

Corso di Laurea in Ingegneria dell'Informazione

Ranging per Smart Grid

Paolo Rossignoli

29 Novembre 2011

Relatore: Prof. Tomaso Erseghe

Indice

1	Introduzione	3
1.1	Progetto PRIME	4
1.1.1	Trasformata di Fourier	5
1.1.2	Autocorrelazione	6
1.2	HomePlug	8
1.2.1	Trasformata di Fourier	8
1.2.2	Autocorrelazione	11
2	Analisi Algoritmi	12
2.1	Course & Fine Estimators	12
2.1.1	Coarse Estimator	14
2.1.2	Fine estimator	15
2.2	Kurtosis Analysis	16
2.2.1	Selezione della soglia basata sull'analisi di Kurtosis	17
3	Simulazioni in laboratorio	18
3.1	Chirp	18
3.1.1	Analisi stima con Coarse Estimator	21
3.1.2	Analisi stima con kurtosis	22
3.2	Preambolo HomePlug	23
3.2.1	Analisi stima con Coarse Estimator	25
3.2.2	Analisi stima con kurtosis	26
3.2.3	Conclusioni	27
4	Appendice: codici Matlab utilizzati	28
	Riferimenti bibliografici	33

1 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 mostra i due segnali preambolo utilizzati per questa tesi, il Chirp e l'HomePlug; nel capitolo 2 vengono spiegati i due algoritmi utilizzati per le misure di ranging: il Coarse Estimator e il kurtosis; il capitolo 3 mostra una simulazione del ranging utilizzando i segnali del capitolo 1 e gli algoritmi del capitolo 2; infine nel capitolo 4 si possono analizzare i codici sviluppati tramite Matlab, utilizzati per ottenere i risultati nel capitolo 3.

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. Si è scelto di utilizzare tali segnali di preambolo, ovvero i segnali che si trovano all'inizio di ogni pacchetto trasmesso, perchè 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 studiandone 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

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

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

$$\text{rect}(t) = \begin{cases} 1 & \text{per } -\frac{1}{2} \leq t \leq \frac{1}{2} \\ 0 & \text{altrove} \end{cases}$$

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

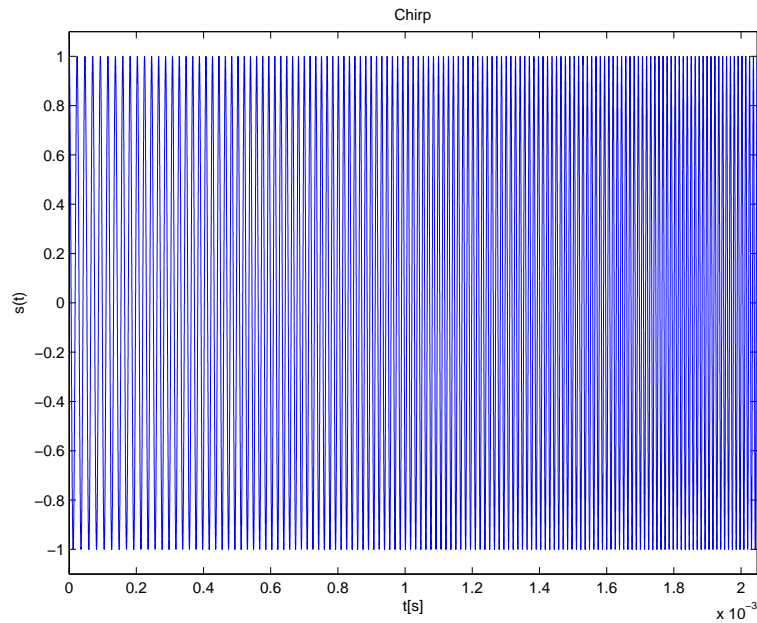


Figura 1: Segnale Chirp $s(t)$

1.1.1 Trasformata di Fourier

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.

Il calcolo per ricavare la trasformata di Fourier di un segnale Chirp è molto conosciuto e vengono riportati i calcoli ritrovati in letteratura [2].

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 $a = t\sqrt{\frac{\mu}{2}}$ e $b = \pi f\sqrt{2/\mu}$, ovvero

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

Si può dunque scrivere la trasformata come

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

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

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

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

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

$$S(f) = \left(\frac{\pi}{\mu}\right)^{1/2} e^{-j\frac{(2\pi f)^2}{2\mu}} C(X_2) + C(X_1) + j[S(X_2) + S(X_1)]$$

L'andamento è illustrato in Figura 2.

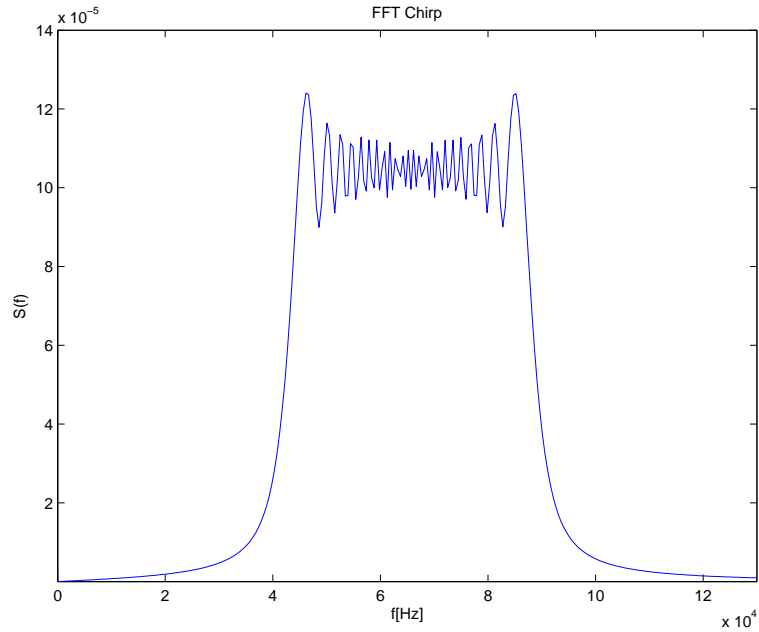


Figura 2: Trasformata di Fourier Chirp $S(f)$

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

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

Quindi, applicando la formula di Eulero, si ottiene

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

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

$$R_s(t) = \frac{1}{\pi\mu t} \text{sen}[\pi\mu t(T - |t|)] = (T - |t|) \text{sinc}[\mu t(T - |t|)]$$

In Figura 3 viene illustrato l'andamento dell'autocorrelazione.

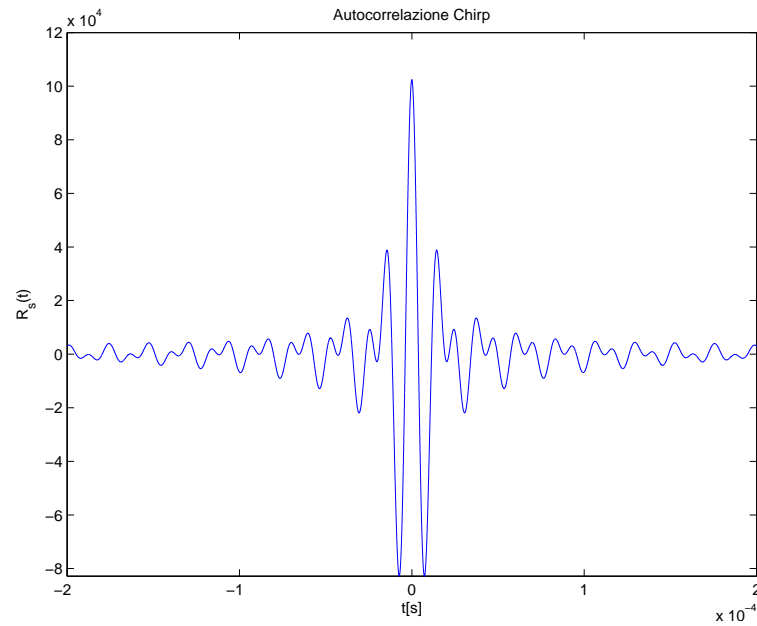


Figura 3: Autocorrelazione del segnale Chirp

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 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 SYNCNP in quanto vengono moltiplicati per 1 dalla funzione segno, gli ultimi SYNCN perchè vengono moltiplicati per -1 dalla stessa funzione.

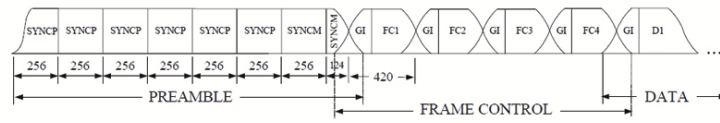


Figura 4: Struttura di un frame secondo le specifiche HomePlug

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

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

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

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

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

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

I valori di ϕ che formano il preambolo vengono riportati in Tabella 1.

In Figura 5 viene riportato l'andamento di uno dei simboli centrali.

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 riscivo la forma d'onda di un simbolo centrale del preambolo come

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

Tabella 1: La funzione ϕ_c

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	$8(\pi/8)$

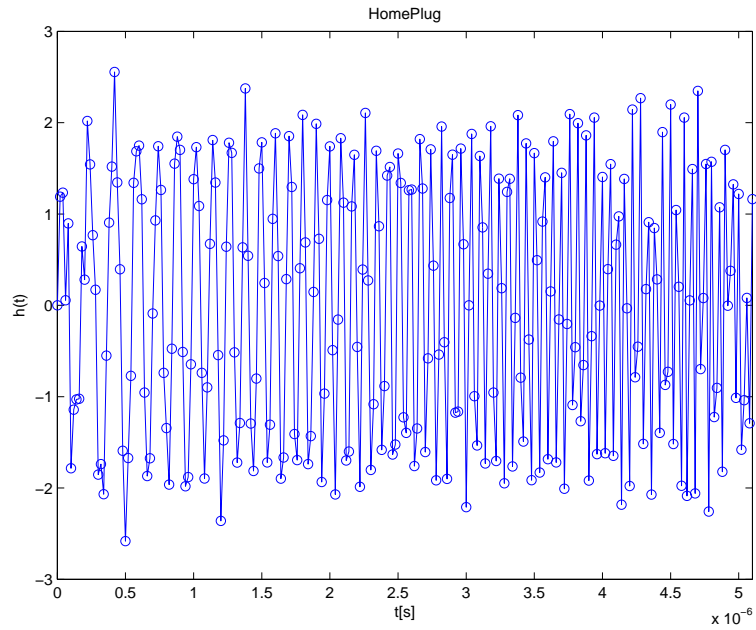


Figura 5: Segnale HomePlug $h(nT)$

la cui trasformata (Figura 6) è

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

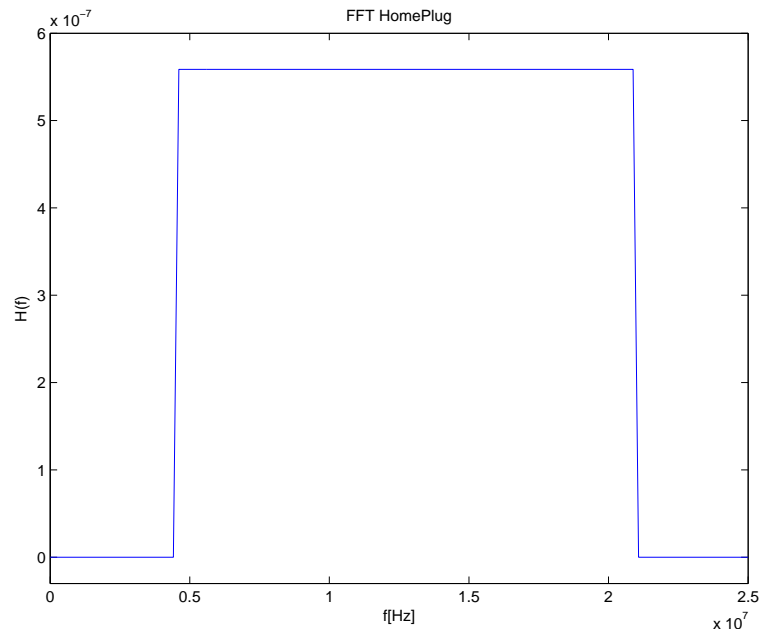


Figura 6: Trasformata di Fourier del segnale HomePlug

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 7.

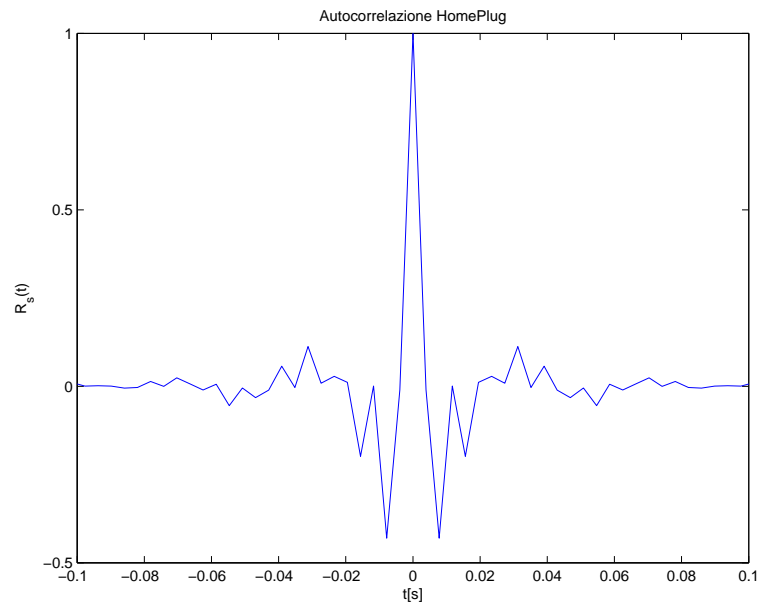


Figura 7: Autocorrelazione del segnale HomePlug

2 Analisi Algoritmi

2.1 Course & Fine Estimators

Questo tipo di stimatore permette il calcolo del ranging tramite TOA ed è basato sulla tecnica dei minimi quadrati (LS). Possiamo trattare 2 tipi di stimatori:

1. Coarse Estimator: stimatore standard che useremo per i nostri esperimenti.
2. Fine Estimator: permette una riduzione dell'errore, ma non lo useremo perchè viene usato quando nella rete ci sono L percorsi.

Il modello del segnale si può modellare nel seguente modo:

$$s(t) = a_k w(t) = w(t)$$

dove abbiamo assunto che $a_k = 1$, $w(t)$ è un impulso di durata T_p (in Figura 8 si ha $T_p = 14$).

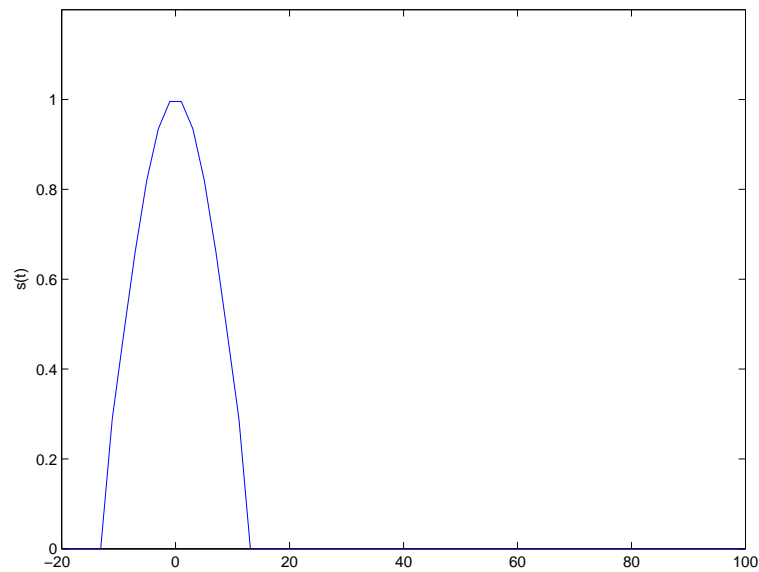


Figura 8: Esempio segnale

In generale il segnale viene trasmesso in un canale a L percorsi. Consideriamo che il segnale che vogliamo ricevere sta sul percorso L_i : per determinare il segnale ricevuto è necessario considerare tutti i percorsi perchè il segnale subisce le riflessioni degli altri percorsi. Questi segnali riflessi risultano comunque attenuati rispetto al segnale che si vuole stimare. Quindi la risposta all'impulso $w(t)$ nel nodo ricevente si può scrivere come:

$$r(t) = \sum_{l=0}^{L-1} w_l(t - \tau_l) + n(t)$$

dove con $n(t)$ è il rumore.

Il percorso dell'impulso lungo il canale che va dal trasmettitore al ricevitore è il più breve, quindi il primo impulso che si presenta al ricevitore è quello che vogliamo stimare. Chiamiamo τ il tempo che vogliamo stimare nel ricevitore, il più corto ritardo tra gli impulsi ricevuti. La seguente Figura 9 illustra il segnale ricevuto nel caso di singolo percorso.

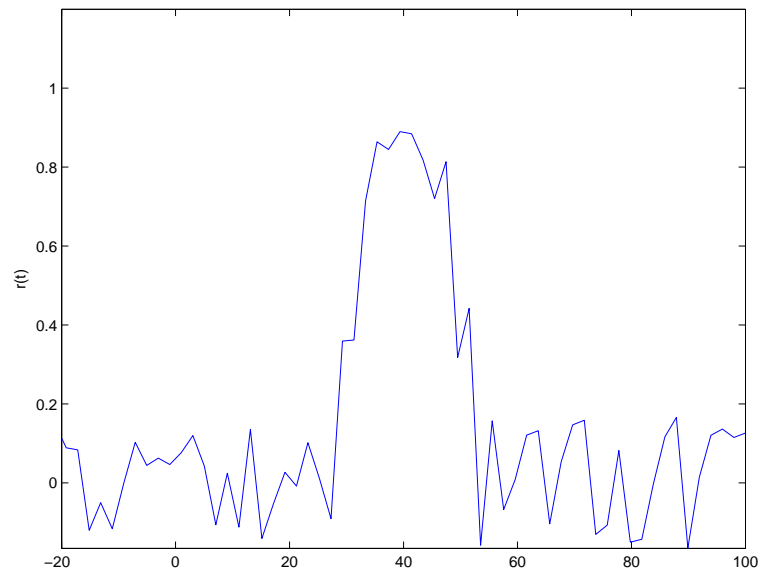


Figura 9: Esempio segnale ricevuto

2.1.1 Coarse Estimator

Ora che abbiamo modellato $s(t)$ (segnale inviato) e $r(t) = s(t-\tau) + n(t)$ (segnale ricevuto con ritardo e con rumore), cerchiamo di stimare τ con la tecnica dei minimi quadrati in questo modo, cioè vogliamo minimizzare J :

$$J(\tau) = \int_0^{T_b} [As(t-\tau) - r(t)]^2 dt$$

dove con T_b si intende un intervallo di tempo che contiene entrambi i segnali. Il termine A è un fattore di scala, necessario in quanto il segnale ricevuto sarà attenuato rispetto al segnale inviato; definisco quindi $h(t) = As(t)$. In altre parole, si esegue una differenza con nostro modello creato della risposta al canale $r(t)$ con il segnale inviato $s(t)$ traslato e scalato di un fattore A , elevando poi al quadrato e successivamente calcolando l'area risultante tramite integrazione (Figura 10).

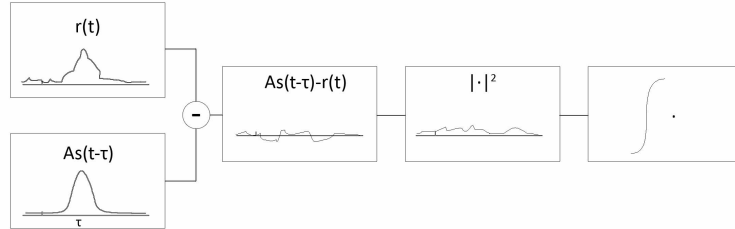


Figura 10: Algoritmo Coarse Estimator

Minimizzare J significa trovare la migliore sovrapposizione tra il modello del segnale e la sua realizzazione, stimando quindi τ . Procediamo sviluppando J , in modo da trovare una formula per la stima di $\hat{\tau}$:

$$\begin{aligned} J(\tau) &= \int_0^{T_b} [h(t-\tau) - r(t)]^2 dt = \int_0^{T_b} h(t-\tau)^2 + r(t)^2 - 2h(t-\tau)r(t) dt \\ &= \int_0^{T_b} h(t-\tau)^2 dt + \int_0^{T_b} r(t)^2 dt - 2 \int_0^{T_b} [h(t-\tau)r(t)] dt \end{aligned}$$

in cui si nota nell'ultimo termine la correlazione tra $h(t)$ e $r(t)$. Minimizzare J significa massimizzare la correlazione, ma bisogna anche tenere conto del termine $h(t-\tau)$ perchè dipende dal parametro A da stimare, cioè:

$$\tau = \arg \max_{\tau} \int_0^{T_b} [h(t-\tau)r(t)] dt - \int_0^{T_b} [h(t-\tau)]^2 dt$$

Considerando che:

$$(a - b)^2 = a^2 - 2ab - b^2$$

cioè

$$2ab - a^2 = b^2 - (a - b)^2$$

possiamo riscrivere i due termini interessati come:

$$2 \int_0^{T_b} [h(t - \tau)r(t)]dt - \int_0^{T_b} h(t - \tau)^2 dt = \int_0^{T_b} r(t)^2 dt - \int_0^{T_b} [h(t - \tau) - r(t)]^2 dt$$

e quindi la stima di τ diventa:

$$\tau = \arg \min_{\tau} \int_0^{T_b} [h(t - \tau) - r(t)]^2 dt - \int_0^{T_b} r(t)^2 dt$$

Se $h(t)$ esiste tra $[-T_h, T_h]$, allora $h(t - \tau)$ esiste tra $[\tau - T_h, \tau + T_h]$ e possiamo riscrivere:

$$\begin{aligned} \tau &= \arg \min_{\tau} \int_{\tau - T_h}^{\tau + T_h} [h(t - \tau) - r(t)]^2 dt - \int_{\tau - T_h}^{\tau + T_h} r(t)^2 dt = \\ &= \arg \min_{\tau} \int_{-T_h}^{T_h} [h(t) - r(t + \tau)]^2 dt - \int_{-T_h}^{T_h} r(t + \tau)^2 dt \end{aligned}$$

Supponendo che si riesce a trovare una sovrapposizione tra $h(t)$ e $r(t + \tau)$ che porta a 0 il termine $\int_{-T_h}^{T_h} [h(t) - r(t + \tau)]^2 dt$, risulta che la nostra stima è:

$$\hat{\tau} = \arg \max_{\tau} \int_{-T_h}^{T_h} [r(t)(t + \tau)]^2 dt$$

Si può quindi interpretare l'algoritmo di Coarse Estimator così: il segnale $r(t)$ viene filtrato con un rect di lunghezza $2T_h$ e si deduce il ritardo τ quando viene raggiunto il massimo (Figura 11).

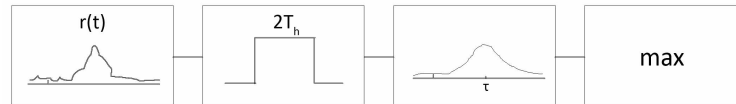


Figura 11: Interpretazione algoritmo Coarse Estimator

2.1.2 Fine estimator

Per quanto riguarda questo tipo di stimatore, mi limiterò a spiegare il funzionamento, essendo abbastanza complicato. Si tratta di uno stimatore che localizza il primo impulso in arrivo (quello che vogliamo stimare) nel seguente procedimento: dopo aver elevato al quadrato il segnale al ricevitore, gli impulsi (quello diretto, che sarà quello col tempo di arrivo minore, e quelli riflessi) vengono cancellati uno ad uno, ricordando il tempo di arrivo per ogni impulso. Per eseguire

questa operazione è necessario sapere il numero degli L percorsi, così da iterare questo procedimento L volte. Ora vengono confrontati i tempi di arrivo degli L impulsi ricevuti (questi impulsi è necessario che non si sovrappongano) e quello più piccolo sarà il τ che volevamo stimare.

2.2 Kurtosis Analysis

Questo tipo di stimatore prevede l'invio di un impulso di breve durata. Questo impulso viaggia nel canale e raggiunge il ricevitore, che stima il tempo di arrivo del segnale quando il livello di tensione supera una certa soglia λ . La soglia può essere scelta basandosi solamente sul livello di rumore. Ci sono due limitazioni:

1. è difficile stimare l'SNR;
2. Usando solo l'SNR del segnale ricevuto non si tiene conto della realizzazione del canale.

La stima di Kurtosis cattura sia la statistica della realizzazione del canale e l'SNR del segnale ricevuto, e può essere fatta analizzando i campioni del segnale, come sarà spiegato in seguito.

Il modello del segnale si può modellare nel seguente modo:

$$s(t) = dw(t)$$

dove abbiamo assunto che $d = 1$, $w(t)$ è un impulso di durata T_c . In questo caso consideriamo che il segnale viaggi su un percorso singolo, il trasmettitore da una parte e il ricevitore dall'altra. Dopo aver percorso la linea, il segnale ricevuto assume una forma del tipo:

$$r(t) = s(t - \tau) + n(t)$$

dove $n(t)$ è il rumore gaussiano bianco (AWGN) con media 0, potenza spettrale $N_0/2$ e varianza σ^2 . Essendo l'analisi di Kurtosis fatta per segnali a tempo discreto, è necessario campionare il segnale a intervalli di t_s , e per la nostra analisi utilizziamo il quadrato del segnale ricevuto:

$$r[n] = \int_{nt_s}^{(n+1)t_s} |r(t)|^2 dt$$

La stima di τ viene eseguita osservando il tempo in cui il segnale ricevuto supera una certa soglia λ , cioè

$$\hat{\tau} = \min[n : |s(t)|^2 \geq \lambda]$$

dove con n si intende il campione n -esimo. La Figura 12 mostra un esempio di tale procedimento.

La soglia λ deve essere preimpostata in base alle statistiche del segnale ricevuto. Dati i valori di campionamento di minima e massima energia, può essere utilizzata la seguente soglia normalizzata:

$$\lambda_{norm} = \frac{\lambda - \min r[n]}{\max r[n] - \min r[n]}$$

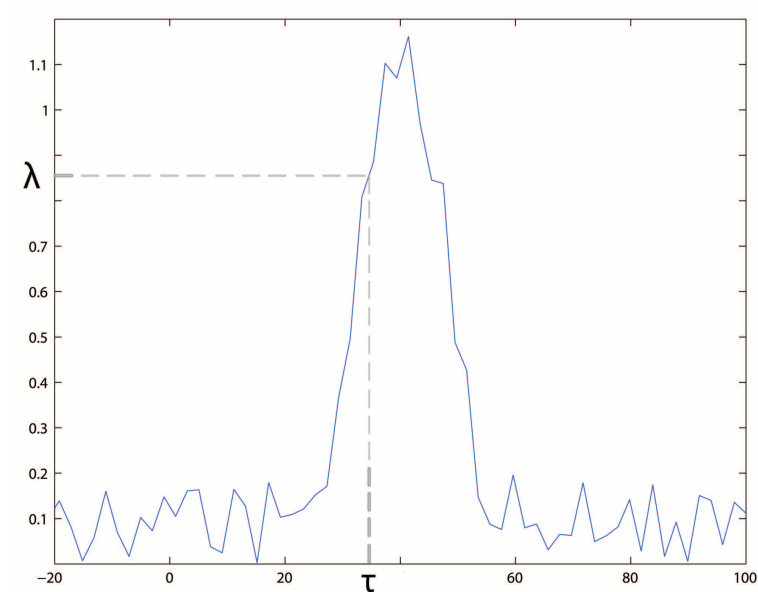


Figura 12: Esempio soglia λ

2.2.1 Selezione della soglia basata sull'analisi di Kurtosis

La soglia di Kurtosis, che viene chiamata kurtosis, è stata utilizzata per la stima dell'SNR nei sistemi di comunicazioni digitali, ed è applicabile per i nostri scopi. La kurtosis dei campioni dei segnali ricevuti $r[n]$ è calcolata usando il secondo e il quarto momento (vengono usati i momenti centrali, cioè $r[n]$ deve avere media nulla):

$$\kappa(r[n]) = \frac{E[r^4[n]]}{E^2[r^2[n]}}$$

applicata per gli n campioni ricevuti. La kurtosis relativa a una variabile aleatoria gaussiana può essere definita come $K(r[n]) = \kappa(z[n]) - 3$, che è zero per una distribuzione gaussiana. La kurtosis è utilizzata anche per vedere che una variabile aleatoria non è gaussiana, in quanto un grande valore di K implica una forte non Gaussianità. In assenza di segnale, $r[n]$ è considerato con distribuzione gaussiana, quindi $K=0$. Se l'SNR aumenta, K tende ad aumentare. Per i nostri scopi sono state trovate 2 soglie ottimali:

$$\lambda_{opt}^{(4ns)} = 0.673e^{-0.75\log_2 K} + 0.154e^{-0.001\log_2 K}$$

$$\lambda_{opt}^{(1ns)} = -0.082\log_2 K + 0.77$$

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, 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 100Mhz, 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 studiare il comportamento con 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 250MHz.

In un secondo momento è stato utilizzato per collegare un cavo BNC RG58 da 50 ohm 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.

Le caratteristiche del cavo utilizzato, rintracciabili facilmente su Internet [5], indicano una velocità di propagazione all'interno del cavo pari a 65.9% della velocità della luce e un ritardo di propagazione pari a 1.54ns/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 è provato, infine, ad implementare tramite Matlab (vedi Appendice) l'algoritmo di ricerca del ritardo τ sfruttando il Coarse estimator e la stima di Kurtosis descritto nel capitolo 2.

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 4.62MHz e come

frequenza finale 20.9Mhz. Questo perchè le frequenze utilizzate dal preambolo HomePlug sono molto più alte e quindi 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.

Poichè la frequenza di lavoro del generatore di funzioni è pari a 100Mhz, il segnale è stato interpolato in maniera tale da avere la stessa frequenza di campionamento. In Figura 13 viene mostrato il segnale come è stato trasmesso, con l'andamento in frequenza (per frequenze positive) e la sua autocorrelazione.

In Figura 14 si può osservare come tale segnale è stato ricevuto dall'oscilloscopio (l'oscilloscopio campiona a 250MHz), mentre in Figura 15 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 alla sua autocorrelazione 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.

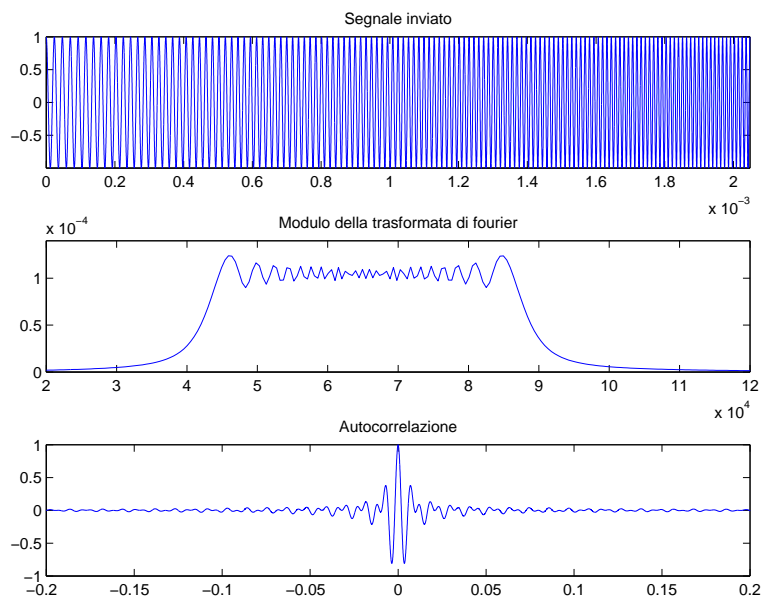


Figura 13: Segnale Chirp trasmesso

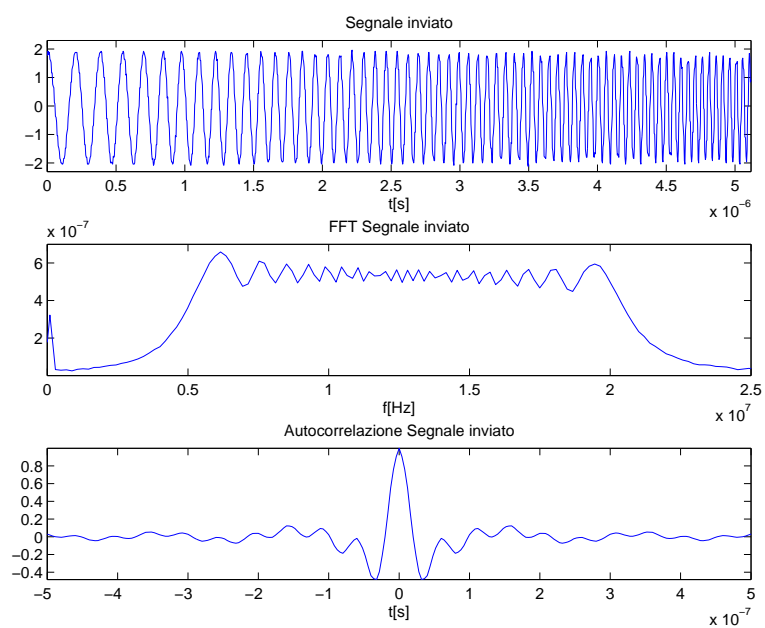


Figura 14: Segnale Chirp ricevuto a distanza 0

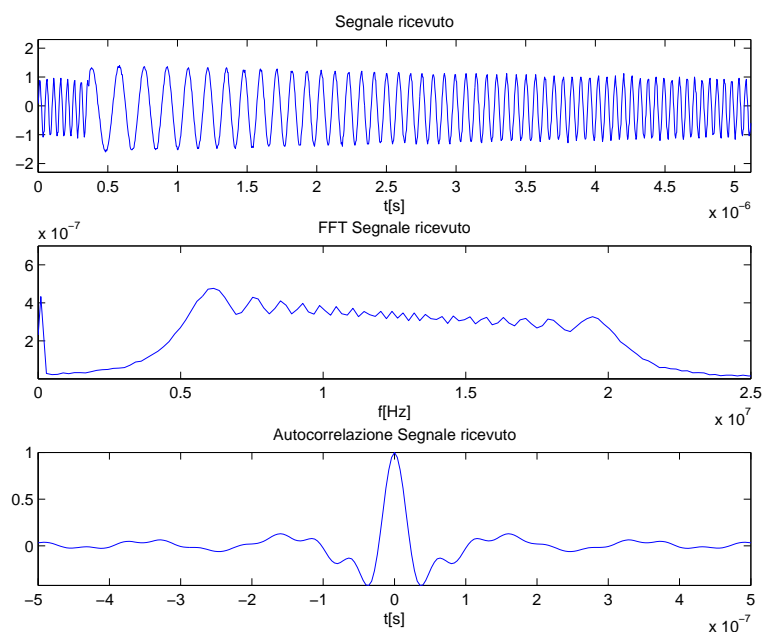


Figura 15: Segnale Chirp ricevuto da cavo BNC lungo 75 metri

3.1.1 Analisi stima con Coarse Estimator

In questa sezione viene cercato il tempo di propagazione del segnale nel cavo di lunghezza 75 metri con l'utilizzo della stima definita dal Coarse Estimator. In Matlab è stato implementato il codice, presente in Appendice, che trova il tempo di propagazione (vedere Figura 16): viene eseguita anzitutto la correlazione normalizzata tra il segnale inviato e il segnale ricevuto, poi il modulo quadro; è stato poi creato un segnale $\text{rect}(t)$ di estensione pari a $T_h = 40\text{ns}$, che è circa l'estensione del picco; dopo aver eseguito il filtraggio si ottiene il terzo grafico in figura. Risulta che il tempo di propagazione cercato è 372ns, che è un'ottima approssimazione. Infatti risulterebbe che il cavo è lungo 73.7 metri, un errore relativo di solo 0.017%. Quindi il Coarse Estimator è un algoritmo che funziona molto bene per la determinazione della distanza tra due nodi.

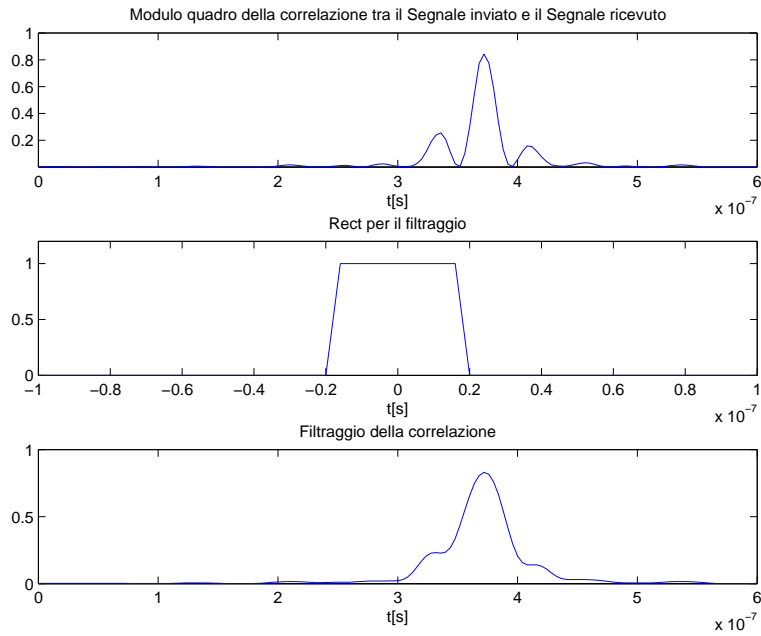


Figura 16: Analisi per la stima Coarse

3.1.2 Analisi stima con kurtosis

In questa sezione viene cercato il tempo di propagazione del segnale nel cavo di lunghezza 75 metri con l'utilizzo della stima definita dallo stimatore kurtosis. In Matlab è stato implementato il codice, presente in Appendice, che trova il tempo di propagazione: viene eseguita prima la correlazione normalizzata tra il segnale inviato e il segnale ricevuto, poi il modulo quadro (vedere Figura 17). Successivamente è stata calcolata la soglia λ attraverso l'algoritmo descritto dal kurtosis (vedi codice). Si ottiene un valore di λ pari a 0.4238: osservando la figura si nota che tale valore va a considerare la curva del picco più alto, che è quella riferita all'impulso inviato e quindi quella corretta. Risulta che il tempo di propagazione cercato è 364ns, che è una buona stima della lunghezza del cavo, ma peggiore rispetto all'analisi precedente. Infatti la lunghezza del cavo risulterebbe di 72.1 metri commettendo un errore relativo di 0.039%. Entrando nel dettaglio, la soglia λ cattura il momento in cui il nostro impulso sta salendo, ma che non ha ancora raggiunto il massimo. Per migliorare l'algoritmo sarebbe necessario aggiungere un piccolo ritardo in modo da raggiungere il picco dell'impulso.

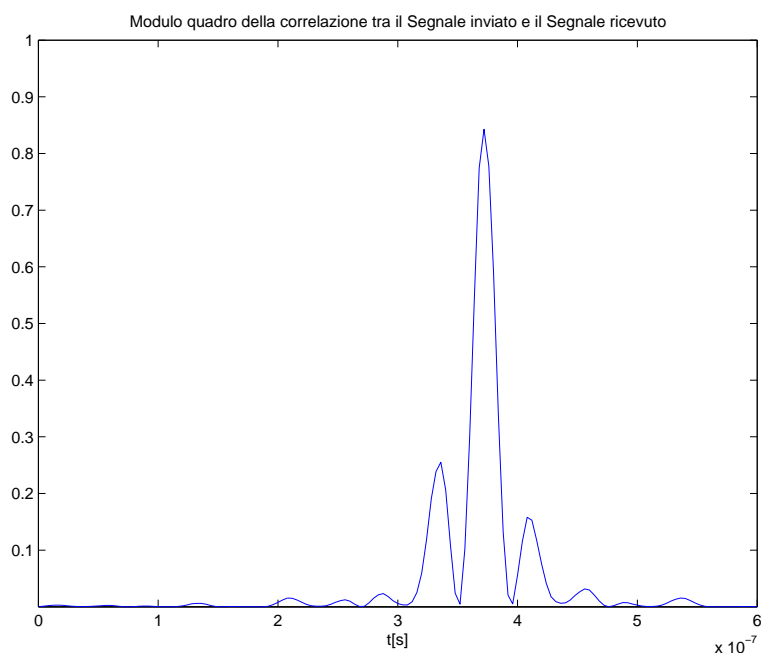


Figura 17: Analisi per la stima kurtosis

3.2 Preambolo HomePlug

Anche in questo caso prima di essere trasmesso il segnale è stato interpolato in maniera da raggiungere una frequenza di campionamento di 100MHz (Figura 18).

Di seguito si può osservare come tale segnale è stato ricevuto dall'oscilloscopio nel caso di connessione diretta (Figura 19) e nel caso di un cavo lungo (Figura 20). Anche in questo caso si osservino le differenze nel caso di cavo di connessione più lungo e in particolare il ritardo temporale conseguente. Si noti anche l'attenuazione maggiore per le frequenze maggiori.

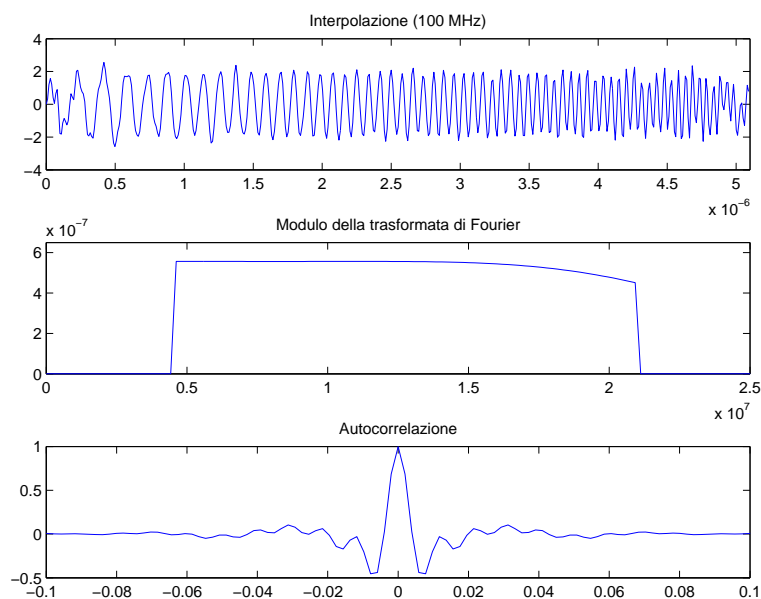


Figura 18: Segnale HomePlug trasmesso

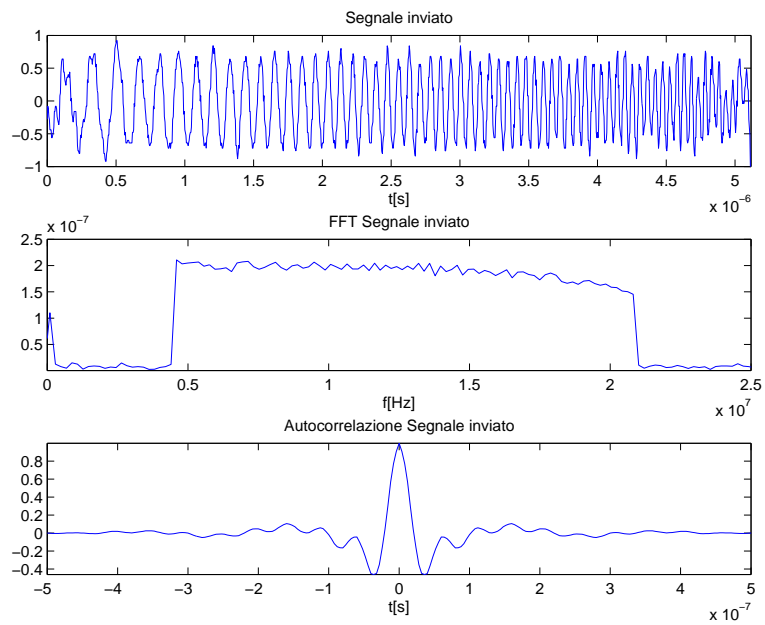


Figura 19: Segnale HomePlug ricevuto

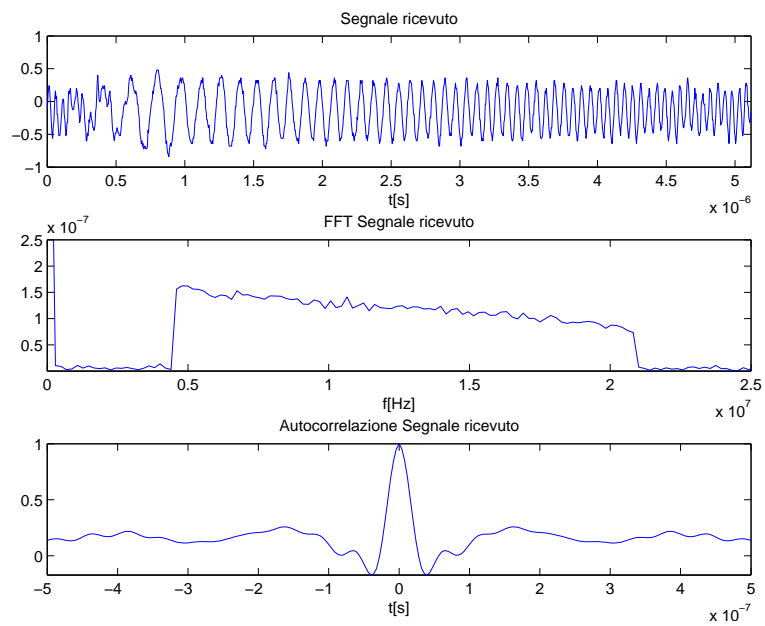


Figura 20: Segnale HomePlug ricevuto da cavo BNC lungo 75 metri

3.2.1 Analisi stima con Coarse Estimator

In questa sezione viene cercato il tempo di propagazione del segnale nel cavo di lunghezza 75 metri con l'utilizzo della stima definita dal Coarse Estimator. In Matlab è stato implementato il codice, presente in Appendice, che trova il tempo di propagazione (vedere Figura 21): viene eseguita anzitutto la correlazione normalizzata tra il segnale inviato e il segnale ricevuto, poi il modulo quadro; è stato poi creato un segnale $\text{rect}(t)$ di estensione pari a $T_h = 40\text{ns}$, che è circa l'estensione del picco come definito sopra; dopo aver eseguito il filtraggio si ottiene il terzo grafico in figura. Risulta che il tempo di propagazione cercato è 376ns, che, come prima, è un'ottima approssimazione. La lunghezza del cavo risulta essere 74.4 metri, commettendo un piccolissimo errore relativo di 0.008%. Quindi il Coarse Estimator è un algoritmo che funziona molto bene anche per questo tipo di segnale.

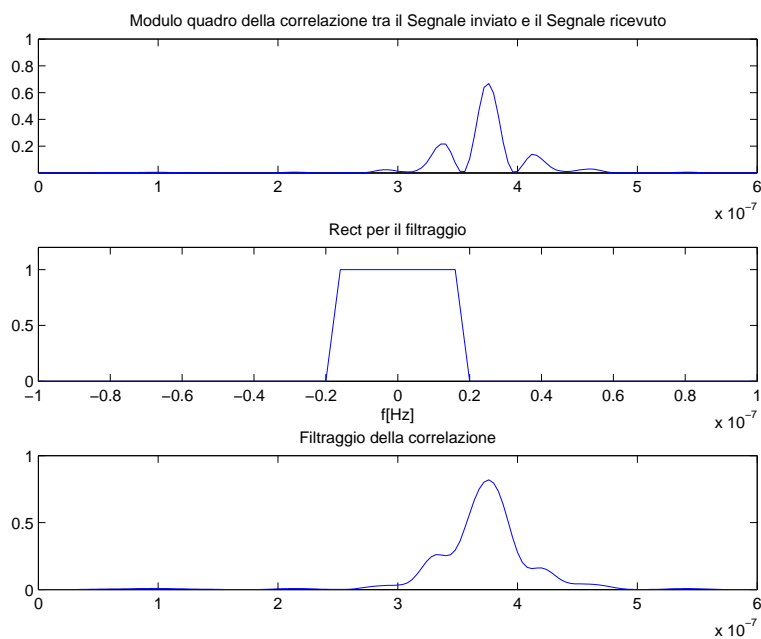


Figura 21: Analisi per la stima Coarse

3.2.2 Analisi stima con kurtosis

In questa sezione viene cercato il tempo di propagazione del segnale nel cavo di lunghezza 75 metri con l'utilizzo della stima definita dallo stimatore kurtosis. In Matlab è stato implementato il codice, presente in Appendice, che trova il tempo di propagazione: viene eseguita prima la correlazione normalizzata tra il segnale inviato e il segnale ricevuto, poi il modulo quadro (vedere Figura 22). Successivamente è stata calcolata la soglia λ attraverso l'algoritmo descritto dal kurtosis (vedi codice). Si ottiene un valore di λ pari a 0.4301: osservando la figura si nota che, come nel caso del Chirp, tale valore va a considerare la curva del picco più alto, che è quella riferita all'impulso inviato e quindi quella corretta. Risulta che il tempo di propagazione cercato è 368ns, che è una buona stima della lunghezza del cavo, ma peggiore rispetto all'analisi precedente. Infatti la lunghezza del cavo risulterebbe di 72.9 metri commettendo un errore relativo di 0.028%. Entrando nel dettaglio, la soglia λ cattura il momento in cui il nostro impulso sta salendo, ma che non ha ancora raggiunto il massimo. Per migliorare l'algoritmo sarebbe necessario aggiungere un piccolo ritardo in modo da raggiungere il picco dell'impulso.

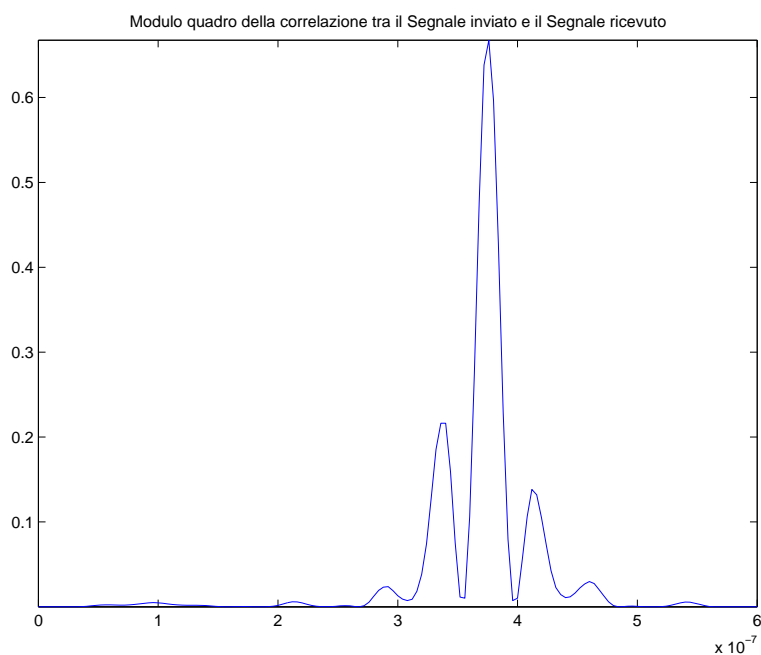


Figura 22: Analisi per la stima kurtosis

3.2.3 Conclusioni

Osservando i dati trovati, si può concludere che il segnale preambolo HomePlug rivela con maggiore precisione la reale distanza del cavo; il segnale Chirp risulta comunque un ottimo segnale con errore molto basso.

Quindi concludo dicendo che il Chirp e l'HomePlug sono segnali validi, che riescono ad individuare con buona approssimazione la distanza di un cavo di 75 metri, commettendo un errore al massimo di 3 metri (considerando i nostri esperimenti).

4 Appendice: codici Matlab utilizzati

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

Segnale Chirp

```
%Segnale Chirp

Tc=10*(10^-9);      % 100MHz
T=2048*(10^-6);    % Estensione del segnale
5 t=0:Tc:T-Tc;     % Asse dei tempi
A=1;
f0=41992;          % frequenza iniziale
f1=88867;          % frequenza finale
m=(f1-f0)/T;
10 s = A*cos(2*pi*(f0*t+0.5*m*(t.^2)));
    % segnale Chirp

figure(1);
plot(t,s);
axis([-inf inf -1.1 1.1]);
15 xlabel('t[s]');
ylabel('s(t)');
title('Chirp');
print -depsc CHIRP.eps;
    % salvo il grafico in .eps

20 % Modulo della trasformata di Fourier
N=length(s);
figure(2);
fft_abs=abs(fft(s,Tc)); % funzione che mi grafica il modulo della fft
25 axis([-0 130000 -inf 1.4*(10^-4)]);
xlabel('f[Hz]');
ylabel('S(f)');
title('FFT Chirp');
print -depsc FTCHIRP.eps;

30 % Autocorrelazione normalizzata
corr=xcorr(s,s); % Calcolo l'autocorrelazione
C=length(corr);
t1=(-(C-1)/2:1:(C-1)/2)*Tc;
35 figure(3);
plot(t1,corr);
axis([-0.0002 0.0002 -inf 120000]);
xlabel('t[s]');
ylabel('R_s(t)');
40 title('Autocorrelazione Chirp');
print -depsc CORRCHIRP.eps;
```

Segnale HomePlug

```
% Segnale HomePlug

% Preambolo di HomePlug con queste semplificazioni:
% ho tolto le funzioni sign, w(n), e ho preso solo 256 valori di n
5

load('pc.mat'); % importa la tabella delle fasi
T=5.12*10^(-6); % estensione del segnale
fun = zeros(1, 256); % crea funzione finale, per ora vuota
10 fc = (23:1:106)/256; % definisce la frequenza (c+23)/256
for n=0:1:(256-1) % implementa la funzione
    a = 2*pi*fc*n + pc; % calcola argomento del coseno
    b= cos(a); % calcola i coseni
```

```

    c = 2/sqrt(84)*sum(b(:)); % esegue la sommatoria
15   fun(1, n+1) = c;         % salva il risultato nella funzione finale
    end;

    Tc = 20*(10^-9);        % tempo di campionamento
    t=0:Tc:Tc*255;         % asse dei tempi
20   figure(1)
    plot(t, fun, 'o');
    hold on
    plot(t, fun);
    xlabel('t[s]');
25   ylabel('h(t)');
    axis([0 Tc*255 -3 3])
    title('HomePlug')
    print -depsc hp.eps;

30 % Modulo della trasformata di Fourier
    figure(2)
    FUN = fft_abs(fun, Tc);
    axis([-0 inf -0.3*10^-7 6*10^-7]);
    xlabel('f[Hz]');
35   ylabel('H(f)');
    title('FFT HomePlug')
    print -depsc hp_fft.eps;

% Autocorrelazione normalizzata
40 t2 = ((-T+Tc):Tc:(T-Tc))/T;
    r = xcorr(fun, fun, 'coeff');
    figure(3)
    plot(t2, r)
    xlabel('t[s]');
45   ylabel('R_s(t)');
    axis([-0.1 0.1 -0.5 1])
    title('Autocorrelazione HomePlug')
    print -depsc hp_corr.eps;

```

Funzione - Modulo trasformata di Fourier

```

% Calcola il modulo della trasformata di Fourier della funzione s
% Tc tempo di campionamento della funzione

function func = fft_abs(s, Tc)
5   Fc = 1/(Tc*(length(s)-1)); %Fc: passo di campionamento in frequenza
    F = 1/Tc;                  %F: periodo in frequenza
    f = -(F/2):Fc:(F/2);
    u = fftshift(s);          %Ricostruisco il periodo fondamentale
    S = fftshift(Tc*fft(u));  %Vado in frequenza
10   func = abs(S);           %Calcolo il modulo della trasformata
    plot(f, func)
    end

```

Calcolo del tau del segnale Chirp con Coarse Estimator

```

% importo i 2 segnali da analizzare
% in Chirp.dat è il segnale inviato
% in c_lungo.dat il segnale ricevuto
5
    Fc=250*10^6;              % frequenza campionamento 250MHz
    Tc=1/Fc;                  % passo di campionamento
    N=1280;                   % numero campioni considerati
    t=(0:N-1)*Tc;            % asse dei tempi globale
10
    % importo il segnale inviato, preso con l'oscilloscopio a distanza 0
    load Chirp.dat;

```

```

chirp_inv=Chirp(1:N);

15 % importo il segnale ricevuto, preso con l'oscilloscopio con cavo lungo
    75m
    load c_lungo.dat;
    chirp75=c_lungo(1:N);

    % trovo autocorrelazione tra il segnale inviato e il segnale ricevuto
20 r=xcorr(chirp75, chirp_inv, 'coeff');
    t_r=(-N+1:N-1)*Tc;
    figure
    plot(t_r, r);
    axis([-0*10^-6 0.6*10^-6 -inf inf]);
25 title('Correlazione tra il Segnale inviato e il Segnale ricevuto');

    % Modulo quadro
    r=r.*r;
    figure
30 subplot(311);
    plot(t_r, r);
    xlabel('t[s]');
    axis([-0*10^-6 0.6*10^-6 -inf 1]);
    title('Modulo quadro della correlazione tra il Segnale inviato e il
        Segnale ricevuto');

35 % supponiamo, guardando il grafico, che l'impulso ricevuto ha estensione
    % Th=0.4*10^-7.
    % quindi, seguendo l'algoritmo di stima del coarse, si esegue un
        filtraggio
    % con un rect centrato in 0 e di estensione Th:
40 Th=0.4*10^-7;
    % visto che xcorr vuole 2 segnali della stessa lunghezza, definisco il
        rect
    % utilizzando l'asse dei tempi di r:
    rect=t_r*0; % definisco il vettore rect
45 x=find(abs(t_r)<(Th/2));
    rect(x)=1;
    subplot(312);
    plot(t_r, rect);
    xlabel('t[s]');
50 axis([-0.1*10^-6 0.1*10^-6 -inf 1.2]);
    title('Rect per il filtraggio');

    % eseguo una correlazione che corrisponde al filtraggio
    h=xcorr(r, rect, 'coeff');
55 t_h=(-2*N+2:2*N-2)*Tc;
    subplot(313);
    plot(t_h, h);
    xlabel('t[s]');
    axis([-0*10^-6 0.6*10^-6 -inf 1]);
60 title('Filtraggio della correlazione');

    print -depsc chirp_coarse.eps

    % il massimo corrisponde al valore cercato
65 m=max(h);
    tau=(find(h==m)-((length(h)+1)/2))*Tc

```

Calcolo del tau del segnale Chirp con kurtosis

```

% La prima parte è identica al codice precedente, fino al modulo quadro

% ricerca del lambda tramite la stima di kurtosis
% suppongo valori equiprobabili e calcolo aspettazione di un pezzo di
    segnale
5 % che contiene l'impulso

```

```

r1=r(1300:1450);           % pezzo di segnale su cui calcolare le
    aspettazioni
m = mean(r1);             % media del segnale
sigma2 = var(r1);        % secondo momento
mu = mean((r1-m).^4);    % quarto momento
10 k=mu/(sigma2^2);
    K=k-3;
    % Utilizzo la formula riferita al tempo di lns
    % lambda=0.673*exp(-0.75*log2(K))+0.154*exp(-0.001*log2(K))
15 lambda=-0.082*log2(K)+0.77;

% trovo la stima della soglia del lambda
m=find(r>(lambda));
tau=(m(1)-((length(r)+1)/2))*Tc

```

Calcolo del tau del segnale HomePlug con Coarse Estimator

```

% La prima parte è identica al codice precedente, fino al modulo quadro
% a parte che vengono caricati i segnali dell'HomePlug

% supponiamo, guardando il grafico, che l'impulso ricevuto ha estensione
5 % Th=0.4*10^-7.
% quindi, seguendo l'algoritmo di stima del coarse, si esegue un
    filtraggio
% con un rect centrato in 0 e di estensione Th:
Th=0.4*10^-7;
% visto che xcorr vuole 2 segnali della stessa lunghezza, definisco il
    rect
10 % utilizzando l'asse dei tempi di r:

    rect=t_r*0;           % definisco il vettore rect
    x=find(abs(t_r)<(Th/2));
    rect(x)=1;
15 subplot(312);
    plot(t_r, rect);
    xlabel('f [Hz]');
    axis([-0.1*10^-6 0.1*10^-6 -inf 1.2]);
    title('Rect per il filtraggio');
20 % eseguo una correlazione che corrisponde al filtraggio
h=xcorr(r, rect, 'coeff');
t_h=(-2*N+2:2*N-2)*Tc;
subplot(313);
25 xlabel('t [s]');
    plot(t_h, h);
    axis([-0*10^-6 0.6*10^-6 -inf 1]);
    title('Filtraggio della correlazione');
30 print -depsc hp_coarse.eps

% il massimo corrisponde al valore cercato
m=max(h);
tau=(find(h==m)-((length(h)+1)/2))*Tc

```

Calcolo del tau del segnale HomePlug con kurtosis

```

% La prima parte è identica al codice precedente, fino al modulo quadro

% ricerca del lambda tramite la stima di kurtosis
% suppongo valori equiprobabili e calcolo aspettazione di un pezzo di
    segnale
5 % che contiene l'impulso
r1=r(1300:1450);           % pezzo di segnale su cui calcolare le
    aspettazioni

```

```

m = mean(r1);           % media del segnale
sigma2 = var(r1);      % secondo momento
mu = mean((r1-m).^4);  % quarto momento
10 k=mu/(sigma2^2);
K=k-3;
% Utilizzo la formula riferita al tempo di lns
% lambda=0.673*exp(-0.75*log2(K))+0.154*exp(-0.001*log2(K))
15 lambda=-0.082*log2(K)+0.77;

% trovo la stima della soglia del lambda
m=find(r>(lambda));
tau=(m(1)-((length(r)+1)/2))*Tc

```


Riferimenti bibliografici

- [1] T. Erseghe, F. Lorenzon, S. Tomasin, A. Costabeber and P. Tenti, “Distance Measurement over plc for Dynamic Grid mapping of Smart Micro Grids”.
- [2] A. Gemito, “Particle Swarm Optimization per lo Shaping del segnale radar meteorologico”.
- [3] A. D’Amico, U. Mengali and L. Taponecco, “Energy-based toa estimation”, *Wireless Communications, IEEE Transactions on*, vol. 7, no. 3, pp. 838 - 847, Mar. 2008.
- [4] I. Guvenc and Z. Sahinoglu, “Threshold selection for uwb toa estimation based on kurtosis analysis”, *Communication Letters, IEEE*, vol. 9, no. 12, pp. 1025 - 1027, Dec. 2005.