DESCRIZIONE GENERATORE DATABASE CON INTERFERENTE

Nel seguente documento verrà descritto il codice della sorgente “db4ra\_jammer\_database\_gen.m”.

PRIMA SEZIONE: INIZIALIZZAZIONE VARIABILI STATICHE

Nella prima sezione si inizializza l’oggetto “db4ra\_collector” che emula un array lineare costituito da “n\_antennas” antenne omnidirezionali, spaziate “lambda” metri tra di loro.  
 L’oggetto accetta un segnale in banda base in ingresso e un angolo, e restituisce “n\_antennas” uscite complesse, che rappresentano gli output delle quattro antenne opportunamente sfasati in modo da emulare il comportamento del sistema reale.  
Ingresso e uscita sono in banda base, ma il sistema si occupa di emulare internamente la presenza di una portante sul segnale in ingresso e di restituire in uscita il segnale “demodulato”.  
Frequenza di campionamento del segnale in banda base e frequenza della portante a RF sono descritte rispettivamente dalle variabili “CollectorSampleRate “ e “CollectorCarrierFrequency”.

SECONDA SEZIONE: INIZIALIZZAZIONE VARIABILI DINAMICHE

Lo scenario che si intende emulare è il seguente: l’array di antenne riceve un segnale utile e un disturbo.

Segnale Utile

Il segnale utile è una chirp lineare di durata “sample\_width” (target 100 us) e incremento variabile.  
La frequenza di partenza è 0 Hz, le varie frequenze di arrivo sono nel vettore “f\_stop\_array”.   
Il numero di chirp che si intende utilizzare è descritto dalla variabile “n\_scenarios”.

Il sistema deve inoltre funzionare con un fattore di oversampling della chirp non superiore al 30%, per cui al variare della chirp verrà applicato al segnale utile un sottocampionamento variabile.   
I fattori di sottocampionamento associati a ciascuna chirp sono descritti nel vettore “downsample\_array”. Il fattore di oversampling implementato al momento è pari all’11% (10/9).  
  
Il segnale utile può arrivare da una porzione dell’angolo di azimuth compresa tra “azimuth\_start” e “azimuth\_stop”. Il range è spazzato con uno step descritto dalla variabile “azimuth\_step”.

Le variabili sono divise in due gruppi “woj” e “wj”: without jammer e with jammer. Una porzione del dataset sarà generata senza interferente: in questo modo si ha più elasticità nel poter bilanciare il dataset.

Interferente

Il segnale interferente è di tipo sinusoidale. La frequenza della sinusoide è all’interno della banda del segnale utile, dalla frequenza 0 Hz alla maggiore delle frequenze descritte nel vettore “f\_stop\_array”.   
In modo da rendere il dataset completo e alla luce del sottocampionamento variabile, le frequenze delle sinusoidi di jamming sono generate come frazioni della frequenza della chirp di riferimento.

Il numero di sinusoidi interferenti generate per ciascun segnale utile è descritto dalla variabile “n\_sin\_jammers” e vengono generate attraverso la seguente formula:

con i ϵ [1; n\_sin\_jammers]

Esempio:   
f\_stop\_array = [150 50], n\_sin\_jammes = 4.  
Freq = [0.125 0.375 0.625 0.875]  
Per la chirp a 150 MHz le interferenti saranno generate alle frequenze: 18.75, 56.25, 93.75 e 131.25 MHz.  
Per la chirp a 50 MHz le interferenti saranno generate alle frequenze: 6.25, 18.75, 31.25 e43.75 MHz.

Il segnale utile può arrivare da una porzione dell’angolo di azimuth compresa tra “azimuth\_start” e “azimuth\_stop”. Il range è spazzato con uno step descritto dalla variabile “azimuth\_step”.

È presente una limitazione però agli angoli di arrivo del segnale interferente: non è contemplato un segnale interferente che si trovi entro “t2j\_distance” gradi dalla direzione di arrivo del segnale utile. Per ogni angolo di arrivo del segnale utile, verranno valutate tutte le combinazioni con gli angoli di arrivo del segnale interferente, a meno di un range attorno all’angolo del segnale utile.

Esempio:   
azimuth compreso tra -60° e 60° per segnale utile  
azimuth compreso tra -80° e 80° per segnale interferente  
t2j\_distance = 20°

Per segnale utile con azimuth = 30°, il range di angoli ammessi per il segnale interferente sarà (-80;10) e (50;80).

Rumore

Al segnale utile e al segnale interferente viene sommato un rumore bianco a media nulla.   
Il rumore viene definito in dB, attraverso il parametro “snr”. Il rumore avrà una potenza tale che il rapporto segnale / rumore con la chirp a frequenza maggiore sarà pari a quanto configurato tramite il parametro “snr”.   
il parametro “snr” è un vettore: assegnando più valori a questo vettori, verranno generati più dataset con rapporti snr diversi.

TERZA SEZIONE: GENERAZIONE DATASET

Il dataset è composto da un certo numero di sample, descritti in seguito. Ciascun sample è composto di 5 attributi, ciascuno nelle rispettive colonne della matrice di dataset:

* Colonna 1: uscita dell’array di antenne, sottoposto a sottocampionamento. Ogni sample è composto di 4 vettori complessi di lunghezza 33'334;
* Colonna 2: angolo di arrivo del segnale utile in gradi;
* Colonna 3: angolo di arrivo del segnale interferente in gradi;
* Colonna 4: frequenza dell’interferente in Hz;
* Colonna 5: presenza o meno dell’interferente. Se ‘0’, nessun interferente, se ‘1’, presente interferente.

La prima sezione del dataset è composta dal solo segnale utile, in assenza di interferente.   
Per ogni chirp nel vettore “f\_stop\_array” viene generato un sample per ciascun angolo spazzato nel range configurato. Il processo viene ripetuto per ciascun livello di SNR impostato.

Il numero totale di sample è pari a:

samples = n\_scenarios \* snr\_length \*

Esempio:  
f\_Stop\_array = [150 50 25] 🡪 n\_scenarios = 3  
snr = [20 10] 🡪 snr\_length = 2  
azimuth start = 60  
azimuth stop = -60  
azimuth step = 0.5

Samples = 3 \* 2 \* (240+1) = 1446

La seconda sezione del dataset è composta da samples in cui al segnale utile è sommato un interferente. Per ciascuna delle casistiche esposte nel punto sopra, vengono generati tanti casi quante sono le possibili direzioni di ciascun segnale interferente. La dimensione del dataset è da moltiplicare quindi per il numero di jammer e per il numero di step che l’interferente impiega per spazzare il range di angoli in azimuth.

Il numero totale di sample è pari a:

samples = n\_sin\_jammers \* n\_jammer\_steps \* n\_scenarios \* snr\_length \*

n\_jammer\_steps =

Prendendo l’esempio esposto sopra e considerando inoltre i parametri:  
n\_sin\_jammers = 3  
azimuth start = 80  
azimuth stop = -80  
azimuth step = 1  
t2j\_distance = 20

Samples = 3 \* 2 \* (240+1) \* 3 \* (120+1)= 524.898