Progetto MATLAB

Fondamenti di Segnali e Trasmissione

Problema 10

FRANCESCO LENZI 10610504

Punto 1

Definisco dei parametri k e T ed assegno loro dei valori arbitrari. Fatto ciò, calcolo la Trasformata di Fourier di g(t) tramite la funzione DFT.

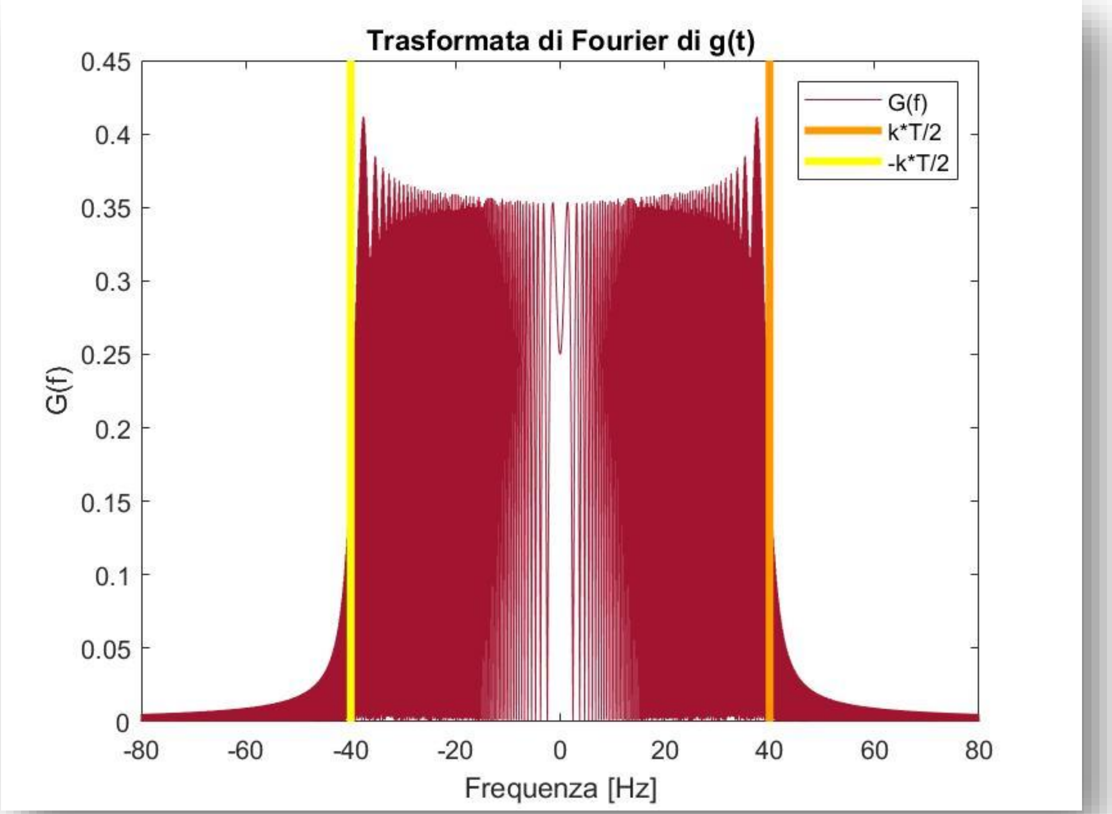
Si evidenzia come, al variare dei parametri, la banda rimane circa uguale al prodotto dei due.

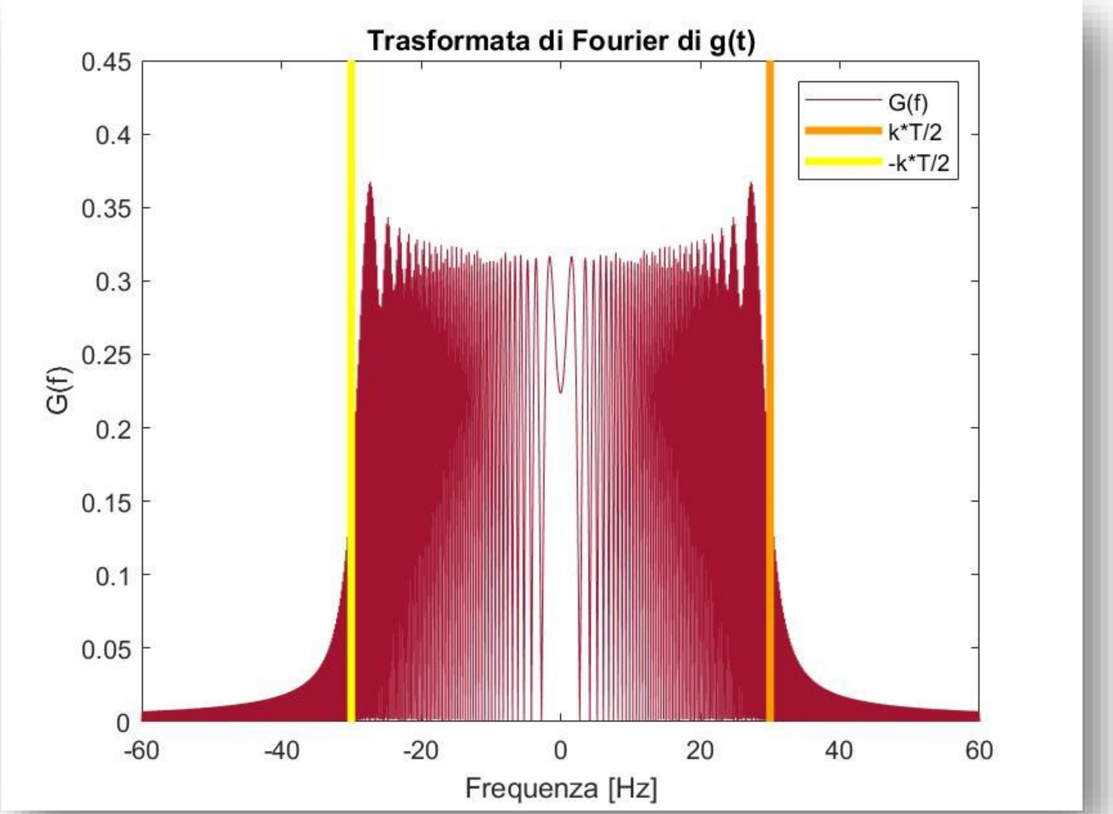
```
k = 8;
T = 10;
```

→ $B \simeq 80$ Hz

```
k = 10;
T = 6;
```

→ $B \simeq 60$ Hz

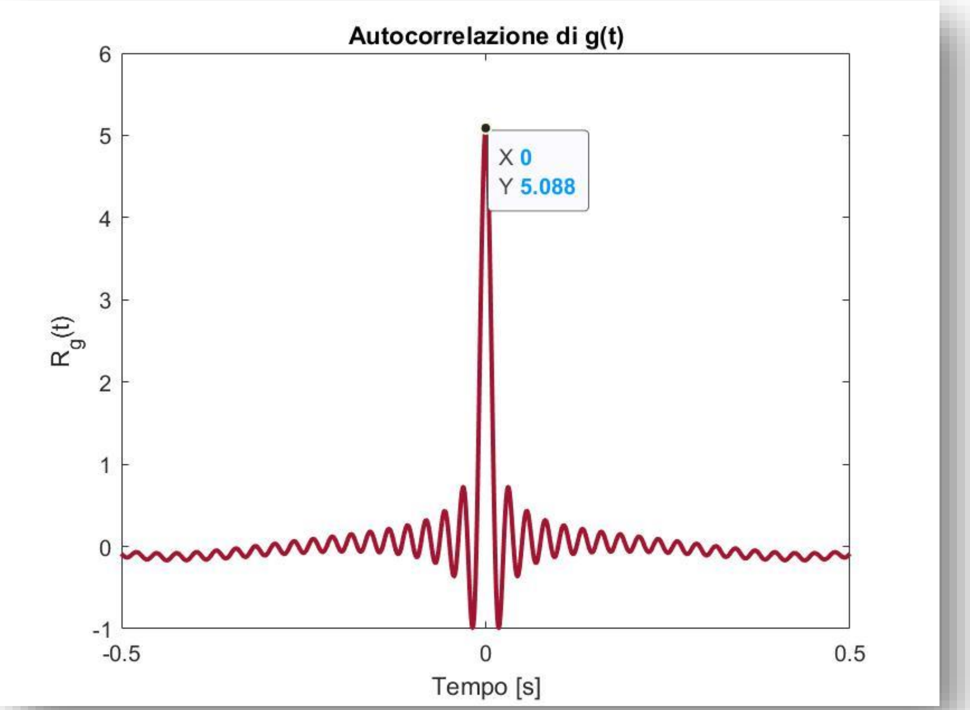




Calcolo l'autocorrelazione di g(t) mediante la seguente formula:

$$R_g(t) = g(t) * g * (-t)$$

La rappresento graficamente e valuto l'ampiezza sia teoricamente che sperimentalmente.



Metodo Teorico

Il massimo teorico, per le proprietà dell'autocorrelazione sarà in t = 0. Inoltre posso valutare l'ampiezza massima calcolando l'energia del segnale g(t):

$$R_g(0) = E_g = \int |g(t)|^2 dt = 5{,}088$$

Metodo sperimentale

Sperimentalmente, tramite la funzione max(), calcolo direttamente la massima ampiezza di $R_g(t)$:

```
tmax_Rg = 0

Max_Rg = 5.0884
```

In conclusione, come possiamo notare, il risultato è lo stesso con entrambi i metodi utilizzati.

Punto 2

Come abbiamo visto nel **Punto 1**, la banda di g(t) è circa k·T .
Affinché la Banda sia di 1 Hz, dato che T = 100 s, il valore di k sarà: $k = \frac{B}{T} = \frac{1\ Hz}{100\ s} = 0.01$

Richiami di Teoria

Considero un segnale ritardato $y(t) = x(t - \tau)$, è nota la seguente proprietà:

$$R_{yx}(t) = y(t) * x * (-t) = R_x(t - \tau)$$

Sfruttando quest'ultima, posso elaborare un metodo per la valutazione di τ calcolando la cross-correlazione fra $x_1(t)$ e g(t). Quest'ultima presenterà un picco traslato di τ rispetto a quello dell'autocorrelazione del segnale g(t). Pertanto, calcolando la distanza fra i picchi, posso trovare il valore di τ.

Mi aspetto che, con l'aggiunta del rumore w(t), la cross-correlazione venga disturbata e dunque il calcolo di τ venga impreciso.

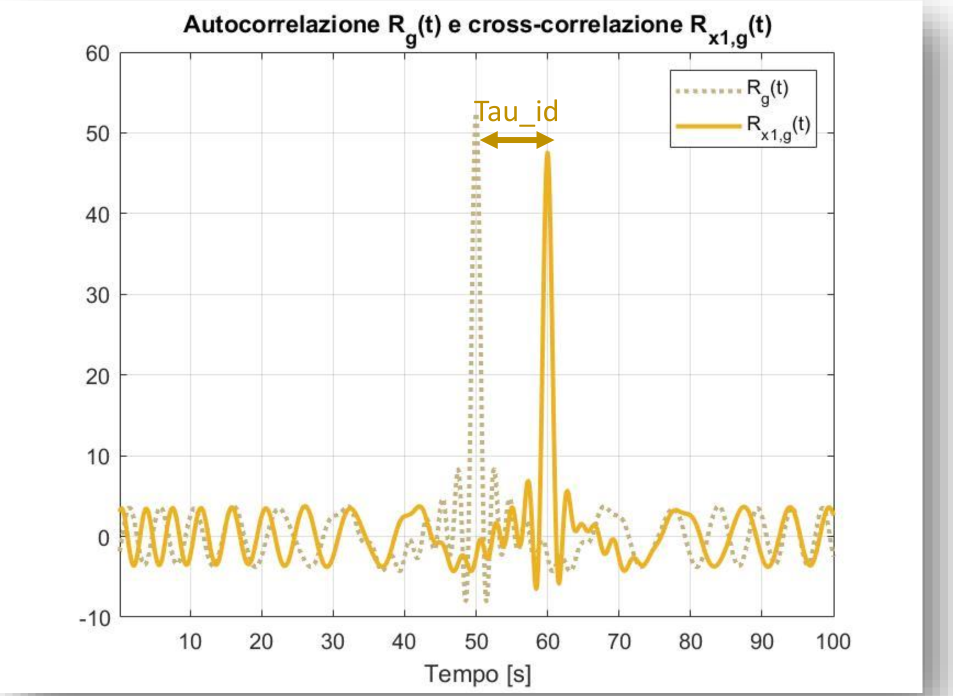
Osservazione

Dato che l'asse dei tempi non è simmetrico ma da 0 s a T = 100 s, il picco dell'autocorrelazione di g(t) sarà perfettamente a metà dell'asse dei tempi, ovvero a t = 50s.

Verifichiamo che il metodo funzioni inserendo un valore arbitrario a τ, e analizzando la distanza fra i picchi dei seguenti grafici:

```
54 -      tau = 10;
```

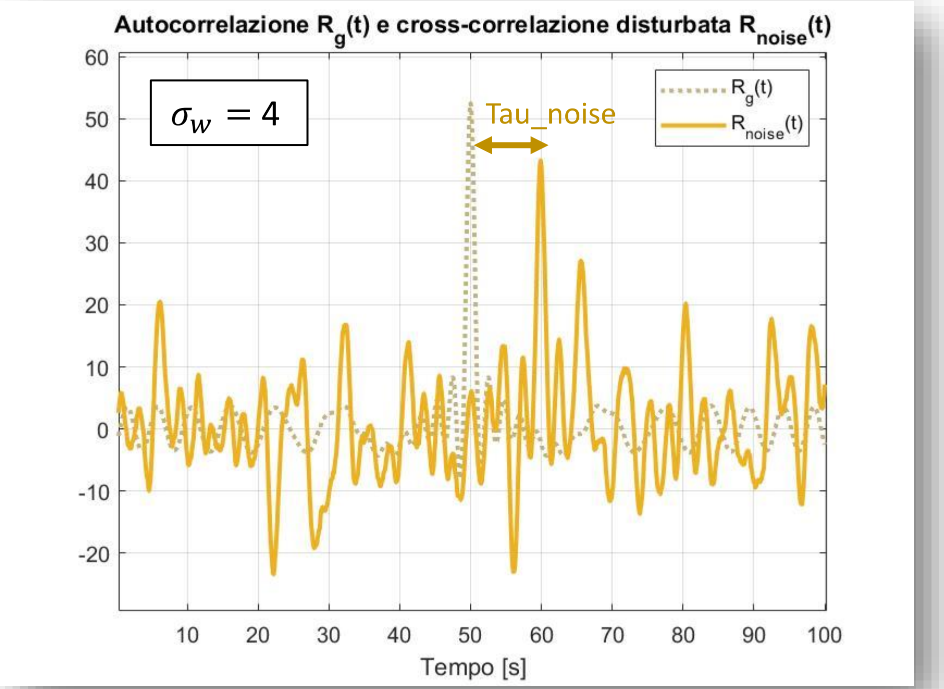
Caso ideale (x1 senza rumore)




```
tau_id = pmax_Rx1g - pmax_Rg        tau_id = 10
```

Come possiamo notare, senza la presenza del rumore, la distanza fra i picchi è perfettamente uguale al valore di τ.

Caso Reale (x1 con il rumore w(t))




```
tau_noise = pmax_Rnoise - pmax_Rg        tau_noise = 9.8000
```

In questo caso, invece, considerando anche il contributo del rumore (Incorrelato perché gaussiano bianco), il risultato è simile ma distorto dal rumore.

Cosa succede per SNR molto Bassi? (Lascio il valore di τ = 10)

Per SNR molto bassi, la potenza del Rumore e del segnale diventano comparabili ed il segnale non si distingue più dal rumore. Per questo motivo minore sarà l’SNR, peggiore sarà la stima di τ.

Consideriamo il rumore w(t) con deviazione standard pari a 12:

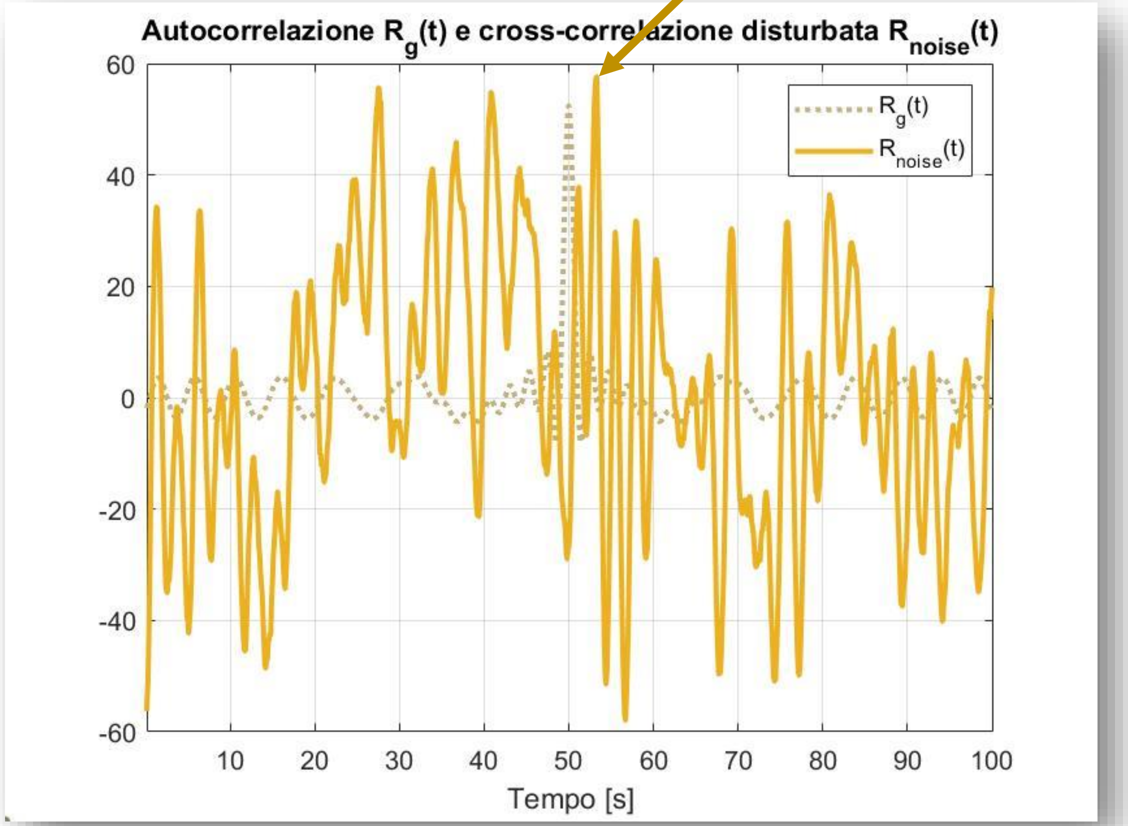
```
sigmaw = 12;
w = sigmaw*randn(1,Nt);
```

Calcoliamo l’SNR e verifichiamo che sia molto basso:

$$SNR = \frac{Potenza_{segnale}}{Potenza_{rumore}}$$

```
SNR = 0.0036
SNRdB = -24.3821
```

Picco di R_{noise}



Disegnando ora lo stesso grafico mostrato nella slide precedente, notiamo come il segnale è molto più disturbato ed è quasi impossibile individuare il picco della crosscorrelazione R_{noise}.

In questo caso il ritardo elaborato è di:

```
tau_noise = 3.3000
```

L’errore è del: **67%!!!**

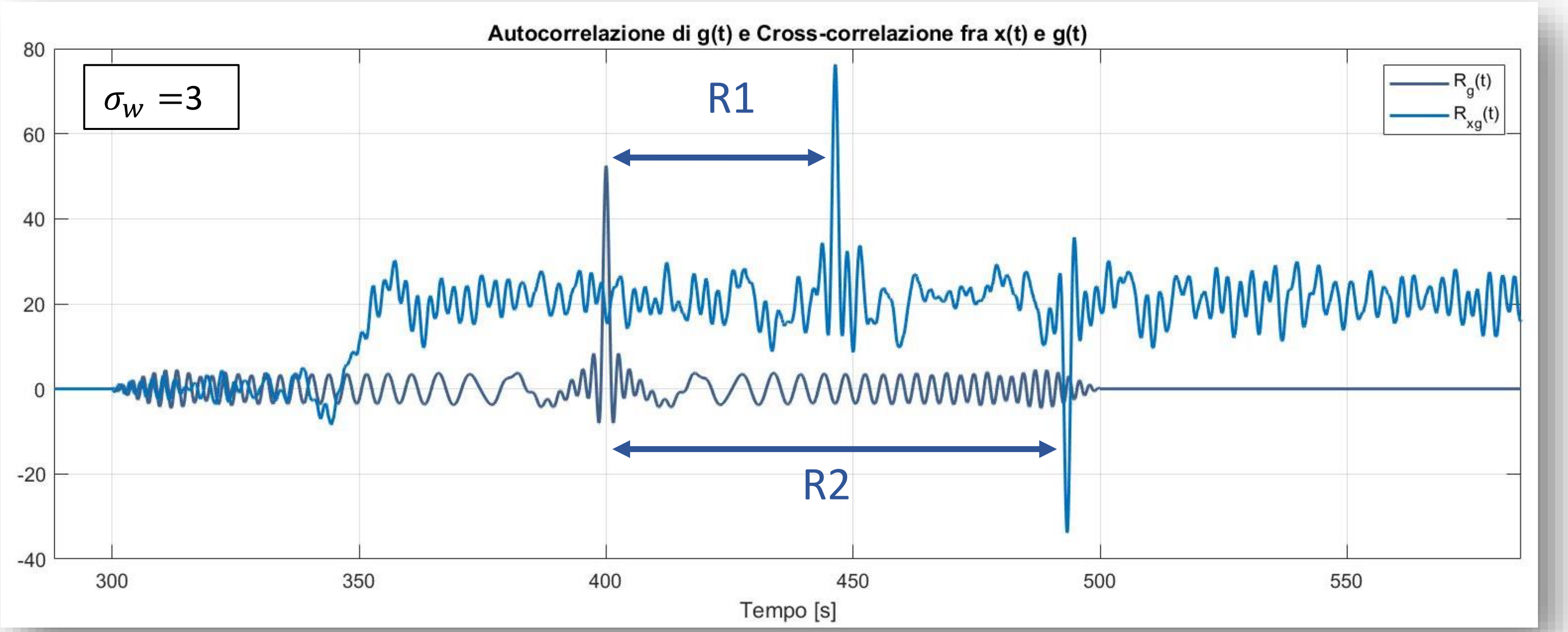
```
tau_err = (abs(pmax_Rnoise-pmax_Rx1g)/tau)*100
```

```
tau_err = 67.0000
```

Punto 3

Il segnale x(t) è la somma dei segnali x1 e x2 forniti dalla funzione mod_1.

Dato che v = 1 e x1,x2 hanno ampiezza opposta, la cross-correlazione fra x(t) e g(t) presenterà un picco positivo ed un picco negativo.
Posso, dunque, utilizzare il metodo applicato nel **Punto 2** e, in particolare, calcolo lo scostamento del picco positivo per R1 e lo scostamento dal picco negativo per il calcolo di R2.




```
R1 = pmax_Rxg - pmax_Rg
R2 = pmin_Rxg - pmax_Rg
```

```
R1 = 46.4000
R2 = 93.3000
```

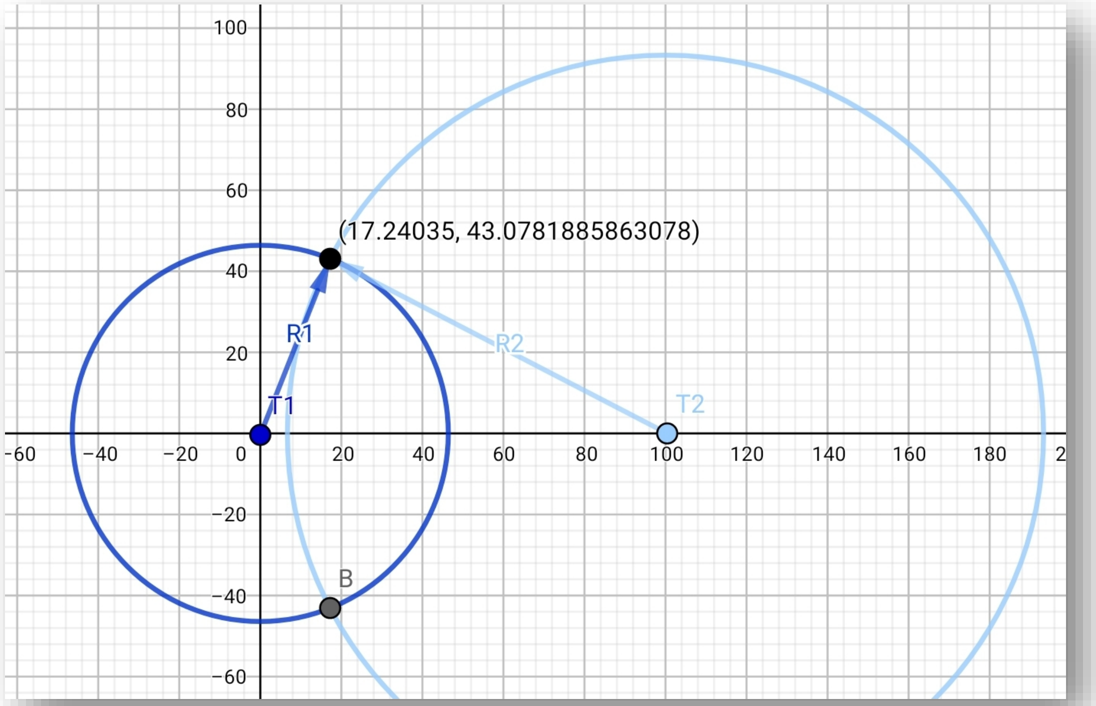
Osservazione

Come possiamo notare, la cross-correlazione fra x(t) e g(t) presenta uno scalino che aumenta all'aumentare della variazione standard passata alla funzione mod_1. Ciò avviene perché l'errore è uniforme e dunque presenta un valor medio non nullo. Mi aspetto, dunque, che per valori di *sigmaw* alti il minimo di R_{xg} non venga più letto correttamente.

Calcolo delle coordinate $[x_0, y_0]$

Per ottenere le coordinate $[x_0, y_0]$ della mia posizione a partire dalle distanze dai ricevitori R1 e R2 utilizzo come metodo per calcolarle l'INTERSEZIONE FRA CIRCONFERENZE.

Utilizziamo le distanze calcolate nella slide precedente: **R1 = 46.4000** **R2 = 93.3000**



Per trovare il punto di intersezione, metto a sistema le equazioni delle due circonferenze:

$$\begin{cases} x^2 + y^2 = R1^2 \\ (x-100)^2 + y^2 = R2^2 \end{cases}$$

Risolvendo il sistema trovo le coordinate:

$$\begin{cases} x_0 = \dfrac{R1^2 - R2^2 + 100^2}{200} \\ y_0 = \sqrt{R1^2 - x_0{}^2} \end{cases}$$

I punti di intersezione possono essere al massimo due. Ai fini del rilevamento della posizione, considero sempre quello nel semipiano positivo.

Inserisco le seguenti formule nel codice MATLAB e calcolo le coordinate. Successivamente calcolo l'accuratezza della misura confrontando i risultati con quelli restituiti dalla funzione mod_1.

Calcolo delle coordinate

```
x0 = abs((R1^2 - R2^2 + 10000)/200)
y0 = abs(sqrt(R1^2 - x0^2))
```

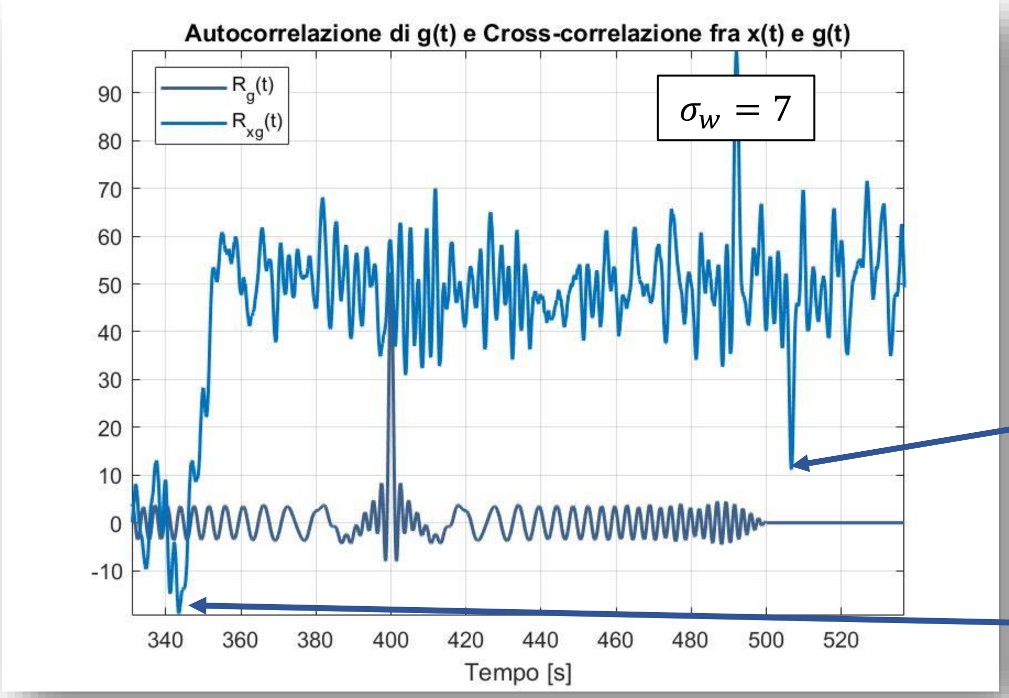
```
x0 = 17.2403
y0 = 43.0782
```

Calcolo l'accuratezza della misura

```
err_x = abs(x0-par.P(1))
err_y = abs(y0-par.P(2))
```

```
err_x = 0.0368
err_y = 0.0290
```

I calcoli svolti sono stati fatti considerando un rumore basso. Vediamo cosa succede se aumento *sigmaw*:



Come accennato in precedenza, aumentando *sigmaw,* lo scalino diventa sempre più grande.
Pertanto, la funzione min non troverà più il picco negativo nella posizione esatta ma bensì in una totalmente differente.

Minimo nella posizione esatta

Minimo letto dalla funzione min()

Di conseguenza, i valori dell'errore di accuratezza aumenteranno:

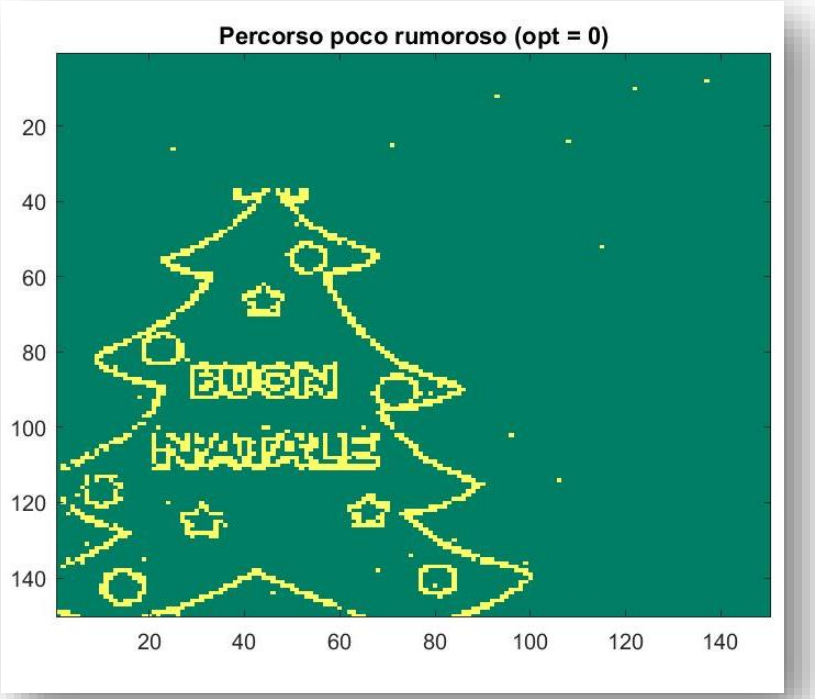
```
err_x = 40.8150
err_y = 33.7269
```

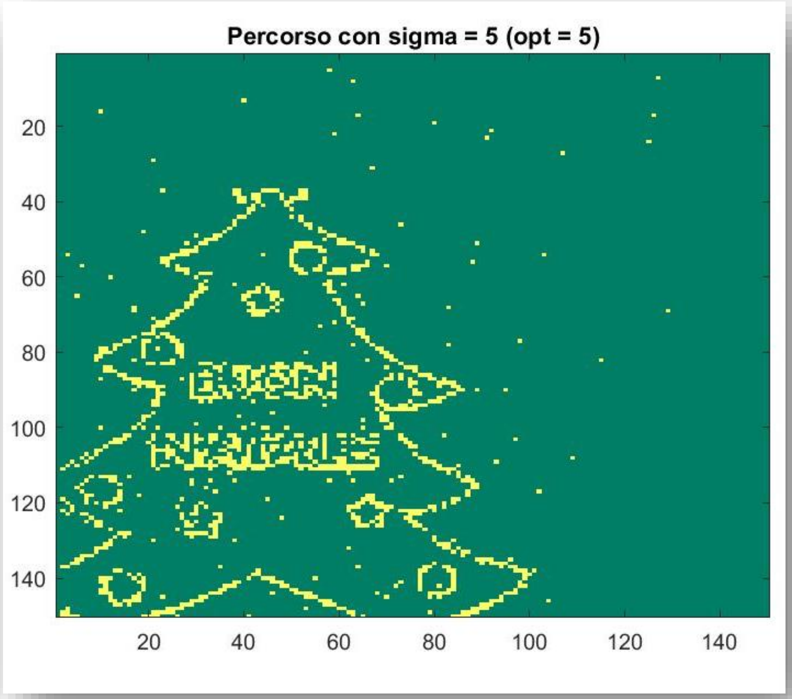
Punto 4

Definisco lo spazio del parco come una matrice 150 x 150 , in particolare analizzo il caso con opt = 0 (opzione poco rumorosa), il caso con opt = 5 (opzione con rumore avente σ_w=5) ed il caso con opt = 7 (opzione con rumore avente σ_w=7).

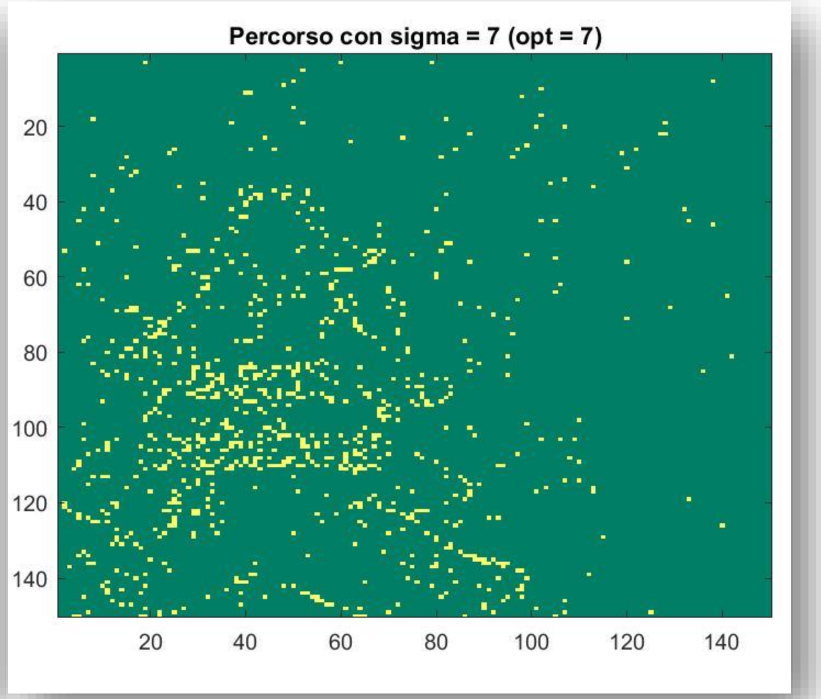
Tramite un ciclo iterativo, analizzo ogni volta l'n-esimo segnale generato dalla funzione mod_2 e assegno 1 all'elemento della matrice corrispondente alla posizione calcolata.

Trasformando la matrice in immagine possiamo notare come all'aumentare del rumore il percorso diventa sempre più distorto ed aumenta il numero delle false misure.





False misure : ~191



False misure : ~402