ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA

CORSO DI LAUREA IN INGEGNERIA
ELETTRONICA PER L'ENERGIA E L'INFORMAZIONE

TITOLO DELL'ELABORATO

LOCALIZZAZIONE DI SORGENTI RF TRAMITE CROSS CORRELAZIONE

Tesi in:
COMUNICAZIONI DIGITALI

Relatore:
Chiar.mo Prof. Davide Dardari

Candidato:
Nicolò Calafiore

Correlatore:
Ing. Nicolò Decarli

Sessione III
Anno accademico 2016-17

Sommario

Acronimi:

ADC: Analog to Digital Converter

AM: Amplitude Modulation

AOA: Angle of Arrival

DAC: Digital to Analog converter

DSB-SC: Double Sideband-Suppressed Carrier

DSP: Digital Signal Processing

DVB-T: Digital Video Broadcasting - Terrestrial

EEPROM: Electrically Erasable Programmable Read-Only Memory

ESD: ElectroStatic Discharge

FPGA: Field Programmable Gate Array

GIS: Geographic Information System

GPS: Global Positioning System)

IF: Intermediate Frequency

IoT: Internet of Things

IP: Internet Protocol

ITSs: Intelligent Transportation Systems

LO: Local Oscillator

MCX: Micro Coaxial (connector)

MIMO: Multiple Input Multiple Output

PCB: Printed Circuit Board

RF: Radio Frequency

RSS: Received Signal Strength

RTLS: Real-Time Locating System

SD: Secure Digital

SDR: Software Defined Radio

SNR: Signal Noise Rapport

SRAM: Static Random Access Memory

TCP/IP: Transfert Control Protocol/Internet Protocol

TDOA: Time Difference of Arrival

TOA: Time of Arrival

USB: Universal Serial Bus

USRP2: Universal Software Radio Peripheral 2

Introduzione

La localizzazione è il processo usato per determinare la posizione di un corpo in relazione alla posizione di ciò che lo circonda e ha avuto un'importanza fondamentale sin dalla nascita del genere umano. Basta pensare infatti che già nell'antichità i marinai cercavano di capire dove si trovassero e quale fosse la rotta da seguire osservando le stelle.

Nell'era tecnologica in cui ci troviamo è facile rendersi conto che la nostra posizione è costantemente tracciata e "online" visto che ogni dispositivo elettronico che utilizziamo determina continue localizzazioni in tempo reale. In tutti gli smartphone e computer infatti, è presente un GIS (Geographic Information System) capace di individuarci. Lo stesso Google, ad esempio, ad ogni ricerca ci chiede di poter localizzare la nostra posizione per poter ottimizzare le ricerche in base a ciò che si trova nelle vicinanze. Ecco perché gli studi per migliorare l'affidabilità e l'accuratezza delle applicazioni che implementano la localizzazione è da anni al centro di numerosi studi di ricerca.

La tecnica di localizzazione sicuramente più famosa è quella conosciuta come GPS (Global Positioning System) basata sulla localizzazione satellitare e impiegata maggiormente nell'ambito degli ITSs (Intelligent Transportation Systems), ad esempio per tracciare un pacco in transito, identificare le strade meno trafficate o l'itinerario più breve. Si può capire l'importanza di questa tecnologia anche riferendola alla sicurezza visto che nel momento in cui digitiamo un numero di emergenza la nostra posizione è subito tracciata per favorire un intervento più rapido. Negli ultimi anni però, è emersa sempre di più l'esigenza di sviluppare sistemi capaci di operare in ambienti ostici per la propagazione di trasmissioni radio come ad esempio all'interno degli edifici. Questo tipo di applicazioni indoor trova realizzazione nell'ambito della logistica, per verificare l'autorizzazione ad aree ad accesso limitato, per monitore i pazienti in ospedale o ancora nell'ambito della domotica e della sicurezza automobilistica.

Proprio in questo settore si sono sviluppate svariate tecniche di RTLS (Real Time Locating System) basate su schemi differenti, come per esempio l'AOA (Angle of Arrival) che identifica la direzione di propagazione, l'RSS (Received Signal Strength) per misurare la potenza ricevuta e il TOA (Time of Arrival) che caratterizza la differenza del ritardo di propagazione del segnale. In questo progetto di tesi invece, viene trattata una specifica tecnica TDOA (Time Difference of Arrival) che calcola la differenza del ritardo di propagazione tra segnali e sulla base di questa determina la localizzazione.

L’intento ultimo è quello di riuscire a localizzare dispositivi che emettono segnali radio a bassa frequenza e con banda stretta per supportare l’imminente arrivo della tecnologia basata su reti 5G e sull’IoT (Internet of Things). Il narrowband infatti, sarà importante tanto quanto il broadband dal momento che sempre più oggetti che necessitano di piccole quantità di segnali e lunghi tempi di autonomia saranno connessi alla rete per migliorare la nostra qualità di vita.

Nella prima parte dell’elaborato si illustra il sistema di localizzazione che utilizza il TDOA spiegando il funzionamento e lo scopo di ogni risorsa necessaria. Successivamente, si è scelto di concentrarsi sulla particolarità dell’impiego di ricevitori modificati per renderli coerenti, verificandone i reali vantaggi con test e considerazioni.

Capitolo 1

Architettura del sistema di localizzazione

1.1 Presentazione del progetto

Scopo di questa tesi è quello di porre le basi per un sistema capace di localizzare un dispositivo che emette segnali radio mediante il processo di cross correlazione, per valutarne l'affidabilità ed i limiti.

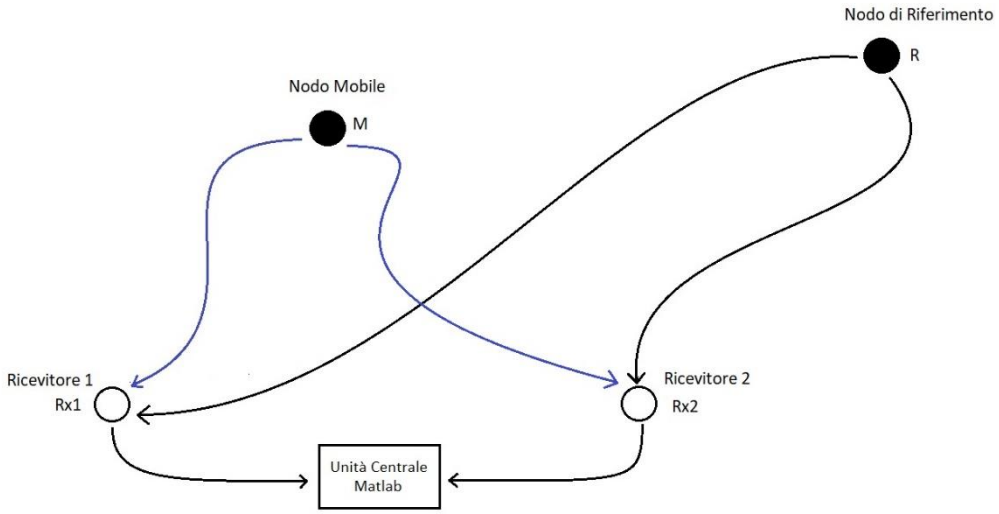

Figura 1: Schematico del progetto

Il sistema è composto dai seguenti elementi:

- Nodo mobile (indicato con M): rappresenta il dispositivo da localizzare, trasmette un segnale $tx(t)$ sotto forma di rumore bianco ad una frequenza e banda nota.
- Nodo di riferimento (indicato con R): trasmette anch'esso un segnale rumoroso $tx_{ref}(t)$ ad una frequenza diversa, ma vicina, a quella del primo nodo. La sua posizione è nota.
- Ricevitore 1 (indicato con Rx1): riceve $s_1(t)$, composto dalla somma dei segnali provenienti da nodo mobile e nodo di riferimento, con una banda sufficientemente larga da poter catturare i segnali provenienti da entrambi i trasmettitori.
- Ricevitore 2: (indicato con Rx2): riceve $s_2(t)$ agendo nello stesso modo del ricevitore 1.
- Unità centrale: computer che elabora i segnali provenienti da entrambi i ricevitori (in questa tesi utilizzando Matlab).

I due ricevitori catturano i segnali provenienti da entrambi i trasmettitori contemporaneamente quindi ciascuno registrerà la sovrapposizione di due segnali

$$s_1(t) = tx(t - t_1) + tx_{ref}(t - t_2) + n_1(t)$$

$$s_2(t) = tx(t - t_3) + tx_{ref}(t - t_4) + n_2(t)$$

dove $n_1(t)$ e $n_2(t)$ rappresentano il rumore ricevuto insieme al segnale utile che ovviamente è indipendente fra i due ricevitori mentre t_1, t_2, t_3, t_4 indicano rispettivamente i ritardi dovuti alla distanza tra M-Rx1, R-Rx1, M-Rx2, R-Rx2

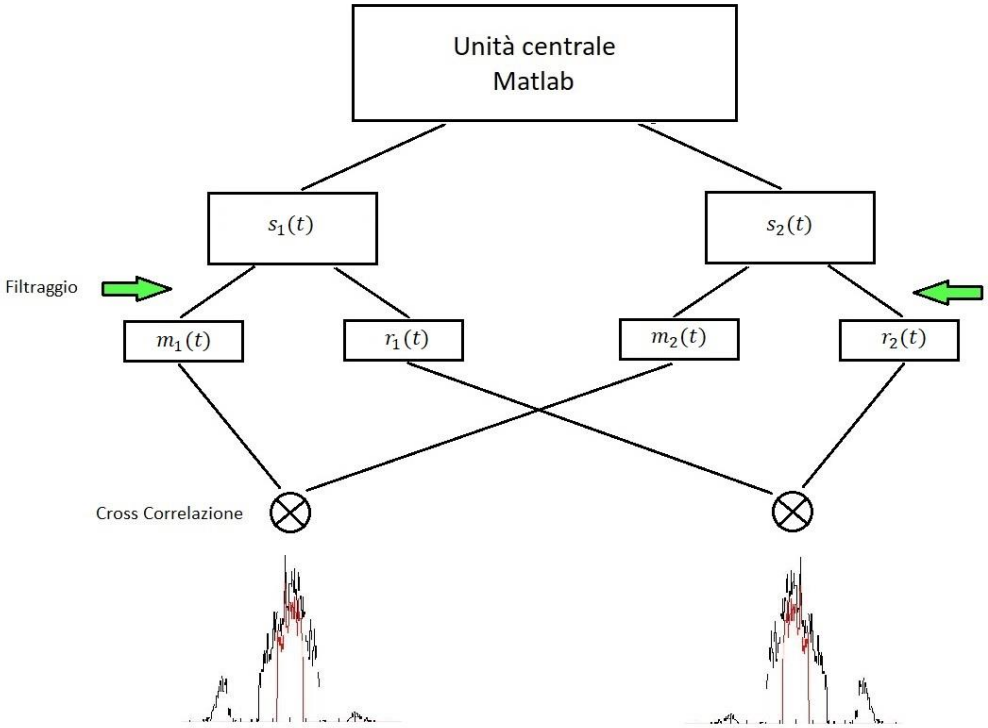

Figura 2: Schema riassuntivo dei passaggi di elaborazione a cui sono sottoposti i segnali ricevuti

Avendo trasmesso i segnali a frequenze diverse, essi possono essere separati mediante un opportuno filtraggio. I due vettori infatti, vengono filtrati prima alla frequenza di trasmissione del nodo mobile e poi a quella del nodo di riferimento ottenendo quattro segnali distinti.

$$s_1(t) \text{ viene scisso in} \begin{cases} m_1(t) = tx(t - t_1) + n_1(t) \\ r_1(t) = tx_{ref}(t - t_2) + n_1(t) \end{cases}$$

$$s_2(t) \text{ viene scisso in} \begin{cases} m_2(t) = tx\big(t - t_3\big) + n_2(t)\big) \\ r_2(t) = tx_{ref}(t - t_4) + n_2(t) \end{cases}$$

Successivamente, la coppia di segnali trasmessa dal nodo mobile e ricevuta da Rx1 e Rx2 viene sottoposta all'operazione di cross-correlazione così da calcolare il ritardo:

$$\mathrm{TDOA_M} = \max \int m_1(t)^* \, m_2(t+\tau) \, dt$$

In condizioni ideali, il ritardo trovato corrisponde alla differenza dei tempi di arrivo (TDOA) del segnale inviato dal nodo mobile e captato dai due ricevitori. Questo è il punto cruciale del progetto perché moltiplicando il TDOA per la velocità della luce il risultato sarà la differenza della distanza del nodo mobile dai due ricevitori nonché la mia incognita. Tutto questo è stato discusso ammettendo alcune condizioni ideali non replicabili con strumenti reali, prima fra tutte la perfetta sincronia degli orologi dei dispositivi riceventi; essi infatti, non inizieranno mai a campionare nello stesso instante quindi la misura della differenza dei tempi di arrivo dei segnali sarà la vera incognita con l'aggiunta di un certo offset non noto.

$$\mathrm{TDOA_M} = \mathrm{TDOA_{Mtrue}} + offset$$

Da questo punto si evince l'utilità del nodo di R; attuando una cross-correlazione anche tra i segnali trasmessi dal nodo di riferimento si otterrà la TDOA tra R-Rx1 e R-Rx2.

$$\mathrm{TDOA_R} = \max \int r_1(t)^* \, r_2(t+\tau) \, dt$$

$$\mathrm{TDOA_R} = \mathrm{TDOA_{Rtrue}} + offset$$

In questo caso però, conoscendo la posizione del nodo di riferimento è possibile calcolare a priori la $\mathrm{TDOA_{Rtrue}}$ esatta e, una volta sottratta alla misura derivante dalla cross-correlazione, estrapolare l'offset tra i due ricevitori per poi sostituirlo nella formula della $\mathrm{TDOA_M}$. In altre parole il nodo di riferimento funge da sincronizzatore per calcolare $\mathrm{TDOA_{Mtrue}}$.

1.2 Cross Correlazione

Questa funzione viene utilizzata per misurare la somiglianza e la differenza temporale tra due segnali.
Si ipotizzi di avere un segnale $s_1\,(t)$ e una sua versione anticipata $s_2(t)$

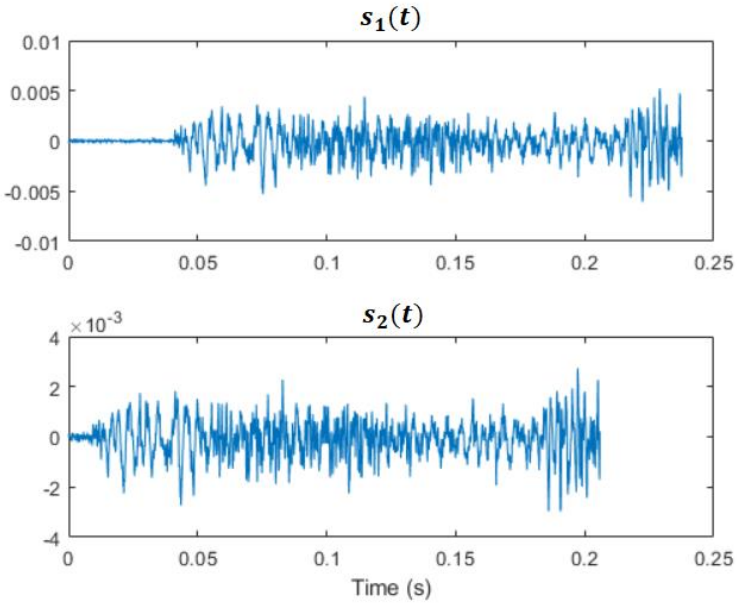

Figura 3: I due segnali $s_1(t)$ e $s_2(t)$ traslati (tratta da [5])

La funzione che implementa la cross-correlazione fra due segnali campionati $s_1(t)$ ed $s_2(t)$, effettua una moltiplicazione campione per campione ed effettua la somma così da ottenere un valore. Questa operazione viene ripetuta ogni volta traslando s2 di un campione (in questo caso posticipandolo) avendo come output un vettore di diversi valori.

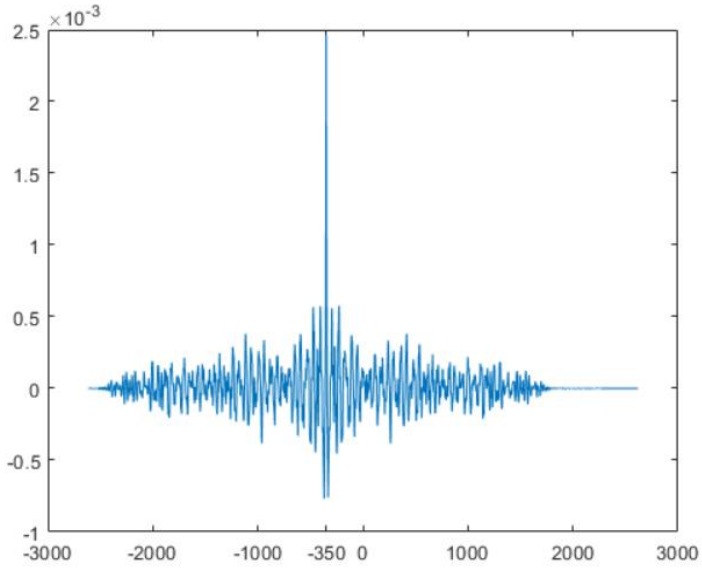

Figura 4: Vettore derivante dalla somma delle moltiplicazioni campione per campione di $s_1(t)$ e $s_2(t)$ (tratta da [5])

Il picco del vettore risultante indica quando i due segnali sono perfettamente sovrapposti (se uguali in partenza) e l'indice di tale picco riporta il numero di campioni di cui sono ritardati (Lag). Infine, dividendo il Lag per la frequenza di campionamento si ottiene di quanti secondi sono ritardati i due segnali.

Capitolo 2

Generazione di segnali arbitrari a RF e loro ricezione

I test effettuati in questa tesi sono stati realizzati tramite dei radiocollegamenti dov'è stata curata sia la parte di trasmissione che quella di ricezione.

2.1 Trasmissione: USRP2 (Universal Software Radio Peripheral 2)

Per la trasmissione dei segnali è stata utilizzato il ricetrasmettitore USRP2, seconda versione dell'omonima primo tipo sviluppata nel 2008 da parte di Ettus Reasearch, azienda leader nel campo dei SDR.

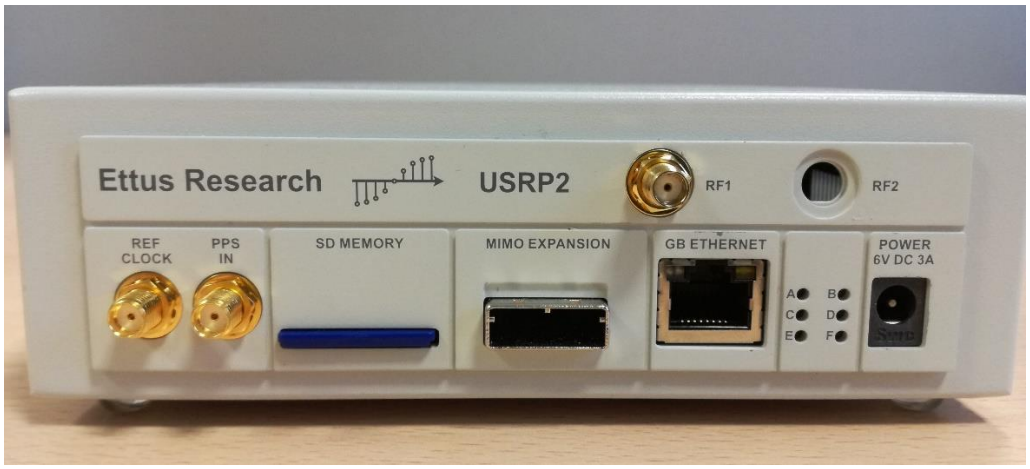

Figura 5: Pannello frontale dell'USRP2

2.1.1 Architettura interna

L'architettura interna dell'USRP2 è composta dai seguenti componenti:

- Interfaccia Gigabit Ethernet per collegare il dispositivo ad un host computer che funga da workstation
- FPGA Xilinx Spartan 3-2000 che si occupa dei filtraggi e dei campionamenti
- 2 ADC da 14 bit a 100MS/s;
- 2 DAC da 16 bit a 400MS/s;
- Ingresso per cavo MIMO
- Lettore SD-Card su cui è riposto il firmware della piattaforma
- SRAM da 1 Mbyte

Oltre a questi, l'USRP possiede la capacità di supportare varie tipologie di doughterboard intercambiabili a seconda dell'utilizzo necessario. Per questo progetto si è impiegata la WBX che possiede una banda si lavoro compresa tra 50 MHz e 2.2GHz.

2.1.2 Interfacciamento dell'USRP2 con Matlab/Simulink

Premessa: per questo progetto di tesi è stato utilizzato un host computer con Windows 10 e Matlab versione 2017b.

Prima di iniziare con la procedura di configurazione della USRP2 occorre verificare di possedere alcuni requisiti. Innanzitutto bisogna assicurarsi che il PC impiegato come workstation possieda un processore da 64bit ed anche una porta Gigabit Ethernet poiché, in caso contrario, il computer non riconoscerebbe il dispositivo. Tuttavia, se ne è sprovvisti, si può ricorrere ad una valida alternativa per altro utilizzata in questo progetto; basta infatti utilizzare un adattatore Gigabit Ethernet-USB 3.0 capace di creare una scheda di rete esterna e dedicata che riesce a riconoscere la piattaforma sfruttando la velocità della porta USB di terza generazione. Se si dispone soltanto di porte USB 2.0 questa soluzione non è attuabile. Riguardo al cavo Ethernet di collegamento, verificare che sia almeno di categoria cat5E in quanto le versioni precedenti gestiscono una velocità massima di soli 100Mbps (Fast Ethernet). L'altro requisito fondamentale è il download dei seguenti toolbox:

- MathWorks DSP System Toolbox
- MathWorks Communications System Toolbox
- MathWorks Signal Processing Toolbox
- Communications System Toolbox Support Package for USRP Radio

Giunti a questo punto, accedendo alle impostazioni della scheda di rete dal pannello di controllo di Windows, bisogna cambiare l'indirizzo IP della porta del PC impostandolo su 192.168.10.1 con subnet 255.255.255.0 così che si possa interfacciare con l'USRP2 che di default possiede l'IP 192.168.10.2.

Ovviamente se si utilizzano due USRP2 simultaneamente il loro indirizzo IP dovrà essere diverso.

Per verificare che la configurazione della piattaforma sia avvenuta con successo si può innanzitutto provare a pingare dal prompt dei comandi l'indirizzo dell'USRP2 precedentemente impostato per vedere se si riceve risposta oppure si può digitare *findsdru* direttamente dal command window di Matlab e attendere il riconoscimento del dispositivo.

Per il corretto impiego dell'USRP2 non bisogna dimenticare alcune avvertenze illustrate sul sito dell'azienda costruttrice, tra cui:

- In fase di trasmissione deve esserci sempre un'antenna collegata o un carico da 50Ω
- Non applicare mai più di -15dBm di potenza in input
- Applicare almeno 30dB di attenuazione se si sta lavorando in loopback

2.2 Ricezione: RTL-SDR (Software Defined Radio)

L'RTL-SDR è un dispositivo USB a basso costo (meno di 20$), in grado di ricevere segnali radio. Inizialmente fu pensato come ricevitore DVB-T (Digital Video Broadcast-Terrestrial) ossia per acquisire i segnali della TV digitale su computer e solo successivamente si è scoperto potesse ricevere un'intera porzione di banda nello spettro 25MHz-1.7GHz.

Figura 6: RTL-SDR con la sua antenna omnidirezionale collegato al PC

2.2.1 RTL_SDR PCB

Il dispositivo, estratto dall'involucro plastico, appare come illustrato nell'immagine sottostante, dove sono evidenziati i componenti principali:

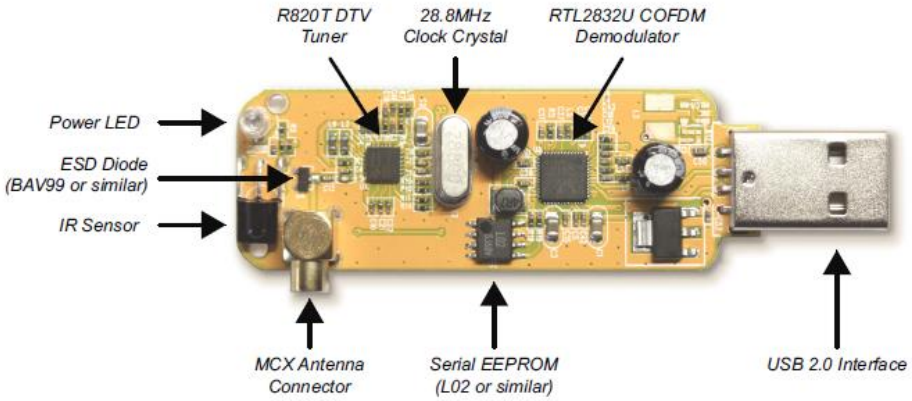

Figura 7: Circuito stampato dell'RTL_SDR (tratto da [2])

- **MCX:** connettore dell'antenna del dispositivo
- **ESD (Electrostatic Discharge Diode):** diodo di protezione contro eventuali scariche elettromagnetiche provenienti dall'antenna
- **R820T:** Tuner chip che seleziona una porzione dello spettro RF del segnale in ingresso e lo converte alla frequenza intermedia IF
- **RTL2832U:** chip demodulatore che converte la frequenza IF in banda base, digitalizza il segnale e riduce la frequenza di campionamento
- **Clock Crystal da 28.8MHz:** genera l'oscillatore locale e il clock
- **Interfaccia USB 2.0:** invia i campioni IQ in banda base al PC
- **Sensore infrarosso (IR):** permette il controllo remoto del dispositivo
- **EEPROM (Elettronically Eraseble Programmable Read Only Memory):** contiene le informazioni necessarie a configurare l'SDR tramite USB.

2.2.2 Architettura NooElec NESDR RTL-SDR

L'Architettura interna della RTL-SDR consiste nella cascata di due componenti principali:

- **Rafael Micro R820T Silicon Tuner:** questo tuner applica una downconversion del segnale RF in ingresso con banda massima di 6 MHz ad una IF (intermediate frequency) di 3.57 MHz.
- **Realtek RTL2832U:** il segnale IF uscente dal tuner arriva a questo secondo blocco e viene convertito una seconda volta per portarlo in banda base. Successivamente un ADC con un rate di 28.8MHz campiona il segnale quantizzandolo in 8 bit corrispondenti a 256 livelli per poi applicare una demodulazione AM in fase e in quadratura producendo i campioni I e Q. Infine, attuando un'operazione di decimazione per ridurre la frequenza di campionamento a 2.8 MHz, i campioni vengono inviati al PC tramite la porta USB.

Tutto il processo è ben illustrato nell'immagine sottostante

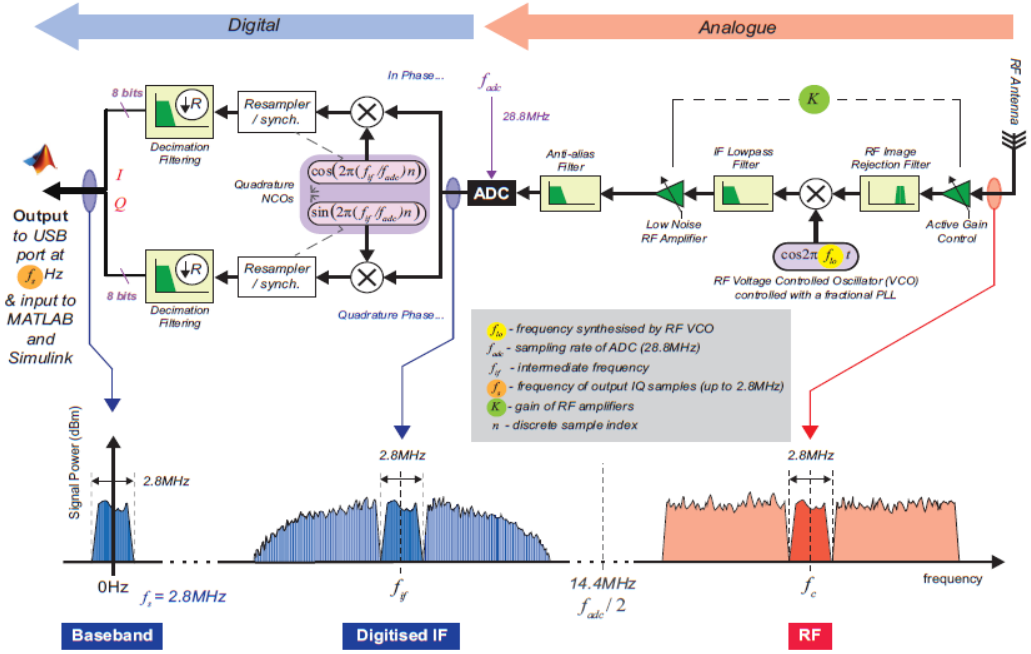

Figura 8: Processo di elaborazione del segnale all'interno della RTL-SDR (tratto da [2])

2.2.3 Interfacciamento dell'SDR con Matlab:

Per interfacciare RTL-SDR a Matlab, oltre a possedere il ricevitore USB e una versione di Matlab R2014 o successiva, è necessario scaricare i seguenti add-ons:

- MathWorks DSP System Toolbox
- MathWorks Communications System Toolbox
- MathWorks Signal Processing Toolbox
- Communications System Toolbox Support Package for RTL-SDR Radio

Se si possiedono tutti i Toolbox, collegando l'RTL-SDR al PC tramite USB i driver di installazione verranno scaricati automaticamente e in pochi minuti è possibile configurare il dispositivo.

Per conferma dell'avvenuta installazione basta digitare *sdrinfo* dal command window e attendere che compaiano le caratteristiche della radio.

Grazie al comando *comm.SDRRTLReceiver* è possibile impostare via Software tutti i paramenti di ricezione e quindi bufferizzare i vettori contenenti i campioni delle vie I e Q.

2.2.4 Gestione della ricezione tramite Matlab

Matlab permette di modificare via software vari parametri che definiscono il settaggio del ricevitore. Essi sono:

- *rtlsdt_id:* identificativo dell'SDR collegato al pc per primo
- *rtlsdr_tunerfreq:* frequenza del segnale che si vuole ricevere espressa in Hz
- *rtlsdr_gain:* guadagno del ricevitore espresso in dB
- *rtlsdr_fs:* frequenza di campionamento espressa in Hz. Per assicurare un campionamento accurato senza perdita di dati si consiglia di non superare il valore di 2.8MHz
- *rtlsdr_frmlen:* size di ciascun frame catturato dall'SDR. Il valore massimo consentito è 375000
- *rtlsdr_datatype:* tipo di dato in output
- *rtlsdr_ppm:* correzione di errore in termini di parti per milione
- *sim_time:* numero frame acquisiti dal dispositivo.

Sono stati collegati al PC due RTL-SDR per emulare i ricevitori del sistema di localizzazione

```
4   % PARAMETRI
5   rtlsdr_id            = '0';
6   rtlsdr_id_1          = '1';
7   rtlsdr_tunerfreq    = 73e6;
8   rtlsdr_gain         = 30;
9   rtlsdr_fs            =1e6;
10  rtlsdr_frmlen      = 256*5;
11  rtlsdr_datatype   = 'single';
12  rtlsdr_ppm          = 0;
13  sim_time             = 1000;
```

Figura 9: Script del settaggio dei due SDR_RTL

2.2.5 Funzione Parfor

Per fare in modo che con un unico *Run* Matlab avvii la ricezione su entrambi i ricevitori è stata usata la funzione *parfor*, disponibile all'interno del Parallel Computing Toolbox, in grado di assegnare funzioni distinte ad un pool di più workers che le eseguono parallelamente. Questo ha reso possibile l'implementazione della funzione *rtlsdr_rx* su due thread paralleli che avviano simultaneamente la ricezione e salvano i campioni IQ nelle variabili *samplesIQ* e *samplesIQ1* rispettivamente per ognuna delle due SDR. L'uso di Parfor però, ha portato delle complicazioni in quanto le variabili definite nel corpo della funzione non possono essere riutilizzate all'esterno, quindi è stato necessario salvare i valori in formato .mat per poi poterli caricare su nuove variabili con il comando *load*. Riguardo al salvataggio si è dovuto ricorrere ad una funzione esterna piuttosto che al classico *save* per non invalidare la trasparenza del *parfor*.

Per quanto comoda, la creazione del pool (*parpool*) richiede un tempo che nel migliore dei casi oscilla intorno ai 22 secondi e che aumenta tanto più sono i workers impiegati. Questo effetto indesiderato rallenta notevolmente l'elaborazione perché, ad ogni avvio, il pool necessita di essere inizializzato nuovamente nonostante la funzione non subisca modifiche.

```
15 -    parpool('local', 2);
16
17 -    parfor i = 1:2
18
19 -        if i == 1
20            samplesIQ=rtlsdr_rx(rtlsdr_id,rtlsdr_tunerfreq,rtlsdr_gain,
21                                rtlsdr_fs,rtlsdr_frmlen,rtlsdr_datatype,rtlsdr_ppm,sim_time);
22 -          salva(sprintf('campioniIQ%d.mat'), samplesIQ);
23 -        else
24            samplesIQ1=rtlsdr_rx(rtlsdr_id_1,rtlsdr_tunerfreq,rtlsdr_gain,
25                                 rtlsdr_fs,rtlsdr_frmlen,rtlsdr_datatype,rtlsdr_ppm,sim_time);
26 -          salva1(sprintf('campioniIQ1%d.mat'), samplesIQ1);
27 -        end
28
29 -    end
```

Figura 10: Script Parfor

Occorre puntualizzare che, nonostante il *parfor*, l'inizio della ricezione da parte delle due SDR non sarà mai perfettamente sincrona quindi ci sarà sempre un offset a viziare le misure de tempi di arrivo per cui si rende necessario l'utilizzo del nodo di riferimento.

Capitolo 3

Risultati sperimentali

Sul finire del progetto di tesi si è scelto di focalizzarsi su alcuni aspetti fondamentali per poter implementare un localizzatore che utilizzi una tecnologia basata su TDOA.

3.1 Ricevitori Coerenti

Affinché il sistema di localizzazione possa individuare correttamente la posizione del nodo mobile, occorre puntualizzare alcuni aspetti propri della ricezione. Per prima cosa si valuti come avviene la modulazione AM DSB-SC dei segnali trasmessi tramite USRP2 e la corrispondente demodulazione da parte delle due RTL-SDR nel momento in cui li ricevono:

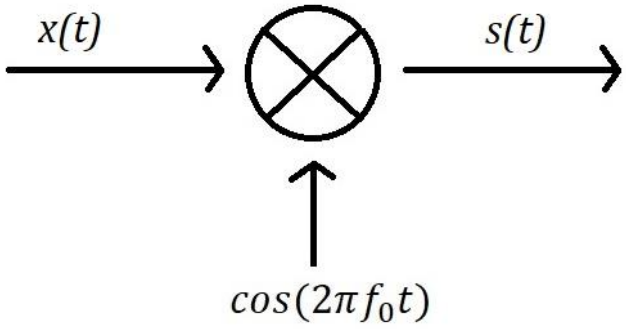

Figura 11: schema a blocchi del modulatore DSB-SC

L'USRP2 modula il segnale $x(t)$ moltiplicandolo per $\cos(2\pi f_0 t)$ ottenendo:

$$s(t) = x(t)\cos(2\pi f_0 t)$$

Il modulato $s(t)$ viene poi inviato all'antenna e trasmesso in aria.

Trasmettitore e ricevitore non sono collegati tra loro quindi non hanno un riferimento temporale comune, questo comporta che la portante in trasmissione e il segnale di riferimento in ricezione non siano esattamente lo stesso coseno ma saranno sfasati di una quantità Δ. In ricezione infatti si ha

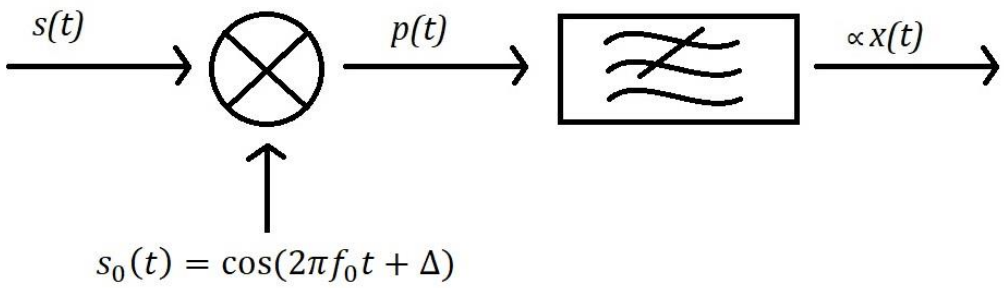

Figura 12: schema a blocchi del demodulatore DSB-SC

$s(t)$ ricevuto dalle due RTL-SDR, viene moltiplicato per il segnale di riferimento $s_0(t)$ imposto dall'oscillatore locale ottenendo

$$p(t) = x(t)\cos(2\pi f_0 t)\cos(2\pi f_0 t + \Delta) = \frac{1}{2}\, x(t)\cos(\Delta) + \frac{1}{2}x(t)\cos(4\pi f_0 t + \Delta)$$

Dopodiché il filtro passa-basso elimina la componente a frequenza $2f_0$ ottenendo all'uscita un segnale proporzionale a $x(t)$.

Considerando due ricevitori distinti però, essi sono dotati di due oscillatori locali indipendenti e non controllabili via software quindi la demodulazione processata dalle due SDR utilizzerà due segnali di riferimento sfasati tra loro

All'uscita di Rx1: $\frac{1}{2}\, x(t)\cos(\Delta_1)$

All'uscita di Rx2: $\frac{1}{2}\, x(t)\cos(\Delta_2)$

In questo modo i due ricevitori invieranno a Matlab due segnali diversi rendendo inutile la cross-correlazione (il tutto sia per la via I che la via Q).

3.2 Modifica Hardware

La modifica hardware implementata in questo progetto è servita per risolvere proprio il problema sopra esposto. Per fare in modo che i due oscillatori siano in fase è stato dissaldato il clock crystal di una delle due radio (slave) e ne sono stati collegati i pin all'oscillatore del secondo ricevitore (master) mediante un cavo coassiale da 75 Ohm.

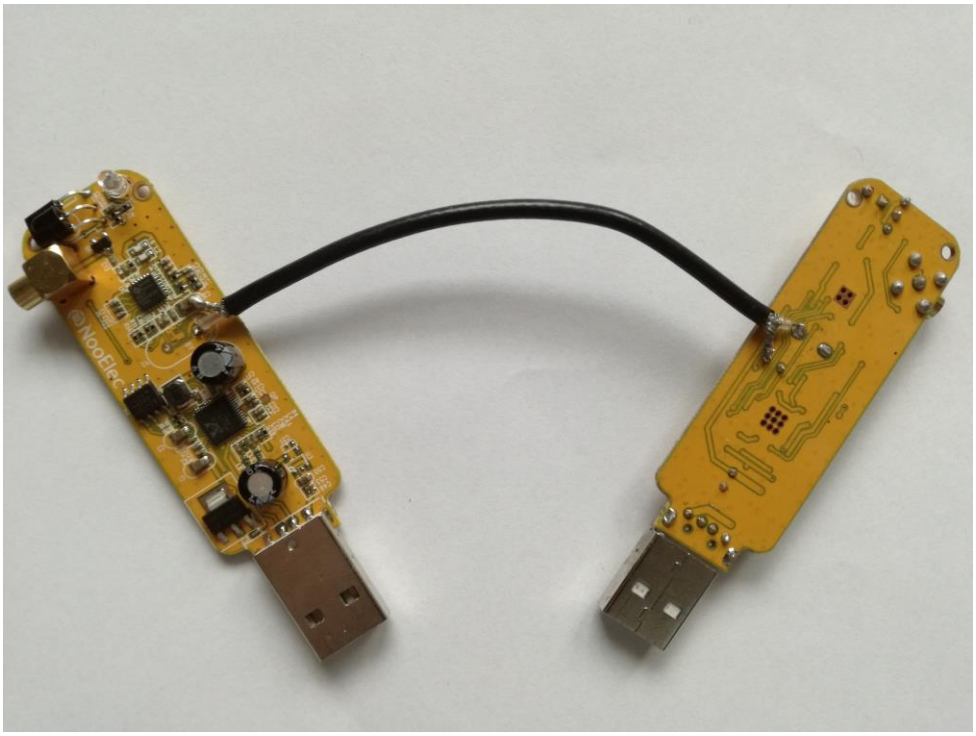

Figura 13: Le due RTL-SDR rese coerenti grazie alla modifica Hardware

In questo modo si sono sincronizzate le due SDR in modo che operassero con lo stesso oscillatore locale così che abbiano un riferimento temporale comune e possano restituire in output lo stesso segnale solamente ritardato a causa della differenza del tempo di arrivo.

3.3 Test

Per verificare l'efficacia e la correttezza della modifica sono stati effettuati alcuni test; il più importante e significativo è stato quello di trasmettere del rumore bianco a banda di 1Mhz e riceverlo prima con due ricevitori non coerenti e successivamente con due coerenti per poi valutare la differenza tra le cross-correlazioni dei segnali ricevuti nei due casi.

Per allestire il radiocollegamento si è reso necessario l'utilizzo di due PC, uno che si interfacciasse con l'USRP2 per la trasmissione e uno con l'RTL-SDR per la ricezione.

In entrambi i casi la trasmissione è stata gestita con la seguente catena Simulink

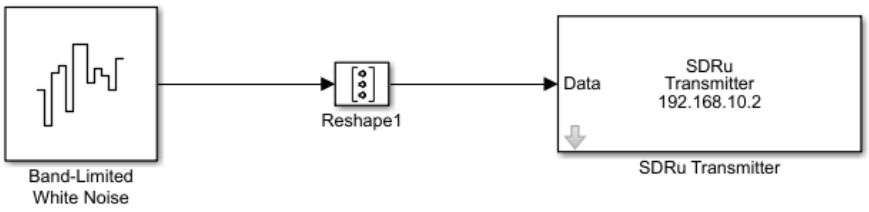

Figura 14: Schema a blocchi Simulink della trasmissione

Con i parametri

Band-Limited White Noise:

- Noise Power: [0.1] (default)
- Sample time :10e-4 (corrisponde ad una frequenza di campionamento di 10Khz)
- Seed: [23341] (default)

Reshape: (necessario perché il blocco trasmettitore richiede l'input come vettore colonna)

- Output dimensionality: Column Vector

SDRu Transmitter:

- USRP IP address: 192.168.10.2
- Channel Mapping: 1
- Center Frequency: 433.9e6 Hz
- LO offset: 0 Hz
- Gain: 8 dB
- Master clock rate: 100e6 Hz (default)
- Interpolation: 500

Occorre inoltre specificare che il Simulation Stop time è stato impostato su “inf” per poter trasmettere per un tempo illimitato e il Simulaton mode su “Accelerator” per sfruttare la velocità massima computazionale di Simulink.

Inoltre, le antenne dei ricevitori sono state posizionate il più vicino possibile in modo da limitare al minimo la differenza dei tempi di arrivo tra i segnali ricevuti cosicché la cross-correlazione restituisca solo il ritardo dovuto al campionamento non sincrono. Per quanto riguarda l’antenna trasmittente dell’USRP2, è stata anteposta da un attenuatore di potenza di 20dB affinché i segnali ricevuti non fossero troppo elevati così da evitare saturazione al ricevitore. I ricevitori sono stati posizionati ad una distanza di circa 1.20 metri dal trasmettitore.

La ricezione invece è stata gestita con il codice Matlab discusso nel capitolo 2 con i seguenti parametri

```
% PARAMETRI
rtlsdr_id             = '0';
rtlsdr_id_1           = '1';
rtlsdr_tunerfreq    = 433.9e6;
rtlsdr_gain          = 30;
rtlsdr_fs             =1e6;
rtlsdr_frmlen       = 256*50;
rtlsdr_datatype    = 'single';
rtlsdr_ppm           = 0;
sim_time              = 500;
```

Figura 15: Settaggi delle due RTL-SDR in ricezione

Sono state effettuate dieci catture con conseguente calcolo della cross-correlazione in entrambi i casi.

```
% Calcolo l'offset
close all;
clc;
x=load('campioniIQ.mat');
x1=load('campioniIQ1.mat');

s1=x.samplesIQ';
s2=x1.samplesIQ1';

fs=1e6;
[acor,lag] = xcorr(real(s1),real(s2));
[~,I] = max(abs(acor));
lagDiff = lag(I)
offset = lagDiff/fs

figure
plot(lag,acor)
a3 = gca;
a3.XTick = sort([0 lagDiff]);
```

Figura 16: Script Matlab che implementa la cross-correlazione tra le parti reali dei segnali ricevuti

3.4 Risultati ottenuti

Ricevitori non coerenti:

La tabella 1 mostra il risultato del ritardo (in numero di campioni) fra i due segnali ottenuto nelle dieci prove con chiavette non modificate per avere il medesimo clock. Si ottiene un ritardo medio pari a 426567.2 campioni e una deviazione standard di 284342.55 campioni.

La figura 17 mostra un esempio di funzione di cross-correlazione fra due segnali collezionati in uno dei dieci test. Come possibile notare, non si ottiene la funzione delta di Dirac attesa, come riportato nel capitolo 1.2.

	Lag
Test 1	-21262
Test 2	-540843
Test 3	-221225
Test 4	578368
Test 5	744355
Test 6	266014
Test 7	388972
Test 8	-884338
Test 9	64918
Test 10	555377

Tabella 1: dieci catture con ricevitori non coerenti e rispettivi Lag

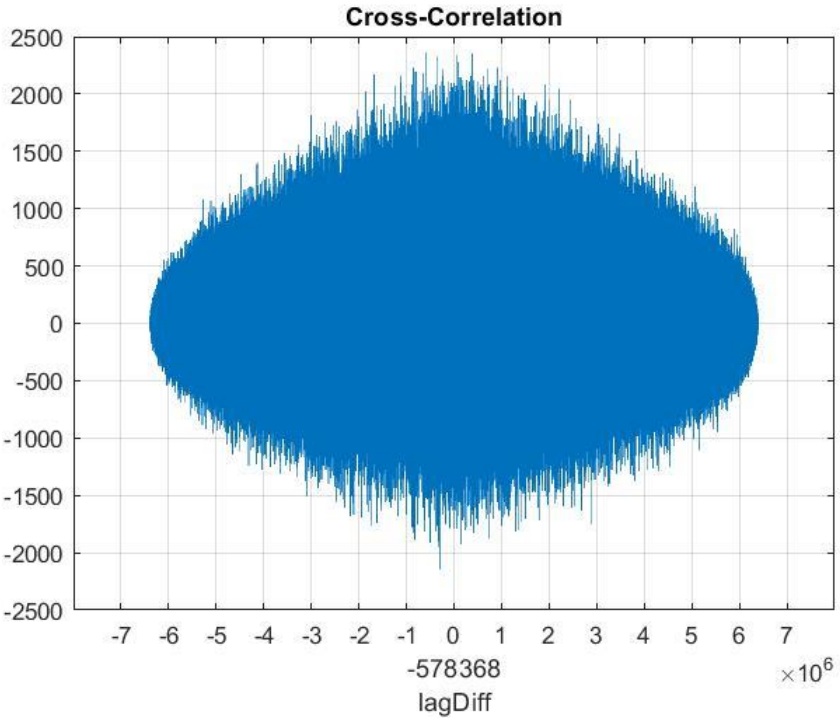

Figura 17: plot Matlab della cross-correlazione tra due segnali acquisiti con ricevitori non coerenti

Ricevitori coerenti:

La tabella 2 mostra invece, il risultato del ritardo (in numero di campioni) fra i due segnali ottenuto nelle dieci prove con chiavette modificate. Si ottiene un ritardo medio pari a 2866.8 campioni e una deviazione standard di 5607.97 campioni.

La figura 18 mostra un esempio di funzione di cross-correlazione fra due segnali collezionati in uno dei dieci test. Come possibile notare, si ottiene la funzione delta di Dirac attesa.

	Lag
Test 1	344
Test 2	-498
Test 3	-542
Test 4	-503
Test 5	3232
Test 6	1663
Test 7	-187
Test 8	18527
Test 9	405
Test 10	2767

Tabella 2: dieci catture con ricevitori coerenti e rispettivi Lag

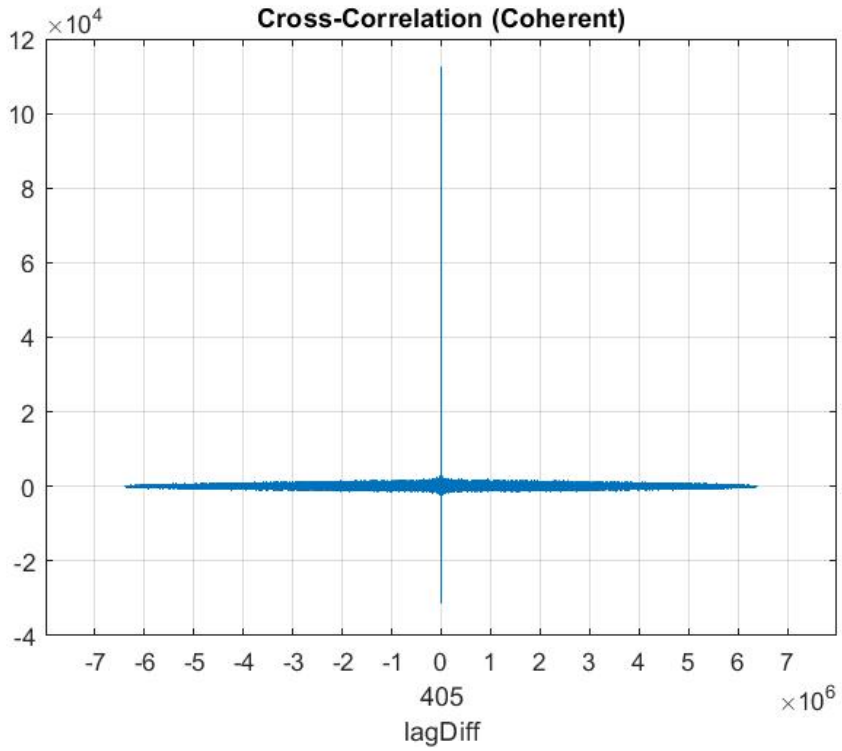

Figura 18: plot Matlab della cross-correlazione tra due segnali acquisiti con ricevitori coerenti

3.5 Calcolo dell'offset

Per calcolare l'offset medio tra i due ricevitori è bastato dividere la media dei Lag per la frequenza di campionamento in ricezione:

$$offet_{medio} = \frac{Lag_{medio}}{fs} = 0{,}0029\ s$$

Questo valore indica l'offset medio che vizia le misure del calcolo della differenza del tempo di arrivo implementata dalla cross-correlazione che si utilizzerebbe nel sistema di localizzazione finale.

Per mancanza di tempo non si è potuto concludere il test del sistema di localizzazione complessivo mentre si è scelto di approfondire gli aspetti sopra esposti relativi alla ricezione con RTL-SDR coerenti.

Conclusioni

Dai test effettuati si nota innanzitutto che realizzare la cross-correlazione tra segnali derivanti da due ricevitori non coerenti riporta valori di ritardo casuali visto che ha senso farla solo con due segnali identici e ritardati tra loro; questo è testimoniato da valori di ritardo molto alti e soprattutto dal grafico della cross-correlazione che in tutte le catture appare molto lontano dall'essere una delta di Dirac. I reali valori di ritardo espresso in termini di campioni infatti, si hanno solo nel caso di utilizzo delle due RTL_SDR coerenti dove le demodulazioni avvengono con lo stesso segnale di riferimento. Trasmettendo rumore bianco, la vera conferma dell'efficacia della modifica è rappresentata dal plot della cross-correlazione che in tutti i test eseguiti risulta essere molto simile ad una delta di Dirac che sarebbe perfetta solo nel caso in cui i due segnali fossero perfettamente identici (ovvero nel medesimo canale di propagazione, in assenza di rumore e di ulteriori non-idealità).

Nella seconda fase di test si è cercato di capire su quali parametri in ricezione lavorare per migliorare il valore di offset medio. Provando a cambiare frequenza di campionamento, numero di workers impiegati, numero di frame e campioni acquisiti non si sono rilevati miglioramenti significativi quindi il valore di offset trovato è molto vicino a quello rappresentato dal limite della tecnologia impiegata. Considerando l'economicità e la semplicità delle RTL-SDR utilizzate, l'offset calcolato rappresenta un buon valore perché molto vicino alla grandezza di tolleranza di un pacchetto TCP/IP.

In ultima analisi si può affermare che la modifica hardware dei ricevitori RTL-SDR è assolutamente necessaria per la corretta implementazione del sistema di localizzazione basato su cross-correlazione.

Per lo sviluppo futuro del sistema, al fine di ottenere risultati migliori possibile bisogna fare alcune considerazioni relative soprattutto alla trasmissione. La frequenza della portante dei due segnali trasmessi è preferibile che sia scelta al di sotto dei 100MHz perché, a basse frequenze, gli ostacoli e gli echi interagiscono in minor misura con la propagazione delle onde elettromagnetiche. Un secondo fattore fondamentale è la banda del segnale modulato, in quanto, se da un lato è interessante capire fin dove la tecnologia possa lavorare con una banda più piccola possibile, dall'altro essa determina la precisione di stima del tempo di arrivo del segnale; ci si trova quindi nella situazione di dover ricercare il giusto compromesso.

Appendice

Richiami

A.1 Modulazione

Si consideri un generico segnale modulante $x(t)$ (il portatore di informazione), di natura passa basso, con banda di frequenza massima f_m e la portante sinusoidale che oscilla ad una frequenza $f_0 \gg f_m$ espressa dalla formula:

$$s_0 = V_0 \cos(2\pi f_0 t + \varphi_0)$$

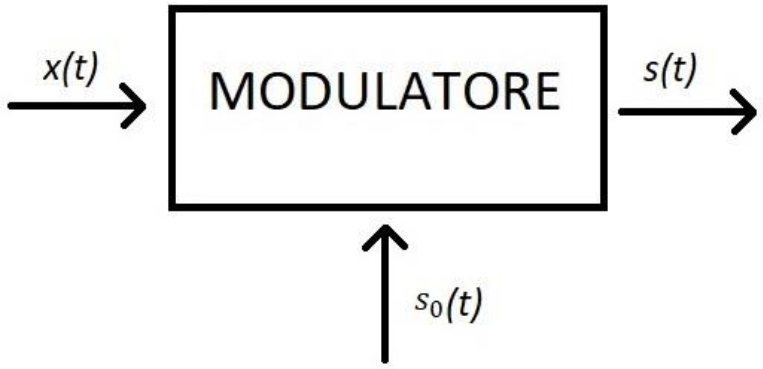

Figura A1: Schema Input-Output del modulatore

Il modulatore modula la portante sulla base alla legge di modulazione ottenendo un segnale con spettro traslato attorno alla frequenza portante; ne consegue che si passi da un segnale passa basso $x(t)$ ad un segnale passa banda $s(t)$.

Un segnale genericamente modulato ha la seguente formulazione:

$$s(t) = V(t)\cos\big(\varphi(t)\big)$$

Se all'interno di questa espressione si sostituiscono le deviazioni

- Deviazione relativa d'ampiezza: $\frac{V(t)-V_0}{V_0}$
- Deviazione di fase: $\alpha(t) = \varphi(t) - (2\pi f_0 t + \varphi_0)$
- Deviazione di frequenza: $\Delta f(t) = f(t) - f_0 = \frac{\dot{\alpha}(t)}{2\pi}$

  Con $\alpha(t) = 2\pi \int_{-\infty}^{t} \Delta f(\xi) d\xi + k$

si ottiene la definizione ancora più generale e completa di segnale modulato:

$$s(t) = [1 + m(t)]V_0\cos[2\pi f_0 t + \alpha(t) + \varphi_0]$$

È interessante notare che se $m(t)$=0 e $\alpha(t) = 0$ degenera in

$$s_0 = V_0\cos(2\pi f_0 t + \varphi_0)$$

Occorre specificare che esistono varie tipologie di modulazioni che vengono impiegate a seconda di alcuni fattori come la distanza fra trasmettitore e ricevitore, il mezzo trasmissivo, la natura dell'informazione e la qualità di trasmissione desiderata. Esse sono:

- Modulazione AM: modulazione di ampiezza
- Modulazione FM: modulazione di frequenza
- Modulazione PM: modulazione di fase
- Modulazioni ibride: quando la legge di modulazione non è univoca

Non è scopo di questa tesi approfondirle tutte ma si riporta qualche accenno teorico relativo alla modulazione d' ampiezza perché è quella implementata nei test svolti.

A.2 Modulazione AM

Questo schema di modulazione modula la sinusoide portante facendone variare l'ampiezza in accordo all'ampiezza del segnale modulante.

Legge di modulazione:

$$\begin{cases} m(t) = k_a x(t) \\ \alpha(t) = 0 \end{cases}$$

Con k_a detta sensibilità

Espressione generica di un segnale modulato in ampiezza:

$$s(t) = V_0\,[1 + k_a x(t)]\cos(2\pi f_0 t + \varphi_0)$$

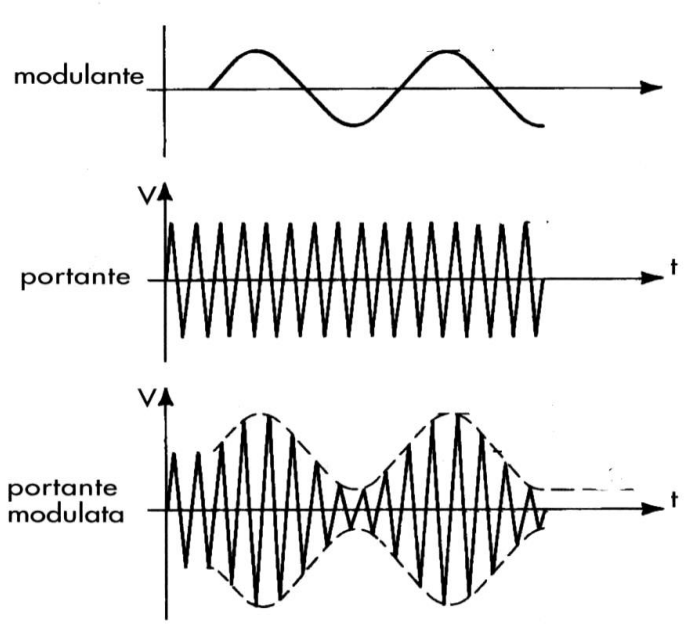

Figura A2: Modulazione AM (tratto da [4])

A.3 Rumore Bianco a banda limitata

Il rumore bianco $x(t)$ si ottiene da un insieme di variabili aleatorie indipendenti con la stessa densità di probabilità e avente densità spettrale di potenza costante $G_x(f) = \frac{N_0}{2}$.

Definendo la funzione di autocorrelazione come

$$R(\tau) = E[x(t)x(t-\tau)]$$

E sapendo che essa è legata alla densità spettrale di potenza dalla relazione

$$G_x(f) = \mathcal{F}\{R(\tau)\}$$

Risulta evidente che, applicando l'antitrasformata di Fourier

$$R(\tau) = \frac{N_0}{2}\delta(\tau)$$

In altre parole la funzione di autocorrelazione è pari a una delta di Dirac che è la miglior funzione di autocorrelazione che ci può ottenere perché si ottiene un unico valore $\frac{N_0}{2}$ per $\tau = 0$ e tutti gli altri a zero. Questo fa sì che un segnale di questo tipo possa essere riconosciuto, se ritardato, senza ambiguità.

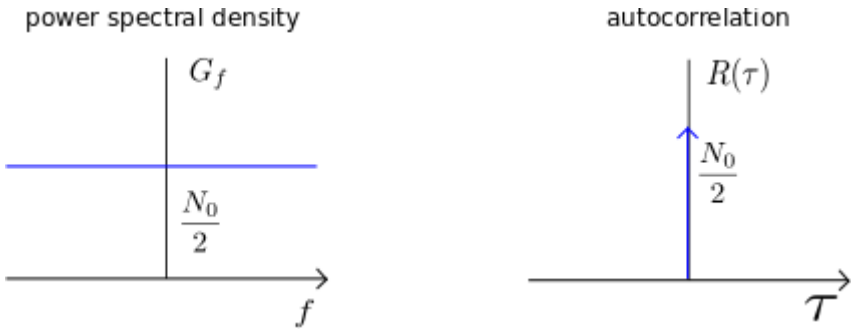

Figura A3: Relazione tra $G_x(f)$ e $R(\tau)$ (tratta da [4])

Questa proprietà è vera per un rumore bianco a banda illimitata che di fatto è un'idealizzazione perché nessun apparecchio elettronico può generarlo. Nella pratica si utilizza lo stesso rumore ma con una banda limitata la cui la funzione di autocorrelazione sarà simile ad una delta di Dirac tanto più larga sarà la banda considerata.

Bibliografia

[1] https://github.com/PetarV/TikZ/tree/master/Frequency%20modulation

[2] Software Defined Radio using Matlab & Simulink and the RTL-SDR di Bob Stewart, Kenneth Barlee, Dale Atkinson e Louise Crockett

[3] http://www.dsplog.com/2012/03/25/thermal-noise-awgn/

[4] https://wordpress.com/4-elettronica/

[5] https://it.mathworks.com/help/signal/ref/xcorr.html

[6] Leonardo Calandrino e Marco Chiani, *Lezioni di Comunicazioni Elettriche*, Pitagora Editrice Bologna, 2013

[7] Davide Dardari, Emanuela Falletti e Marco Luise, *Satellite and Terrestrial Radio Positioning Techniques*, Elsevier Academic Press, 2012

[7] Alexander M. Wyglinski, *Digital Communication Systems Engineering with Software-Defined Radio*, Artech House, 2013

[8] B. Stewart,K. Barlee, D. Atkinson, L. Crockett, *Software defined radio using Matlab and Simulink and the RTL-SDR*, Strathclyde Academic Media, 2015