

# Sistemi di elaborazione e trasmissione delle informazioni - Laboratorio V°G

OGGETTO: Sorgente in Processing che realizzi graficamente lo sviluppo in serie di Fourier.

## CENNI TEORICI

Lo sviluppo in serie di Fourier permette di esprimere un segnale periodico ( $T=1/f$ ) con la somma di un numero infinito di funzioni sinusoidali.

Si esprime con la seguente formula:

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} A_n \sin(2\pi nft) + \sum_{n=1}^{\infty} B_n \cos(2\pi nft)$$

Dove

$$A_n = \frac{2}{T} \int_0^T g(t) \sin(2\pi nft) dt$$

$$B_n = \frac{2}{T} \int_0^T g(t) \cos(2\pi nft) dt$$

$$C = \frac{2}{T} \int_0^T g(t) dt$$

Le funzioni sinusoidali con cui viene espresso il segnale prendono il nome di armoniche. Si considera "armonica fondamentale" quell'armonica che ha la stessa frequenza del segnale originale, mentre le successive armoniche hanno tutte frequenze multiple rispetto alla frequenza del segnale originale.

Quanto più velocemente varia nel tempo il segnale  $s(t)$ , tanto più numerose sono le armoniche necessarie a descriverlo.

Lo spettro di frequenze, invece, rappresenta l'ampiezza di tutte le armoniche, calcolato tramite il valore RMS. Le ampiezze delle armoniche sono utili perché ci danno un'idea sull'energia trasmessa alle varie frequenze.

$$RMS = \sqrt{A_n^2 + B_n^2}$$

Definiamo Banda di frequenza, quindi, l'insieme delle frequenze utili a definire il segnale.

Per rappresentare un esempio di sviluppo in serie di Fourier ci siamo serviti di Processing, un'ambiente di sviluppo basato su Java (<http://processing.org>).

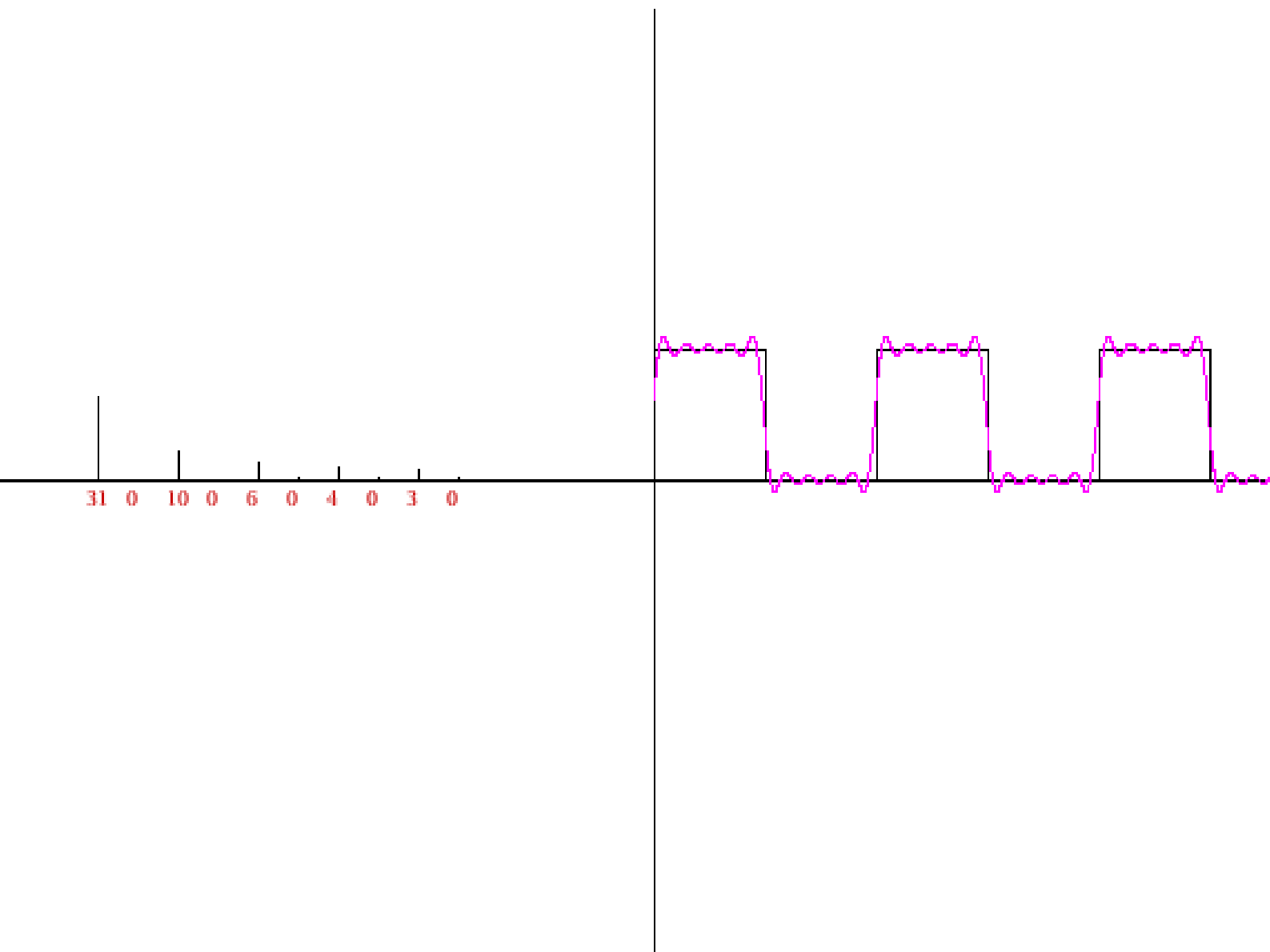
## PROCESSING

Processing è un linguaggio e un ambiente di sviluppo orientato all'Interaction Design. La sintassi è ripresa dal linguaggio Java, ma semplificata soprattutto per quel che riguarda la realizzazione di Applet.

Un tipico programma in Processing prevede un metodo “setup()” in cui viene definita la finestra del programma (dimensioni, sfondo ecc..) e un metodo “draw” che equivale al “while (true)” del java, in quanto viene eseguito in modo ciclico. Nel nostro programma abbiamo inserito l'istruzione “noLoop()” che fa eseguire il draw una volta soltanto.

#### IL PROGRAMMA – Sviluppo in serie di Fourier

Il programma, scritto con Processing, fornisce in output la rappresentazione grafica di un'onda quadra (segnale originale) e la rappresentazione del segnale ricostruito. Inoltre, sulla sinistra, viene visualizzato lo spettro di frequenza.



*Illustrazione 1: Output fornito dal programma. In nero il segnale originale, in magenta quello ricostruito.*

Ecco il programma con gli opportuni commenti:

```

float deltax, passo=0.1, x, y, y0, y1, a, t, f;
int n_armoniche;

void setup(){
  size (600, 400);
  background(255, 255, 255);
  t=100;
  a=50;
  n_armoniche=10;
  deltax=t/100;
  fill(200,0,0);
  line (0, height/2, width, height/2);
  line(width/2, 0, width/2, height);
  noLoop();
}

void draw(){
  g();
  fourier();
  spettri();
}

void g(){
  stroke(0, 0, 0);
  for (x=width/2; x<width; x=x+t){
    line(x, height/2, x, height/2-a);
    line(x, height/2-a, x+t/2, height/2-a);
    line(x+t/2, height/2-a, x+t/2, height/2);
    line(x+t/2, height/2, x+t, height/2);
  }
}

void fourier(){
  stroke(255, 0, 255);
  for (x=width/2; x<width; x=x+passo){
    y=c()/2+sinusoide(x, n_armoniche)+cosinusoide(x,
n_armoniche);
    y=height/2-y;
    point(x, y);
  }
}

float c(){
  float areatd=0;
  for (float i=width/2; i<width/2+t/2; i=i+deltax){

```

//Definizione delle variabili globali

//Metodo setup(). In questo metodo viene definita la dimensione della finestra (600\*400, in pixel), lo sfondo (bianco), il valore di alcune variabili e vengono disegnati gli assi cartesiani con l'istruzione "line(x0, y0, x1, y1)".

Tra le variabili la più importante è "n\_armoniche", che definisce il numero di armoniche che comporrà il segnale ricostruito. Un numero basso di armoniche renderà il segnale poco preciso, mentre un numero alto rende lento il calcolo ma precisa la ricostruzione. L'istruzione noLoop() impone che il successivo metodo "draw()" venga eseguito una volta sola.

//Questo metodo richiama altri tre metodi: g(), fourier() e spettri().

//Il metodo g() disegna un'onda quadra in nero a partire dall'origine. Il periodo dell'onda quadra è definito dalla variabile t globale, inizializzata a 100 in "setup()". Height e Width sono attributi dell'oggetto finestra, creato automaticamente con i valori impostati in size(600, 400) in setup().

il metodo stroke(0, 0, 0) imposta a nero il colore con il quale saranno disegnate le linee.

//Metodo che esegue lo sviluppo in serie di Fourier.

Il metodo stroke(255, 0, 255) imposta a magenta il colore con il quale saranno disegnate le armoniche con la funzione point(x, y).

Il ciclo for calcola per le ascisse ad un passo predefinito i valori dello sviluppo in serie di Fourier. Per il calcolo della costante 'c' e per la sommatoria di seni e coseni vengono richiamati tre metodi definiti in seguito.

// Calcolo della costante c, utile per traslare la funzione sull'asse y. Per il calcolo dell'integrale, in questa funzione e in quelle successive, viene utilizzato

```

    areatd=areatd+deltax*a;
}
println(areatd);
areatd=areatd*2/t;
return (areatd);
}

```

```

float an(int n){
    float areatd=0;
    for (float i=width/2; i<width/2+t/2; i=i+deltax)
        areatd=areatd+deltax*a*sin(2*PI*n*i/t);
    println(areatd);
    areatd=areatd*2/t;
    return (areatd);
}

```

```

float bn(int n){
    float areatd=0;
    for (float i= width/2; i<width/2 + t/2; i=i+deltax)
        areatd=areatd+deltax*a*cos(2*PI*n*i/t);
    println(areatd);
    areatd=areatd*2/t;
    return (areatd);
}

```

```

float sinusoide(float q, int n){
    float seno=0;
    for (int j=1; j<=n; j++)
        seno=seno+an(j)*sin(2*PI*j*q/t);
    return (seno);
}

```

```

float cosinusoide(float q, int n){
    float coseno=0;
    for (int j=1; j<=n; j++)
        coseno=coseno+bn(j)*cos(2*PI*j*q/t);
    return (coseno);
}

```

```

void spettri(){
    int incremento=18, x=50, y=height/2, i=1, valint;
    stroke (0, 0, 0);
    for (i=1; i<=n_armoniche; i++){
        line (x, y, x, y-sqrt(sq(an(i))+sq(bn(i))));
    }
}

```

il metodo dei rettangoli. Essendo la funzione da sviluppare un'onda quadra con un periodo equamente suddiviso fra valore logico alto (di ampiezza a) e valore logico basso (di ampiezza 0), il ciclo for provvede a calcolare l'area dei rettangoli moltiplicando la base (deltax) per l'altezza (a) solo sul semiperiodo nel quale il segnale assume valore logico alto.

//Calcolo di An con lo stesso metodo utilizzato per c(). In questo caso l'altezza del rettangolo è data dal prodotto di a (ampiezza del segnale) per il seno relativo all'armonica n passata come parametro.

//Calcolo di Bn.

//Questa funzione calcola per ogni ascissa (q) l'ordinata relativa alla sommatoria delle sinusoidi per tutte le armoniche (n).

//Questa funzione calcola per ogni ascissa (q) l'ordinata relativa alla sommatoria delle cosinusoidi per tutte le armoniche (n).

//Metodo che disegna lo spettro di frequenza.

```
valint=int(sqrt(sq(an(i))+sq(bn(i))));
    PFont font = loadFont("LucidaBright-Demi-
8.vlw");
textFont (font);
text(valint, x-5, y+10);
x=x+incremento;
}
}
```

#### *ALLEGATI*

*Il file compresso allegato (Fourier.zip) contiene:*

*Fourier.pde: Codice sorgente del programma. Per modificarlo e compilarlo è necessario il programma Processing (Open Source, liberamente scaricabile dal sito <http://processing.org>)*

*Cartella Data: contiene il font utilizzato per visualizzare i valori delle ampiezze delle armoniche.*

*Cartella Applet: contiene la pagina index.html e l'applet che viene utilizzata per visualizzare lo sviluppo in serie di Fourier.*

*Cartelle con l'applicazione per i sistemi operativi Linux, Mac e Windows.*

*Marco Lochi & Davide Sabatelli*