mmWave Radar Notes

# IWR1843BOOST Antennas

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

# IWR1843BOOST SOP modes

Imagen que contiene Tabla

Descripción generada automáticamente

* Flash programming: Uniflash
* Functional mode: mmWave Demo Visualizer
* Debug or development mode: DCA1000

# Chirps

En la banda base compleja, el modelo en el dominio del tiempo de la señal del pulso generada por el mixer (frecuencia intermedia), que consiste en componente en fase (I, in-phase) y en cuadratura (Q, quadrature), se define como

𝑠(𝑡) = 𝑠𝐼(𝑡) + 𝑠𝑄(𝑡) = exp(𝑗2𝜋(𝑓0𝜏 + *k*𝑡𝜏 − 0.5𝐾𝜏2)) , 0 ≤ 𝑡 ≤ 𝑇

donde 𝑓0 es la frecuencia inicial del chirp, *k* = 𝐵/𝑇 es la pendiente calculada a partir del ancho de banda válido 𝐵, y la duración del chirp 𝑇. 𝜏 es el desfase de ida y Vuelta de la señal reflejada asumiendo un único dispersor ideal en la escena, y el término 𝐾𝜏 es la frecuencia del pulso que lleva la información sobre el rango de este dispersor.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Podemos observar que el dispositivo no utiliza siempre el ancho de banda total del radar, mientras que el ancho de banda efectivo depende de otros parámetros. El dispositivo transmite una sucesión de chirps, que se agrupan dentro de un frame y cada frame tiene una duración denominada . El ADC Sampling Time es el periodo dentro del cual el dispositivo muestrea la señal del pulso. El ADC Valid Start Time es usado para evitar la primera parte de la rampa, que no es lineal y puede causar distorsión dentro de la señal. El Idle Time se utiliza para esperar el reinicio del generador de la rampa. El tiempo de adquisición depende de estos parámetros, y viene dado por el valor de multiplicado por el número de frames. El valor de debe ser al menos igual al tiempo total de un chirp multiplicado por el número de chirps por frame.

The FMCW waveform with L consecutive linear FM chirps (or sweeps) can be expressed as s(t) = ∑L−1 l=0 x(t − lTc), where the individual chirps are given by x(t) = ejφ(t)rectTv (t), φ(t) = 2π(fct+0.5αt2). Here, α = BWv/Tv is the chirp slope, BWv denotes the sweep bandwidth, Tv represents the chirp duration, fc is the carrier frequency, rectTv (t) is rectangular pulse of duration Tv of the victim radar and Tc is the pulse repetition time (PRT).

# Bandwidth

Para calcular el ancho de banda del chirp, se utiliza el tiempo de sampleo ADC, que puede obtenerse sabiendo la frecuencia de muestreo y el número de muestras ADC como:

Teniendo la pendiente del chirp, , es fácil calcular el ancho de banda como:

# Máximo rango medible

Para una configuración de parámetros dada, es posible calcular el máximo rango medible de varias formas. Se puede calcular el mínimo rango detectable tras hacer la FFT como

Y si la FFT tiene , entonces el máximo rango medible es .

Otra forma de calcularlo es sustituir el valor de calculado en el apartado del ancho de banda y llegar a la ecuación

Que también se obtiene a partir de la relación entre rango y frecuencia intermedia, donde se ve que el factor de conversión entre y es , y como la FFT da las frecuencias en múltiplos de la frecuencia de muestreo, al multiplicar el vector de frecuencias por y utilizar el factor de conversión se llega a la misma expresión para el rango máximo.

# Máximo Doppler medible

Para una configuración de parámetros dada, es posible calcular el mínimo Doppler medible tras hacer la FFT como

Siendo el número de chirps de cada frame y el tiempo de separación entre dos chirps consecutivos.

El tiempo de separación entre chirps debe ser el tiempo inactivo más el tiempo hasta el final de la rampa, es decir . No obstante, este último es el tiempo entre chirps emitidos por una misma antena. Si se utilizan varias antenas transmisoras, el chirp de la antena TX1 no volverá a emitirse hasta que se emita el de la antena TX2, con lo cual el tiempo entre chirps de la antena TX1 es dos veces el tiempo entre chirps cuando sólo se utilizaba una antena. Es decir, el tiempo entre chirps es el número de antenas transmisoras por , .

Para calcular el Doppler se promedian todas las antenas y luego se hace la FFT a lo largo de los chirps, por lo tanto, el tiempo entre un chirp y el siguiente que ve la FFT se incrementa tantas veces como antenas promediadas. La recepción en las antenas RX debe ser síncrona, sino no tendría sentido el funcionamiento del dispositivo, por tanto, aunque se promedien todas las antenas virtuales, sólo importa el número de antenas de transmisión, que indica el número de chirps distintos emitidos y que provoca el retraso en una nueva emisión del mismo chirp.

El valor máximo de Doppler se obtiene multiplicando el mínimo Doppler por el número de bins y dividiendo entre 2, ya que el rango tiene valores positivos y negativos, no como en el rango que únicamente valen los valores positivos (?).

Comparando con la tesis de Jose, el valor máximo allí descrito es

Donde es el tiempo de separación entre un mismo chirp.

# Rango Azimuth medible

El rango de azimuth tras una FFT de 64 puntos se calcula con la expresión vectorial siguiente:

Pero, para ángulos mayores a 75° no hay suficiente resolución, así que eliminamos el primer ángulo (-90°) para azimuth (y también para elevación). Teniendo un total de 63 puntos en el heatmap.

# Rango Elevación medible

El rango de elevación tras una FFT de 32 puntos se calcula con la expresión vectorial siguiente:

# Slow-time – Fast-time

El fast-time es el tiempo de duración de un chirp. El slow-time es el tiempo de duración de varios chirps (un frame).

También puede ser el fast-time el tiempo de duración de un frame y el slow-time el de duración de varios frames.

# Time Division Multiplexing (TDM)

Para los radares xWR16xx/IWR6843 la forma de almacenar los datos es no entrelazada, es decir, primero está toda la información de la antena receptora 1, luego de la 2, etc. Para xWR14xx se recomienda el modo entrelazado (y es la única que lo soporta), en el que se almacena la información de todas las antenas para la primera muestra sampleada, luego la de todas las antenas para la segunda, etc.

Imagen de la pantalla de un celular

Descripción generada automáticamente con confianza media

*Non-interleaved Complex Data Format*

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

*Interleaved Complex Data Format*

# Resultado mmWave Demo Visualizer IWR1843BOOST

*Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente*

# FFT

Al hacer la FFT de puntos sobre una señal de muestras se obtiene un vector complejo con puntos cuya parte real e imaginaria se calculan como:

Las componentes de modo que el resto de componentes son independientes (recordemos que la señal tiene muestras). La primera es cero porque el seno de 0 es 0, y la segunda porque al sustituir se obtiene .

El valor es la media de los puntos de la señal en el dominio del tiempo, por lo que se conoce a este valor como DC offset.

Teniendo en cuenta la expresión del seno o coseno para una onda y comparando con las ondas por las que se multiplican las componentes de la señal para encontrar , teniendo en cuenta que se ve que

Así, la máxima frecuencia que podemos encontrar en la señal muestreada es 0.5 veces la frecuencia de muestreo.

La DFT ve la señal como periódica, por tanto, el espectro también lo es, no hay más que ver que múltiplos de una frecuencia dan los mismo valores de seno y coseno. El periodo de la señal va de 0 a 1 veces la frecuencia de muestreo, o de -0.5 a 0.5. Y tiene el siguiente aspecto:

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

# Lectura correcta de datos

Los números se mandan en formato de 16 bits y en complemento a 2, es decir, cuando un número empiece por b’1XXXXXXXXXXXXXXX será negativo, y su valor positivo se calcula restándole . Esto es lo que hacen en MATLAB. En la librería de Python lo leen igual que en MATLAB, como entero de 16 bits sin signo, uint16, pero no tienen en cuenta el complemento a 2. Se puede leer como int16, con signo, y automáticamente se hace la conversión de complemento a 2, se puede probar viendo que np.uint16(-1) es 1111111111111111=65535, y asumiendo que es complemento a dos, podemos obtener el valor con signo haciendo 65535- que es -1.

Los de la librería de *ti77radar* leen como int16 pero *openradar* lo hace como uint16.

No sé si bastaría con leer del radar en int16 o convertir antes de entrar en organize a int16 con np.frombuffer(adc\_data, dtype=np.int16). Por cierto, cuando se convierte a int16 o uint16 usando Numpy, se divide el número de samples a la mitad, porque Python lo lee como enteros de 8 bits. Por ahora he convertido el buffer leído a int16 antes de devolverlo como salida de la función DCA1000.read().

El script data\_format\_test.py y readFileMATLAB.m muestran el mismo frame para comprobar que el resultado obtenido es el mismo.

Otra cosa que pasa es que a veces ellos leen dos I y dos Q (readDCA1000\_2.m) y otras veces leen I y luego Q (read\_ADC\_bin\_TDA2\_separateFiles.m), pero si se hace la FFT de la señal para ver el rango, sólo cuando se leen dos I y luego dos Q se obtiene algo con sentido.

He probado a ver si la organización de los datos en canales RX, chirps y muestras por chirp es la misma en el script de MATLAB y lo que hacen en la librería openradar. En el archivo readFileMATLAB.m está la función que lee los datos y los transforma a una matriz cuyas filas son todas las muestras de un canal RX (numSamplePerChirp\*numChirps). Aquí leen dos I y luego dos Q, y si comparo todas las muestras de la primera antena receptora con esto mismo del array obtenido con la función organize se obtiene el mismo. Con lo cual, en principio, la organización de las antenas es la correcta. Además, si hago la gráfica del primer chirp de la primera antena receptora con Python se obtiene lo mismo que con las primeras numSamplePerChirp de la primera antena RX de la matriz leída en el programa de MATLAB:

Gráfico, Histograma

Descripción generada automáticamente