



**Autore: Francesco Marino**  
<http://www.francescomarino.net>  
[info@francescomarino.net](mailto:info@francescomarino.net)

Esercitazione n. 8

## Determinazione della FFT di segnali corrispondenti a vari codici di linea con MATLAB

Classe:

Gruppo:

Data	Alunni assenti

La presente esercitazione riguarda la determinazione con l'ausilio di MATLAB della FFT per diverse forme d'onda corrispondenti a diversi codici di linea, in modo tale da poter verificare quanto esposto nel Vol. 2, Unità didattica 7, in merito agli spettri dei codici stessi. Il segmento di riferimento su cui calcolare la FFT ha una durata pari a 8 cicli di clock. Le forme d'onda possono essere costruite nel modo più semplice possibile, ovvero codificando quanti meno bit è possibile per ottenere una forma d'onda tipica del dato codice. Di seguito sono elencati i bit da codificare per ottenere una forma d'onda tipica:

- NRZ, RZ e Manchester: un solo bit di valore «1»;
- bifase differenziale: assimilabile al Manchester;
- codice AMI: due «1»;
- HDB-3: assimilabile all'AMI;
- CMI: assimilabile al Manchester;
- MLT-3: quattro «1» o sequenza «100111» (nel secondo caso si evidenzia la presenza di una componente continua)
- 2B1Q: sequenza «00-10-01-11» (anche in questo caso cambiando la sequenza comparirebbe una componente continua);
- 4B3T: sequenza «0010»;
- MS43: assimilabile al 4B3T.

La forma d'onda può essere costruita su 128 punti, e in tal modo ogni bit corrisponderà a 16 punti ( $16 = 128/8$ ). Per la costruzione della forma d'onda si può inizialmente utilizzare con il comando

```
v=zeros(1,N);
```

che assegna il valore «0» agli  $N$  punti del vettore  $v$  (con  $N = 128$ ). Successivamente si provvederà a cambiare il valore di alcune serie di punti per costruire la forma d'onda. Per esempio, per codificare un bit «1» in Manchester si può utilizzare il seguente codice:

```
N=128;
v=zeros(1,N);
v(1:8)=-1;
v(9:16)=1;
```

Per la determinazione della FFT e del relativo grafico si rimanda al Vol.2, Unità Didattica 1, par. 1.3 e all'esempio ivi riportato. Nella pagina seguente è riportato, ad uso del docente e anche, se ritenuto da questi opportuno, degli alunni, il codice per ottenere le FFT delle diverse forme d'onda e il relativo output (ottenuto ridimensionando la finestra grafica).

```

N=128;
T=8;
t=linspace(0,T,N);
f=(0:N-1)/N*N/T;
ck(1:8)=1;
ck(8:16)=0;
ck(17:32)=ck(1:16);
ck(33:64)=ck(1:32);
ck(65:128)=ck(1:64);
subplot(9,2,1)
plot(t,ck)
axis([-inf inf -1.2 1.2])
ylabel('clock')
title('Forma d\'onda')

```

```

%NRZ
v=zeros(1,N);
v(1:16)=1;
F=fft(v);
subplot(9,2,3)
plot(t,v)
axis([-inf inf -1.2 1.2])
ylabel('NRZ')
subplot(9,2,4)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])
title('DFT')

```

```

%RZ
v(9:16)=0;
F=fft(v);
subplot(9,2,5)
plot(t,v)
axis([-inf inf -1.2 1.2])
ylabel('RZ')
subplot(9,2,6)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])

```

```

%MANCHESTER-CMI
v(1:8)=-1;
v(9:16)=1;
F=fft(v);
subplot(9,2,7)
plot(t,v)
axis([-inf inf -1.2 1.2])
ylabel('Manch.-CMI')
subplot(9,2,8)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])

```

```

%AMI
v(1:8)=1;
v(9:16)=0;
v(17:24)=-1;
F=fft(v);
subplot(9,2,9)
plot(t,v)
axis([-inf inf -1.2 1.2])
ylabel('AMI')
subplot(9,2,10)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])

```

```

%MLT-3
v(1:8)=0;
v(9:24)=1;
v(25:40)=0;
v(41:56)=-1;
F=fft(v);
subplot(9,2,11)
plot(t,v)
axis([-inf inf -1.2 1.2])
ylabel('MLT-3')
subplot(9,2,12)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])

```

```

%MLT-3 (2)
v(1:8)=0;
v(9:56)=1;
v(57:72)=0;
v(73:88)=-1;
F=fft(v);
subplot(9,2,13)
plot(t,v)
axis([-inf inf -1.2 1.2])
ylabel('MLT-3 (2)')
subplot(9,2,14)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])

```

```

%2B1Q
v=zeros(1,N);
v(1:32)=-1;
v(33:64)=1;
v(65:96)=-1/3;
v(96:128)=1/3;
F=fft(v);
subplot(9,2,15)
plot(t,v)
axis([-inf inf -1.2 1.2])
ylabel('2B1Q')
subplot(9,2,16)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])

```

```

%4B3T
v=zeros(1,N);
v(1:21)=1;
v(22:43)=-1;
F=fft(v);
subplot(9,2,17)
plot(t,v)
axis([-inf inf -1.2 1.2])
xlabel('t (ms)')
ylabel('4B3T')
subplot(9,2,18)
plot(f,abs(F*T/N))
axis([-inf 4 -inf inf])
xlabel('f (kHz)')

```

