

Indice

Introduzione	1
1 Standard PLC	3
1.1 Progetto PRIME	3
1.1.1 Trasformata di Fourier	4
1.1.2 Autocorrelazione	6
1.2 HomePlug	7
1.2.1 Trasformata di Fourier	9
1.2.2 Autocorrelazione	9
1.2.3 Analisi nel continuo	9
2 Ranging	13
2.1 Ranging su PLC	13
3 Simulazioni in Laboratorio	15
3.1 Chirp	15
3.2 Preabolo HomePlug	18
3.3 Algoritmo di ricerca a soglia	20
4 Segnali equivalenti in banda base	23
4.1 Idea di base	23
4.1.1 Calcolo per il Chirp	24
4.1.2 Calcolo per l'HomePlug	25
4.2 Equivalente in banda base dei segnali ricevuti	26
4.3 Algoritmo di ricerca a soglia	29
Appendice: codici Matlab utilizzati	31
Bibliografia	37

Introduzione

Il lavoro che viene descritto in questa tesina riguarda delle misure di ranging, cioè di stima della distanza, che sono alla base di una futura implementazione di una smart grid.

Le smart grid, o “reti intelligenti”, sono una nuova tecnologia che ha come scopo una migliore gestione delle reti di distribuzione di energia elettrica. Queste ultime, infatti, soffrono di un numero elevato di problematiche: elevate perdite, l'impossibilità di una gestione dei flussi per una loro modifica dinamica, tempi di attesa lunghi in caso di blackout o di interruzione di una linea. Per risolvere questi problemi si è pensato di ricorrere all'uso delle telecomunicazioni: se si fa infatti un confronto con una rete di calcolatori si può notare che gli stessi ostacoli vengono superati grazie ad una gestione attiva dei vari nodi che compongono la rete, i quali intervengono prontamente e modificano il percorso delle informazioni in caso di necessità. Ed è proprio dall'idea di implementare questi benefici nella rete elettrica che nascono le smart grid.

Una smart grid è, quindi, in grado di evitare sprechi distribuendo l'energia coordinando tra di loro, tramite protocolli e flussi informativi, i vari generatori e carichi della rete. Per fare ciò è necessario un cambiamento radicale dei nodi della rete, non più nodi passivi in cui l'energia viaggia in maniera unidirezionale, ma nodi attivi in grado di dialogare costantemente tra di loro.

Il Capitolo 1 di questa tesina si concentra su uno studio analitico di due diversi segnali che vengono utilizzati per le comunicazioni di dati sulla rete elettrica e che sono stati usati nelle prove in laboratorio.

Il Capitolo 2 si sofferma sul ranging e sugli algoritmi per la stima della distanza.

Nel Capitolo 3 viene descritto in particolare il lavoro di misurazione svolto in laboratorio per verificare la bontà di un algoritmo di ranging da utilizzare poi in queste tipologie di reti.

Nel Capitolo 4, infine, vengono esaminati i segnali equivalenti in banda base e compiuti su di essi le stesse prove di ranging.

1 Standard PLC

Come prima cosa, lavorando sulle smart grid, è necessario parlare di powerline communication (PLC), ovvero della tecnologia utilizzata per la trasmissione di dati impiegando la stessa rete di distribuzione di energia elettrica come mezzo trasmissivo.

Per sfruttare la presenza di tale rete sul territorio a questo scopo sono stati pensati diversi standard e create diverse tecnologie in quanto, normalmente, le reti elettriche vengono utilizzate per trasmettere tensioni, potenze e frequenze completamente diverse da quelle che caratterizzano i segnali dati.

In particolare ci si sofferma sullo studio di due standard diversi, il progetto PRIME e l'HomePlug, quest'ultimo di gran lunga il più conosciuto e utilizzato nell'ambito PLC. Per entrambi gli standard vengono analizzati i segnali del preambolo, in quanto sono i segnali che sono stati poi utilizzati nelle prove di laboratorio (Capitolo 3). Si è scelto di utilizzare tali segnali di preambolo, ovvero i segnali che si trovano all'inizio di ogni pacchetto trasmesso, perché essi hanno come scopo la sincronizzazione con il dispositivo ricevente e quindi possiedono particolari caratteristiche di correlazione che risultano essere molto comode.

In questo primo capitolo vengono quindi analizzati questi due preamboli studiane le caratteristiche nel tempo e nella frequenza.

1.1 Progetto PRIME

Un primo tipo di segnale che è possibile utilizzare come preambolo nella trasmissione powerline è il Chirp, segnale molto conosciuto ed utilizzato fin dalla nascita del radar. Tale segnale viene utilizzato, ad esempio, nel progetto PRIME (Powerline Intelligent Metering Evolution) lanciato nel 2007 che definisce una serie di standard a livello fisico e MAC. In particolare a livello fisico PRIME si basa sul multiplexing OFDM nella banda CENELEC-A (9-95 KHz), centrando la prima portante a 41992,1875 Hz, l'ultima portante a 88867,1875 Hz e inserendo uno spazio tra una portante e l'altra di 488,28125 Hz.

Andando ad analizzare in particolare il preambolo che viene utilizzato da tale progetto, notiamo che esso ha una durata di 2,048 ms e viene descritto come segue

$$s(t) = A \cdot \text{rect}(t/T) \cdot \cos[2\pi(f_0 t + (1/2)\mu t^2)]$$

dove $T = 2048 \mu s$, $f_0 = 41992 Hz$ (frequenza d'inizio), $f_1 = 88867 Hz$ (frequenza finale), $\mu = (f_1 - f_0)/T$ e

$$\text{rect}(t) = \begin{cases} 1 & \text{per } 0 \leq t \leq 1 \\ 0 & \text{altrove} \end{cases}.$$

In Figura 1.1 viene riportato il grafico supponendo $A = 1$.

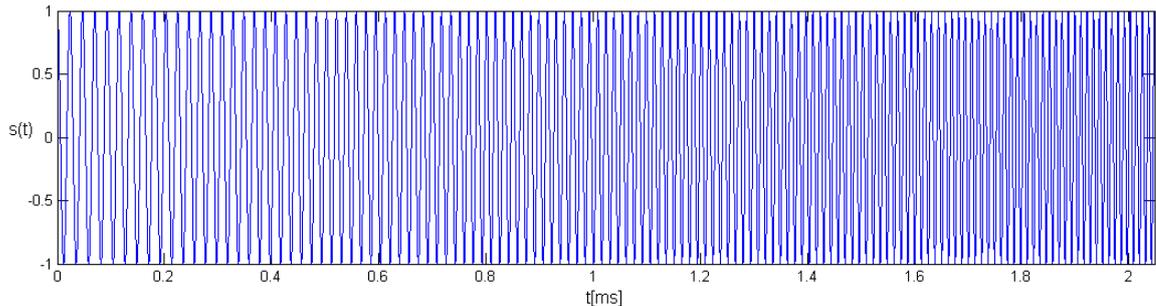


Figura 1.1: Andamento del Chirp

Ora verrà analizzato nello specifico tale segnale calcolando, sia in forma analitica che attraverso Matlab, il modulo della trasformata di Fourier e la sua autocorrelazione.

1.1.1 Trasformata di Fourier

Il calcolo per ricavare la trasformata di Fourier di un segnale Chirp è molto conosciuto, vengono riportati i calcoli ritrovati in letteratura[3].

La trasformata è data da

$$S(f) = \int_{-T/2}^{T/2} e^{j(\mu \frac{t^2}{2} - 2\pi ft)} dt$$

la quale si può modificare riconducendo la quantità tra parentesi tonde ad una differenza di quadrati del tipo

$$\mu \frac{t^2}{2} - 2\pi ft = (a - b)^2 - b^2.$$

Da qui si ricava $a = t\sqrt{\frac{\mu}{2}}$ $b = \pi f\sqrt{\frac{2}{\mu}}$, ovvero

$$\mu \frac{t^2}{2} - 2\pi ft = \left(t\sqrt{\frac{\mu}{2}} - \pi f\sqrt{\frac{2}{\mu}} \right)^2 - \frac{2(\pi f)^2}{\mu}$$

Si può dunque scrivere la trasformata come

$$S(f) = e^{-j\frac{(2\pi f)^2}{2\mu}} \int_{-T/2}^{T/2} e^{j\left(t\sqrt{\frac{\mu}{2}} - \pi f\sqrt{\frac{2}{\mu}}\right)} dt.$$

Applicando la sostituzione

$$t\sqrt{\frac{\mu}{2}} - \pi f\sqrt{\frac{2}{\mu}} = y\sqrt{\frac{\pi}{2}}$$

si ottiene

$$S(f) = \sqrt{\frac{\pi}{\mu}} e^{-j\frac{(2\pi f)^2}{2\mu}} \int_{X_1}^{X_2} e^{j\frac{\pi y^2}{2}} dy$$

dove

$$X_1 = \sqrt{\frac{\mu}{\pi}} \left(\frac{T}{2} + \frac{2\pi f}{\mu} \right) \quad X_2 = \sqrt{\frac{\mu}{\pi}} \left(\frac{T}{2} - \frac{2\pi f}{\mu} \right)$$

Ricordando gli integrali di Fresnel

$$\begin{cases} C(X) = \int_0^X \cos\left(\frac{\pi y^2}{2}\right) dy \\ S(X) = \int_0^X \sin\left(\frac{\pi y^2}{2}\right) dy \end{cases}$$

si ha

$$S(f) = \left(\sqrt{\frac{\pi}{\mu}} \right) e^{-j\frac{(2\pi f)^2}{2\mu}} \{C(X_2) + C(X_1) + j[S(X_2) + S(X_1)]\}$$

L'andamento è illustrato in Figura 1.2

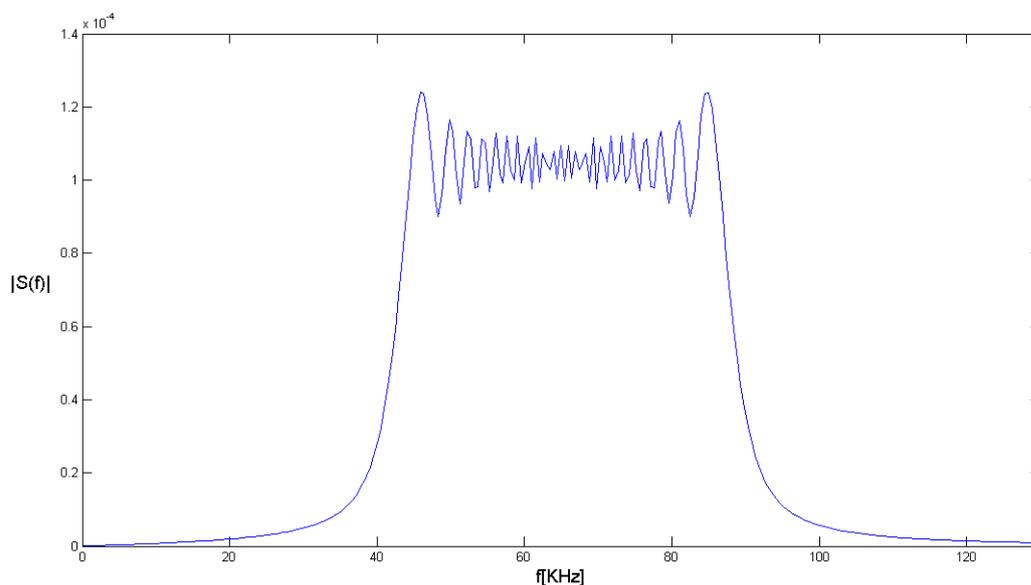


Figura 1.2: La trasformata di Fourier del Chirp a frequenze positive

1.1.2 Autocorrelazione

L'autocorrelazione può essere scritta come

$$\begin{aligned} R_s(t) &= \int_{-\infty}^{+\infty} e^{-j\pi\mu\tau^2} \text{rect}\left(\frac{\tau}{T}\right) e^{j\pi\mu(t+\tau)^2} \text{rect}\left(\frac{t+\tau}{T}\right) d\tau \\ &= e^{j\pi\mu t^2} \int_{-\infty}^{+\infty} \text{rect}\left(\frac{\tau}{T}\right) e^{j2\pi\mu t\tau} \text{rect}\left(\frac{t+\tau}{T}\right) d\tau \end{aligned}$$

che, per $0 < t < T$, restituisce

$$\begin{aligned} R_s(t) &= e^{j\pi\mu t^2} \int_{-\frac{T}{2}}^{\frac{T}{2}-t} e^{j2\pi\mu t\tau} d\tau \\ &= \frac{e^{j\pi\mu t^2}}{j2\pi\mu t} \left[e^{j\pi\mu t(T-2t)} - e^{-j\pi\mu t T} \right] \\ &= \frac{e^{j\pi\mu t(T-t)} - e^{-j\pi\mu t(T-t)}}{j2\pi\mu t} \end{aligned}$$

Quindi, applicando la formula di Eulero, si ottiene

$$R_s(t) = \frac{1}{\pi\mu t} \sin[\pi\mu t(T-t)]$$

Analogamente, calcolando per $-T < t < 0$ si trova un'espressione identica alla precedente.

In definitiva unendo i due risultati si ottiene la funzione di autocorrelazione per $-T < t < T$, ovvero

$$\begin{aligned} R_s(t) &= \frac{1}{\pi\mu t} \sin[\pi\mu t(T-|t|)] \\ &= (T-|t|)\text{sinc}[\mu t(T-|t|)] \end{aligned}$$

in Figura 1.3 viene illustrato l'andamento dell'autocorrelazione.

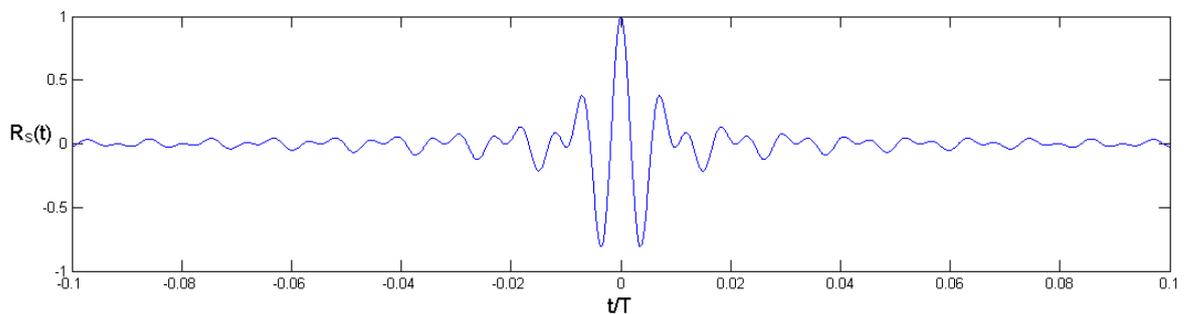


Figura 1.3: Autocorrelazione del Chirp

1.2 HomePlug

L'Homeplug è una serie di standard sulle reti powerline creati da un'alleanza di circa 50 aziende che ne hanno definito le specifiche. Attualmente le versioni disponibili sono HomePlug 1.0, HomePlug AV e HomePlug GP.

In tutti i casi come tecnica di trasmissione viene utilizzata l'OFDM o il ROBO (ROBust OFDM), con 128 portanti di cui 84 riservate alla trasmissione e viene utilizzata una banda tra 0 e 25 MHz.

Nella Figura 1.4 viene riportata la struttura di un frame, da essa si può notare che il preambolo è costituito da 7.5 simboli, ognuno dei quali è formato da 256 campioni. I primi 6 simboli sono denominati SYNCN in quanto vengono moltiplicati per 1 dalla funzione segno, gli ultimi SYNCN perché vengono moltiplicati per -1 dalla stessa funzione.

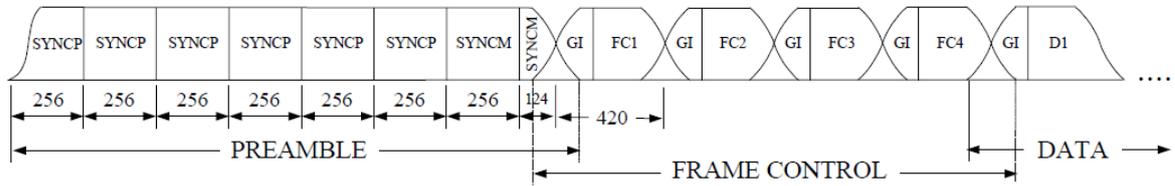


Figura 1.4: Struttura di un frame secondo le specifiche HomePlug

La forma d'onda del preambolo viene descritto dall'espressione

$$\tilde{h}(nT_s) = \frac{\text{sign}(nT_s) \cdot w(nT_s) \cdot 2}{\sqrt{84}} \sum_{c=0}^{83} \cos(2\pi f_c n + \phi_c)$$

Dove $T_s = 0.02 \mu s$, $T_0 = 256$, $T = 5.12 \mu s$, $f_c = \frac{(c+23)}{256}$ e

$$\text{sign} = \begin{cases} 1 & 0 \leq n \leq 6 \cdot 256 - 1 \\ -1 & 6 \cdot 256 - 1 \leq n \leq 7.5 \cdot 256 - 1 \end{cases}$$

La funzione $w(nT_s)$ invece modifica solo i primi e gli ultimi 7 campioni creando una rampa iniziale e finale.

$$w(nT_s) = \begin{cases} 0.5 \left[1 - \cos\left(\frac{\pi n}{8}\right) \right] & 0 \leq n \leq 7 \\ 1 & 8 \leq n \leq 7.5 \cdot 256 - 8 \\ 0.5 \left[1 - \cos\left(\frac{\pi(n-7.5 \cdot 256)}{8}\right) \right] & 7.5 \cdot 256 - 7 \leq n \leq 7.5 \cdot 256 - 1 \end{cases}$$

I valori di ϕ_c che generano il preambolo vengono riportati in tabella 1.1

In Figura 1.5 viene riportato l'andamento di uno dei simboli centrali.

c	ϕ	c	ϕ	c	ϕ	c	ϕ
0	0	21	$4(\pi/8)$	42	$4(\pi/8)$	63	0
1	0	22	0	43	$12(\pi/8)$	64	$4(\pi/8)$
2	$15(\pi/8)$	23	$11(\pi/8)$	44	$3(\pi/8)$	65	$7(\pi/8)$
3	$15(\pi/8)$	24	$7(\pi/8)$	45	$11(\pi/8)$	66	$11(\pi/8)$
4	$14(\pi/8)$	25	$2(\pi/8)$	46	$2(\pi/8)$	67	$14(\pi/8)$
5	$13(\pi/8)$	26	$13(\pi/8)$	47	$9(\pi/8)$	68	$\pi/8$
6	$12(\pi/8)$	27	$8(\pi/8)$	48	0	69	$4(\pi/8)$
7	$11(\pi/8)$	28	$3(\pi/8)$	49	$7(\pi/8)$	70	$7(\pi/8)$
8	$9(\pi/8)$	29	$13(\pi/8)$	50	$13(\pi/8)$	71	$9(\pi/8)$
9	$7(\pi/8)$	30	$7(\pi/8)$	51	$3(\pi/8)$	72	$11(\pi/8)$
10	$6(\pi/8)$	31	$2(\pi/8)$	52	$10(\pi/8)$	73	$14(\pi/8)$
11	$3(\pi/8)$	32	$11(\pi/8)$	53	$15(\pi/8)$	74	$15(\pi/8)$
12	$\pi/8$	33	$5(\pi/8)$	54	$5(\pi/8)$	75	$\pi/8$
13	$15(\pi/8)$	34	$15(\pi/8)$	55	$11(\pi/8)$	76	$3(\pi/8)$
14	$12(\pi/8)$	35	$8(\pi/8)$	56	0	77	$4(\pi/8)$
15	$9(\pi/8)$	36	$\pi/8$	57	$5(\pi/8)$	78	$5(\pi/8)$
16	$6(\pi/8)$	37	$10(\pi/8)$	58	$10(\pi/8)$	79	$6(\pi/8)$
17	$3(\pi/8)$	38	$3(\pi/8)$	59	$15(\pi/8)$	80	$7(\pi/8)$
18	$15(\pi/8)$	39	$11(\pi/8)$	60	$3(\pi/8)$	81	$7(\pi/8)$
19	$12(\pi/8)$	40	$4(\pi/8)$	61	$8(\pi/8)$	82	$8(\pi/8)$
20	$8(\pi/8)$	41	$12(\pi/8)$	62	$12(\pi/8)$	83	$8(\pi/8)$

Tabella 1.1: La funzione ϕ_c

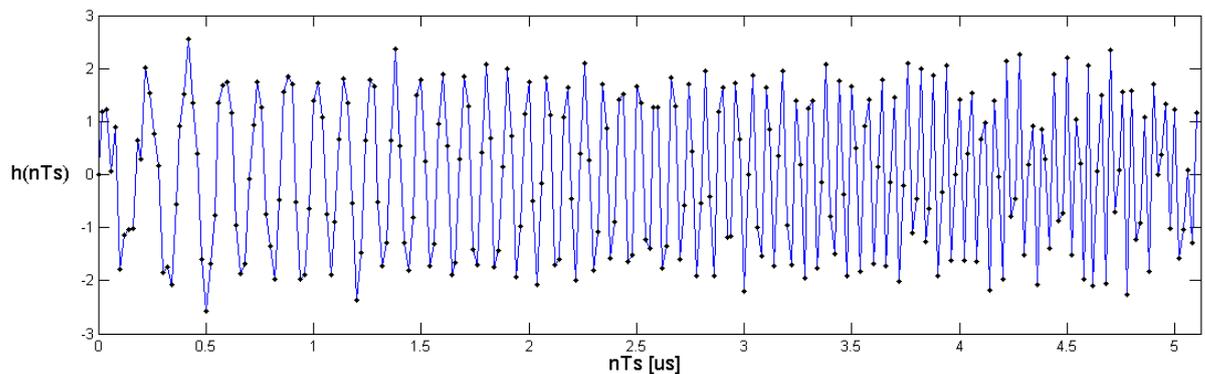


Figura 1.5: Andamento del preambolo HomePlug

Ora, come abbiamo fatto per il segnale Chirp, verrà analizzato questo simbolo calcolandone la trasformata di Fourier e l'autocorrelazione.

1.2.1 Trasformata di Fourier

Tramite Eulero riscrivo la forma d'onda di un simbolo centrale del preambolo come

$$\tilde{h}(nT_s) = \frac{1}{\sqrt{84}} \sum_{c=0}^{83} [e^{j(2\pi f_c n + \phi_c)} + e^{-j(2\pi f_c n + \phi_c)}]$$

la cui trasformata è

$$\tilde{H}(f) = \frac{1}{\sqrt{84}} \sum_{c=0}^{83} e^{j\phi_c} \delta(f - f_c) + e^{-j\phi_c} \delta(f - f_c)$$

1.2.2 Autocorrelazione

L'espressione analitica dell'autocorrelazione è molto complessa da calcolare. Ci siamo limitati a calcolarla in Matlab ed è riportata in Figura 1.6.

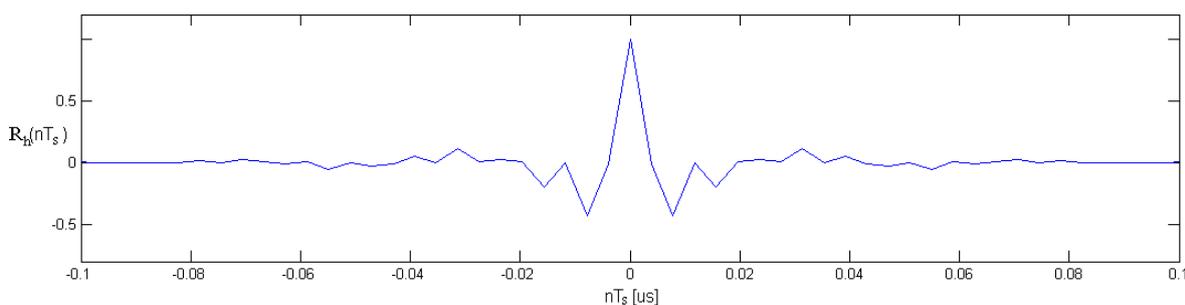


Figura 1.6: Autocorrelazione del preambolo HomePlug

1.2.3 Analisi nel continuo

Il segnale definito dallo standard HomePlug viene descritto su tempi discreti. Risulta quindi interessante andare ad osservarne l'andamento nel continuo. Per farlo è stata effettuata un'interpolazione con Matlab. Il risultato si può osservare in Figura 1.7. Nelle Figure 1.8 e 1.9 è possibile invece osservare l'andamento in frequenza e l'autocorrelazione.

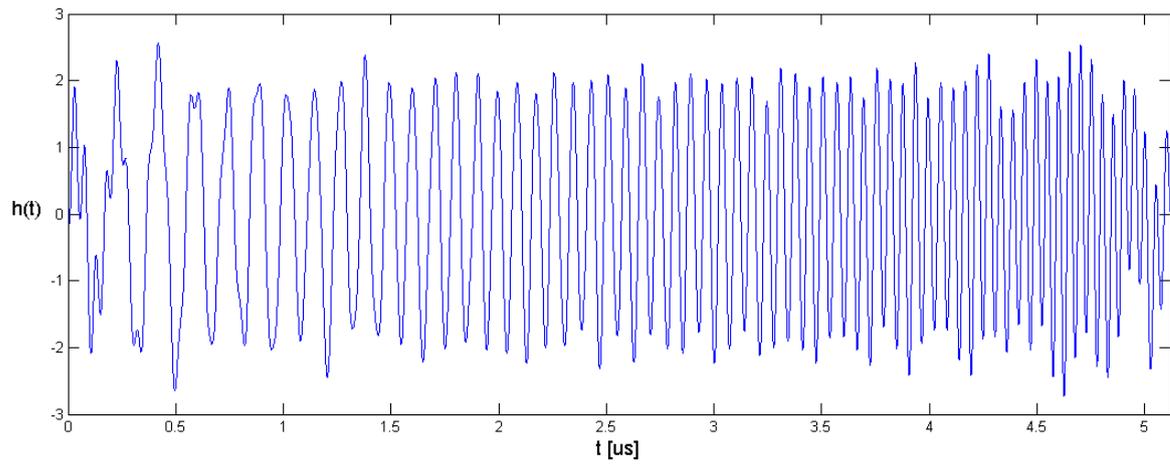


Figura 1.7: Andamento del preambolo HomePlug nel continuo

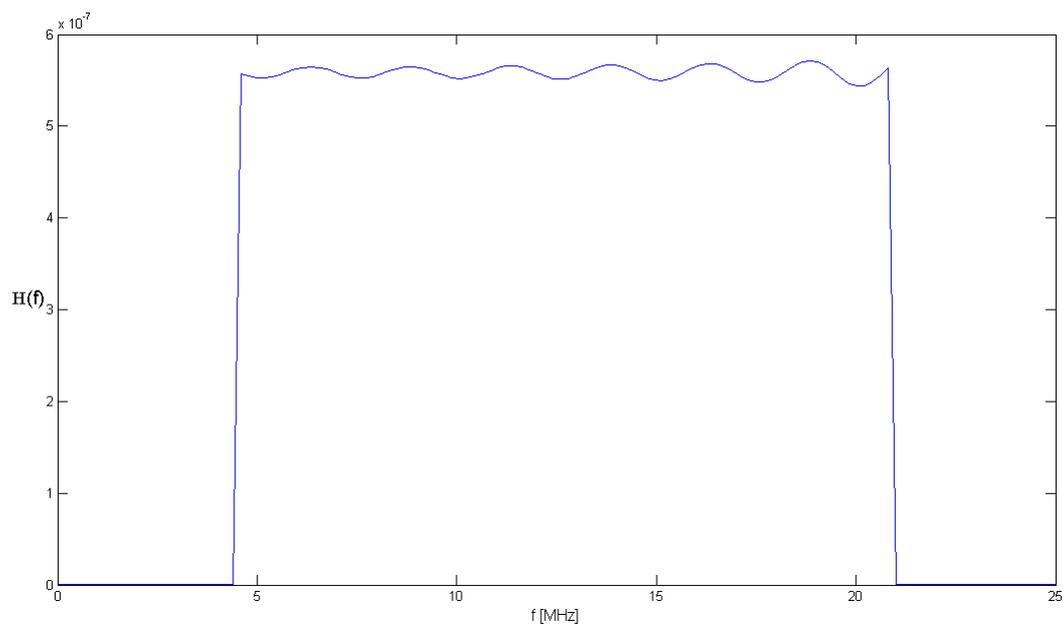


Figura 1.8: Trasformata di Fourier del preambolo HomePlug nel continuo a frequenze positive

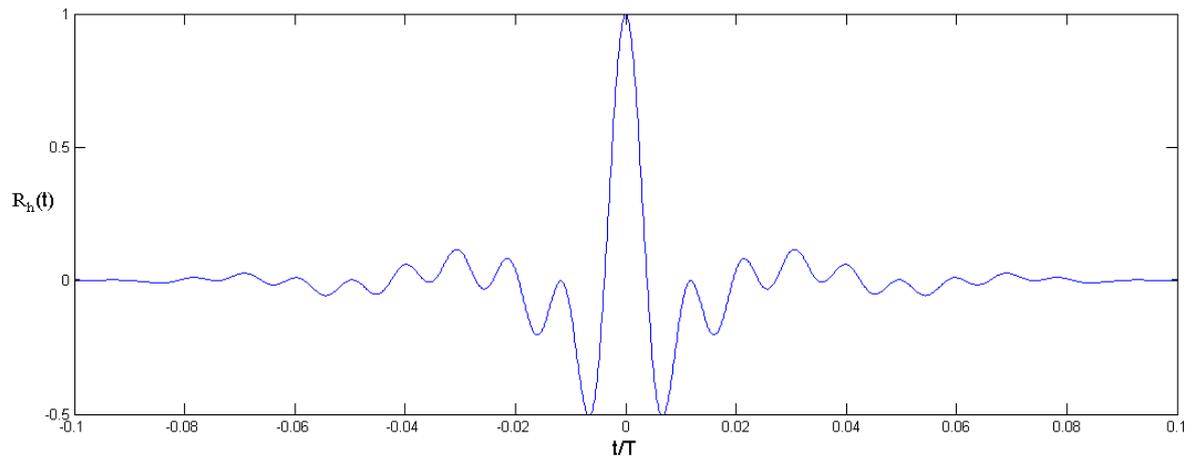


Figura 1.9: Autocorrelazione del preambolo HomePlug nel continuo

2 Ranging

Lo scopo delle tecniche di ranging è quello di calcolare, attraverso opportuni algoritmi, la posizione di un certo dispositivo e quindi la sua distanza da un punto definito. Questo calcolo è estremamente importante nel caso delle smart grid in quanto ogni nodo della rete, per poter lavorare in maniera attiva modificando la direzione del flusso di energia elettrica, deve conoscere in ogni istante la tipologia della rete stessa di cui fa parte ma soprattutto la distanza abbastanza precisa degli altri nodi collegati direttamente.

Le più utilizzate tecniche di ranging sono raggruppabili in due categorie: quelle che sfruttano la potenza del segnale ricevuto e quelle che si basano sui tempi di arrivo (TOA, Time Of Arrival). Nel nostro caso sono assai più semplici da implementare e anche più performanti quelle del secondo tipo[4]. La tecnica denominata *two way handshake* (Fig. 2.1), ad esempio, si basa sul calcolo del round-trip time (RTT). Il nodo A trasmette un pacchetto al nodo B il quale lo riceve dopo un tempo τ_B e risponde al mittente con un pacchetto di acknowledgment dopo un tempo conosciuto T . In tal modo, il primo nodo riceverà il pacchetto di riscontro dopo un tempo $\tau_A = 2\tau_B + T$. Da cui si può ricavare facilmente la distanza, visto che $\tau_B = d/c$, ovvero: $d = c \frac{(\tau_A - T)}{2}$.

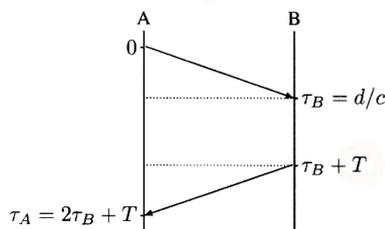


Figura 2.1: Rappresentazione schematica della tecnica two way handshake

2.1 Ranging su PLC

Quando si implementa tale tecnica di ranging su dei segnali powerline il problema principale che ne nasce è il multipath. Interferenze costruttive e distruttive dovute ad echi del segnale stesso arrivano al ricevitore rendendo difficile riconoscere il momento in cui viene ricevuto il pacchetto. Per superare questo problema sono stati ideati diversi algoritmi per identificare il momento di arrivo; anche in questo caso

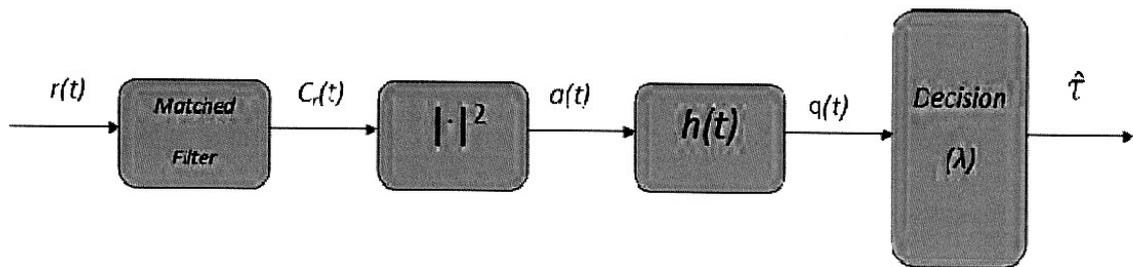


Figura 2.2: Schema a blocchi dell'algoritmo a soglia

l'algoritmo più semplice, l'algoritmo a soglia, risulta essere abbastanza performante per i nostri scopi [4].

La Figura 2.2 descrive lo schema a blocchi dell'algoritmo. In primo luogo il segnale ricevuto, $r(t)$, viene filtrato dal matched filter. Del segnale che ne esce viene preso il modulo ed elevato al quadrato. A questo punto l'algoritmo esegue un secondo filtraggio, questa volta con un filtro $h(t) = \text{rect}(t/K)$, dove con K si indica la durata temporale, variabile tra 0 e 10. Il segnale uscente viene infine processato secondo la seguente espressione al fine di ricavare un valore stimato di tempo di arrivo

$$\frac{q(t) - Q_{min}}{Q_{MAX} - Q_{min}} \leq \lambda$$

dove con Q_{min} ed Q_{MAX} sono indicati rispettivamente il minimo ed il massimo del segnale $q(t)$. Si ottiene quindi un segnale normalizzato con un'ampiezza che varia tra 0 ed 1. Con λ si indica invece la soglia oltre la quale il campione viene salvato e dal quale verrà estratto il valore del ritardo temporale corrispondente. Questa deve essere scelta in maniera da ottenere una stima il più precisa possibile. Se il suo valore dovesse essere troppo basso, infatti, si rischierebbe di selezionare un valore temporale corrispondente a del rumore e calcolare quindi una distanza troppo bassa rispetto alla realtà. Se invece il valore dovesse essere troppo alto, il rischio è di perdere il picco corrispondente al cammino diretto e di calcolare quindi una distanza più alta rispetto alla realtà.

Per avere un confronto su cui basarsi sono stati presi in esame diversi valori di λ : $\{0.02, 0.04, 0.06, 0.1, 0.2, 0.3, 0.35, 0.4, 0.45, 0.5\}$.

3 Simulazioni in Laboratorio

In questo capitolo verranno esposti i risultati ottenuti durante le prove in laboratorio che sono state eseguite allo scopo di testare la bontà degli algoritmi esposti sinora.

Per condurre tali verifiche sono stati utilizzati un generatore di funzioni Tektronix AFG3010 e un oscilloscopio Tektronix TDS5000. Per poter generare i segnali creati con Matlab essi sono stati prima esportati in formato .csv (vedi Appendice), formato supportato dal programma Arbexpress, creato dalla Tektronix. Una volta collegato il computer al generatore di funzioni, questo programma gestisce il dialogo tra le due macchine. In tal modo è stato quindi possibile generare i segnali esattamente come erano stati creati. Tramite lo stesso programma è stato anche possibile variare alcune funzioni del generatore stesso (ampiezza del segnale, durata del periodo di trasmissione). La frequenza di campionamento non era invece modificabile dall'utente ma imposta dalla macchina. Nel nostro caso risultava essere di 100 MHz, si è quindi provveduto a cambiare i segnali da trasmettere in maniera da avere la stessa frequenza.

A questo punto il generatore è stato collegato direttamente con l'oscilloscopio per verificare l'esattezza delle onde generate e studiarne il comportamento in condizioni ottimali. Il connettore Trigger Output sul pannello anteriore del generatore consente di prelevare il segnale di riferimento da collegare all'oscilloscopio permettendo così di sincronizzare i due strumenti tra di loro. Dall'oscilloscopio è stato possibile salvare la memoria di visualizzazione. La frequenza di acquisizione del dispositivo varia a seconda della scala e della risoluzione che vengono scelte operando direttamente sul pannello dello strumento. Si è scelto di svolgere tutte le prove ad una frequenza di campionamento di 250 MHz.

In un secondo momento è stato utilizzato per il collegamento un cavo BNC RG58 da 50Ω lungo 75 metri. In questa maniera è stato possibile studiare i problemi derivanti da attenuazioni, ritardi e rumore. I segnali acquisiti durante tutte queste prove sono stati quindi riportati sul computer per poterli poi acquisire ed analizzare tramite Matlab.

3.1 Chirp

Il segnale Chirp utilizzato nelle prove in laboratorio non è esattamente quello descritto ed analizzato nel Capitolo 1. Si è pensato infatti di modificarlo in maniera che occupasse la stessa banda del preambolo descritto dallo standard HomePlug, ovvero utilizzando come frequenza iniziale 4.62 MHz e come frequenza finale 20.9 MHz. Questo perché le frequenze utilizzate dal preambolo HomePlug sono molto

più alte e quindi molto più interessanti da analizzare. Per un confronto ottimale, inoltre, si è pensato di modificarne anche l'altezza in maniera che potesse avere anche la stessa energia del segnale HomePlug.

Poiché la frequenza di lavoro del generatore di funzioni è pari a 100 MHz, il segnale è stato interpolato in maniera tale da avere la stessa frequenza di campionamento. In Figura 3.1 viene mostrato il segnale come è stato trasmesso, con l'andamento in frequenza (per frequenze positive) e la sua autocorrelazione.

In Figura 3.2 si può osservare come tale segnale è stato ricevuto dall'oscilloscopio, mentre in Figura 3.3 viene riportato il segnale ricevuto nel caso di cavo di connessione più lungo. In entrambi i casi viene riportato anche l'andamento in frequenza del segnale e la sua correlazione con il segnale trasmesso.

Nel caso di cavo BNC lungo 75 metri si può notare il ritardo temporale e un'attenuazione maggiore a frequenze elevate.

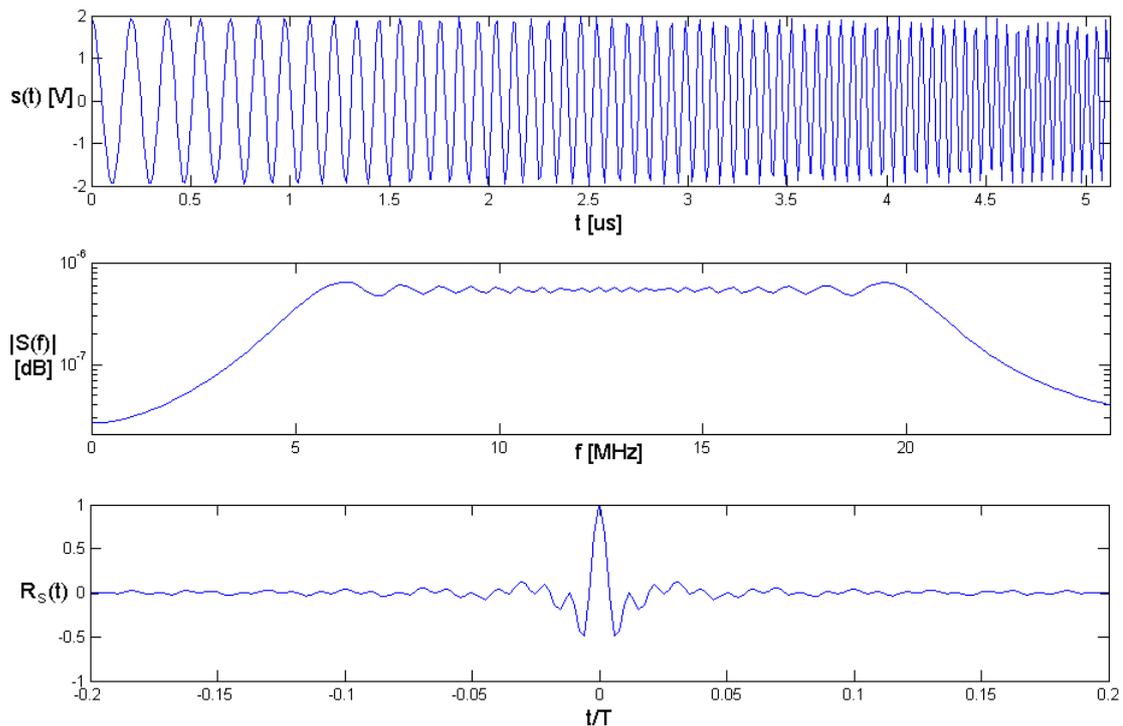


Figura 3.1: Segnale Chirp trasmesso

3.1 Chirp

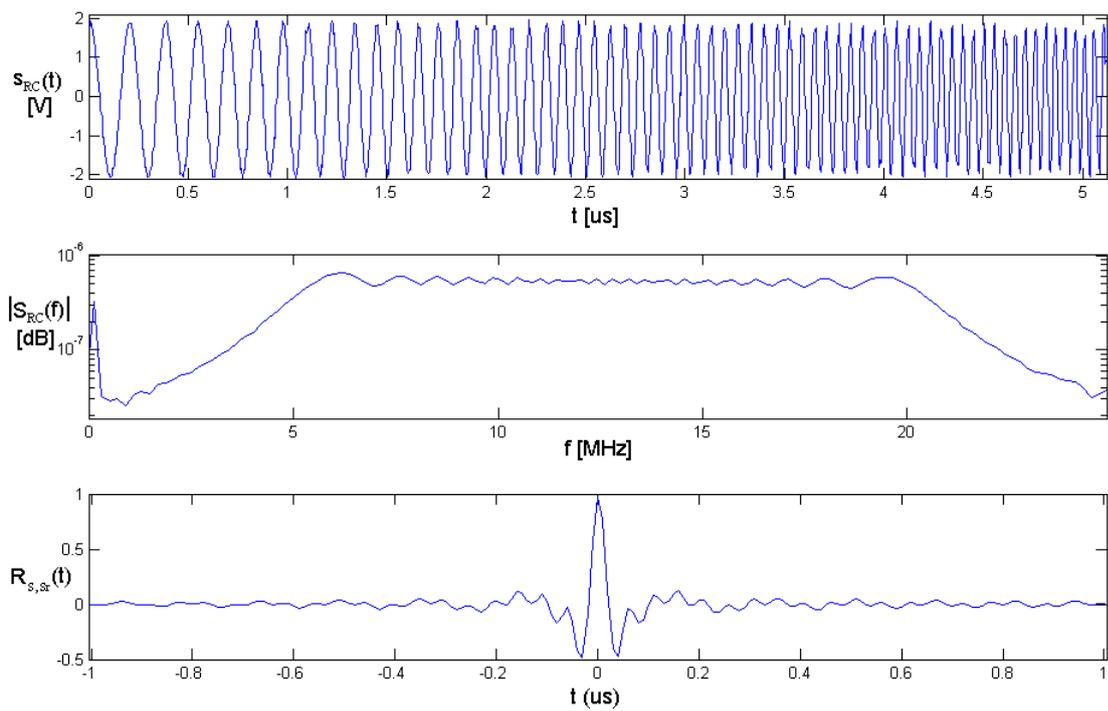


Figura 3.2: Segnale Chirp ricevuto

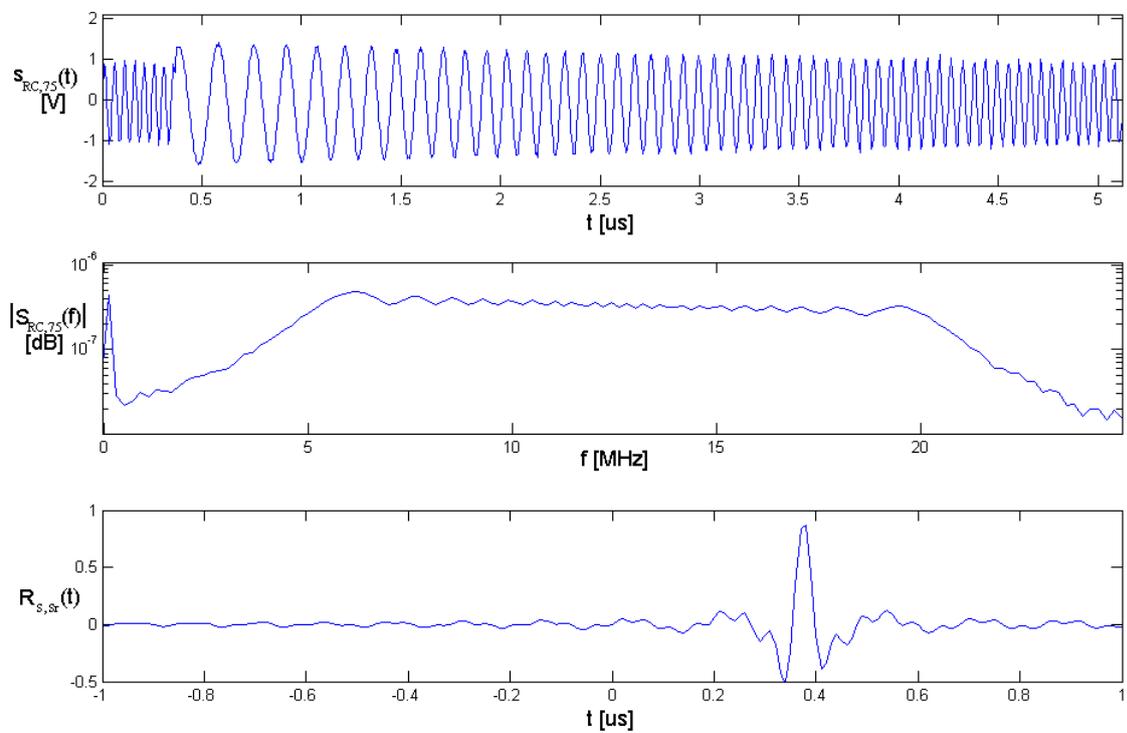


Figura 3.3: Segnale Chirp ricevuto da cavo BNC lungo 75 metri

3.2 Preabolo HomePlug

Anche in questo caso prima di essere trasmesso questo segnale è stato interpolato in maniera da raggiungere una frequenza di campionamento di 100 MHz (Fig. 3.4).

Di seguito si può osservare come tale segnale è stato ricevuto dall'oscilloscopio nel caso di connessione diretta (Fig. 3.5) e nel caso di cavo lungo (Fig. 3.6). Anche in questo caso si osservino le differenze nel caso di cavo di connessione più lungo e in particolare il ritardo temporale conseguente.

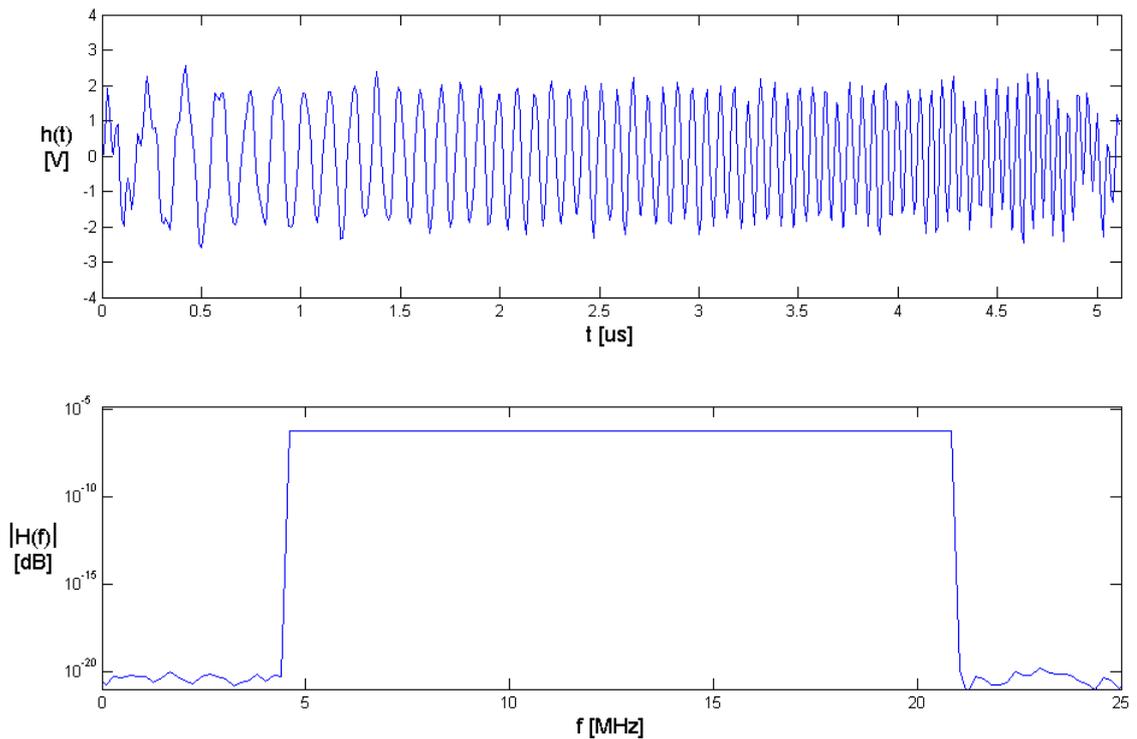


Figura 3.4: Segnale HOMEPLUG trasmesso

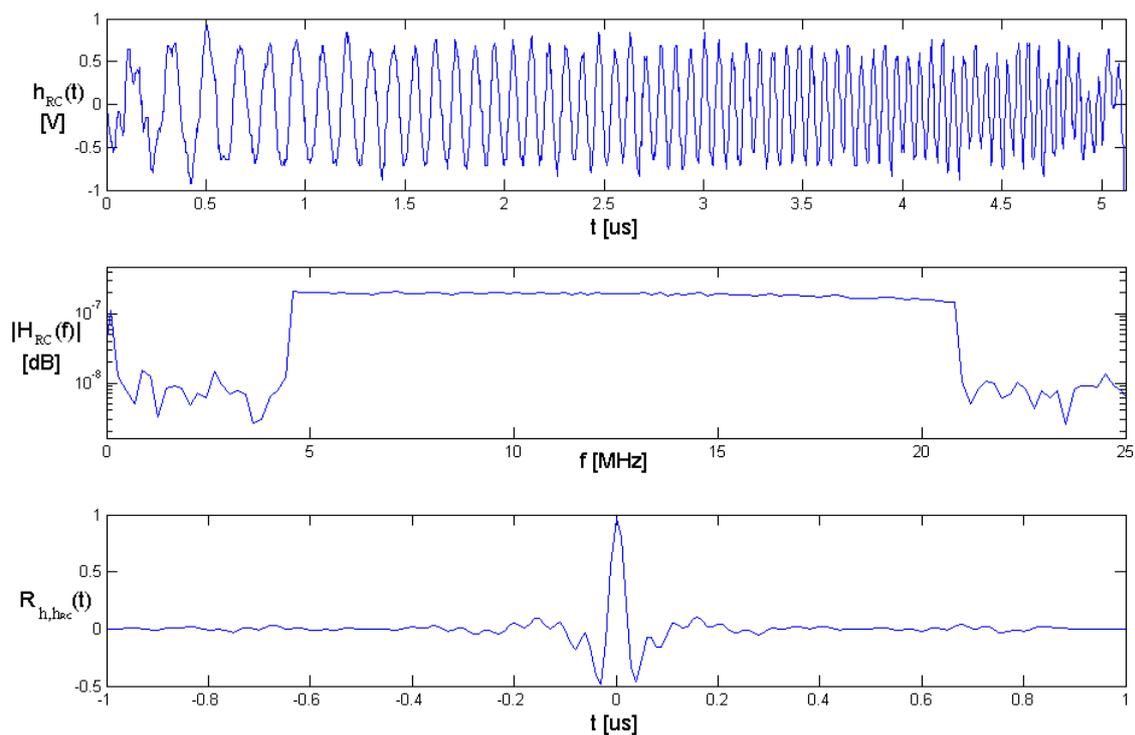


Figura 3.5: Segnale HOMEPLUG ricevuto

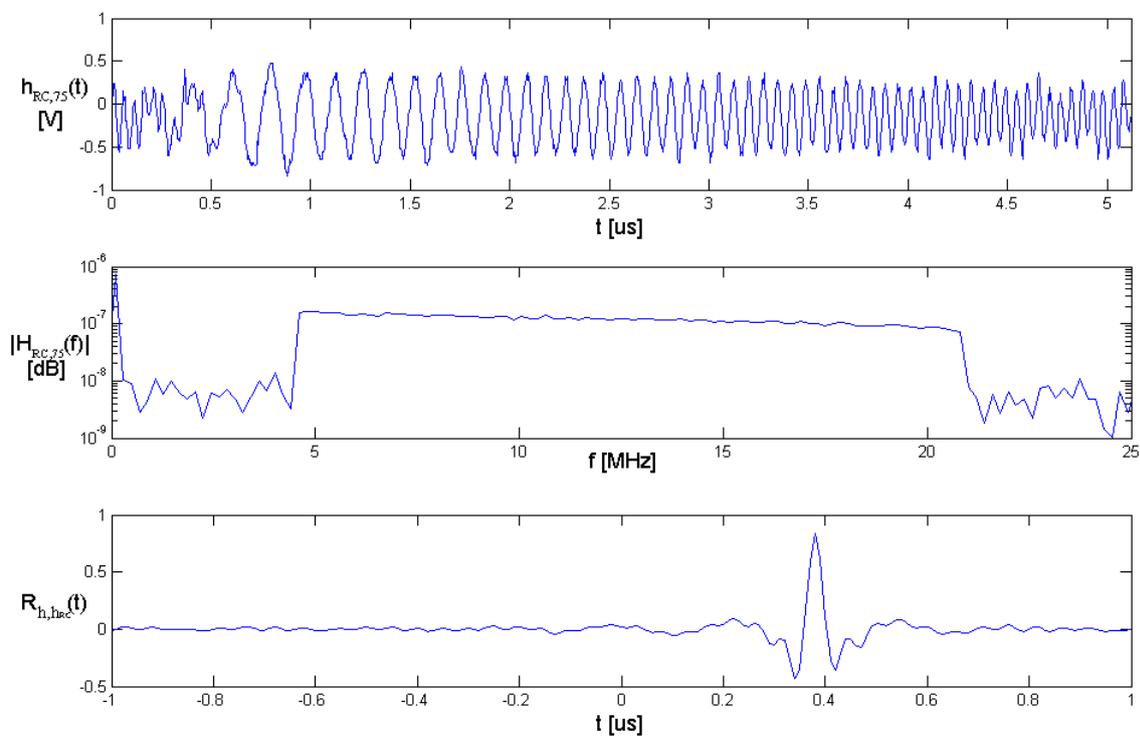


Figura 3.6: Segnale HomePlug ricevuto da cavo BNC lungo 75 metri

3.3 Algoritmo di ricerca a soglia

Le caratteristiche del cavo utilizzato, rintracciabili facilmente su Internet [5], indicano una velocità di propagazione all'interno del cavo pari al 65.9% della velocità della luce e un ritardo di propagazione pari a 1.54 ns/ft, ovvero 5.05 ns/m. Il che implica, su un cavo di 75 metri come nel nostro caso, un ritardo di 378.75 ns. Si può osservare infatti da ciò che è risultato in laboratorio che in entrambe le prove il picco della correlazione tra il segnale trasmesso e quello ricevuto ha un ritardo pari a 380 ns.

Si è provato infine ad implementare tramite Matlab (vedi Appendice) l'algoritmo di ricerca a soglia descritto nel Capitolo 2. È interessante osservare in Figura 3.7 il grafico del segnale $\tilde{q}(t)$ con ampiezza normalizzata tra 0 e 1, sul quale viene compiuta la ricerca della soglia.

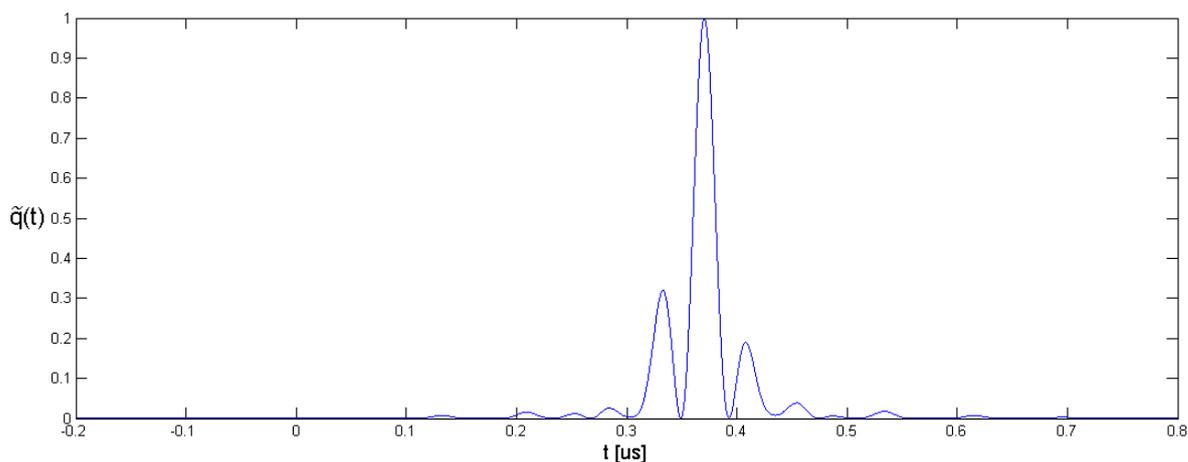


Figura 3.7: Segnale uscente dall'algoritmo di ricerca a soglia

Per ottenere dei risultati ancora migliori dall'algoritmo è stata applicata una correzione temporale per rimuovere l'errore causato dalla larghezza non nulla del picco centrale. Calcolando la durata di tale picco (applicando una prima volta l'algoritmo o semplicemente guardando il grafico) è stato possibile correggere i risultati direttamente sul codice Matlab aggiungendo uno scostamento temporale (10 ns) a quello effettivamente trovato dalla soglia.

In Tabella 3.1 vengono riportati i risultati ottenuti utilizzando i diversi livelli di soglia che sono stati poi graficati in Figura 3.8.

λ	L [m]
0.02	58.41
0.04	65.54
0.06	66.93
0.1	66.33
0.2	67.32
0.3	68.5
0.35	74.05
0.4	74.25
0.45	74.25
0.5	74.45

Tabella 3.1: Risultati dell'algoritmo di ricerca a soglia

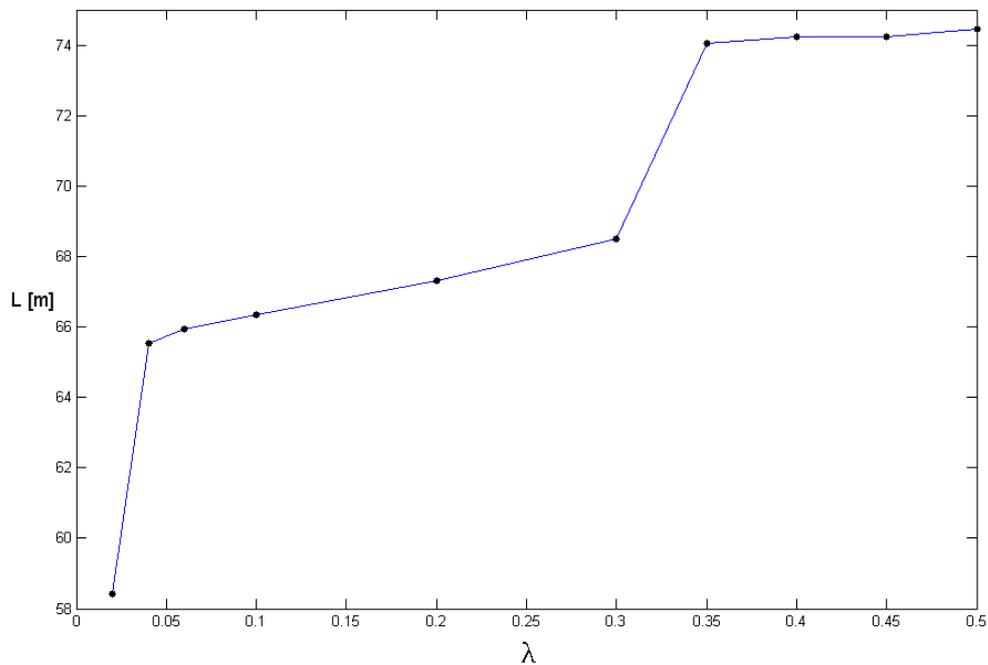


Figura 3.8: Risultati dell'algoritmo di ricerca a soglia

Si nota subito che a partire da $\lambda = 0.35$ si ha un risultato quasi perfetto, per ottenere buoni risultati è quindi necessario che la soglia di decisione sia maggiore o uguale di tale valore.

4 Segnali equivalenti in banda base

In questo capitolo verrà presentato il lavoro svolto per modificare i segnali utilizzati precedentemente portandoli in banda base, per poi applicare su di essi gli stessi calcoli di ranging. Lo scopo di questa trasformazione è quello di poter ottenere gli stessi risultati con un minor utilizzo di memoria. Trasformare i segnali in banda base infatti, come si vedrà meglio nei prossimi paragrafi, permette di mantenere le informazioni necessarie utilizzando una quantità notevolmente minore di campioni dei segnali.

4.1 Idea di base

Anzitutto è necessario capire esattamente che cosa sia un segnale equivalente in banda base. Lo scopo che si vuole raggiungere è, dato un segnale qualsiasi $x(t)$, quello di trovare un segnale $d(t)$ tale che

$$D(f) = X^{(a)}(f + f_m)$$

dove f_m indica la frequenza centrale e

$$X^{(a)}(f) = \begin{cases} 2X(f) & \text{per } f > 0 \\ X(0) & \text{per } f = 0 \\ 0 & \text{per } f < 0 \end{cases}$$

$D(f)$ si ottiene cioè trasportando in banda base la parte delle frequenze positive di $x(t)$. E' facile notare che $d(t)$ mantiene ancora tutte le informazioni del segnale di partenza, è infatti possibile tornare al segnale originale attraverso la relazione

$$x(t) = \text{Re}[e^{j2\pi f_m t} \cdot d(t)]$$

che corrisponde a

$$x(t) = \cos(2\pi f_m t)a(t) - \sin(2\pi f_m t)b(t)$$

dove si è posto $d(t) = a(t) + jb(t)$. Infatti tornando in frequenza si ricava

$$X(f) = \frac{1}{2}[D(f - f_m) + D^*(-(f + f_m))]$$

Ora è quindi necessario andare a calcolare l'equivalente in banda base $d(t)$ per i due segnali con cui stiamo lavorando, descritti nel Capitolo 1.

4.1.1 Calcolo per il Chirp

$$s(t) = \text{Arect}_T \cdot \cos\left[2\pi t\left(f_0 + \frac{1}{2}\mu t\right)\right]$$

Ossevando che

$$f_0 = f_m - \frac{\Delta f}{2}$$

dove $\Delta f = f_1 - f_0$ e utilizzando la proprietà del coseno si ricava

$$\begin{aligned} \cos\left[2\pi t\left(f_m - \frac{\Delta f}{2} + \frac{1}{2}\mu t\right)\right] &= \cos\left[2\pi t f_m + 2\pi t\left(\frac{1}{2}\mu t - \frac{\Delta f}{2}\right)\right] \\ &= \cos(2\pi t f_m) \cos(\pi t \cdot (\mu t - \Delta f)) - \sin(2\pi t f_m) \sin(\pi t \cdot (\mu t - \Delta f)) \\ &= \cos(2\pi t f_m) \cdot a(t) - \sin(2\pi t f_m) \cdot b(t) \end{aligned}$$

da cui ricaviamo quindi che

$$d_s(t) = \begin{cases} e^{j\pi t(\mu t - \Delta f)} & 0 \leq t \leq T \\ 0 & \text{altrove} \end{cases}$$

Nella figura 1.4 è riportata la trasformata di Fourier di $d_s(t)$ campionato a 50 MHz.

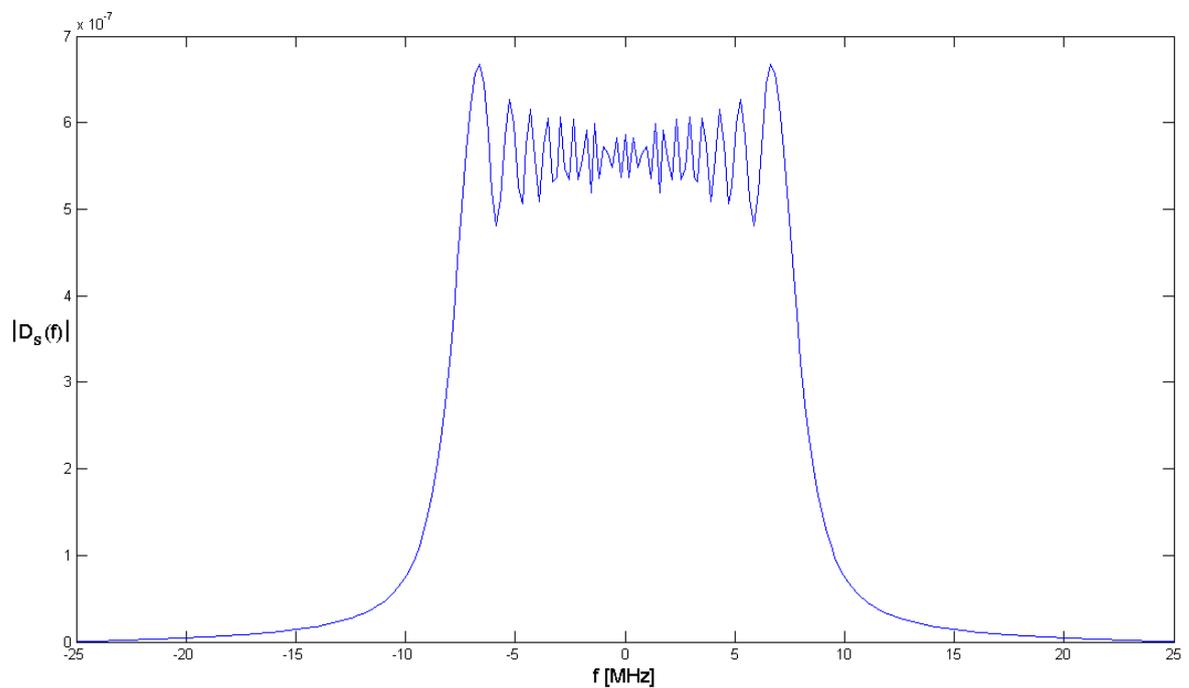


Figura 4.1: Trasformata di Fourier del segnale Chirp in banda base

4.1.2 Calcolo per l'HomePlug

$$h(n) = \sum_{c=0}^{83} \cos(2\pi f_c n + \phi_c)$$

Osservando che

$$f_c = f_m - \frac{\Delta f}{2}$$

dove $\Delta f = f_{83-c} - f_c$ con $0 \leq c \leq 83$, allora

$$\begin{aligned} x(n) &= \sum_{c=0}^{106} \cos(2\pi f_c n + \phi_c) \\ &= \sum_{c=0}^{83} \cos(2\pi (f_m - \frac{\Delta f}{2})n + \phi_c) \\ &= \sum_{c=0}^{83} \cos(2\pi f_m n + (\phi_c - \pi \Delta f \cdot n)) \\ &= \sum_{c=0}^{83} [\cos(2\pi f_m n) \cos(\phi_c - \pi \Delta f \cdot n) - \sin(2\pi f_m n) \sin(\phi_c - \pi \Delta f \cdot n)] \\ &= \cos(2\pi f_m n) \sum_{c=0}^{83} \cos(\phi_c - \pi \Delta f \cdot n) - \sin(2\pi f_m n) \sum_{c=0}^{83} \sin(\phi_c - \pi \Delta f \cdot n) \\ &= \cos(2\pi f_m n) \cdot a(n) - \sin(2\pi f_m n) \cdot b(n) \end{aligned}$$

e, quindi, in questo caso

$$d_h(t) = \begin{cases} \sum_{c=0}^{83} e^{j(\phi_c - \pi \Delta f \cdot n)} & 0 \leq n \leq 256 \\ 0 & \text{altrimenti} \end{cases}$$

In figura 4.2 è riportata la trasformata di Fourier di $d_h(t)$ campionato a 50MHz.

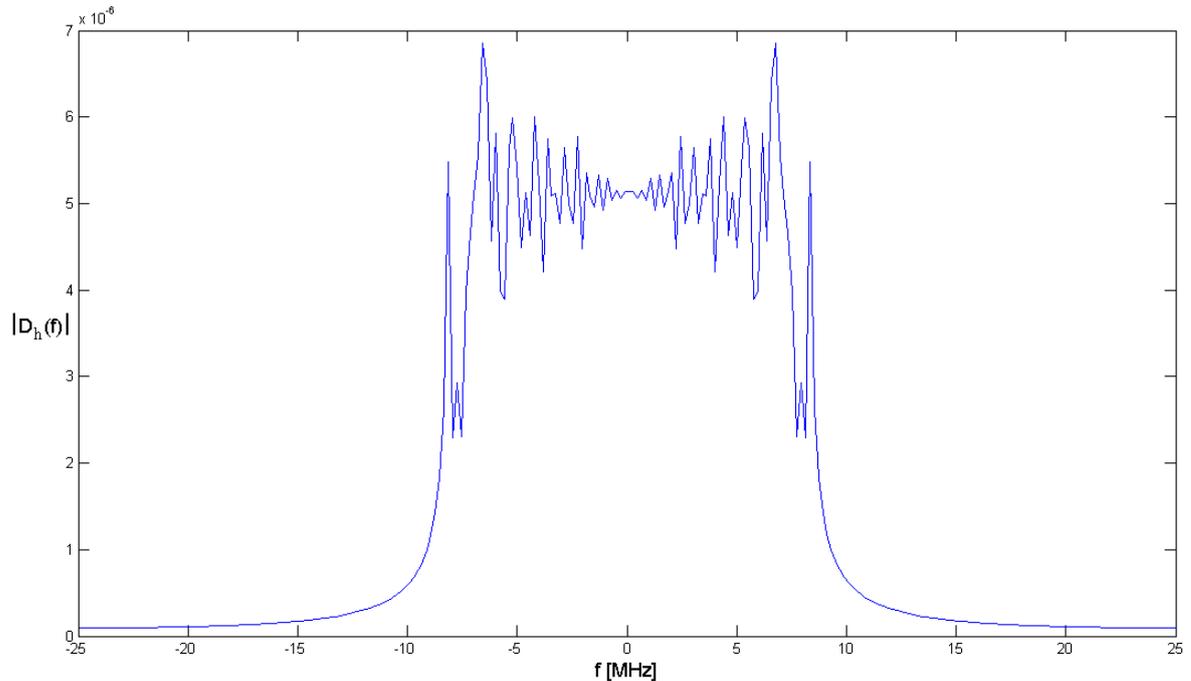


Figura 4.2: Trasformata di Fourier del segnale HomePlug in banda base

4.2 Equivalente in banda base dei segnali ricevuti

I segnali che sono stati salvati dalle misure con l'oscilloscopio hanno una frequenza di campionamento di 250 MHz, dovuta alle caratteristiche di funzionamento dello strumento stesso. Un tale valore è decisamente superiore a quanto sia necessario, considerando che i segnali sono stati inizialmente descritti con una larghezza di banda di circa 20 MHz e con una frequenza di campionamento di 50 MHz. Per di più, trasportando tali segnali in banda base, la loro larghezza di banda si dimezza; è evidente quindi che mantenere tutti i dati raccolti significa immagazzinare elementi superflui ed inutili che vanno ad appesantire la memoria e prolungare i tempi necessari per i calcoli.

Tramite Matlab, quindi, (vedi Appendice) sono stati calcolati gli equivalenti in banda base dei due segnali ricevuti tramite il cavo BNC lungo 75 metri e sono stati campionati mantenendo un valore ogni 10, portando quindi il segnale ad avere una frequenza di campionamento di 25 MHz senza perdere alcuna informazione (Fig. 4.3 per il Chirp e Fig. 4.5 per l'HomePlug). Questi sono stati poi correlati con i relativi segnali calcolati nel paragrafo precedente (creati sempre ad frequenza di campionamento di 25 MHz) per verificare che effettivamente non siano state perse informazioni a fronte di una quantità di dati 10 volte minore (Fig. 4.4 per il Chirp e 4.6 per l'HomePlug).

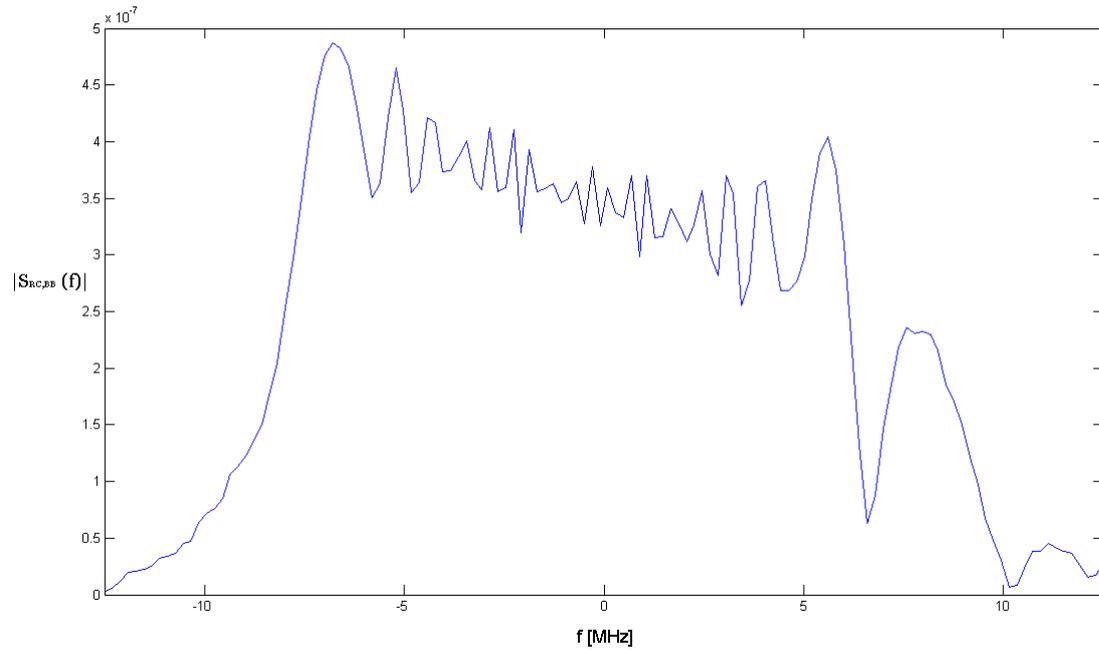


Figura 4.3: Equivalente in banda base del segnale Chirp ricevuto

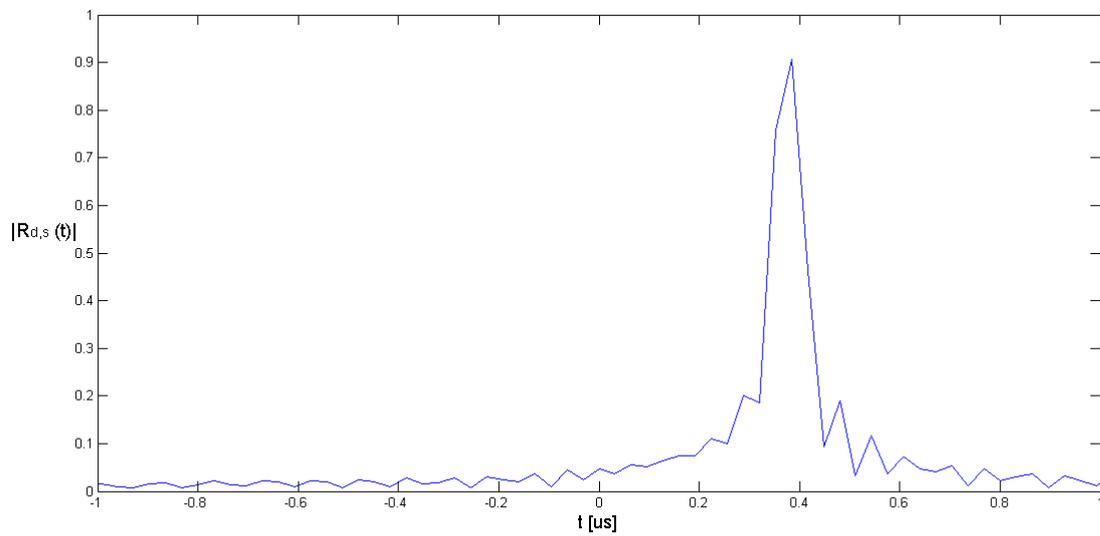


Figura 4.4: Correlazione tra gli equivalenti in banda base del segnale Chirp trasmesso e ricevuto

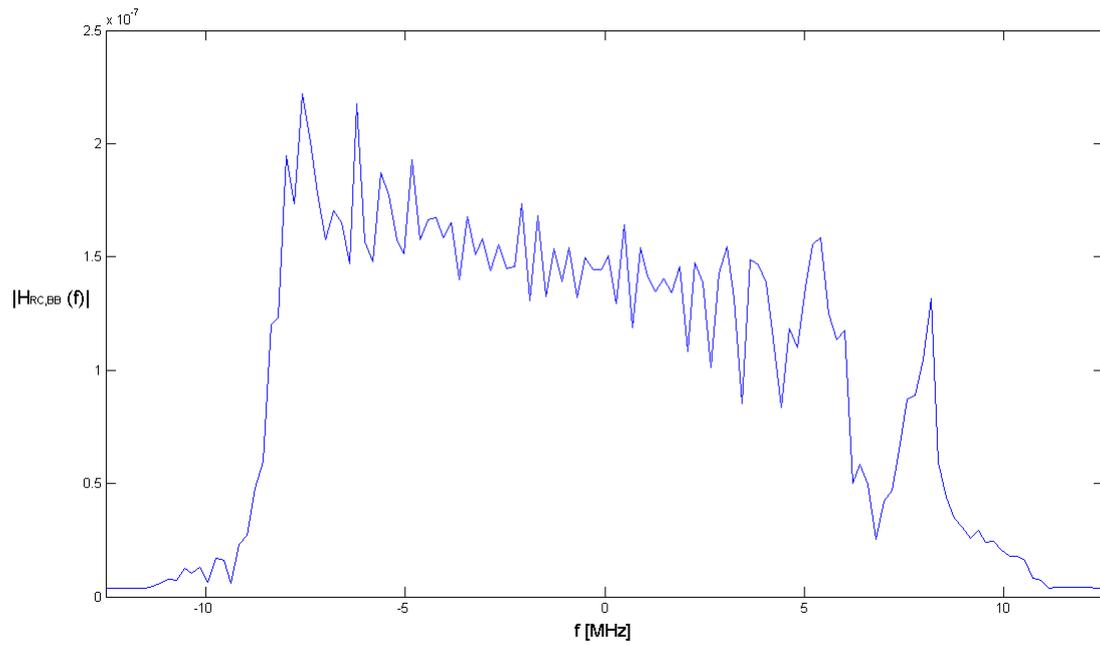


Figura 4.5: Equivalente in banda base del segnale HomePlug ricevuto

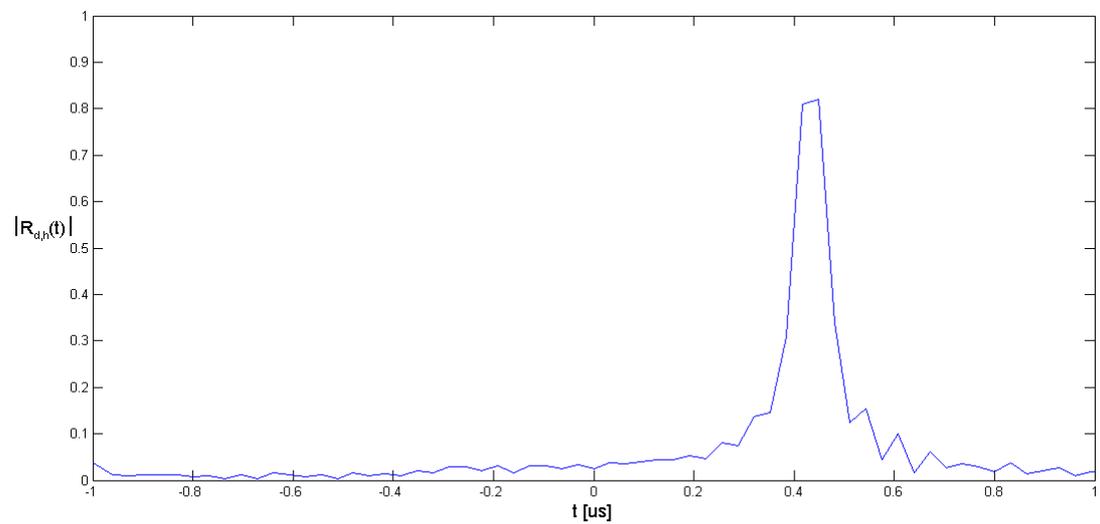


Figura 4.6: Correlazione tra gli equivalenti in banda base del segnale HomePlug trasmesso e ricevuto

4.3 Algoritmo di ricerca a soglia

È importante a questo punto andare a controllare quali valori di lunghezze si ottengono applicando l'algoritmo di ricerca a soglia. Per poter avere un confronto col caso precedente sono state utilizzate le stesse soglie viste nel Paragrafo 3.3. Anche in questo caso è stata implementata una correzione temporale, in questo caso di 18 ns, che permette di migliorare i risultati pervenuti.

λ	L [m]
0.02	59
0.04	59
0.06	59
0.1	74.84
0.2	74.84
0.3	74.84
0.35	74.84
0.4	74.84
0.45	74.84
0.5	74.84

Tabella 4.1: Risultati dell'algoritmo di ricerca a soglia con i segnali in banda base

Possiamo notare che, a partire dal valore di soglia 0.1, si hanno ottimi risultati, anche migliori di quanto si sia ottenuto con i segnali in banda passante.

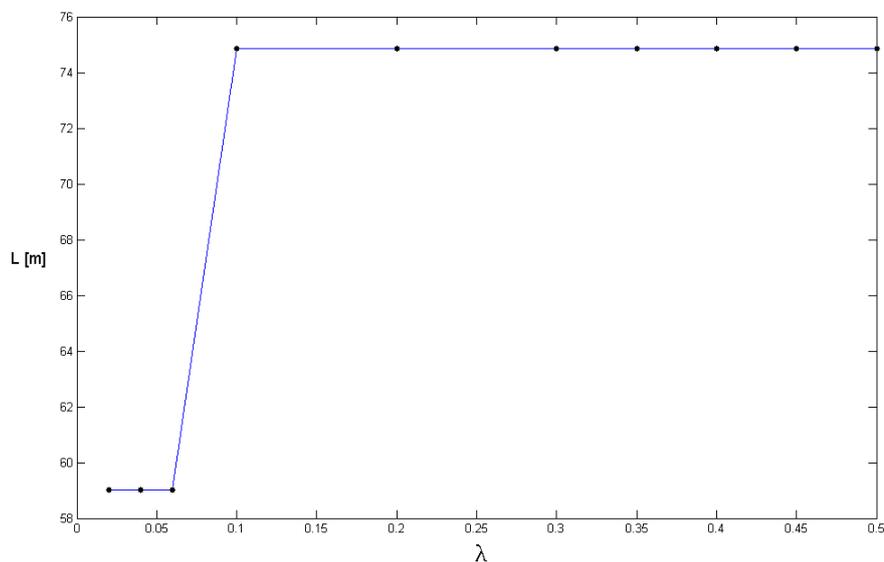


Figura 4.7: Risultati dell'algoritmo di ricerca a soglia con i segnali in banda base

Appendice: codici Matlab utilizzati

Qui di seguito vengono riportati i codici utilizzati in Matlab per la creazione e lo studio delle diverse funzioni.

Per primi sono stati trascritti i codici che creano i segnali Chirp e HomePlug aventi la stessa banda e la stessa energia. Subito dopo si trova il codice per copiare i dati dei campioni dei segnali in un file in formato .csv (comma-separated values). Questa conversione è necessaria per l'acquisizione dei valori col generatore di funzioni. Il codice successivo è quello utilizzato per trasformare il segnale acquisito in laboratorio in un segnale in banda base (viene riportato il caso dell'HomePlug, ma nel caso del Chirp il procedimento è identico). Infine è possibile trovare l'algoritmo per il calcolo della distanza usando la ricerca a soglia.

Preambolo HomePlug

Il codice che segue crea il segnale HomePlug a partire dall'elenco delle fasi definite nella tabella 1.1

```
pc = [0 0 5.8905 5.8905 5.4978 5.1051 4.7124 4.3197 3.5343 2.7489 ...
2.3562 1.1781 0.3927 5.8905 4.7124 3.5343 2.3562 1.1781 5.8905 ...
4.7124 3.1416 1.5708 0 4.3197 2.7489 0.7854 5.1051 3.1416 1.1781 ...
5.1051 2.7489 0.7854 4.3197 1.9635 5.8905 3.1416 0.3927 3.9270 ...
1.1781 4.3197 1.5708 4.7124 1.5708 4.7124 1.1781 4.3197 0.7854 ...
3.5343 0 2.7489 5.1051 1.1781 3.9270 5.8905 1.9635 4.3197 0 ...
1.9635 3.9270 5.8905 1.1781 3.1416 4.7124 0 1.5708 2.7489 4.3197 ...
5.4978 0.3927 1.5708 2.7489 3.5343 4.3197 5.4978 5.8905 0.3927 ...
1.1781 1.5708 1.9635 2.3562 2.7489 2.7489 3.1416 3.1416];
T = 5.12*(10^-6);
shp = zeros(1, 256); % Crea la matrice dove verrà salvata la funzione
fc = (23:1:106)/256; % Definisce le frequenze (c+23)/256
for n = 1:1:256
    a = cos(2*pi*fc*n + pc);
    b = 2/sqrt(84)*sum(a(:)); % Esegue la sommatoria
    shp(1, n) = b; % Salva il risultato nella funzione finale
end;

%Interpolazione per portare il segnale a 100 MHz
shp = [shp,shp,shp];
shp = interp(shp,2,10,1);
```

```
shp = shp(513:1024);
```

Chirp

Qui di seguito viene creato il segnale Chirp con le stesse frequenze e la stessa energia che caratterizzano il segnale HomePlug.

```
Tc1 = 10*(10^-9); %100MHz
T = 5.12*(10^-6);
t = 0:Tc1:T-Tc1;
f0 = 4.6169*10^6;
f1 = 2.0923*10^7;
m = (f1-f0)/T;
A = 1;
sch = A*cos(2*pi*(f0*t+0.5*m*(t.^2)));

% Calcola il Chirp con la stessa energia del segnale HomePlug
E = Tc1*sum(abs(sch).^2);
E1 = Tc1*sum(abs(shp).^2); % hp è definita nel codice precedente
s2 = sqrt(E1/E)*sch;
```

Conversione in .csv

Viene qui riportato il codice utilizzato per trasportare i dati dei segnali in un file con estensione .csv, nel caso di segnale HomePlug. La funzione 'cell2csv' trascritta è stata trovata su Internet [6], nel commento si trova il nome dell'autore. Una volta compiuta questa conversione il file può essere aperto dal programma ArbExpress, il quale può dialogare col generatore di funzione dopo essere stato con esso sincronizzato tramite l'apposito pulsante 'connect' nel menù 'File Transfer & Control'. Dopo aver fatto ciò il segnale può essere trasmesso cliccando semplicemente su 'send to Arb'.

```
% Si crea un cell array dove viene riportata la frequenza di clock,
% il numero di campioni del segnale e di seguito i campioni stessi
% incolonnati. La funzione cell2csv trasforma questo cell array
% in un file .csv che può essere acquisito dal generatore di funzioni.

clock_string = ['#CLOCK=1.0000000000e+008'];
size_string = ['#SIZE=512'];
clock_cell = cellstr(clock_string); % cellstr trasforma una stringa
size_cell = cellstr(size_string); % in cell array
hdr = [clock_cell size_cell];
data = num2cell(shp); % num2cell trasforma un numero in cell array
data = data';
```

```
fine = [hdr data];
fine = fine';
cell2csv('hp_csv.csv',fine,',');

function cell2csv(filename,cellArray,delimiter)
% Converte un cell Array in un file .csv
% by Sylvain Fiedler, KA, 2004

if nargin<3
    delimiter = ',';
end

datei = fopen(filename,'w');
for z=1:size(cellArray,1)
    for s=1:size(cellArray,2)
        var = eval(['cellArray{z,s}']);
        if size(var,1) == 0
            var = '';
        end
        if isnumeric(var) == 1
            var = num2str(var, 14);
        end
        fprintf(datei,var);
        if s ~= size(cellArray,2)
            fprintf(datei,[delimiter]);
        end
    end
    fprintf(datei,'\n');
end
fclose(datei);
end
```

Una volta che il segnale è in trasmissione esso può essere esaminato dall'oscilloscopio. Dallo strumento si possono salvare i dati acquisiti tramite 'Save Wfm to File' nella finestra che si apre selezionando 'Reference Waveforms', quindi 'Save Wfm' nel menu File. L'onda può essere salvata in formato .dat, acquisibile senza problemi da Matlab.

Equivalente in banda base del segnale ricevuto

Il codice sotto riportato descrive la procedura per trasformare il segnale ricevuto HomePlug. Si tratta comunque di una procedura generica che può essere applicata a qualsiasi funzione.

```

Tc = 4*(10^-9); %250 MHz
Tc2 = 40*(10^-9); %25 MHz
T = 5.12*10^(-6);
f0 = 4.6169*10^6;
f1 = 2.0923*10^7;
fm = (f0+f1)/2;

% Crea il segnale HomePlug equivalente in banda base
load('pc.mat'); % pc.mat contiene l'elenco delle fasi del segnale HomePlug,
                % per l'elenco completo si veda il codice 'Preambolo HomePlug'
fun = zeros(1, 256); % Crea la funzione finale, per ora vuota
fcr = (23:1:106)/256; % Frequenze crescenti
fdc = (106:-1:23)/256; % Frequenze decrescenti
for n=1:1:256
    e = exp(1i*(pc - pi*n*(fdc-fcr))); % Esegue la Sommatoria
    a = sum(e(:));
    fun(1, n) = a;
end;
d = a(1:2:length(fun)); % Ora è campionato a 25 MHz

%Carica l'HP, ricava l'equivalente in banda base e lo campiona a 25 MHz
load hp_lungo.dat; % 6 periodi del segnale HP ricevuto
hp_lungo1p = hp_lungo(1:1280); % Prende un periodo
t = 0:Tc:T-Tc;
hp_shift = (hp_lungo1p').*(exp(-1j*(2*pi*fm*t))); % Spostamento in frequenza
t_ircos = -(T-Tc)/2:Tc:(T-Tc)/2;
h = conv(hp_shift,ircos(t_ircos*21*10^6,0.1)); % Filtro in frequenza
h = h(640:1919);
hBB = h(1:10:length(es)); % Campionamento a 25 MHz

% Correlazione tra i due segnali in banda base
x = xcorr(hBB,d,'coeff');

```

Algoritmo di ricerca a soglia

%Calcolo della lunghezza tramite algoritmo a soglia nel caso di segnali
 %in banda passante e di segnali in banda base

```

Tc = 10*10^6; %100 MHz
Tc2 = 40*(10^-9); %25 MHz
load c_lungo.dat % 6 periodi del segnale HP ricevuto tramite cavo lungo
chirp_lungo1p = c_lungo(1:1280); % Prende un periodo
e = chirp_lungo1p(1:5:1280); % Porta il segnale da 250 MHz a 100 MHz,
e2 = interp(e,2); % come quello trasmesso

```

```
corr = xcorr(e2,s2,'coeff'); % s2 è il chirp in banda ed energia dell'HP
t = -(T-Tc):Tc:(T-Tc);
L = find_dist(corr,t,0.4)

% Calcola la correlazione tra i segnali in banda base
% definiti nel codice precedente.
x = xcorr(hBB,d,'coeff');
tx = -(T-Tc2):Tc2:T-Tc2;
L2 = find_dist(x,tx,0.4)

% funzione per il calcolo della distanza
function d = find_dist(fun, t, alpha)
v = 3*10^8*0.66; % Velocità di propagazione nel cavo
correction = 10*10^-9; %Correzione temporale: è 18 ns nel caso in BB
f = abs(fun);
f2 = f.^2;
f3 = (f2 - min(f2))/(max(f2)-min(f2));
l = find(f3>alpha); % Trova i valori dei segnali maggiori della soglia
Ts = t(l(1))+correction; %Trova il tempo corrispondente al primo valore
d = v*Ts; % Calcola la distanza
end
```


Bibliografia

- [1] “Smart grids, stato dell’arte e principali problematiche”, Francesco Grasso
- [2] “Tecniche di Ranging su sistemi wireless UWB e wired PLC”, tesi di laurea di Francesco Lorenzon
- [3] “Particle Swarm Optimization per lo Shaping del segnale radar meteorologico”, Alessio Gemitto
- [4] “Distance measurement over PLC for dynamic grid mapping of smart micro grids”, T.Erseghe, F.Lorezon, S.Tomasin, A.Costabeber, P.Tenti
- [5] <http://www.tech-faq.com/rg-58.html>
- [6] <http://www.mathworks.com/matlabcentral/fileexchange/4400-cell-array-to-csv-file-cell2csv-m>