Progetto MATLAB
Fondamenti di Calcolo Numerico

Emanuele Intagliata
Prof. Carlo De Falco

A.A. 2018-2019

Indice

1 Introduzione

Questo progetto vuole illustrare un algoritmo adattivo per la ricostruzione di sample mancanti in segnali audio campionati che possono essere localmente descritti attraverso processi auto-regressivi.
In particolare, in questo elaborato discutiamo l'implementazione in ambiente MATLAB dell'algoritmo presentato nell'articolo [1].

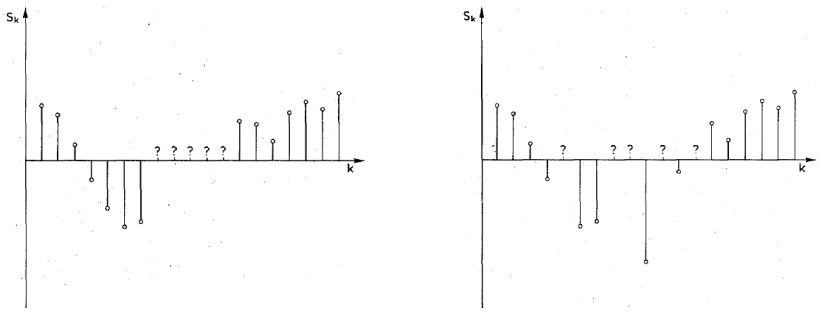
Si è mostrerà come questo metodo dia risultati soddisfacenti, in particolare per segnali audio digitali. Il metodo descritto nel paragrafo 3 usa l'informazione contenuta nei sample vicini noti, minimizzando, in funzione dei sample incogniti e dei coefficienti del modello AR, la somma dei quadrati degli errori residui. Questo metodo può essere usato sia in modo non-iterativo che in modo iterativo. Nel caso iterativo ad ogni passo la stima corrente dei sample mancanti è usata per calcolare una nuova stima. Il metodo è infine testato su segnali, di tipo generico e audio, artificialmente compromessi.

Tutte i codici MATLAB commentati di cui tratteremo nel corso della relazione sono disponibili in allegato per verificare i risultati ottenuti.

1.1 Organizzazione della relazione

Questa relazione è organizzata come segue. Nella Sezione 2 è presentata una trattazione teorica del metodo di interpolazione. In particolare vengono presentate le ipotesi fatte, viene discusso l'algoritmo e vengono esposte alcune considerazioni qualitative sull'errore di interpolazione. Nella Sezione 3 viene discusso il codice che implementa l'algoritmo in ambiente MATLAB. Vengono discusse le scelte effettuate e il funzionamento del codice allegato.

Nella Sezione 4 sono mostrati i risultati sui test effettuati e vengono presentate alcune conclusioni. In coda a questa relazione è presente un'Appendice sulla fattorizzazione di Cholesky.



(a) Burst unico di sample incogniti consecutivi. (b) Pattern casuale di sample incogniti.

Figura 1: Sequenza contenente bursts di sample incogniti.

2 Trattazione teorica

Premessa: Poiché l'algoritmo è stato sviluppato in ambiente MATLAB, la trattazione matematica in questo progetto, pur seguendo quella dell'articolo [1] di riferimento, differisce da essa in indici, pedici e apici, ridefiniti in conformità alla sintassi del linguaggio utilizzato.

2.1 Il problema e le ipotesi

L'articolo [1] tratta il problema della ricostruzione (o interpolazione) di sample mancanti in segnali campionati, in particolare in segnali audio. Viene presentato un algoritmo capace di ricostruire in modo soddisfacente i sample mancanti.
Le ipotesi che facciamo sono:

- la *posizione* dei sample mancanti è nota;
- i sample mancanti sono circondati da un numero sufficiente ampio di sample dal valore noto;
- il segnale è a banda limitata[1].

Le sequenze contenenti sample mancanti possono presentarsi in gruppi isolati e consecutivi (detti *bursts*, Figura 1a) o in pattern sparsi e casuali (Figura 1b). L'algoritmo presentato risulta efficace in entrambi i casi[2].

La stima del valore dei sample mancanti è ottenuta minimizzando la somma dei quadrati dell'errore residuo. Un'analisi statistica contenuta nell'articolo [1, Section II.B] dimostra che, per un *burst* di sample mancanti, l'errore quadratico medio atteso per sample converge alla varianza del segnale quando la lunghezza della sequenza mancante tende a infinito.

2.1.1 Modellizzazione dei segnali audio come processi AR

Si assume che i segnali da interpolare possano essere modellizzati come *processi auto-regressivi (AR) di ordine finito*[3].
Il metodo presentato nell'articolo [1] di riferimento è *adattivo* nel senso che, data una sequenza finita di dati, per prima cosa stimiamo i parametri del modello AR. Noti questi i sample ignoti possono essere ottenuti come *soluzione di un sistema lineare di equazioni.*
In realtà sia i parametri del modello AR che i sample ignoti potrebbero essere ottenuti in un unico passaggio minimizzando una funzione dipendente da entrambi. Tuttavia questa funzione contiene termini del quarto ordine e minimizzarla è un problema non banale. In questo progetto, come anche nell'articolo

[1] Ipotesi sempre verificata per segnali audio digitali.
[2] Il numero di sample sconosciuti è da intendersi come numero totale dei sample mancanti.
[3] La scelta di modellizzare i segnali audio come processi AR può essere motivata dal fatto che molti segnali che incontriamo comunemente (segnali vocali, segnali audio tonali) possono essere così modellizzati.

[1] di riferimento, adotteremo un **approccio sub-ottimale**. In questo approccio i parametri del modello AR sono stimati da una sequenza incompleta di dati. Partendo da essi viene fatta una stima del valore dei sample mancanti. Questo può essere considerato il primo passo di un algoritmo iterativo rapidamente convergente, come discusso nella Sezione 3.

Si assuma che s_k , $k = -\infty, \cdots, +\infty$, sia una realizzazione di un processo auto-regressivo stazionario $\tilde{s}_k$, $k = -\infty, \cdots, +\infty$. Ciò significa che esistono un intero positivo p, detto **ordine** del modello AR, e dei numeri $a_1, \cdots, a_{p+1}$, con $a_1 = 1$ detti **coefficienti** del modello AR. Sia inoltre $\tilde{e}_k$, $k = -\infty, \cdots, +\infty$ un processo di **rumore bianco** con varianza σ_e^2 tale che:

$$a_1\tilde{s}_k + a_2\tilde{s}_{k-1} + \cdots + a_{p+1}\tilde{s}_{k-p} = \tilde{e}_k \tag{1}$$

con $k = -\infty, \cdots, +\infty$.
Per convenzione si assuma che $a_k = 0$ per $k < 1$ o $k > p + 1$.

I **dati disponibili** consistono nel segmento s_k , $k = 1, \cdots, N$, di una realizzazione del processo AR $\tilde{s}_k$, $k = -\infty, \cdots, +\infty$. Come ipotizzato nell'introduzione, si assuma che i sample incogniti si presentino agli istanti di tempo noti $t(1), \cdots, t(m)$, dove:

$$1 < (p + 1) \leq t(1) < \cdots < t(m) \leq (N - p)$$

Il **problema** in esame consiste nella *stima* degli m sample incogniti $s_{t(1)}, \cdots, s_{t(m)}$, dei p parametri del modello AR $a_2, \cdots, a_{p+1}$ e di σ_e^2 dai dati disponibili in modo che la sequenza ricostruita sia il più coerente possibile con il modello assunto. Quantitativamente la ricostruzione dovrà essere tale da minimizzare la somma dei quadrati dell'errore residuo $e_{p+1}, \cdots, e_N$.
Esistono molti metodi per la stima dell'ordine di processi auto-regressivi (come riportato nell'articolo [2]), così come delle funzioni MATLAB già implementate. Nonostante ciò, l'articolo [1] in esame assume che, se p è incognito, verrà scelto p in funzione di m, numero dei sample incogniti:

$$p = 3m + 2 \tag{2}$$

Questa relazione piuttosto arbitraria si dimostra dare buoni risultati di interpolazione.

2.2 Presentazione del metodo di interpolazione

Adottiamo la seguente nomenclatura:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_{p+1} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} s_{t(1)} \\ \vdots \\ s_{t(m)} \end{bmatrix}$$

Inoltre siano $\hat{\mathbf{a}}$ la stima di $\mathbf{a}$ e $\hat{\mathbf{x}}$ la stima di $\mathbf{x}$.
Come anticipato, la stima di $\mathbf{a}$ e $\mathbf{x}$ è formulata sotto forma di *problema di minimizzazione*. Sceglierò $\hat{\mathbf{a}}$ e $\hat{\mathbf{x}}$ tali che a funzione

$$Q(\mathbf{a}, \mathbf{x}) = \sum_{k=p+1}^{N} \left| \sum_{l=0}^{p} a_k s_{k-l} \right|^2 = \sum_{k=p+1}^{N} |e_k|^2 \tag{3}$$

sia minima in funzione di $\mathbf{a}$ e $\mathbf{x}$.
Determinati $\hat{\mathbf{a}}$ e $\hat{\mathbf{x}}$, potrò stimare σ_e^2 come:

$$\hat{\sigma}_e^2 = \frac{1}{N-p-m} Q(\hat{\mathbf{a}}, \hat{\mathbf{x}}) \tag{4}$$

La particolare scelta di minimizzare la funzione $Q(\mathbf{a}, \mathbf{x})$ per ottenere le stime di $\mathbf{a}$ e $\mathbf{x}$ ha molteplici motivazioni, discusse nel dettaglio nell'articolo [1]. In particolare, nell'*Appendice A* di [1] si dimostra che, ipotizzando che i valori dei campioni abbiano una funzione di densità di probabilità di tipo Gaussiano, minimizzare $Q(\mathbf{a}, \mathbf{x})$ rispetto a $\mathbf{x}$ sia lo stesso che trovare la minima stima della varianza di $\mathbf{x}$, noti p ed $\mathbf{a}$.

Si nota immediatamente come minimizzare la funzione $Q(\mathbf{a}, \mathbf{x})$ rispetto a $\mathbf{a}$ e $\mathbf{x}$ sia un problema tutt'altro che banale in quanto essa contiene termini del quarto ordine come $a_2^2 s_{t(m)}^2$. Useremo quindi un **approccio sub-ottimale**:

1. Scegliamo una stima iniziale $\hat{\mathbf{x}}^{(0)}$ per il vettore $\mathbf{x}$ dei sample incogniti, per esempio $\hat{\mathbf{x}}^{(0)} = \mathbf{0}$.
2. Successivamente minimizziamo $Q(\mathbf{a}, \hat{\mathbf{x}}^{(0)})$ in funzione di $\mathbf{a}$ per ottenere una stima di $\hat{\mathbf{a}}$.
3. Infine minimizziamo $Q(\hat{\mathbf{a}}, \mathbf{x})$ in funzione di $\mathbf{x}$ per ottenere una stima di $\hat{\mathbf{x}}$ dei sample incogniti.

Entrambe le minimizzazioni sono possibili in quanto $Q(\mathbf{a}, \mathbf{x})$ è quadratica sia in funzione di $\mathbf{a} \in \mathbb{R}^p$ che in funzione di $\mathbf{x} \in \mathbb{R}^m$. Si può dimostrare che

$$Q(\mathbf{a}, \mathbf{x}) = \mathbf{a}^T C(\mathbf{x}) \mathbf{a} + 2\mathbf{a}^T \mathbf{c}(\mathbf{x}) + c_{11}(\mathbf{x}) \tag{5}$$

dove $C(\mathbf{x})$ è la *matrice di auto-covarianza* di dimensioni $p \times p$ definita come:

$$C(\mathbf{x}) = (c_{ij}(\mathbf{x}))_{i,j=1,\cdots,p}$$

e

$$\mathbf{c}(\mathbf{x}) = \begin{bmatrix} c_{0,1}(\mathbf{x}) \\ \vdots \\ c_{0,p}(\mathbf{x}) \end{bmatrix}$$

con

$$c_{ij}(\mathbf{x}) = \sum_{k=p+1}^{N} s_{k-i} s_{k-j}, \qquad i, j = 0, 1, \cdots, p.$$

Allo stesso modo, si può dimostrare che

$$Q(\mathbf{a}, \mathbf{x}) = \mathbf{x}^T B(\mathbf{a})\mathbf{x} + 2\mathbf{x}^T \mathbf{z}(\mathbf{a}) + D(\mathbf{a}) \tag{6}$$

dove

$$B(\mathbf{a}) = (b_{(t(i)-t(j))+(p+1)})_{i,j=1,\cdots,m} \tag{7}$$

$$\mathbf{z}(\mathbf{a}) = \begin{bmatrix} z_1(\mathbf{a}) \\ \vdots \\ z_m(\mathbf{a}) \end{bmatrix}$$

con

$$z_i(\mathbf{a}) = \sum_{k=1}^{2p+1} b_k s_{t(i)-k+(p+1)}, \qquad i = 1, \cdots, m,$$

$$b_{l+(p+1)} = \sum_{k=1}^{p+1} a_{k+p} a_{k+p+l}, \qquad l = -p, \cdots, +p \tag{8}$$

and $D(\mathbf{a}) \in \mathbb{R}$ dipendente da $\mathbf{a}$ e dai sample noti.
Quindi $\hat{\mathbf{a}}$ e $\hat{\mathbf{x}}$ saranno dati rispettivamente da

$$C(\hat{\mathbf{x}}^{(0)})\hat{\mathbf{a}} = -\mathbf{c}(\hat{\mathbf{x}}^{(0)}), \tag{9}$$

e

$$B(\hat{\mathbf{a}})\hat{\mathbf{x}} = -\mathbf{z}(\hat{\mathbf{a}}). \tag{10}$$

Il metodo appena descritto per il calcolo dei coefficienti del modello AR attraverso una sequenza di campioni è noto come **metodo dell'autocovarianza** ed è descritto nel dettaglio nell'articolo [3]. Un metodo diverso, il metodo dell'auto-correlazione, verrà esposto nella Sezione 2.2.1.

Sostituendo la (9) nella (4) segue che

$$\hat{\sigma}_e^2 = \frac{1}{N-p-m}(c_{0,0}(\hat{\mathbf{x}}) + \hat{\mathbf{a}}^T \mathbf{c}(\hat{\mathbf{x}})). \tag{11}$$

Il metodo di interpolazione appena descritto può essere considerato come il primo passo di un **algoritmo iterativo** nel quale, ad ogni passo, nuovi coefficienti $\hat{\mathbf{a}}$ del modello sono stimati attraverso la (9) usando, anziché $\hat{\mathbf{x}}^{(0)}$, il vettore $\hat{\mathbf{x}}$ dei campioni incogniti precedentemente stimato attraverso la (10). I coefficienti ottenuti potranno essere usati per ottenere una nuova stima dei sample mancanti e così via. Sembra chiaro, almeno intuitivamente, che $Q(\mathbf{a}, \mathbf{x})$ che decresca all'aumentare dei passi di iterazione, tendendo a qualche numero non-negativo. Si potrebbe sperare che la sequenza ottenuta converga nel punto in cui $Q(\mathbf{a}, \mathbf{x})$ raggiunge il suo minimo globale. Sfortunatamente, è estremamente difficile provare un risultato del genere[4] (vedi Sezione 3).

[4] Si può tuttavia dimostrare che questo procedimento iterativo di minimizzazione assomiglia molto all'algoritmo EM (Expectation-Maximisation) di stima di parametri per le distribuzioni, vedi[1, Appendice B].

2.2.1 Calcolo dei coefficienti del modello AR

Il calcolo di $\hat{\mathbf{a}}$ attraverso la (9) è, in realtà, un problema noto come **metodo dell'auto-covarianza**. Esso è ampiamente discusso nel dettaglio nell'articolo [3]. Nello stesso articolo è esposto un algoritmo molto efficiente per il calcolo di $\hat{\mathbf{a}}$ attraverso la (9) in $O(p^2)$ operazioni: il **metodo dell'auto-correlazione**[5] , che qui riportiamo.

Metodo dell'auto-correlazione In questo metodo anziché risolvere il sistema lineare (9) verrà risolto il sistema

$$\mathbf{R}\hat{\mathbf{a}} = -\mathbf{r}$$

dove R è la *matrice di auto-correlazione* definita come:

$$\mathbf{R} = \left(r_{(i-j)+(p+1)}\right)_{i,j=1,\cdots,p} \tag{12}$$

e

$$\mathbf{r} = \begin{bmatrix} r(1) \\ \vdots \\ r(p) \end{bmatrix}$$

dove

$$r_{j+p+1} = \frac{1}{N}\sum_{k=1}^{N-|j|} s_k s_{k+|j|}, \quad j = -p, \cdots, +p$$

è una stima pesata dell'autocorrelazione di $j-esimo$ lag di auto-correlazione di $\tilde{s}_k, k = -\infty, \cdots, +\infty$. Esistono molteplici modi di risolvere il sistema (12). Per semplicità, noi risolveremo il sistema con la funzione `mldivide`, nativa in ambiente MATLAB.[6]

2.2.2 Calcolo dei campioni incogniti

Per il calcolo di $\hat{\mathbf{x}}$ attraverso la (10) ci affidiamo alla funzione MATLAB per la risoluzione di sistemi lineari `mldivide`. Per comprendere cosa succede quando questa funzione viene chiamata è utile analizzare la struttura della matrice $\mathbf{B}(\hat{\mathbf{a}})$ definita nel dettaglio in (9) e (8). Una proprietà molto utile della matrice $\mathbf{B}(\hat{\mathbf{a}})$ è che è **definita positiva**. In ambiente MATLAB questa proprietà è facilmente *verificabile* con l'ausilio della funzione `chol` (vedi anche Appendice A). Matematicamente, questa proprietà non è in generale semplice da dimostrare: fortunatamente lo è per la matrice $\mathbf{B}(\hat{\mathbf{a}})$.

[5] Un calcolo approssimato di $\hat{\mathbf{a}}$ da comunque un risultato di interpolazione soddisfacente: dunque vari e altri metodi possono essere applicati indifferentemente.

[6] Il sistema può essere risolto in $O(p^2)$ operazioni dall'algoritmo di Levinson-Durbin[4]. Tuttavia non ci sono differenze significanti nei risultati di interpolazione ottenuti dall'autore attraverso differenti metodi.

Dimostrazione. Una *matrice simmetrica* B a coefficienti reali si dice definita positiva se, per ogni vettore riga $\mathbf{v}$, il prodotto $\mathbf{v}B\mathbf{v}^T$ è *strettamente* positivo. Come si può vedere dalla sua definizione in (7) la matrice $\mathbf{B}(\hat{\mathbf{a}})$ risulta essere simmetrica a coefficienti reali. Si può anche notare che essa ha valori costanti b_{p+1} sulla sua diagonale principale. Inoltre, per un generico vettore riga $\mathbf{v} = (v_1, \cdots, v_m)$:

$$\mathbf{vB}(\hat{\mathbf{a}})\mathbf{v^T} = \sum_{i=1}^{m}\sum_{j=1}^{m}\mathbf{B}(\hat{\mathbf{a}})_{ij}v_iv_j = \sum_{k}\left|\sum_{i=1}^{m}\hat{a}_{k+t(i)}v_i\right|^2$$

che possiamo ottenere inserendo la (7) e la (8) nel termine centrale. Sia $\mathbf{v}$ non banale, ovvero abbia *almeno un elemento non nullo*, e sia i' l'indice più grande per cui $v_{i'} \neq 0$. Allora la sommatoria (presente nel termine a destra) per $k = -t(i')$ è uguale a $v_{i'}^2$, essendo $\hat{a}_l = 0$ per $l \notin [1, p+1]$, $a_1 = 1$ e $v_i = 0$ per $i > i'$. Quindi, se $\mathbf{v}$ non è banale, il termine sulla destra consiste in termini non-negativi di cui almeno uno è positivo. Ciò dimostra che $\mathbf{B}(\hat{\mathbf{a}})$ è *definita positiva.* □

Il fatto che $\mathbf{B}(\hat{\mathbf{a}})$ sia definita positiva ci permette di utilizzare la *decomposizione di Cholesky* [7] per calcolare $\hat{\mathbf{x}}$ in $O(m^3)$ operazioni.

Decomposizione della matrice $\mathbf{B}(\hat{\mathbf{a}})$. Nella decomposizione di Cholesky la matrice $\mathbf{B}(\hat{\mathbf{a}})$ è scomposta nel prodotto:

$$B(\hat{\mathbf{a}}) = LL^T \tag{13}$$

dove L è una matrice $m \times m$ triangolare inferiore.
In MATLAB, data una matrice B definita positiva, è possibile trovare la matrice L con il comando `L = chol(B)`. Il sistema lineare $B(\hat{\mathbf{a}}) = LL^T\hat{\mathbf{x}} = -\mathbf{z}(\hat{\mathbf{a}})$ viene risolto risolvendo per sostituzione all'indietro, trovando $\mathbf{y}$ da $L\mathbf{y} = -\mathbf{z}(\hat{\mathbf{a}})$ e quindi $\hat{\mathbf{x}}$ da $L^T\hat{\mathbf{x}} = \mathbf{y}$.
Per gli elementi della matrice L in [1, Appendix C] viene trovato il seguente risultato:

$$1 \leq L_{jj} \leq \sqrt{b_{p+1}}, \quad j = 1, \cdots, m, \tag{14}$$

$$\sum_{i=1}^{m} L_{ij}^2 = b_{p+1}, \quad j = 1, \cdots, m, \tag{15}$$

così che

$$|L_{ij}| \leq \sqrt{b_{p+1} - 1}, \qquad i = 1, \cdots, j-1, \quad j = 1, \cdots, m. \tag{16}$$

Il limite inferiore nella (14) è da tenere in considerazione in quanto i gli elementi L_{jj}, $j = 1, \cdots, m$, sono usati come divisori nel processo di sostituzione

[7] Come illustrato in [1, Sezione III.B], esistono molteplici metodi per la risoluzione di $\mathbf{B}(\hat{\mathbf{a}})$. Tuttavia in questo progetto (come nell'articolo di rifermento) verrà esposta la sola decomposizione di Cholesky.

all'indietro: se fossero troppo piccoli si potrebbe perdere accuratezza nel calcolo. Per esperienza dell'autore dell'articolo [1], nel caso di musica digitalizzata, b_{p+1} ha valori modesti, del tipo $b_{p+1} < 4$, tale che gli elementi L_{jj} non diventino troppo piccoli.

Cosa si nasconde dunque dietro il comando Matlab `mldivide` ? Nel caso di una matrice B simmetrica con elementi sulla diagonale principale (il che non implica che la matrice sia definita positiva), MATLAB tenta un una *fattorizzazione di Cholesky* (`chol`). Se la matrice è una matrice sparsa viene prima applicato un algoritmo di preordinamento [5, p.166].
Nell'Appendice A è descritto un algoritmo per trovare gli elementi di L.

2.3 Alcune considerazioni qualitative sull'errore di interpolazione

Nella presente Sezione sono riassunte alcune conclusioni *qualitative* tratte da [1, Sezione II.B], utili all'interpretazione dei risultati presentati nella Sezione 4 dei test.
Analizziamo dapprima il caso di un burst unico di m campioni **consecutivi** poiché merita più attenzione rispetto al caso più generale di burst sparsi. Per primo caso si può dimostrare che:

- di solito, gran parte dell'energia dell'errore si concentra al centro del burst;
- per burst molto lunghi l'errore quadratico di interpolazione per sample tende all'energia del segnale per sample.

Il risultato trovato per il suddetto caso particolare può essere sfruttato anche per derivare conclusioni per il caso più generale. Infatti si può dimostrare che:

- in questo caso l'errore di interpolazione risulta minore del caso di un unico burst di lunghezza m;
- se $\tilde{e}_k$ ha funzione di densità di probabilità *gaussiana* allora l'errore di interpolazione può essere analizzato più nel dettaglio: si può ad esempio osservare che la $var(\tilde{e})$ è più piccola se lo spettro in frequenza del segnale ha molti picchi (e.g. *Sinusoidi*).

3 L'algoritmo

L'algoritmo MATLAB in allegato mette in pratica l'*approccio sub-ottimale* descritto nella Sezione 2.2. Per prima cosa l'algoritmo controlla che i dati in ingresso rispettino le ipotesi fatte nella Sezione 2.1. Dopo aver definito le variabili, inizia un ciclo iterativo in cui:

1. Viene calcolata una stima dei coefficienti del modello AR tramite la funzione `a_estimator`.

2. Viene calcolata una stima dei campioni incogniti tramite la funzione `x_estimator`. I campioni stimati sostituiscono i campioni incogniti nel segnale ricostruito.

3. Con il segnale così stimato si potrà iterare dal punto (1), trovando una nuova stima dei parametri del modello AR.

Alcune note sulla versione iterativa del metodo di interpolazione Le *proprietà di convergenza* di questo metodo sono descritte nel dettaglio in [1, Appendix B]. Si trova che iterare questo algoritmo può migliorare i risultati ottenuti se il numero di sample disponibili è relativamente piccolo. Sebbene il metodo iterativo risulti, nella pratica e nei test della Sezione 4, convergere molto rapidamente [8], non sembra facile dimostrare risultati di convergenza soddisfacenti.

L'algoritmo si serve di alcune funzioni ausiliarie.

La funzione `a_estimator`

Sintassi `[a] = a_estimator(sig, p, met)`

Descrizione Questa funzione adatta un modello auto-regressivo (AR) del p-esimo ordine al segnale in ingresso `sig`. In quanto $a(1) = 1$ sempre, il vettore restituisce solo gli elementi $\mathbf{a} = (a(2), \cdots, a(p+1)$.

Nel nostro progetto, per il calcolo dei coefficienti del modello AR, abbiamo messo a confronto due differenti metodi: il **metodo dell'auto-covarianza** e il **metodo dell'auto-correlazione**.

`a_estimator(__, met)` specifica il metodo da usare per il calcolo dei coefficienti:

- se `met == 'acov'` verrà usato il **metodo dell'auto-covarianza**
- se `met == 'acor'` verrà usato il **metodo dell'auto-correlazione**

I due metodi sono descritti nel dettaglio nella Sezione 2.2. Il metodo dell'auto-covarianza è per semplicità implementato attraverso la funzione `arcov` di MATLAB. Questa funzione, tramite appunto il metodo dell'auto-covarianza, stima i parametri di un modello AR di ordine p.
Il metodo dell'auto-correlazione è invece interamente implementato come descritto nella Sezione 2.2.1 . Ciò implica la risoluzione del sistema (12). Esistono molteplici modi per risolverlo. Per semplicità, avendo verificato con il comando `chol` che $\mathbf{R}$ è *definita positiva*, noi risolveremo il sistema con la funzione `mldivide`, nativa in ambiente MATLAB.
Si è lasciata possibilità di scelta per l'utente per far vedere come il calcolo approssimato di $\hat{\mathbf{a}}$ dia comunque un risultato di interpolazione soddisfacente. Dai

[8] cfr. la Sezione 4 di questa relazione e [1, Section 4] per maggiori dettagli.

test della Sezione 4 è possibile infatti vedere che, nella maggior parte dei casi, non c'è una notevole differenza tra i risultati di interpolazione ottenuti tramite il metodo dell'auto-covarianza o il metodo dell'auto-correlazione.

La funzione `x_estimator`

Sintassi `[sig] = x_estimator(a, t, sig_comp)`

Descrizione La funzione ritorna una stima del segnale ricostruito `sig`, noti i coefficienti del modello auto-regressivo `a`, le posizioni dei sample incogniti contenute in `t` ed il segnale compromesso `sig_comp`. L'algoritmo usato è quello descritto nella Sezione 2.2 . In particolare, viene trovato il vettore dei campioni incogniti `x` tramite l'equazione (10) che, successivamente, viene inserito nel segnale compromesso alle posizioni `t` per ottenere il segnale in uscita `sig`.

L'algoritmo consente all'utente alcune scelte per la ricostruzione del segnale. L'utente può decidere:

- il *numero di iterazioni* per cui ripetere i passaggi descritti sopra
- il *metodo di stima dei parametri del modello AR* (vedi Paragrafo precedente)
- il metodo per il calcolo di σ_e^2

Inoltre, nella versione dimostrativa allegata, l'utente può scegliere quale segnale compromettere e poi ricostruire. I segnali a disposizione sono quelli presentati nei test della Sezione 4.

4 Test e risultati

In quest'ultima Sezione discutiamo i risultati del metodo adattivo di interpolazione discusso in questa relazione ottenuti nella ricostruzione dei seguenti segnali di test.

1. *Unica sinusoide pura.* Sequenza di 1 secondo di una sinusoide a frequenza 500 Hz campionata a $F_s = 44100Hz$, unico burst di $m = 4$.
2. *File audio percussivo.* Campione di rullante della durata di $93ms$, campionato a $F_s = 44100Hz$, unico burst di $m = 8$.
3. *Due sinusoidi pure.* Sequenza di 512 sample del segnale dato da

   $$s_n = sin(0.23\pi n + 0.3\pi) + 0.6sin(0.4\pi n + 0.3\pi) \quad con\, n = 1, \cdots, 512.$$

   I sample incogniti sono $m = 12$, divisi 3 burst da 4 sample.

4. *Processo auto-regressivo generato artificialmente.* Realizzazione, artificialmente generata di un processo AR di ordine 10 con spettro in frequenza generico. Sequenze di 512 sample, pattern dei burst da $m = 8$. Nella Tabella 1 si possono trovare i coefficienti del modello AR.

5. *Multiple sinusoidi corrotte da rumore bianco.* Due sinusoidi, le stesse descritte nel Test 3, corrotte da rumore bianco. Viene considerato un rapporto segnale-rumore di $40dB$. Sequenza di 512 sample, con 3 bursts da 4 sample ($m = 12$).

6. *File audio musicale.* Una porzione di circa $17secondi$ del brano di *Ludwig van Beethoven*, *Ode to Joy - Symphony No. 9 in D minor Choral Op. 125*, campionata a $F_s = 44100Hz$, viene corrotta da molteplici bursts di 16 sample con un rate di $10s^{-1}$.

In questa demo i segnali di test sono creati o importati e poi compromessi artificialmente, ponendo arbitrariamente a 0 il valore di alcuni sample.

Come anticipato nella Sezione 3, la nostra demo dell'algoritmo dà la possibilità di scegliere uno dei due metodi di interpolazione descritti nella Sezione 2.2.1. In questa Sezione presentiamo i risultati ottenuti nell'interpolazione dei segnali di test con entrambi i metodi, in particolare i coefficienti trovati sono indicati con c_i per *metodo dell'auto-covarianza* e con r_i per il *metodo dell'auto-correlazione*, dove il pedice i indica il numero di iterazioni effettuate.

Per tutti i segnali di test, i risultati del metodo di interpolazione adattivo sono giudicate in base alla media dell'errore quadratico relativo di interpolazione e

$$e = \frac{\frac{1}{m}\sum_{i=1}^{m}(\hat{s}_{t(i)} - s_{t(i)})^2}{\frac{1}{N}\sum_{i=1}^{N} s_i^2} \qquad (17)$$

Tabella 1: Coefficienti del processo AR utilizzato come segnale di Test 4.

i	a_i
1	1
2	-0.7185
3	0.5502
4	0.7970E-1
5	0.1586E-2
6	0.3802E-1
7	0.5296E-1
8	0.2792E-1
9	-0.11538E-1
10	-0.54262E-1
11	-0.5943E-2

Un media dei valori di e per i segnali di test è presentata nelle Tabelle 2 - 10. Per ogni segnale di test è poi mostrato un grafico del segnale originale sovrapposto al segnale ricostruito nella zona compromessa artificialmente. Sull'asse delle ascisse è indicata la posizione dei sample del segnale campionato. Per questioni di leggibilità tutti i grafici riportano una porzione del segnale, seppur l'algoritmo lavori con l'intero segnale. In particolare, viene rappresentata la porzione di segnale artificialmente compromessa.
Per i Test 2 e 6, oltre le tabelle e i grafici qui riportati, l'algoritmo fornisce anche un ascolto del segnale originale, compromesso e ricostruito per verificare il risultato di ricostruzione ottenuto.

Test 1 Il segnale è una sequenza di 1 secondo di una sinusoide pura a frequenza 500 Hz campionata a $F_s = 44100Hz$ con un unico burst di $m = 4$.
Nella Tabella 2 sono mostrati l'errore quadratico relativo di interpolazione e e lo scostamento massimo Δ del segnale ricostruito dal segnale originale, per 1, 3 e 5 interazioni. Nella Figura 2 è presentato il caso particolare c_5, ovvero il segnale compromesso è stato interpolato attraverso il metodo dell'auto-covarianza per 5 iterazioni.

	c_1	c_3	c_5	r_1	r_3	r_5
e	3.3747E-17	1.3499E-16	3.0372E-16	6.1346E-17	2.4451E-16	5.4948E-16
Δ	0%	0%	0%	0%	0%	0%

Tabella 2: Test 1. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale.

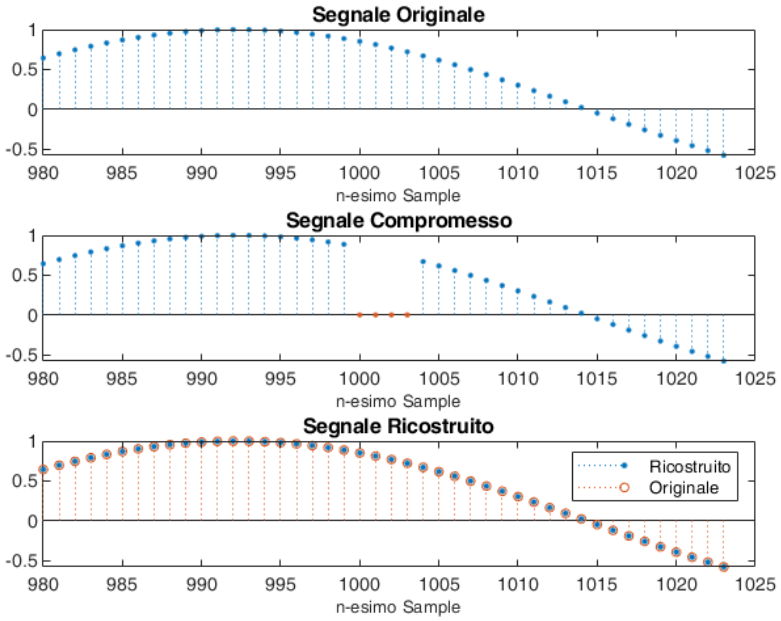


Figura 2: Test 1. Segnale interpolato attraverso il metodo dell'auto-covarianza per 5 iterazioni.

Test 2 Il segnale è un file audio di un campione di rullante della durata di $93ms$, campionato a $F_s = 44100Hz$. Il file è compromesso con un unico burst di $m = 8$.
Nella Tabella 3 è mostrato l'errore quadratico relativo di interpolazione e e lo

scostamento massimo Δ del segnale ricostruito dal segnale originale, per 1 e 3 interazioni. Nella Figura 3 è presentato il caso particolare c_1, ovvero il segnale compromesso è stato interpolato attraverso il metodo dell'auto-covarianza per 1 iterazione.
Per questo segnale la nostra demo da anche un test d'ascolto.

	c_1	c_3	r_1	r_3
e	0.28286	0.27258	0.28286	0.27257
Δ	16.7%	16.4%	16.7%	16.4%

Tabella 3: Test 2. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale.

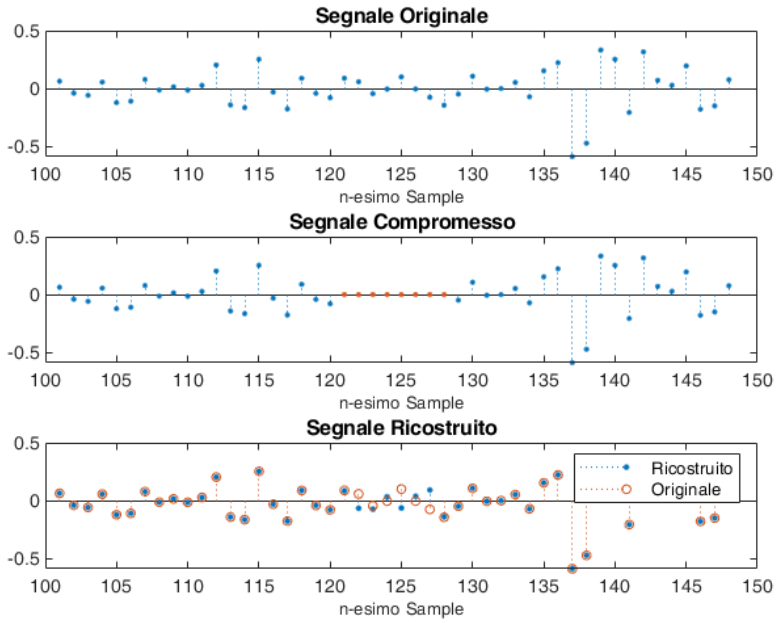


Figura 3: Test 2. Segnale interpolato attraverso il metodo dell'auto-covarianza per 1 interazione.

Test 3 In questo test viene compromesso un segnale costituito da due sinusoidi *pure*. Il segnale è costituito da una sequenza di 512 sample dato dall'equazione descritta sopra. I sample incogniti sono $m = 12$, divisi 3 burst da 4 sample.
Nella Tabella 4 è mostrato l'errore quadratico relativo di interpolazione e e lo scostamento massimo Δ del segnale ricostruito dal segnale originale, per 1 e 3 interazioni. Nella Figura 4 è presentato il caso particolare r_3, ovvero il segnale

compromesso è stato interpolato attraverso il metodo dell'auto-correlazione per 3 iterazioni.

	c_1	c_3	r_1	r_3
e	1.40E-09	5.60E-09	1.23E-07	4.17E-07
Δ	0%	0%	0%	0.1%

Tabella 4: Test 3. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale.

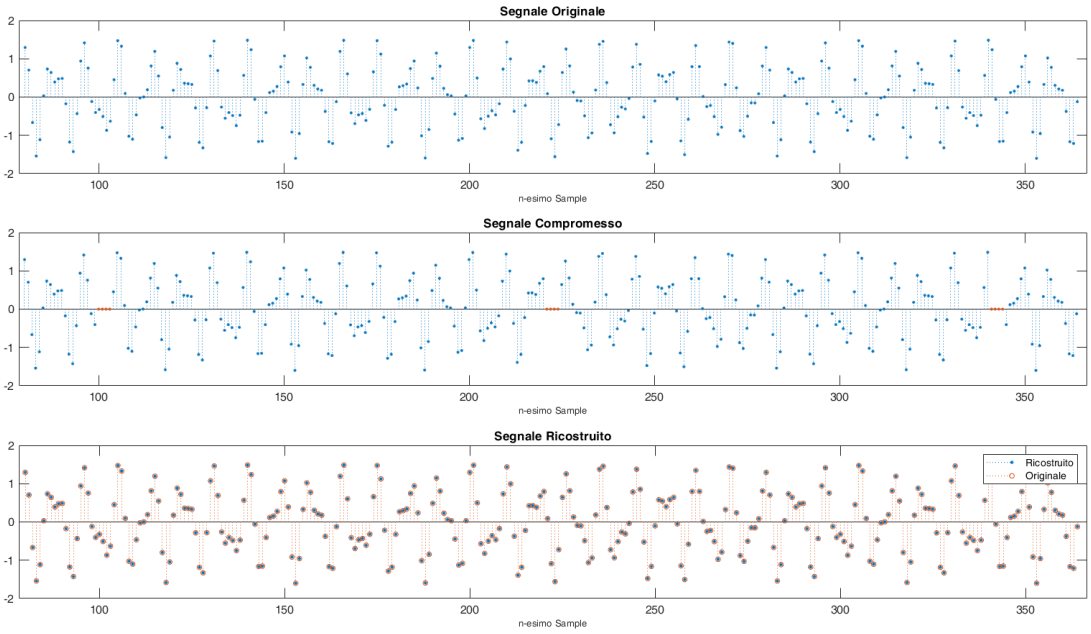


Figura 4: Test 3. Segnale interpolato attraverso il metodo dell'auto-correlazione per 3 interazioni.

Test 4 Il segnale compromesso è una realizzazione, artificialmente generata, di un processo auto-regressivo di ordine 10 con spettro in frequenza generico. La sequenza è di 512 sample e i sample incogniti sono costituiti da 2 pattern di burst da 8 sample ($m = 16$).
In questo caso, avendo $p = 3m + 2 = 50$, ci si può chiedere cosa accada forzando l'algoritmo a lavorare con $p = 10$. Nella nostra demo principale è presentata una versione semplificata di questo test. In allegato è però presente una versione più avanzata di questo test, `test4_b.m` che permette alcune scelte supplementari:

- Si può scegliere se forzare far lavorare l'algoritmo forzando p al valore originale o se farlo scegliere in automatico.
- Si può scegliere se lavorare con i *veri* coefficienti del modello AR o se stimarli con uno dei due metodi descritti nella Sezione 2.2.1.

Nella Tabella 5 si possono trovare i coefficienti del modello AR originali affiancati a quelli stimati dall'algoritmo. In questo caso abbiamo forzato l'algoritmo a lavorare con $p = 10$.
Nelle Tabelle 6 e 7 è mostrato l'errore quadratico relativo di interpolazione e e lo scostamento massimo Δ del segnale ricostruito dal segnale originale, per 1 e 3 interazioni. Nella Tabella 6 l'algoritmo è stato forzato a lavorare con $p = 10$, mentre nella Tabella 7 l'algoritmo sceglie in automatico i coefficienti p. Nella prima Tabella inoltre i risultati di interpolazione sono comparati con quelli ottenuti utilizzando i veri coefficienti del modello AR, indicati con f (che sta per *fixed coefficients*).
Si può vedere come l'errore di interpolazione per entrambi i metodi di stima dei coefficienti non differisce significativamente dall'errore di interpolazione attraverso i veri coeffecienti del modello AR. Come anticipato nella Sezione 2.2.1, una stima grossolana dei coefficienti di predizione, magari da un set incompleto di dati, non influenza significativamente la qualità della ricostruzione.
Nella Figura 5 è presentato il caso particolare r_3, ovvero il segnale compromesso è stato interpolato attraverso il metodo dell'auto-covarianza per 3 iterazioni usando $p = 10$.

i	f	r_1	r_3	c_1	c_3
1	1	1	1	1	1
2	-0.7185	−0.9738	−0.9747	−0.9594	−0.9603
3	0.5502	0.6264	0.6273	0.6050	0.6059
4	0.7970E-1	−0.4140	−0.4141	−0.3962	−0.3962
5	0.1586E-2	−0.1080	−0.1086	−0.1223	−0.1229
6	0.3802E-1	−0.0210	−0.0199	−0.0240	−0.0228
7	0.5296E-1	−0.0221	−0.0237	−0.0245	−0.0262
8	0.2792E-1	−0.0576	−0.0551	−0.0558	−0.0531
9	-0.11538E-1	−0.1005	−0.1027	−0.0967	−0.0992
10	-0.54262E-1	0.1072	0.1085	0.1063	0.1077
11	-0.5943E-2	0.0080	0.0075	0.0128	0.0123

Tabella 5: Test 4. Stima dei parametri del modello AR. Parametri originali e parametri stimati.

	f	r_1	r_3	c_1	c_3
e	0.16298	0.13385	0.13076	0.13021	0.12721
Δ	25.2%	23.5%	22.7%	23.4%	22.4%

Tabella 6: Test 4. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale per $p = 10$.

	r_1	r_3	c_1	c_3
e	0.17117	0.15928	0.22853	0.20017
Δ	24.6%	23.4%	28.3%	26%

Tabella 7: Test 4. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale utilizzando il numero di coefficienti p scelti dall'algoritmo, $p = 3m + 2$.

Si può vedere dalle Tabelle 6 e 7 come iterare l'algoritmo non dia significativi miglioramenti.

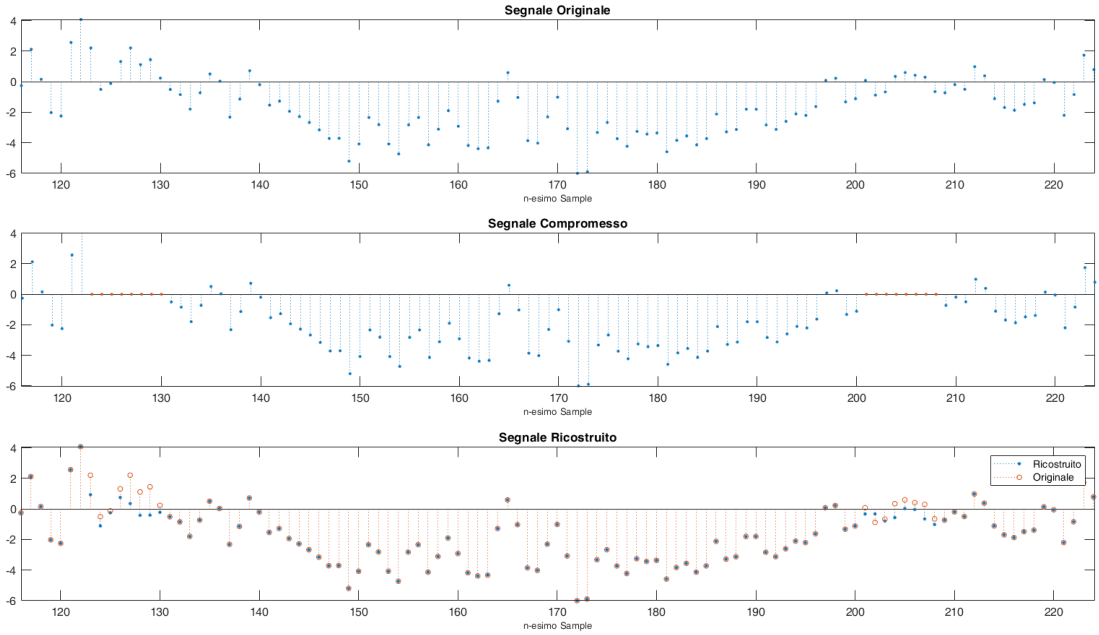


Figura 5: Test 4. Segnale interpolato attraverso il metodo dell'auto-covarianza per 3 iterazioni usando $p = 10$.

Test 5 Il segnale è costituito dalle stesse due sinusoidi del Test 3 corrotte da rumore bianco. Viene considerato un rapporto segnale-rumore di $40dB$ e una sequenza di 512 sample, con 3 bursts da 4 sample ($m = 12$).
Nella Tabella 8 sono mostrati l'errore quadratico relativo di interpolazione e e lo scostamento massimo Δ del segnale ricostruito dal segnale originale, per 1 e 3 interazioni.
Nella Figura 6 è presentato il caso particolare r_3, ovvero il segnale compromesso è stato interpolato attraverso il metodo dell'auto-correlazione per 3 iterazioni.

	r_1	r_3	c_1	c_3
e	2.57E-04	8.74E-05	2.14E-04	2.83E-04
Δ	1.5%	1.8%	1.3%	2.4%

Tabella 8: Test 5. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale.

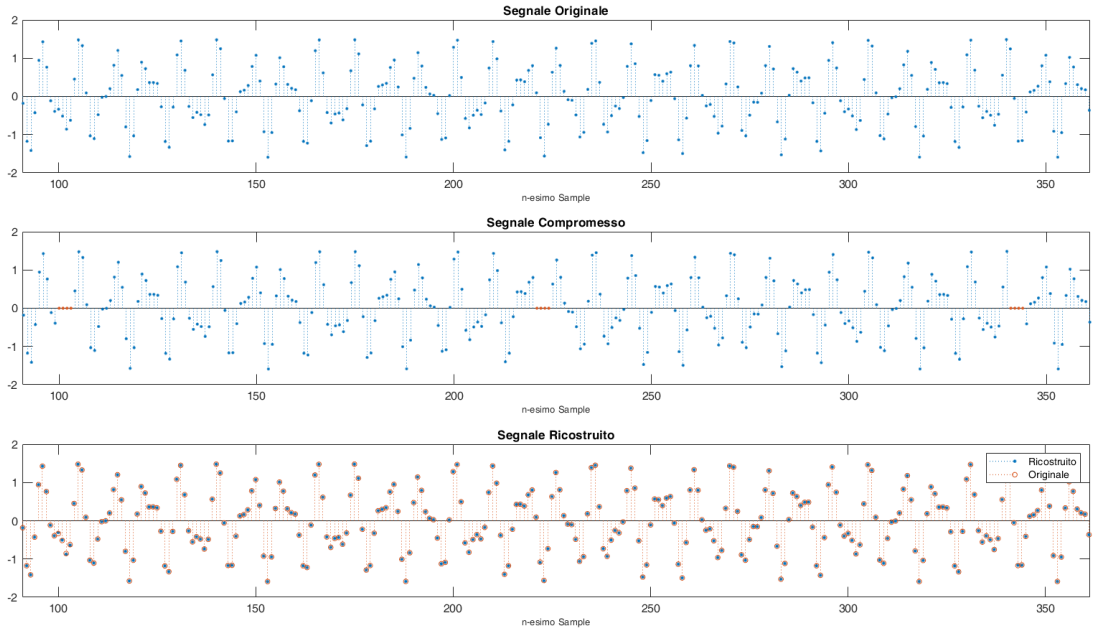


Figura 6: Test 5. Segnale interpolato attraverso il metodo dell'auto-correlazione per 3 iterazioni.

Al decrescere del rapporto segnale-rumore, i risultati di interpolazione del nostro algoritmo peggiorano leggermente. Come si può vedere dalle Tabelle 4 e 8 essi non differiscono significativamente.

Test 6 L'ultimo test è eseguito su un file audio musicale. Una porzione di circa 17 secondi del brano di *Ludwig van Beethoven*, *Ode to Joy - Symphony No. 9 in D minor Choral Op. 125*, campionata a $F_s = 44100Hz$, viene corrotta da molteplici bursts di 16 sample con un rate di $10s^{-1}$. Il numero totale di sample compromessi è molto alto (m è dell'ordine di 10^3). Inoltre, anche per questo segnale, la nostra demo da anche un test d'ascolto: in questo caso la compromissione è perfettamente udibile. La porzione di segnale compromessa inizia al secondo 5 e finisce al secondo 10.
In questo caso p è estremamente alto. Nella Sezione 4 dell'articolo di riferimento per segnali audio musicali viene scelto arbitrariamente $p = min(3m + 2, 50)$.

Nella nostra demo principale non è possibile questa scelta. In allegato è però presente una versione modificata di questo test, `test6_b.m` che la permette. Si può vedere dai risultati come questa scelta arbitraria dia buoni risultati di interpolazione.
Nella Tabella 9 sono mostrati l'errore quadratico relativo di interpolazione e e lo scostamento massimo Δ del segnale ricostruito dal segnale originale, per 1 e 3 interazioni per $p = 3m + 2$. Nella Tabella 9 sono mostrati gli stessi dati per $p = min(3m + 2, 50)$. [9]
Nella Figura 7 è presentato il caso particolare r_1, ovvero il segnale compromesso è stato interpolato attraverso il metodo dell'auto-correlazione per 1 iterazione. Poichè i burst sono molteplici, per migliorarne la leggibilità nella Figura 8 viene mostrata solo una porzione della parte di segnale compromesso.

	r_1	r_3
e	0.032408	0.042897
Δ	10.5%	12.6%

Tabella 9: Test 6. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale per $p = 3m + 2$.

	r_1	r_3	c_1	c_3
e	0.10642	0.10642	0.089467	0.10642
Δ	22%	22%	18.8%	22%

Tabella 10: Test 6. Errore quadratico relativo di interpolazione e e scostamento massimo Δ del segnale ricostruito dal segnale originale per $p = min(3m + 2, 50)$.

Per segnali audio musicali, l'errore quadratico relativo di interpolazione è dello stesso ordine di grandezza di quello di un processo auto-regressivo (Test 4). Si noti come questo metodo di interpolazione sia riuscito a ricostruire in modo soddisfacente burst consecutivi di 16 sample che, in un segnale audio campionato a $F_s = 44100Hz$, corrispondono ad un intervallo temporale di circa $0.36ms$ per burst. Aumentando la lunghezza dei bursts (ad esempio, a 50 sample) l'interpolazione risulta ancora abbastanza buona ma alcuni errori di interpolazione diventano udibili. Può sembrare strano che questo metodo funzioni per bursts di questa entità, che rappresentano un intervallo di tempo di $1.1ms$. Spesso gli errori di ricostruzione fatti non sembrano ridurre la qualità dell'esperienza di ascolto soggettiva: è probabile che ciò sia dovuto a effetti di mascheramento.

[9] Per questioni computazionali il metodo dell'auto-covarianza, per p troppo alti, non viene preso in considerazione in questi test.

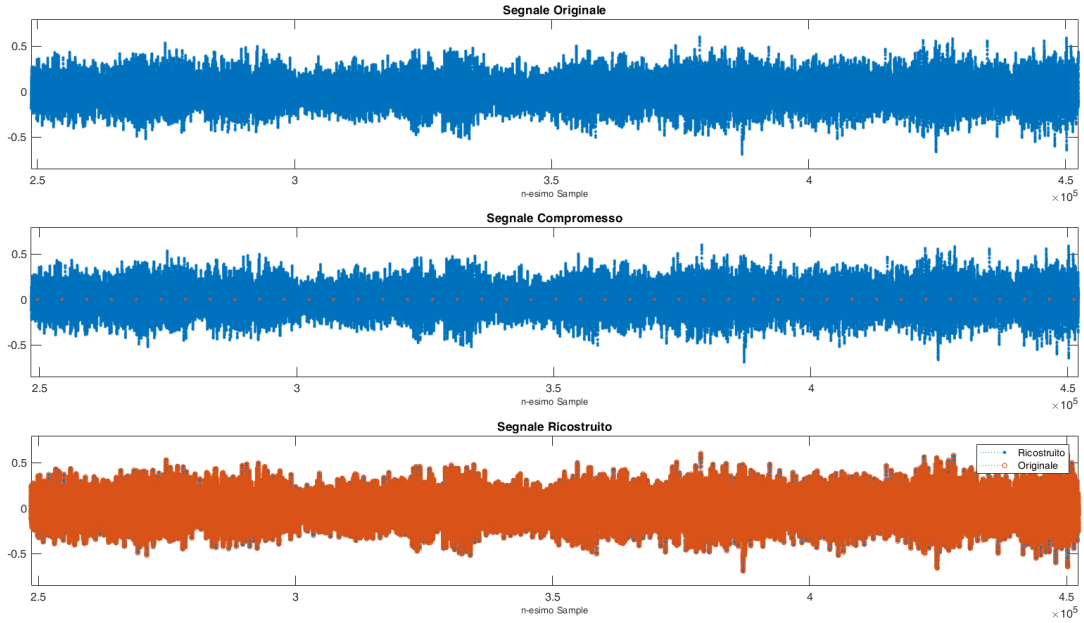


Figura 7: Test 6. Segnale interpolato attraverso il metodo dell'auto-correlazione per 1 iterazione. Porzione di segnale compromessa.

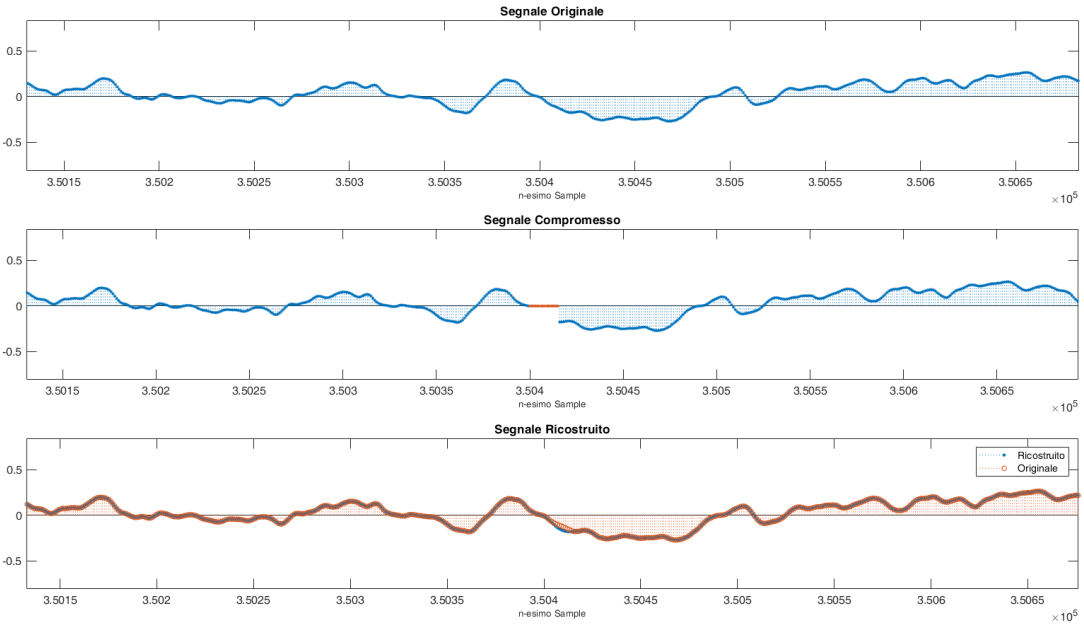


Figura 8: Test 6. Segnale interpolato attraverso il metodo dell'auto-correlazione per 1 iterazione. Particolare.

4.1 Conclusioni

In questo progetto abbiamo presentato un algoritmo adattivo per l'interpolazione di sample corrotti o incogniti in segnali tempo discreti che possono essere modellizzati come processi auto-regressivi, descritto nell'articolo di riferimento [1]. L'obiettivo di questo progetto è stato quello di implementare il suddetto algoritmo in ambiente MATLAB.
I test mostrano come questo algoritmo dia risultati soddisfacenti, in particolare per segnali audio e per bursts piccoli. Questo metodo può essere usato sia iterativamente che non. Nel caso iterativo vengono eseguite minimizzazioni successive utilizzando, ad ogni passo, nuove stime dei sample incogniti. Si è mostrato come iterare migliori la qualità dell'interpolazione nel caso sia abbia una sequenza di dati disponibili relativamente piccolo.
L'algoritmo è stato testato su vari segnali artificialmente compromessi, quali sinusoidi, processi AR artificialmente generati, segnali audio brevi e tracce musicali di durata modesta. I risultati ottenuti sono stati giudicati sia oggettivamente che soggettivamente, con test di ascolto. Dai test della sezione precedente possiamo trarre alcune conclusioni.

- Si può notare come uno scostamento massimo locale Δ non trascurabile non influenzi la qualità della ricostruzione. Ad esempio, i test d'ascolto per i Test 2 e 6 rivelano che l'errore di interpolazione di questi segnali risulta praticamente non udibile.

- Come regola generale si può vedere che pattern generici di bursts sono interpolati con un errore di interpolazione minore rispetto a bursts unici con lo stesso numero di sample.

- Come anticipato nella Sezione 2.3, l'errore di interpolazione per segnali il cui spettro contiene picchi è molto più piccolo di processi il cui spettro è più piatto. Per delle sinusoidi pure in particolare (Test 1 e 3) avremmo $\sigma_e^2 = 0$ e quindi idealmente una ricostruzione con errore di interpolazione nullo. Infatti le Tabelle 2 e 4 mostrano un errore molto piccolo.

- Nella pratica il metodo dell'auto-correlazione da risultati lievemente peggiori rispetto al metodo dell'auto-covarianza perché usa una stima pesata della funzione di auto-correlazione. Se p è scelto alto, ciò ha meno influenza sui risultati. Se si usa il metodo dell'auto-covarianza p va scelto non troppo grande. I questo caso, per più di una iterazione la matrice dell'auto-covarianza diventa *quasi singolare* e i coefficienti di predizione non possono più essere calcolati risolvendo il sistema lineare (9).

- Come si vede dalle Tabelle 3, 6, 7, e 10, per segnali *non* sinusoidali, non ci sono significanti differenze tra i risultati ottenuti utilizzando il metodo dell'auto-covarianza o il metodo dell'auto-correlazione.

Questo metodo di interpolazione si è mostrato avere una solida base matematica. In particolare si è mostrato come le varie minimizzazioni possano essere

effettuate risolvendo in modo efficiente e in maniera stabile particolari sistemi di equazioni lineari.

Appendici

A Fattorizzazione di Cholesky

La matrice $\mathbf{B}(\hat{\mathbf{a}})$ può essere scomposta nel prodotto $B(\hat{\mathbf{a}}) = LL^T$ come abbiamo visto nella Sezione 2.2.2. Possiamo trovare un algoritmo che calcoli gli elementi della matrice triangolare inferiore L scrivendo elemento per elemento la relazione 13:

$$B_{ij} = \sum_{k=1}^{m} L_{ik} L_{kj}^T = \sum_{k=1}^{i} L_{ik} L_{jk} \qquad per\ i \geq j.$$

Per gli elementi della diagonale principale di L ($j = i$) otteniamo:

$$B_{jj} = \sum_{k=1}^{j} L_{jk}^2$$

da cui, togliendo il termine L_{jj} dalla sommatoria, otteniamo:

$$L_{jj} = \sqrt{B_{jj} - \sum_{k=1}^{j-1} L_{jk}^2}$$

Per gli altri elementi di L otteniamo

$$L_{ij} = \frac{1}{L_{jj}} \left(B_{ij} - \sum_{k=1}^{j-1} L_{ik} L_{jk} \right)$$

Il calcolo della matrice L è possibile in MATLAB grazie al comando `L = chol(B)`.

Se la matrice B non fosse definita positiva, il processo di fattorizzazione si bloccherebbe in quanto troverebbe un radicando negativo. In questo caso la funzione `chol` viene terminata e viene ritornato un errore. Nella pratica, questa funzione è un buon modo per verificare se una matrice data è definita positiva.

Riferimenti bibliografici

[1] A. Janssen, R. N. J. Veldhuis, and L. B. Vries, "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, p. 317–330, Apr 1986.

[2] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, pp. 716–723, December 1974.

[3] S. M. Kay and S. L. Marple, "Spectrum analysis—a modern perspective," *Proceedings of the IEEE*, vol. 69, pp. 1380–1419, Nov 1981.

[4] J. Durbin, "The fitting of time-series models," *Rev. Inst. Int. Stat.*, 1960.

[5] *Scientific Computing with MATLAB and Octave.* Springer-Verlag Berlin Heidelberg, 2006.