



# Cause di disturbo in tomografia

---

- Rumore e sua propagazione
- Conservazione della componente continua (analisi quantitativa)
- Dati insufficienti



# Noise

---

Si possono identificare due tipi di noise sui dati (proiezioni).

- Il primo è dovuto all'elettronica (amplificatori, convertitori analogico/digitali etc..) e si può considerare **gaussiano** (bianco) **addittivo** (ovvero **non dipendente** dal segnale).
- Il secondo è intrinseco nel conteggio fotonico, ha, quindi statistica poissoniana ed è **dipendente** dal segnale.



# Noise poissoniano

---

Questa fonte di rumore è quella che incide di più sui dati, soprattutto nella tomografia a emissione, dove l'efficienza è molto bassa.

Infatti, per ovvie ragioni mediche, non si può irradiare il corpo umano con dosi troppe elevate. Di conseguenza il **numero di conteggi** rilevato è in genere **basso** e il rumore poissoniano alto, infatti:

- Sia  **$\langle N \rangle$**  il **numero di medio fotoni** che arrivano sul rivelatore; se il processo è poissoniano **la varianza è  $\langle N \rangle$**  e la deviazione standard  $\sqrt{\langle N \rangle}$



# Simulazione (noise poissoniano)

---

In ogni linguaggio di programmazione esistono modi per creare sequenze di numeri pseudo-casuali che abbiano una distribuzione di probabilità che obbedisce ad una certa legge.

In IDL esiste la funzione **radomn**

```
result=radomn(seed,n,poisson=valore_medio)
```

crea un vettore di **n** elementi con valore medio assegnato e distribuiti con statistica poissoniana

```
print, moment(result)
```

stampa i momenti del vettore (media, varianza, etc..)



# Simulazione (noise poissoniano)

---

Alcune prove:

```
pro poisson
```

```
result=make_array(1000,/double)
```

```
for i=100,1000,100 do begin
```

```
result=randomn(seed,1000,poisson=i)
```

```
print,i,min(result),max(result),mean(result),stddev(result), stddev(result)/mean(result)
```

```
end
```

```
end
```



# Simulazione (noise poissoniano)

---

<b>n</b>	<b>min</b>	<b>max</b>	<b>media</b>	<b>std-dev</b>	<b>errore relativo</b>
5	0.000	14.0000	4.98700	2.25697	0.452571
10	2.00000	22.0000	9.90000	3.18813	0.322033
100	62.0000	133.000	99.6470	9.80533	0.0984006
200	154.000	248.000	200.108	13.5908	0.0679175
300	247.000	355.000	300.312	17.3389	0.0577364
400	334.000	475.000	399.863	20.9588	0.0524148
500	436.000	576.000	500.580	22.7528	0.0454529
600	521.000	673.000	600.313	23.3565	0.0389072
700	623.000	781.000	700.900	25.8623	0.0368987
800	715.000	908.000	800.689	27.8283	0.0347555
900	814.000	982.000	900.179	29.3433	0.0325972
1000	901.000	1110.00	1000.11	30.9398	0.0309362



# Implementazione

---

Come si applica il rumore poissoniano ad un sinogramma?

```
for i=0,n_bin ; indice sui bin
```

```
    for j=0, n_angl ; indice sugli angoli
```

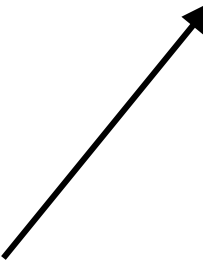
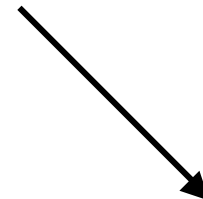
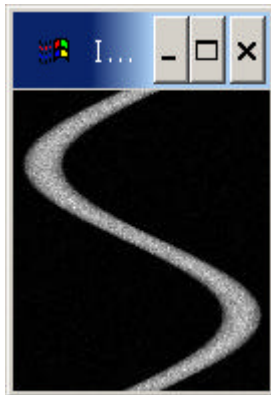
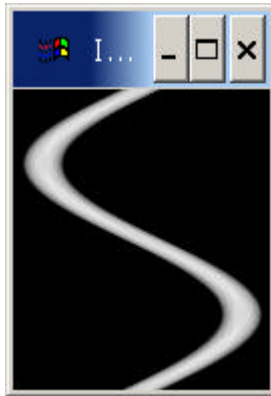
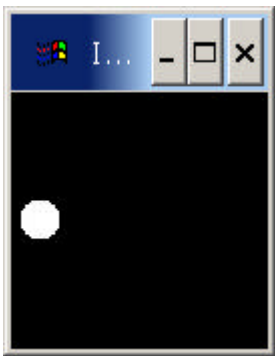
```
        sino_noise[i,j]=randomn(seed,1,poisson=sino[i,j])
```

```
    end
```

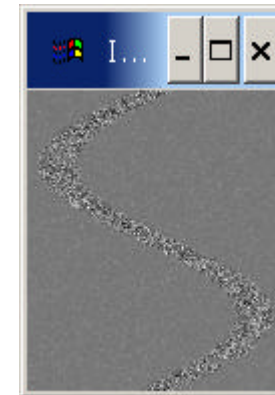
```
end
```

I valori nel sinogramma devono essere **opportunamente scalati**

# Implementazione



-







## Conservazione componente continua

---

Se si opera una simulazione completa con il filtro a rampa e si osserva la ricostruzione si può notare uno "shift" nei valori medi. Questo, che è certamente una fonte di errore per le valutazioni quantitative delle immagini TAC, SPECT etc., è dovuto al fatto che, in Fourier, vengono messe a zero tutte le frequenze che stanno in un intorno della frequenza zero.

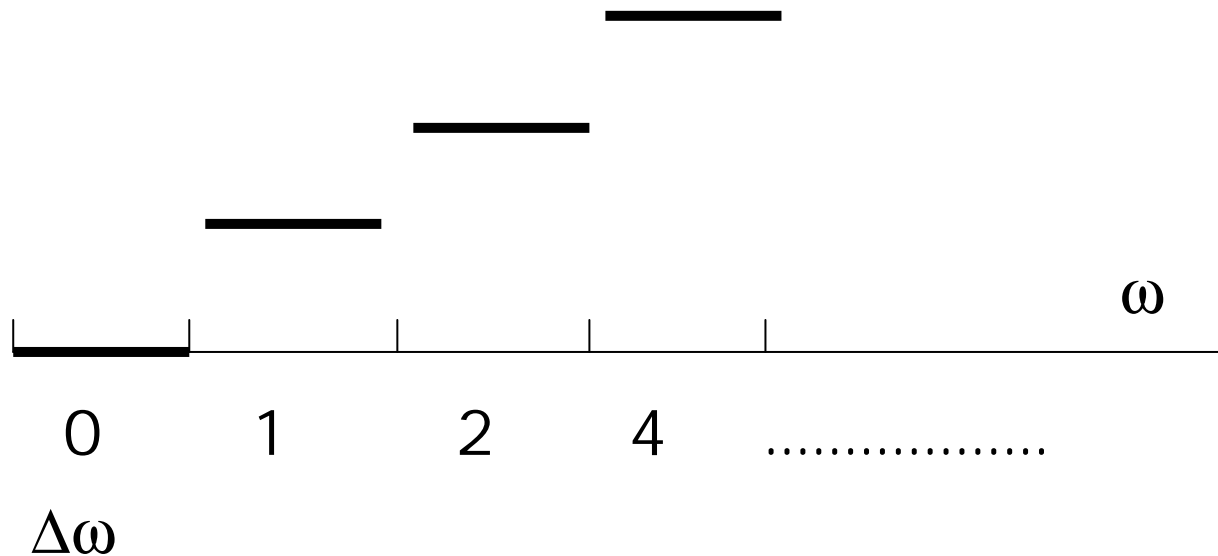
Kak e Slaney paragrafo 3.3.3



# Conservazione componente continua

---

Disegniamo il filtro a rampa nel discreto facendo uno zoom vicino alla frequenza zero:





# Conservazione componente continua

---

Con il filtro a rampa mettiamo a zero non solo la frequenza zero ma tutte le frequenze che stanno nell'intervallo  $[0, \Delta\omega]$ .

Nella ricostruzione questo si ripercuote sul valore medio (componente continua). In pratica si trova un valore leggermente più basso, che può essere fonte di errore nell'analisi quantitativa delle immagini.



## Banda di un segnale e ... dintorni

---

Come ripasso, diamo alcune definizioni che ci serviranno in seguito.

Banda di un segnale è l'intervallo di frequenze presenti nel segnale: avere **banda limitata** vuol dire che le frequenze del segnale sono comprese tra  $[-\Omega, \Omega]$

Dal teorema del campionamento sappiamo **che un segnale che viene campionato ha una banda limitata.**

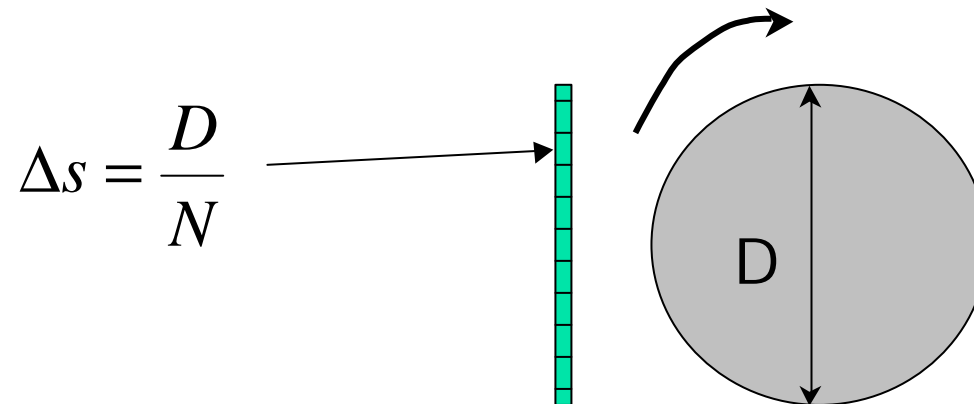
$\Omega$  è determinata dal passo di campionamento, ovvero, nel caso della tomografia, dalla larghezza del rivelatore (o del pixel)



## Banda di un segnale e ... dintorni

---

La larghezza del rivelatore è data dall'estensione fisica della regione sotto osservazione diviso il numero di elementi:



Allora la massima (frequenza) pulsazione contenuta nel segnale così campionato è:

$$\Omega = \frac{\mathbf{p}}{\Delta s} = \frac{\mathbf{p}}{D} N$$



# Campionamento e zero padding

---

Ora osserviamo come vengono campionate le frequenze.  
L'intervallo è  $[-\Omega, \Omega]$  e abbiamo  $N$  punti:

$$\Delta \mathbf{w} = \frac{2\Omega}{N} = \frac{2p}{ND} N = \frac{2p}{D}$$

Se operiamo lo zero padding raddoppiamo  $N$  ma anche  $D$ .  
Prima osservazione **la banda non cambia!!!**  
L'intervallo di campionamento nelle frequenze si dimezza.

$$\Delta \mathbf{w}_z = \frac{\Delta \mathbf{w}}{2}$$



# Filtro RALA

---

Abbiamo visto ora che lo zero-padding aiuta a correggere l'errore sulla componente continua, ciò non basta allora si utilizza un nuovo filtro detto di **Ramachandran-Lakshminarayanan (abbr. RALA)**

Questo filtro si scrive nello spazio ordinario come anti-trasformata continua del filtro a rampa, limitata tra  $[-\Omega, \Omega]$ :

$$H_{\Omega}(s) = \frac{1}{2p} \int_{-\Omega}^{\Omega} |w| e^{-isw} dw$$



# Filtro RALA

---

Questo calcolo si svolge analiticamente e da il seguente risultato:

$$H_{\Omega}(s) = \frac{\Omega^2}{2p} \left[ 2\text{sinc}\left(\Omega \frac{s}{p}\right) - \text{sinc}^2\left(\Omega \frac{s}{2p}\right) \right]$$

dove  $H$  è detto nucleo (kernel) del filtro.

Si può notare che per calcolare questo filtro occorre calcolare la banda ovvero fissare il campionamento.



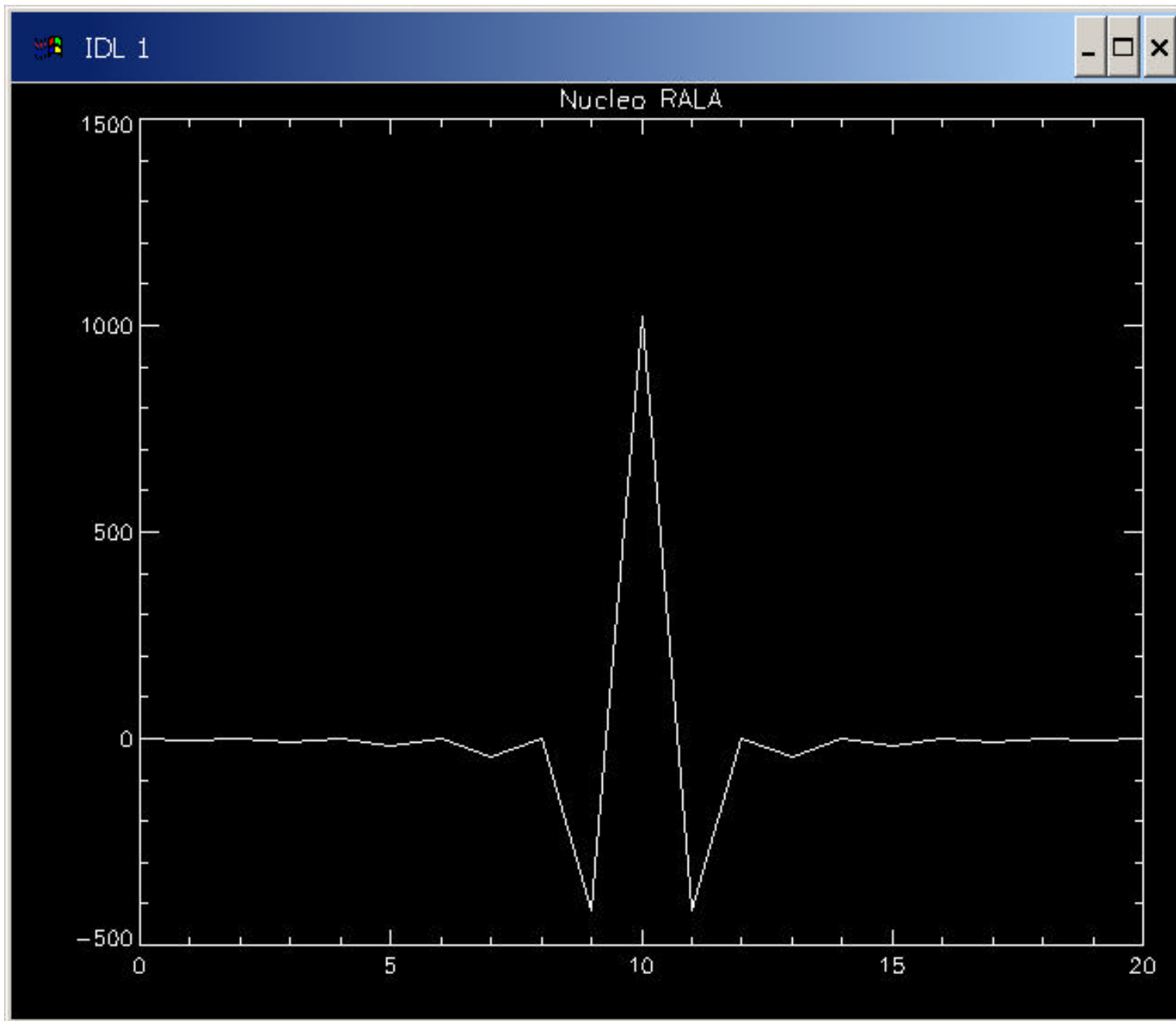


# Filtro RALA

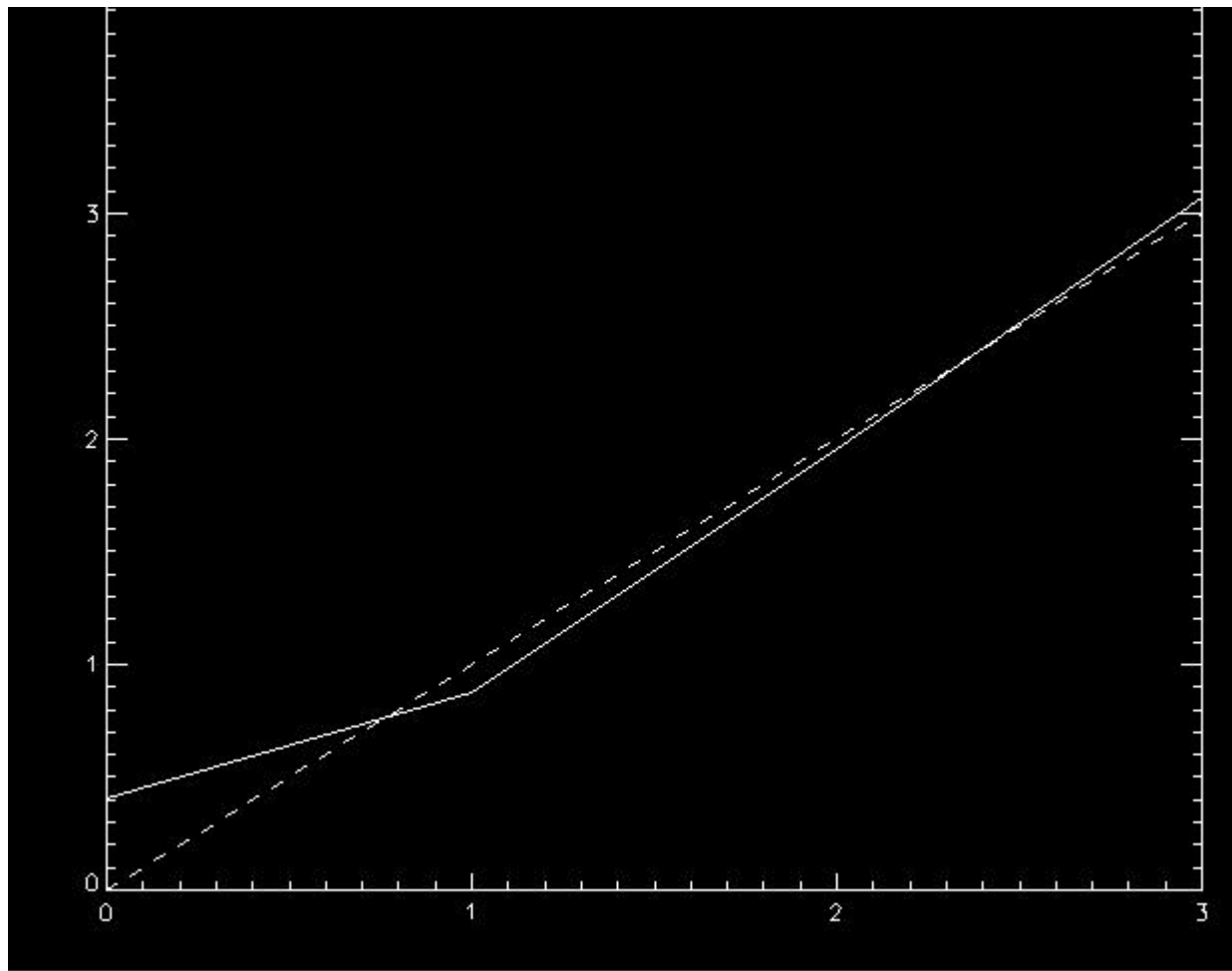
---

La versione discreta del nucleo è

$$H_{\Omega}(m\Delta s) = \begin{cases} 0 & \text{per } m \neq 0 \text{ e pari} \\ \frac{\Omega^2}{2\mathbf{p}} = \frac{1}{2\Delta s^2} & \text{per } m = 0 \\ \frac{-\Omega^2}{\mathbf{p}^3 m^2} = \frac{-1}{\mathbf{p} m^2 \Delta s^2} & \text{per } m \text{ dispari} \end{cases}$$



# Differenza RALA- RAMPA



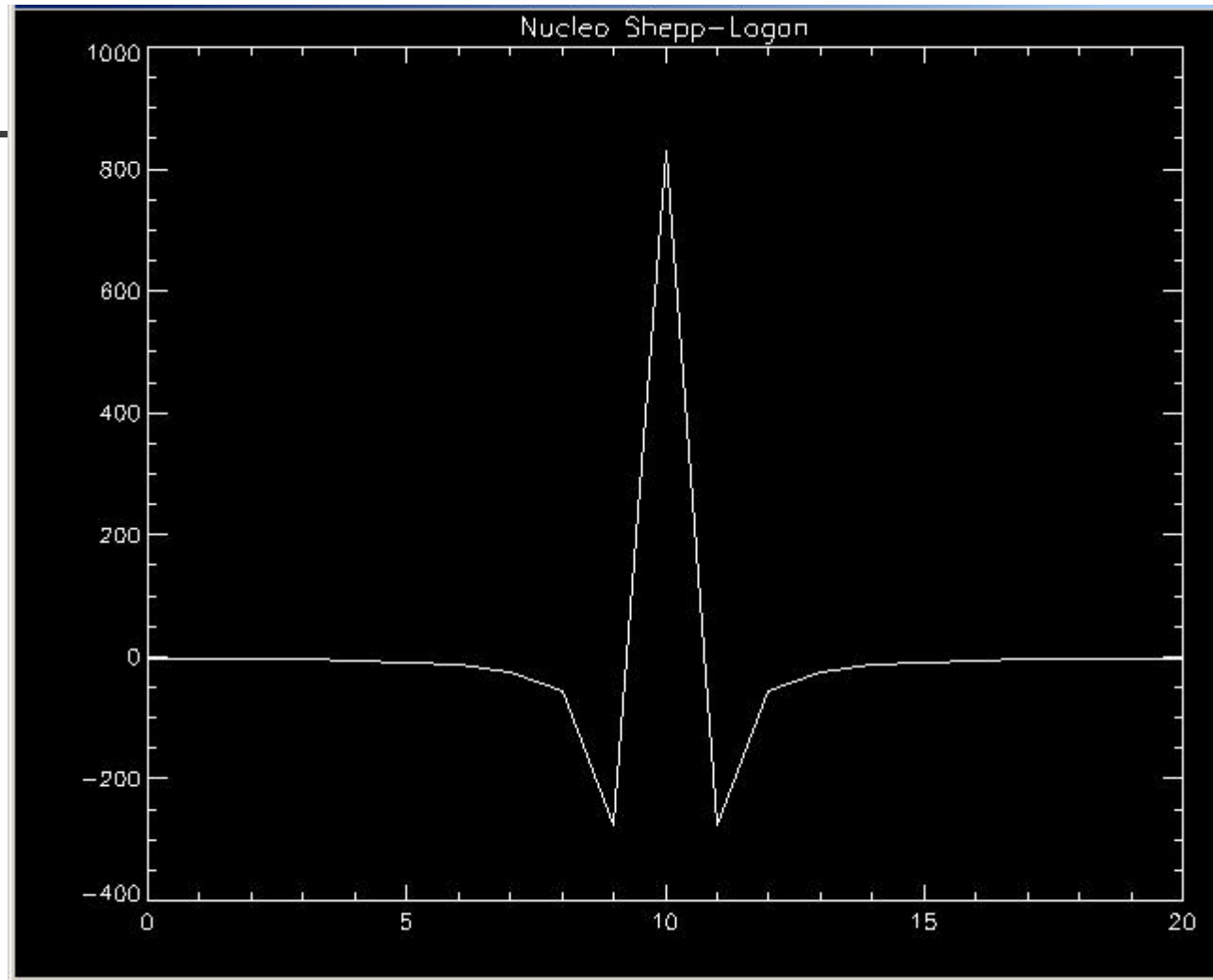
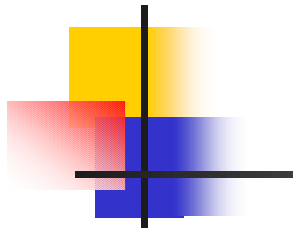


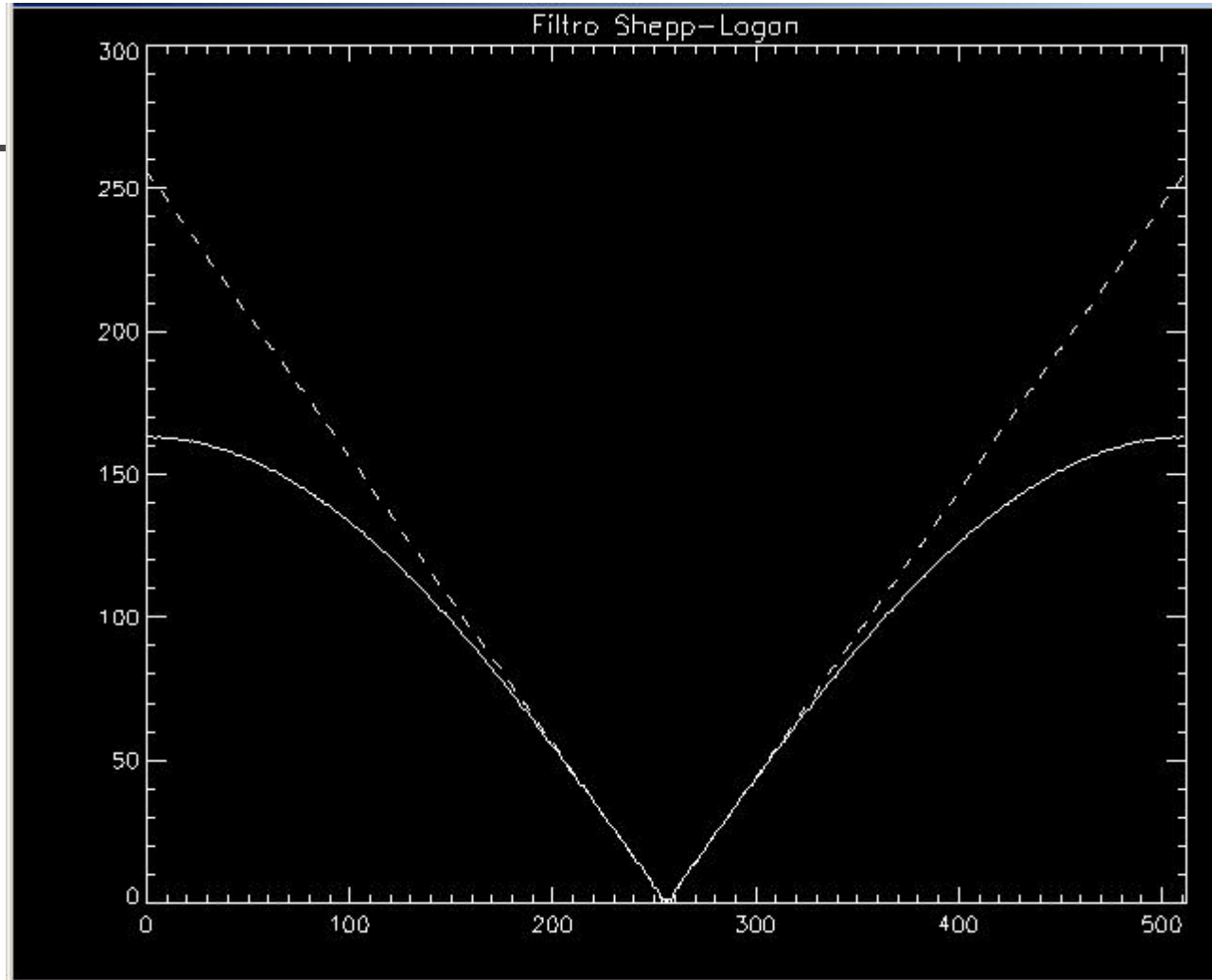
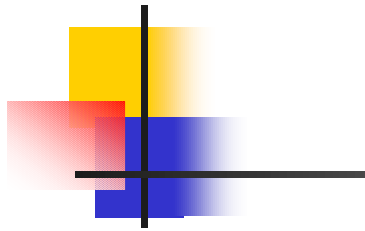
# Filtro di Shepp- Logan

---

Il filtro RALA ha un altro difetto alle frequenze alte amplifica il rumore quindi si usano altri filtri che per le frequenze basse assomigliano al RAMPA ma filtrano maggiormente quelle alte. Uno di questi è il filtro Shepp e Logan.

$$S_{\Omega}(m\Delta s) = -\frac{4}{p^3} \frac{\Omega^2}{4m^2 - 1} = -\frac{4}{p\Delta s^2} \left( \frac{1}{4m^2 - 1} \right)$$







# Altri filtri

---

A volte si applicano filtri passa basso moltiplicati per il filtro a rampa: Hamming, Hann, Parzen, Butterworth, tutti disegnati sulla banda.

$$\text{Hamming} \quad \left\{ \begin{array}{ll} 0.5 + 0.5 \cos\left(\frac{pw}{\Omega}\right) & \text{per } |w| \leq \Omega \\ 0 & \text{per } |w| > \Omega \end{array} \right.$$

$$\text{Hann} \quad \left\{ \begin{array}{ll} 0.5 + 0.46 \cos\left(\frac{pw}{\Omega}\right) & \text{per } |w| \leq \Omega \\ 0 & \text{per } |w| > \Omega \end{array} \right.$$



# Altri filtri

---

Parzen

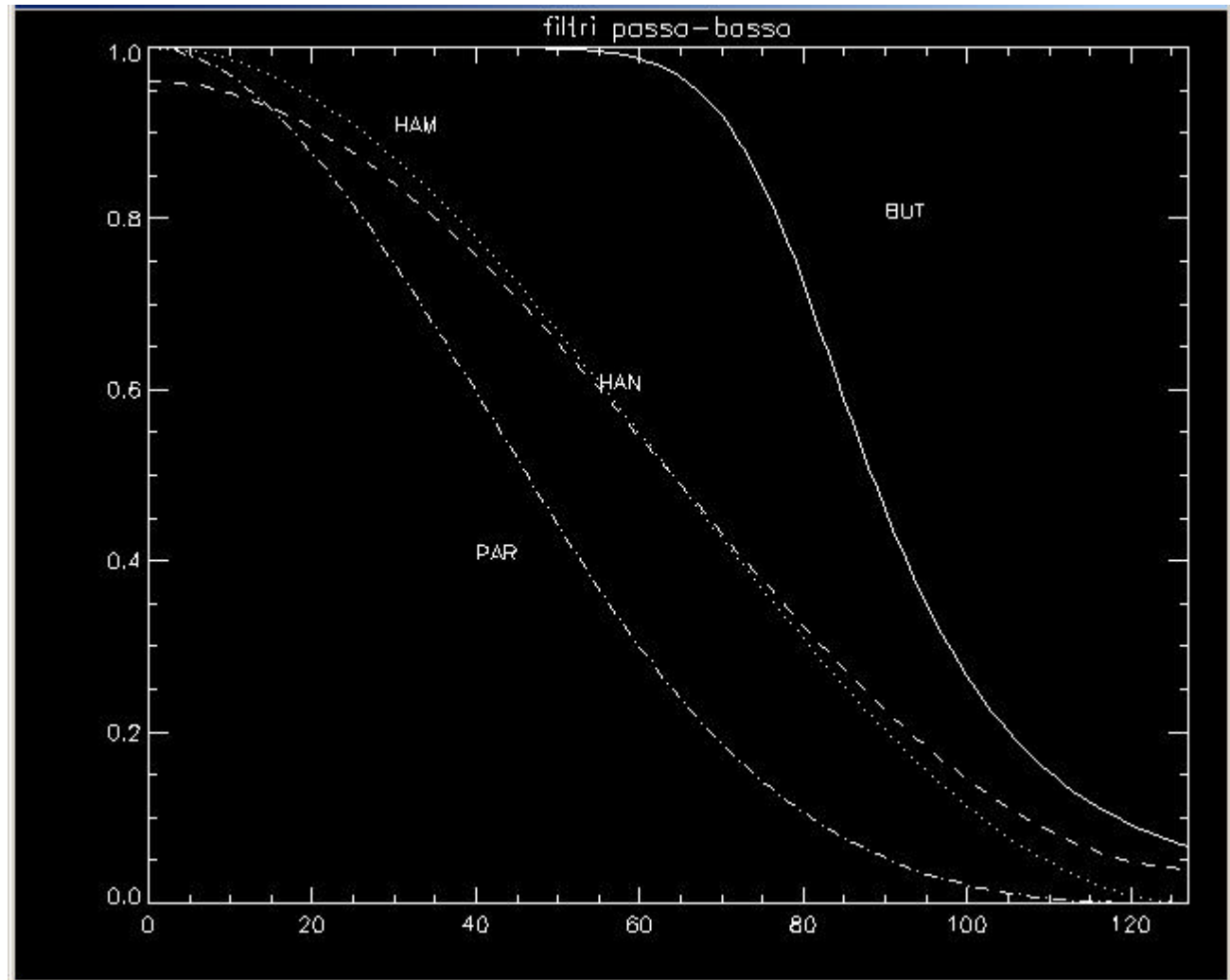
$$\begin{cases} 1 - 6 \left( \frac{|\mathbf{w}|}{\Omega} \right)^2 \left( 1 - \frac{|\mathbf{w}|}{\Omega} \right) & \text{per } |\mathbf{w}| \leq \Omega / 2 \\ 2 \left( 1 - \frac{|\mathbf{w}|}{\Omega} \right)^3 & \text{per } \Omega / 2 < |\mathbf{w}| \leq \Omega \\ 0 & \text{per } |\mathbf{w}| > \Omega \end{cases}$$

Butterworth

$$\frac{|\mathbf{w}|}{\sqrt{1 + \left( \frac{\mathbf{w}}{\mathbf{w}_s} \right)^{2n}}}$$



# Altri filtri



# Altri filtri

