Esercizi facoltativi su FFT

francesco.fuso@unipi.it; http://www.df.unipi.it/∼fuso/dida

(Dated: version 1 - FF, 12 aprile 2017)

Questa nota ha lo scopo di introdurre l'argomento dell'*analisi di Fourier* condotta numericamente (*Discrete Fourier Transform* - DFT) attraverso il metodo della *Fast Fourier Transform (FFT)*. Dal punto di vista concettuale, si tratta di un argomento potentissimo e di importanza ubiqua in tantissimi settori della ricerca scientifica. Il nostro approccio al problema è minimale, e ci accontentiamo di impiegare una versione dell'algoritmo FFT implementato in Python per trattare dati acquisiti con Arduino in alcune esperienze di laboratorio. Questa operazione si presta ad essere argomento di un esercizio che, per quest'anno, si mantiene completamente *facoltativo*. La nota, dopo una breve introduzione, fornisce dei suggerimenti pratici per lo svolgimento dell'esercizio.

I. ANALISI DI FOURIER

In termini molto generali e volutamente privi dei contenuti concettuali e matematici che avete trovato, o troverete, in altri corsi, l'analisi di Fourier può essere considerata un meraviglioso strumento per esaminare sistemi fisici, in particolare quelli che si comportano in modo "lineare", in cui, per esempio, i segnali in "uscita" e "ingresso" sono legati tra loro da una relazione causale che può essere modellata da una funzione (semplice e possibilmente analitica) di trasferimento.

Questa funzione, generalmente complessa, descrive il comportamento del sistema in un dato dominio, che è molto spesso quello delle frequenze, o frequenze angolari, come abbiamo più volte avuto modo di incontrare nel nostro corso. La *trasformata* di Fourier, che è alla base del metodo di analisi che vogliamo trattare, consente di passare al dominio "coniugato", che è quello dei tempi. Naturalmente esistono altri domini coniugati tra loro, cioè tali che il prodotto tra le variabili (coordinate) dei due domini sia adimensionale; tuttavia in questa nota, in accordo con le nostre esigenze e possibilità, ci restringiamo a esaminare la trasformazione dal dominio dei tempi a quello delle frequenze. L'operazione inversa, che può anche essere eseguita, è chiamata *anti-trasformata*: nei nostri esempi, essa permette di passare dal dominio delle frequenze a quello dei tempi.

Per essere estremamente grossolani e animati da senso pratico, possiamo vedere la trasformata di Fourier come un'estensione del concetto di *serie* di Fourier che già abbiamo incontrato e utilizzato nel nostro corso. Abbiamo in particolare già mostrato come una generica funzione *periodica* di frequenza angolare ω possa essere espressa come somma di (tante) funzioni seno e coseno, ognuna di frequenza angolare $k\omega$, con k intero. I pesi della somma li abbiamo definiti *coefficienti* dello sviluppo in serie di Fourier, e, all'epoca, abbiamo anche stabilito una regola per la loro determinazione.

La trasformata di Fourier rappresenta, in pratica, l'applicazione dello stesso concetto a funzioni *non periodiche* ma dipendenti dal tempo secondo un andamento "qualsiasi". Detta $g(t)$ una tale funzione, si ha che la sua trasformata di Fourier è, a parte eventuali coefficienti di normalizzazione,

$$\mathscr{F}\left\{g(t)\right\} = \tilde{g}(f) = \int_{-\infty}^{\infty} g(t)\exp(j2\pi ft)dt\ , \tag{1}$$

dove $j=\sqrt{-1}$ è l'unità immaginaria e con $f=\omega/(2\pi)$ indichiamo una frequenza. La $\tilde{g}(f)$, calcolata per una data f (ovvero per l'intervallo infinitesimo $f, f+df$), rappresenta in sostanza il valore dei coefficienti (complessi) dell'espansione di Fourier della $g(t)$.

L'anti-trasformata può essere definita come

$$\mathscr{F}^{-1}\left\{\tilde{g}(f)\right\} = g(t) = \int_{-\infty}^{\infty} \tilde{g}(f)\exp(-j2\pi ft)df\ , \tag{2}$$

sempre a parte eventuali coefficienti di normalizzazione.

La bellezza dell'analisi di Fourier può essere facilmente compresa ricordando quanto abbiamo fatto, per esercizio, con la serie di Fourier. All'epoca abbiamo osservato come il comportamento di un sistema (un filtro passa-basso o passa-alto, o una combinazione dei due) per un segnale periodico non sinusoidale $g_{in}(t)$ (un'onda quadra o triangolare) potesse essere ben riprodotto utilizzando la funzione di trasferimento $T(f)$ determinata per segnali sinusoidali, e ricostruendo il segnale di uscita $g_{out}(t)$ come sovrapposizione di tante componenti che rappresentavano l'applicazione della funzione di trasferimento alle componenti del segnale in ingresso. Questa procedura può essere generalizzata, affermando che

$$g_{out}(t) = \mathscr{F}^{-1}\left\{\tilde{g}_{out}(f)\right\} \tag{3}$$

$$\tilde{g}_{out}(f) = T(f)\tilde{g}_{in}(f) \tag{4}$$

$$\tilde{g}_{in}(f) = \mathscr{F}\{g_{in}(t)\}\ , \tag{5}$$

cioè, in sostanza, che il comportamento del sistema nel dominio delle frequenze è descritto da una semplice operazione di prodotto tra funzione di trasferimento e trasformata di Fourier del segnale in ingresso, e che il segnale di uscita può essere ricostruito anti-trasformando tale prodotto (tutto questo, a parte eventuali coefficienti di normalizzazione ha, come sapete, il nome di "prodotto di convoluzione").

In questa nota, e almeno per questo anno accademico, non applicheremo l'analisi di Fourier a questa interessantissima casistica, limitandoci solo a eseguire trasformate di Fourier $\tilde{g}(f)$ di segnali $g(t)$ dipendenti dal tempo e acquisiti sperimentalmente con Arduino in qualche esperienza. Usando un altro linguaggio, si può dire che la $\tilde{g}(f)$ rappresenta lo *spettro* (la parola spettro ha a che fare con la dipendenza dalla frequenza) delle componenti di Fourier a frequenza $f, f + df$ della $g(t)$ considerata. Dunque in questa nota ci proponiamo di costruire lo spettro di alcuni segnali acquisiti in laboratorio.

Osserviamo che tale spettro è, per sua definizione, complesso. Tralasciando per quest'anno commenti, pur interessantissimi, di carattere formale e concettuale, se si è interessati a conoscere l'*ampiezza* di questi coefficienti ci si può limitare a calcolare il *modulo* della Eq. 1. Lo spettro così ottenuto, cioè il grafico di $|\tilde{g}(f)|$, rappresenta la (densità) dell'ampiezza delle componenti del segnale a diverse frequenze che, sovrapposte fra loro, danno la $g(t)$. Questo spettro non è necessariamente normalizzato, per cui nel seguito non daremo importanza al "valore numerico" della grandezza $|\tilde{g}(f)|$.

A. DFT e FFT

Dal punto di vista matematico, l'espressione di Eq. 1 è ricchissima di conseguenze e implicazioni, sia nella fisica classica che in quella quantistica (e relativistica). Tuttavia, come al solito, la fisica non è la patria degli integrali, e neanche delle derivate, e il calcolo esplicito dell'integrale che compare nella definizione può essere affrontato solo per casi "modello", quelli che interessano la soluzione di semplici, o semplicissimi, esercizi da libro di testo.

Esistono dei validi strumenti numerici che permettono di eseguire il calcolo in maniera *discreta*, cioè partendo da una $g(t)$ definita per punti, che dunque non è una funzione, ma rappresenta, per esempio, una misura campionata nel tempo; il risultato della trasformazione è anche uno spettro discreto, generalmente complesso. Questi strumenti, uno dei quali è appunto la FFT, sono presenti in tutti i software di analisi dati, compreso, ovviamente (e con diverse implementazioni che qui non discuteremo), Python. Se siete interessati, potete sicuramente trovare in rete, o altrove, informazioni su come funzionano gli algoritmi per la DFT, e verificare come l'aggettivo fast della FFT sia giustificato da un metodo di calcolo molto efficiente in termini di risorse.

Prima di procedere, diamo qualche chiarimento sull'implementazione FFT in Python a cui facciamo esplicito riferimento. Il problema principale da risolvere riguarda la *scala* di f, cioè quanto valgono il minimo e il massimo valore delle f da usare come asse orizzontale per lo spettro. Infatti, se l'Eq. 1 non dà alcuna indicazione sulla presenza di limiti in f, il calcolo numerico (discreto) impone delle regole precise.

Senza entrare nei dettagli, le regole principali sono le seguenti:

1. la FFT funziona al meglio, cioè senza la necessità di applicare artifici particolari, se l'array di partenza, quello che rappresenta la $g(t)$ (in forma discreta, per punti), è composto da 2^n punti, con n intero. Vedete che, spesso, questo requisito è automaticamente soddisfatto quando i dati di partenza sono quelli acquisiti con una delle varie combinazioni di sketch e script che abbiamo impiegato con Arduino nel corso dell'anno.

2. Nell'implementazione da noi usata, l'array trasformato è fatto da $2^{n-1} + 1$ punti. Supponendo array di partenza costituiti da $2^{11} = 2048$ punti: la nostra FFT è rappresentata da array reali di 1025 punti. Il punto "in più", quello che rende dispari questo valore, si riferisce a $|\tilde{g}(f = 0)|$ e, per i nostri scopi, ha scarsa importanza. Esso infatti rappresenta il valore medio del segnale, calcolato nella durata complessiva dell'acquisizione e influenzato, ad esempio, da eventuali offset, bias, etc..

3. Lo spettro di Fourier ottenuto parte sempre da $f = 0$ e si estende fino a $f_{max} = 1/(2\Delta t_{eff})$, dove, per noi, Δt_{eff} rappresenta l'intervallo temporale effettivo tra due campionamenti successivi.

4. L'affermazione precedente accende un campanello di allarme: usando Arduino, l'intervallo Δt_{eff} è normalmente definito con un'accuratezza piuttosto limitata a causa dei (noti) limiti nella definizione e nella misura dei tempi intrinseci nel funzionamento del microcontroller. Nella sua implementazione ordinaria, l'algoritmo FFT richiederebbe dati equispaziati temporalmente, per cui l'uso di Arduino è inevitabilmente accompagnato da una incertezza, difficilmente valutabile, nella costruzione dello spettro.

5. La risoluzione in frequenza dello spettro ottenuto, cioè la distanza Δf (in unità di f) tra due punti consecutivi dello spettro, è ovviamente pari a $f_{max}/(2^{n-1})$. Poiché l'intervallo complessivo di durata dell'acquisizione è $\Delta T \simeq 2^n \Delta t_{eff}$, dove il simbolo $\simeq$ può essere sostituito per i nostri scopi dal simbolo $=$, si ha anche $\Delta f = 1/\Delta T$. Essendo Δt_{eff} fluttuante, nelle nostre condizioni, nella pratica ci serviremo proprio della misura di ΔT per stabilire la scala in frequenza dei nostri spettri, assumendo implicitamente che il campionamento sia equispaziato temporalmente.

6. Sulla base di quanto appena affermato, è evidente che la risoluzione temporale aumenta all'aumentare della durata temporale del record di dati. Visto che nelle nostre acquisizioni operiamo tipicamente con un intervallo nominale di campionamento relativamente breve (decine o centinaia di μs, in genere), costruire un record di durata sufficiente per garantire una ragionevole risoluzione in frequenza implica acquisizioni di record relativamente lunghi, cioè costituiti da un numero relativamente grande di dati. Come sappiamo, questo può essere facilmente rea-

lizzato (con segnali ripetitivi) usando strategie di sincronizzazione per Arduino.

Dal punto di vista pratico, poiché le grandezze campionate sono, nel nostro caso, *reali* e il nostro interesse è quello di determinare il *modulo* della trasformata di Fourier, il comando di Python che possiamo usare è `abs(numpy.fft.rfft(V))`, dove V rappresenta l'array di dati della nostra misura (il segnale) e `abs` indica l'operazione di calcolo del modulo dell'array complesso prodotto dalla FFT. Questo comando genera un nuovo array, che possiamo graficare in funzione di un array, da costruire a parte, che riporta i valori discreti della frequenza f determinati come sopra specificato. Il grafico ottenuto rappresenta lo spettro del segnale $\tilde{V}$.

II. L'ESERCIZIO

L'esercizio, che ha carattere assolutamente facoltativo e a cui si può rispondere con qualche grafico e qualche riga di commento (da inviare, al solito, via e-mail), prevede di calcolare tramite FFT gli spettri di qualche acquisizione realizzata con Arduino nel corso delle esperienze. Per i motivi prima esposti, si consiglia di impiegare acquisizioni con record lunghi, per esempio quelle create dalla combinazione di sketch e script `synclong2016` e `harmlong`. Normalmente queste combinazioni creano record composti da 2048 coppie di dati, tempo (in μs) e valore digitalizzato (in unità arbitrarie, ovvero, nel nostro linguaggio, digit): la trasformata di Fourier di questi dati dà luogo ad un array composto da 1025 punti, compreso lo zero.

L'esercizio può essere compiuto su qualsiasi file in vostro possesso, che rappresenti l'andamento temporale di un segnale, con caratteristiche preferibilmente note e riferibili a precise situazioni fisiche di interesse nel corso.

Qualche esempio di dati che potrebbero essere impiegati è elencato qui nel seguito.

A. Onde quadre, triangolari (e sinusoidali)

Nelle prime esperienze in cui abbiamo usato Arduino ci siamo posti il problema di verificare la ricostruzione dei segnali periodici prodotti dal generatore di forme d'onda. Noi già conosciamo il metodo, quello della serie di Fourier, adatto per determinare i coefficienti dell'espansione. Come ricordate, in un precedente esercizio abbiamo proprio usato la serie di Fourier per ricostruire numericamente delle forme d'onda periodiche. Qui la finalità dell'esercizio è diversa: si parte da dati "reali", necessariamente fisici, cioè dotati di "imperfezioni" dovute al generatore o al processo di campionamento/digitalizzazione.

Per intenderci, se avete un record che rappresenta la digitalizzazione di un'onda quadra o triangolare, la FFT dovrebbe mostrare un picco principale alla frequenza dell'onda stessa (a parte il contributo della frequenza zero, che corrisponde alle componenti continue di offset o bias, e che, per i nostri scopi, non è rilevante), più altri picchi minori che corrispondono alle diverse armoniche. La frequenza di questi picchi dovrebbe essere nel corretto rapporto con la frequenza fondamentale e anche la loro ampiezza dovrebbe soddisfare le aspettative (ripensate all'espansione in serie di seni e coseni per queste forme d'onda: vedrete che le armoniche presenti dovrebbero essere solo le dispari e che i coefficienti dovrebbero seguire un andamento specifico per la forma d'onda). Aveste anche un'acquisizione di "pinna di squalo" con un numero sufficiente di punti, potrebbe anche essere interessante studiarne la trasformata di Fourier.

I condizionali che ho volutamente impiegato stanno a ricordare che la realtà sperimentale è, appunto, una realtà: infatti potrebbero essere presenti dei rumori sovrapposti al segnale (tipica è la presenza di rumori attorno alla frequenza di rete), oppure potrebbero diventare rilevanti eventuali effetti legati al campionamento (intervallo fluttuante, digitalizzazione non perfetta, etc.). Infine, poiché le forme d'onda sono anch'esse "reali", cioè prodotte da un dispositivo reale quale il generatore di forme d'onda, potrebbero presentare delle evidenti differenze rispetto alle aspettative. Per rendersene conto, può essere interessante eseguire lo spettro dell'acquisizione di una forma d'onda sinusoidale: in questo caso, la "teoria" (la matematica delle funzioni) prevede una singola componente armonica, centrata alla frequenza dell'onda e tale da approssimare una funzione delta della frequenza. Molto probabilmente la realtà suggerirà come il segnale prodotto non sia perfettamente sinusoidale (nessun segnale può essere perfettamente armonico, come ben sapete) attraverso l'allargamento del picco spettrale e la comparsa di ulteriori contributi armonici.

B. Oscillatore smorzato

Un valido esercizio per apprezzare la bellezza dell'analisi di Fourier è costituito dalla trasformazione del segnale acquisito con l'oscillatore armonico smorzato RLC. Nel nostro corso abbiamo analizzato il comportamento di questo circuito prima nel dominio dei tempi, ottenendo dei segnali rappresentativi di un'oscillazione smorzata, e poi nel dominio delle frequenze, ricostruendo la funzione di trasferimento del circuito e analizzando il suo andamento al variare della frequenza.

La trasformata di Fourier permette di passare da un dominio all'altro. Dunque, almeno tralasciando le "imperfezioni" dovute al campionamento, il modulo dello spettro FFT del segnale acquisito in determinate condizioni deve riprodurre l'andamento del guadagno, o attenuazione, nelle stesse condizioni. Potete facilmente verificarlo, osservando in particolare come lo spettro ottenuto sia piccato attorno alla frequenza di risonanza e come la sua larghezza dipenda dal fattore di qualità, ovvero dallo smorzamento dell'oscillatore. Se, in particolare, disponete di acquisizioni riferite all'oscillatore senza e con nuclei conduttori inseriti nella bobina, potete facilmente verificare come lo spettro si “abbassi” e si “allarghi” quando lo smorzamento diventa più rilevante.

Potete credermi se affermo che tutto questo io l’ho verificato, anche se, per motivi di tempo e di voglia, in questa prima versione della nota non vi mostro l’esito delle mie prove. Dunque siete invitati a svolgere in prima persona l’esercizio che, secondo me, ha un valido contenuto educativo!