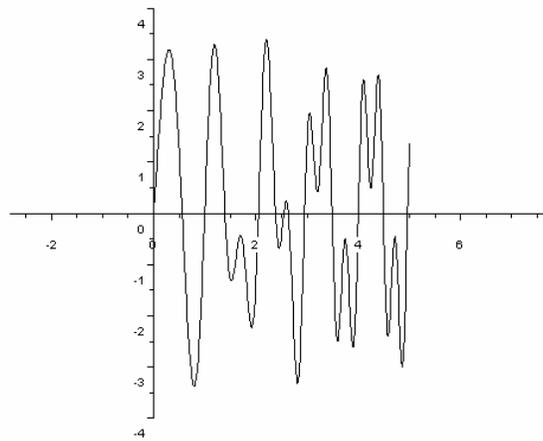


Simulazioni al Calcolatore

by
Federico Milan



Prefazione

Nella realtà di tutti i giorni siamo sempre più sommersi da minuscoli e a volte giganteschi computer che predicono e modellano la nostra realtà; si pensi ai centri per le previsioni meteorologiche, ai programmi di simulazione per circuiti integrati o perché no!, ai sistemi per elaborare previsioni per il lotto ...

Questo documento non spazierà così ampiamente il campo delle simulazioni, nemmeno tratterà in modo rigoroso l'argomento, non è un manuale e non può e non vuole sostituire scritti e trattati sicuramente più illustri e più adeguati ad una materia che merita il "massimo del rispetto.

Questo documento vuole piuttosto essere un punto di partenza, mostrando le potenzialità del calcolatore e come sfruttarle con un pizzico di matematica e di curiosità.

Il seguente documento è di proprietà esclusiva dell'autore il quale si riserva tutti i diritti.

L'utilizzo del seguente materiale o parte di esso è consentito se e solo se viene citata la fonte, l'autore, nonché l'autore stesso venga messo al corrente dell'uso che se ne intende fare (eventualmente con diritto di veto ©).

© Copyright – Federico Milan, 2006

Qualunque FeedBack è sempre apprezzato.

L'autore non è responsabile di false citazioni, errori, danni a cose e/o persone derivanti dalla lettura del seguente materiale.

Gli indirizzi di riferimento per contattare l'autore sono:

milan1@interfree.it
federico@plcforum.it

Federico Milan
Cittadella, PD
Italy

Indice

Capitolo I

1.1 Scalari

1.2 Vettori

1.2.1 Operazioni elementari tra vettori

1.2.2 Vettori al calcolatore

1.2.3 Vettori e numeri complessi al calcolatore

1.3 Trasformata di Fourier

1.3.1 La trasformata di Fourier e il calcolatore

Capitolo II

2.1 Spazio, tempo, velocità, accelerazione

2.2 Moti composti

2.3 Cenni su sistemi complessi

Capitolo III

3.1 Presentazione del sistema elettronico

3.2 Considerazione sulla simulazione

3.3 Simulazione al calcolatore del circuito

Capitolo

I

Strumenti algoritmico/matematico di base

Nello stile del documento, non vi è alcuna pretesa di perfezione e rigorosità nel presentare alcuni dei più importanti e funzionali strumenti che la matematica e l'informatica ci mettono a disposizione; anzi, la trattazione sarà la più basilare e semplice possibile, a maggior ragione non vi saranno noiose dimostrazioni, le quali potranno essere reperite in un qualunque testo dedicato.

*Questo capitolo vuole essere solo una piccola premessa alla semplice simulazione del **Capitolo III**.*

• 1.1 Scalari

Quando si parla di scalari ci si riferisce a "semplici" numeri; esempi di scalari sono:

- La temperatura globale di una stanza
- La lunghezza di un oggetto
- I numeri Naturali
- I numeri Reali

In poche parole possiamo pensare uno scalare come un elemento che determina una singola grandezza, ad esempio:

- Il peso di un oggetto
- La distanza tra terra e luna
- Il numero di pesci pescato

Dal punto di vista di un calcolatore, lo scalare, è un valore: il risultato di una semplice operazione, il numero contenuto in una variabile o il numero di iterazioni di un ciclo. Lo scalare è l'elemento più semplice e maneggevole che si possa utilizzare, non per questo meno importante.

• 1.2 Vettori

I vettori sono strumenti matematici molto importanti, hanno notevoli proprietà, tra le quali la capacità di descrivere in modo conciso aspetti multipli di una grandezza, a tal proposito si pensi ad esempio all'individuazione di un punto nello spazio tridimensionale, come è logico aspettarsi servono 3 lunghezze per definire la posizione. Tramite i vettori, con una unica scrittura matematica del tipo \vec{S} si riesce a rappresentare in modo conciso e univoco un punto nello spazio tridimensionale (figura 1). La rappresentatività di questo strumento non finisce ovviamente qui, ma va ben oltre gli scopi di questo capitolo. I vettori non sono strumenti monodimensionali, ma possono essere definiti a dimensioni generiche (Vettori bi, tri, quadri-dimensionali ... spesso queste strutture sono chiamate anche con il termine di matrice).

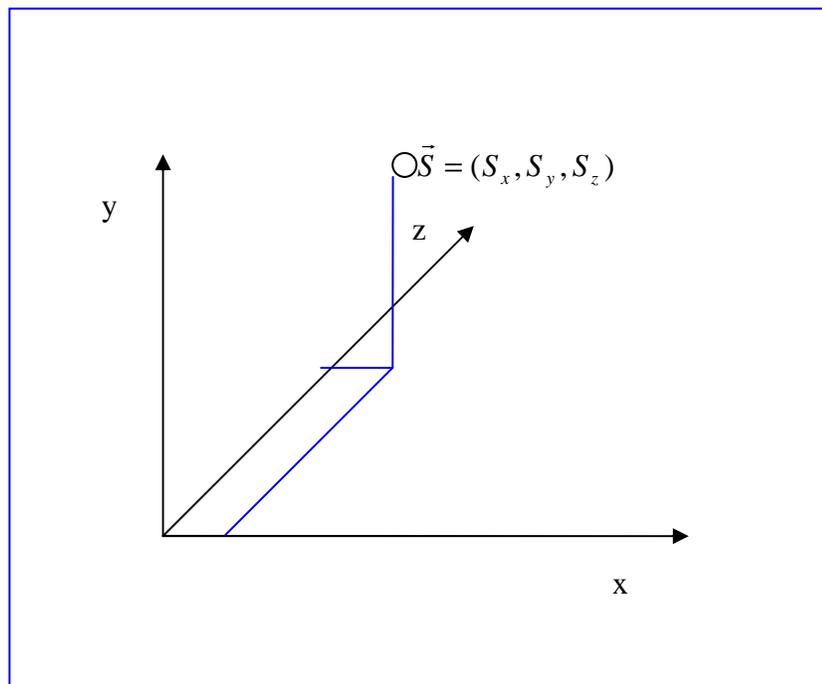


Figura 1 – Rappresentazione di un punto nello spazio tridimensionale

I vettori oltre che simbolicamente, possono essere rappresentati anche tramite forme tabellari (figura 2). Sotto certe ipotesi è possibile definire delle operazioni di somma e prodotto tra vettori e vettori e tra vettori e scalari

Dal punto di vista del calcolatore, un vettore non è altro che una serie contigua di celle di memoria (simile alla forma tabellare), dove possono essere registrati dati omogenei tra di loro.

$$\vec{V} = [a \quad b \quad c \quad \dots \quad z];$$

$$\vec{W} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ \dots \\ n \end{bmatrix};$$

$\vec{V} \bullet \vec{W} = 1 * a + 2 * b + \dots n * z$ che è uno scalare.

$$\vec{W} \bullet \vec{V} = \begin{bmatrix} \alpha_{(1 \ 1)} & \alpha_{(1 \ 2)} & \dots & \alpha_{(1 \ z)} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \alpha_{(n \ 1)} & \alpha_{(n \ 2)} & \dots & \alpha_{(n \ z)} \end{bmatrix}$$

$$\vec{\Phi} = \begin{bmatrix} 34.5 \\ 4.55 \\ 4.3 \\ 4.6 \end{bmatrix} \rightarrow \Phi_2 = 4.55$$

Figura 2 – Vettori rappresentati come tabelle

La capacità più grande di un calcolatore è di saper trattare e usare dati in forma di vettore in modo nativo e con grande velocità. Tramite i vettori si possono memorizzare serie numeriche, indicizzare dati quindi utilizzarli attraverso appositi algoritmi. Per il calcolatore, quindi, un vettore è un ottimo posto dove conservare i dati.

Vista l'importanza della struttura, conviene prendere un po' di confidenza con i vettori e alcune notazioni di utilizzo in pseudo linguaggio (il linguaggio utilizzato è molto simile a programmi di matematica di alto livello quali ©MatLab, ©SciLab, ...)¹.

```
t = [0:(% pi/10) : 2*% pi]; // notazione  $\pi = \%pi$ 
Vm=1.5;
v = Vm.* sin(t);           // .* moltiplicazione tra vettore e scalare

plot(t, v, xtitle('Andamento funzione v(t) = Vm * sin(t)'))
```

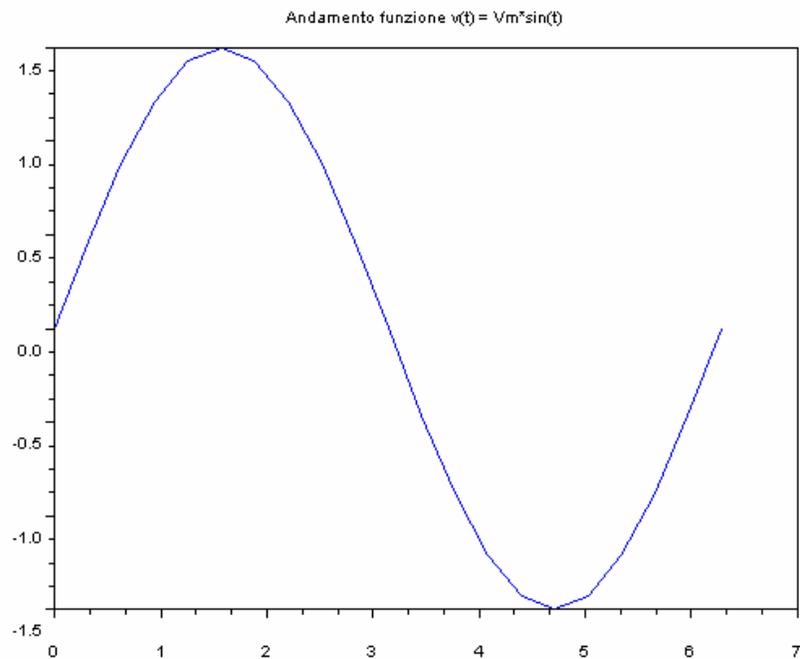


Figura 3 – grafico di una funzione e relativo pseudo codice

- **1.2.1 Operazioni elementari tra vettori**

Come accennato precedentemente, tra vettori si possono definire alcune operazioni. Le operazioni di maggiore interesse sono ovviamente la somma e il prodotto, purtroppo, a diversità degli scalari, la somma e il prodotto di

¹ MatLab e SciLab sono marchi registrati dai relativi proprietari.

vettori tra vettori o scalari è possibile solo sotto certe ipotesi relativamente. Per semplicità vedremo solo casi semplici e di interesse pratico:

1. La somma tra vettore e uno scalare non è definita.
2. La somma tra vettori è possibile se e solo se sono vettori congruenti (vettore riga con vettore riga, vettore colonna con vettore colonna) con dati omogenei e di eguali dimensioni, es:

$$\begin{aligned}\bar{s} &= [a_1 \quad a_2 \quad a_3]; \\ \bar{z} &= [b_1 \quad b_2 \quad b_3]; \\ \bar{s} + \bar{z} &= [a_1 + b_1 \quad a_2 + b_2 \quad a_3 + b_3];\end{aligned}$$

Le stesse regole si applicano per eventuali vettori colonna. Stesso discorso si applicano per vettori a dimensioni multiple, vedi le matrici:

$$\begin{aligned}\bar{s} &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}; \\ \bar{z} &= \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}; \\ \bar{s} + \bar{z} &= \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \\ a_{31} + b_{31} & a_{32} + b_{32} & a_{33} + b_{33} \end{bmatrix}\end{aligned}$$

3. La moltiplicazione tra vettore e uno scalare è sempre definita (per i nostri scopi). Questa operazione spesso è utilizzata per una scalatura del segnale:

$$\begin{aligned}\bar{s} &= [a_1 \quad a_2 \quad a_3]; \\ z &= \alpha; \\ z \cdot \bar{s} &= \alpha \cdot [a_1 \quad a_2 \quad a_3] = [\alpha \cdot a_1 \quad \alpha \cdot a_2 \quad \alpha \cdot a_3];\end{aligned}$$

Stesse regole si applicano per il prodotto tra scalare e matrice.

4. Il prodotto tra vettori è molto particolare e non ha una forma “naturale”, di seguito alcuni esempi per rendere l’idea di cosa sia un prodotto “vettoriale” ed alcune sue proprietà:

$$\vec{s} = [a_1 \quad a_2 \quad a_3];$$

$$\vec{z} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix};$$

$\vec{s} \cdot \vec{z} = a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3$; che è uno scalare e non più un vettore

$$\vec{z} \cdot \vec{s} = \begin{bmatrix} a_1 \cdot b_1 & a_2 \cdot b_1 & a_3 \cdot b_1 \\ a_1 \cdot b_2 & a_2 \cdot b_2 & a_3 \cdot b_2 \\ a_1 \cdot b_3 & a_2 \cdot b_3 & a_3 \cdot b_3 \end{bmatrix}$$

- **1.2.2 Vettori al calcolatore**

Oltre alle operazioni definite matematicamente, al calcolatore è possibile definire altre operazioni, spesso le più importanti sono la moltiplicazione elemento ad elemento tra due vettori lineari:

$$\vec{s} = [a_1 \quad a_2 \quad a_3];$$

$$\vec{z} = [b_1 \quad b_2 \quad b_3];$$

$$\vec{s} \cdot * \vec{z} = [a_1 \cdot b_1 \quad a_2 \cdot b_2 \quad a_3 \cdot b_3];$$

- **1.2.3 Vettori e numeri complessi al calcolatore**

Anche i numeri complessi possono essere visti come vettori. Su di essi sono definite tutte le operazioni dei vettori e inoltre “possono essere trattati” come scalari (anche se questa affermazione non è proprio corretta):

$$\vec{c} = a + j \cdot b = (a, b) = [a \quad b];$$

$$\vec{s} = [s_1 \quad s_2 \quad s_3];$$

$$\vec{c} \cdot \vec{s} = [a \quad b] \cdot [s_1 \quad s_2 \quad s_3] = [s_1 \cdot [a \quad b] \quad s_2 \cdot [a \quad b] \quad s_3 \cdot [a \quad b]]$$

Al calcolatore quindi è possibile implementare queste strutture dati in modo molto semplice e “nativo”. La potenzialità di questi strumenti consente di ottenere trattazioni semplici e concise di aspetti della realtà altrimenti difficilmente rappresentabili.

La trattazione di questo argomento non è esaustiva ma si è voluto soffermarci solamente sulle possibili applicazioni al calcolatore di questi strumenti matematici.

• 1.3 Trasformata di Fourier

In questo paragrafo si fa un salto ulteriore sugli strumenti matematici che abbiamo a disposizione, per la tipologia del documento viene volutamente saltato argomenti molto importanti e molto interessanti quali l'analisi complessa e l'utilizzo e risoluzione delle equazioni differenziali, strumenti importanti, ma per i nostri scopi non così semplici e veloci da utilizzare. Le trasformate, in questo caso la "Trasformata di Fourier" è uno strumento molto potente ma allo stesso modo semplice da utilizzare con il PC, purtroppo è necessario una piccola trattazione teorica per sfruttarla.

Tralasciando i dettagli, e sempre nello spirito della semplicità, Fourier si "accorse" che alcune classi particolari di funzioni, potevano essere rappresentate come:

$$s(t) = \alpha_0 + \alpha_1 \sin(\varpi \cdot t) + \alpha_2 \sin(2 \cdot \varpi \cdot t) + \dots + \alpha_n \sin(n \cdot \varpi \cdot t) + \beta_1 \cos(\varpi \cdot t) + \beta_2 \cos(2 \cdot \varpi \cdot t) + \dots + \beta_n \cos(n \cdot \varpi \cdot t)$$

$$\alpha_0 = \frac{1}{T} \cdot \int_0^T s(t) \cdot \delta t$$

$$\alpha_n = \frac{1}{T} \cdot \int_0^T s(t) \cdot \sin(n \cdot \varpi \cdot t) \cdot \delta t$$

$$\beta_n = \frac{1}{T} \cdot \int_0^T s(t) \cdot \cos(n \cdot \varpi \cdot t) \cdot \delta t$$

La scoperta fatta da Fourier è di estrema importanza, in quanto, il significato della sua serie sta a mostrare come alcune classi di funzioni possono essere scomposte in serie di funzioni sinusoidali. Questo significa che un segnale complesso, purchè rispetti alcune ipotesi, può essere studiato analizzando semplicemente le sue singole componenti della serie di Fourier (un caso di ***divide et impera***), quindi semplificandone l'analisi. Analizzando la serie di Fourier, si nota come possa essere (limitando la serie ad un numero N) completamente contenuta su di un vettore di dimensione N. Questo significa che, conosciuti i coefficienti alfa e beta, un segnale può essere memorizzato e analizzato al calcolatore semplicemente analizzando gli elementi del vettore stesso.

Pur essendo la serie di Fourier molto interessante, per i casi pratici conviene utilizzare la trasformata di Fourier, che altro non è che una generalizzazione della serie stessa. Osservando la serie infatti possiamo fare le seguenti considerazioni:

$$s(t) = \frac{\alpha_0}{2} + \sum_{i=1}^{N \rightarrow \infty} (\alpha_i \sin(i \cdot \varpi \cdot t) + \beta_i \cos(i \cdot \varpi \cdot t))$$

$$\alpha_n = \frac{2}{T} \cdot \int_{\frac{T}{2}}^{\frac{T}{2}} s(t) \cdot \sin(n \cdot \varpi \cdot t) \cdot \delta t$$

$$\beta_n = \frac{2}{T} \cdot \int_{\frac{T}{2}}^{\frac{T}{2}} s(t) \cdot \cos(n \cdot \varpi \cdot t) \cdot \delta t$$

Che consentono di compattare ulteriormente la scrittura. Per compattare ulteriormente e arrivare alla definizione di trasformata si procede come di seguito:

$$j = \sqrt{-1}$$

$$\cos(n \cdot \omega \cdot t) + j \cdot \sin(n \cdot \omega \cdot t) = e^{j \cdot n \cdot \omega \cdot t}$$

$$c_n = \frac{2}{T} \cdot \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t) \cdot e^{-j \cdot n \cdot \omega \cdot t} \cdot \delta t$$

$$s(t) = \sum_{i=1}^{N \rightarrow \infty} c_i \cdot e^{j \cdot n \cdot \omega \cdot t}$$

La trasformata di Fourier viene così definita :

$$S(\omega) = \int_{-\infty}^{\infty} s(t) \cdot e^{j \cdot \omega \cdot t} \cdot \delta t$$

La trasformata, consente quindi di interpretare un segnale secondo le sue componenti in frequenza. L'interesse pratico sta infatti nella semplificazione nello studio dei sistemi complessi e una semplificazione nell'implementare questi strumenti sul calcolatore.

La trasformata di Fourier, consente inoltre di rappresentare alcuni segnali particolari solamente con un minimo contenuto di informazioni. Si pensi per esempio alla rappresentazione di una sinusoide di frequenza e ampiezza nota. La sua rappresentazione quindi è possibile solamente memorizzando l'ampiezza e la pulsazione (o la frequenza) della sinusoide in una struttura come la seguente $\hat{A} = A \cdot \sin(\omega \cdot t) = [A \quad \omega];$.

• 1.3.1 La trasformata di Fourier e il calcolatore

Come stile del documento non si entrerà nel dettaglio, ma si cercherà di dare una trattazione molto pratica e introduttiva all'utilizzo della trasformata di Fourier attraverso il calcolatore.

Come visto prima la trasformata è un potente strumento matematico per l'analisi di alcune classi di segnali. Al calcolatore è possibile implementarla con alcune accortezze; è necessario, però, saperne interpretare il significato. La trasformata di Fourier è una trasformata "continua", mentre il calcolatore, per sua natura, tratta solo grandezze discrete. Per questo motivo sono date una serie di classi di trasformate, necessarie per lavorare con il calcolatore. Tra le più importanti troviamo le DFT (Discrete Fourier Transform) da cui derivano le FFT (Fast Fourier Transform) le quali di regola lavorano solo su serie di campioni con dimensione multipla di 2.

Di seguito (figura 4) un esempio semplice di come utilizzare la trasformata di Fourier discreta al calcolatore:

```
n = [0:255]; // 255 campioni
Vm = 1.5;
v = Vm .* cos(2 * %pi * n / 255);
V = fft(v);
f = [-255/2 : 255/2] / 255;
w = 2 * %pi .* f;
```

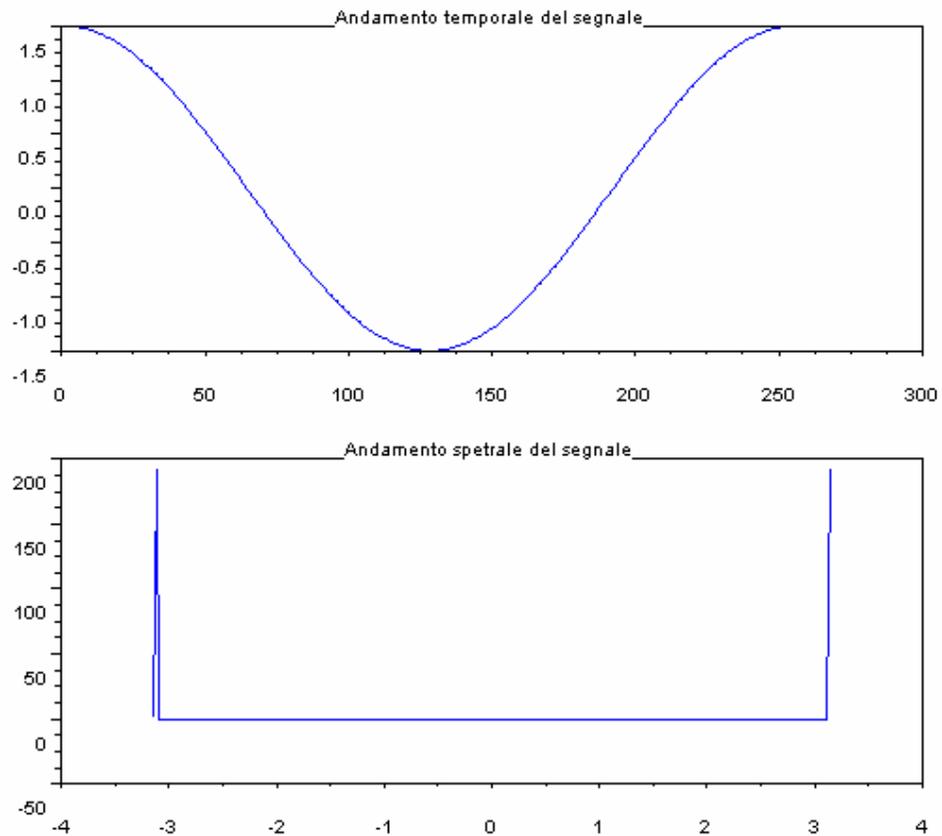


Figura 4 – Dimostrazione della FFT.

Capitolo

II

Semplici simulazioni

In questo capitolo si mostreranno alcune semplici simulazioni e alcuni metodi di analisi, utili per l'interpretazione dei dati quando si utilizza il calcolatore. Bisogna, infatti, ricordare che il calcolatore è un potente mezzo di elaborazione di serie numeriche, ma non è un conoscitore della realtà. Questo significa che un dato errato non può essere scoperto, e solo la conoscenza del processo può verificare l'attendibilità dei dati. Del resto, parlando di simulazioni, si deve parlare di attendibilità dei dati più che correttezza, in quanto il valore assoluto non ha significato, ma quello che si cerca in una simulazione è l'andamento più che semplici numeri.

• 2.1 Spazio, tempo, velocità, accelerazione

Nella fisica esistono molti aspetti da simulare, in particolare la dinamica, presenta casi di studio di interesse pratico, visibili direttamente nella realtà, e cosa molto interessante molto intuitivi.

La simulazione più semplice che si può realizzare è graficare l'andamento di un punto materiale secondo le leggi della dinamica classica. A tale proposito, per entrare nel vivo della simulazione, si vuole provare a simulare l'andamento di un punto (privo di massa) soggetto ad una accelerazione. Lo scopo sarà quindi di graficare l'andamento della velocità e dello spazio percorso dal punto in funzione del tempo (che riteniamo la nostra variabile “indipendente”).

Prima di simulare si deve porre delle ipotesi e dei “punti fissi”:

1. ipotizzeremo che il nostro punto sia soggetto ad una accelerazione

$$\text{iniziale di } a = 0.5 \frac{m}{s^2};$$

2. ipotizziamo una velocità iniziale nulla $v_0 = 0 \frac{m}{s}$;

3. ipotizziamo di partire dal punto $S_0 = 0$.

Per poter simulare il sistema descritto occorre analizzare prima matematicamente come si comporta il sistema:

$$a = 0.5 \cdot \left[\frac{m}{s^2} \right]$$

$$v_0 = 0 \cdot \left[\frac{m}{s} \right]$$

$$s_0 = 0 \cdot [m]$$

$$v = v_0 + \int_0^t a \cdot \delta\tau = v_0 + a \cdot t = a \cdot t = 0.5 \cdot \left[\frac{m}{s^2} \right] \cdot t \cdot [s]$$

$$s = s_0 + \int_0^t v \cdot \delta\tau = \int_0^t 0.5 \cdot \tau \cdot \delta\tau = 0.5 \cdot \int_0^t \tau \cdot \delta\tau = 0.5 \cdot \left[\frac{m}{s^2} \right] \cdot \frac{1}{2} \cdot t^2 \cdot [s]^2$$

Che come si vede si ha a disposizione tutto quello che serve per graficare su carta, senza l'ausilio di altri strumenti l'andamento delle curve cercate. Nell'esempio, si è utilizzata una teoria semplice, dove l'andamento è esprimibile tramite formule, ma per generalizzare nella simulazione al calcolatore non useremo le formule dirette, ma proveremo a simulare gli andamenti con un metodo generico, e con un grado di precisione decidibile a priori.

$$N = 256;$$

$$\Delta t = 0.01 \cdot [s];$$

$$a = 0.5 \cdot \left[\frac{m}{s^2} \right];$$

$$v_0 = 0;$$

$$s_0 = 0;$$

$$v_n = v_{n-1} + a \cdot \Delta t;$$

$$s_n = s_{n-1} + v_n \cdot \Delta t;$$

Con la teoria esposta, in figura 4, si presenta uno dei possibili pseudo codici per eseguire la simulazione.

```
n = [0 : 255];  
v = n; s = n; t = 0.01.* n;  
a = 0.5; dt = 0.01;  
for i = 2 : 256,  
    v(i) = v(i - 1) + a * dt;  
end;  
for i = 2 : 256,  
    s(i) = s(i - 1) + v(i) * dt;  
end;
```

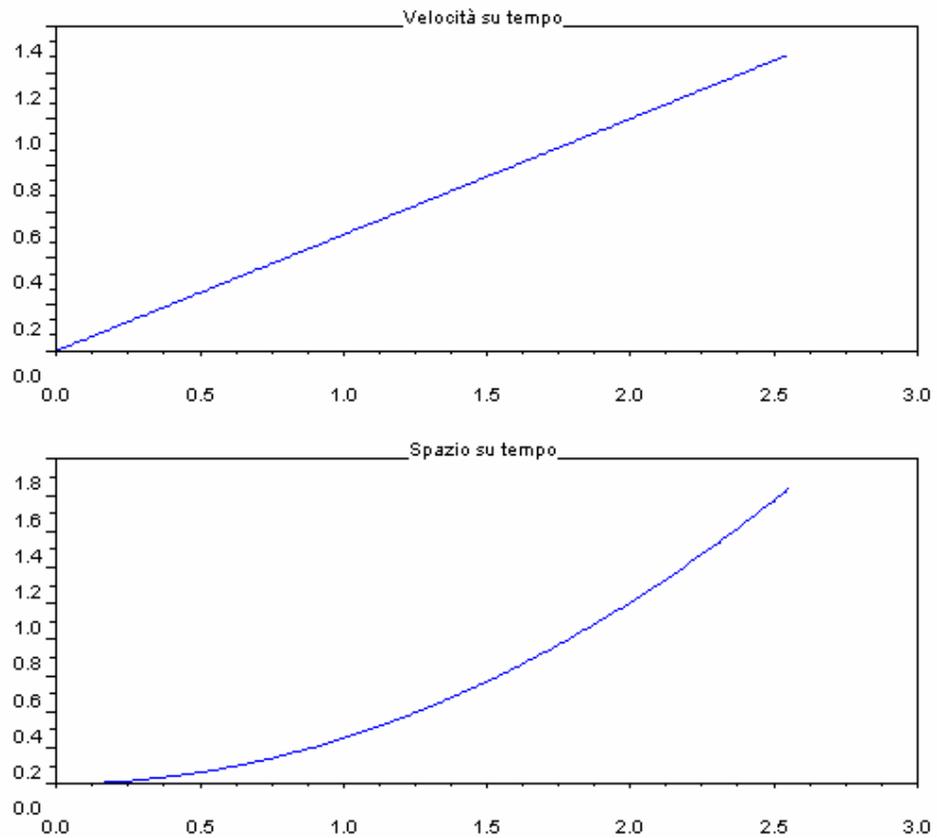


Figura 5 – simulazione al calcolatore dell'andamento spazio temporale di un punto soggetto ad una accelerazione

Si può anche verificare la precisione della simulazione, infatti l'ultimo elemento del vettore s contiene lo spazio totale percorso dal punto,

confrontandolo con il valore calcolato dalla formulazione precedente è possibile verificare l'errore (l'incertezza commessa):

$$s(256) = 1.632;$$

$$s = 0.5 \cdot \int_0^{0.01 \cdot 255} \tau \cdot \delta\tau = 0.5 \cdot \left[\frac{m}{s^2} \right] \cdot \frac{1}{2} \cdot (0.01 \cdot 255)^2 \cdot [s]^2 = 1.625625;$$

• 2.2 Moti composti

Spesso si devono simulare oggetti che si muovono su di uno spazio bidimensionale e o tridimensionale; nulla toglie la generalità del metodo esposto sul paragrafo 2.1. La tecnica è la medesima, solo che si dovranno eseguire tante simulazioni quante sono le dimensioni (esistono però una serie di casistiche in cui le grandezze sulle varie dimensioni sono legate tra di loro, ma per la trattazione di questo documento, e per semplicità espositiva, vengono volutamente non considerate).

Ipotizziamo di avere un punto che si muove l'ungo l'asse x con una accelerazione pari a $\sin(2 \cdot \pi \cdot t)$ e sull'asse y con una accelerazione pari a e^t .

Come per il paragrafo precedente, si creeranno i vettori velocità e poi i vettori spazio. Come unica complessità, so dovrà preparare anche un vettore accelerazione, in quanto l'accelerazione non è costante.

Da notare che la simulazione dà un andamento coerente con il modello fisico, ne approssima con buona precisione l'andamento (e i valori effettivi di spazio). Per aumentare la precisione del sistema basta diminuire il valore della variabile dt, che rappresenta il quanto temporale discreto utilizzato per realizzare la simulazione stessa.

In figura 6 la simulazione completa di grafico.

```

n = [0:255];
vx = vy = sx = sy = n; t = 0.01.*n;
ax = sin(2.*pi.*t); ay = e^t;
for i = 2:256,
    vx(i) = vx(i-1) + ax(i)*dt;
    vy(i) = vy(i-1) + ay(i)*dt;
end;
for i = 2:256,
    sx(i) = sx(i-1) + vx(i)*dt;
    sy(i) = sy(i-1) + vy(i)*dt;
end;

```

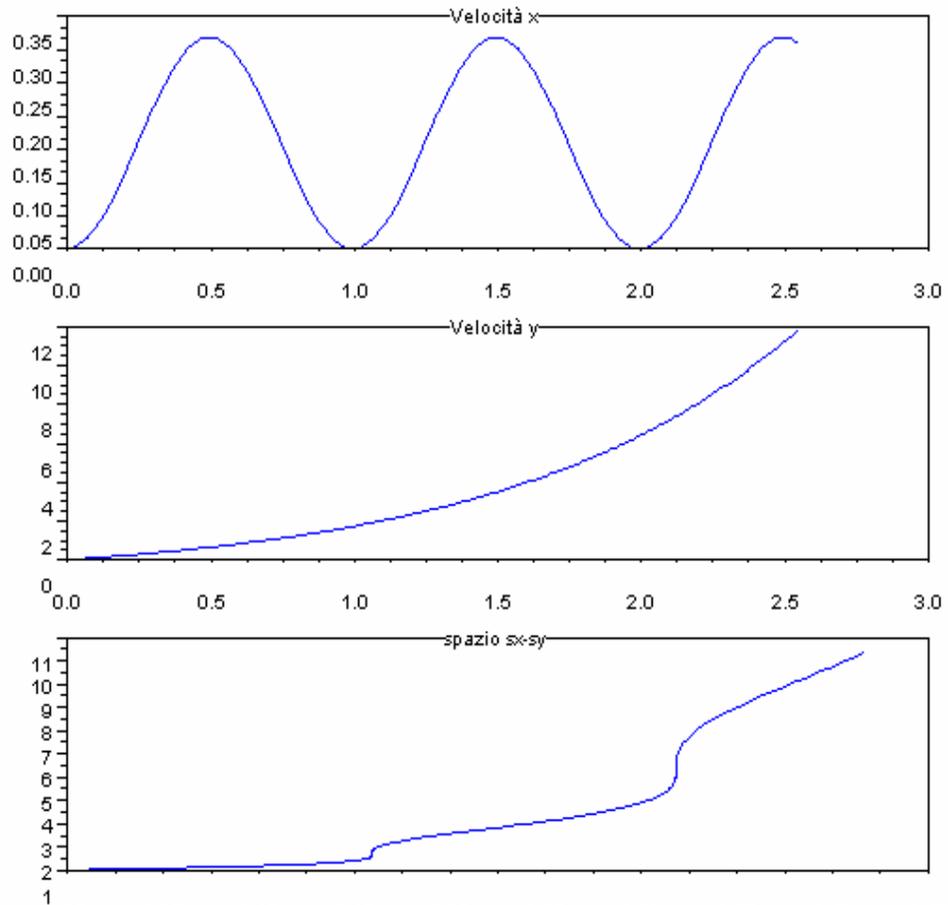
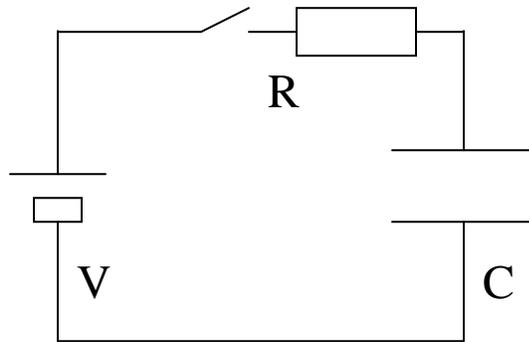


Figura 6 – simulazione al computer dell'andamento di un punto soggetto ad accelerazioni sull'asse x e y

• **2.3 Cenni su sistemi complessi**

Non tutti i sistemi possono essere simulati in modo semplice. Si pensi a sistemi elettrici composti da condensatori e induttanze. La simulazione di questi sistemi necessitano strumenti matematici un po' più efficaci. Per esempio consideriamo il semplice sistema elettrico definito in figura 7.



$$V - R \cdot C \cdot \frac{d}{dt} v_c - v_c = 0 \rightarrow \frac{d}{dt} v_c = -\frac{1}{R \cdot C} \cdot v_c + \frac{1}{R \cdot C} \cdot V$$

Figura 7 – sistema elettrico di una certa complessità

La risoluzione del sistema elettrico di figura 7, prevede la soluzione di una equazione differenziale. In questo caso l'equazione differenziale cade in una classe di equazioni differenziali conosciute e in cui esiste la soluzione, per completezza si ha quindi che:

$$\begin{cases} \frac{d}{dt} v_c = -\frac{1}{R \cdot C} \cdot v_c + \frac{1}{R \cdot C} \cdot V \\ v_c(0) = 0 \end{cases}$$

$$v_c = e^{-\frac{1}{R \cdot C} t} \cdot v_{c0} + \int_0^t e^{-\frac{1}{R \cdot C}(t-\tau)} \cdot \frac{1}{R \cdot C} \cdot V \cdot \delta\tau$$

$$v_c = \frac{1}{R \cdot C} \cdot V \cdot \int_0^t e^{-\frac{1}{R \cdot C}(t-\tau)} \cdot \delta\tau = V \cdot (1 - e^{-\frac{1}{R \cdot C} t})$$

Per la simulazione di questi sistemi sono stati definiti degli algoritmi particolari, in grado di risolvere le equazioni differenziali. Questi algoritmi vengono raggruppati in classi chiamate ODE (Ordinary Differentiated Equation).

Tramite programmi di matematica, spesso chiamati CAS (Computer Algebra System) è possibile risolvere queste classi di problemi.

In figura 8 un esempio di come procedere per risolvere questa tipologia di problemi.

```
R = 100;  
C = 0.0047;  
V = 5;  
function dvc = f(t, vc)  
    dvc = (1/R*C)*V - (1/R*C)*vc;  
endfunction  
y0 = 0; t0 = 0; t = 0 : 0.01 : 5;  
vc = ode(y0, t0, t, f);
```

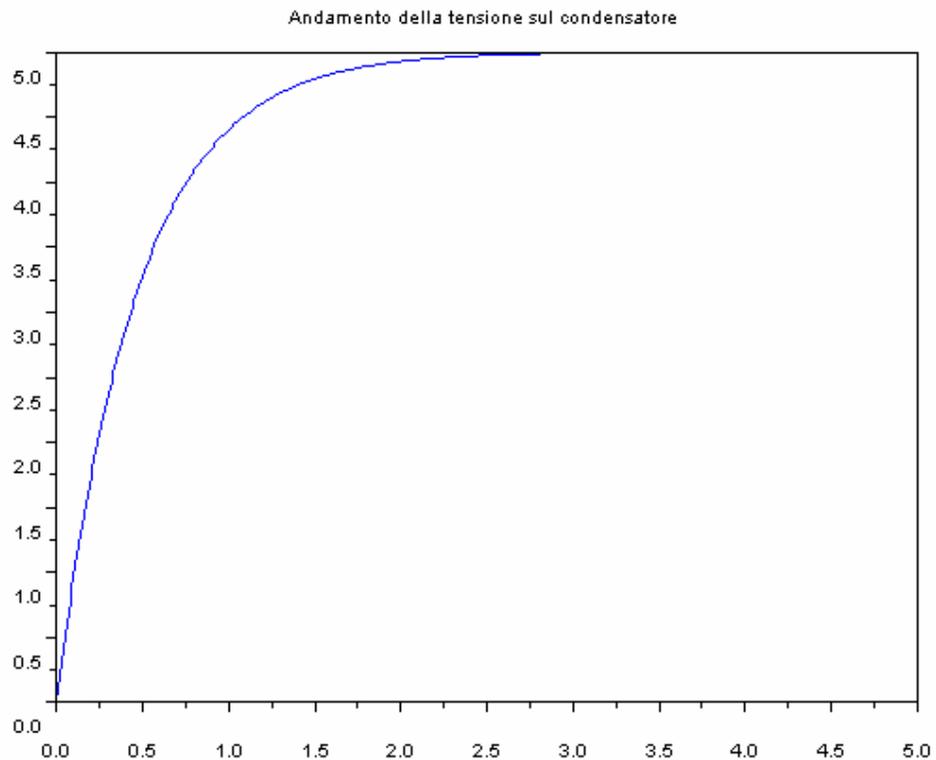


Figura 8 – sistema elettrico di una certa complessità

Come si vede la complessità delle simulazioni per sistemi descritti da equazioni differenziale aumenta esponenzialmente. Purtroppo, nell'automazione, questi casi non sono rari, anzi sono la routine!

Tra le classi di sistemi che possono essere studiati mediante equazioni differenziali, esistono delle sottoclassi di sistemi (molto interessanti per l'ingegneria) che si prestano ad essere studiati nel dominio della frequenza, piuttosto che nel dominio del tempo. A tale proposito esistono strumenti matematici molto efficaci, tra i quali oltre alla Trasformata di Fourier troviamo la Trasformata di Laplace (caso più generico di Fourier), la trasformata Cisoidale e molti altri strumenti derivanti dalla Teoria dei Sistemi.

Capitolo

III

Simulazione di un sistema elettronico

In questo ultimo capitolo si mostrerà come si potrebbe simulare un sistema elettronico relativamente complesso. Il sistema verrà scomposto nella funzione di trasferimento, quindi ipotizzato un ingresso ad onda quadra viene realizzata una analisi attraverso l'utilizzo della FFT e la sua inversa.

- **3.1 Presentazione del sistema elettronico**

Sebbene le funzioni di trasferimento (FdT) siano fondamentali per l'analisi e simulazione di sistemi, non verranno trattate. In questo paragrafo e per l'intero capitolo si presuppone che il lettore abbia almeno una vaga nozione di cosa sia una FdT e di come essa va ricavata dal sistema in esame.

In questo paragrafo verrà visto un sistema elettronico composta da amplificatore operazionale realizzante un filtro passa-banda. Non ci soffermeremo sulle tecniche di analisi di tale struttura, ne analizzeremo le proprietà di un tale sistema. Andremo ad estrapolare la funzione di trasferimento del sistema elettronico considerato, tralasciando le eventuali tecniche di analisi.

In figura 9 viene presentato il circuito elettronico che realizza la seguente funzione:

$$W(s) = -\frac{R_2}{R_1} \cdot \frac{1}{1 + s \cdot R_2 \cdot C_2} \cdot \frac{s \cdot R_1 \cdot C_1}{1 + s \cdot R_1 \cdot C_1} \Bigg|_{s=j\omega}$$

L'espressione della FdT mostra come si tratti di un filtro passa banda. Questo significa che la FdT lascerà passare "intatte" solo alcune frequenze, mentre altre verranno attenuate.

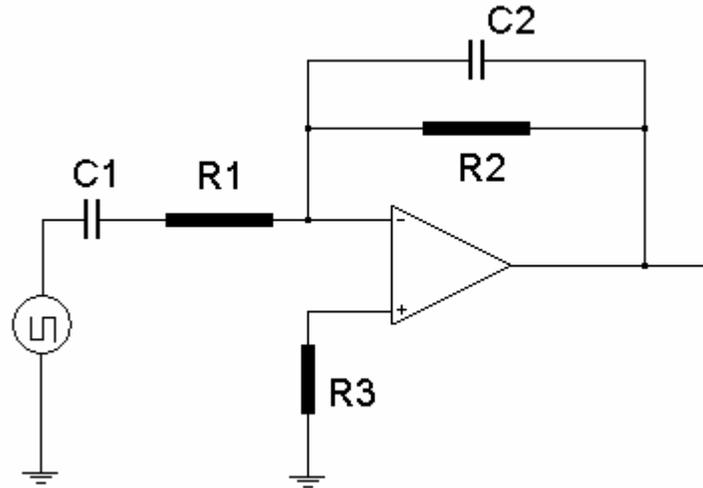


Figura 9 – Sistema elettronico con OpAmp

La figura 9 oltre a presentare il circuito, presenta anche un generatore di onde quadre. Lo studio-simulazione del circuito, infatti, verrà svolta utilizzando un generatore di onde quadre.

• 3.2 Considerazione sulla simulazione

Prima di procedere con la simulazione vera e propria, è opportuno focalizzare quale siano i punti salienti per effettuare una tale simulazione. Per preparare la simulazione del circuito, si effettueranno i seguenti punti:

1. Memorizzazione dei valori noti (resistenze, capacità)
2. Creazione di un vettore Onda quadra, simulante il generatore di onde quadra rappresentato in figura 9.
3. Creazione del vettore Trasformata di Fourier (discreta) del vettore onda quadra.
4. Creazione della FdT.

5. Calcolo della risposta del filtro (tramite FdT) sollecitato dal segnale ad onda quadra.
6. Trasformazione della risposta del sistema dal dominio della frequenza al dominio del tempo (antitrasformata di Fourier).

L'utilizzo della FFT sebbene sia "automatico", presenta alcune sorprese. Infatti, per rendere l'algoritmo efficiente, la FFT di un segnale genera un vettore contenente i coefficienti della trasformata, ma ribaltati rispetto alla posizione centrale. Questo è il motivo della strana forma del vettore pulsazioni che verrà proposto nel paragrafo successivo.

• 3.3 Simulazione al calcolatore del circuito

Finalmente siamo arrivati alla simulazione del sistema, sfruttando le potenzialità del calcolatore e dei sistemi CAS. Di seguito si presenterà il pseudo codice per realizzare la simulazione, i commenti del codice saranno esplicativi sulla funzionalità dello stesso.

Il codice proposto non è ottimizzato e non sfrutta appieno le potenzialità dei moderni CAS, a è strutturato per essere il più chiaro possibile. Inizieremo con la costruzione delle funzioni basilari, quelle necessarie per eseguire la simulazione e dove possibile sono rese parametriche per realizzare uno studio più completo e semplice, cioè senza riscrivere codice.

```
function [y] = Onda_Quadra(x)
y = x;           // x è un vettore contenente i campioni, è un vettore di appoggio
dim = length(y); // ricavo il numero di campioni previsto
for i=1:dim,     // ciclo per la generazione dell'onda quadra
    if (i < (dim/2)) then
        y(i) = 1;
    else
        y(i) = -1;
    end
end
endfunction
```

Figura 10 – codice generazione onda quadra dato un vettore di campioni

```
function [y] = Pulsazione(N,T)
y = [1:1:N];
for i=0:N-1,
    if (i < (N/2)) then
        y(i+1) = i*2*pi/T;
    else
        y(i+1) = (i-N)*2*pi/T;
    end
end
endfunction
```

Figura 12 – codice generazione vettore pulsazioni

```
function [y] = Fdt(R1, R2, C1, C2, w)
y = w;
j=sqrt(-1);

for i=1:length(w),
    y(i) = -R2/R1*(1/(1+j*w(i)*R2*C2))*(j*w(i)*R1*C1/(1+j*w(i)*R1*C1));
end
endfunction
```

Figura 13 – codice generazione FdT

Come già accennato queste tre funzioni sono il cuore della simulazione, infatti si è realizzato il vettore pulsazione necessario per creare assieme alla FdT il vettore contenente i coefficienti della risposta del filtro. Il vettore onda quadra, necessario per simulare il generatore di onde quadre.

Di seguito il rimanente codice per realizzare la simulazione.

```

//-----
// Valori resistenze
//-----
R1 = 3300;
R2 = 47000;

//-----
// Valori capacità
//-----
C1 = 100.000D-9;
C2 = 470.000D-12;

//-----
// Calcolo costanti di tempo (s)
//-----
T1 = R1*C1;
T2 = R2*C2;

//-----
// Calcolo costanti di appoggio
//-----
T = max(T1,T2); // Massimo valore ta le due costati temporali, serve per il periodo dell'onda quadra
N = 256; // 256 campioni

```

Figura 14 – codice generazione dati di contorno per la simulazione

```

function [t, n, q, w, f, W, Wf, out] = Simula(T, N, R1, R2, C1, C2)

_n = [0:1:N-1];
_t = _n;
_t = (T/(N-1)).*_n;

_q = Onda_Quadra(_n); // Onda quadra
_w = Pulsazione(N,T); // vettore pulsazione
_f = fft(_q,-1); // trasformata di Fourier dll'onda quadra
_W = Fdt(R1, R2, C1, C2, _w); // Calcolo della Fdt
_Wf = _W.*_f; // FdT per la funzione onda quadra di ingresso
_out = ifft(_Wf); // antitrasformata

t = _t; // riporto in uscita alla funzione I valori trovati
n = _n;
q = _q;
w = _w;
f = _f;
W = _W;
Wf = _Wf;
out = _out;
endfunction

```

Figura 15 – codice finale

In figura 15 viene proposto il codice finale, che altro non fa che mettere assieme i blocchi principali di simulazione. La funzione è parametrizzabile, questo consente di scrivere una volta solo il codice e modificando i parametri simulare una classe di “infinite” varianti della FdT e del sistema stesso.

Di seguito i risultati della simulazione in forma grafica.

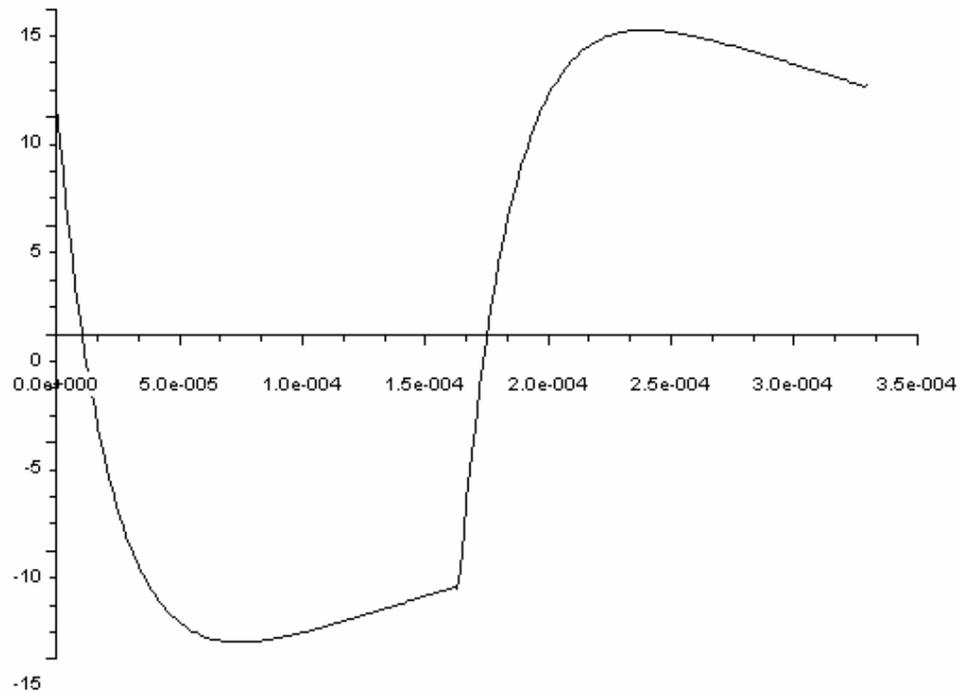


Figura 16 – Grafico del segnale in uscita al filtro sollecitato da un'onda quadra.

Il risultato finale mostra l'andamento del segnale in uscita al circuito.

• 3.4 Conclusioni

La simulazione è un aspetto molto interessante nella vita quotidiana, soprattutto per chi si occupa di aspetti tecnici. In questo documento abbiamo visto la simulazione di alcuni sistemi dinamici, ma l'argomento è talmente ampio da meritare molto più spazio.

L'aspetto più importante per chi si voglia occupare in futuro di simulazioni è la reperibilità di un modello il più simile possibile alla realtà. Da questo modello,

poi dovranno estrapolare i parametri di interesse, separare i parametri maggiormente influenzati da errori da quelli che incidono poco. In prima analisi semplificare il tutto per velocizzare l'analisi, quindi introdurre man mano parametri sempre più dettagliati.

È importante, per qualunque modello, renderlo generico e parametrizzabile, in modo da crearsi una libreria di sistemi componibili e semplicemente adattabili.