

**ALMA MATER STUDIORUM  
UNIVERSITÀ DEGLI STUDI DI BOLOGNA  
SEDE DI CESENA**

---

Seconda Facoltà di Ingegneria con sede a Cesena  
Corso di Laurea in Ingegneria Elettronica e Telecomunicazioni

**ANALISI E IMPLEMENTAZIONE DI  
ALGORITMI DI DETECTION PER  
SISTEMI AD ANTENNE  
MULTIPLE SU PIATTAFORMA  
PROGRAMMABILE USRP2**

Tesi in  
Laboratorio di Elettronica e Telecomunicazioni

Presentata da:  
FORTI ENRICO

Relatore:  
Prof. Ing.  
ANDREA GIORGETTI

Correlatore:  
Ing.  
ANDREA MARIANI

---

SESSIONE I  
ANNO ACCADEMICO 2011-2012



# **PAROLE CHIAVE**

Cognitive Radio

Spectrum Sensing

USRP2

AGM Detection



# Indice

<b>Capitolo1 – Introduzione.....</b>	<b>1</b>
<b>Capitolo2 – Radio Cognitiva.....</b>	<b>3</b>
2.1 Coral.....	5
2.2 White, Gray e Black holes.....	6
<b>Capitolo3 – Spectrum Sensing.....</b>	<b>9</b>
3.1 Definizione.....	9
3.2 Metodi di Spectrum Sensing per CR.....	10
3.3 Cooperative Spectrum Sensing.....	11
3.3.1 Indipendence Test.....	13
3.3.2 Sphericity test.....	13
3.3.3 Arithmetic Geometric Mean Detection.....	14
<b>Capitolo4 – USRP2.....</b>	<b>17</b>
4.1 Introduzione.....	17
4.2 Caratteristiche USRP2.....	18
4.3 Strumenti Utilizzati.....	19
<b>Capitolo5 – Implementazione Algoritmo di Detection..</b>	<b>23</b>
5.1 Introduzione a Simulink.....	23
5.2 AGM Simulato.....	24
5.3 Algoritmo su segnali reali.....	32
5.4 Misure Sperimentali.....	36
<b>Capitolo6 – Conclusioni.....</b>	<b>43</b>
<b>Bibliografia.....</b>	<b>45</b>

## Acronimi

<b>AWGN</b>	Additive White Gaussian Noise Rumore additivo Gaussiano bianco che rappresenta il rumore termico
<b>CR</b>	Cognitive Radio Nuovo approccio all'utilizzo della banda mediante metodi innovativi per la trasmissione e la ricezione di dati.
<b>DVB-T</b>	Digital Video Broadcasting-Terrestrial È lo standard utilizzato per la trasmissione del segnale televisivo digitale terrestre.
<b>FCC</b>	Federal Communication Commission È un'agenzia governativa degli Stati Uniti che monitora l'uso dello spettro radio e delle telecomunicazioni degli USA.
<b>GNL</b>	Generalized Likelihood Ratio Rapporto di verosimiglianza generalizzato
<b>GSM</b>	Global System for Mobile Communications È lo standard di telefonia mobile cellulare.
<b>IEEE</b>	Institute of Electrical and Electronics Engineers Associazione internazionale che si pone come obiettivo quello di cercare nuove applicazioni, teorie, standard, nei campi dell'elettronica, dell'informatica, della scienza biomedica e dell'elettrotecnica.
<b>PU</b>	Primary User Utente che ha la licenza di trasmettere su una determinata banda.
<b>SDR</b>	Software Defined Radio Piattaforma radio adattabile tramite software.
<b>SNR</b>	Signal to Noise Ratio Rapporto fra la potenza di segnale e quella di rumore.
<b>SU</b>	Secondary User Utente che utilizza una banda attraverso la cognitive radio e spectrum sensing senza interferire con la trasmissione del primary user.

# Capitolo 1

## Introduzione

Al giorno d'oggi parole come Wi-Fi, GSM o dello stesso "Digitale Terrestre" (DVB-T), sono diventate quasi di uso comune, per via del fatto che ormai la maggior parte della popolazione possiede un cellulare, o un accesso ad internet o una TV. Ma come fare a gestire un insieme di trasmissioni così complesso? Come si può mantenere un buon servizio a tutti gli utenti con una banda di frequenze che è limitata?

La cognitive radio, ormai da decenni, si pone come obiettivo quello di risolvere questi problemi, attuando un approccio ragionato ed intelligente alla risoluzione di questi ultimi.

In questo lavoro di tesi si propone un approccio riguardante l'implementazione di algoritmi di detection i quali prendono come fondamento e spunto per l'elaborazione, la radio cognitiva e lo spectrum sensing. In particolare l'algoritmo di arithmetic-geometric detection verrà applicato ad un sistema di antenne fin'ora mai sperimentato. Infatti sono state utilizzate due USRP2 (piattaforme interfacciabili al computer utili per la trasmissione e la ricezione radio) in parallelo e solo in ricezione (quindi un sistema multi-antenna in ricezione), al fine di migliorare il sensing e giungere dei risultati migliori rispetto al caso di una singola antenna in ricezione. Per gestire il tutto si è utilizzato il programma Simulink in ambiente Matlab, che è un software per l'analisi e il calcolo, il quale utilizza un'interfaccia grafica al fine di semplificare la trattazione e ridurre al minimo la scrittura diretta di codice.

Questa tesi è strutturata in sei capitoli di cui il primo è questa stessa introduzione:

Capitolo Secondo: viene esposto il concetto base di questo documento, ossia la radio cognitiva, definendone le caratteristiche e un esempio di applicazione.

Capitolo Terzo: vengono analizzati altri argomenti fondamentali per la trattazione finale: l'analisi dello spettro, l'individuazione di eventuali parti di esso in cui poter trasmettere e l'introduzione degli algoritmi di detection con particolare attenzione all'AGM.

## Capitolo 1 – Introduzione

- Capitolo Quarto: viene definita del dettaglio la strumentazione utilizzata in laboratorio con particolare attenzione alle funzionalità delle periferiche USRP2.
- Capitolo Quinto: viene proposta una breve introduzione all'interfaccia grafica utilizzata sul PC per interfacciare le USRP2 (Simulink); inoltre è riportata l'implementazione dell'algoritmo di detection sia nel caso simulato che nel caso reale.
- Capitolo Sesto: vengono riportate le conclusioni della tesi.



## CAPITOLO 2

### Radio Cognitiva

La radio cognitiva è il presente e il futuro della comunicazione senza fili e rappresenta un approccio intelligente all'utilizzo dello spettro radio; infatti il problema fondamentale riguarda lo sfruttamento di questa preziosa risorsa naturale la quale, a causa dello sviluppo tecnologico e della maggior richiesta di servizi via wireless da parte degli utenti, ha sempre più bisogno di regolamentazioni e metodi alternativi di utilizzo. La *cognitive radio* (CR) si pone come obiettivo quello di utilizzare la banda in un modo intelligente, ossia sfruttando le frequenze in base al non utilizzo da parte di altri utenti delle stesse. Il termine “*cognitive*” descrive (come detto nel documento [1]) “*tre punti di vista computazionali della conoscenza:*

- 1) *Gli stati mentali* (quindi le invenzioni, le scoperte, ecc.) *ed i processi che intervengono tra gli stimoli e le risposte di input/output.*
- 2) *Gli stati mentali e processi sono descritti da algoritmi.*
- 3) *Gli stati ed i processi mentali si prestano ad indagini scientifiche”.*

Analizzando questa definizione e il concetto stesso di radio cognitive, si può descrivere la CR mediante sei aggettivi fondamentali: consapevolezza, intelligenza, apprendimento, capacità di adattamento a diverse situazioni, affidabilità ed efficienza.

Per un'analisi più dettagliata del lavoro svolto dalla CR, si definiscono:

- Utente primario (PU = primary user): colui che ha diritto e priorità di utilizzo della banda
- Utente secondario (SU = secondary user): colui che può utilizzare una banda se e solo se non crea interferenza dannosa al primary.

Infatti come esposto in [2] : “*Numerose osservazioni indicano che molte porzioni dello spettro della banda radio, non sono utilizzate per significativi periodi di tempo, e che l'utilizzo di questi “white spaces” (sia temporali che geografici) nello spettro possono essere incrementati significativamente*”, è da questo presupposto che (tramite i concetti proposti dalla CR) si può trovare la possibilità, da parte di un SU, di trasmettere in una certa banda.

I “white spaces” sono dei “buchi” (poiché vengono definiti anche “spectrum holes”) ossia delle parti di spettro le quali non vengono u-

tilizzate dall'utente primario in quel preciso istante e in quella posizione geografica nella quale viene fatta la rilevazione della banda.

In Figura 2.1 (documento [3]) viene mostrato come, mediante quattro differenti *sniffer* (analizzatori di spettro) posti in posizioni diverse nello spazio, sia possibile trovare degli spectrum hole; nell'esperimento svolto da CRC (società canadese che utilizza la CR per sviluppare prodotti), si utilizzano frequenze Wi-Fi (quindi si cerca di implementare la CR in sistemi comuni e noti da tutti gli utenti). Si nota che sullo Sniffer-2 è presente uno spectrum hole e quindi vi potrebbe essere l'eventualità di trasmettere in quella determinata banda.

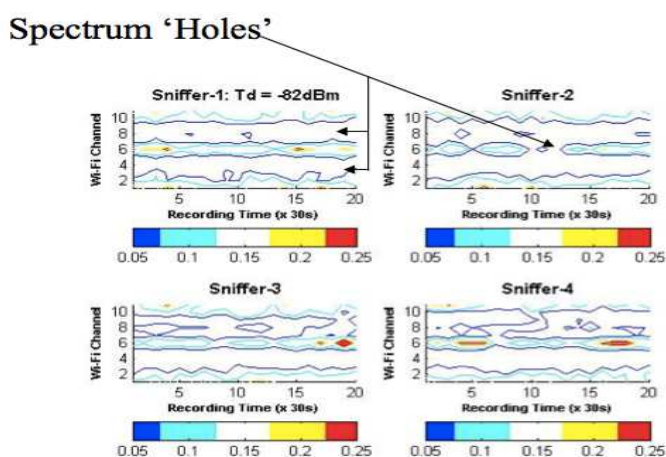


Figura 2.1: Analisi di spectrum holes mediante quattro differenti sniffer

L'importanza della radio cognitiva è quindi fondamentale perché si basa direttamente sulle esigenze dell'utente e formula una soluzione ai problemi di trasmissione su bande sempre meno disponibili, a causa dell'incremento di utilizzo delle frequenze. Infatti Joseph Mitola definisce radio cognitivi quei sistemi che riescono [4]:

- a scovare le necessità dell'utente
- a provvedere ai metodi, servizi e algoritmi che possano soddisfare tali necessità.

Per cui il concetto iniziale su cui si basano le telecomunicazioni, ossia di assegnare frequenze fisse per un certo tipo di servizi agli utenti, viene parzialmente stravolto dal fatto che, con la CR, un utente secondario può utilizzare una determinata banda per uso proprio a patto che non interferisca con il primario. Questo fatto è molto importante poiché permetterebbe di ridurre notevolmente i costi dato che, in teoria, non ci sarebbe più la necessità di comprare una specifica banda da parte di molti PU, ma basterebbe istituire regolamentazioni e dispositivi secondari di tipo CR. I due più importanti limiti da superare per rendere reali le ipotesi descritte qui sopra sui sistemi cognitivi, sono:

- uno di natura scientifica: la necessità di analizzare l'intero spettro al fine di individuare i corretti White Spaces su cui trasmettere;
- il secondo è di natura burocratica: la necessità di descrivere regolamentazioni, leggi, limiti che possano permettere all'utente secondario di trasmettere seppur esigendo una priorità certa di trasmissione all'utente primario.

### 2.1 Coral

Un esempio molto importante di applicazioni pratiche della radio cognitiva, è stato fatto dal CRC [5] (Communication Research Centre in Canada) che ha creato un sistema basato su protocolli Wi-Fi (802.11) e radio cognitiva chiamato Coral. Questo dispositivo integra alle funzioni di un comune router wireless con quelle relative alla radio cognitiva; cerca quindi di ottimizzare al massimo un'applicazione che ormai è utilizzata da tutti gli utenti. Coral è gestito tramite un software diviso in tre parti:

- Network Management System (NMS): gestisce direttamente la rete svolgendo le funzioni di un comune router wireless;
- Cognitive Engine (CE): parte fondamentale che analizza lo spettro nei dintorni del dispositivo ed elabora i dati;
- Database (REAM): insieme di dati che sono frutto del rilevamento derivato dalla CR.

Come funziona? Il sistema ottiene periodicamente informazioni dalla rete e le immagazzina nel database (REAM), il sistema di gestione (NMS) informa il CE che sono presenti nuovi dati relativi alle rilevazioni, a questo punto vengono calcolati dal CE nuovi parametri al fine di ottimizzare la rete wireless.

Per cui questo strumento è molto importante poiché riesce a coniugare aspetti comuni a tutti gli utenti (router wireless) con l'innovazione e la sperimentazione relativa alla cognitive radio, all'unico scopo di ottimizzare sempre più la trasmissione e la ricezione di dati.

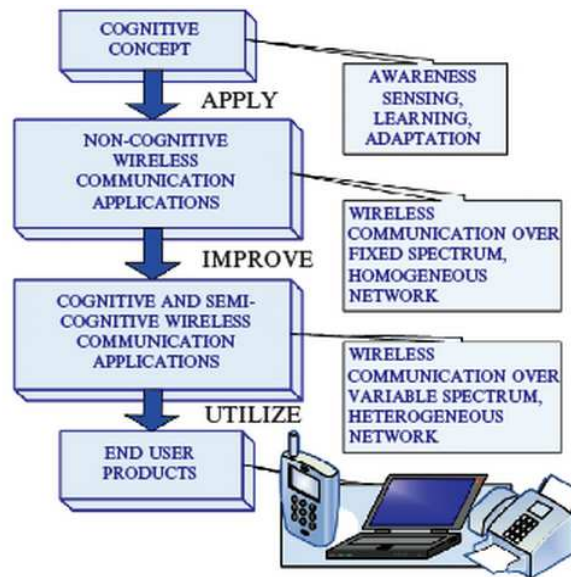


Figura 2.2: Applicazione, miglioramento e utilizzo di una rete wireless con metodi di radio cognitiva [6]

## 2.2 White, Gray e Black holes

Gli spectrum hole sono classificati in base all'occupazione di banda o meno:

- **Black Spaces:** sono bande di frequenze completamente occupate da segnali a media o alta potenza;
- **Gray Spaces:** sono bande parzialmente occupate da segnali a bassa potenza;
- **White Spaces:** si hanno quando il canale è completamente libero da interferenze a meno di quelle relative al rumore termico.

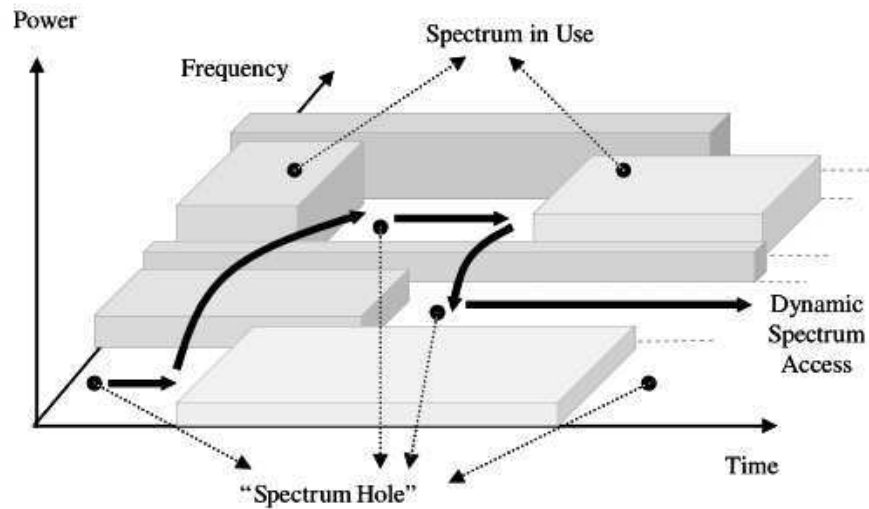


Figura 2.3: Rappresentazione grafica di un “hole” in cui vi è la possibilità di trasmettere utilizzando la tecnica dell’accesso dinamico alla banda [7].

Queste tre definizioni descrivono in modo abbastanza preciso tutti gli eventuali casi che un utente secondario può incontrare nell’analisi di una banda di frequenze.

Giunti a questo punto ci si può chiedere: “come è possibile passare dalla radio cognitiva alla vera e propria ricerca degli spazi di banda atti alla trasmissione di utenti secondari?”. Per fare ciò si utilizzeranno le tecniche di spectrum sensing che implementano algoritmi specifici per queste funzionalità (che verranno descritti minuziosamente nel prossimo capitolo).



## Capitolo3

### Spectrum Sensing

Come è noto, l'evoluzione dei sistemi radio ha portato all'incremento dell'uso dello spettro. Purtroppo la carenza di banda è un problema rilevante nelle telecomunicazioni dei giorni nostri. D'altro canto l'accesso dinamico allo spettro (dynamic spectrum access, DSA) si pone come obiettivo quello di soddisfare questa richiesta di frequenze utilizzando tecnologie, come la cognitive radio, fondamentali al fine di non sprecare la banda.

Lo **spectrum sensing** è una parte fondamentale di questo ragionamento poiché si occupa direttamente della gestione dell'analisi di spettro, per ottenere la miglior realizzazione alla specifica applicazione da implementare.

#### 3.1 Definizione

*“Lo spectrum sensing è un'essenziale funzionalità che abilita le radio cognitive ad individuare gli spectrum holes e in modo opportunistico usa le bande di frequenza sotto-utilizzate senza causare interferenza dannosa alle reti primarie” [8].*

Lo spectrum sensing è quindi la capacità di trovare un range di frequenze nello spettro in cui vi sia la possibilità di trasmettere senza interferire con l'utente primario. Esistono altre tecniche per fare ciò (**Database e Broadcasting**), ma esse comprendono azioni di gestione ed invio di informazioni direttamente dall'utente primario verso gli altri utenti (avendo anche un ulteriore problema di dedicare un canale in cui il PU invia informazioni relative alla banda occupata). Lo spectrum sensing pone il problema in maniera opposta e più coerente con le tecniche di CR. Esso agisce in modo tale che l'utente secondario analizza da sé lo spettro e trova i white spaces su cui trasmettere, senza che il PU tenga in considerazione la presenza di eventuali utenti secondari. Però questa funzionalità deve tenere conto di alcuni fattori fondamentali per far sì che la trasmissione avvenga in modo corretto. Infatti, per effettuare il sensing dello spettro, si possono utilizzare due architetture [9]:

- Single-radio: utilizza un canale nel quale viene effettuato il sensing solo periodicamente per un determinato intervallo di tempo in cui non si può trasmettere;
- Dual-radio: utilizza un canale per il sensing e un canale per la trasmissione e la ricezione dei dati.

I problemi rilevati da entrambe le ipotesi non sono da sottovalutare:

- per la prima si ha che, durante la trasmissione, non viene fatta alcuna analisi dello spettro (potendo quindi avere l'eventualità di una trasmissione di dati nel momento in cui un Primary User richiede la suddetta banda).
- per la seconda si ha a che fare con un'elevata richiesta di potenza (poiché si necessita delle due funzioni di sensing e trasmissione in contemporanea), un aumento di costi dell'hardware ed un aumento della velocità di trasmissione (vedi figura 3.1).

	Single-Radio	Double-Radio
<b>Advantages</b>	<ul style="list-style-type: none"> <li>- Simplicity</li> <li>- Lower cost</li> </ul>	<ul style="list-style-type: none"> <li>- Higher spectrum efficiency</li> <li>- Better sensing accuracy</li> </ul>
<b>Disadvantages</b>	<ul style="list-style-type: none"> <li>- Lower spectrum efficiency</li> <li>- Poor sensing accuracy</li> </ul>	<ul style="list-style-type: none"> <li>- Higher cost</li> <li>- Higher power consumption</li> <li>- Higher complexity</li> </ul>

Tabella 1: Pro e contro di single e double radio [9]

Bisogna quindi trovare un compromesso ottimale fra le due soluzioni che permetta di raggiungere lo scopo prefissato dalla generica applicazione che si vuole implementare, al fine di avere il miglior risultato possibile.

### 3.2 Metodi di Spectrum Sensing per CR

Di seguito vengono brevemente descritti i principali approcci all'utilizzo di popolari algoritmi che implementano in sé il concetto di spectrum sensing e di CR, per la rilevazione di segnali trasmessi [9]:

- **Energy Detector (ED):** è il modo più semplice per realizzare lo spectrum sensing per via dei bassi costi e della bassa complessità di implementazione; esso analizza direttamente l'energia del segnale ricevuto senza conoscerlo, questo però



potrebbe creare dei problemi per segnali a bassa potenza poiché gli algoritmi di ED sono sensibili al rumore;

- **Waveform-Based Sensing:** metodo utilizzato principalmente per la sincronizzazione; l'implementazione di questo algoritmo non è molto semplice e comporta innanzitutto la conoscenza dei campioni o di un preambolo (ossia una sequenza di campioni conosciuta e trasmessa ad un certo istante della trasmissione ciclicamente). Questo metodo può quindi essere applicato facendo la correlazione tra il segnale ricevuto e la sua copia conosciuta in precedenza. Oggettivo limite di questo algoritmo è quello della verosimiglianza della trasmissione e ricezione del preambolo;
- **Eigenvalue-Based Detection:** si basa sull'analisi della matrice di covarianza del segnale ricevuto e in particolare sugli autovalori che la compongono. Vedi paragrafo 3.3.3 Arithmetic-Geometric Mean Detection.
- **Cyclostationarity-Based Sensing:** è un metodo di sensing utilizzato grazie alla periodicità del segnale o di alcune grandezze statistiche quali la media, l'autocorrelazione, ecc..

### 3.3 Cooperative Spectrum Sensing

Spesso l'attenuazione, lo shadowing ed il rumore dei singoli ricevitori, possono compromettere il sensing e determinare un'acquisizione di dati non consona a ciò che si ha nella realtà sullo spettro. Un modo per attenuare il più possibile questi inconvenienti, è quello di realizzare algoritmi di sensing cooperativo i quali permettono lo scambio di informazioni relative all'analisi dello spettro, ad un insieme di utenti secondari.

Nella teoria molte reti di spectrum sensing cooperativo hanno nodi per i quali viene considerata la stessa potenza di rumore, quindi la teoria è difficilmente applicabile alla realtà, poiché il rumore varia da nodo a nodo, sia a causa della temperatura, che delle caratteristiche differenti che distinguono ogni singolo ricevitore. Questo problema verrà affrontato direttamente nella pratica (come si vedrà nel capitolo 5) introducendo due fondamentali algoritmi (**independence test** e **AGM**) che svolgono l'analisi di frequenza seppur avendo ricevitori (quindi un sistema ad **antenne multiple**) non calibrati (con differenti potenze di rumore) e un algoritmo (**sphericity test**) il quale invece necessita di ricevitori propriamente configurati e calibrati.

Prima di descrivere i tre algoritmi, è necessario dare un'introduzione alla teoria della decisione, problema fondamentale delle telecomunicazioni radio.

### Capitolo 3 – Spectrum Sensing

Un ricevitore ha il difficile compito di interpretare il segnale ricevuto e analizzarlo in modo da carpire da esso il segnale utile, cercando di eliminare la componente spuria derivante dal rumore termico e dalle interferenze nella trasmissione. Il problema rilevante si ha quando la potenza di segnale è paragonabile a quella di rumore; in questo caso è necessario trovare un criterio ottimale, adottabile in tutte le situazioni, che permetta di compiere una decisione corretta (qui entra in gioco la teoria della decisione).

Per descrivere questo metodo sperimentale, si farà riferimento all'acquisizione di un determinato segnale da parte di un ricevitore, il quale potrà distinguere fra due ipotesi:

- $H_0$  : caso in cui ho solo rumore termico  $n(t) \sim N(0, \sigma^2)$
- $H_1$  : caso in cui oltre al rumore ho anche il segnale utile

$$\text{Ipotesi } H_0 : \quad \mathbf{y}(t) = \mathbf{n}(t) \quad (3.3.1)$$

$$\text{Ipotesi } H_1 : \quad \mathbf{y}(t) = \mathbf{s}(t) + \mathbf{n}(t) \quad (3.3.2)$$

Per cui esistono quattro diverse possibilità a seconda della decisione che prenderà il ricevitore sul segnale ricevuto, riportate in tabella 2;

Tabella 2: Descrizione delle decisioni possibili in base ai segnali ricevuti

Probabilità delle decisioni condizionate al segnale trasmesso	Descrizione
$\text{Prob} \{ \hat{H}_0   H_0 \}$	Decisione corretta: <i>correct rejection</i> ( $P_R$ ), non è stato trasmesso nessun segnale
$\text{Prob} \{ \hat{H}_1   H_0 \}$	Decisione errata: <i>false allarme</i> ( $P_{FA}$ ), non è stato trasmesso alcun segnale
$\text{Prob} \{ \hat{H}_0   H_1 \}$	Decisione errata: <i>missed detection</i> ( $P_{MD}$ ), il segnale è stato trasmesso ma non visto
$\text{Prob} \{ \hat{H}_1   H_1 \}$	Decisione corretta: <i>correct detection</i> ( $P_D$ ), il segnale è stato trasmesso e visto dal RX

Per arrivare ad una decisione ( $\Lambda$ ), bisogna introdurre il Lemma fondamentale di Neyman-Pearson, secondo il quale:

$$T = \frac{|\hat{\Sigma}_1|}{|\hat{\Sigma}_0|} \underset{H_1}{\overset{H_0}{>}} \xi \quad (3.3.3)$$

ossia, scegliendo in modo analitico (e questo sarà uno dei problemi dell'indipendenza e dello sphericity test) una soglia  $\xi$ , si può giungere alla decisione e calcolare le probabilità di essa in base alla ta-

bella qui sopra; il rapporto  $\frac{|\hat{\Sigma}_1|}{|\hat{\Sigma}_0|}$  viene detto *generalized likelihood*

*ratio* o *GNL* (rapporto di verosimiglianza generalizzato).  $\Sigma_0$  e  $\Sigma_1$  sono le rappresentazioni della matrice di covarianza ( $S$ ) nei casi  $H_0$  (in questo caso  $\Sigma_0$  è diagonale) e  $H_1$ .

### 3.3.1 Independence Test

“L'Independence Test è in realtà un rapporto di verosimiglianza generalizzato dove gli utenti secondari non sono calibrati” [10], da questa definizione si può dedurre che questo algoritmo può essere implementato nel caso in cui le potenze di rumore relative alle antenne in ricezione sono differenti, e quindi non hanno la stessa varianza; infatti per giungere ad una decisione si può esprimere il rapporto di verosimiglianza come [10]:

$$T^{(\text{ind})} = \frac{|S|}{\prod_{k=1}^{N_R} s_{k,k}} \begin{matrix} H_0 \\ < \xi \\ > \xi \\ H_1 \end{matrix} \quad (3.3.4)$$

dove  $s_{k,k}$  sono gli elementi della diagonale principale della matrice di covarianza del segnale e  $N_R$  il numero di ricevitori.

Nei casi di spectrum sensing cooperativo, questo algoritmo risulta più robusto (rispetto allo sphericity) alla non uniformità della potenza di rumore, nonostante ciò il prossimo algoritmo rimane una valida alternativa all'indipendenza test.

### 3.3.2 Sphericity Test

Come detto in precedenza, questa implementazione necessita di ricevitori fra di loro calibrati. Per fare ciò tutti gli utenti secondari devono avere lo stesso livello di rumore. Anche in questo caso, per prendere una decisione, bisognerà impostare una soglia che permetta di esprimere il GLR:

$$T^{(\text{ind})} = \frac{|S|}{(\text{tr}\{S\}/N_R)^{N_R}} \begin{matrix} H_0 \\ < \xi \\ > \xi \\ H_1 \end{matrix} \quad (3.3.5)$$

dove  $\text{tr}\{\mathbf{S}\}$  rappresenta la traccia della matrice  $\mathbf{S}$ . In questo caso l'algoritmo riesce a trovare delle probabilità di detection o di falso allarme (a seconda che ci sia il segnale o meno) accettabili e abbastanza vicine al caso di paragrafo 3.3.1, resta sempre il rilevante problema di avere tutte le antenne calibrate (cosa irrealizzabile nella realtà di laboratorio).

### 3.3.3 Arithmetic – Geometric Mean Detection

Questa implementazione rappresenta un nuovo approccio al problema dell'acquisizione di segnali da parte di più ricevitori (sistemi di spectrum sensing multi-antenna), poiché non necessita (come invece lo era per gli ultimi due algoritmi) di impostare una soglia. Nonostante ciò, per introdurre l'algoritmo stesso, l'*arithmetic-geometric mean* sfrutta gli stessi presupposti dell'indipendenza e dello sphericity, ossia applicando la *generalized likelihood ratio* giunge ad una decisione grazie ad una certa soglia  $\xi$ :

$$T^{(\text{AGM})} = \frac{\frac{1}{M} \sum_m \lambda_m}{\left(\prod_m \lambda_m\right)^{1/M}} \begin{array}{l} H_0 \\ < \xi \\ > \xi \\ H_1 \end{array} \quad (3.3.6)$$

in questo caso si denotano con  $M$  il numero di campioni, con  $\lambda_m$  gli autovalori della matrice di covarianza. Il rapporto visto nella formula (2.3.6) da il nome all'algoritmo stesso infatti è un rapporto tra la media aritmetica e la media geometrica degli autovalori della matrice di covarianza del segnale. La (2.3.6) ha ancora il problema della soglia il quale verrà risolto con l'implementazione dell'AGM mediante la stima del numero di segnali realmente presenti nella selezione ricevuta dall'antenna. Per descrivere ciò si definisce un numero di modelli  $\hat{k}$ , i quali sono l'insieme dei segnali il più possibile simili a quello ricevuto [11] e [12]:

$$\hat{k} = \arg \min_k \{-\log(\mathbf{Y}; \hat{\boldsymbol{\theta}}^{(k)}) + L(\nu(k), N_s)\} \quad (3.3.7)$$

$$k = 0, \dots, k_{\text{MAX}}$$

Nella quale  $\mathbf{Y}$  è la matrice composta dai vettori dal segnale che arriva al ricevitore (quindi rumore sommato a segnale utile);  $k_{\text{MAX}}$  è il massimo numero di modelli;  $\hat{\boldsymbol{\theta}}^{(k)}$  è la stima della massima verosimi-

glianza al vettore  $\theta^{(k)}$ ;  $N_s$  sono in numero di istanti di osservazione del segnale ricevuto;  $\nu(k)$  è un numero di parametri liberi.

$$L(\nu(k), N_s) = \frac{\nu}{2} \log(N_s) \quad (3.3.8)$$

viene definita come funzione di penalità ed è fondamentale per la definizione generale per la stima del numero di segnali ottenuta come:

$$\hat{N}_T = \arg \min_{k \in \{0, \dots, k_{\max}\}} \left\{ \log \left[ \left( \frac{1}{N_R - k} \sum_{i=k+1}^{N_R} l_i \right)^{(N_R - k) N_s} \right] + L(k(2N_R - k), N_s) \right\} \quad (3.3.9)$$

Nella (2.3.9)  $l_i$  sono gli autovalori stimati della matrice di covarianza (quindi saranno uguali ai  $\lambda_m$  stimati visti nella formula 2.3.6).

L'obbiettivo dell'implementazione di questo algoritmo, è quello di trovare la molteplicità dell'autovalore minimo che si adatta al meglio alla matrice di covarianza stimata. Inoltre, per arrivare ad una formula dell'AGM che non comporti la definizione di una soglia basterà semplificare il tutto imponendo  $k_{\max} = 1$  e di fatto trovando due uniche ipotesi possibili ossia Ipotesi  $H_0$  e  $H_1$  viste nelle formule 2.3.1 e 2.3.2. Nel capitolo 5 verranno descritte le seguenti formule relative alla stima del numero di segnali:

$$\hat{N}_T(0) = \log \left[ \left( \frac{1}{N_R} \sum_{i=1}^{N_R} l_i \right)^{N_R N_s} \right] \quad (3.3.10)$$

$$\hat{N}_T(1) = \log \left[ \left( \frac{1}{N_R - 1} \sum_{i=2}^{N_R} l_i \right)^{(N_R - 1) N_s} \right] + \frac{2N_R - 1}{2} \log N_s \quad (3.3.11)$$

### Capitolo 3 – Spectrum Sensing

Il rapporto tra le formule 2.3.10 e 2.3.11 andrà a costituire il test:

$$\frac{\hat{N}_T(0)}{\hat{N}_T(1)} \underset{H_1}{\overset{H_0}{<}} 1 \quad (3.3.12)$$

# Capitolo 4

## USRP2

### 4.1 Introduzione

Prima di descrivere che cosa è un USRP2 bisogna definire un concetto fondamentale nella cognitive radio sviluppatosi soprattutto grazie al continuo miglioramento della tecnologia, ossia: la **software defined radio** (SDR).

Gli apparati di comunicazione wireless (più in generale tutti gli apparati relativi alle telecomunicazioni e alla trasmissione di informazioni) sono caratterizzati da tre aspetti principali:

- Il *signaling*: scambio di informazioni riguardanti la creazione e il controllo del circuito di telecomunicazione e la gestione della rete;
- L'hardware fisico: insieme dei componenti che costituiscono l'apparato in sé;
- Le funzionalità stesse dell'apparato.

Nel tempo lo sviluppo scientifico ha permesso di creare strumenti di comunicazione sempre più sofisticati e performanti grazie alla significativa evoluzione della tecnologia hardware. Nonostante ciò è rimasto il problema dell'adattabilità; infatti, fino a qualche anno fa, gli apparati venivano costruiti appositamente per una determinata applicazione, generando un netto aumento dei costi relativi ad una eventuale riconfigurazione. Il problema principale è quello di creare *hardware* che si possa adattare il più possibile alle molteplici realizzazioni, o esperimenti, o studi, o calcoli, ecc.. La *software defined radio* crea hardware *virtuali* (che completano quelli fisici su scheda) che servono ad implementare una specifica applicazione, ottenendo così un notevole vantaggio in termini di costi (poiché conviene modificare "virtualmente" i componenti tramite software che andarli a sostituire "fisicamente" in un circuito). La SDR viene quindi definita dall'FCC come "*una radio che include un trasmettitore, nella quale i parametri operativi del range di frequenze, del tipo di modulazione, o della massima e minima potenza in uscita possono essere alterati tramite modifiche al software, senza effettuare nessuna modifica alla componente hardware*" [14].

Partendo dalle basi di questa definizione, Matt Ettus, insieme alla sua azienda (Ettus Research LLC), ha creato una periferica chiamata

**USRP** (Universal Software Radio Peripheral) che integra in sé il concetto di SDR. Successivamente nel 2008 viene rilasciata la seconda versione (per risolvere alcuni problemi della prima periferica) che è quella che andremo ad analizzare nel dettaglio, ovvero la **USRP2**.



Figura 4.1: Pannello frontale della USRP2

## 4.2 Caratteristiche USRP2

Nell'elaborazione di questa tesi di laurea sono state utilizzate due distinte USRP2 le quali entrambe sono composte da una motherboard nella quale sono presenti l'FPGA e vari ADC e DAC come in figura:

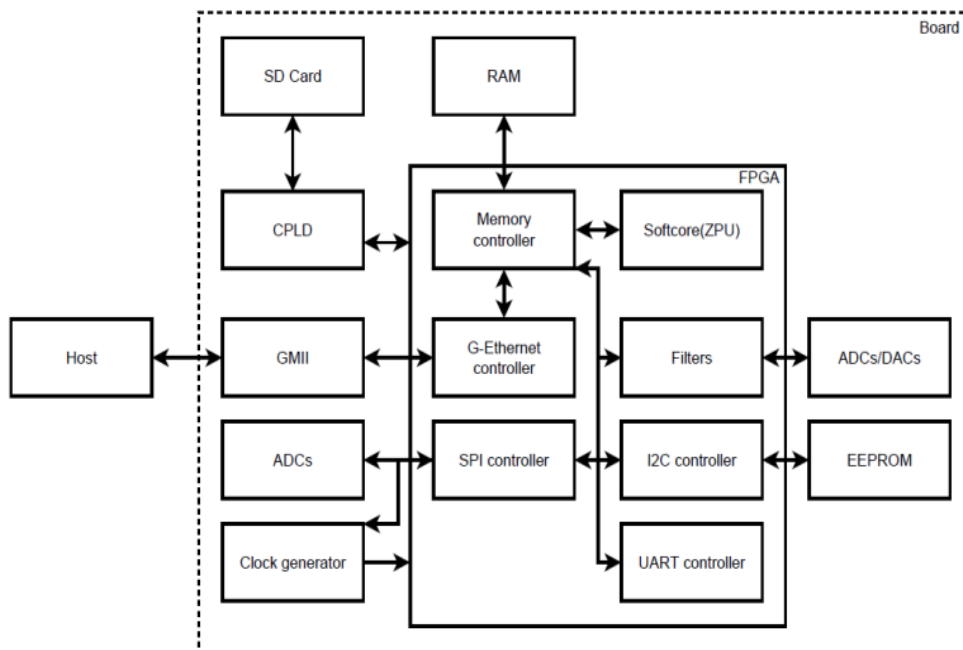


Figura 4.2: architettura USRP2



Il principale compito della motherboard è il campionamento, infatti l'USRP2 riceve il segnale dall'ingresso (RF1 vedi figura 4.1), gli ADC ne fanno una digitalizzazione (quindi il segnale campionato ad una frequenza di 100Ms/s che è quella di campionamento degli ADC), poi viene direttamente elaborato in modo da avere una Digital Down Conversion. In sostanza il segnale di ingresso (che sarà un segnale continuo a una certa frequenza, ad esempio, per gli esperimenti del capitolo 5 viene utilizzata una sinusoide a 870 MHz) viene campionato, portato in banda base e poi filtrato, in modo da poter essere letto correttamente dal software il quale potrà elaborare i dati ricevuti.

Le daughterboard invece hanno il compito di distinguere le USRP2 in base alle potenze, al guadagno, al rumore e alle frequenze di utilizzo; infatti, nel caso visto in laboratorio, verranno utilizzate in entrambe le USRP2, le daughterboard WBX che hanno una banda di lavoro che va dai 50MHz ai 2.2GHz (quindi tra le Very e le Ultra High Frequency).

Dalla figura 4.2 si possono distinguere i parametri più significativi, dei quali vengono riportati in seguito i parametri principali:

- FPGA Xilin Spartan 3-2000;
- 2 ADC da 14 bit a 100MS/s;
- 2 DAC da 16 bit a 400MS/s;
- 1 Mbyte di SRAM
- 2 slot: 1 RX e 1 TX;
- Connessione USRP2/Host tramite Gbit Ethernet;
- Lettore SD-card;
- Modalità stand-alone;
- Un ingresso per Mimo Cable

### 4.3 Strumenti Utilizzati

Come descritto già in precedenza, l'approccio e i sistemi utilizzati in questa tesi, non erano stati presi in considerazione prima d'ora; infatti, per lo svolgimento delle prove di laboratorio, sono state utilizzate due USRP2 distinte in parallelo per la ricezione. Questo sistema è un sistema multi-antenna che riceve direttamente un segnale generato da uno Sweeper (vedi figura 4.3).

## Capitolo 4 – USRP2



Figura 4.3: HP Sweeper 83751A utilizzato per le misure in laboratorio; si può notare a sinistra come sia impostata la frequenza a 870MHz e a destra la potenza a -84dBm

Lo Sweeper (HP Sweeper 83751A) è un generatore di forme d'onda molto preciso e soprattutto permette di creare segnali modulabili a piacere. Le due principali forme d'onda che può generare sono: sinusoidi e sweep.

Nel caso di misure in laboratorio, per interfacciare le due USRP2 fra loro, sarebbe stato utile utilizzare il collegamento MIMO tramite l'omonimo cavo (fondamentale per sincronizzare in modo molto preciso due USRP2), ma purtroppo per due dispositivi in parallelo, nel modo in cui si vedrà nel prossimo capitolo, non è possibile fare ciò.

Le USRP2 sono collegate via ethernet ad un PC (Pentium® 4 dual 3.40GHz, 1 Gb di RAM) precedentemente assemblato (vedi [15]) nel quale sono presenti due schede di rete, ognuna per le singole USRP2, di modo che l'acquisizione da parte di un unico PC dei dati ricevuti sia fatta in contemporanea sulle due periferiche.

Basta quindi impostare lo Sweeper sulla forma d'onda desiderata (ad esempio sinusoidale), impostare con il tasto "CW" la frequenza centrale del suddetto segnale (nel caso in laboratorio: 870MHz), decidere un valore di potenza del segnale con il tasto "Power Level" (ad esempio -84dBm come in figura 4.3), collegare il tutto con cavi *sma* alle USRP2, collegate a loro volta (con cavi ethernet) al pc, e si ottiene quindi una configurazione finale che è illustrata nella figura 4.4.



Figura 4.4: Si nota nel dettaglio come le due piattaforme siano collegate direttamente tramite cavo ethernet al pc (vedi figura 9)

Per fare in modo che le USRP2 siano riconosciute dal PC (nel quale è installato Ubuntu), per installare i firmware, e per l'installazione di Matlab e Simulink, si rimanda a [7], [11] e [15] nei quali viene descritto passo per passo come realizzare tutto ciò.

## Capitolo 4 – USRP2



Figura 4.5: Particolare delle due schede di rete nelle quali vengono inseriti i cavi ethernet che arrivano dalle USRP2

## Capitolo 5

# Implementazione Algoritmo di Detection

### 5.1 Introduzione a Simulink

L'algoritmo di detection che verrà implementato in questa tesi, sarà l'*arithmetic-geometric mean detection*, il quale verrà implementato in ambiente Matlab, più in particolare, utilizzando Simulink.

Innanzitutto Matlab (Matrix Laboratory) è un software fondamentale per il calcolo, l'elaborazione, lo sviluppo di dati e per l'implementazione di algoritmi. Nelle prove di laboratorio e nei risultati che verranno proposti in questo testo, è stata utilizzata la versione R2011a (che può essere trovata tramite il documento [16] in versione di prova) che è la prima versione di Matlab nella quale sono disponibili dei blocchi, creati dalla Ettus e da MathWorks, al fine di interfacciare le USRP2 con il software ed in particolare con Simulink.

*“Simulink è un ambiente per la simulazione multidominio e per la progettazione basata sulla modellazione per sistemi dinamici e integrati. Mette a disposizione un'interfaccia grafica interattiva e una serie di librerie personalizzabili per sviluppare, simulare, implementare e testare numerose categorie di sistemi variabili nel tempo, comprendenti tematiche quali comunicazioni elettriche, controlli, elaborazione di segnali, video e immagini”*[17]. Per cui è una branca di Matlab molto importante e abbastanza facile da usare per via del fatto che, invece del codice scritto, utilizza un'interfaccia grafica che permette, tramite il susseguirsi di blocchi (raggruppati in librerie) e connessioni, di implementare algoritmi o svolgere calcoli in modo relativamente semplice. Questo programma è direttamente integrato a Matlab e permette anche di passare direttamente dalla scrittura in codice implementando anche le funzioni del programma principale.

Per implementare l'algoritmo AGM si utilizzeranno due approcci: **simulato** (ossia generando dei segnali solo da software quindi da blocchi in libreria di Simulink) e **reale** (ossia prendendo direttamente dei campioni che arrivano dalle USRP2).

## 5.2 AGM Simulato

In figura 5.16 viene presentato l'algoritmo AGM simulato mediante l'interfaccia di Simulink. Per giungere a quel risultato, è stato necessario introdurre blocco per blocco ogni singola funzionalità dell'algoritmo: dal generatore di rumore, al generatore di segnale, alla creazione della matrice di covarianza, alla trasposizione dell'algoritmo vero e proprio in linguaggio Matlab/Simulink.

Per prima cosa è stato necessario impostare alcuni valori (che si possono trovare anche in [7]) per l'esecuzione degli algoritmi (vedi figura 5.1):

```

1  %-----Variabili globali-----
2
3  %-----Cooperative Detection Variables-----
4
5  nS=500;           % Numero di campioni osservati in una realizzazione
6  nP=10000;        % Numero di realizzazioni del segnale ricevuto
7  nR=2;            % Numero di antenne in ricezione
8  nT=1;            % Numero di sorgenti trasmettenti
9  pFad=0.1;        % Probabilità di falso allarme desiderata
10 tS=0.1;          % Passo di campionamento per i segnali simulati
11 SNRdB=-10;       % Signal to Noise Ratio (dB) fissato per la simulazione
12 SNR=10^(SNRdB*0.1); % Signal to Noise Ratio lineare
13 dec=512;         % Decimazione USRP2 RX
14 samp=dec/100e6; % Sample Time per USRP2
15
16 %-----
17 % DON'T CONNECT TX AND RX!!!
18 % Need an attenuator about 40-50 dB.
19 %
20 % WBX Transmit Power 30 to 100 mW (14.77 - 20 dbm)
21 % WBX Receiver:
22 % Primary and second stage Max power limit -10dbm
23 % Suggested max -30dbm direct cable connection.
24 % (typical use -60dbm on-air)
25 %-----
    
```

Figura 5.1: Modello in Matlab nel quale vengono definite le variabili

Del modello in figura 5.1 è necessario ricordare alcuni valori già visti negli algoritmi di indipendente, sphericity e AGM; in particolare:

- nS sono il numero di campioni che viene considerato in ogni singola realizzazione;
- nP sono il numero di prove quindi di realizzazioni del segnale ricevuto;
- nR è il numero di ricettori (quindi nel caso visto in laboratorio si ha a che fare con due USRP2 in ricezione);
- nT è il numero di sorgenti trasmettenti (l'unico Sweeper che manda il segnale alle periferiche);

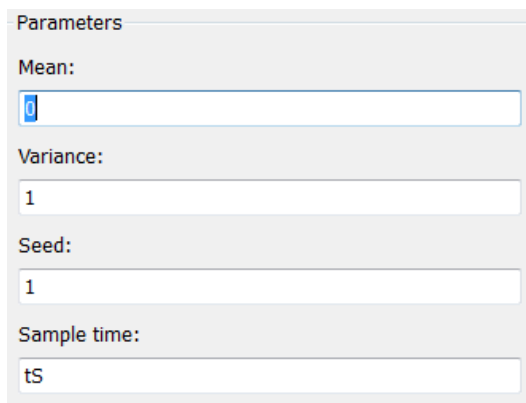
- $tS$  è il tempo di campionamento per i segnali simulati (*Sample time*);
- SNR e SNRdB sono rispettivamente il rapporto segnale rumore in lineare e in dB del segnale simulato.
- Dec e samp sono variabili utili nel caso reale (vedi paragrafo 5.3)

Ora si andranno a descrivere passo per passo i singoli blocchi che rappresentano l'implementazione di questo algoritmo.



Figura 5.2:  
Blocco *Random Number*

Il *Random Number* (fig. 5.1) genera una sequenza di numeri casuali secondo una distribuzione Normale; nel caso visto in fig. 5.16 sono stati inseriti quattro blocchi che generano del rumore e due blocchi nei quali è stato impostato di creare il segnale vero e proprio.



Parameters

Mean:

Variance:

Seed:

Sample time:

Figura 5.3: Parametri impostabili da *Random Number*

Nel caso di rumore i parametri impostati sono: valor medio nullo, varianza unitaria e sample time (ossia tempo di campionamento) impostato a  $tS$ . Questo valore è molto importante ed è stato scelto inizialmente ed impostato in tutti i generatori di numeri casuali, questo perché è il tempo di campionamento che, se uguale permette una perfetta sincronizzazione tra rumore e segnale.

Il *seed* (seme in inglese) è invece un valore che va cambiato in ogni blocco *Random Number* che viene introdotto nell'algoritmo, questo perché il simulatore non crea un vero e proprio numero casuale, ma ha un insieme di numeri che si ripetono ciclicamente con un periodo molto lungo, quindi è necessario scegliere seed diversi per avere delle sequenze di numeri casuali anche fra loro.

Impostare lo stesso *seed*, significa impostare il suddetto blocco dallo stesso punto di partenza, rendendo i generatori di rumore uguali (cosa che renderebbe l'algoritmo simulato lontano dalla realtà). Per impostare il segnale invece, si impone la varianza pari ad  $SNR/2$  nei due blocchi (questo perché andandoli a sommare, si ottiene l'SNR voluto), e il valore medio sempre nullo.



Figura 5.4:  
*Real-Imag to Complex*

Il blocco *Real-Imag to Complex* restituisce un segnale complesso a fronte dei blocchi che generano numeri casuali.

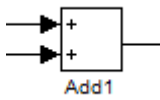


Figura 5.5: Sum

Il blocco *Sum* somma semplicemente i campioni ricevuti dai generatori di rumore simulato e dal segnale simulato, aggiungendo quindi alle informazioni utili un rumore additivo come avviene nella realtà.

Il problema sta nel fatto che ciò che si vuole simulare, è un segnale che ad ogni campione abbia una parte di rumore (come è nella realtà). Nel segnale simulato questo problema è subito risolto, poiché tutti i generatori visti in figura 5.2 sia per il segnale, che per il rumore, hanno lo stesso Sample time e quindi sono perfettamente sincronizzati e replicano ciò che avviene realmente.

La parte descritta fin'ora è la vera e propria parte simulata, nel senso che, questa sezione dell'implementazione dell'algoritmo, va a rappresentare ciò che le USRP2 fanno nella realtà. Infatti esse eseguono un campionamento del segnale e lo riportano al computer ovviamente aggiungendo rumore termico. La restante parte dell'algoritmo sarà uguale a quella che verrà descritta nel capitolo 5.4.

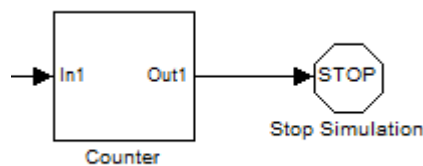


Figura 5.6: Counter e Stop Simulation

Il *Counter* (figura 5.7) conta semplicemente i campioni che gli arrivano in modo da fermare (Stop Simulation fig. 5.6) la simulazione non appena il contatore ha raggiunto il numero di campioni impostato a priori prima dell'inizio della simulazione stessa (per una stima accettabile bisogna considerare almeno 10000 campioni).



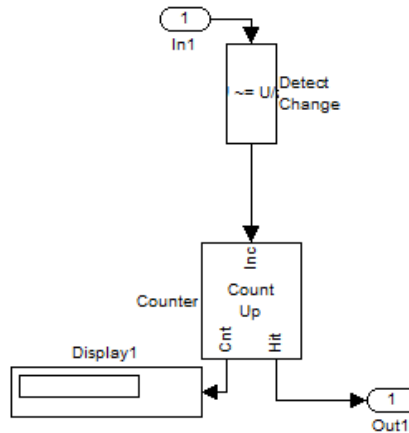


Figura 5.7: Interno del blocco Counter; cliccando su Count up si imposta il numero di campioni da contare prima di dare in uscita il valore Hit che di fatto poi stoppa la simulazione

Per introdurre il blocco relativo alla matrice di covarianza normalizzata (fig. 5.8), bisogna prima definire quest'ultima:

$$\mathbf{S} = \frac{1}{N_s} \mathbf{Y}\mathbf{Y}^H \quad (5.3.1)$$

Dove  $\mathbf{Y}$  è la matrice vista nella formula 3.3.7,  $\mathbf{Y}^H$  è la sua hermitiana e  $N_s$  è il numero di campioni impostato nel modello in figura 5.1. Per cui il blocco della Normalized Sample Covariance Matrix, (vedi figura 5.9) rappresenta al suo interno l'operazione vista per la formula 5.3.1 a meno di un buffer che ha il compito di raccogliere il numero di campioni per realizzazione che si vuole considerare:  $N_s$  (ossia 500, valore sempre impostato nel modello di fig. 5.1).

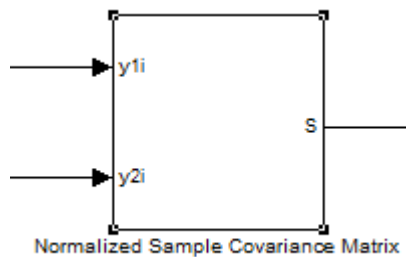


Figura 5.8: Normalized Sample Covariance Matrix

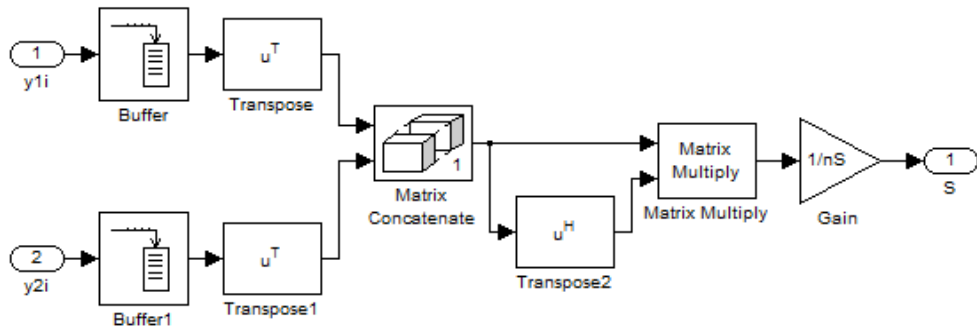


Figura 5.9: Implementazione della formula teorica della matrice di covarianza, mediante i blocchi di Simulink. Si può notare come i vettori  $y1i$  e  $y2i$  siano quelli che vanno a comporre la matrice  $\mathbf{Y}$ , questi vengono bufferizzati, poi trasposti e concatenati in una matrice (che è proprio la matrice  $\mathbf{Y}$ ); giunti a questo punto se ne fa la trasposta e si moltiplicano le due matrici, dividendo tutto per gli  $N_S$  campioni considerati.

Implementando quest'ultima parte, si stima la matrice di covarianza, che, nel caso visto al capitolo 5, sarà una matrice con due righe e due colonne. Ora, grazie agli strumenti di Matlab, è possibile scrivere direttamente l'algoritmo di arithmetic-geometric mean detection che ricaverà i due valori di  $\hat{N}_T(0)$  e  $\hat{N}_T(1)$ , come in figura 5.10.

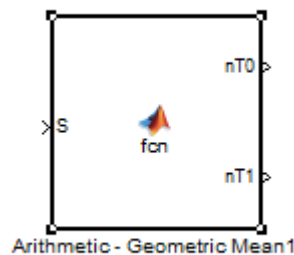


Figura 5.10: Implementazione dell'algoritmo AGM in Simulink, mediante codice Matlab

All'interno del blocco in figura 5.10 si possono infatti distinguere nettamente i parametri visti nella teoria dell'AGM al capitolo 3 (vedi figura 5.11).

```

1 function [nT0, nT1] = fcn(S)
2 %#codegen
3 % definisco le variabili nR e nS
4 nR=2;
5 nS=500;
6
7 % funzione 'eig' crea un vettore colonna con gli autovalori della matrice S
8
9 t=eig(S);
10 t1=t(1,1);
11 t2=t(2,1);
12
13
14
15 % calcolo la stima del numero di segnali presente, nel caso segnale assente
16 nT0=log(((1/nR)*(t1+t2))/(((t1)^(1/nR))*(t2)^(1/nR))))^(nR*nS);
17
18 % calcolo la stima del numero di segnali presente, nel caso segnale
19 % presente
20 nT1=(log(((1/(nR-1))*t2)/(t2^(1/(nR-1))))^(nR-1)*nS))+((log(nS))*((2*nR)-1)/2);
    
```

Figura 5.11: Codice Matlab che rappresenta come trovare i valori  $\hat{N}_T(0)$  e  $\hat{N}_T(1)$



Figura 5.12: Divide



Figura 5.13: Abs

A questo punto basta applicare la formula 3.3.12 utilizzando il blocco *Divide* (figura 5.12), il quale fa il rapporto fra il numero di segnali presenti stimati e farne il modulo (figura 5.13).

Infine il blocco *Probability Calculator* (fig. 5.14) da in uscita la probabilità di detection e quella di falso allarme.

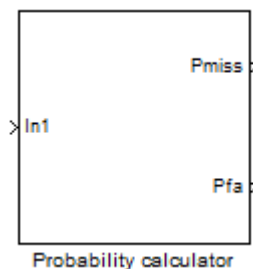


Figura 5.14: Probability calculator

All'interno di questo blocco infatti è presente l'operatore relazionale che è il maggiore (o minore a seconda dell'algoritmo che si vuole implementare) il quale può essere visto nella formula 3.3.12 dell'AGM. Il blocco *Error Rate Calculation*, come definito dalla stessa MathWork, [18]: “calcola il tasso di errore dei dati ricevuti confrontandolo con una versione ritardata dei dati trasmessi. L'uscita del blocco è costituita dal tasso di errore seguito dal numero di errori rilevati e il numero totale di simboli a confronto”.

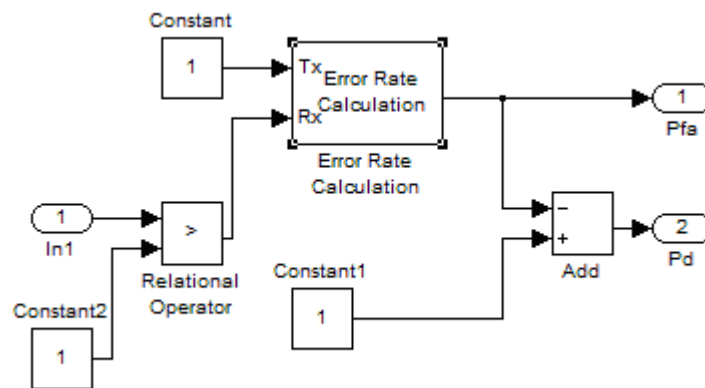


Figura 5.15: interno del blocco in figura 5.14

Lanciando la simulazione in figura 5.16, si può infine trovare la probabilità di aver deciso per la rilevazione del segnale, quando in realtà non c'era ( $P_{fa}$ ) e la probabilità di decidere per la rilevazione di esso quando era stato effettivamente trasmesso dalla sorgente ( $P_d$ ).

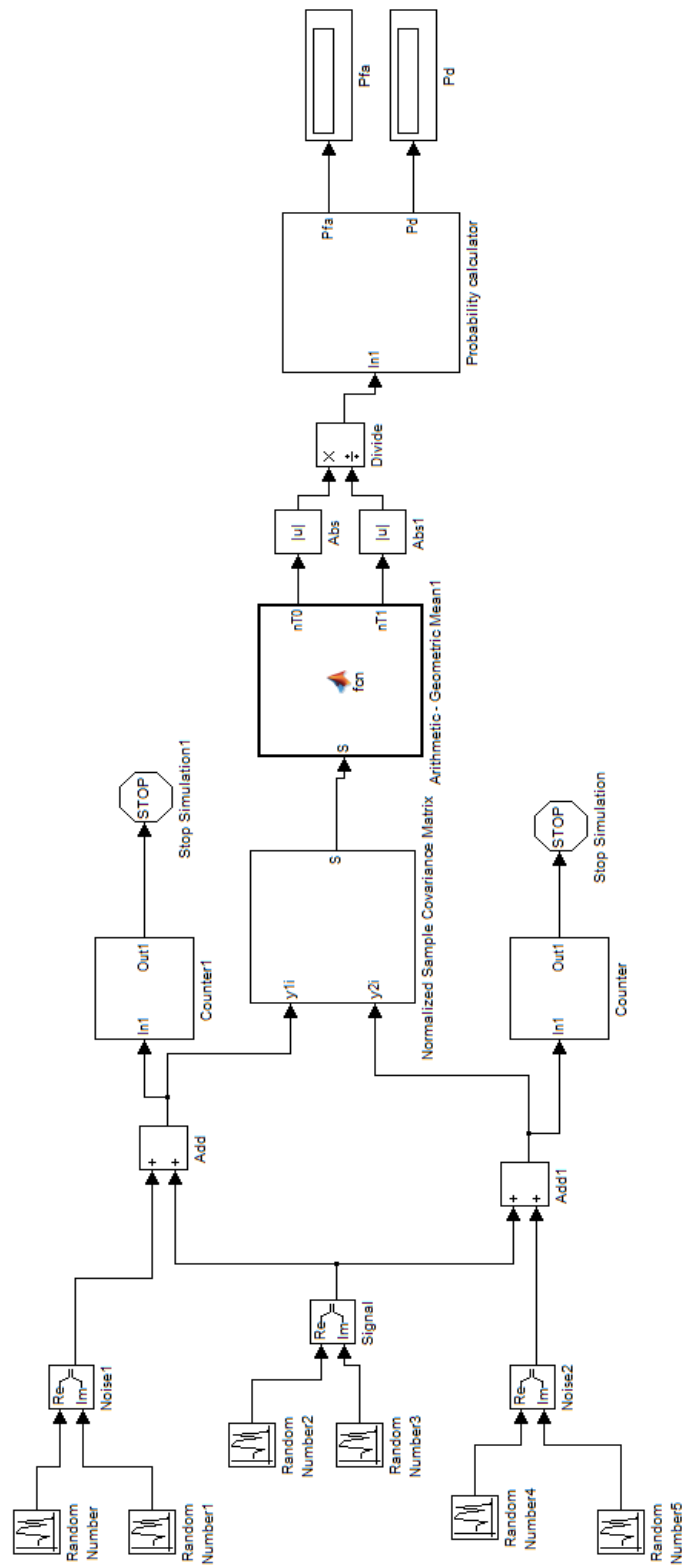


Figura 5.16: Rappresentazione grafica in Simulink, dell’algoritmo AGM nel caso simulato

### 5.3 Algoritmo su segnali reali

L'implementazione reale dell'algoritmo AGM, si intende l'utilizzo non più di generatori di numeri e quindi di segnali e rumori (vedi paragrafo precedente), ma di veri e propri segnali che partono dallo Sweeper (fig. 4.3) e arrivano al computer passando attraverso l'elaborazione delle USRP2 (vedi capitolo 4). Il problema ora è come far capire a Simulink che sono presenti due periferiche USRP2 e, in particolare, in parallelo sullo stesso segnale di ingresso. In realtà questo problema è stato risolto già dalla MathWorks poiché, dalla versione di Matlab R2011a in poi, sono stati creati (con l'aiuto della NI e della Ettus) dei blocchi che supportano la comunicazione tra Simulink e le schede stesse. Come si denota dalle figure 5.3.1 e 5.3.2, Simulink può implementare entrambe le possibili funzionalità delle USRP2, ossia utilizzandole o in trasmissione e/o in ricezione.

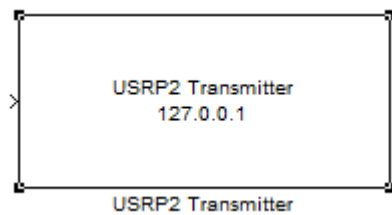


Figura 5.3.1 Esempio di Blocco USRP2 per la trasmissione

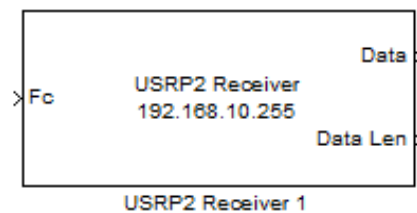


Figura 5.3.2: Blocco USRP2 per la ricezione utilizzato nell'algoritmo

Schematicamente la connessione Simulink-USRP2 per la trasmissione (sopra) e la ricezione (sotto), può essere descritta come in figura:

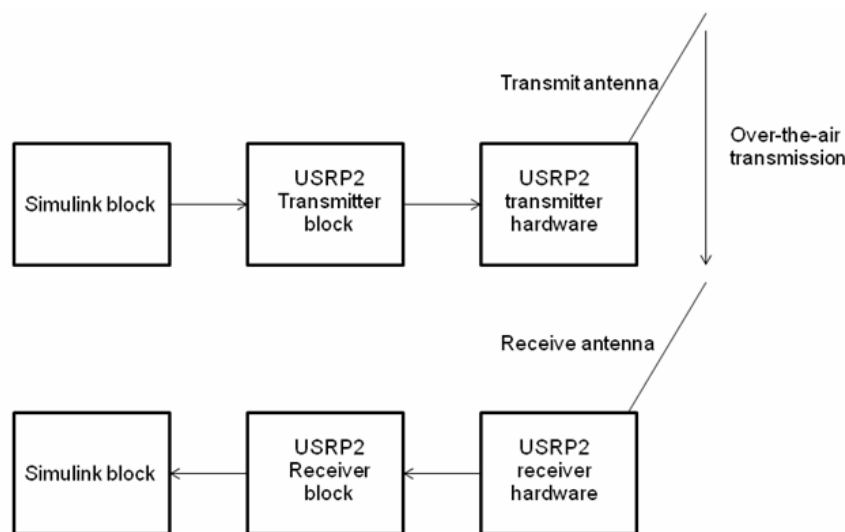


Figura 5.3.3: Schema dell'implementazione Tx e Rx per USRP2 su Simulink

In questo lavoro di tesi è stato utilizzato unicamente il blocco in ricezione poiché, come già detto più volte, si è voluto implementare un sistema multi-antenna con due USRP2 che ricevessero lo stesso segnale. Il blocco *USRP2 Receiver* riceve il segnale e controlla le informazioni che arrivano dalla periferica utilizzando pacchetti UDP (User Datagram Protocol), che è il protocollo utilizzato a livello di trasporto di dati che di solito è associato all'indirizzo IP (infatti si necessita di un indirizzo IP per ogni USRP2). Nella figura e nella descrizione seguenti, viene spiegato nel dettaglio come impostare il ricevitore da Simulink nello stesso modo in cui sono stati impostati i parametri in laboratorio per il raggiungimento dei risultati proposti in seguito.

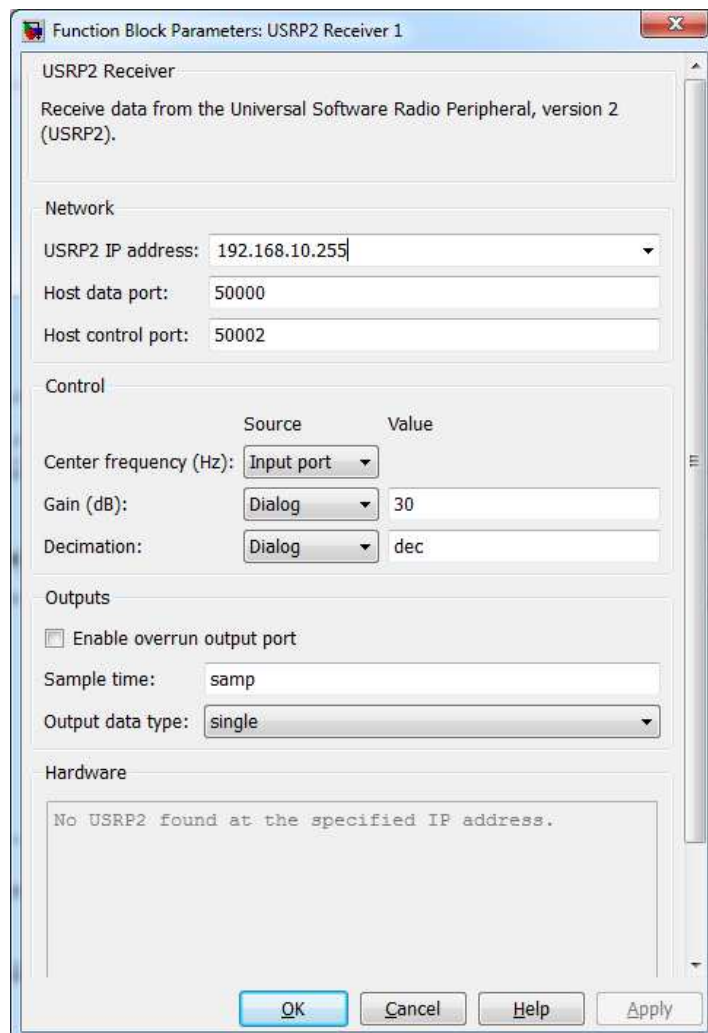


Figura 5.3.4: Pannello di impostazioni del blocco di fig. 5.3.2

Per prima cosa si necessita di impostare un indirizzo IP (in questo caso è un indirizzo privato) con il quale si denota univocamente

l'USRP2 (nel caso visto in laboratorio è stato impostato 192.168.10.255 per la periferica inserita sull'eth0, e 192.168.20.255 per quella sulla eth2) e le porte di controllo e ricezione dati della board (per maggiori dettagli di configurazione di rete e periferiche vedere il documento [15]). Con *Centre Frequency* si imposta la frequenza centrale del segnale in ingresso alla USRP2 (vedi figura 5.3.5); questo parametro, insieme ai prossimi due, può essere direttamente impostato con valori decisi dall'utilizzatore (*Dialog* vedi figura 5.3.4), oppure può direttamente considerarli impostando dei nuovi ingressi nei quali andranno inseriti i suddetti valori (*Input port* vedi figura 5.3.4). *Gain* è il guadagno che il blocco applica all'ingresso della USRP2. *Decimation* è invece il fattore di decimazione che serve all'USRP2 per convertire in banda base la frequenza di utilizzo (512 è il miglior valore di decimazione che si potesse utilizzare; infatti è impostato nel modello di figura 5.1). *Sample time* è invece il tempo di campionamento utilizzato da Simulink; esso viene impostato (come in figura 5.1) applicando la seguente formula:

$$\text{Samp} = \frac{dec}{100 \cdot 10^6} \quad (5.3.1)$$

Quindi, nel caso visto in laboratorio, poiché  $dec=512$ :

$$\text{Samp} = \frac{512}{100 \cdot 10^6} = 5.12 \cdot 10^{-6} \text{ s} \quad (5.3.2)$$

Impostati tutti questi valori, mettendo in modo corretto i blocchi delle USRP2, e riprendendo tutta la parte a blocchi dell'algoritmo del paragrafo precedente, a parte quella simulata, si ottiene la configurazione in figura 5.3.5. In particolare ciò che si nota è il blocchetto *Center Frequency 870MHz* che è la frequenza del segnale sinusoidale che esce dallo Sweeper ed entra nelle USRP2; inoltre l'uscita *Data len* del blocco *Receiver*, rappresenta sempre il numero 358, che è il numero di frame in uscita, scelto da Ettus, per avere un'ottimizzazione hardware e un uso efficiente dei pacchetti Ethernet da 1500 byte.



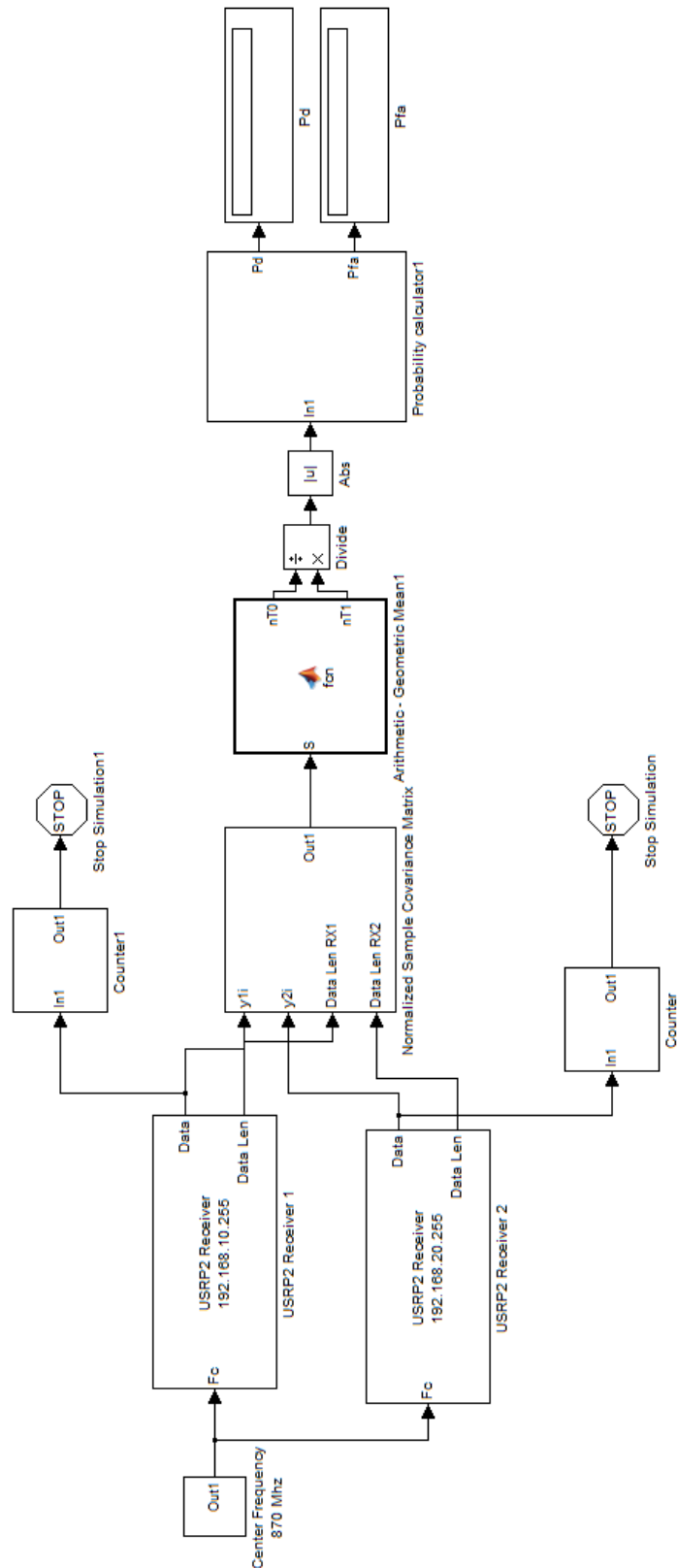


Figura 5.3.5: Rappresentazione grafica in Simulink, dell'algoritmo AGM nel caso reale

## 5.4 Misure sperimentali

Per effettuare alcune misure sperimentali (come ad esempio per il calcolo dalla potenza di rumore), si è utilizzato lo Sweeper (figura 4.3) impostandolo in modo che generasse una sinusoide di potenza pari a -84dBm e frequenza 870MHz e collegandolo alle USRP2. In realtà si è visto (grazie all'algoritmo in figura 5.3.6) come la probabilità di detection sia ancora molto alta (vicina al valore 1), nonostante un'impostazione di potenza di segnale molto bassa (-84dBm). L'obiettivo principale al quale giungere, è quello di tentare di abbassare la potenza di segnale, in modo da raggiungere livelli per i quali questa potenza sia paragonabile a quella di rumore e di conseguenza vedere fino a quali valori del segnale utile, l'algoritmo AGM sia utilizzabile. A questo punto però sorge un grosso problema derivante dal fatto che lo Sweeper non può generare segnali con potenze inferiori ai -85dBm. Per cui al fine di svolgere i calcoli si è giunti ad una realizzazione empirica che permette di rappresentare il segnale con una Potenza Virtuale ( $P_V$ ) che possa andare sotto il livello determinato dal generatore di forme d'onda.

In prima istanza si definisce il rapporto segnale rumore:

$$SNR = \frac{P}{\sigma_U^2} \quad (5.3.3)$$

dove  $P$  è la potenza del segnale utile e  $\sigma_U^2$  è la potenza di rumore delle USRP2. Ora per fare in modo che questo rapporto diminuisca, bisogna diminuire il più possibile  $P$  ma, non riuscendo a superare certi livelli, l'altra opzione è quella di aggiungere del rumore aggiuntivo ( $\sigma_A^2$ ) che sarà il parametro da determinare, poiché la Potenza Virtuale è quella dovrebbe avere il segnale stesso:

$$SNR = \frac{P}{\sigma_U^2 + \sigma_A^2} \quad (5.3.4)$$

Il nuovo rapporto segnale rumore potrà essere definito come il rapporto tra una Potenza Virtuale (che comprenderà anche i contributi relativi al rumore aggiuntivo) e la potenza di rumore delle USRP2; basterà quindi eguagliare le due equazioni 5.3.3 e 5.3.4 per ottenere  $\sigma_A^2$ :

$$\frac{P}{\sigma_U^2 + \sigma_A^2} = \frac{P_V}{\sigma_U^2} \quad (5.3.5)$$

Con semplici calcoli matematici si ottiene (5.3.6):

$$\sigma_A^2 = \sigma_U^2 \left( \frac{P}{P_V} - 1 \right) \quad (5.3.6)$$

L'unico valore incognito che deve ancora essere svelato è la potenza di rumore delle USRP2 ( $\sigma_U^2$ ); per trovarlo si è utilizzato un algoritmo rappresentato in figura 5.3.9.

L'algoritmo si presenta simile a quello AGM reale (infatti per alcuni blocchi già visti, come la frequenza a 870MHz della sinusoide di ingresso, o i blocchi delle USRP2, si rimanda ai capitoli precedenti), nella realtà i blocchi centrali sono configurati in modo completamente diverso: nei blocchi denominati *Noise* è presente un filtro FIR (figura 5.3.7) impostato su valori  $\frac{1}{2}$  e  $-\frac{1}{2}$ .



Figura 5.3.7: filtro FIR

Questo filtro infatti analizza l'*i*-esimo campione e il suo precedente (per questo il meno nel secondo valore impostato); avendo una sinusoide in ingresso con frequenza di campionamento molto elevata, la differenza fra i due campioni contigui può essere considerata nulla, per cui l'unico fattore che può rendere i due campioni diversi è il rumore.

Per calcolare una media di questo rumore è stato impostato il blocco *Noise Medium* (vedi figura 5.3.9) che è composto dai parametri rappresentati in figura 5.3.8: il buffer raccoglie un numero di campioni sufficientemente grande (almeno 10000), poi si esegue il modulo (*Abs*) al quadrato (*Math Function*), e si fa una media (*Mean*) di tutti i campioni trovati.

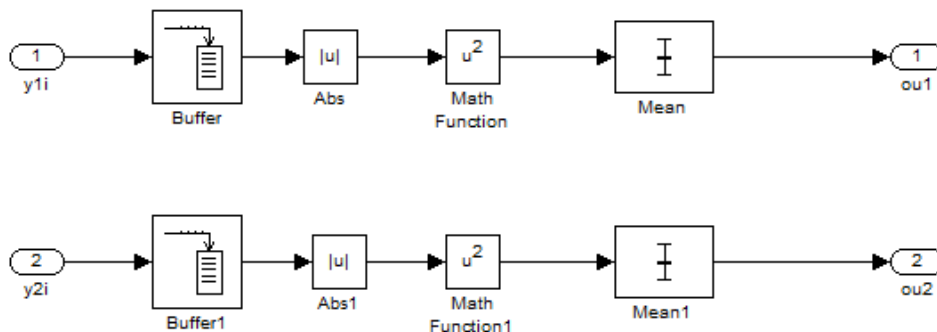


Figura 5.3.8: Particolare del blocco Noise Medium

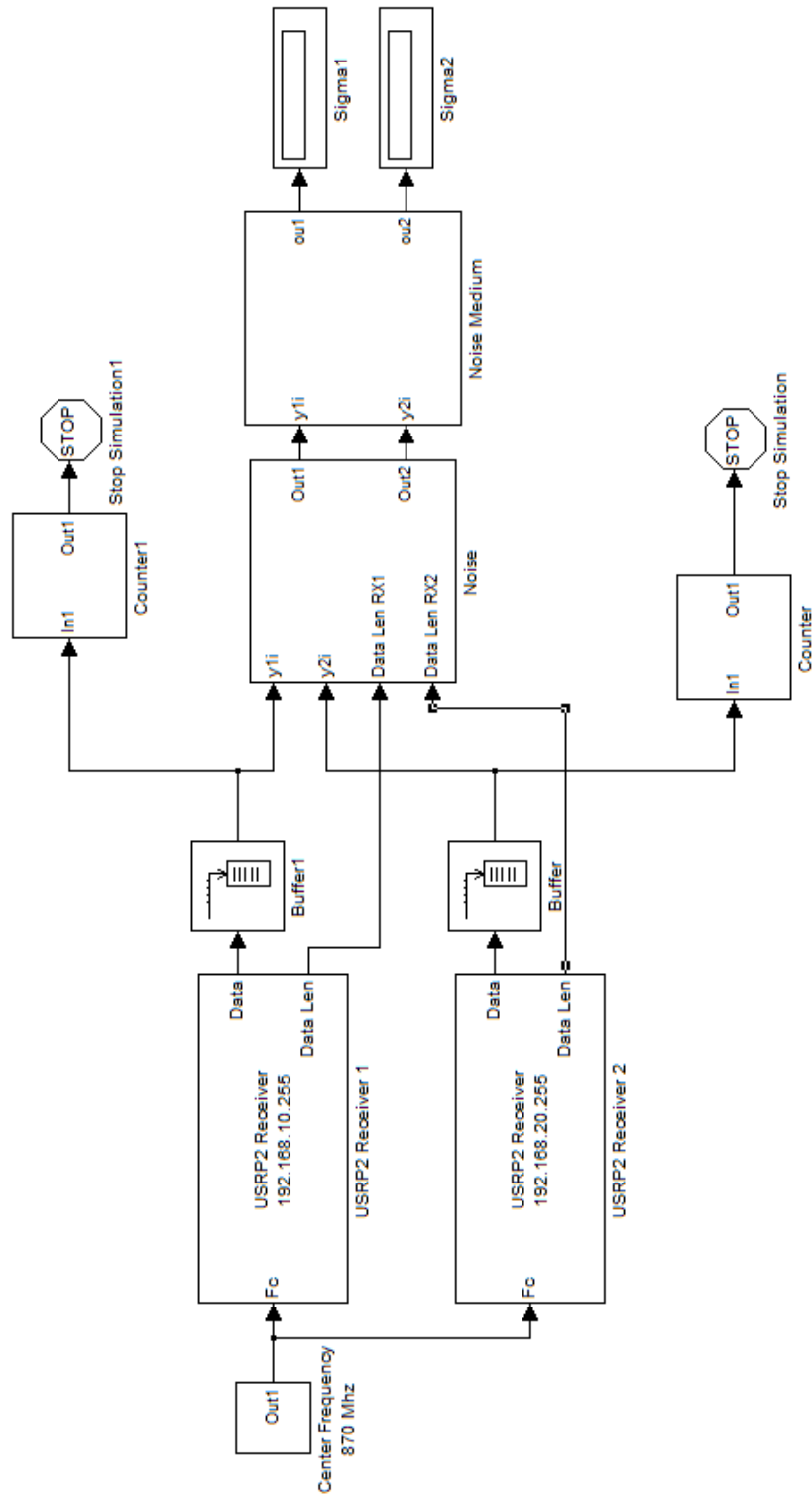


Figura 5.3.9: Algoritmo per la determinazione della potenza di rumore della prima antenna ( $\sigma_1^2$ ) e della seconda antenna ( $\sigma_2^2$ )

Eseguendo l'algoritmo in figura 5.3.9 si riscontrano i valori:

$$\sigma_1^2 = 9.346e^{-10} \text{ V}^2$$

$$\sigma_2^2 = 9.815e^{-10} \text{ V}^2$$

In particolare avendo due antenne in parallelo con potenza di rumore differente, è noto che il rapporto segnale rumore totale delle due antenne è uguale alla media dei rapporti segnale rumore:

$$SNR = \frac{SNR_1 + SNR_2}{2} = \frac{\frac{P}{\sigma_1^2} + \frac{P}{\sigma_2^2}}{2} \quad (5.3.7)$$

Di conseguenza, facendo i calcoli, si trova che la potenza di rumore delle USRP2 è pari a:

$$\sigma_U^2 = 2 \frac{\sigma_1^2 * \sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (5.3.8)$$

Riprendendo in mano la formula 5.3.6 è quindi possibile ottenere da essa tutti i risultati della potenza di rumore aggiuntiva. Per fare ciò è stato imposto un passo di Potenza Virtuale che va dai -84dBm ai -120dBm, considerando potenze ogni 2dBm (vedi tabella 2).

Per implementare il tutto nell'algoritmo AGM reale, è quindi opportuno porre in uscita dai blocchi Reciever delle USRP2, un blocco *random number* che abbia la varianza uguale a quella trovata dalla tabella delle potenze di rumore aggiuntive (vedi figura 5.3.10). Facendo così, si aggiunge potenza di rumore al segnale, potendo quindi raggiungere potenze virtuali anche sotto i -85dBm (limite minimo raggiungibile dallo Sweeper), trovando quindi i punti nei quali l'algoritmo comincia a funzionare male nel senso che la potenza di rumore diventa paragonabile a quella di segnale utile e si ha una diminuzione della probabilità di detection del segnale.

Capitolo 5 – Implementazione Algoritmo di Detection

Tabella 3: In questa tabella sono rappresentati i valori di potenza virtuale, di potenza virtuale in dBm e di potenza di rumore aggiuntiva

$\sigma_1$	$\sigma_2$	$\sigma_{usrp2}$	Potenza	Potenza dBm
9,347E-10	9,815E-10	9,575E-10	3,981E-12	-84
Potenza virtuale		Potenza virtuale (dbm)		$\sigma$ aggiuntivo
3,981E-12		-84		0
2,512E-12		-86		5,6E-10
1,585E-12		-88		1,45E-09
1E-12		-90		2,85E-09
6,31E-13		-92		5,08E-09
3,981E-13		-94		8,62E-09
2,512E-13		-96		1,42E-08
1,585E-13		-98		2,31E-08
1E-13		-100		3,72E-08
6,31E-14		-102		5,95E-08
3,981E-14		-104		9,48E-08
2,512E-14		-106		1,51E-07
1,585E-14		-108		2,4E-07
1E-14		-110		3,8E-07
6,31E-15		-112		6,03E-07
3,981E-15		-114		9,57E-07
2,512E-15		-116		1,52E-06
1,585E-15		-118		2,4E-06
1E-15		-120		3,81E-06

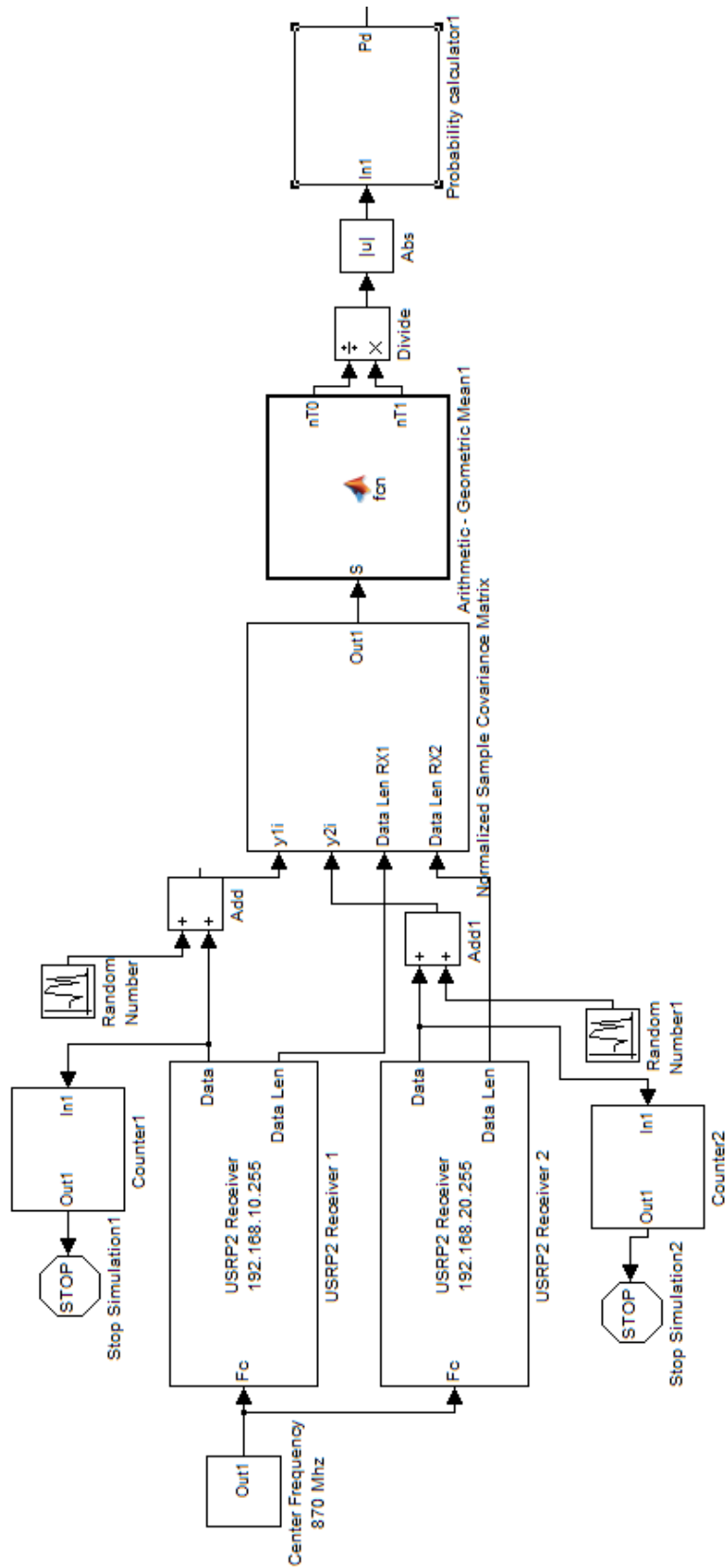


Figura 5.3.10: AGM reale con aggiunta di rumore per  $P_v = -120\text{dBm}$





## CAPITOLO 6

### Conclusioni

In questa tesi è stato ripreso e sviluppato, mediante l'implementazione di algoritmi di detection, il concetto di radio cognitiva. L'approccio che propone questo concetto, nato già da almeno un decennio ma sviluppato solo da qualche anno a questa parte, è innovativo poiché cerca di sfruttare una risorsa molto importante come lo spettro radio in modo intelligente e da non sprecarlo. L'aggettivo "intelligente" è stato più volte proposto in questa tesi nei riguardi della cognitive radio, questo perché essa cerca una soluzione ragionata, funzionale, efficiente al problema della saturazione della banda utilizzabile per le trasmissioni, basandosi su una utilizzazione di frequenze funzionale ai fini dell'utente secondario ed inoltre dinamica a seconda della trasmissione dell'utente primario.

Successivamente è stato preso in considerazione il concetto più importante che è parte della radio cognitiva: lo spectrum sensing. Questa tecnica è fondamentale nell'implementazione di dispositivi cognitivi; infatti sono i secondary user ad eseguire l'analisi dello spettro, determinando la possibilità o meno di trasmettere, senza che gli utenti primari, che hanno pagato per quella banda su cui si vuole trasmettere, debbano inviare eventuali informazioni.

Sono stati proposti gli algoritmi independence, sphericity e arithmetic-geometric mean. Quest'ultimo in particolare è stato analizzato più in dettaglio poiché permette di analizzare la banda e riscontrare la presenza o meno di segnali senza avere l'obbligo di settare una soglia, impostando un rapporto fra le stime del numero di segnali ottenuti. Per creare una arithmetic-geometric mean, si è partiti dalla stima della matrice di covarianza del segnale poi, mediante algoritmi basati sugli autovalori, si è giunti al risultato finale.

Ovviamente per implementare il tutto si è utilizzata un'interfaccia user friendly accessibile da tutti e integrata nelle versioni di Matlab: Simulink. Questo programma semplifica molto l'analisi e l'implementazione di algoritmi, anche complessi, utilizzando un'interfaccia grafica che permette all'utente di capire (già solo controllando i blocchi in cascata) come è composto e strutturato l'algoritmo in sé.

Per l'elaborazione dei dati, l'interfacciamento della ricezione e l'invio dei dati elaborati al computer, sono state utilizzate due perife-

## Capitolo 6 – Conclusioni

riche innovative: USRP2. Questi strumenti sono stati fondamentali per lo sviluppo di questa tesi. Essi fanno propri i concetti base della cognitive radio e della software defined radio, essendo hardware il più possibili adattabili a tutte le realizzazioni che un'utente voglia implementare. Queste periferiche sono facili da utilizzare, configurare ed interfacciare al computer, ed hanno una precisione elevata, nonostante ad esempio la bassa potenza di segnale utilizzata (vedi i -84dBm di potenza di segnale utilizzati negli esperimenti nei capitoli precedenti).

Purtroppo non si è potuto giungere a risultati nel caso dell'AGM reale (vedi capitolo 5). Il problema sta probabilmente nella sincronizzazione; infatti, per il caso simulato, sono stati considerati blocchi che ad ogni campione di segnale aggiungevano rumore. Nel caso reale non è nota la velocità di campionamento delle due USRP2 (che nel caso simulato è sempre uguale per ogni blocco a  $t_s$ , e quindi tutto sincrono) di conseguenza sarà senza dubbio presente uno sfasamento che comporta la non veridicità dei risultati ottenuti in laboratorio. Questo problema potrebbe essere dato sia dal non sincronismo tra le due USRP2, che dal blocco di generazione del rumore il quale purtroppo ha un tempo di campionamento non uguale a quello delle USRP2 (poiché lo “eredita” evidentemente non dalle due periferiche).

Comunque è possibile concludere che gli algoritmi che sfruttano gli autovalori della matrice di covarianza per lo spectrum sensing, in un sistema ad antenne multiple in ricezione con USRP2, offrono buone prestazioni e non richiedono la conoscenza del segnale trasmesso, fattore che permette di utilizzarli nella realtà come ottimo metodo di ricezione di segnali con dispositivi di radio cognitiva.

## Bibliografia

- [1] S. Haykin, “*Cognitive Radio: Brain-Empowered Wireless Communications*”, IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, Feb. 2005.
- [2] C. R. Stevenson, “*Comments of IEEE 802.18 on the report of the commission’s spectrum policy task force*”, IEEE 802.18 Radio Regulatory Technical Advisory Group, January 2003.
- [3] S. Palaninathan, J. Sydor, B. Doray, J. Hu, “*Cognitive Radio Networking in the ISM Band*”, [http://www.crc.gc.ca/files/crc/home/wifi\\_cr/knowledge/winnforum\\_\\_crc\\_29sept2011.pdf](http://www.crc.gc.ca/files/crc/home/wifi_cr/knowledge/winnforum__crc_29sept2011.pdf), Canada, 2011.
- [4] J. Mitola, “*Cognitive Radio An Integrated Agent Architecture for Software Defined Radio*”, PhD thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2000.
- [5] [http://www.crc.gc.ca/en/html/crc/home/wifi\\_cr/coral\\_brochure](http://www.crc.gc.ca/en/html/crc/home/wifi_cr/coral_brochure)
- [6] H. Arslan, “*Cognitive Radio, Software Defined Radio, and Adaptive Wireless System*”, Edizione Springer, 2007.
- [7] Patrick Sanzani, “*Implementazione di algoritmi per lo Spectrum Sensing su piattaforme per Radio Cognitive*”, Tesi di Laurea magistrale, Università di Bologna, Relatore Prof. Andrea Giorgetti, 2011
- [8] Z. Quan, S. Cui, A. H. Sayed, and H. V. Poor, “*Spatial-spectral joint detection for wideband Spectrum Sensing in Cognitive Radio networks*”, Department of Electrical Engineering, University of California, Los Angeles, CA
- [9] T. Yücek and H. Arslan, “*A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications*”, IEEE Tutorials, vol. 11, no. 1, 2009.
- [10] A. Mariani, A. Giorgetti, e M. Chiani, “*Test of Independence for Cooperative Spectrum Sensing with Uncalibrated Receivers*”, proc. IEEE Global Commun. Conf. (Globecom), Anaheim, USA, Dec. 2012.

## Bibliografia

- [11] Riccardo Crociani, “*Spectrum Sensing per Radio Cognitive: Implementazione su Piattaforma Software Defined Radio e Misure Sperimentali*”, Tesi di Laurea Magistrale, Università di Bologna, Relatore Prof. Marco Chiani, 2010.
- [12] M. Wax and T. Kailath, “*Detection of Signals by Information Theoretic Criteria*”, IEEE Transactions On Acoustics, Speech, And Signal Processing, Vol. Assp-33, No. 2, April 1985.
- [13] M. Dilinger, K. Madani, N. Alonistioti, “*Software Defined Radio: Architectures, Systems and Function*”, Editor Wiley, 2003.
- [14] Report and Order FCC 01-264 (Docket No. 00-47), <http://apps.fcc.gov/eas/comments/GetPublishedDocument.html?id=171&tn=726620>
- [15] Matteo Parrini, *Implementazione di Algoritmi di Spectrum Sensing Cooperativo su Piattaforma USRP2*, Tesi di Laurea, Università di Bologna, Relatore Prof. Andrea Giorgetti, luglio 2012.
- [16] <http://www.mathworks.it/products/matlab/trial.html;jsessionid=786e25dba9d242c10d5a3e5cf5bd>
- [17] <http://www.mathworks.it/products/simulink/>
- [18] <http://www.mathworks.it/help/toolbox/comm/ref/errorrateratecalculation.html>

## Elenco delle Figure

2.1	Analisi di spectrum holes mediante 4 differenti sniffer.....	4
2.2	Applicazione, miglioramento e utilizzo di una rete wireless con metodi di radio cognitiva.....	6
2.3	Rappresentazione grafica di un “hole” in cui vi è la possibilità di trasmettere utilizzando la tecnica dell’accesso dinamico alla banda.....	7
4.1	Pannello frontale della USRP2.....	18
4.2	Architettura USRP2.....	18
4.3	HP Sweeper 83751A utilizzato per le misure in laboratorio; si può notare a sinistra come sia impostata la frequenza a 870 Mhz e a destra la potenza a -84dBm.....	20
4.4	Si nota come in particolare come le due piattaforme siano collegate direttamente tramite cavo ethernet al pc.....	21
4.5	Particolare delle due schede di rete nelle quali vengono inseriti i cavi ethernet che arrivano dalle USRP2.....	22
5.1	Modello in Matlab nel quale vengo definite le variabili.....	24
5.2	Blocco Random Number.....	25
5.3	Parametri impostabili da Random Number.....	25
5.4	Real-Imag to Complex.....	26
5.5	Sum.....	26
5.6	Counter e Stop Simulation.....	26
5.7	Interno del blocco Counter; cliccando su Count up si imposta il numero di campioni da contare prima di dare in uscita il valore Hit che di fatto poi stoppa la simulazione.....	27
5.8	Normalized Sample Covariance Matrix.....	27
5.9	Implementazione della formula teorica della matrice di covarianza, mediante i blocchi di Simulink. Si può notare come i vettori $y_{1i}$ e $y_{2i}$ siano quelli che vanno a comporre la matrice $\mathbf{Y}$ , questi vengono bufferizzati, poi trasposti e concatenati in una matrice (che è proprio la matrice $\mathbf{Y}$ ); giunti a questo punto se ne fa la trasposta e si moltiplicano le due matrici, dividendo tutto per gli $N_S$ campioni considerati.....	28
5.10	Implementazione dell’algoritmo AGM in Simulink, mediante codice Matlab.....	28
5.11:	Codice Matlab che rappresenta come trovare i valori $\hat{N}_T(0)$ e $\hat{N}_T(1)$ .....	29
5.12	Divide.....	29
5.13	Abs.....	29
5.14	Probability calculator.....	29
5.15	Interno del blocco in figura 5.14.....	30
5.16	Rappresentazione grafica in Simulink, dell’algoritmo AGM	

## Elenco delle Figure

nel caso simulato.....	31
5.3.1 Esempio di Blocco USRP2 per la trasmissione.....	32
5.3.2 Blocco USRP2 per la ricezione utilizzato nell'algoritmo.....	32
5.3.3 Schema dell'implementazione Tx e Rx per USRP2 su Simulink.....	32
5.3.4 Pannello di impostazioni del blocco di fig. 5.3.2.....	33
5.3.5 Rappresentazione grafica in Simulink, dell'algoritmo AGM nel caso reale.....	35
5.3.7 Filtro FIR.....	37
5.3.8 Particolare del blocco Noise Medium.....	37
5.3.9 Algoritmo per la determinazione della potenza di rumore della prima antenna ( $\sigma_1^2$ ) e della seconda antenna ( $\sigma_2^2$ ).....	38
5.3.10 AGM reale con aggiunta di rumore per $P_v=-120\text{dBm}$ .....	41

## Elenco delle Tabelle

Tabella 1: Pro e contro di single e double radio [9].....	10
Tabella 2: Descrizione delle decisioni possibili in base ai segnali ricevuti.....	12
Tabella 3: In questa tabella sono rappresentati i valori di potenza virtuale, di potenza virtuale in dBm e di potenza di rumore aggiuntiva.....	40





## Ringraziamenti

In realtà questa parte della tesi dovrebbe essere la più immediata e veloce da scrivere, ma nel mio caso non è stato così, perché non è facile ricordare e trasporre in una misera pagina o poco più, ogni singola persona che mi ha aiutato a raggiungere questo obiettivo.

Innanzitutto è necessario un grosso ringraziamento al Professor Giorgetti che, nonostante la mia crescente ansia all'avvicinarsi del giorno della discussione e la mia insistenza, è riuscito a sopportarmi con estrema disponibilità, professionalità e pazienza. Un altrettanto grande "grazie" va all'Ingegnere Mariani il quale nonostante la mole di lavoro, i viaggi, gli studi che la Ricerca gli impongono, è sempre riuscito a trovare il tempo per inviarmi una e-mail o a darmi una mano per risolvere i miei dubbi. In particolare li ringrazio poiché, tramite questa tesi, mi hanno permesso di vedere come realmente funziona il mondo dell'Università, al di fuori della didattica.

Un'importante ringraziamento va ad una delle mie poche certezze: i miei genitori e le mie sorelle i quali ci sono stati, ci sono e ci saranno sempre per me.

Ora incomincia la parte veramente difficile: cercare di ricordare tutte le persone che in questi quattro anni sono riusciti a capire le mie paranoie e le mie difficoltà cercando di aiutarmi in qualche modo.

Innanzitutto comincio con ringraziare i più temerari, quelli che ogni singolo giorno di Università sono riusciti a sopportarmi.

Grazie a Mario, l'uomo con più pazienza del mondo per un solo motivo: è riuscito sempre a sopportarmi in questi quattro anni di studio, lezioni, disperazione per gli esami e anche serate, senza battere ciglio, grazie davvero.

Grazie a Stefano e Maikol, compagni di studio matto e disperato, senza i quali non ce l'avrei MAI fatta, sia per all'aiuto negli esami, che soprattutto al di fuori dell'Università.

Grazie a Francesca, senza il suo aiuto non sarei mai riuscito a passare gli ultimi esami e riuscire in questa impresa.

Grazie a Fabio, Pietro, Simone e a tutti i membri del "L'allegria" che mi ha sempre supportato e soprattutto hanno permesso di svagarmi dallo stress derivante da studio, esami, ecc..

Grazie a Cesare, Barbara, Monica e Chiara che, nonostante i mille impegni di tutti e cinque, riusciamo ad incontrarci ogni tanto e constatare che nulla cambia fra noi: carichi ed uniti come sempre.

Grazie a tutti gli amici di Forlimpopoli che, nonostante il mio poco tempo, ad ogni ritrovo è come ogni volta, come se non fosse mai passato il tempo.

Grazie a Matteo, che mi ha aiutato all'inizio e in tutto il percorso dell'elaborazione della tesi.

Grazie agli amici, ai compagni di Università, a quelli che, anche se per un minimo secondo, hanno ascoltato le mie lamentele, il mio pessimismo e le mie paranoie, mi hanno compreso e mi sono stati vicini.

Infine, a tutti quanti a cui voglio bene e che senza dubbio mi sarò dimenticato di menzionare:

**GRAZIE!**