

SETTIMANA MATEMATICA

Laboratorio “Matematica, Musica, e Computer”

Parte 2: Modificare i suoni con effetti

20 – 22 aprile 2022

1 Qualche nuovo comando MATLAB

Per gli esercizi di oggi ci faranno comodo delle istruzioni di MATLAB leggermente più avanzate di quelle che abbiamo utilizzato fino ad ora.

1.1 Cicli for

Come in quasi tutti i linguaggi di programmazioni, in MATLAB è possibile effettuare cicli for, ovvero istruzioni che vengono ripetute al variare di un indice. Ad esempio, potremmo calcolare il fattoriale di N con le istruzioni:

```
f = 1;
for j = 2 : N
    f = f * j;
end
```

Questo vi potrà essere utile in alcuni esercizi, come ad esempio quello sulla piramide di Chichen-Itza.

Esercizio 1.1. *Si provi a replicare la costruzione del vettore dei numeri dispari fra 1 e 31 tramite un ciclo for, inizializzando prima un vettore vuoto $\mathbf{x} = []$, e poi assegnando $\mathbf{x}(j) = 2*j - 1$; per $j = 1, \dots, 16$.*

2 Simulare l’eco nel Grand Canyon

Cominciamo con un esempio semplice: una sorgente sonora (ad esempio, una persona che batte le mani) si trova in un punto A , ad una certa distanza da una parete rocciosa, posta in un punto B ; supponiamo che il tutto avvenga in uno spazio unidimensionale, per non complicarci eccessivamente la vita.

Le onde sonore viaggiano nell’aria a circa $340 \frac{m}{s}$, e dunque possiamo determinare il tempo che impiegheranno a raggiungere il punto B , venire riflesse, e tornare indietro ad A . Se la distanza $|B - A| = l$, ci aspettiamo che questo tempo sia di $t_1 = 2 \cdot \frac{l}{340}$. Il suono di ritorno sarà più debole di quello dell’andata, per via della dispersione nell’aria e della parziale riflessione da parte della parete rocciosa.



Come abbiamo già discusso, questo ritardo corrisponde ad effettuare una *convoluzione*, ovvero un prodotto di polinomi, fra $s(z)$ e $w(z)$, dove $w(z) = 1 + \lambda z^k$, dove k è il ritardo (come numero di sample, nel nostro caso $k = \lfloor 44100 \cdot t_1 \rfloor$); la convoluzione si può calcolare in MATLAB con il comando

```
w = [ 1, zeros(1, k-1), 1 ];
eco = conv(s, w);
```

Notiamo che il comando `zeros` permette di creare un vettore di zeri. Può essere utile sapere che la funzione `floor` calcola la parte intera inferiore:

```
>> floor(2.3)
```

```
ans = 2
```

Nota: Per leggere un suono da un file, possiamo usare il comando `audioread` come segue:

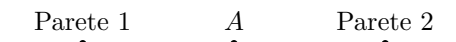
```
[s, Fs] = audioread('nomefile.wav');
```

In output, `s` sarà il vettore con il sampling del segnale $s(t)$, e `Fs` la frequenza di campionamento (per tutti i nostri esempi, `Fs = 44100`).

Esercizio 2.1. Si provi a simulare questo effetto con MATLAB, realizzando uno script `eco.m` che, preso un vettore di sample rappresentante un suono con frequenza 44100 Hz, ci aggiunga un eco. La distanza dalla parete rocciosa può essere scelta a piacimento. Riprodurre il suono ottenuto utilizzando il comando `sound`.

Una volta che l'eco generate ci soddisfa, possiamo provare a riprodurre un esempio più complesso. Supponiamo di trovarci in una specie di Grand Canyon, compresi fra due pareti rocciose (ad esempio una davanti ed una dietro di noi); sempre per non complicarci la vita, assumeremo sempre di vivere in uno spazio unidimensionale, in cui le pareti rocciose e la sorgente sonora siano puntiformi.

In ogni caso, possiamo schematizzare la situazione in questo modo:



Nel disegno qui sopra le pareti sono alla stessa distanza dalla sorgente A , che semplifica un po' la trattazione.

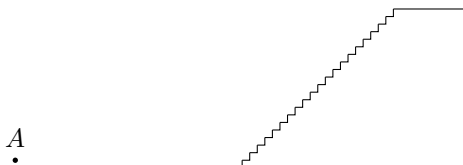
Esercizio 2.2. Si provi ad adattare lo script realizzato in precedenza per simulare l'eco in questa nuova situazione. Si tenga conto del fatto che il suono rimbalzerà prima su una parete e poi sull'altra, riducendosi di intensità ad ogni rimbalzo. Questo processo in teoria continuerebbe all'infinito, ma si può ignorarlo una volta che il suono passa sotto la soglia dell'udibile, ad esempio quando si è ridotto di intensità del 90%. Si provi a testare questo nuovo eco su vari suoni.

3 La piramide di Chichen-Itza

La piramide Maya di Chichen-Itza, in Messico, è famosa per un curioso fenomeno: si dice che battendo le mani di fronte alla sua grande scalinata, sarà possibile ottenere risposta dall'uccello sacro dei Maya, il Quetzal.

In effetti, provate a cercare qualche video su Youtube con le parole chiave “Chichen-Itza handclap echo” (ad esempio <https://www.youtube.com/watch?v=bvJGRpaqaV0>) e troverete i video di molti turisti che si registrano mentre battono le mani di fronte alla piramide. La spiegazione completa di questo fenomeno richiederebbe un po' di lavoro, ma possiamo provare a produrne un modello semplificato.

Schematizziamo la situazione immaginando una persona (per noi rappresentata da un punto A) alla base di una scalinata.



Quando A batte le mani, il suono viene riflesso sui vari gradini, e torna indietro. Ogni gradino si trova però ad una distanza diversa da A , e dunque il suono impiegherà un tempo diverso per tornare alla sorgente, traducendosi in una sequenza di echi leggermente sfasati ed uguale al numero di gradini.

Esercizio 3.1. Si provi a simulare l'eco sentito da A considerando come distanza dalla base della piramide 10 metri, e 150 gradini di altezza e profondità di 20 centimetri. Si consideri come velocità del suono 340 metri al secondo, determinando il tempo necessario ad ogni eco per tornare ad A .

Si costruisca il vettore w e si provi ad applicarlo al suono ottenuto leggendo il file `handclap.wav`, che contiene un battito di mani. Si imponga un'attenuazione del suono riflesso, dovuto all'assorbimento dei gradoni, senza tener conto della distanza.

Per caricare il suono in MATLAB sarà sufficiente dare il comando:

```
>> s = audioread('handclap.wav');
```

e come al solito, si potrà applicare il filtro con

```
>> s2 = conv(s, w); sound(s, 44100);
```

Notiamo che il nostro modello non è estremamente raffinato; per simulare davvero il suono prodotto dalla piramide bisognerebbe tenere conto del fatto che i gradini più alti riflettono solo le basse frequenze, mentre quelli più bassi le alte frequenze. Tuttavia, questo primo esperimento è sufficiente per produrre dei suoni interessanti.

4 Riverbero

Ora che abbiamo capito come far funzionare l'eco, possiamo spingerci a realizzare effetti più avanzati. Un classico è il cosiddetto riverbero, che simula la sensazione di diffusione del suono all'interno di un ambiente (ad esempio, pensiamo al “rimbombo” della musica all'interno di una cattedrale).

La simulazione accurata di questo effetto richiederebbe molte raffinatezze tecniche, che noi ignoreremo; con un po' di impegno, riusciremo però a ottenere comunque un effetto soddisfacente.

Una possibile strategia per ottenere un effetto di riverbero soddisfacente è il seguente:

- Si scelgono n_e istanti di tempo fra 0 e T .
- Per ognuno di questi istanti di tempo, si aggiunge un effetto di eco con quel ritardo, e con un certo coefficiente di attenuazione fra 0 e $\frac{1}{2}$.
- Si trasforma il segnale originale aggiungendo questi echi.

Esercizio 4.1. Si provi a realizzare un buon riverbero seguendo lo schema introdotto con $n_e = 50$ e $T = 2$. Per quanto sorprendente, si ottengono dei buoni risultati anche scegliendo gli istanti dell'eco casualmente, e similmente scegliendo casualmente anche le loro attenuazioni.

Il risultato si può comunque migliorare tenendo conto che echi più lunghi potrebbero avere attenuazioni più forti. Si provi a migliorare l'effetto incorporando nel modello queste considerazioni.

Consiglio: Si utilizzino i comandi `rand` e `randi` per ottenere numeri casuali fra 0 ed 1 oppure interi casuali in un determinato intervallo. Per scoprire come si usa un comando, utilizzare `help rand` oppure `help randi`.

Esercizio 4.2. Si provi a testare i vari effetti realizzati su una propria registrazione. È possibile registrare un breve audio di 5 secondi tramite il microfono in dotazione alle cuffie con la funzione `registra`, fornita fra i file `.m` per il secondo laboratorio.

5 Un sintetizzatore più avanzato (opzionale)

Il sintetizzatore che abbiamo implementato ieri in `crea_suono` era molto rudimentale, ed in effetti non si può dire che il flauto (o gli eventuali altri strumenti che avessimo identificato) siano così realistici. Si può provare a migliorarlo aggiungendo alcuni effetti che sono presenti nella vita reale.

Ad esempio, un flauto non avrà una frequenza esattamente costante nel tempo, ma oscillerà intorno a quella desiderata (di poco). Lo stesso discorso vale per il volume.

Esercizio 5.1. Si modifichi la funzione `crea_suono` aggiungendo un effetto di vibrato (sulla frequenza e/o sul volume); ad esempio, per modificare l'altezza della nota, è sufficiente “distorcere il tempo”, rimpiazzando la funzione $s(t)$ con

$$s_V(t) := s(t + \epsilon \sin(\gamma t)),$$

dove γ regola quanto veloce è l'effetto di vibrato, e ϵ di quanto varia la frequenza. Valori ragionevoli potrebbero essere $\epsilon \approx 0.01$, ed $\gamma \approx 50$. In maniera analoga per il volume, si potrebbe utilizzare $s_A(t) = s(t) \cdot (1 + \epsilon \sin(\gamma t))$. Chiaramente, i due effetti possono essere combinati. Si provi ad ottenere il sintetizzatore migliore possibile.