

# Effective Return Loss (ERL)

CLASS PROJECT DEL CORSO  
LABORATORIO DI SIGNAL INTEGRITY



Studenti:

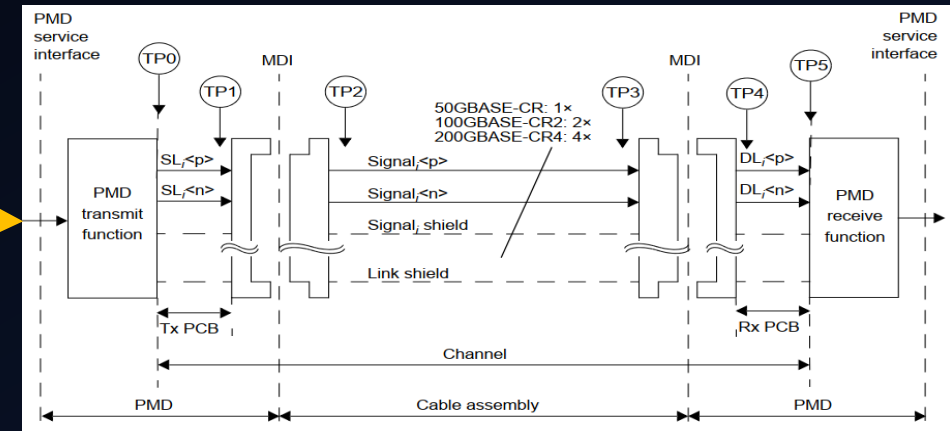
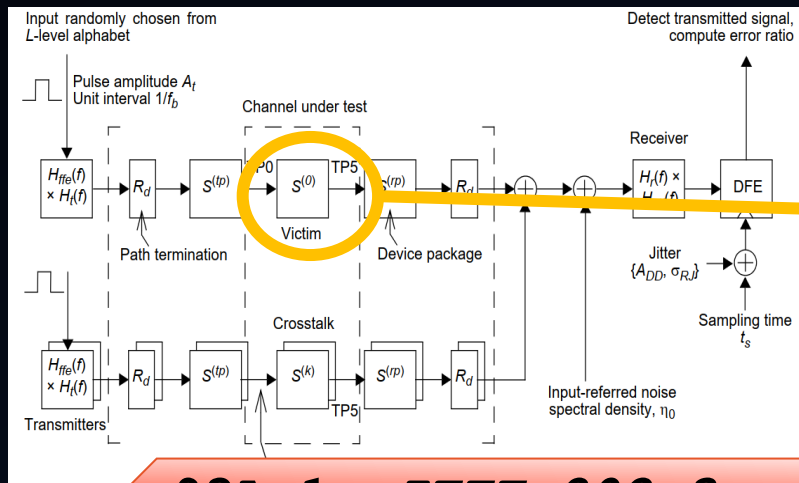
Mattia Ragnoli  
Davide Colaiuda

Professore:

Francesco De Paulis

# QUAL È LA DIFFERENZA TRA RL ED ERL?

- Il RL rappresenta la sezione fisica di canale (TP0-TP5) quindi  $S_{ii}(f)$ ;
- Progettare SOLO tramite maschera su  $S_{ii}(f)$  può dare **constrain troppo rigidi** (*non si considera equalizzazione*);
- L'ERL è una *fom* che rappresenta la quantità di segnale riflessa all'interfaccia tra dispositivi o canali, includendo gli effetti dell'equalizzazione; [\(nomenclatura\)](#)



Esempio TP0-TP5 link 50GBASE-CR, 100GBASE-CR2, 200GBASE-CR4

93A-1, IEEE 802.3 cd

vedi APP.1

# STEP PRINCIPALI NEL CALCOLO DELL'ERL

Calcolo dei parametri S filtrati

Calcolo della PTDR(t)

Calcolo della deviazione standard

Calcolo della PDF e della CDF

Risultato ERL



# PARAMETRI PER IL CALCOLO DELL'ERL

Parametro	Simbolo	Unità
Signaling rate	$f_b$	GBd
Tempo di transizione dell'impulso	$T_r$	ns
Banda a 3dB del ricevitore	$f_r$	GHz
Livelli del segnale	$L$	—
Lunghezza del segnale riflesso	$N$	UI
Numero di sample per UI	$M$	—
Equalizer length	$N_{bx}$	UI
Signal Loss Factor	$\beta_x$	GHz
Riflessione permessa da linee esterne al DUT	$\rho_x$	—
Detector Error Ratio	$DER_0$	—

## Esempi dal codice:

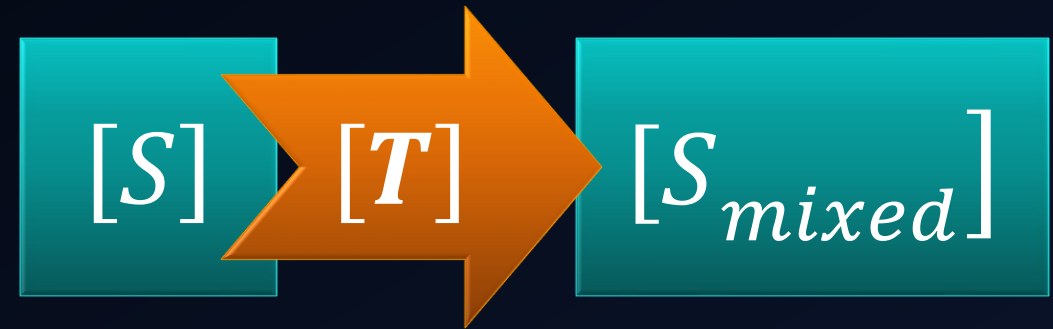
```
param.specBER =  
xls_parameter(parameter, 'DER_0');  
  
param.ndfe =  
xls_parameter(parameter, 'N_b');  
  
param.samples_per_ui =  
xls_parameter(parameter, 'M', 32);  
  
-com_ieee8023_93a.m
```

# LETTURA $[S]$ ED ESTRAZIONE $[S_{DD}]$

- Parametri  $[S]$  misurati  $\rightarrow$  Conversione in  $[S_{diff}]$

- $[S_{mixed}] = [T] [S] [T]^{-1}$

- $[T] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$



- $[S_{mixed}] = \begin{bmatrix} S_{cc11} & S_{cd11} & S_{cc12} & S_{cd21} \\ S_{dc11} & S_{dd11} & S_{dc12} & S_{dd12} \\ S_{cc21} & S_{cd21} & S_{cc22} & S_{cd22} \\ S_{dc21} & S_{dd21} & S_{dc22} & S_{dd22} \end{bmatrix}$

# LETTURA [S] ED ESTRAZIONE [S<sub>DD</sub>]

```
%% FUNCTION :: read_sp4_sparams  
% Description  
%   Read the fid of single-ended 4-port complex S-parameters  
%   in Touchstone format 'file' and convert to the internal format
```



# CALCOLO DEI PARAMETRI S FILTRATI

- Conversione in  $[S_{diff}] \rightarrow$  Calcolo  $H_{ii}(f)$

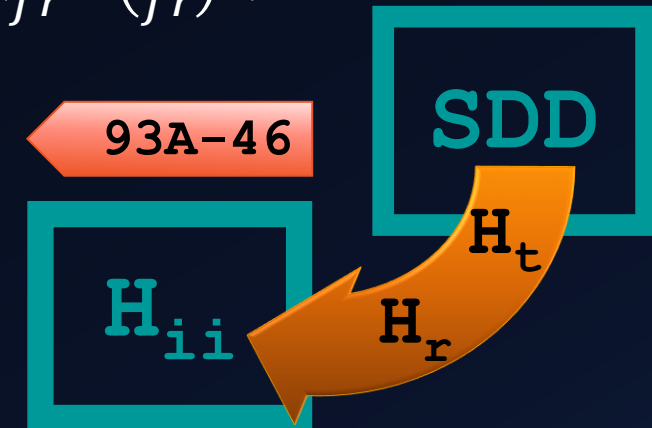
- $H_{ii}(f) = H_t(f) \cdot S_{dd,ii}(f) \cdot H_r(f) \quad i = 1,2$  93A-58

- Filtro in ricezione:

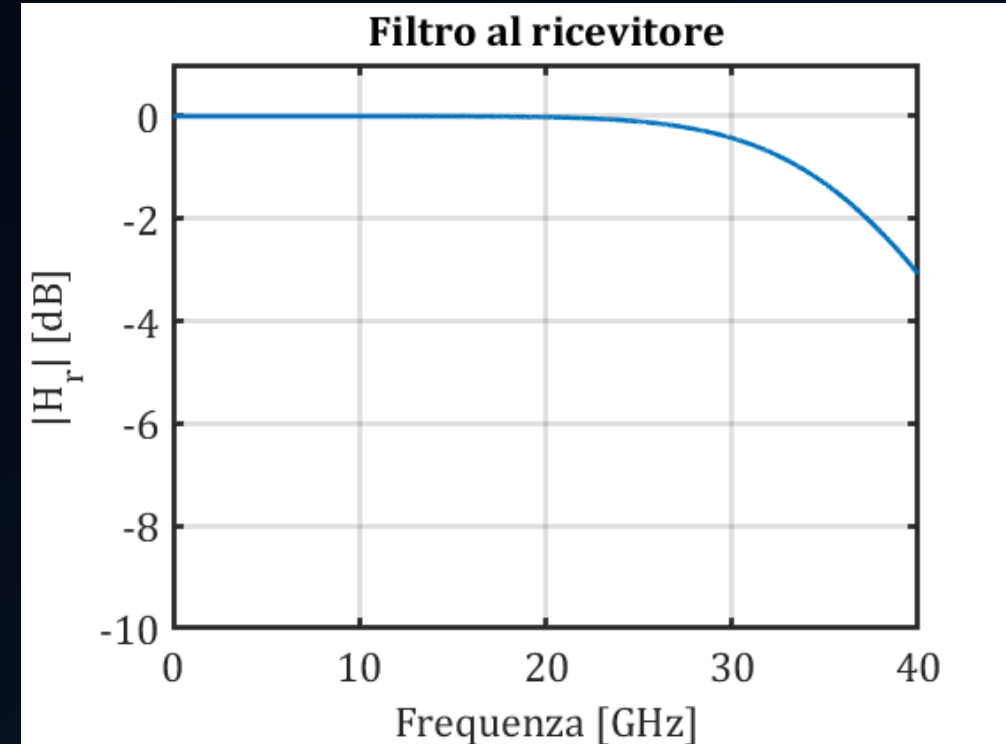
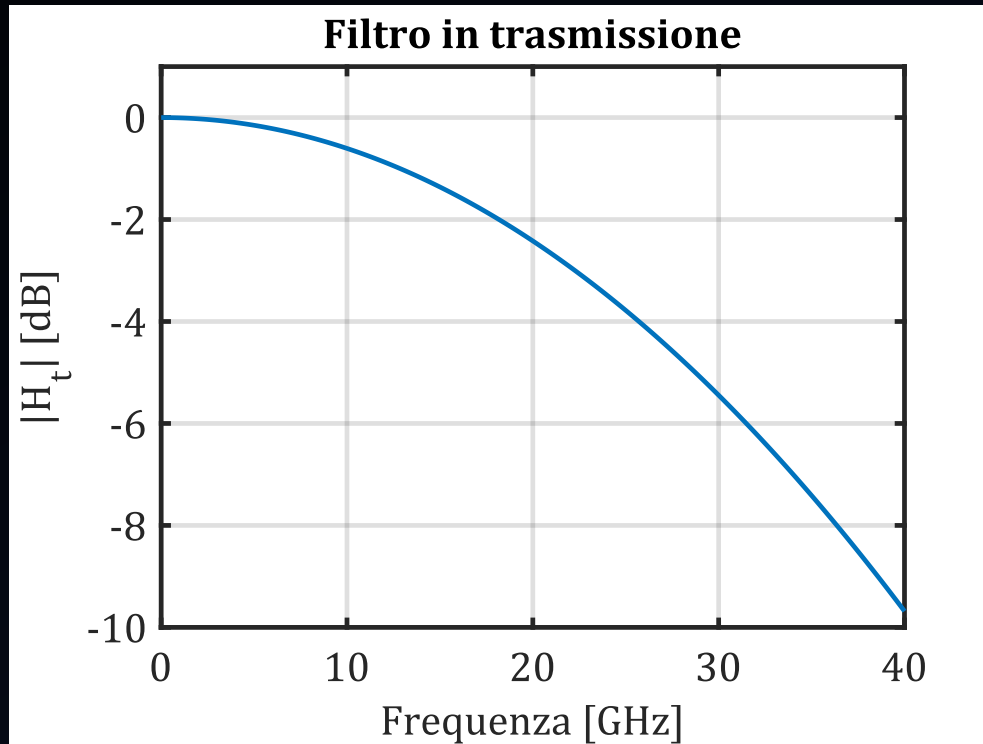
$$H_r(f) = \frac{1}{1 - 3.414214 \left(\frac{f}{f_r}\right)^2 + \left(\frac{f}{f_r}\right)^2 + j2.613126 \left(\frac{f}{f_r} - \left(\frac{f}{f_r}\right)^3\right)}$$
93A-20

- Filtro al trasmettitore:  $H_t(f) = e^{-2\left(\frac{\pi f T_r}{1.6832}\right)^2}$  93A-46

- $T_r$  tempo dal 20% all'80% della transizione



# CALCOLO DEI PARAMETRI S FILTRATI



Dai parametri S filtrati si ricava la riflettometria (TDR)

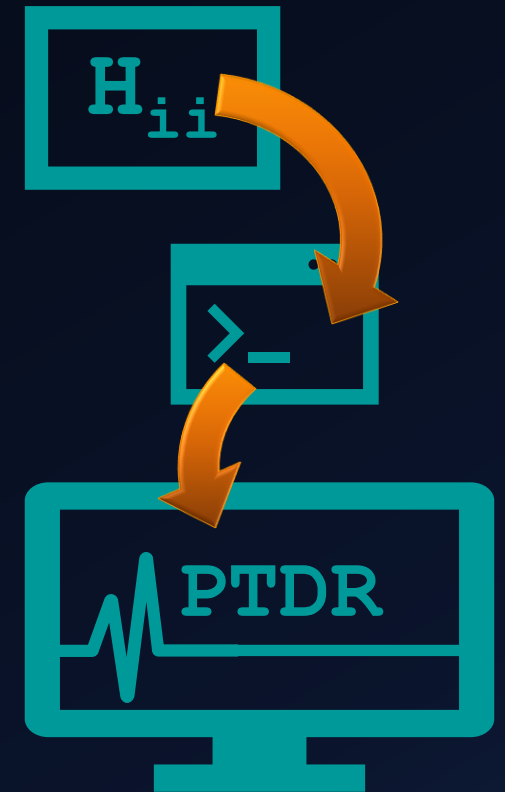


# PTDR

Calcolo  $H_{ii}(f) \rightarrow$  Calcolo della *PTDR*

- Rappresenta la risposta dovuta alla combinazione di riflessioni a stimulate dalla trasmissione di **un impulso**;
- $PTDR(t) = \int_{-\infty}^{\infty} X(f) H_{ii}(f) e^{j2\pi ft} df$  93A-59
- Dato l'impulso  $X(f) = A_t T_b \text{sinc}(fT_b)$  93A-23
- $T_b = \frac{1}{f_b}$  larghezza impulso,  $A_t$  ampiezza impulso;
- Nel Tool COM viene calcolata anche la  $TDR(t)$ . vedi APP.2

```
function TDR_results = get_TDR(sdd, OP, param, ZT, np)
```



# COME SI OTTIENE LA PTDR DAL TDR?

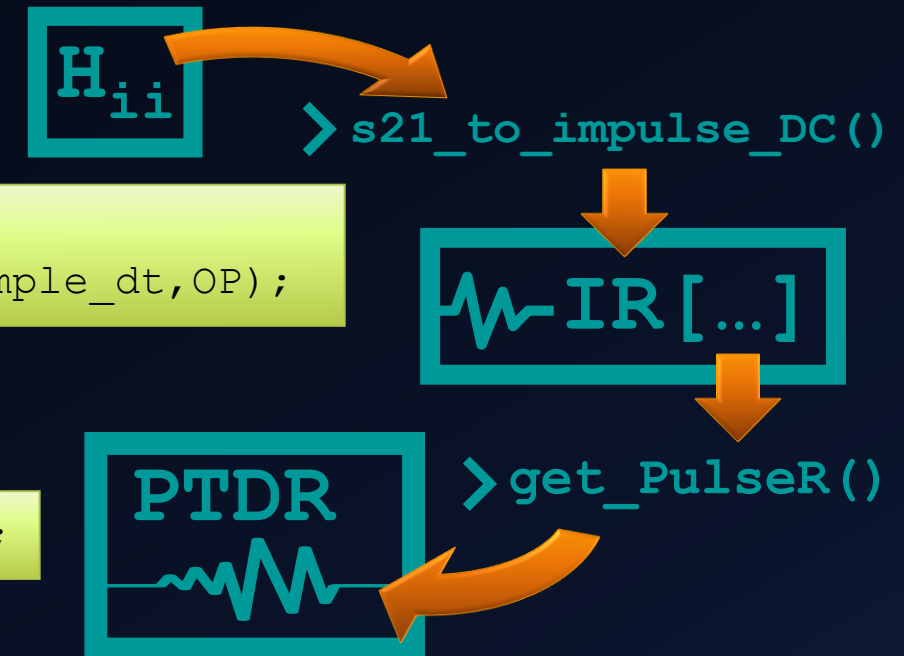
- Nel codice viene implementata la funzione get\_TDR;
- All'interno della funzione vengono richiamate subroutine ausiliarie;

$[H_{ij}] \rightarrow$  Impulse Response (IR)

```
[IR, t, causality_correction_dB, truncation_dB] =  
s21_to_impulse_DC(RLf,  sdd.Frequencies(:), param.sample_dt, OP);
```

Impulse Response (IR)  $\rightarrow$  PTDR

```
PTDR=get_PulseR(IR(tstart:tend), param, OP.cb_step, ZT);
```



```
function result=get_PulseR(ir,param,cb_step,ZT)
%ir = impulse response
%t_base=time array with equal time steps
%samp_UI = number of samples per UI for ir
% t for debug
t=(1/param.fb)/param.samples_per_ui*(0:length(ir)-1);
```

```
if cb_step
```

```
    Ag=1;
    dt=1/param.fb/param.samples_per_ui;
    edge_time=param.TR_TDR*1e-9;
    fedge=1/edge_time;
    tedge=0:dt:edge_time*2;
    %
    edge=Ag*(2*cos(2*pi*(tedge)*fedge/16-pi/4).^2-1);
    drive_pulse=[edge ones(1,param.samples_per_ui)];

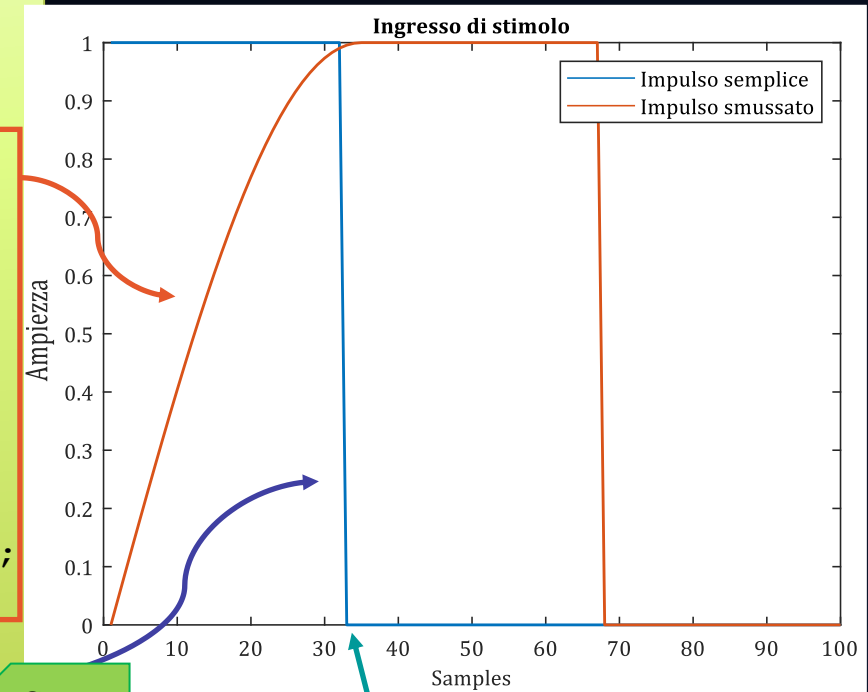
    t=(1/param.fb)/param.samples_per_ui*(0:length(ir)-1);
    pulse=filter(drive_pulse,1,ir);
```

```
else
```

```
    pulse=filter(ones(1,param.samples_per_ui),1,ir);
```

```
end
```

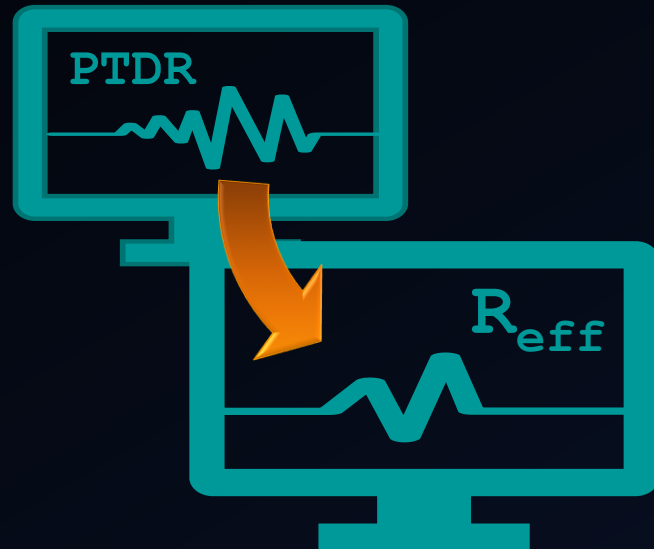
```
PDR_response=(1+pulse)./(1-pulse).*ZT*2;
result.PDR=PDR_response;
result.pulse=pulse;
```



1 UI

# EFFECTIVE REFLECTION WAVEFORM

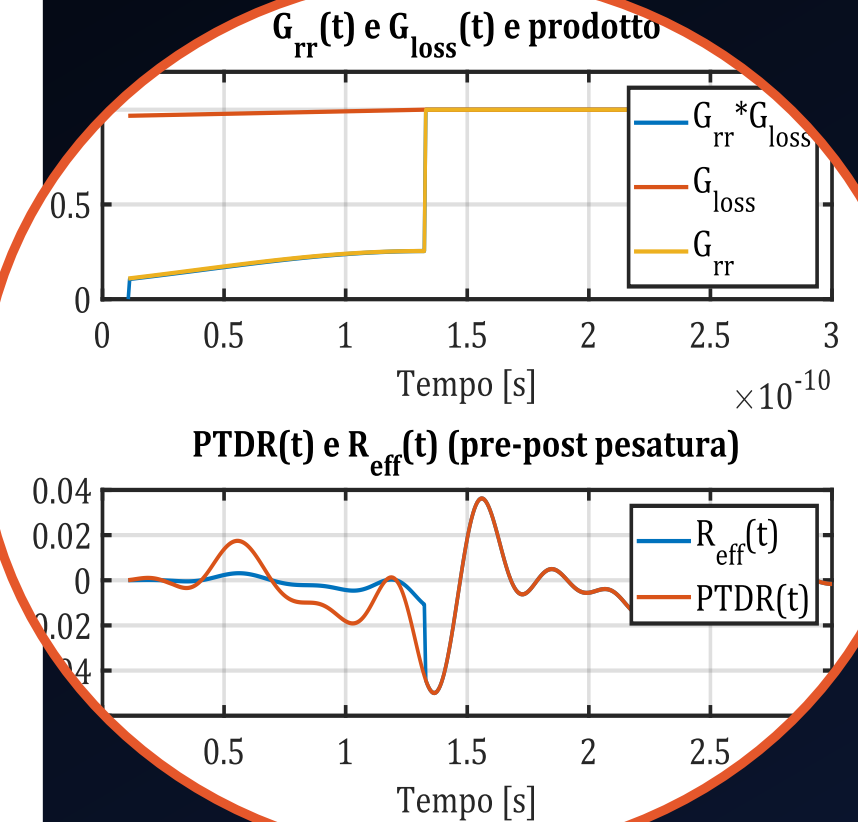
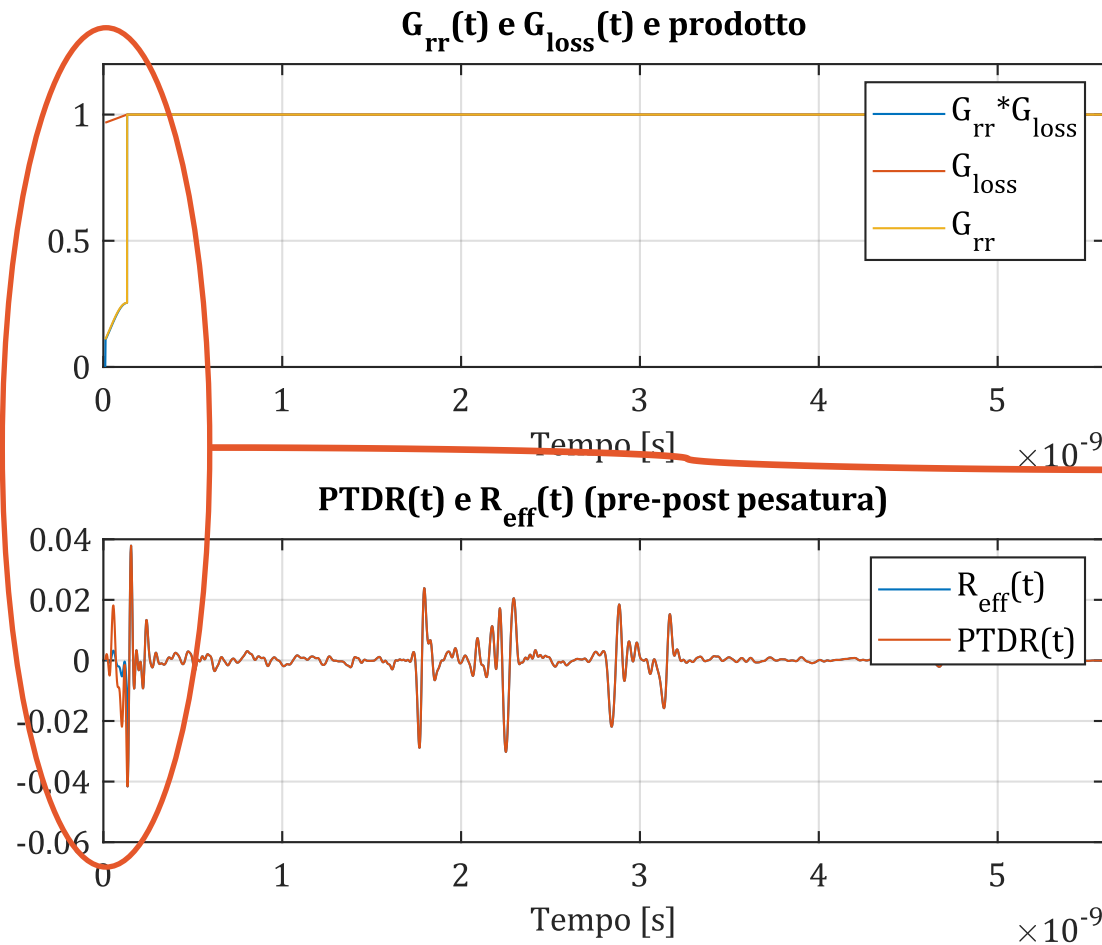
- Possiamo calcolare quindi la PTDR equalizzata;
- $R_{eff}(t) = PTDR(t) \cdot G_{rr}(t) \cdot G_{loss}(t)$  93A-60;



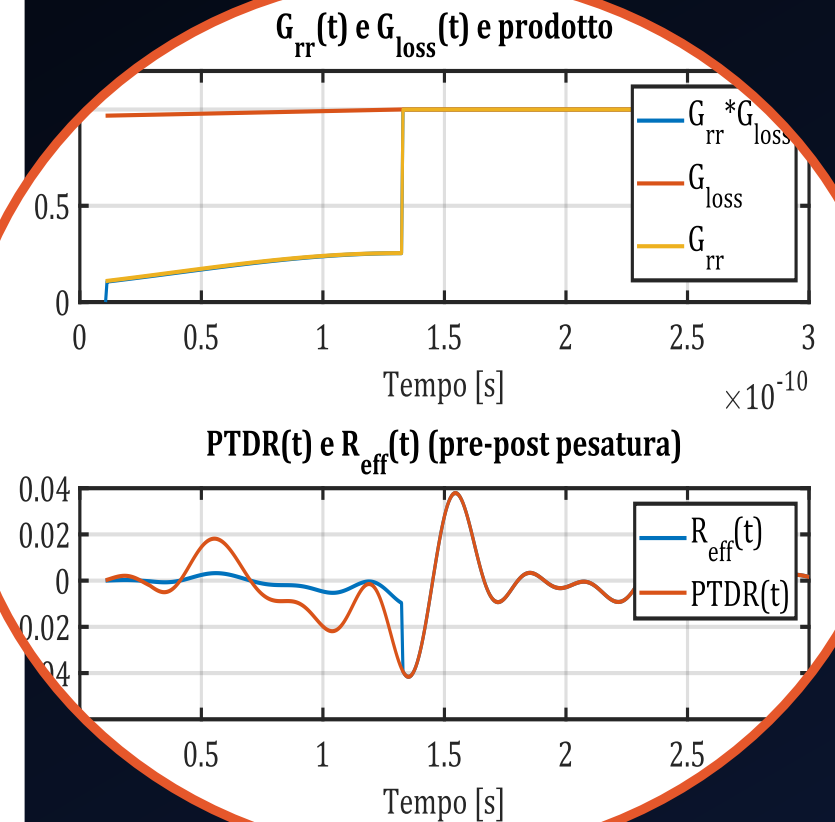
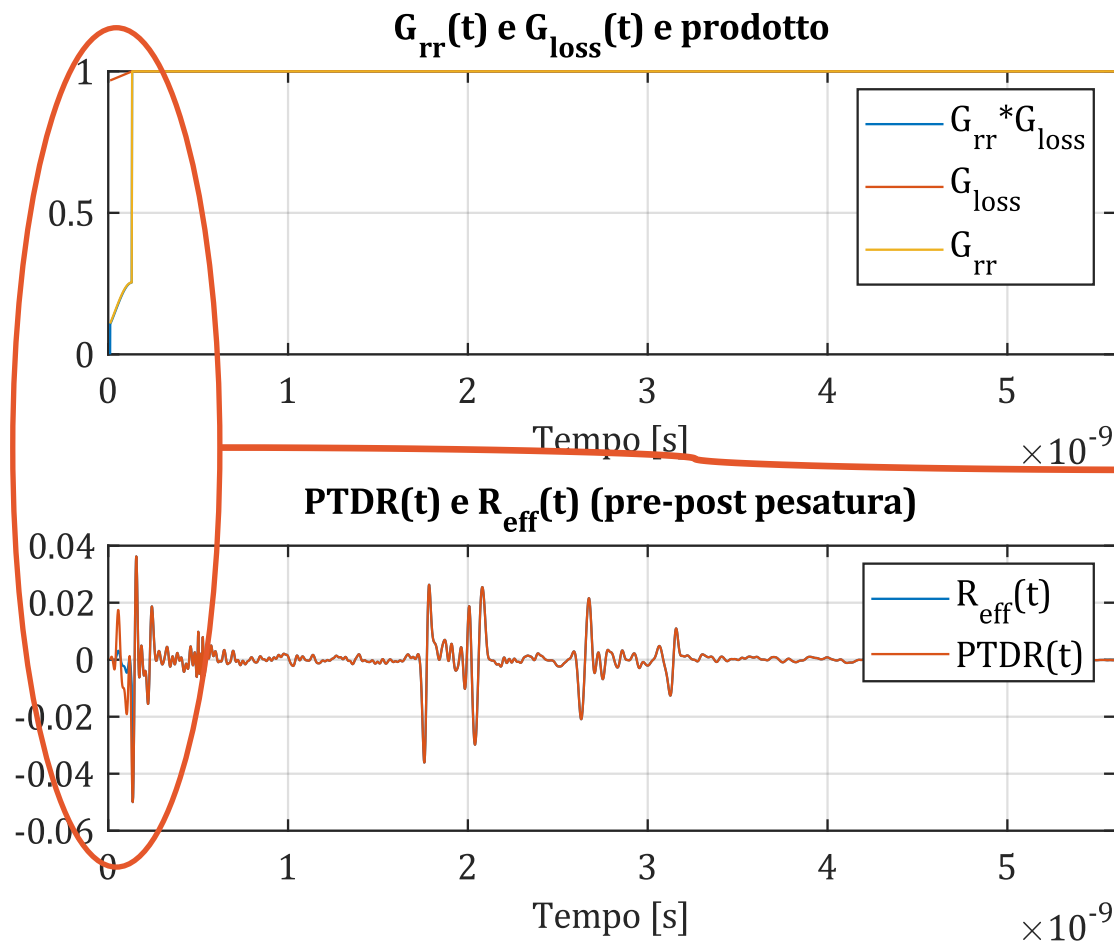
- $G_{rr}$ : effetto dell'eq., elimina riflessioni per la durata del DFE
- $G_{loss}$ : effetto dell'IL a smorzare le riflessioni vedi APP.3
- Applicate entrambe per  $N_b$  UI

```
fctrx(ii) = Gloss(ii).*Grr(ii);
          [...]
PTDR.pulse = PTDR.pulse.*fctrx;
```

# EFFETTO DELLA PESATURA ( $S_{dd,11}$ )



# EFFETTO DELLA PESATURA ( $S_{dd,22}$ )



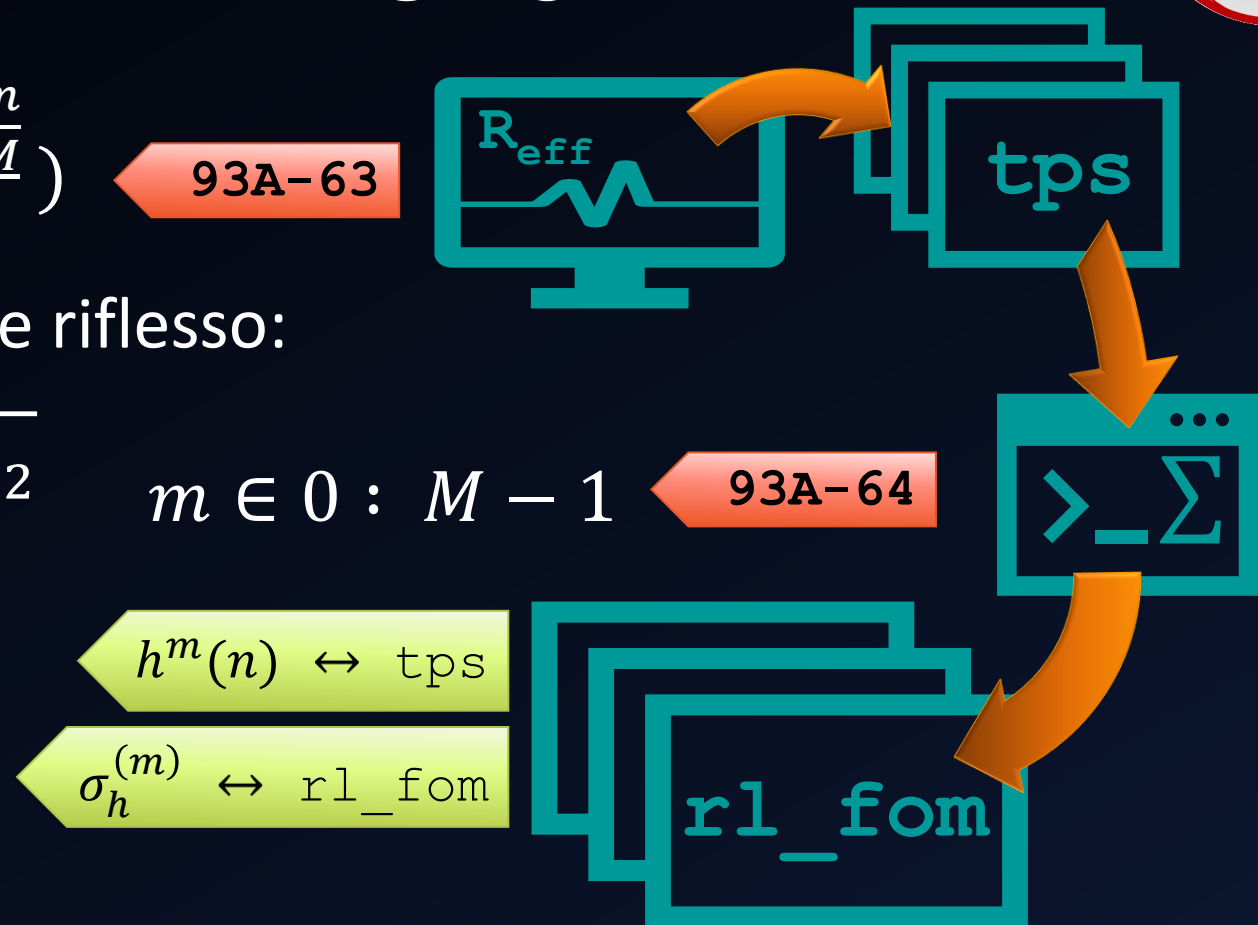
# SAMPLED EFFECTIVE REFLECTION

- $h^m(n) = R_{eff} \left( T_{fx} + \frac{n + \frac{m}{M}}{f_b} \right)$  93A-63

- Deviazione std del segnale riflesso:

$$\sigma_h^{(m)} = \sqrt{\sum_{n=1}^N h^m(n)^2} \quad m \in 0 : M - 1$$
93A-64

- $N$ : numero UI
- $M$ : samples per UI





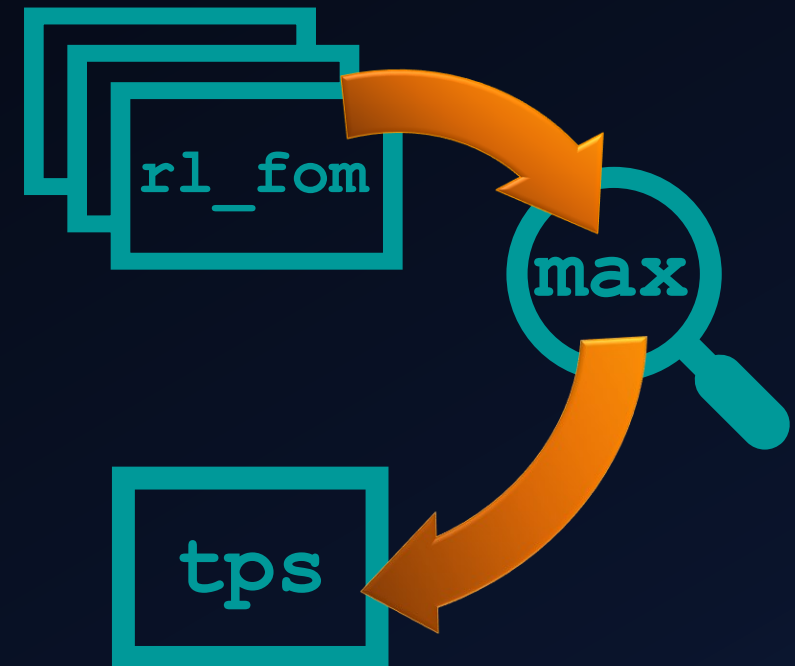
# SCELTA DI $m$ (SAMPLES PER UI)

- Si utilizza la  $h^{(m)}(n)$  a cui è associata la  $\sigma_h^{(m)}$  massima  $\rightarrow m_{best}$
- $h(n) = h^{(m_{best})}(n)$

PTDR  $\rightarrow h(n)$

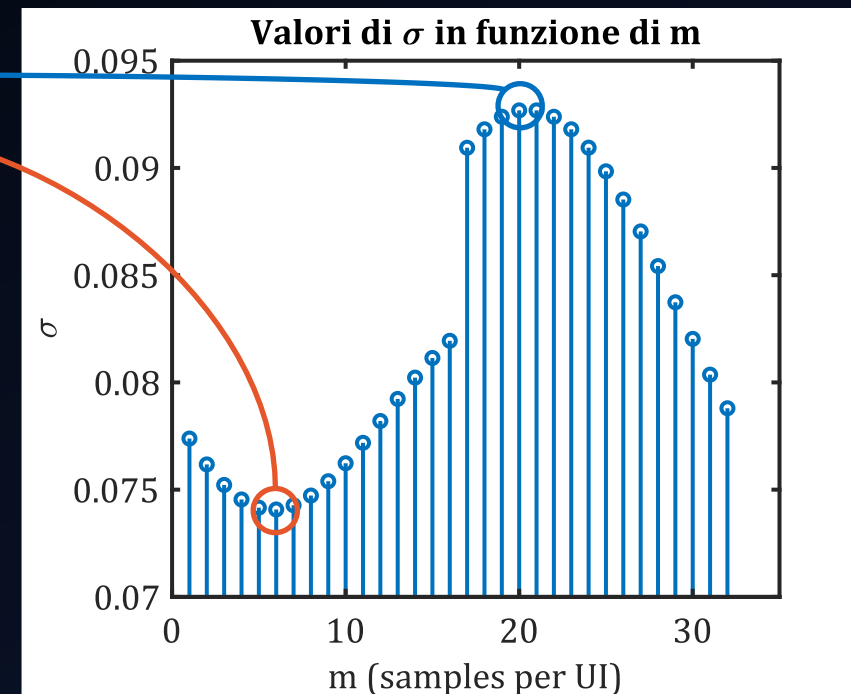
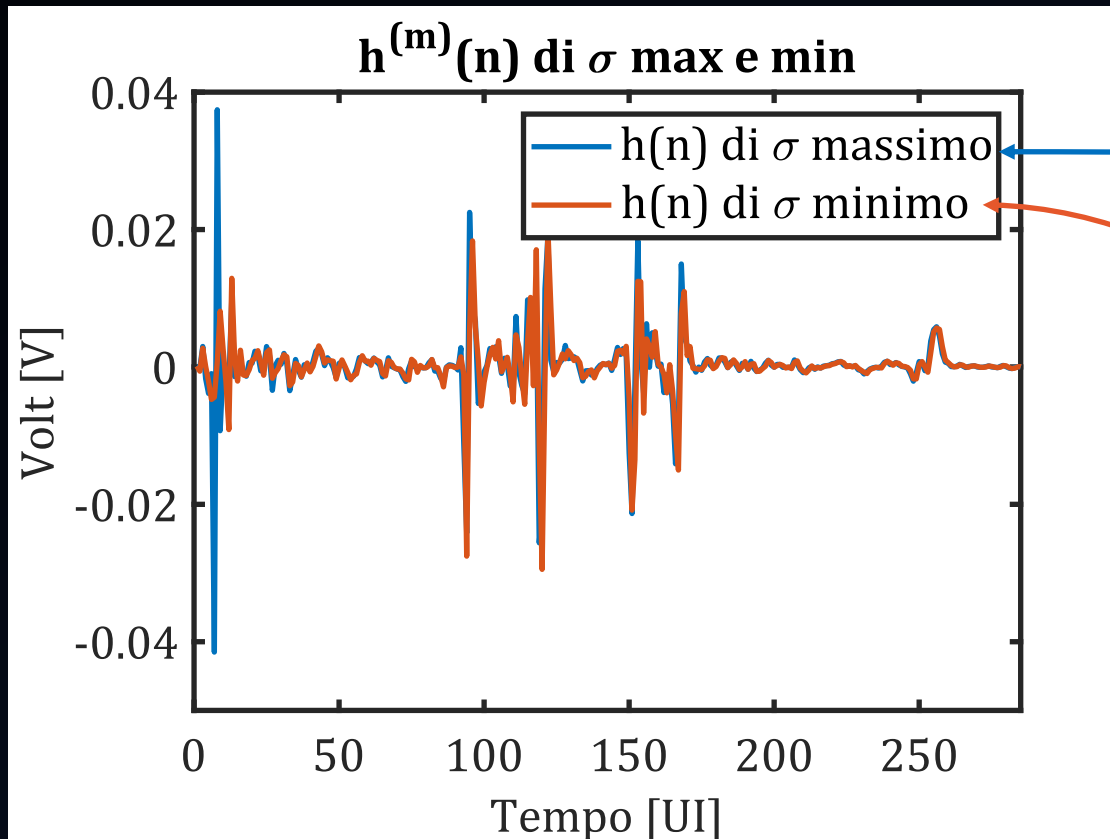
$m \leftrightarrow ki$

```
for ki=1:param.samples_per_ui
    tps=PTDR.pulse(ki:param.samples_per_ui:end);
    rl_fom=(norm(tps));
    if rl_fom > RL_equiv;
        RL_equiv=rl_fom;
        best_ki=ki;
    [...]
tps=PTDR.pulse(best_ki:param.samples_per_ui:end);
```





# $h^{(m)}(n)$ AL VARIARE DEI SAMPLES PER UI



# INTERFERENCE AMPLITUDE DISTRIBUTION

- Trovata  $h(n)$  si ricavano  $N$  *interference amplitude distribution*:

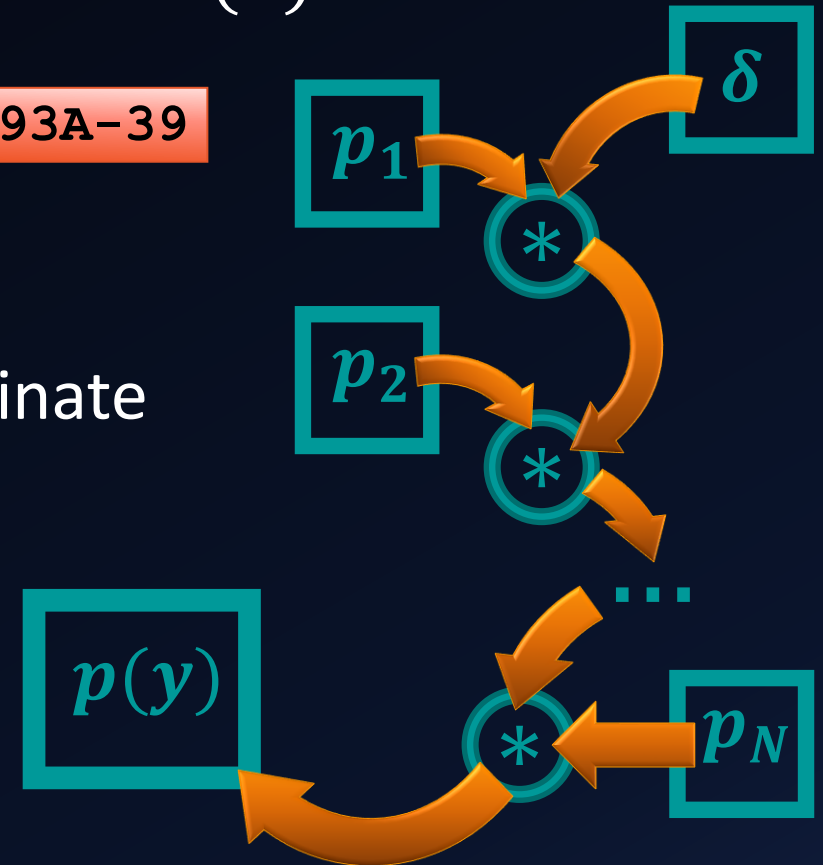
$$p_n(y) = \frac{1}{L} \sum_{l=0}^{L-1} \delta \left( y - \left( \frac{2l}{L-1} - 1 \right) h(n) \right) \quad \text{93A-39}$$

- $L$ : possibili valori per simboli i.i.d.
- Le  $N$  componenti così calcolate sono combinate per convoluzioni iterative in questo modo:

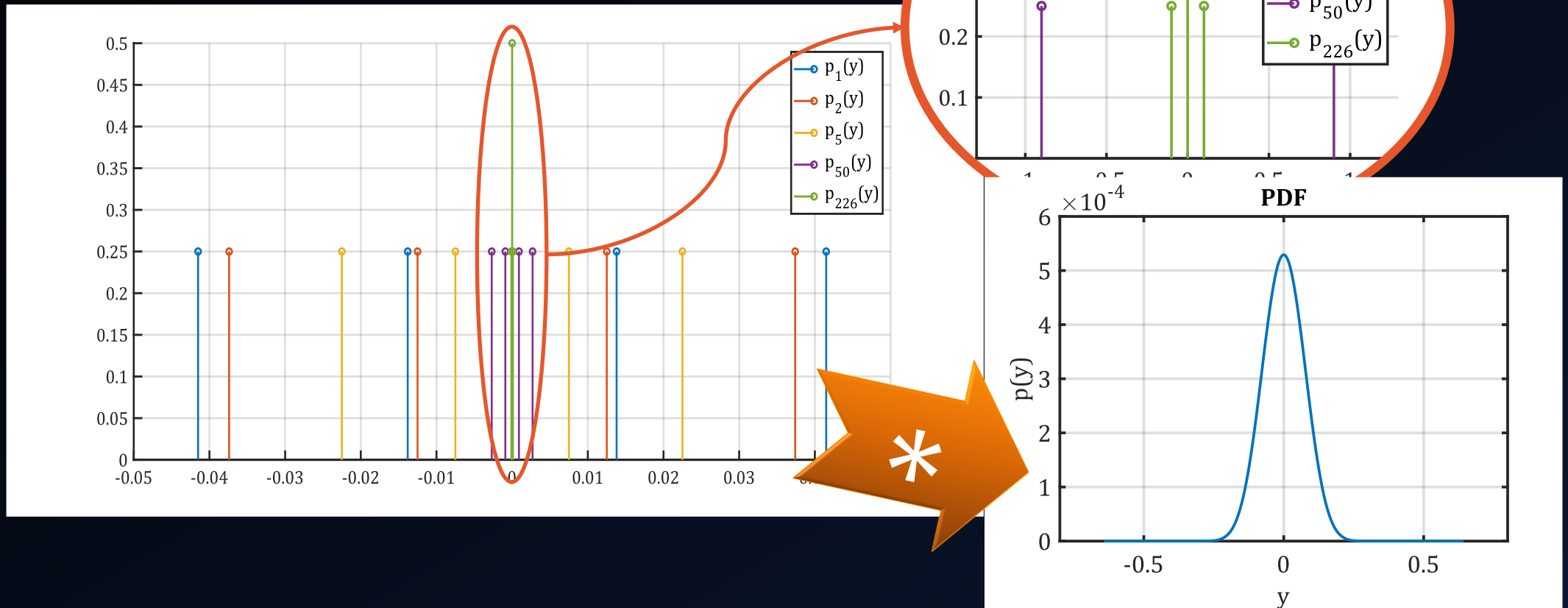
$$p(y) = p(y) * p_n(y) \quad \text{93A-40}$$

(per  $p_0(y)$  la convoluzione è con una delta)

$h(n) \rightarrow \text{PDF}$



# DALLE IAD ALLA PDF



# CALCOLO DELLA PDF E DELLA CDF

```
testpdf=get_pdf_from_sampled_signal(tps,L, BinSize*10, FAST_NOISE_CONV);
[...]
```

%% Equation 93A-39 %%

```
values = 2*(0:L-1)/(L-1)-1;
prob = ones(1,L)/L;
%% Initialize pdf to delta at 0
pdf=d_cpfd(BinSize, 0, 1);
for k = 1:length(input_vector)
    pdfn=d_cpfd(BinSize, abs(input_vector(k))*values, prob);
    pdf=conv_fct(pdf, pdfn);
end
[...]
```

```
cdf_test=cumsum(testpdf.y);
```

PDF → CDF

$\delta(y) \leftrightarrow \text{pdf}$

$p_n(y) \leftrightarrow \text{pdfn}$



# CDF E RELAZIONE CON DER

- la CDF si ricava integrando la PDF così ottenuta:

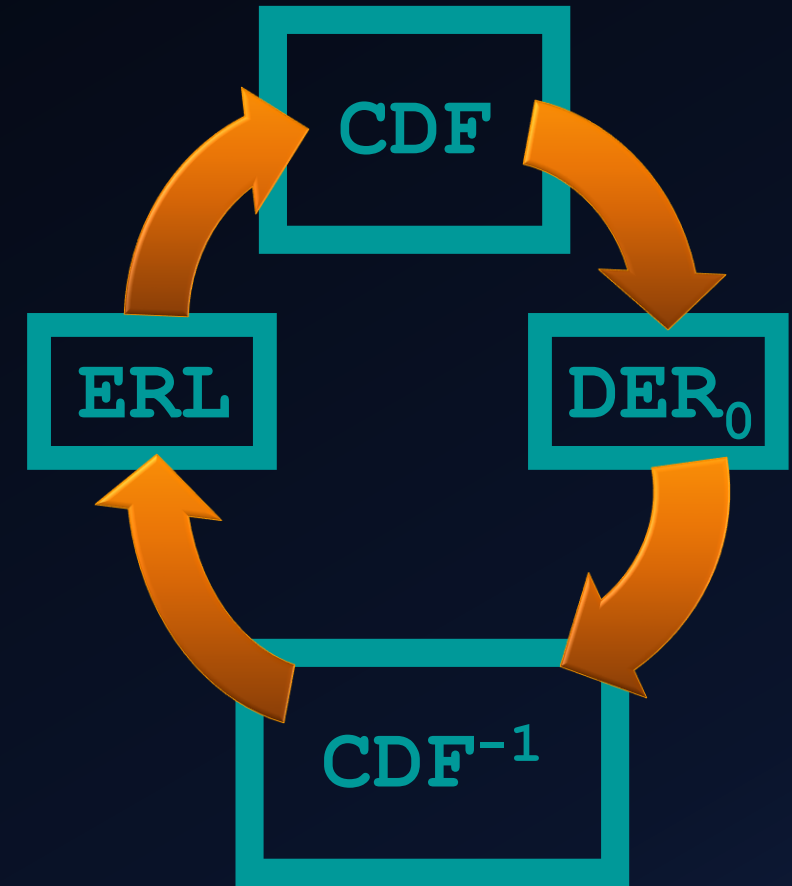
$$P(y) = \int_{-\infty}^y p(y) dy$$

93A-37

- Il quantile  $x$  della distribuzione  $P(y)$  è la quantità di  $y$  tale che:

$$P(y) = x$$

- Il  $DER_0$  è il quantile della distribuzione per  $y = ERL$



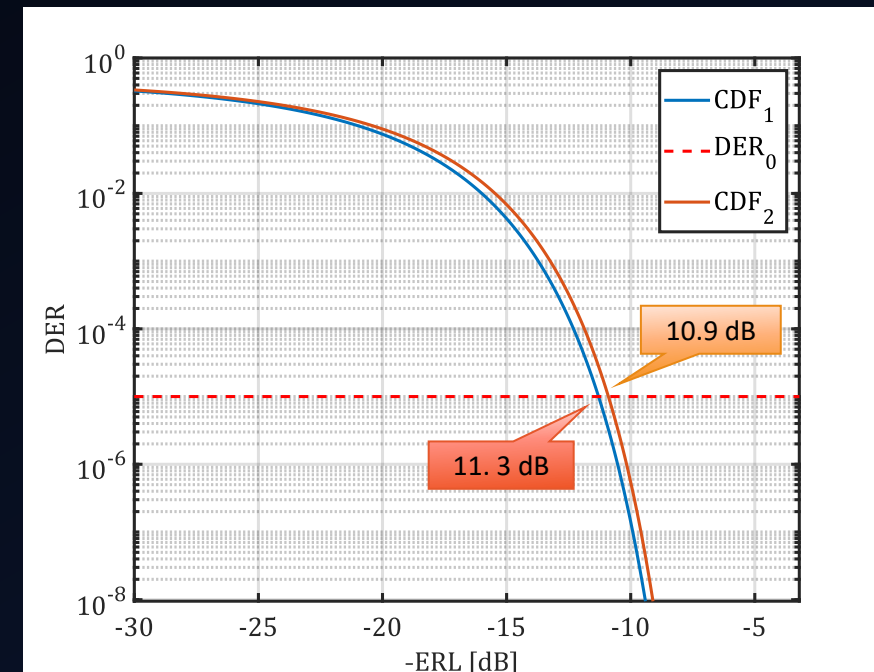
# CALCOLO DELL'ERL

- Trovata la CDF, l'ERL è definito come:  

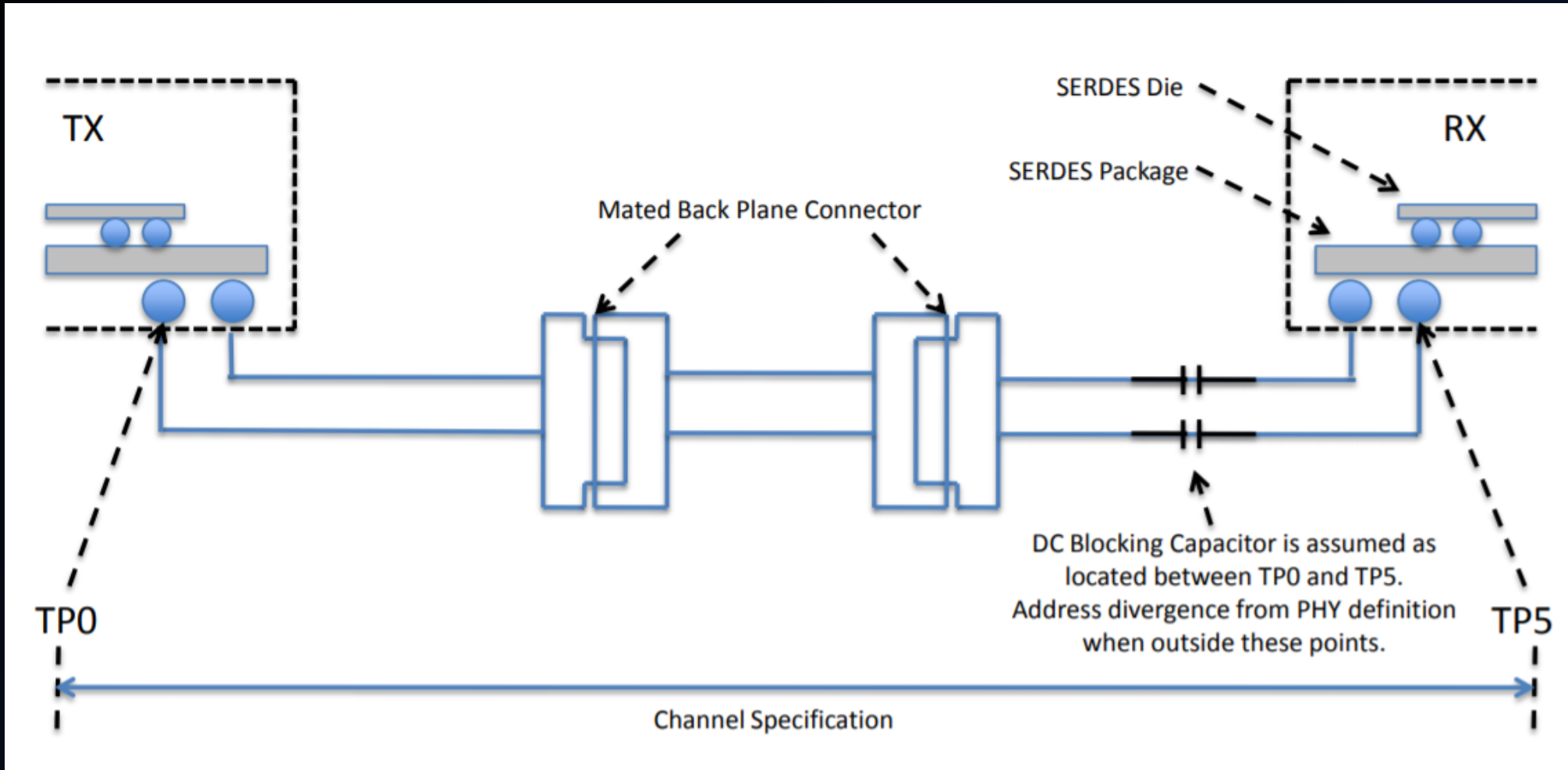
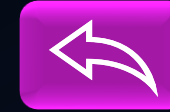
$$\text{ERL} = -20 \log_{10} P^{-1}(\text{DER}_0)$$
- È il valore (in dB) per il quale la CDF diventa uguale al  $\text{DER}_0$
- L'ERL considerato è il più basso (caso peggiore)



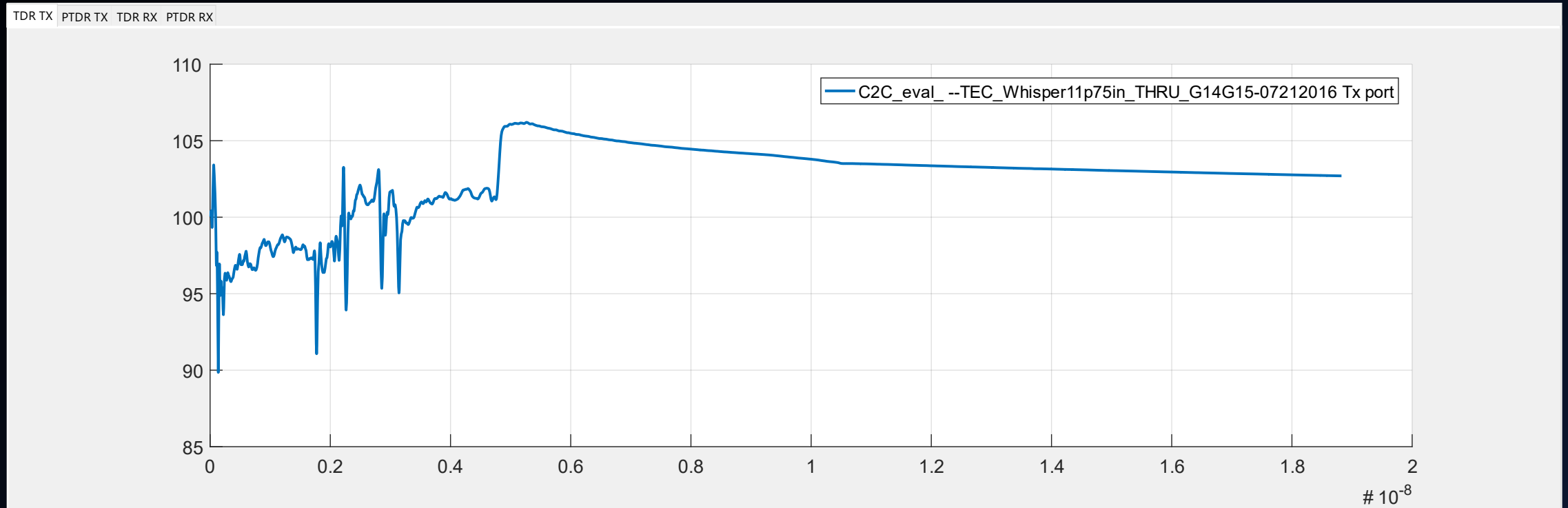
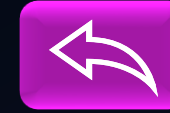
CDF → ERL



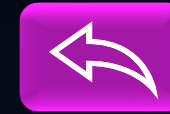
# APPENDICE 1



# APPENDICE 2







# APPENDICE 3

$$G_{rr}(t) = \begin{cases} 0 & \text{per } t \leq T_{fx} \\ \rho_x(1 + \rho_x) \exp\left(\frac{[(t - T_{fx})f_b - (N_{bx} + 1)]^2}{(N_{bx} + 1)^2}\right) & \text{per } T_{fx} \leq t \leq T_{fx} + \frac{N_{bx} + 1}{f_b} \\ 1 & \text{per } t \geq T_{fx} + \frac{N_{bx} + 1}{f_b} \end{cases}$$

$$G_{loss}(t) = \begin{cases} 0 & \text{per } t \leq T_{fx} \\ 10^{\frac{\frac{\beta_x}{f_b}[(t - T_{fx})f_b - (N_{bx} + 1)]}{20}} & \text{per } T_{fx} \leq t \leq T_{fx} + \frac{N_{bx} + 1}{f_b} \\ 1 & \text{per } t \geq T_{fx} + \frac{N_{bx} + 1}{f_b} \end{cases}$$

DFE (Decision Feedback Equalizer)