for objects localization and tracking. Consider a scenario where you have two objects at the same range (Equidistant objects) and same relative velocity to the RADAR as seen in 2.13. The range- velocity plot resulting from the 2D-FFT will have single peak, since they have the same range and velocity. Hence, we need to deteremine their angle to know their exact location. **To separate these two objects we need multiple antennas to estimate the angle of arrival (AoA).**

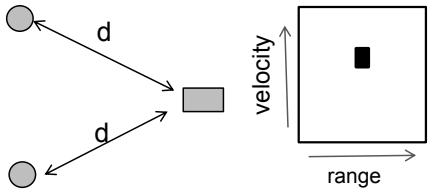


Figure 2.13. Range-velocity of two objects equidistant from the radar and with the same velocity relative to the radar.

Similarly, you may also consider a multiple-object scenario in which two or more targets are approaching the radar at the same range and with the same relative velocity. Again, in both cases we are going to have single-tone peak in range-Doppler map. To be able to separate objects moving at the same distance and velocity, we use further signal processing algorithm so called 3D FFT (Angular FFT) to get the Angle information. It is important to re-emphasize, that angle estimation needs at least two receiving antennas. In this extend, we are going to generalize a standard way for measuring AoA of multiple objects at the same range and velocity. Accordingly we will start answering the questions regarding single vs multiple objects scenario, maximum field of view and angular resolution as illustrated in [4].

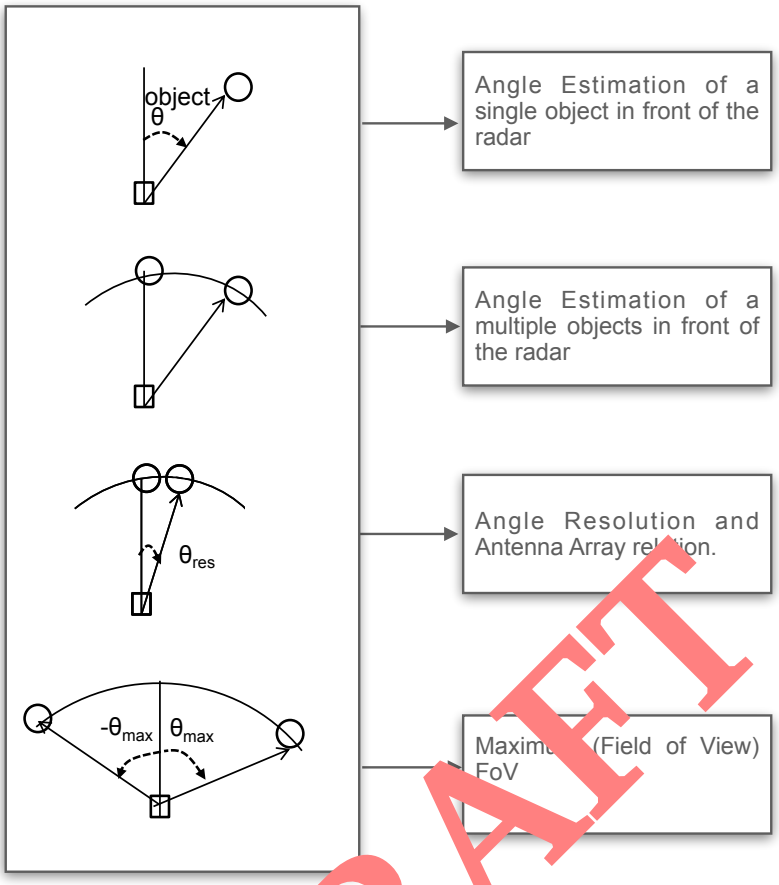


Figure 2.14. Common scenarios to address in the sensing dimension (Angle Estimation)

2.3.1 Naive Approach: Single Object Angle Estimation

Before, we go deep; sometimes it's very simple to measure the Angle for a single object in front of the radar. Recall (from velocity estimation) that, the phase of the IF signal is very sensitive to small changes in the distance of the object, specifically a small change Δd in the distance of the object results in a phase change ω given by

$$\omega = \frac{4\pi\Delta d}{\lambda}$$

.

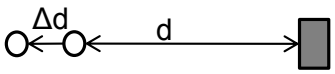


Figure 2.15. Recall that a small change in the distance of the object result in a phase change ω in the peak of the range-FFT

Angle estimation exploits a similar concept, where it requires at least 2 RX antennas. As seen in 2.16 what is exploited here is the differential distance (small change in distance Δd) of the object to each of these antennas. The transmit antenna transmits a chirp which is then reflected off the object. Hence, for 2 RX antennas we will have 2 rays; one ray going from the object to the first RX antenna and the second rayto the second RX antenna.

Estimating the angle of arrival is achieved by measuring the phase difference across two RX antennas as in this example **??**, the ray to the second RX antenna has to travel a little longer, an additional distance of Δd, to get there. This additional distance results in an additional phase of omega (omega is the phase difference of the signal across both antennas) equal to

$$= \frac{2\pi\Delta d}{\lambda}$$

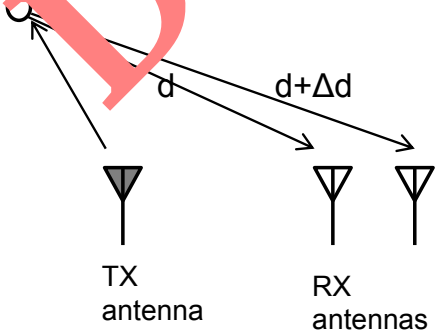


Figure 2.16. Angle Estimation requires at least 2 RX antennas. The differential distance from the object to each of the antennas results in a phase change in the 2D-FFT peak which is exploited to estimate the angle of arrival

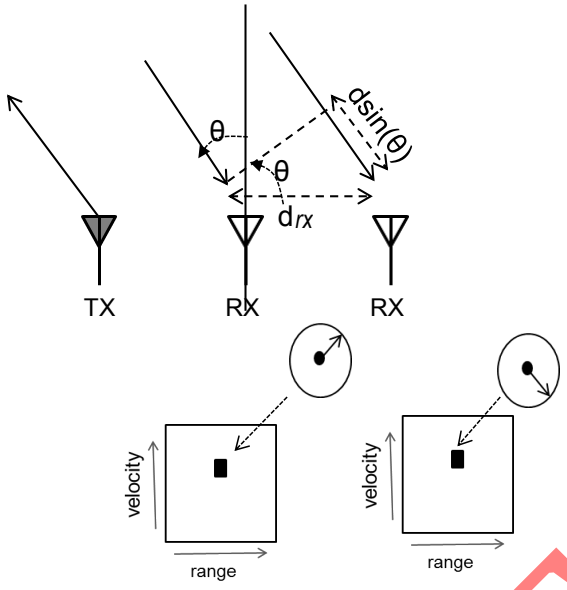


Figure 2.17. The dependency of this additional distance on the angle of arrival.

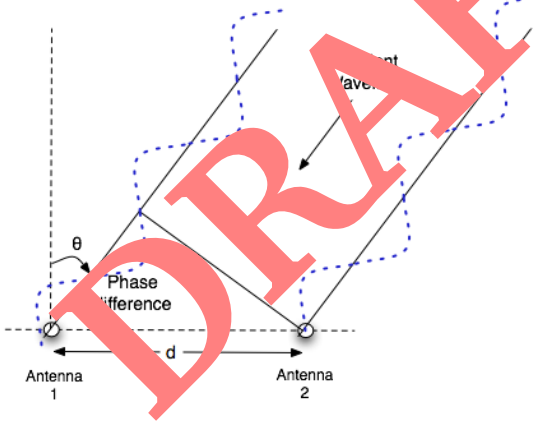


Figure 2.18. The dependency of this additional distance on the angle of arrival. d :=d_{rx}

- To measure the AoA of a single object using 2 RX antennas
  - TX antenna transmits a frame of chirps
  - The 2D-FFT corresponding to each RX antenna will have peaks in the same location but with differing phase. $d_{rx}sin(\Theta)$
  - The measured phase difference ω can be used to estimate the angle of arrival of the object

$$\omega = \frac{2\pi d_{rx} sin(\Theta)}{\lambda}$$

$$\Theta = sin^{-1}(\frac{\lambda\omega}{2\pi d_{rx}})$$

Henceforth, d_{rx} refers to the distance between two consecutive rx antennas and theta is the angle of arrival of the object with respect to the radar and $d_{rx}sin(\Theta)$ is the additional distance of the ray to the second antenna compared to the first antenna.

The scenario is that, the transmitter antenna transmits a frame a chirps, and the data is received at each of these antennas where each of these antennas processes the data to create a 2D-FFT matrix with a peak corresponding to the range and velocity of the object as seen in 2.17. Note that the location of the peak is going to be virtually identical for both of these 2D-FFTs. The peak location is very insensitive to small changes in distance between the radar and the object. However, the phase difference between these two peaks is going to be given by

$$\omega = \frac{2\pi d sin(\Theta)}{\lambda}$$

And once you have measured this phase difference by comparing these two signals, the signals at these two peaks, you can just invert this equation to calculate the angle of arrival.

$$\Theta = sin^{-1}(\frac{\lambda\omega)}{2\pi d})$$

Estimation accuracy depends AoA

If you examine this expression $\omega = \frac{2\pi d sin(\Theta)}{\lambda}$ the relationship between the quantity that we want to estimate, namely Θ/AoA, and the measured phase difference ω is a **nonlinear** one because of $sin(\Theta)$.

Reecalling the expression from velocity estimation, which is given by $\omega = \frac{4\pi V T_c}{\lambda}$ It's clear that the estimated phase difference depends linearly on the velocity. Likewise, the relationship between the IF frequency and the range is linear.

Considering the plot in 2.19 of $sin(\theta)$ versus θ, close to $\theta = 0$, $sin(\Theta)$ is very sensitive to θ. Therefore, small changes in θ produce comparable changes in $sin(\theta)$. But this sensitivity decreases as theta increases especially close to $\theta = 90$, $sin(\theta)$ becomes quite insensitive to θ. Hence, estimation of θ is more error prone as θ.

Angle estimation is best when the object is directly in front of the radar. That is theta equal to 0. And as theta increases and approaches 90 degrees, the estimation accuracy decreases, of course, in the presence of noise.

RANGE and VELOCITY have linear relationships w.r.t distance and angle respectively, BUT this is not the case in AoA estimation where we have non-linear relationship

- Note that the relationship between ω and Θ is a non-linear relationship unlike in the case of velocity where $\omega = \frac{2\pi d sin(\Theta)}{\lambda}$
- At $\Theta = 0$, ω is most sensitive to changes in $\Theta = 0$. The sensitivity of ω, to Θ, reduces as Θ, increases (becoming 0 at $\Theta = 90$)
- Hence estimation of Θ is more error prone as Θ increases.

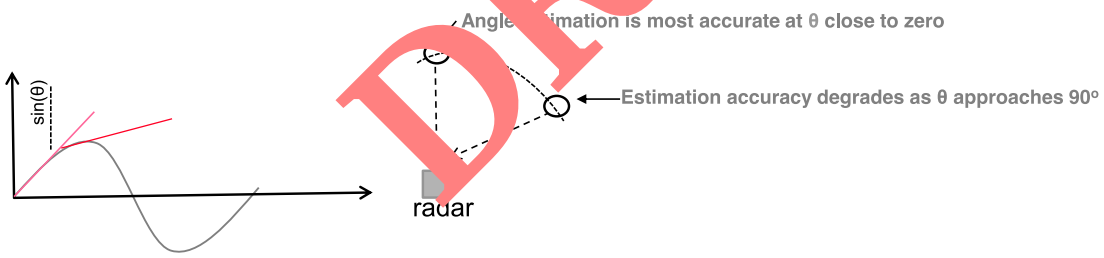


Figure 2.19. Angle non-linear relation with respect to Theta

2.3.2 Angle FFT Approach: Multiple Objects Angle Estimation

The Naive phase comparison technique will not work since the value at the peak has many phasor components from many objects which should be resolved. To measure the angle of multiple objects in front of the radar, we use antenna array by increasing the number of RX antenna from two and create an area of N antennas; N receive antennas. The 2D FFT at all these antennas is going to have a peak at the same location. Hereinafter, the signal at this series of peaks is going to create a discrete sequence consisting of two rotating phases, as shown in 2.20

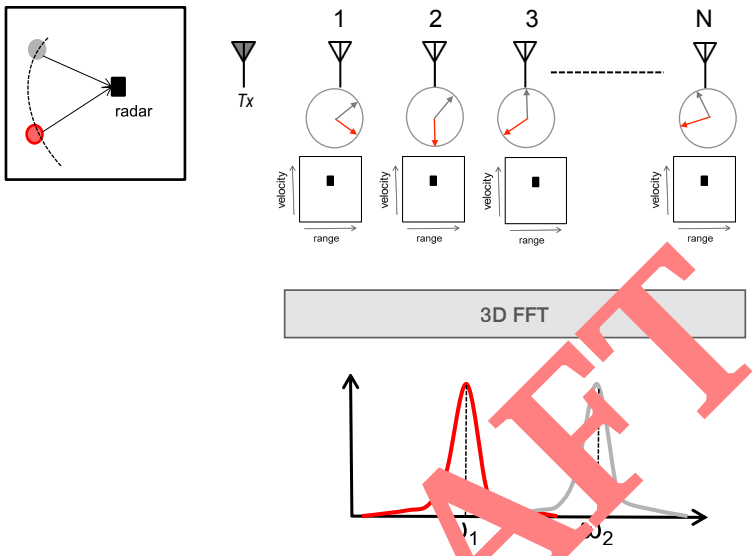


Figure 2.20. Resolving objects in space; where omega1 and omega2 correspond to the phase difference between consecutive chirps for the respective objects

An FFT on the sequence of phasors is corresponding to the 2D-FFT peaks resolves the two objects. This is called angle-FFT (3D FFT). FFT on the sequence will show up as two peaks at $\omega 1$ and $\omega 2$, where $\omega 1$ and $\omega 2$ are the rates of rotation in radians per sample for the two objects. Therefore, we can read the location of these two peaks from the FFT, and then just back-calculate the angle of arrivals for the two objects, as shown here. We call this particular FFT here, which is performed across RX antennas, as the angle FFT.

$$\Theta_1 = sin^{-1}(\frac{\lambda\omega 1}{2\pi d})$$

$$\Theta_2 = sin^{-1}(\frac{\lambda\omega 2}{2\pi d})$$

2.3.3 Angle resolution

Angle resolution Θ_{res} is the minimum angle separation for the two objects to appear as separate peaks in the angle-FFT. in Other words, for two objects with AoA of Θ and $\Theta + \Delta\Theta$ relative to the radar. How small can $\Delta\Theta$ get with the two objects showing up as distinct peaks in the angle FFT.

- An object with an angle of arrival of Θ has a discrete frequency of ω given by $\omega = \frac{2\pi dsin(\Theta)}{\lambda} < \pi < sin^{-1}(\frac{\lambda}{2d})$ in the angle-FFT
- The criterion for separation in the frequency domain is that the separation of angular frequencies $\Delta\omega > \frac{2}{N}$ where, N being the number of samples in the FFT. So we get angle resolution as below.

$$\Theta_{res} = \frac{\lambda}{Ndcos(\Theta)}$$

Two objects with angles of arrival of Θ and $\Theta+\Delta\Theta$ will have their discrete frequencies separated by this expression:

$$\Delta\omega = \frac{2\pi d}{\lambda} * (sin(\Theta + \Delta\Theta)) - sin(\Theta$$

Now to simplify this expression, remember from calculus that the derivative of sine theta is cos theta, which means that the ratio of change in sine theta for a corresponding small change delta theta is equal to cos theta.

$$\frac{(sin(\Theta + \Delta\Theta) - sin(\Theta)}{\Delta\Theta} = cos(\Theta)$$

$$\Delta\omega = \frac{2\pi d}{\lambda} cos(\Theta)\Delta\Theta$$

Also recall from the properties of discrete Fourier transforms that we discussed DFT, that for our two peaks to be separated in the frequency domain, the separation of their angular frequencies should be greater than 2 pi by N, N being the number of samples input to the FFT (Observation Window or Bandwidth), which in this case is also **N** is the number of antennas in the array.

$$\Delta\omega > \frac{2\pi}{N}$$

$$\frac{2\pi d}{\lambda} cos(\Theta)\Delta\Theta > \frac{2\pi}{N}$$

$$\Delta\Theta > \frac{\lambda}{Ndcos(\Theta)}$$

$$\Theta_{res} = \frac{\lambda}{Ndcos(\Theta)}$$

Also note that since N is the number of antennas in the array, and d is the space in between consecutive antennas, N times d is actually the length of the antenna array. So you can say that the

angular resolution increases as the length of the receiver antenna array increases.

Now, resolution is often quoted assuming that the interval antenna spacing is lambda by 2 and that theta is 0. And substituting that in this expression you get this expression for the angle of resolution:

$d = \frac{\lambda}{2}$ and $\Theta = 0$

$$\Theta_{res} = \frac{2}{N} radians$$

One thing to remember is that this expression here is actually in units of radians. So you would need to multiply this by 180 divided by pi to convert it to degrees.

Resolution best at $\Theta = 0$ (Why?)

In 2.21 For two objects separated by delta theta, their angular frequencies in the angle FFT are actually further apart at theta equal to 0 and come closer to each other as theta increases, even though the separation delta theta is the same in both cases.

As you might have probably figured out by now, angle estimation and velocity estimation in FMCW radar depend on very similar concepts. Actually, the mathematical underpinnings are almost identical.

Figure 2.21. Resolution degrades as AoA increases

By synthesising a 12 virtual Rx antennas will be able to achieve 7 and 14.3 degrees in azimuth and elevation respectively

2.3.4 Angular Field of View

In 2.22 this representing the phasor corresponding to the peak of the 2D-FFT for an object to the left of the radar, this phasor moves anticlockwise when you go from the first RX antenna to the second RX antenna. Likewise, for an object to the right of the radar, the phasor moves clockwise.

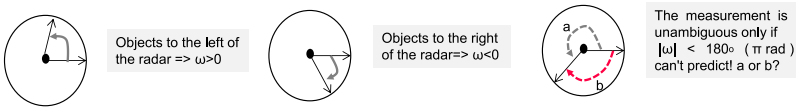


Figure 2.22. Unambiguous measurement of angle

So the measurement of this movement is unambiguous, only as long as the movement either in the clockwise or in the anticlockwise direction is less than 180 degrees or pi radians. If that were not the case. One cannot say if this movement is because of movement of a degrees in the anticlockwise direction or a movement of b degrees in the clockwise direction.

Unambiguous measurement of angle => $|\omega| < 180$

So basically, unambiguous measurement of velocity requires that the phase change between the two antennas be less than 180 degrees. And plugging in the value of omega from the previous slides and rearranging this a little bit, we see that the maximum angle that can be measured by the radar has to be less than this expression over here. Sine inverse of lambda by 2d, while d is the spacing between the two antennas.

$$\omega = \frac{2\pi d sin(\Theta)}{\lambda} < \pi < sin^{-1}(\frac{\lambda}{2d})$$

Therefore, The maximum field of view that can be serviced by two antennas spaced d apart is

$$\Theta_{max} = sin^{-1}(\frac{\lambda}{2d})$$

A spacing d of $\frac{\lambda}{2}$ results in the largest field of view (+/- 90 degree)

Summary of Angle Estimation This is summarize the angle estimation process and places it in context within the FMCW radar system. So you have synthesizer synth, also known as the LO or local oscillator, which generates a chirp. The chirp is transmitted by the transmit antenna. It is reflected off objects in front of the radar. And the reflective signal is received at each of these receive antennas.

The synth signal is routed to each of these RX antennas, and the received signal is mixed with the LO signal to create an IF signal. The IF signal is low pass filter and samples by an ADC. This ADC data is processed across an entire frame to create a 2D-FFT grid, one such 2D-FFT grid for each of these RX antennas. Peaks in this 2D-FFT grid correspond to the range and velocity of objects. An FFT across the corresponding peaks in this series of TX antennas is called an angle FFT, and peaks in this angle FFT directly correspond to the angle of arrival of objects.

- A single TX, RX chain can estimate the range and velocity of multiple objects.
- Besides range, angle information is needed for localization
- Multiple RX antennas are needed for angle estimation. Hence the 2D FFT grid is generated at each RX chain (corresponding to each antenna). FFT on the corresponding peak across antennas is used to estimate the angle.

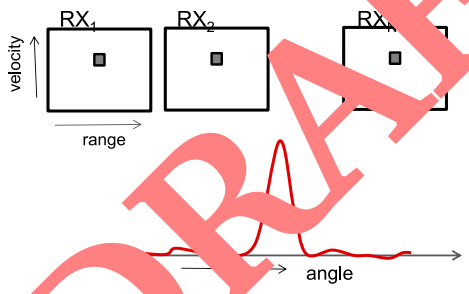


Figure 2.23: Summarizing of Angle estimation process

3 Mapping requirements to Chirp Parameters

Maximum Range	The maximum measurable distance $dmax$ is limited by ADC sampling rate Fs. Larger IF Bandwidth means faster chirps, therefore, better maximum distance.
Range Resolution	How close any two objects can be and still be resolved as two peaks in the IF spectrum. $dres$ is directly proportional to the bandwidth (B) spanned by the chirp.
Maximum Velocity	The maximum measurable velocity in Fast FMCW modulated radars depends on the chirp cycle time Tc, that is, the time difference between the start of two consecutive chirps. This in turn depends on how fast the frequency sweep can be performed and the minimum inter-chirp time allowed. Hence, TC is the total Chirp time ,which includes chirp time+idle time
Velocity Resolution	$vres$ resolution can be improved by increasing frame time (Tf)=> No hardware cost. .
Angular Resolution	Improving angle resolution requires increasing the number of receive antennas. Each receive antenna has its own receive chain (LNA, mixer,LPF, ADC). Cost and area constraints limit most radar on a chip solutions to a small number RX chains (Further improvements possible via multi-chip cascading) .

Given range resolution (dres), max range (dmax), velocity resolution (vres), max velocity (vmax), how do we design a frame?.

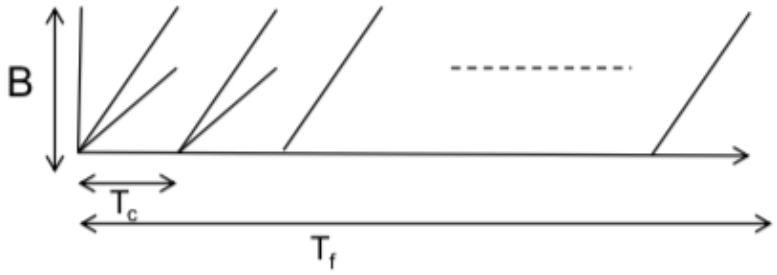


Figure 3.1. Mapping objectives and Requirements into a chirp parameters and accordingly the frame

Terms, Units	Equation	Definition
`Maximum Velocity` km/h	$v_{max} = \frac{\lambda}{4T_c}$	$\Rightarrow T_c = \frac{\lambda}{4v_{max}}$
`Velocity Resolution` m/sec	$v_{res} = \frac{\lambda}{2T_f}$	$\Rightarrow T_f = \frac{\lambda}{2v_{res}}$
`Range Resolution` W/m^2	$d_{res} = \frac{c}{2B}$	$\Rightarrow B = \frac{c}{2d_{res}}$

- Tc determined using vmax.
- B determined using dres. Note that with B and Tc known, the slope S is now (LOCKED) which is determined by $S = \frac{B}{T_c}$
- Frame Time Tf can be determined using vres

$$f_{ifmax} = \frac{S2d_{max}}{c}$$

It's very critical to define our constraints according to the device/target:

- The maximum required IF bandwidth might not be supported by the device since $f_{ifmax} = \frac{S2d_{max}}{c}$ a trade-off in either S or dmax might be needed
- The device must be able to generate the required Slope.
- Device specific requirements for idle time between adjacent chirps need to be honored
- Device must have sufficient memory to store the range-FFT data for all the chirps in the frame. Note that range-FFT data for all the chirps in the frame must be stored before Doppler-FFT computation can start.

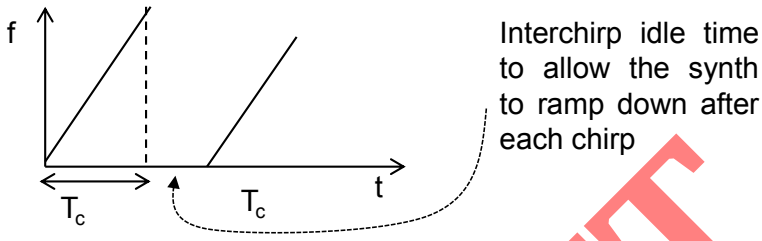


Figure 3.2. Interchirp idle time to allow the synth to ramp down after each chirp

- The product sd_{max} is limited by the available IF bandwidth in the device. Hence as dmax increases => S has to be decreased.
- Assuming Tc is frozen based on vmax, a smaller slope directly translates to poorer resolution

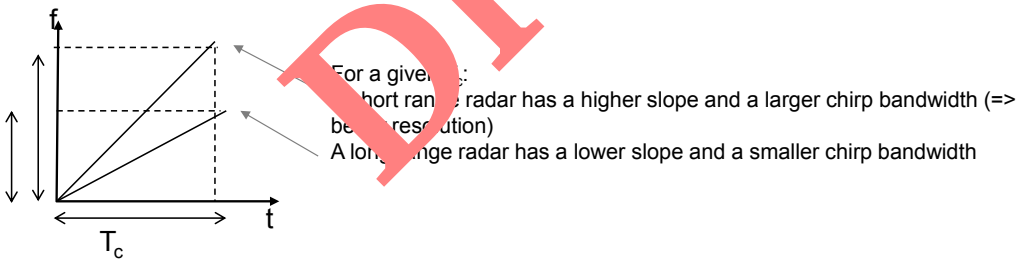


Figure 3.3. Interchirp idle time to allow the synth to ramp down after each chirp

Knobs/Configuration	Value
Idle Time (us)	**5**
ADC Start Time (us)	**3**
Ramp End Time (us)	**44**
Num of ADC Samples	**[illegible]12**
Frequency Slope (MHz/us)	**[illegible]25**
ADC Sampling Frequency (K[illegible])	**[illegible]00**
MIMO	**Enabled**
Number of Chips pe[illegible]rofile	**256**
Effecti[illegible]hirp Ti[illegible] (us)	**42**
Bandwidth (M[illegible])	**3456**
Frame Length (ms)	**10.5**
Memory Requirement	**512**

3.1 Sensing Profiles and Chirp Configuration

FMCW Radar configurations can be broadly categorized into three categories namely short-range radar with a maximum distance of about 30 meters, medium range radar with about 60 meters and long-range radar with about 250 meters.

3.1.1 Chirp configuration for short range radar (SRR)

Short Range Radar can be used in blind spot detection, rear and forward collision mitigation, parking assist, lane change assist etc. While, Medium range Radar can be used for forward collision warning, cross traffic alert, stop & go etc. The industry has not formulated the specific definitions and distinctions between SRR and MRR and neither about their use case.

In FMCW Radar, the chirp configurations control all the basic requirements like maximum range, range resolution, maximum velocity and velocity resolution. The maximum range is given by the equation:

$$MaxRange = \frac{IF_{max}T_c c}{(2BW)}$$

$$RangeResolution = \frac{c}{(2BW)}$$

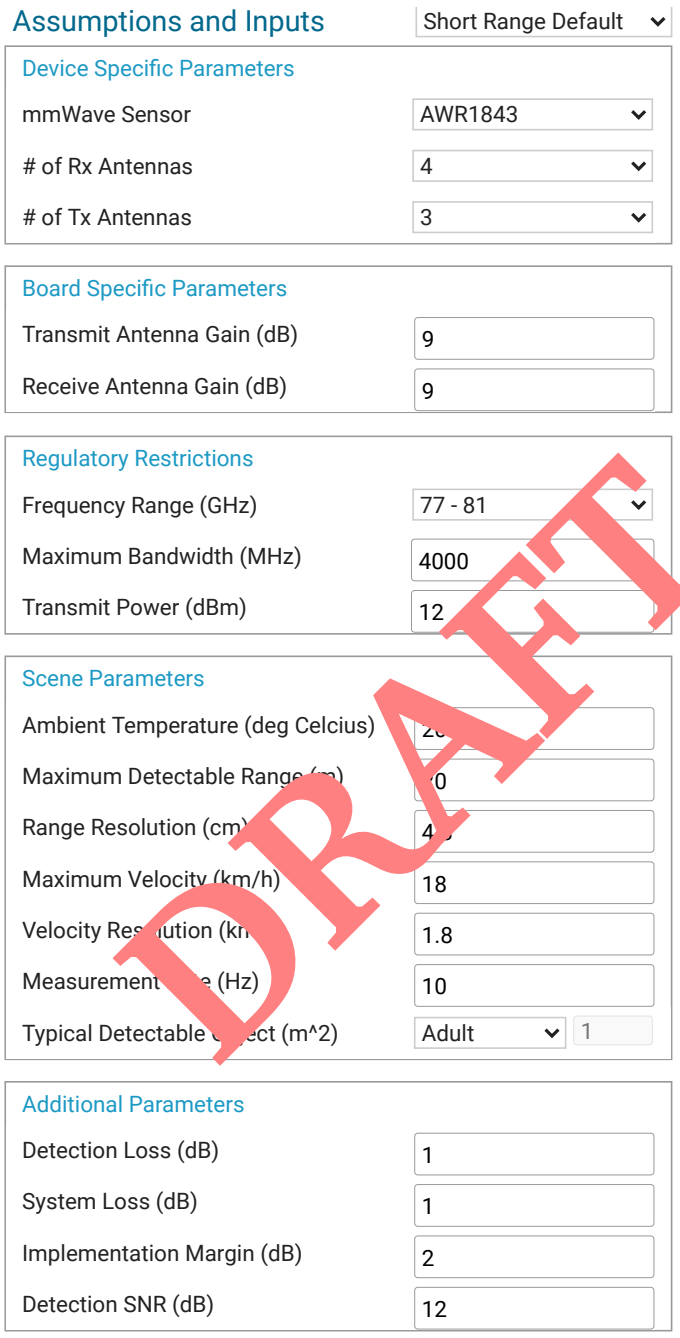
Where IFmax = maximum supported intermediate frequency. This is directly related to the sampling rate of the chirp, S = slope of the chirp, c = speed of light and Tc = chirp time.

The sensing profiles for our system have been done manually using the standard equations with a single objective to achieve higher angle resolution, therefore we used specifically the USRR (Ultra-Short Range Resolution profile). The Sensing profile have been tested and re-evaluated and checked accordingly using the **mmWave Sensing Estimator**.

mmWave Sensing Estimator is an online free accessible TI API for calculating chirp configuration based on scene parameters. It enables rapid prototyping of the chirp through real-time feedback of the chirp configuration and out of bounds checking. It allows to prototype chirp designs rapidly with real-time feedback!

- Load sensible default values to quickly start from a valid configuration. I.e: Short Range, Long Range
- Select the device family and number of active Tx and Rx antennas.
- Input the transmission gain of the antenna on your PCB. TI EVMs are 8dB for both
- Input regulatory restrictions based on where the sensor will deploy. I.e. Frequency Range, BW .. etc.
- Input the desired maximum parameters of your scene.
- Input additional parameters to build in engineering margin into the design.

The Design Rules, Restrictions and Hardware configuration for the AWR1843 device is shown below

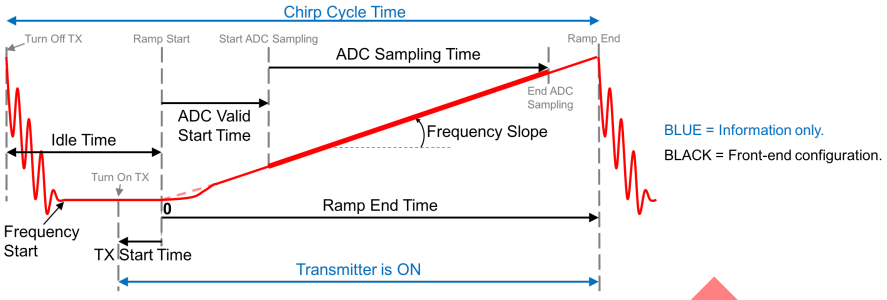


Once we setup the design rules for the SRR configuration, the chirp design parameters are auto-calculated with respect to the hardware limitation such as available RAM and sampling frequency of the sensor, as shown below.

Outputs and Chirp Design

Detectable Object Range

Max Range for Typical Detectable Object (m)	60.16	Min RCS Detectable at Max Range (m^2)	0.01



Chirp Configuration Parameters

Frequency Start (GHz)	77	# of Chirp Loops	21
Frequency Slope (MHz/us)	69.96	Frame Periodicity (ms)	100
Frequency Slope Constant	1449	Idle Time (us)	7
Sampling Rate (ksps)	10364	ADC Valid Start Time (us)	5.40
# of Samples per Chirp	516	Ramp End Time	56.19

Information Only Parameters

Bandwidth (MHz)	3930.3	# of Range FFT Bins	1024
Beat Frequency (MHz)	9.33	# of Doppler FFT Bins	32
Chirp Cycle Time (us)	63	Range Interbin Resolution (cm)	2.17
Chirp Repetition Period (us)	190	Velocity Interbin Resolution (m/s)	0.33
Active Frame Time (ms)	19.99	Radar Cube Size (KB)	1008

[1]

SRR subframe consists of three alternating chirps. Each chirp utilizes one of the three Txs available on the AWR1843 device. Combined processing of this subframe allows the generation of a virtual Rx array of twelve Rx antennas, which consequently has better angular resolution (approximately 14.3°) than the 4 Rx antenna array.

3.1.2 Chirp configuration for medium range radar (MRR)

While moving from SRR to MRR, for the added distance, either the velocity resolution (assuming same range resolution) or the range resolution (assuming same velocity parameters) has to be compensated. If the radar chip supports sufficient higher sampling rate, then MRR can be achieved with the similar configuration as SRR but with more amount of data in radar cube.

Reserved Sub-Section for future design/application

3.1.3 Chirp configuration for long range radar (LRR)

Now the case of LRR is extremely different. In general scenarios, while scanning at 150 meters, the range resolution is not so crucial. A range resolution of 1m is sufficient. This helps in reducing the bandwidth to 1GHz and there by reduces the radar data cube size, which results in lesser data for processing. It is also known that the noise level from both the chip and attenuation from atmosphere is lesser for lesser frequency. Hence it is preferred to use 76-77 GHz band for LRR, even though, literally this doesn't fall within the 77-81 GHz band.

Reserved Sub-Section for future design/application

3.2 Antenna Configuration

One way to double the angle resolution is to double the number of RX antennas (from four to eight).

However, to satisfy our main objective, to achieve higher angular resolution, we can also use a time division multiplexed MIMO, TDM-MIMO configuration (that is, alternate chirps in a frame transmit on TX1, TX2 and TX3 respectively.). The MIMO configuration synthesizes an array of twelve virtual RX antennas (8 antennas in azimuthal plane and 4 antennas in the elevation plane), as shown in 3.4. This configuration improves the angle resolution by a factor of three (compared to a single TX configuration). Accordingly, the angular resolution is increased from $14.3239°$ i to $4.77465°$

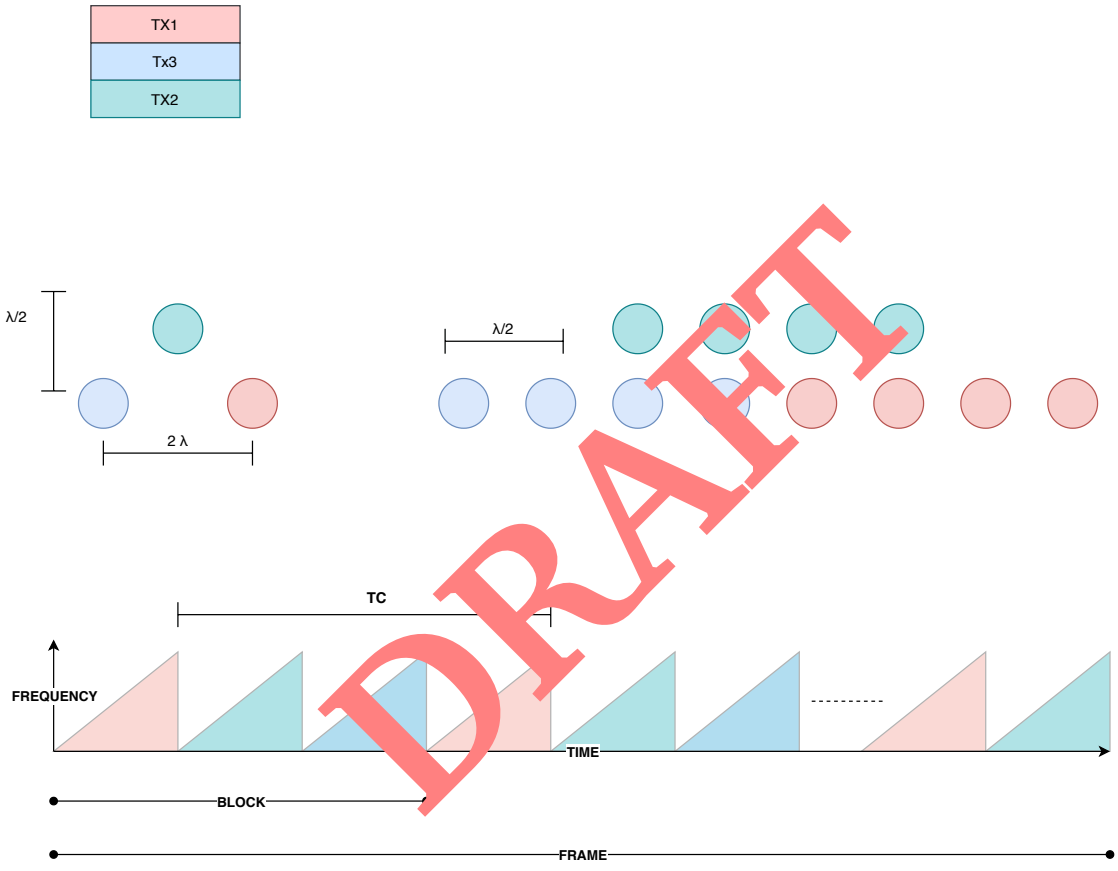


Figure 3.4. TDM-MIMO, Virtual Antenna Geometery

In TDM-MIMO, the orthogonality is in time. Each frame consists of several blocks, with each block consisting of N_{TX} time slots each corresponding to transmission by one of the N_{TX} transmitter antennas. As in our system, for an FMCW radar with $N_{TX} = 3$, alternate time slots are dedicated to TX1, TX2, and TX3. TDM-MIMO is the most simple way to separate signals from the multiple TX antennas and is therefore widely used.

In a typical processing scheme for TDM-MIMO FMCW radar, the 2D-FFT (range-Doppler FFT) is performed for each Tx-Rx pair. Each 2D-FFT corresponds to one virtual antenna. A radar with $N_{Tx} = 3$ and $N_{Rx} = 4$, would compute $N_{Rxv} = N_{Tx} * N_{Rx} = 3x4 = 12$ virtual antennas.

Such range-Doppler matrices as shown in 3.5.

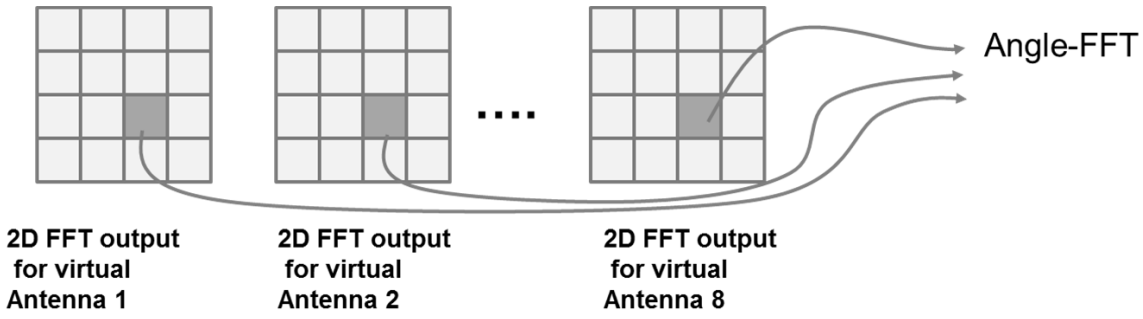


Figure 3.5. Angle Estimation in MIMO Radar over virtual antennas

The 2D- FFT matrices are then noncoherently summed to create a power detection matrix, and then a detection algorithm (CFAR) identifies peaks in this matrix that correspond to valid objects. For each valid object, an angle- FFT is performed on the corresponding peaks across these multiple 2D-FFTs, to identify the angle of arrival of that object. Prior to applying angle-FFT, a Doppler correction step must be performed in order to correct any velocity induced phase change.

3.2.1 TDM-MIMO Radar implementation

AWR1843 has the analog front end closely coupled with digital logic. This coupling allows considerable flexibility in designing the Tx signal. Further, the state machine within the digital logic allows multiple chirp types and various kinds of frame sequences to be programmed up front, relieving the processor from the burden of controlling the front end on a real-time basis. APIs in mmWaveSDK abstract out all the registers in the digital logic and present a simple and intuitive interface to the programmer are also provided. All this content amounts to a programming model that is easy to learn and easy on the processor.

Just hree concepts to keep in mind when programming a Tx signal: profile, chirp, and frame. Each of these concepts is briefly described as follows. Hence, programming the device for a specific MIMO use case amounts to suitably configuring the profile, chirp, and frame.

Profile	It is a template for a chirp and consists of various parameters that are associated with the transmission and reception of the chirp. This includes RF parameters such as the start frequency, slope, duration, and idle time, and ADC parameters such as ADC sampling rate. Up to four different profiles can be defined and stored. **One profile is defined in our system, since we use only one subframe.*
Chirp	Each chirp type is associated with a profile and inherits all the properties of the profile. Additional properties that can be associated with each chirp include the Tx antennas on which the chirp should be transmitted and any binary phase modulation that should be applied. Up to 512 different chirps types can be defined (each associated with one of the four predefined profiles). **Three Chirps are defined in our system, one for each transmitter antenna.*
Frame	Frame is constructed by defining a sequence of chirps using the previously defined chirp types. It also possible to sequence multiple frames, each consisting of a different sequence of chirps.

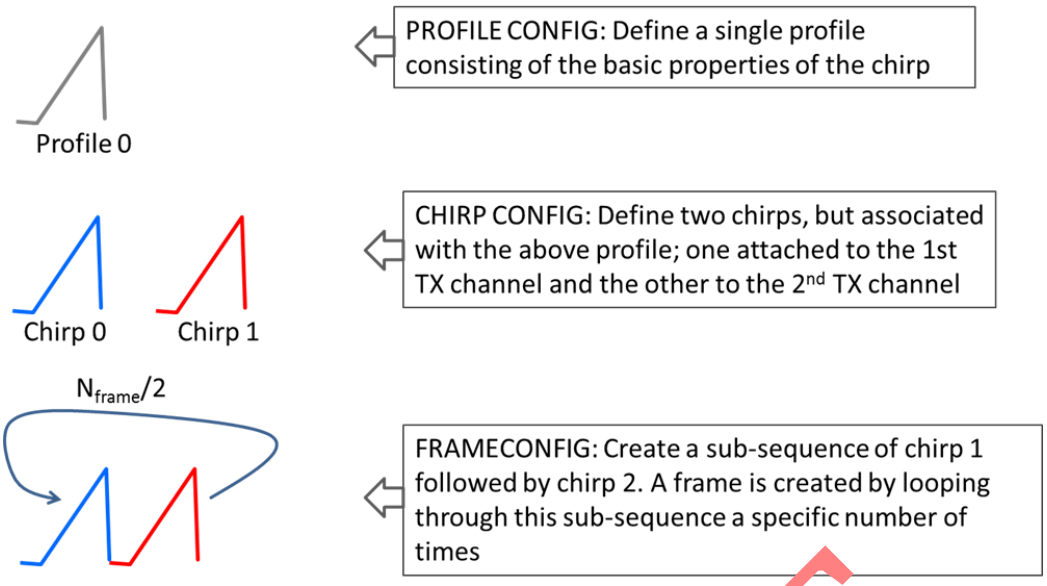


Figure 3.6. Procedures to configure the device for TDM-MIMO Mode Operation

4 Digital Signal Processing Data Path

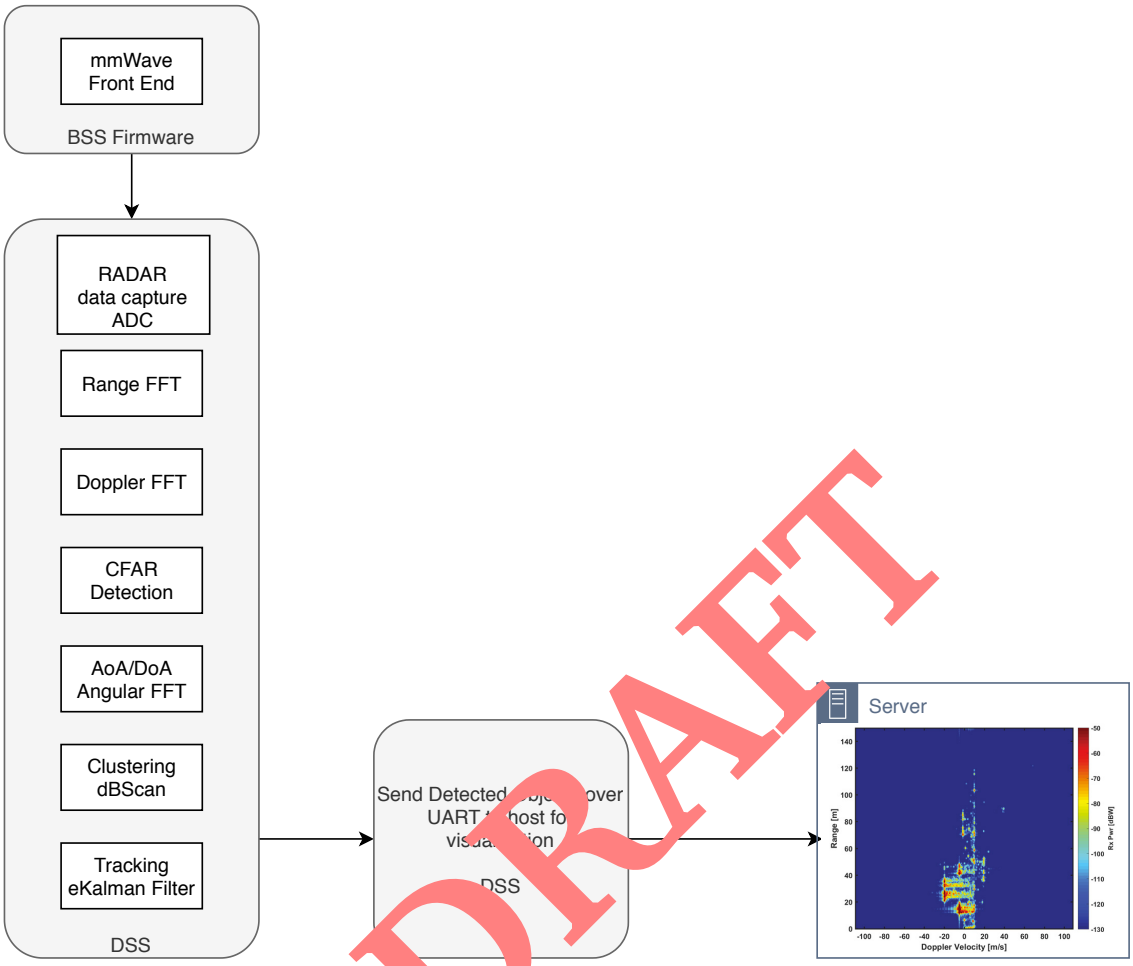


Figure 4.1. FMCW 2D FFT processing in a nutshell 1

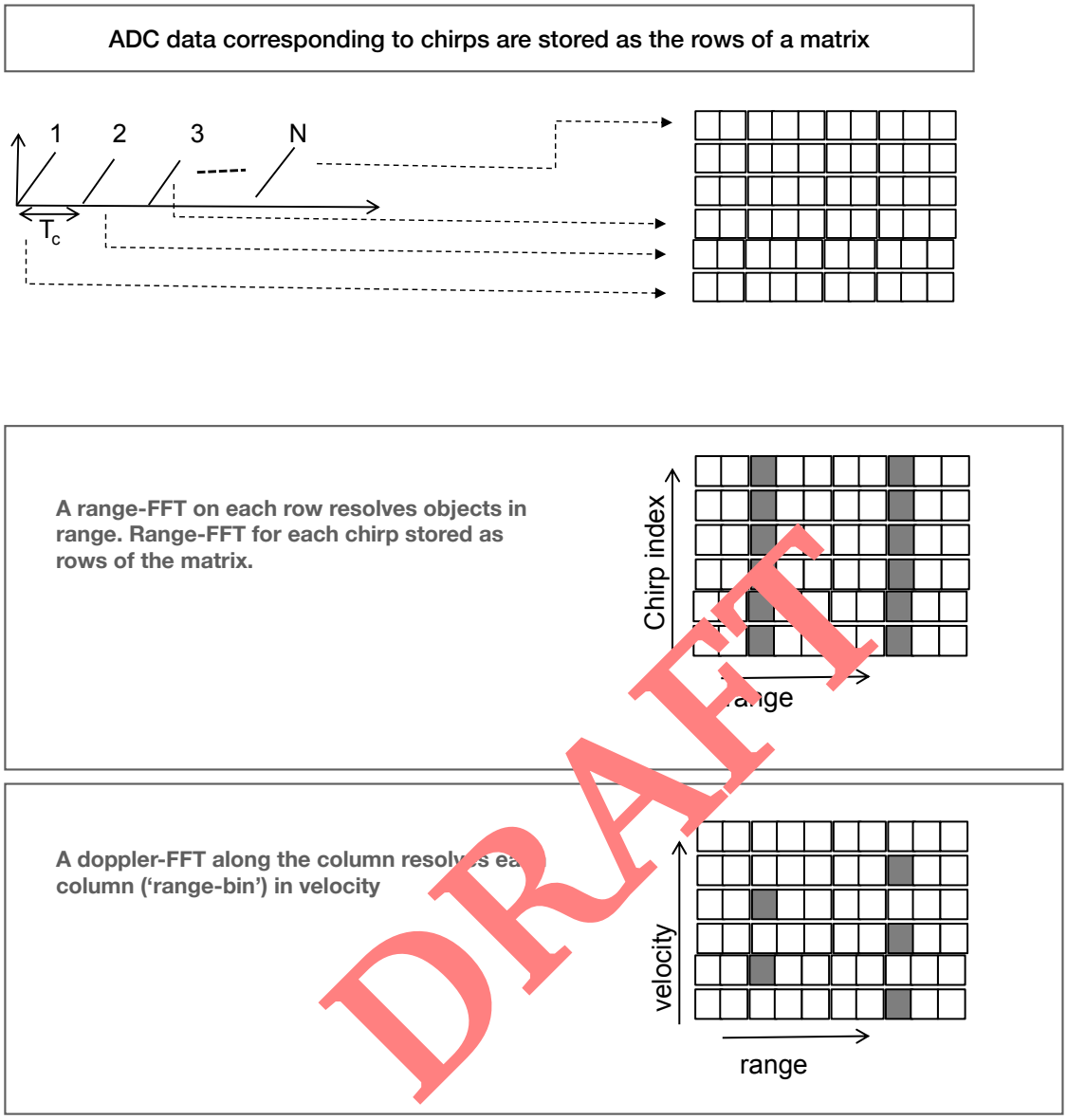


Figure 4.2. FMCW 2D FFT processing in a nutshell 1

The RF front end is configured by the BIST subsystem (BSS). Therefore, the raw data obtained from front end channels is taken by the C67x DSP subsystem (DSS) for processing.

Processing during the chirps consists of:

- 1D (range) FFT processing performed by the C674x that takes input from multiple receive antenna from the ADC buffer for every chirp (corresponding to the chirping pattern on the transmit antenna)
- Transferring transposed output into the L3 RAM by enhanced direct memory access (eDMA)

Processing during the idle or cool down period of the RF circuitry following the chirps until the next chirping period. This processing consists of:

- 2D (velocity) FFT processing performed by C674x that takes input from 1D output in L3 RAM and performs FFT to give a (range, velocity) matrix in the L3 RAM.
- CFAR detection in Doppler direction. CFAR detection in range direction uses the mmWave library.
- Peak grouping (for both Doppler and range) for the MRR subframe and Doppler for the USRR subframe
- Direction of arrival (azimuth and elevation) estimation to map the X-Y-Z location of object.
- Additional pruning based on the SNR and the 2D-FFT magnitude of the object to avoid ground clutter

4.1 Processing Chain

In 4.3 is the processing chain for AWR1843 using the ultra short range chirp and frame design, is implemented on the AWR1843 EVM as shown in

The main processing elements involved in the processing chain consist of the following:

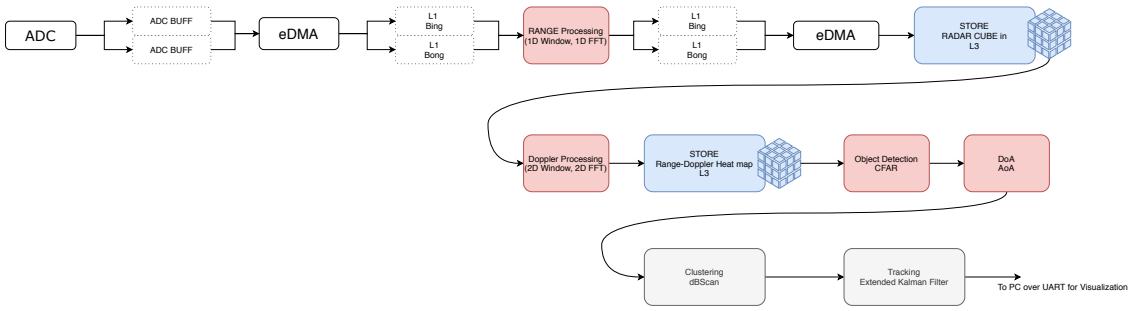


Figure 4.3. Processing chain in AWR1843 RAD AR

Front End – Represents the antennas and the analog RF tr ceiver I lementing the FMCW transmitter and receiver and various hardware-based signal onditioning op ations. This must be properly configured for the chirp and frame settings of the u e.

ADC – The ADC is the main element that interfac to the DS hain. The ADC output samples are buffered in ADC output buffers for access by the d rt of he processing chain.

EDMA controller – This is a user-programmed DM engin employed to move data from one memory location to another without using ano er pro ssor. The EDMA can be programmed to be triggered automatically, and can also be onfi d to eorder some of the data during the movement operations.

C674x DSP – This is the digit signal cess g core that implements the configuration of the front end and executes the low-l l signal ocessing operations on the data. This core has access to several memory resources as no further in the design description.

Range processing – For each ante na, 1D windowing, and 1D fast Fourier transform (FFT). Range processing is interleaved with the active chirp time of the frame

Doppler processing – For each antenna, 2D windowing, and 2D FFT. Then non-coherent combining of received power across antennas in floating-point precision

CFAR – Range-Doppler detection algorithm is based on Constant false-alarm rate algorithm, cell averaging smallest of (CASO-CFAR) detection in range domain, plus CFAR-cell averaging (CACFAR) in Doppler domain detection, run on the range-Doppler power mapping to find detection points in range and Doppler space

Angle estimation – For each detected point in range and Doppler space, reconstruct the 2D FFT output with Doppler compensation, then a beamforming algorithm is applied to calculate the angle spectrum on the azimuth direction with multiple peaks detected. After that the elevation angle is estimated for each detected peak angle in azimuth domain.

Clustering – Collect all detected points and perform DBSCAN-based clustering algorithm for ev-

ery fixed number of frames. The reported output includes the number of clusters and properties for each cluster, like center location and size. After the DSP finishes frame processing, the results consisting of range, doppler, 3D location, and clustering are formatted and written in shared memory (L3RAM) for R4F to send all the results to the host through UART for visualization.

4.1.1 CFAR: Constant False Alarm Rate detection

One important task a radar system performs is target detection and localization in terms of range, velocity and angle. The detection itself is fairly straightforward by performing the FFT in three dimensions as discussed before. However, the extraction of such accurate information is considered as a complex task, due to the time-varying noisy environment. Hence, the measured noise power associated with clutter is unknown and could be time varying. Therefore, the need for adaptive signal processing methods i.e CFAR is mandatory to improve the performance and probability of targets detection.

In CFAR detection the detector is required to maintain a constant false alarm probability while detecting targets in non-stationary clutter. CFAR compares the signal to a an adaptive threshold. The threshold is therefore a function of both the probability of detection P_d and the probability of false alarm P_{fa}. Accordingly, to provide accurate detection results in a realistic interference scenario

The CFAR detector in a radar system detects targets with varying interference by dynamically adjusting to the detection thresholds in order to maintain a constant design-specific false-alarm rate with a corresponding probability of false alarm (failure) P_{fa}. When the detection is needed for a given cell,often termed as the cell under test (CUT), the noise power is estimated from neighboring cells. Then the detection threshold, T, is determined by estimating the interference around the target. The CFAR estimates the interference power for a cell under test (CUT) by averaging the sample values of adjacent cells, under the assumption that interference statistics are homogeneous in a localized area. T is given by is given by

$$T = \alpha P_n$$

Where, $P_n$is the noise power estimate and α the threshold scaling factor. Now, it is clear that the threshold adapts to the data. It can be shown with the appropriate threshold factor, α, the resulting probability of false alarm can be kept at a constant, hence the name CFAR. Now we will discuss the implemented method in our system called Cell Averaging Constant False Alarm Rate detection (CA-CFAR).

In **CA-CFAR: Cell Averaging CFAR** noise samples are extracted from both leading and lagging cells (called training cells) around the CUT. The noise estimate can be computed as belwo:

$$P_n = \frac{1}{N} \sum_{m=1}^{N} X_m$$

where N is the number of training cells and X_m is the sample in each training cell. If X_m happens to be the output of a square law detector, then P_n represents the estimated noise power. In general, the number of leading and lagging training cells are the same. Guard cells are placed adjacent to the CUT, both and leading and lagging it. The purpose of these guard cells is to avoid signal components from leaking into the training cell, which could adversely affect the noise estimate.

The following figure shows the relation among these cells for the 1-D case.

4.1.2 Clustering

Clustering is performed using the dBscan algorithm. dBScan is applied on the detected objects of both the SRR subframe. Accordingly, the output of the clustering algorithm is the mean location of a cluster and its dimensions (assuming the cluster is a rectangle).

For the SRR subframe, the clustering output is sent as is to the graphical user interface (GUI). USRR clusters allow the grouping of dense point clouds to rectangles. In cross- traffic scenarios, these clusters can be used to identify vehicles crossing the field of vision (FoV) of the radar.

4.1.3 Kalman Filter

Clustering output forms the input for the tracking algorithm. The strongest object in the cluster is provided as a representative object for the tracking algorithm.

The tracker is a fairly standard Extended Kalman Filter (EKF) with four states $[x, y, vx, vv]$ and three inputs $[r, v, sin(\Theta)]$ or range, relative velocity, and sine of the azimuth).

Signal to noise ratio is taken as input and associated variance is computed using the Cramer-Rao Lower Bound formula for frequency variance .

5 Device Drivers Configuration

5.1 ADC Buffer

The ADC buffer is on-chip memory arranged as a ping-pong buffer, with ECC (Error-correcting Code) memory support for each ping and pong memory. The raw ADC output data from RADAR-SS(BSS) is stored on this memory, to be consumed by the DSP processor. Hence, The analog signals received on each of the configured receive (Rx) channels in the device passes through a pre-conditioning over the Analog and Digital Front End (DFE) and the resulting data at the configured sampling rate is stored in the ADC buffer. Data corresponding to all the configured Rx channels is stored within this buffer. Once again, the ADC buffer is implemented as a double buffering (ping-pong) mechanism that allows for one buffer to be written(filled) while the other one is being read out(emptied). The size of the ADC buffer is 32 KiB for each ping and pong buffers.

The ADC buffer can be written from DFE in any of the three modes by configuring the control registers or by using the API as in our case.

Single-chirp mode	Multi-chirp mode	Continuous mode

In single-chirp mode, the FMCW chirp data from the DFE is written to the ADC buffer on a per chirp basis, and a chirp available interrupt is generated on the completion of the write data operation at the end of the chirp, as shown in Figure 13-3. ADC buffer control logic generates the Ping_Pong_Sel signal, as shown in Figure 13-3, which controls whether the data is written into either ping or pong buffer. Data write can start from either the ping or pong buffer.

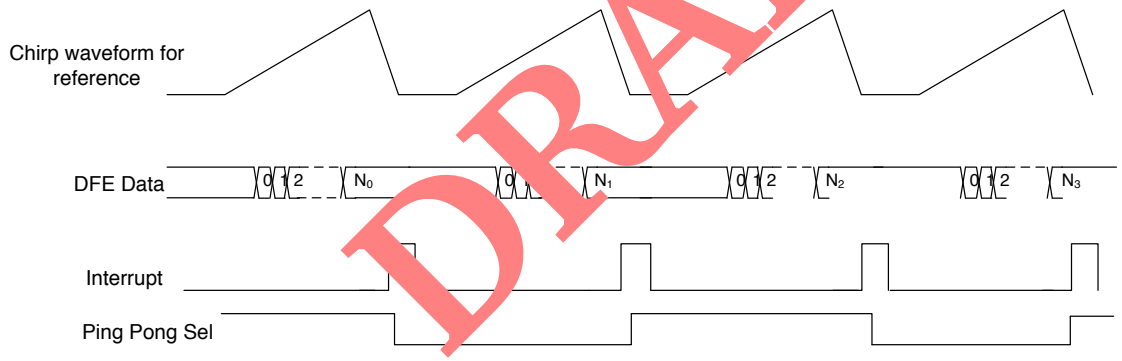


Figure 5.1. ADC Buffer in Single-Chirp Mode

In multi-chirp mode, ADC samples for N chirps are stored in a ping/pong buffer before the Ping Pong Select toggles and the Chirp Available Interrupt is generated. The number of chirps stored in the ping buffer is configured in the register field ADCBUFNUMCHRPPING, and the number of chirps to be stored in the pong buffer is configured in the register field ADCBUFNUMCHRPPONG or using mmWave APIs directly.

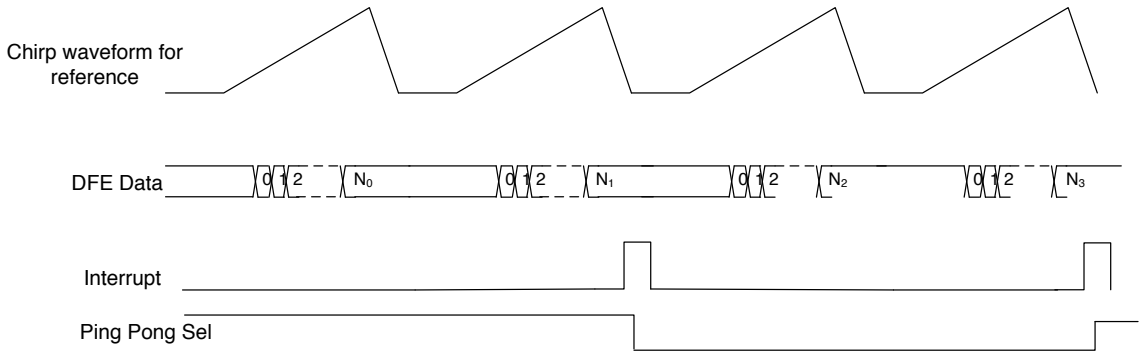


Figure 5.2. ADC Buffer in Multi-Chirp Mode

In continuous mode, where the FMCW transceiver is configured to output a single frequency tone in the range of 76-81 GHz, N ADC samples are stored in a ping/pong buffer before the Ping Pong Select toggles and the Chirp Available Interrupt is generated. In real mode this value N refers to the number of real samples per channel, and in complex mode, this refers to the number of complex samples per channel. This counter increments once for every new sample (as long as 1 or more Rx channels are enabled). Continuous mode is expected to be only used for BIST and ADC buffer test pattern mode.

5.1.1 ADC Data Format

The ADCBuf driver in TI-RTOS samples an analogue waveform at a specified frequency. The resulting samples are transferred to a buffer provided by the application. The driver can either take n samples once, or continuously sample by double-buffering and providing a callback to process each finished buffer.

ADC buffer Data format is written into Non-interleaved data storageto the ADC buffer. Accordingly, each channel data is stored in different memory locations, as shown below.

Rx0(3)	Rx0(2)	Rx0(1)	Rx0(0)
Rx0(7)	Rx0(6)	Rx0(5)	Rx0(4)

Rx1(3)	Rx1(2)	Rx1(1)	Rx1(0)
Rx1(7)	Rx1(6)	Rx1(5)	Rx1(4)

Rx2(3)	Rx2(2)	Rx2( )	Rx2(0)
Rx2(7)	Rx2(6)	Rx2(5)	Rx2(4)

In the non-interleaved mode of storage, the ADC data corresponding to each Rx channel are grouped and stored together allowing easy processing of the related data corresponding to each channel. The storage offset for each of the channels is configurable. Also depending on the number of channels configured, the offset to store the data can be moved to allow for larger amount of data to be stored within the same buffer reduced number of Rx channels.

Rx3(3)	Rx3(2)	Rx3(1)	Rx3(0)
Rx3(7)	Rx3( )	Rx3(5)	Rx3(4)

5.2 MAILBOX

Mailbox mechanism is an important component which is essential for inter-process communication *(IPC)*. Mailbox uses message queue for messaging passing to asynchronously exchange the messages between any two processors. Each processor has a mailbox memory space, and registers designated to be used by other processor that wishes to communicate with it. In addition, processor has two sets of mailbox memory space and registers, and each set is designated per other processor to communicate with it as seen in 5.3.

Mailbox relies on virtual communication channels, where they are a fundamental abstraction provided by the abstract machine. Hence, Channel: is a *link* between two tasks used to exchange data. On the other hand, in Hardware machine, the channel is the communication network (ethernet, PCIe, ..etc) which Can have different semantics, and can be used using different syntax, depending on the abstract machine and on its implementation.

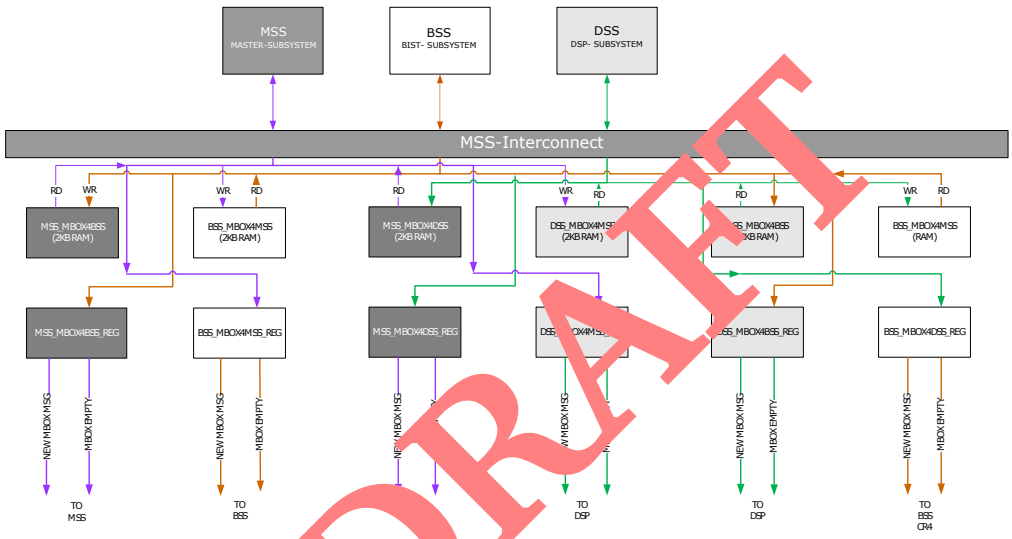


Fig 5.3. Mailbox Block Diagram

Mailbox channel semantics; The behaviour of channel used to communicate between tasks depends on many factor:

- **T**he direction of the communication [Mono-directional/unidirectional or bidirectional]. In Mono-directional, the most commonly used cases is *pipeline* e.g producer consumer mechanism. However, In our case, we use bidirectional communication channel for the communication between MSS and DSS. To be more specific, the Bidirectional channel is implemented using 2 unidirectional virtual channels.
- **T**he number of tasks sending and receiving. In our scenario, we have only two tasks, one on the MSS side and another task on the DSS side. Morever, It could be n to m tasks or 1 to m, or n to 1. Here, we have 1 to 1 tasks communicatating through two virtual channels (MSS to DSS) and (DSS to MSS).
- **T**he kind of synchronization [Asynchronous, Synchronous, Extended]. Message queues provide

an asynchronous communications protocol, meaning that the sender and receiver of the message do not need to interact with the message queue at the same time. Messages placed onto the queue are stored until the receiver retrieves them. Message queues have implicit or explicit limits on the size of data that may be transmitted in a single message and the number of messages that may remain outstanding on the queue. The implementation of mailbox message queues function internally: within TI Real-Time operating system

Maibox Message Scheme: The processor which wishes to send a message to another processor writes the message to the mailbox memory space, then interrupts the receiver processor. The receiver processor acknowledges the interrupt, then reads the message from the mailbox memory space. The receiver informs the sender that the message is read by an interrupt, which is acknowledged back by the sender. The sender must not initiate another message to the same receiver until the previously initiated mailbox interaction with the same receiver is complete. Hence, the the register layer definitions for the Mailbox Module are included in the following path:

mailbox/include/reg_mailbox.h

High level encapsulation (Abstraction) have been defined in the following path:

ti/drivers/mailbox/mailbox.h

Mailbox Initialization and Implementation withing the mmWave Radar application is defined here.

MAILBOX

```
#include <ti/drivers/mailbox/mailbox.h>

/* mboxHandle is defined in a globally accessed structure, here it's defined just for illustration*/
Mbox_Handle      mboxHandle;
Mailbox_Config   mboxCfg;
int32_t          mboxErrCode;

Mailbox_Config_init(&mboxCfg)


mboxCfg.readMode          = MAILBOX_MODE_CALLBACK
mboxCfg.writeMode         = MAILBOX_MODE_CALLBACK
mboxCfg.readTimeout       = MAILBOX_WAIT_FOREVER
mboxCfg.writeTimeout      = MAILBOX_WAIT_FOREVER
mboxCfg.readCallback      = &mmWave_mboxCallbackFxn_MSS_ch1;
mboxCfg.opMode            = MAILBOX_OPERATION_MODE_PARTIAL_READ_ALLOWED
mboxCfg.dataTransferMode  = MAILBOX_DATA_TRANSFER_MEMCPY
mboxCfg.chType            = MAILBOX_CHTYPE_MULT
mboxCfg.chId              = MAILBOX_CH_ID_0

gMSSMCB.peerMailbox       = MAILBOX_open(MALIBOX_TYPE_DSS,&mboxCfg,&mboxErrcode)

if ( (!gMSSMCB.mboxHandle) || (mboxErrCode ==0) ) {

    System_printf("[Error][MAILBOX] Unable to open channel 1 between MSS & DSS")


}

TASK_Params_init(&taskParams)
taskParams.stackSize          = 16*1024
Task_Create(&mmWave_mboxReadTask, &taskParams, NULL)
```

Where we relie on the communication between MSS and DSS, multiple instances of the driver

can be opened (each instance controls one virtual channel). Note that the mailbox driver is instantiated both in the MSS and DSS. The code sample attached is deployed on the MSS side to open a virtual channel to communicate with the DSS. Accordingly, same code is deployed in the DSS side to open a second virtual channel to communicate with the MSS

5.3 UART

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

5.4 eDMA

5.5 CANFD Driver

The application report [4] describes the required procedures to integrate the CAN-FD interface on the mmWave devices.

Many CAN buses have reached 50%-95%+ bus-network load level.

- Standard CAN 111 bits/message for 64 bits of data [Excluding Stuff bits].
- Extended CAN 131 bits/message for 64 bits of data [Excluding Stuff bits].

CANFD Based Functions

- MCANAppErrStatusCallback()
- MCANAppCallback()
- MCANAppInitParams()
- Can_Initialize(void)

Reserved Sub-Section for future design/application.

6 Output Data Structure Format

7 System Specifications: Hardware Specification and limitations, Design Tradeoff, Design Corners bounds

8 System Specifications: Programming Model and Software Architecture

mmWave Suite is the functional software part of the mmWave SDK and would encapsulate these smaller components:

- Drivers
- OSAL
- mmWaveLink
- mmWaveLib
- mmWave API
- Data processing layer (manager, processing units)
- RADARSS Firmware
- Board Setup and Flash Utilities

8.1 Device Drivers

Drivers encapsulate the functionality of the various hardware IPs in the system and provide a well defined API to the higher layers. The drivers are designed to be OS-agnostic via the use of OSAL layer.

On the MSS side running on top of ARM-Cortex R4F 200MHz, we use the following drivers. UART, MCAN (CAN or CAN-FD) and Mailbox. On the DSS side running on top of C674x DSP 600MHz, we use ADC and EDMA.

Our mmWave RADAR application would perform these operations:

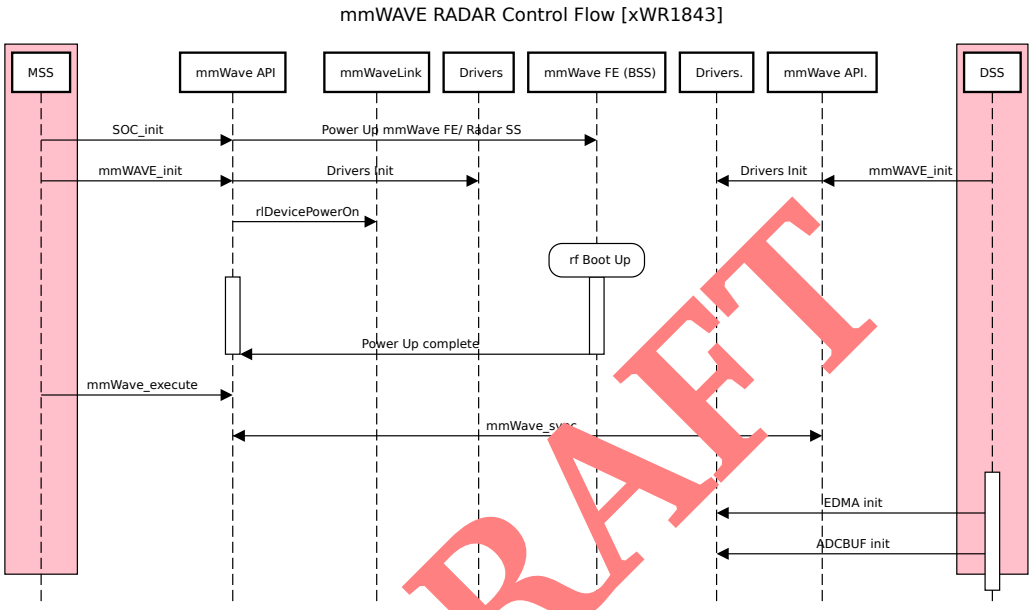
- Control and monitoring of RF front-end through mmaveLink
- Transport of external communications through standard peripheral
- Apply radar data processing using DSP

8.2 Application Sequence Diagram

The MSS application uses the *SOC_init* to initialize the device and powers up the mmWave Front End mmWave. It then uses the *MMWave_init* API to do the basic initialization of the mailbox and the other key drivers. Concurrently, DSP application, on the other hand, do the same initialization procedures. After that, it waits for the (FE) front end boot-up completion. Then it uses the

MMWave_execute API, which sets up the Inter process communication (IPC) to receive the data from the MSS. Both the MSS and the DSS does synchronization to check the health of each other. DSS also initializes the EDMA and the ADC buffer for the data processing.

Once the initialization is complete and the MSS and the DSS are both synchronized, MSS application use the *MMWave_config* API to parse the configuration from the application to the mmWave Front End. mmWave API uses the mmWaveLink API, which constructs the mailbox message and sends it to the mmWave Front End. mmWave Front End, once it receives a message, checks the integrity of the message and sends an acknowledgement back to the mmWaveLink. In this way, all the messages are sent to the front end.



8.2.1 MSS Initialization Task

In the Initialization Task, we initia[illegible] all necessery device drivers UART, MCAN, GPIO, and Mailbox.

References

[1] TI App mmwave sensing estimator. online chirp designer. https://dev.ti.com/gallery/view/mmwave/mmWaveSensingEstimator/ver/1.4.0/. Accessed: 2019-12-01.

[2] C. A. Balanis. Antenna theory: Analysis and design, 4th edition, Feb 2016.

[3] E. Hyun, Y.-S. Jin, and J.-H. Lee. A pedestrian detection scheme using a coherent phase difference method based on 2d range-doppler fmcw radar. *Sensors*, 16(1):124, 2016.

[4] R. Kamath. Adding can-fd tx and rx to an existing mmwave project. *SPRACF7 -Copyright © Texas Instruments Incorporated*, 2018.