



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

CORSO DI LAUREA TRIENNALE IN
INFORMATICA E COMUNICAZIONE
DIGITALE

TESI DI LAUREA
IN
CALCOLO NUMERICO

COMPRESSIVE SENSING
NEL MONITORAGGIO DI SISTEMI SMART-GRID

Relatore:

Chiar.ma Prof.ssa Giuseppina SETTANNI

Laureando:

Matteo LUCERI

ANNO ACCADEMICO 2019-2020

*A Federica,
mia straordinaria compagna di vita.*

Abstract

La tecnologia Smart Grid (SG) trasforma la tradizionale rete elettrica da un livello fisico in sistemi che comprendono due livelli affiancati, quello fisico e quello informativo. Raccolta, trasferimento e analisi dell'enorme quantità di dati che possono essere acquisiti da diversi nodi nella rete, insieme all'incertezza causata dai generatori di energia distribuiti, sfidano i metodi standard per la sicurezza e il monitoraggio nelle future SG. Questa tesi presenta un'efficace soluzione dinamica per il monitoraggio delle SG combinando concetti relativi al Compressive Sensing (CS) e allo Sparse Recovery (SR). A seguito di un'analisi preliminare dei concetti enunciati, si propone uno studio atto a implementare in MATLAB un modello SG, con l'obiettivo di mostrare e infine confrontare alcune applicazioni di tecniche algoritmiche di CS-SR che si propongono di ottenere soluzioni efficaci del problema.

Indice

1	Introduzione	1
1.1	Dalle reti elettriche tradizionali alle smart grid: sfide e opportunità	1
1.2	Piena osservabilità dalle informazioni parziali	2
2	Compressive Sensing	4
2.1	Cos'è il Compressive Sensing	4
2.2	Introduzione al Compressive Sensing: origini e contesto	4
2.3	Cenni storici del Compressive sensing	6
2.4	Ambiti d'uso del Compressive Sensing	7
2.5	Quadro matematico	7
2.5.1	Concetti preliminari e notazioni	7
2.6	Fondamenti del Compressive Sensing	11
2.7	Matrice di rilevamento (<i>Sensing matrix</i>)	13
2.8	Algoritmi di ricostruzione	14
2.8.1	Convex Optimization - ottimizzazione convessa	14
2.8.2	Algoritmi Greedy	14
2.9	Compressive Sensing Distribuito (<i>Distributed Compressive Sensing</i>) .	15
3	Applicazioni Compressive Sensing per le Smart Grid	17
3.1	Modello di rete e grafico corrispondente	17
3.1.1	Grafico di una PN	17
3.1.2	Modello di flusso a corrente continua e grafico corrispondente di una PN	17
3.1.3	Distribuzione gaussiana dei parametri	20
3.2	Identificazione topologia sparsa	21
3.2.1	Impostazione del problema sparso di identificazione	22
3.3	Algoritmi SR risolutivi: struttura sparsa di T_I e il problema della correlazione dei dati	23
3.4	Implementazione degli algoritmi	24
3.4.1	Matrice di rilevamento	25
3.4.2	Segnale sparso x e misurazione y	25
3.4.3	LASSO	27
3.4.4	OMP	29
4	Simulazioni: Test e valutazione	31
4.1	Modello e misure di valutazione	31
4.1.1	Scelta del modello di test	31
4.1.2	Scelta delle misure di valutazione	31

4.2	Metodologia	33
4.3	Misurazione	34
4.4	Risultati e valutazione	35
4.4.1	Valutazione LASSO	36
4.4.2	Valutazione OMP	39
4.4.3	Confronto LASSO vs OMP	40
4.5	Conclusioni	42
Appendici		44
A	Test IEEE a 30 Bus	45
B	Test IEEE a 118 Bus	57
C	Test IEEE a 300 Bus	69

Capitolo 1

Introduzione

In questo capitolo, presentiamo una panoramica delle tecnologie relative alle reti elettriche *smart grid*, mostrandone opportunità e sfide, ed evidenziamo come le tecniche di *compressive sensing* trovino applicazione in questo ambito.

1.1 Dalle reti elettriche tradizionali alle smart grid: sfide e opportunità

Una rivoluzione sta avvenendo nella tecnologia delle reti elettriche che tutti conosciamo, supportando lo sviluppo delle cosiddette *smart grids* (SG). Il termine SG si riferisce a reti elettriche in grado di integrare in modo intelligente il comportamento e le azioni di tutti i parametri, le misure e le utenze, connessi a loro. Questa rivoluzione trasforma una tradizionale rete elettrica ad un singolo livello fisico in un'enorme rete a due livelli coesistenti, uno fisico e uno informativo che scorre attraverso il sistema. Queste informazioni includono lo stato di numerosi parametri della rete come tensione dei collegamenti, potenze attive e reattive, angoli di fase, correnti di diramazione e coefficiente di carico, oltre ai comandi di attuazione o controllo inviati alla rete da responsabili e unità di controllo.

Un'altra caratteristica importante delle *smart grids* di nuova generazione è l'aggregazione di micro-fonti di energia rinnovabile nella rete di distribuzione. Nonostante i vantaggi ambientali apportati dall'uso di fonti rinnovabili, la potenza iniettata da queste, come le turbine eoliche e i pannelli solari, sono intrinsecamente casuali. Questa casualità, insieme all'incertezza dei profili di carico, porta a un comportamento probabilistico nel flusso di potenza e carico sui collegamenti della rete.

Un'altra importante problematica è relativa ai tempi, all'efficienza e all'accuratezza dell'analisi dei dati in tempo reale, all'interno di queste enormi reti complesse e interconnesse. I contatori intelligenti raccolgono i dati in una risoluzione di campionamento ridotta e la trasmettono a una struttura centrale (centrale di controllo). Supponendo che la risoluzione di campionamento vari da 15 minuti a 1 secondo, la quantità di dati generati sarà tra 0,25 TB e 250 TB in un anno. La quantità di dati generati dovrebbe inoltre crescere man mano che i vecchi contatori vengono sostituiti con quelli intelligenti. Raccogliere, trasferire e archiviare l'enorme mole di

RETE SMART GRID VS RETE TRADIZIONALE

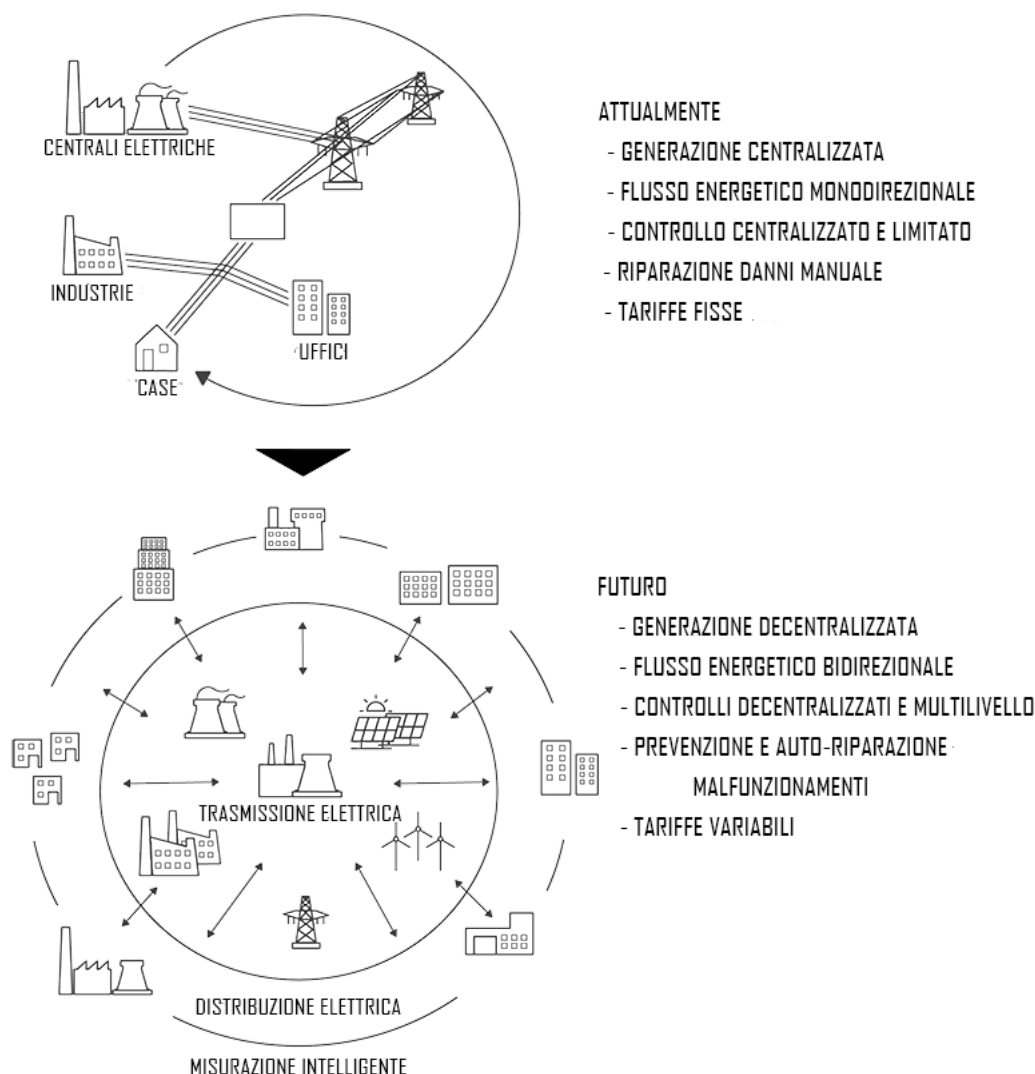


Figura 1.1: Reti smart vs reti tradizionali

dati che fluisce attraverso il livello informativo di una smart grid in grande scala, insieme all'incertezza causata dalla generazione distribuita e dai profili di carico, mette a dura prova gli standard di sicurezza, monitoraggio e controllo della rete.

1.2 Piena osservabilità dalle informazioni parziali

In generale la maggior parte degli approcci di monitoraggio, a livello operativo locale, è altamente dipendente dalla qualità dei dati disponibili forniti dal sistema e funziona male in caso di misurazioni corrotte, in cui potrebbero essere disponibili solo informazioni parziali.

Inoltre, la maggior parte delle metodologie classiche di monitoraggio a livello globale sono state adattate per strutture semplici. Questi metodi di solito soffrono di costi computazionali elevati e la non linearità della formulazione, di fatto, si traduce in

soluzioni intrattabili in caso di applicazioni in tempo reale.

Questi problemi computazionali sono evidenziabili nel caso di reti elettriche su larga scala, dove di solito una grande quantità di dati arriva nelle sale operative di controllo in ogni momento. In molti casi pratici (considerando i tempi, i limiti di memorizzazione e di efficienza energetica), gli operatori potrebbero esser costretti a ridurre in modo drastico la quantità di dati necessari da elaborare, per poi estrarre successivamente in modo accurato tutte le informazioni utili per le analisi del sistema.

Queste intrinseche sfide tecniche hanno portato allo sviluppo di un approccio comune, ovvero all'utilizzo di tecniche di *Compressive Sensing e Sparse Recovery* (CS-SR) nell'analisi delle reti *Smart Grids* e affrontare così i problemi derivati dall'analisi dei big data. Comprimendo i dati, infatti, è possibile trasmetterli sul sistema informativo della SG ad un costo computazionale inferiore e successivamente decomprimerli in modo da elaborarli senza perdere accuratezza nell'informazione.

Nel prossimo capitolo analizzeremo nel dettaglio la tecnica del compressive sensing.

Capitolo 2

Compressive Sensing

In questo capitolo analizzeremo nel dettaglio cos'è il compressive sensing, facendo una panoramica sulle origini, l'evoluzione e le tecniche principali, per poi successivamente studiare le possibili applicazioni alla smart power grid.

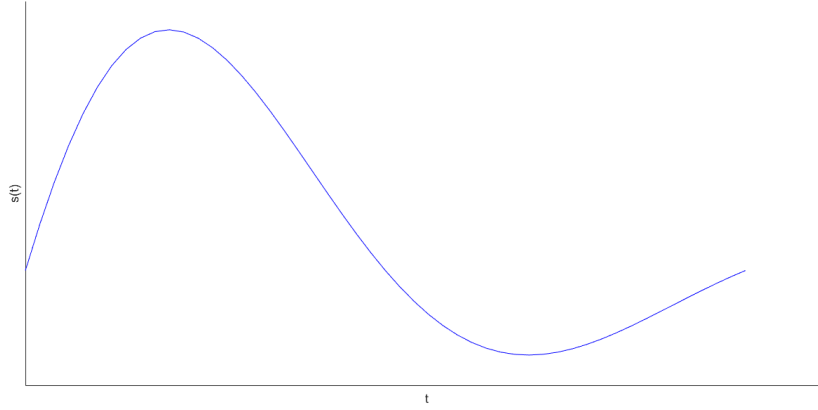
2.1 Cos'è il Compressive Sensing

Il Compressive sensing (CS) è una nuova tecnica di elaborazione del segnale che ha rivoluzionato la scienza dell'analisi dei dati. Grazie alla sua efficienza nel tempo, il framework CS-SR ha trovato un ruolo cruciale nella modellazione e analisi di dati di sistemi su larga scala e ha influenzato significativamente l'era dei Big Data. Dal punto di vista dell'elaborazione del segnale, il CS è un paradigma emergente, che mira ad acquisire direttamente una forma compressa di segnali con comportamento sparso in fase di rilevamento e consente un'elevata qualità di ricostruzione in fase di ricezione. Il CS presenta un paradigma alternativo a quelli di acquisizione tradizionali. Infatti, i paradigmi tradizionali, affermano che il numero di campioni misurati dovrebbero essere almeno uguali al numero di campioni rilevati nel segnale originale al fine di garantire un recupero esatto. Tuttavia, queste condizioni non tengono conto della struttura del segnale. Quindi, se il segnale di interesse è sparso, cioè esso può essere rappresentato da un numero minore di coefficienti diversi da zero rispetto alla sua dimensione originale, il CS afferma che prendendo solo pochi numeri di misurazioni lineari casuali (proiezioni) del segnale sparso, esse saranno sufficienti per acquisire le informazioni salienti dal segnale e così fornire una qualità di ricostruzione accettabile. Il CS mira a spostare la complessità dai sensori, di solito risorse limitate e autoalimentate, al ricevitore che invece è presente su piattaforme informatiche con vincoli più permissivi.

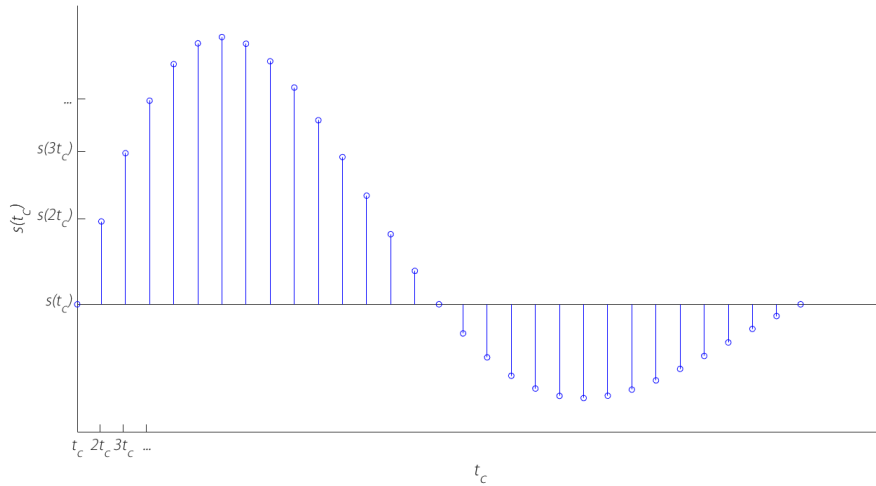
2.2 Introduzione al Compressive Sensing: origini e contesto

Siamo nel mezzo di una rivoluzione digitale che sta guidando lo sviluppo e la diffusione di nuovi tipi di sistemi di rilevamento con fedeltà e risoluzione sempre crescenti. Il fondamento teorico di questa rivoluzione è il lavoro pionieristico di Kotelnikov [32], Nyquist [39], Shannon [44] e Whittaker [52] sul campionamento di segnali. I loro

risultati dimostrano che segnali, immagini, video e altri dati possono essere recuperati esattamente da una serie di campioni distribuiti uniformemente al cosiddetto tasso di Nyquist, il doppio della frequenza più alta presente nel segnale di interesse.



(a) Segnale analogico (continuo)



(b) Segnale campionato (discreto)

Figura 2.1: Segnale analogico e segnale campionato

Sfruttando questa scoperta, gran parte dell'elaborazione del segnale si è spostata dal dominio analogico a quello digitale cavalcando l'onda della legge di Moore. La digitalizzazione ha consentito la creazione di sistemi di rilevamento ed elaborazione più robusti e flessibili, nonché più economici e, di conseguenza, più utilizzati rispetto alle loro controparti analogiche.

Come risultato di questo successo, la quantità di dati generati dai sistemi di rilevamento è cresciuta esponenzialmente. Sfortunatamente però, in molte applicazioni importanti ed emergenti, il tasso di Nyquist risultante è così alto da ottenere sin troppi campioni. Infatti, potrebbe essere semplicemente troppo costoso, o addirittura fisicamente impossibile, costruire dispositivi in grado di acquisire campioni alla velocità necessaria. Nonostante gli straordinari progressi nello sviluppo del-

la potenza computazionale, l'acquisizione e l'elaborazione di segnali continuano a rappresentare una sfida importante da intraprendere.

Per affrontare le sfide logistiche e computazionali legate alla gestione di dati ad alta dimensione, spesso contiamo sulla compressione, che mira a trovare la rappresentazione più concisa di un segnale, in grado di raggiungere un livello di distorsione accettabile.

Una delle tecniche più popolari per la compressione del segnale è nota come *transform coding* e in genere si basa sulla ricerca di una base o di un frame che fornisca rappresentazioni sparse o comprimibili per segnali in una classe di interesse.

Per rappresentazione sparsa, intendiamo che per un segnale di lunghezza n , possiamo rappresentarlo con $k \ll n$ coefficienti diversi da zero; per rappresentazione comprimibile, intendiamo che il segnale è ben approssimato da un segnale con solo k coefficienti diversi da zero.

Sia i segnali sparsi che quelli comprimibili possono essere rappresentati con elevata fedeltà preservando solo i valori e le posizioni dei più grandi coefficienti del segnale. Questo processo è chiamato *approssimazione sparsa* e costituisce la base di schemi di codifica di trasformazione che sfruttano la sparsità e la compressibilità del segnale, inclusi gli standard JPEG, JPEG2000, MPEG e MP3, nonché dello stesso CS.

Sfruttando il concetto di *transform coding*, il CS è emerso come un nuovo framework per l'acquisizione del segnale e la progettazione dei sensori, che consente una riduzione potenzialmente elevata dei costi di campionamento e il calcolo di segnali di rilevamento con rappresentazione sparsa o comprimibile. Mentre il teorema di campionamento di Nyquist-Shannon afferma che è necessario un certo numero minimo di campioni per acquisire perfettamente un segnale arbitrario, quando il segnale è sparso in una base nota, possiamo ridurre notevolmente il numero di misurazioni che devono essere memorizzate. Di conseguenza, quando si rilevano segnali sparsi, potremmo essere in grado di fare meglio di quanto suggerito dai risultati classici. Questa è l'idea fondamentale alla base di CS: piuttosto che prima campionare ad alta velocità e quindi comprimere i dati campionati, vorremmo trovare modi per rilevare direttamente i dati in una forma compressa, cioè ad una frequenza di campionamento inferiore. Il campo del CS è cresciuto dal lavoro di Candès, Romberg, Tao [9, 11, 12, 13, 15] e di Donoho [22], i quali hanno dimostrato che un segnale a dimensioni finite con una rappresentazione sparsa o comprimibile può essere recuperato da un piccolo insieme di misurazioni lineari e non adattative. La progettazione di questi schemi di misurazione, la loro estensione a modelli pratici e a sistemi di acquisizione, sono sfide centrali nel campo del CS.

2.3 Cenni storici del Compressive sensing

Sebbene questa idea abbia recentemente attirato una significativa attenzione nella comunità dell'elaborazione del segnale, ci sono stati suggerimenti in questa direzione che risalgono al diciottesimo secolo. Nel 1795, Prony propose un algoritmo per la stima dei parametri associati a un piccolo numero di esponenziali complessi campionati nella resistenza del rumore [40]. Il successivo salto teorico avvenne all'inizio del 1900, quando Carathéodory mostrò che una combinazione lineare positiva di qualsiasi punto k è determinata in modo univoco dal suo valore in $t = 0$ e in qualsiasi altro punto $2k$ nel tempo [17, 18]. Questo risultato consente di rappresentare molti meno campioni rispetto al numero di campioni di frequenza di Nyquist, quando k è

piccolo e l'intervallo delle frequenze possibili è ampio. Negli anni '90, questo lavoro è stato generalizzato da George, Gorodnitsky e Rao, che hanno studiato la sparsità nella diagnostica per immagini biomagnetiche e in altri contesti [28, 30, 29, 41]. Contemporaneamente, Bresler, Feng e Venkataramani hanno proposto uno schema di campionamento per l'acquisizione di determinate classi di segnali costituiti da componenti k con larghezza di banda diversa da zero (al contrario di sinusoidi puri) sotto restrizioni sui possibili supporti spettrali, sebbene il recupero esatto non sia stato in generale garantito [7, 25, 8, 49]. All'inizio degli anni 2000 Blu, Marziliano e Vetterli hanno sviluppato metodi di campionamento per determinate classi di segnali parametrici che sono governati solo da parametri k , dimostrando che questi segnali possono essere campionati e recuperati da soli $2k$ campioni [50].

Un problema correlato si concentra sul recupero di un segnale dall'osservazione parziale della sua trasformata di Fourier. Beurling ha proposto un metodo per estrapolare queste osservazioni al fine di determinare l'intera trasformata di Fourier [5]. Si può dimostrare che se il segnale è costituito da un numero finito di impulsi, l'approccio di Beurling recupererà correttamente l'intera trasformata di Fourier da qualsiasi pezzo sufficientemente grande della sua trasformata di Fourier. Il suo approccio, per trovare il segnale con la più piccola norma l_1 tra tutti i segnali concordanti con le misurazioni di Fourier acquisite, ha una notevole somiglianza con alcuni degli algoritmi utilizzati in CS. Più recentemente, Candès, Romberg, Tao e Donoho hanno mostrato che un segnale con una rappresentazione sparsa può essere recuperato *esattamente* da un piccolo insieme di misurazioni lineari e non adattative. Questo risultato suggerisce che potrebbe essere possibile rilevare segnali sparsi prendendo molte meno misurazioni, da cui il nome di *compressive sensing* (rilevamento compresso) [9, 11, 12, 13, 15, 22].

2.4 Ambiti d'uso del Compressive Sensing

Il CS ha già avuto un notevole impatto su diverse applicazioni. Un esempio è l'imaging biomedico (diagnostica per immagini), dove ha consentito di accelerare di sette volte la risonanza magnetica pediatrica preservando la qualità diagnostica [48]. Inoltre, l'ampia applicabilità di questo framework ha ispirato la ricerca, che estende l'ambito del CS, proponendo implementazioni pratiche per numerose applicazioni, inclusi i sistemi di campionamento sub-Nyquist, compressive imaging architetturale e reti di sensori compressive.

2.5 Quadro matematico

In questo paragrafo definiremo in linea generale i concetti matematici alla base del CS.

2.5.1 Concetti preliminari e notazioni

Per chiarezza di presentazione, introdurremo nozioni e concetti fondamentali nell'ambito del CS, quali spazi vettoriali, notazioni di base, proprietà di isometria limitata (RIP) e proprietà di coerenza, per poi proseguire con i fondamenti matematico-teorici del CS.

2.5.1.1 Spazi Vettoriali

Per gran parte della sua storia, l'elaborazione dei segnali si è concentrata sui segnali prodotti dai sistemi fisici. Molti sistemi naturali e artificiali possono essere modellati come lineari. Pertanto, è naturale considerare i modelli di segnale che completano questo tipo di struttura lineare. Questa nozione è stata incorporata nella moderna elaborazione del segnale, modellando segnali come vettori che vivono in uno spazio vettoriale appropriato. Questo prende la struttura lineare che spesso desideriamo, vale a dire che se sommiamo due segnali insieme otteniamo un nuovo segnale fisicamente significativo. Inoltre, gli spazi vettoriali ci consentono di applicare intuizioni e strumenti dalla geometria in \mathbb{R}^3 , come lunghezze e angoli, per descrivere e confrontare segnali di interesse.

Tratteremo i segnali come funzioni a valore reale con domini che sono continui o discreti e infiniti o finiti. Nel seguito verrà presa in considerazione solo una breve rassegna di alcuni dei concetti chiave negli spazi vettoriali che saranno richiesti nello sviluppo della teoria del CS. In genere ci occuperemo di spazi vettoriali normalizzati, ovvero dotati di una norma. Nel caso di un dominio finito discreto, possiamo vedere i segnali come vettori in uno spazio euclideo N-dimensionale, indicato con \mathbb{R}^N . Quando avremo a che fare con vettori in \mathbb{R}^N , useremo frequentemente le norme l_p , che sono definite per $p \in [1, \infty]$ come

$$\|x\|_p = \begin{cases} \sqrt[p]{\sum_{i=1}^N |x_i|^p}, & \text{per } p \in [1, \infty); \\ \max_{1 \leq i \leq N} |x_i|, & \text{per } p = \infty; \end{cases} \quad i = 1, 2, 3, \dots, N \quad (1)$$

Nello spazio euclideo possiamo anche considerare il *prodotto interno* classico in \mathbb{R}^N , che denotiamo con

$$\langle x, z \rangle = z^T x = \sum_{i=1}^N x_i z_i \quad (2)$$

Questo *prodotto interno* è riconducibile alla norma l_2 : $\|x\|_2 = \sqrt{\langle x, x \rangle}$.

Definizione - Due vettori x e z della stessa lunghezza sono detti *ortogonali* se $\langle x, z \rangle = z^T x = 0$. Inoltre, se ogni vettore ha lunghezza unitaria, cioè $\|x\|_2 = \|z\|_2 = 1$, allora i due vettori sono detti *ortonormali*.

In alcuni contesti è utile estendere la nozione delle norme l_p al caso in cui sia $p < 1$. In questo caso, la norma definita in (1) non riesce a soddisfare la disuguaglianza triangolare, quindi in realtà è una *quasi-norma*. Faremo uso della notazione $\text{supp}(x) = \{i : x_i \neq 0\}$ per indicare il supporto di x e $|\text{supp}(x)|$ per la sua cardinalità. Si noti che $\|\cdot\|_0$ non è nemmeno una quasi-norma, ma si può facilmente dimostrare, infatti che

$$\lim_{p \rightarrow 0} \|x\|_p = |\text{supp}(x)| \quad (3)$$

giustificando questa scelta di notazione. Le (quasi-)norme l_p hanno particolari proprietà diverse per valori diversi di p . Per illustrare ciò, nella Figura 2.2 mostriamo la sfera unitaria, ovvero $\{x : \|x\|_p = 1\}$, indotta da ciascuna di queste norme in \mathbb{R}^2 . Si noti che per $p < 1$ la sfera unitaria corrispondente non è convessa (riflettendo la violazione della quasi-norma della disuguaglianza del triangolo). In genere le norme

vengono utilizzate come misura dell'intensità di un segnale o della dimensione di un errore. Ad esempio, supponiamo che ci venga dato un segnale $x \in \mathbb{R}^2$ e desidera-

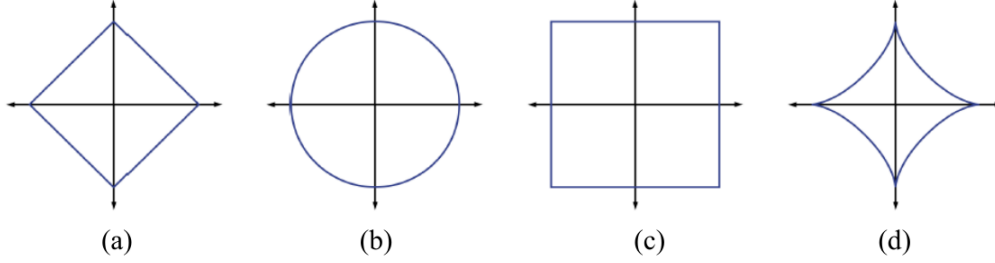


Figura 2.2: Sfera unitaria in \mathbb{R}^2 per la norma l_p con $p = 1, 2, \infty$, e per la quasi-norma l_p con $p = 0.5$. (a) Sfera unitaria con norma l_1 . (b) Sfera unitaria con norma l_2 . (c) Sfera unitaria con norma l_∞ . (d) Sfera unitaria con quasi-norma l_p .

mo approssimarlo usando un punto in uno spazio affine monodimensionale A . Se misuriamo l'errore di approssimazione usando una norma l_p , allora il nostro compito è trovare la $\hat{x} \in A$ che minimizza $\|x - \hat{x}\|_p$. La scelta di p avrà un effetto significativo sulle proprietà dell'errore di approssimazione risultante, come illustrato nella Figura 2.3. Per calcolare il punto più vicino a x in A usando ciascuna norma l_p , possiamo immaginare di far crescere una sfera l_p centrata su x fino a quando essa si interseca con A . Questo sarà il punto $\hat{x} \in A$ che è il più vicino a x nella corrispondente norma l_p .

Va notato che una p maggiore tenda a distribuire l'errore in modo più uniforme tra i due coefficienti, mentre una p più piccola porta a un errore che è distribuito in modo più disomogeneo e tende ad essere scarso. Questa intuizione gioca un ruolo importante nello sviluppo della teoria CS.

Definizione - Un segnale x è K -sparso quando ha al massimo K elementi non nulli, ovvero $\|x\|_0 \leq K$. Denotiamo con $\Sigma_K = \{x : \|x\|_0 \leq K\}$ l'insieme di tutti i segnali K -sparsi.

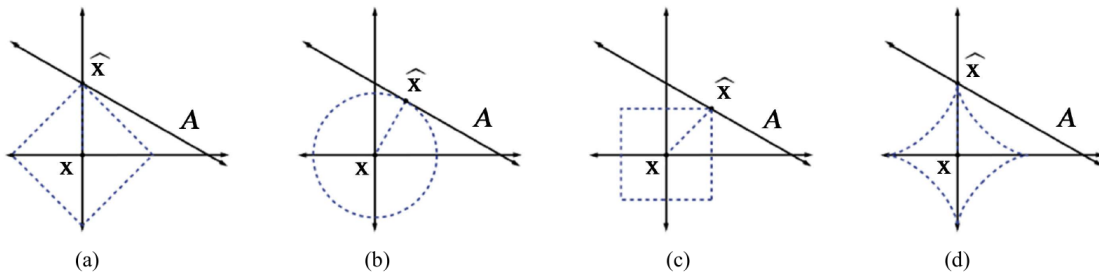


Figura 2.3: Migliore approssimazione di un punto $x \in \mathbb{R}^2$ con un punto \hat{x} in uno spazio affine monodimensionale usando la norma l_p con $p = 1, 2, \infty$, e per la quasi-norma l_p con $p = 0.5$. (a) Approssimazione in norma l_1 . (b) Approssimazione in norma l_2 . (c) Approssimazione in norma l_∞ . (d) Approssimazione in quasi-norma l_p .

2.5.1.2 La proprietà di isometria limitata (*Restricted Isometry Property - RIP*)

In algebra lineare una matrice ortogonale è una matrice quadrata a coefficienti reali le cui colonne e righe sono vettori ortonormali, cioè $A^T A = A A^T = I$, dove I è la matrice identica. Questo porta alla caratterizzazione equivalente: una matrice A è ortogonale se la sua trasposta è uguale alla sua inversa, ovvero $A^T = A^{-1}$. Il determinante di qualsiasi matrice ortogonale è perciò $+1$ o -1 . Come trasformazione lineare, una matrice ortogonale è una matrice che rappresenta una isometria dello spazio euclideo, poiché conserva la distanza come la rotazione o la riflessione.

La RIP caratterizza matrici quasi ortonormali, almeno quando operano su vettori sparsi. Il concetto è stato introdotto da Emmanuel Candès e Terence Tao [14] ed è usato per dimostrare molti teoremi nel campo del CS. Non sono note matrici delimitate di grandi dimensioni con costanti isometriche limitate (e calcolare queste costanti è NP-difficile), ma è stato dimostrato che molte matrici casuali rimangono limitate. In particolare, è stato dimostrato che con probabilità esponenzialmente elevata le matrici casuali Gaussiane, di Bernoulli e quelle parziali di Fourier soddisfano la RIP con un numero di misurazioni quasi lineari a livello di sparsità. Gli attuali limiti superiori più piccoli per le matrici rettangolari di grandi dimensioni sono quelli delle matrici Gaussiane.

Definizione - Sia A una matrice $m \times n$ e sia $1 \leq K \leq n$ un intero. Supponiamo che esista una costante δ_K tale che per ogni vettore $y \in \sum_K$,

$$(1 - \delta_K) \|y\|_2^2 \leq \|Ay\|_2^2 \leq (1 + \delta_K) \|y\|_2^2, \quad (4)$$

di ordine K . Allora diremo che la matrice A soddisfa la proprietà di isometria limitata (RIP) con costante di isometria limitata δ_K .

2.5.1.3 Il concetto di coerenza

Il concetto di coerenza è stato introdotto in un quadro leggermente più specifico in [22], e da allora è stato ampiamente utilizzato nel campo delle rappresentazioni sparse dei segnali. In particolare, viene utilizzato come misura della capacità di algoritmi sub-ottimali come il *matching pursuit* e il *basis pursuit* per identificare correttamente la rappresentazione reale di un segnale sparso. Le attuali assunzioni nel campo del CS e del recupero di segnali sparsi impongono che la matrice di misurazione abbia colonne non correlate. Formalmente si definisce la coerenza o la coerenza reciproca di una matrice A come il valore assoluto massimo delle correlazioni incrociate tra le colonne di A .

Definizione - La coerenza di una matrice A , $\mu(A)$, è il più grande prodotto interno in valore assoluto tra due colonne a_i, a_j di A

$$\mu(A) = \max_{1 \leq i < j} \frac{\| \langle a_i, a_j \rangle \|}{\|a_i\|_2 \|a_j\|_2}$$

Nello specifico, se A è normalizzata in modo tale che $a_i^T a_i = 1$ allora la coerenza reciproca di A viene quindi definita come

$$\mu(A) = \max_{1 \leq i \neq j \leq m} |a_i^T a_j| \quad (5)$$

Diremo che un campione è *incoerente* quando $\mu(A)$ è molto piccolo. Risultati standard richiedono quindi che la matrice di misurazione soddisfi rigorosamente la proprietà di incoerenza, come impone anche la RIP. Se il campione è altamente coerente, allora anche la matrice risultante sarà coerente in generale. La coerenza è in qualche modo una proprietà naturale nel quadro del CS, poiché se due colonne sono strettamente correlate, sarà impossibile in generale distinguere se l'energia nel segnale proviene dall'una o dall'altra.

2.6 Fondamenti del Compressive Sensing

Come detto precedentemente, il campionamento della frequenza di Nyquist descrive completamente un segnale sfruttandone la larghezza di banda. Il nostro obiettivo è ridurre il numero di misurazioni necessarie per descrivere completamente un segnale sfruttando la sua comprimibilità. Il problema sorge quando le nostre misurazioni non sono campioni puntuali, ma funzionali lineari più generali del segnale.

Consideriamo un segnale x reale, finito, monodimensionale e discreto nel tempo, che possiamo vedere come un vettore colonna $N \times 1$ in \mathbb{R}^N con elementi x_n , $n = 1, 2, \dots, N$. Consideriamo un'immagine o dati di dimensione superiore e vettorializziamoli in un lungo vettore unidimensionale. Qualsiasi segnale in \mathbb{R}^N può essere rappresentato in modo univoco come combinazione lineare dei vettori della base $\{\psi_i\}_{i=1}^N$. I vettori ψ_i della base sono linearmente indipendenti e ogni segnale $x \in \mathbb{R}^N$ può essere espresso come

$$x = \sum_{i=1}^N \psi_i s_i = \Psi s \quad (6)$$

con

$$\Psi := [\psi_1 | \psi_2 | \dots | \psi_N]$$

matrice di base $N \times N$ ottenuta impilando i vettori $\{\psi_i\}$ come colonne e $s = (s_1, \dots, s_N)^T \in \mathbb{R}^N$ vettore dei coefficienti.

Per semplicità assumeremo che la base sia ortogonale, ovvero :

$$\langle \psi_i, \psi_j \rangle = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$$

Una base ortogonale ha il vantaggio che i coefficienti s possano essere calcolati come $s_i = \langle x, \psi_i \rangle$ ovvero $s = \Psi^T x$ con Ψ^T matrice trasposta di Ψ .

Chiaramente, x e s sono rappresentazioni equivalenti dello stesso segnale, ma con domini diversi.

Ci concentreremo sui segnali che hanno una rappresentazione sparsa, ovvero quelli per cui il vettore dei coefficienti s ha $K \ll N$ elementi non nulli. Un segnale x si dice *K-sparso* se il vettore s ha $K \ll N$ elementi non nulli. L'insieme degli indici degli elementi di s non nulli è detto *supporto di s* e si denota con I_K . Ψ prende il nome di *matrice di sparsità*.

La sparsità è dovuta al fatto che molti segnali naturali e artificiali sono comprimibili, nel senso che esiste una base Ψ in cui la rappresentazione $x = \Psi s$ ha solo pochi coefficienti di grandi dimensioni e molti coefficienti piccoli.

I segnali comprimibili sono ben approssimati dalle rappresentazioni K -sparse. Questo concetto è alla base del *transform coding*: le immagini tendono a essere comprimibili attraverso la *discrete cosine transform* (DCT) e le *wavelet* che sono alla base delle tecniche di compressione JPEG e JPEG-2000. I segnali audio e molti segnali di comunicazione sono invece comprimibili in una base di Fourier localizzata. La procedura consiste nell'acquisire il campione completo N del segnale x , nel calcolare l'insieme completo dei coefficienti di trasformazione $\{s_i\}$ tramite $s = \Psi^T x$, nell'individuare il coefficiente K più grande e scartare i coefficienti $(N - K)$ più piccoli, ed infine elaborare i valori K e le posizioni dei coefficienti più grandi. (In pratica, convertiamo anche i valori e le posizioni in bit.)

Sfortunatamente, la procedura *campiona-poi-comprimi* soffre di tre inefficienze intrinseche: in primo luogo, dobbiamo iniziare con un numero potenzialmente elevato di campioni N anche se il K desiderato è piccolo. In secondo luogo, il codificatore deve calcolare tutti N i coefficienti di trasformazione $\{s_i\}$, anche se li eliminerà tutti tranne K . In terzo luogo, l'encoder deve affrontare il sovraccarico della codifica delle posizioni dei coefficienti di grandi dimensioni.

Studieremo quindi un approccio di acquisizione dei dati più generale che condensa il segnale direttamente in una rappresentazione compressa senza passare attraverso lo stadio intermedio del prelievo di N campioni. Infatti, si consideri un generico *processo di misurazione lineare* che calcola $M < N$ prodotti interni tra x e una raccolta di vettori $\{\phi_j\}_{j=1}^M$ come $y_j = \langle x, \phi_j \rangle$. Impilando le misure y_j nel vettore y di dimensioni $M \times 1$ e i vettori di misura ϕ_j^T come righe in una matrice $M \times N$, detta Φ , e sostituendo in (6), possiamo scrivere

$$y = \Phi x = \Phi \Psi s = \Theta s. \quad (7)$$

dove Θ è una matrice $M \times N$. Una rappresentazione grafica della equazione (7) è rappresentata in Figura 2.4.

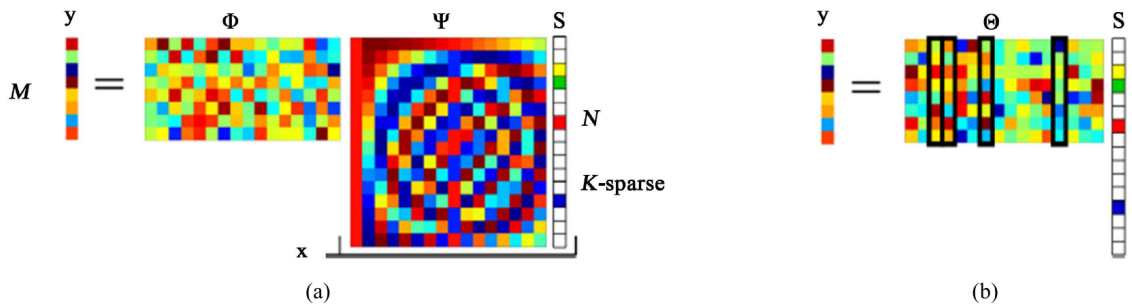


Figura 2.4: (a) Processo di misurazione CS con matrice di misurazione (gaussiana casuale) Θ e matrice Φ derivata dal *DCT*. Il vettore del coefficiente s è sparso con $K = 4$; (b) Processo di misurazione in termini dell'equazione (7) con le quattro colonne diverse da zero evidenziate. Il vettore di misurazione y è una combinazione lineare di queste quattro colonne.

Si noti che il processo di misurazione non è adattivo; cioè, Φ non dipende in alcun modo dal segnale x .

Il nostro obiettivo adesso sarà quello di progettare una matrice di misurazione Φ e un algoritmo di ricostruzione per segnali K -sparsi e comprimibili, che richiedono solo $M \approx K$ misurazioni o poco più. Il nostro approccio si basa sulla teoria CS introdotta precedentemente.

2.7 Matrice di rilevamento (*Sensing matrix*)

Nel primo passo, progettiamo una matrice di rilevamento stabile Φ che assicuri che le informazioni salienti in qualsiasi segnale K -sparso o comprimibile, non siano danneggiate dalla riduzione della dimensionalità da $x \in \mathbb{R}^N$ a $y \in \mathbb{R}^M$.

Innanzitutto, progettiamo la parte riguardante la misurazione del sistema di acquisizione dei dati, che si basa sulla matrice Φ . Miriamo a fare M misurazioni (nella fattispecie il vettore y) da cui possiamo ricostruire stabilmente il segnale x di lunghezza N , o equivalentemente il suo vettore dei coefficienti sparsi s nella matrice base Ψ . Chiaramente la ricostruzione non sarà possibile se il processo di misurazione danneggia le informazioni in x . Sfortunatamente, questo è il caso generico: poiché il processo di misurazione è lineare e definito in termini di matrici Φ e Ψ , risolvere $y = \Phi x = \Phi \Psi s = \Theta s$ è proprio un problema di algebra lineare, dove essendo $M < N$, ci sono meno equazioni che incognite, rendendo la soluzione mal posta in generale. Tuttavia, la K -sparsità di s viene in nostro aiuto. In questo caso il vettore di misura y è solo una combinazione lineare delle colonne K di Θ il cui corrispondente $s_i \neq 0$. Quindi, se sapessimo a priori quali K elementi di s fossero diversi da zero, allora potremmo formare un sistema $M \times K$ di equazioni lineari da risolvere per questi valori diversi da zero, dove ora il numero di equazioni M è uguale o superiore al numero di incognite K . Una condizione necessaria e sufficiente per garantire che questo sistema $M \times K$ sia ben condizionato - e quindi presenta un inverso stabile - è che per ogni vettore v che condivide gli stessi K elementi diversi da zero di s , abbiamo

$$1 - \delta_K \leq \frac{\|\Theta v\|_2}{\|v\|_2} \leq 1 + \delta_K \quad (8)$$

per un generico $0 < \delta_K < 1$. In altre parole, la matrice Θ deve preservare la lunghezza di questi determinati vettori K -sparsi. Questa è la cosiddetta *restricted isometry property* (RIP, *proprietà isometrica limitata*). Un approccio alternativo alla stabilità è quello di garantire che la matrice di rilevamento Φ sia *incoerente* con la matrice base Ψ , nel senso che i vettori $\{\phi_j\}$ non possono rappresentare in modo sparso i vettori $\{\phi_i\}$ e viceversa. L'esempio classico è dato dal principio di incertezza di Fourier il quale produce immediatamente l'incoerenza. Quindi, data una base sparsificante Ψ , come possiamo costruire una matrice di misurazione Φ tale che $\Theta = \Phi \Psi$ soddisfi la RIP? Sfortunatamente, la semplice verifica che una matrice Θ garantisca la (8) è alquanto complessa; infatti occorre verificare la (8) per ciascuna delle $\binom{N}{K}$ combinazioni.

Nel CS, evitiamo questo problema selezionando Φ come matrice casuale. Ad esempio, disegniamo gli elementi della matrice $\phi_{i,j}$, come variabili casuali indipendenti e identicamente distribuite (iid) da una media zero, e da una varianza $1/N$ della densità Gaussiana. Quindi, le misure y sono semplicemente M diverse combinazioni lineari ponderate casualmente degli elementi di x . Una Φ Gaussiana ha due interessanti e potenti proprietà. In primo luogo, Φ è incoerente, con alte probabilità, con la base $\Psi = I$. In secondo luogo, grazie alle proprietà della distribuzione Gaussiana iid che genera Φ , anche la matrice $\Theta = \Phi \Psi$ è iid Gaussiana al di là della scelta della matrice di base (ortogonale) Ψ . Pertanto, le misurazioni Gaussiani casuali Φ sono *universali* nel senso che $\Theta = \Phi \Psi$ rispetta la RIP con alta probabilità per ogni possibile Ψ . Tra le altre, si può anche dimostrare che le matrici di Rademacher,

definite a partire da una distribuzione di probabilità in cui una variabile casuale ha il 50% di probabilità di assumere valore ± 1 , hanno proprietà RIP [4].

2.8 Algoritmi di ricostruzione

Gli algoritmi di ricostruzione veloci ed efficienti sono le chiavi per incorporare il CS nelle applicazioni del mondo reale, quindi lo sviluppo di tali algoritmi è stata la preoccupazione principale nella comunità CS. Diverse classi di algoritmi di ricostruzione sono state proposte in letteratura. Tuttavia, gli algoritmi ben noti rientrano nell'ambito di due importanti approcci algoritmici, ovvero la *convex optimization* (ottimizzazione convessa) e gli algoritmi *greedy* (ricerca soluzione più ghiotta, avida).

2.8.1 Convex Optimization - ottimizzazione convessa

Un approccio semplice per recuperare un vettore sparso x è risolvere il problema di minimizzazione l_0 :

$$\hat{x} = \arg \min \|x\|_0 \text{ soggetto a } y = \Phi x \quad (9)$$

Tuttavia, il problema di minimizzazione l_0 è considerato un NP-difficile per una matrice su larga scala [36], cioè risolvere l'equazione (9) per una qualsiasi matrice di grandi dimensioni Φ è costoso dal punto di vista computazionale. Di conseguenza, per superare l'inconveniente di l_0 , si è considerati la formazione convessa dell'equazione (9), considerando la minimizzazione l_1 .

La minimizzazione l_1 , noto come *Base Pursuit* (BP) [20], considera la seguente soluzione:

$$\hat{x} = \arg \min \|x\|_1 \text{ soggetta a } y = \Phi x \quad (10)$$

In particolare, un modello di acquisizione rumoroso si adatta prendendo in considerazione la soluzione della *basis pursuit denoising* (BPDN)[10]

$$\hat{x} = \arg \min \|x\|_1 \text{ soggetta a } \|y - \Phi x\|_2 \leq \epsilon \quad (11)$$

dove ϵ rappresenta il limite superiore del livello di rumore.

Inoltre, se non si conoscono i livelli di rumore, è possibile esplorare l'approccio *Least Absolute Shrinkage and Selection Operator* (LASSO, *operatore di restringimento e selezione minimo assoluto*) [16]

$$\min_{\hat{x}} \frac{1}{2} \|\Phi \hat{x} - y\|_2 + \gamma \|\hat{x}\|_1 \quad (12)$$

dove $\gamma > 0$ rappresenta un parametro di ottimizzazione per diversi livelli di sparsità.

2.8.2 Algoritmi Greedy

Gli algoritmi greedy sono stati ampiamente sfruttati nelle applicazioni CS grazie al loro framework relativamente semplice che fornisce una ricostruzione veloce con un basso costo di implementazione. Questi metodi migliorano iterativamente l'approssimazione del segnale facendo scelte localmente ottimali. Gli algoritmi greedy consistono in due fasi principali, selezione degli elementi e aggiornamento dei coefficienti. Questi metodi sono generalmente inizializzati con un $r^{[0]} = y$ residuo, la

stima del segnale $\hat{x}^{[0]} = 0$ e l'insieme di supporto T vuoto ($T = \emptyset$). Ad ogni iterazione j , gli elementi singoli o multipli della matrice di rilevamento vengono aggiunti all'insieme di supporto, aggiornando la stima del segnale x ($x^{[j]} = \Phi^T y$), riducendo così l'errore di osservazione residuo r ($r^{[j]} = y - \Phi^T x^{[j]}$). Gli algoritmi si fermano quando la norma residua è inferiore a una soglia predefinita.

Attualmente, gli algoritmi greedy ben consolidati includono il *gradient pursuit* [26], il *matching pursuit* (MP) [35, 33] e l'*orthogonal matching pursuit* (OMP) [47]. L'OMP offre un recupero rapido rispetto agli approcci di *convex optimization*, ma soffre di una cattiva qualità di recupero per segnali con un basso grado di scarsità. Sono state proposte diverse versioni migliorate dell'OMP, come il *compressive sampling matching pursuit* (CoSaMP) [37], il *subspace pursuit* (SP) [21], il *Regularized OMP* [38], lo *Stagewise OMP* [23] e l'*orthogonal multiple matching pursuit* [34].

Le prestazioni di questi algoritmi di recupero dipendono dall'applicazione in oggetto e non viene stabilita alcuna metrica di recupero univoca per determinare la migliore tecnica di recupero per tutti gli scenari. Pertanto, in [6, 19, 27, 1] è possibile trovare un confronto delle prestazioni sia teorico che sperimentale tra le diverse classi di algoritmi di recupero CS. La tabella seguente elenca i dettagli relativi alla complessità e al numero minimo di misurazioni richieste per ciascun algoritmo per raggiungere il limite di recupero ottimale.

Algoritmo	Numero minimo di misurazioni	Complessità
BP	$k \log N$	$O(M^2 N^{1.5})$
OMP	$k \log N$	$O(kMN)$
CoSaMP	$k \log N$	$O(MN)$
SP	$k \log(\frac{N}{k})$	$O(MN \log k)$
Stagewise OMP	$N \log N$	$O(k \log N)$

2.9 Compressive Sensing Distribuito (*Distributed Compressive Sensing*)

La teoria CS convenzionale esplora solo le strutture centrate sul segnale (sparsità e compressibilità) in un singolo sensore. Tuttavia, se lo scenario di interesse presenta un'architettura in cui i dati non vengono raccolti solo da un singolo sensore ma piuttosto con una rete distribuita di sensori, lo sfruttamento della collaborazione di questi sensori porterebbe a un ulteriore aumento delle informazioni raccolte e ad un conseguente perfezionamento della progettazione applicativa. Tale scenario si presenta in diverse applicazioni di monitoraggio *Smart Grid* (SG), in cui viene distribuita una rete di sensori per misurare diverse quantità fisiche/fisiologiche scalari. Se i sensori sono distribuiti densamente, le loro misurazioni sono correlate e probabilmente condivideranno determinate strutture. L'architettura multisensore basata su CS con sensori J può presentare due scenari. Nel primo, i segnali acquisiti sono indipendenti, quindi il problema di ricostruzione equivale a eseguire un'operazione di ricostruzione individuale J per ciascun sensore, e nessuna collaborazione tra i nodi di rilevamento può essere analizzata per il motivo che ciascun sensore non contiene informazioni sui dati acquisiti dagli altri sensori. Nel secondo scenario, che emerge frequentemente nella maggior parte delle applicazioni SG, i segnali acquisiti da tutti

i sensori sono altamente correlati, presentando una struttura di sparsità tra segnali. Di conseguenza, per ricostruire l'intero insieme di segnali, è possibile sfruttare non solo la singola sparsità di ciascuna misurazione dei sensori, ma anche la dipendenza tra i campioni acquisiti da sensori diversi che allo stesso tempo dovrebbero essere analizzate per ridurre il numero richiesto di misurazioni, al fine di assemblare le informazioni più pertinenti relative al fenomeno rilevato. Questo processo, che chiamiamo *recupero delle misure congiunte*, è la motivazione per introdurre il concetto di *distributed compressive sensing* (DCS)[3].

Il DCS presenta un nuovo paradigma di codifica distribuito che sfrutta le strutture di correlazione sia intra che tra segnali. Negli scenari DCS, un certo numero di sensori acquisisce segnali sparsi che sono correlati tra loro. Ogni sensore individualmente acquisisce un segnale compresso prendendo proiezioni casuali e quindi lo trasmette al nodo di fusione. Ancora più importante, il DCS non richiede collaborazione tra i sensori durante la fase di acquisizione, tuttavia il processo di recupero può sfruttare la correlazione dei segnali per ricostruire tutti i segnali contemporaneamente. Gli algoritmi di recupero associati a DCS sono derivati dalle loro controparti CS, come il BPDN-multicanale [45], l'OMP simultaneo (*simultaneous OMP*, SOMP) [46] e il distributed compressive sensing OMP (DCS-SOMP)[3] che è una forma generale di SOMP.

Capitolo 3

Applicazioni Compressive Sensing per le Smart Grid

Abbiamo analizzato cosa sono le Smart Grid, le loro funzionalità e le sfide che pongono. Inoltre, abbiamo studiato cos'è il Compressive Sensing ed evidenziato come possa costituire una soluzione alle problematiche sollevate dal rilevamento e il recupero sparso delle informazioni sulle reti SG. In questo capitolo studieremo, implementeremo e analizzeremo una delle principali tecniche algoritmiche applicabili al contesto di studio. Per fare ciò dovremo avvalerci di un modello di una rete elettrica cui fare riferimento per il nostro studio.

3.1 Modello di rete e grafico corrispondente

In questa sezione sono descritte la rappresentazione grafica di una rete elettrica (*power network*, PN), le sue corrispondenti matrici e la sua connessione con il modello di flusso di corrente continua. Inoltre, discutiamo un'ipotesi di distribuzione gaussiana casuale per i parametri di sistema (potenza e tensione del bus) in un grande SG, con una considerevole quantità di fonti di energia rinnovabile.

3.1.1 Grafico di una PN

Rappresentiamo una PN come un grafico $G(S_N, S_E)$, costituito da un insieme di N nodi $S_N = \{1, \dots, N\}$, in cui ciascun nodo i rappresenta un bus arbitrario della SG, e un insieme di L linee di trasmissione $S_E = \{l_{i,j} : i, j \in S_N\}$, che collegano determinati nodi e formano la struttura generale o la topologia della griglia. Seguendo gli approcci in [31], e quasi tutti gli esempi in [54], assumiamo che i dati dei parametri dei nodi (come gli angoli di potenza e di fase) siano ottenuti da una rete di sensori. L'obiettivo principale è quello di identificare la struttura o la topologia di una PN in tempo quasi reale utilizzando una piccola serie di dati, senza alcuna conoscenza a priori sulla struttura della rete.

3.1.2 Modello di flusso a corrente continua e grafico corrispondente di una PN

Il modello di flusso a corrente alternata, noto anche come modello di flusso di carico alternato, è una tecnica numerica in ingegneria energetica, applicata a un sistema

di alimentazione al fine di analizzare il comportamento di diverse forme di corrente alternata durante il normale funzionamento in regime stazionario. In un tale modello due distinti insiemi di equazioni non lineari (chiamate equazioni di bilanciamento della potenza) indicano la relazione tra potenza attiva e reattiva e la grandezza della tensione e l'angolo di fase come segue [53] :

$$P_{i,j} = g_{i,j}V_i^2 - g_{i,j}V_iV_j \cos(\theta_i - \theta_j) + b_{i,j}V_iV_j \sin(\theta_i - \theta_j) \quad (13)$$

$$Q_{i,j} = b_{i,j}V_i^2 - b_{i,j}V_iV_j \cos(\theta_i - \theta_j) + g_{i,j}V_iV_j \sin(\theta_i - \theta_j) \quad (14)$$

Qui $P_{i,j}$ e $Q_{i,j}$ sono le potenze attive e reattive iniettati, rispettivamente, dal nodo j al nodo i , V_i e θ_i rappresentano rispettivamente l'ampiezza e la fase della tensione sul nodo i , $g_{i,j}$ è la parte reale dell'ammissione della linea $l_{i,j}$ (o della conduttanza) e $b_{i,j}$ è la parte immaginaria dell'ammissione della linea $l_{i,j}$ (o la suscettività). È stato dimostrato che quando il sistema è stabile, in una normale condizione di regime, le differenze di angolo di fase sono piccole, il che significa che $\sin(\theta_i - \theta_j) \approx \theta_i - \theta_j$. Usando questa semplificazione e il metodo del flusso di corrente continua, la potenza iniettata in un distinto bus i segue la legge della sovrapposizione :

$$P_i = \sum_j P_{i,j} = \sum_{j \in N_j} b_{i,j}(\theta_i - \theta_j) \quad (15)$$

$$Q_i = \sum_j Q_{i,j} = \sum_{j \in N_j} b_{i,j}(V_i - V_j) \quad (16)$$

Qui N_i è l'insieme dei bus vicini collegati direttamente al bus i . È utile riscrivere queste sommatorie in un formato vettoriale a matrice, dove abbiamo:

$$p = B\theta, \quad p \in R^N \quad (17)$$

$$q = Bv, \quad q \in R^N \quad (18)$$

In questa notazione i valori di ampiezza di tensione e angolo di tensione del fasore di tutti i nodi sono raccolti rispettivamente in due vettori $v, \theta \in R^N$. Inoltre, i valori di potenza attiva e reattiva di tutti i nodi sono memorizzati rispettivamente nei vettori p e q . La matrice $B \in R^{N \times N}$ è chiamata matrice di ammissione nodale (*nodal-admittance matrix*) la quale descrive una rete di alimentazione di N bus, e i suoi elementi possono essere rappresentati nel seguente formato:

$$B_{i,j} = \begin{cases} -b_{i,j}, & \text{se } l_{i,j} \in S_E \\ \sum_{j \in N_j} b_{i,j}, & \text{se } i = j \\ 0, & \text{altrimenti} \end{cases} \quad (19)$$

La matrice di ammissione nodale dello standard IEEE 118 Bus è mostrata in Figura 3.4, mentre in Figura 3.1 è riportato il relativo grafo.

Questa matrice ha una struttura sparsa. È stato menzionato in letteratura che la matrice di ammissione nodale di una PN può essere interpretata come una versione ponderata della matrice Laplaciana del corrispondente grafico $G(S_N, S_E)$ [53] e [51].

Definizione - Si dice **matrice di adiacenza** normalizzata la matrice così definita:

$$\mathcal{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$$

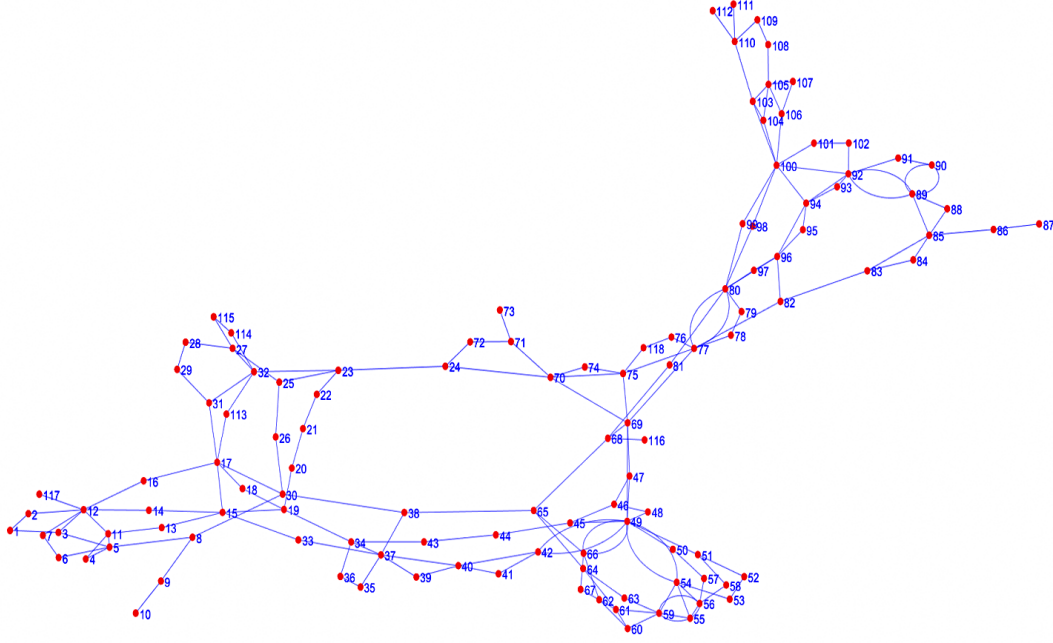


Figura 3.1: Rappresentazione a grafo di una rete Standard IEEE a 188 Bus.

dove A è la matrice di adiacenza di G e $D = \text{diag}(d)$ con $d(i)$ grado del nodo i . Per un grafo G , con vertici non isolati, definiamo così la **matrice dei gradi**:

$$D^{-\frac{1}{2}} = \begin{pmatrix} \frac{1}{\sqrt{d(1)}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{d(2)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sqrt{d(n)}} \end{pmatrix}$$

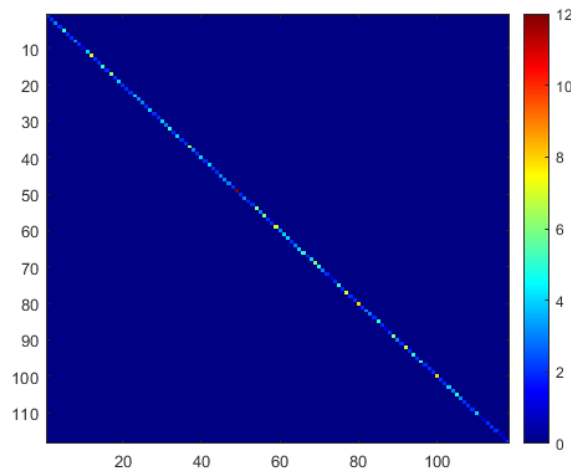


Figura 3.2: Rappresentazione della matrice dei gradi dello Standard IEEE 118 Bus

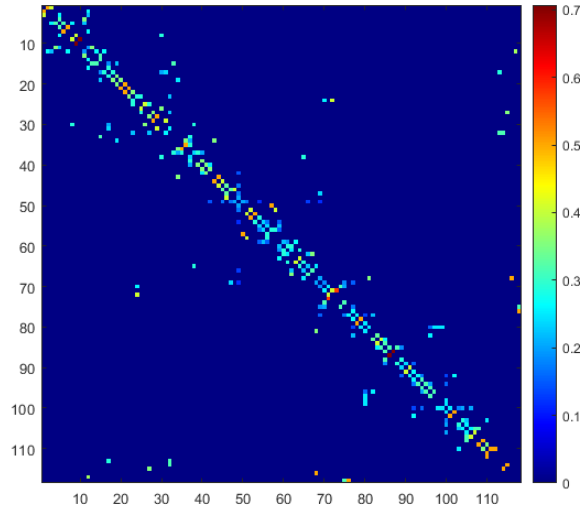


Figura 3.3: Rappresentazione della matrice di adiacenza dello Standard IEEE 118 Bus

In Figura 3.2 - Figura 3.3 sono riportate le matrici dei gradi e di adiacenza dello Standard IEEE 118 Bus.

Definizione - Si dice **matrice Laplaciana** normalizzata, la matrice così definita:

$$\mathcal{L} = I - \mathcal{A}.$$

Da notare che $\mathcal{L} = I - A = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}} = D^{-\frac{1}{2}}L_G D^{-\frac{1}{2}}$ con L_G matrice non-normalizzata Laplaciana.

Ogni elemento $l_{i,j}$ di \mathcal{L} potremmo quindi definirlo come segue:

$$l_{i,j} = \begin{cases} d(n_i), & \text{se } i = j \\ -1, & \text{se } i \neq j, \text{ e } n_i \text{ è adiacente a } n_j \\ 0, & \text{altrimenti} \end{cases}$$

dove $d(n_i)$ è il grado del nodo i .

Poiché la matrice Laplaciana ha una relazione diretta con altre importanti matrici strutturali di un grafo, la matrice di ammissione nodale B può fornire una descrizione completa della struttura della rete elettrica. Di conseguenza, il nostro obiettivo è progettare un metodo appropriato e rapido per determinare la struttura della matrice B dalle misurazioni dei parametri della PN.

3.1.3 Distribuzione gaussiana dei parametri

In [43, 31, 42] è stato discusso a fondo che, a causa dell'incertezza del carico nelle reti di trasmissione su larga scala e del crescente contributo di fonti rinnovabili distribuite in PN su piccola scala, i poteri attivi e reattivi iniettati possono essere modellati come variabili casuali. Inoltre, *“la potenza iniettata può essere modellata come una variabile casuale gaussiana poiché modella la sovrapposizione di molti fattori indipendenti (ad es. la carica)”* [43]. La suddetta relazione lineare in (15) suggerisce

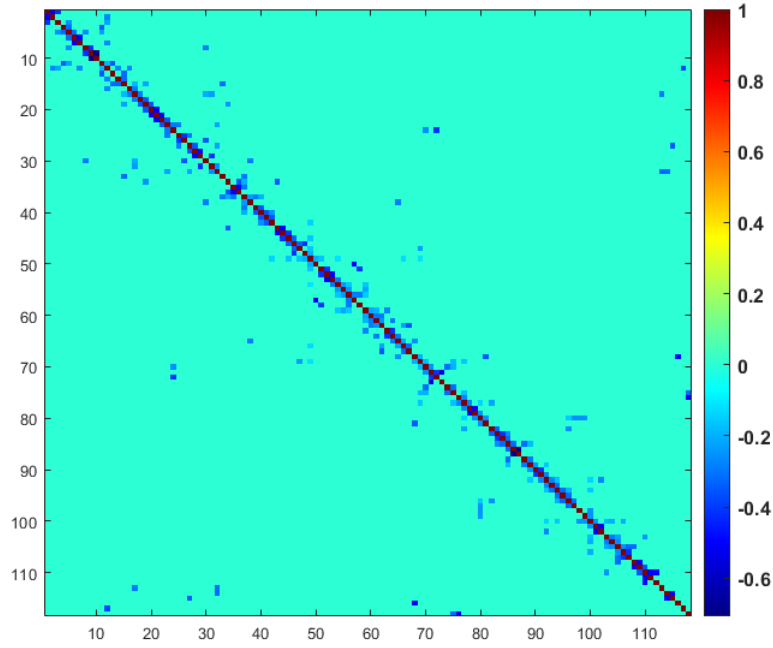


Figura 3.4: Rappresentazione della matrice Laplaciana dello Standard IEEE a 188 Bus.

che la differenza degli angoli di fasore ($\theta_i - \theta_j$) per ogni campione di tempo attraverso un bus può essere approssimata anche da una variabile casuale gaussiana [?]. Inoltre, consentendo il fasore fisso sullo *slack bus*, in condizioni di stato stazionario le singole misurazioni dell'angolo di fasore θ possono essere modellate come variabili casuali gaussiane. Di conseguenza, supponiamo che durante la finestra di osservazione di un problema T_I , le misurazioni dell'angolo di fase del nodo i possano essere ben approssimate da variabili casuali gaussiane. Questo comportamento casuale ha un impatto sulla struttura della matrice di rilevamento nel problema T_I ed è vantaggioso per le prestazioni delle tecniche di recupero sparse.

3.2 Identificazione topologia sparsa

Ci concentriamo ora sul problema del recupero della matrice di *nodal-admittance* (o di topologia) B costruita con i vettori delle misurazioni p e θ (vedi (17)), o equivalentemente, q e v (vedi (18)). Un presupposto chiave nel nostro lavoro è che la SG può essere considerata come un sistema sparso interconnesso. Questo presupposto si basa su un sondaggio di articoli e modelli di reti di alimentazione standard presenti in software e toolbox come MATLAB. Osserviamo che il livello massimo di connettività di un bus in una rete è in genere inferiore al 5-10%, specialmente nei sistemi di trasmissione di potenza molto grandi. Ad esempio, il livello di connettività più elevato di un bus nel modello *IEEE a 30 Bus* è 6 (10%), nell' *IEEE a 118 Bus* è 9 (7%) e nel *IEEE 2383* è 9 (0,3%). Ciò significa che ogni colonna nella matrice di *nodal-admittance* soddisfa la definizione di un segnale sparso. Questa sparsità ci aiuta a formulare T_I come un problema di recupero sparso, che come sostenuto

dalla teoria CS, può essere risolto con un piccolo set di misurazioni in modo rapido e preciso utilizzando gli algoritmi SR. Se l'assunzione di sparsità viene violata, tuttavia, le prestazioni dell'algoritmo ne risentiranno. Nelle prossime sezioni formuleremo T_I come un problema di SR e introdurremo l'algoritmo SR risolutore utilizzato in questo studio.

3.2.1 Impostazione del problema sparso di identificazione

Data una PN interconnessa di N bus, l'angolo di fase e le misurazioni della potenza attiva dell' i -esimo bus sono associate alle seguenti serie temporali di campionamento M :

$$p_i(t) \quad \text{per } t = 1, 2, \dots, M \quad (20)$$

$$\theta_i(t) \quad \text{per } t = 1, 2, \dots, M \quad (21)$$

Come accennato in precedenza, nel modello a corrente continua, la potenza (attiva o reattiva) iniettata in un bus distinto segue la legge di sovrapposizione (15) e (16). Dato che (16) non può essere validato correttamente in alcune situazioni pratiche, preferiamo usare (15) per formulare il problema sparso T_I . Ciò significa che, per ogni nodo i nella rete e ad ogni tempo di campionamento t , abbiamo

$$p_i(t) = \sum_j p_{ij}(t) = \sum_{j \in N_i} b_{i,j}(\theta_i(t) - \theta_j(t)) \quad (22)$$

Come accennato in precedenza, $b_{i,j}$ è la suscettività lungo la linea $l_{i,j}$ sotto il modello a corrente continua, e N_i è l'insieme dei bus vicini collegati direttamente al bus i . Se scriviamo $b_{i,j} = 0$ per $j \notin N_i$, possiamo estendere (22) come segue:

$$p_i(t) = \sum_j p_{ij}(t) = \sum_{j \in S_N^i} b_{i,j}(\theta_i(t) - \theta_j(t)) + u_i(t) + e_i(t) \quad (23)$$

dove S_N^i è l'insieme di tutti i nodi nella rete escluso i (quindi S_N^i conterrà $N - 1$ nodi), u_i è una possibile perdita di potenza attiva nel nodo i stesso, ed e_i è la misura del rumore. Poiché supponiamo che i dati vengano raccolti in M tempi di campionamento, possiamo usare la notazione a matrice vettoriale per scrivere

$$p_i = A_i y_i + u_i + e_i \quad (24)$$

dove, $p_i, u_i, e_i \in \mathbb{R}^M$, e

$$A_i = [a_{1,i}^T, \dots, a_{i-1,i}^T, a_{i+1,i}^T, \dots, a_{N,i}^T] \in \mathbb{R}^{M \times N-1}, \quad (25)$$

$$a_{j,i}^T = (\theta_i(t) - \theta_j(t)) \in \mathbb{R}^M \quad \text{per } t = 1, 2, \dots, M, \quad (26)$$

$$y_i = [b_{i,1}, \dots, b_{i,i-1}, b_{i,i+1}, \dots, b_{i,N}]^T \in \mathbb{R}^{N-1}. \quad (27)$$

Ogni colonna della matrice A_i rappresenta la differenza degli angoli di fase tra il nodo i e un nodo $j \in S_N^i$ nella rete attraverso M campioni di tempo. Come discusso precedentemente, questa differenza può essere modellata come una variabile casuale gaussiana. Il vettore x_i è un vettore sparso, con tutti i suoi elementi uguali a zero tranne una piccola porzione, che si trova in quelle posizioni corrispondenti ai (pochi)

vicini del nodo i . Sommando u_i e e_i (modellato come un vettore di rumore Gaussiano bianco), raggiungiamo la seguente equazione per ogni singolo bus:

$$p_i = A_i y_i + \eta_i \quad (28)$$

Il problema T_I può essere visto come la stima di tutti i vettori $\{y_i\}_{i=1}^N$ che corrispondono meglio alle misurazioni osservate $\{p_i\}_{i=1}^N$. Per risolvere un simile problema, possiamo definire il seguente problema di ottimizzazione:

$$\min_{\{\hat{y}_i\}_{i=1}^N} \sum_{i=1}^N \|A_i \hat{y}_i - p_i\|_2^2 \quad (29)$$

Il problema (29) può essere risolto individualmente per ciascun nodo i . Se A_i ha rango massimo, il numero di misurazioni M supera il numero di incognite $N - 1$ e la varianza del vettore del rumore, η_i , è abbastanza piccola, questo problema di ottimizzazione può essere risolto usando le tecniche di *Least Square* (LS). Poiché il valore di N dipende dal numero di bus nella rete, sarà necessario un gran numero di misurazioni M per risolvere questo problema nel caso di grandi reti elettriche. Per evitare questo problema, è consigliabile utilizzare tecniche di recupero sparse per calcolare i vettori $\{y_i\}_{i=1}^N$. Dato che ogni vettore y_i è sparso, possiamo vedere l'obiettivo in (28) come il recupero di un vettore K -sparso $y_i \in \mathbb{R}^{N-1}$ da una serie di osservazioni $p_i \in \mathbb{R}^M$ dove $A_i \in \mathbb{R}^{M \times N-1}$ è la matrice di rilevamento, $\eta_i \in \mathbb{R}^M$ è un vettore di rumore gaussiano bianco e K è il numero di nodi che sono direttamente collegati al singolo nodo i per cui stiamo risolvendo il problema. In generale vogliamo $M \ll N$, quindi il problema può essere risolto usando una serie ragionevole di osservazioni in enormi reti elettriche. Dopo aver risolto questo problema per ciascun vettore sparso y_i corrispondente a ciascun nodo i , possiamo concatenare tutti i vettori sparsi insieme, formare la matrice Laplaciana ponderata B e il processo è completato. Poiché $\theta_i(t) - \theta_j(t) = 0$ per $i = j$, S_N^i dovrebbe includere $N - 1$ nodi; in altre parole, dovremmo tenere il nodo i fuori da S_N^i poiché produce un vettore di zeri nella colonna corrispondente della matrice A_i . Pertanto, non possiamo calcolare direttamente il valore del parametro B_{ii} dal vettore recuperato y_i ; tuttavia, per quanto riguarda la suddetta definizione della matrice B ,

$$B_{ii} = \sum_{j \in N_j} b_{i,j}. \quad (30)$$

Quindi, dopo aver recuperato il vettore y_i , B_{ii} può essere facilmente calcolato. Nella sezione successiva introdurremo i primi algoritmi risolutivi del problema di SR (28). Generalizzando questi metodi, introdurremo nuove strategie adattabili per affrontare le sfide pratiche legate alla risoluzione del problema sparso T_I .

3.3 Algoritmi SR risolutivi: struttura sparsa di T_I e il problema della correlazione dei dati

In generale, le tecniche standard per risolvere i problemi di SR, come abbiamo visto, possono essere classificate in due gruppi principali: algoritmi basati sulla *convex optimization* e algoritmi *greedy*. In questo lavoro useremo il famoso algoritmo greedy detto *Orthogonal Matching Pursuit* (OMP) e l'algoritmo di *convex optimization* noto come LASSO.

3.3.0.1 Stimatore LASSO

Lo stimatore LASSO è definito come segue:

$$\hat{x} = \operatorname{argmin}_x \|y - Ax\|_2^2 + \lambda \|x\|_1. \quad (31)$$

Questo può essere visto come la forma Lagrangiana del problema BPDN (11), visto precedentemente. Rimandiamo alla sezione 2.8.1 per ulteriori approfondimenti.

Algorithm 1 LASSO (L1- Regularized Least Squares Problem Solver)

Richiede: Matrice A , misurazione y , piccola costante λ

Inizializzazione: $t = 0$ (step Newton), $i = 0$, $y^0 = y$, $\frac{1}{t^0}I = A^T A$, tolleranza $\varepsilon = 10^{-5}$

- 1: Calcolo del *duality gap* $\lambda_t = \max\{\|A^T y^t\|_\infty, \lambda\}$
- 2: Aggiorna i criteri di arresto e t
- 3: Uso del metodo di Newton per ottimizzare la funzione di predizione di \hat{x}
- 4: Uso della *Backtracking line search* per la definizione dei valori di predizione

Output: $\hat{x} = x$

3.3.0.2 OMP - Orthogonal Matching Pursuit

L'OMP costituisce una soluzione più sofisticata. In sintesi, l'algoritmo di recupero OMP tenta di correggere gli errori della stima del minimo quadrato del vettore sparso x causati da minori possibili correlazioni tra le colonne della matrice di rilevamento A . OMP utilizza una procedura iterativa per identificare il "supporto (posizione degli elementi diversi da zero)" corretto del vettore K -sparso. Nell'OMP l'approssimazione per x viene aggiornata in ogni iterazione proiettando ortogonalmente y sulle colonne di A (con $\Theta = A$) associate all'attuale insieme di supporto T . L'OMP quindi minimizza $\|y - A\hat{x}\|_2$ su tutto \hat{x} con supporto T . La procedura è descritta nello schema *Algoritmo 2*. Il costo computazionale dell'OMP è determinato dal prodotto tra matrice e vettore, e dalle possibili grandi dimensioni di dati, ma gode generalmente di prestazioni superiori, nel contesto CS, rispetto ad altri metodi analoghi quali il *Matching pursuit*.

Varie forme estese di OMP sono state sviluppate in letteratura, prendendo in considerazione ulteriori livelli di informazione sulla struttura del segnale x o della matrice di rilevamento, abbattendo notevolmente le barriere computazionali poste precedentemente.

3.4 Implementazione degli algoritmi

Abbiamo definito un modello di riferimento degli standard IEEE e fornito una descrizione generale dei metodi risolutivi del problema di CS-SR. In questa sezione verrà descritta l'implementazione riguardante la creazione della matrice A ($\Phi = A$), del segnale sparso x , delle misurazioni y e l'applicazione dei metodi CS-SR enunciati.

Algorithm 2 OMP - Orthogonal Matching Pursuit**Richiede:** Matrice A , misurazione y , criterio di arresto**Inizializzazione:** $r^0 = y, \hat{x}^0 = 0, i = 0, \text{supporto } T^0 = \emptyset$ **Per** $i = 1$, fino a quando non è soddisfatto il criterio di arresto **fai**:associa $h^i = A^T r^i$ 2: identifica l'indicatore di supporto: $\text{sup}^i = \{\arg\max_j |h^i(j)|\}$ aggiorna il supporto: $T^{i+1} = T^i \cup \text{sup}^i$

4: aggiorna la stima del segnale:

$$\hat{x}^{i+1} = \arg\min_{z: \text{supp}(z) \subseteq T^{i+1}} \|y - Az\|_2$$

6: $r^{i+1} = y - A\hat{x}^{i+1}$ $i = i + 1$ **Fino a:** Raggiungi criterio di arresto**Output:** : $\hat{x} = x^i$

3.4.1 Matrice di rilevamento

Avvalendoci del tool MATPOWER, è possibile definire e creare le matrici degli standard IEEE sui quali, prelevato un segnale x , applicheremo le tecniche LASSO e OMP per il recupero sparso. In particolare :

```

1  %%Creazione matrice di rilevamento
2  function [A,n] = creaMatrice()
3
4  % Directory data di MATPOWER contenente
5  % diversi modelli standard IEEE
6  cd C:\Users\matte\Desktop\matpower7.0\data
7  addpath 'C:\Users\matte\Desktop\matpower7.0\lib'
8  % Richiamo il modello desiderato e definisco
9  % il numero di nodi n
10 caseCall = case300;
11 n = 300;
12 % Richiamo la funzione ext2int che permette
13 % di associare correttamente
14 % inidci e nodi
15
16 mpc = ext2int(caseCall)
17
18 cd ..
19 cd lib
20 % Chiamo la funzione makeYbus per la
21 % creazione del modello completo di
22 % nodi, bus e parametri
23 [Ybus, Yf, Yt] = makeYbus(mpc.baseMVA, mpc.bus, mpc.branch);
24
25 A = full(Ybus);
26 % Rendiamo la phi_mat una Uniform Spherical Ensemble (USE)
27 for col_no = 1:size(A,2)
28     A(:, col_no) = A(:, col_no)/norm(A(:, col_no));
29 end

```

3.4.2 Segnale sparso x e misurazione y

Definiamo le modalità con le quali sono scelti rispettivamente x e y .

3.4.2.1 Creazione del segnale sparso x

Scegliamo arbitrariamente la sparsità del nostro vettore e costruiamo così un segnale. Il segnale è creato casualmente in un dato intervallo. Al termine, è riproposta una

rappresentazione del segnale appena creato, Figura 3.5.

```

1 function [x, s0] = creaSegnaleSparso(K,n)
2
3 % Sparsita' del vettore sparso
4 s0 = K; % Numero di elementi non-nulli in x
5
6 % L'intervallo di x e' una distribuzione uniforme gaussiana
7 % Quindi costruiremo una simulazione del segnale da cui prelevare
8 % casualmente dei campioni
9 pd = makedist('Normal');
10 t = truncate(pd,0.1, 0.9);
11
12 x = zeros(n,1); % Inizializziamo x
13
14 % Decidiamo le posizioni casuali di x,
15 % che contengono valori diversi da zero
16 x_pos = [];
17 while length(x_pos)<s0
18     x_pos = unique(randi(n, [1 s0]));
19 end
20
21 % Associamo i valori diversi da zero da una distribuzione uniforme
22 for j=1:s0
23
24     % Tracciamo il valore casuale uniforme dall'intervallo scelto
25     r = random(t,1);
26     curr_val = r;
27     % Riempimento delle posizioni diverse da zero
28     x(x_pos(j)) = curr_val;
29
30 end
31
32 % Rappresentiamo il vettore x
33 figure(1);
34 subplot(2,1,1);
35 stem(real(x));
36 title('Il vettore sparso x');
```

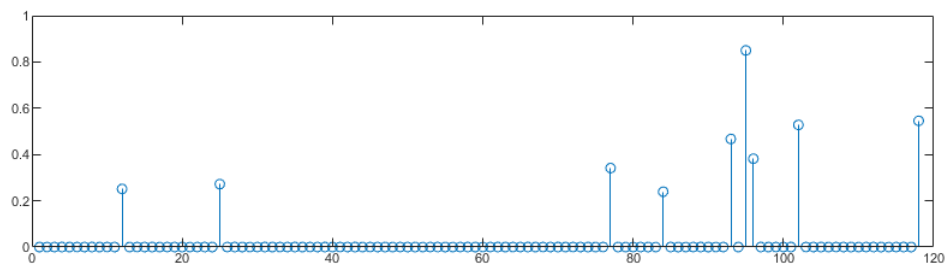


Figura 3.5: Rappresentazione del segnale sparso x nell'IEEE 118 Bus

3.4.2.2 Creazione della misurazione y e generazione del rumore

Prima di calcolare y è necessario stabilire e simulare la presenza di rumore nelle fasi di elaborazione del segnale. Per fare ciò si è definita una funzione generatrice, ovvero:

```

1 function [n_eta, norm_n_eta] = generaRumore(m)
2
3 % Generiamo rumore n_eta
4 sigma_eta = 1; % La varianza del segnale di errore
5 mean_eta = 0; % La media del rumore e' zero (assegnamento)
```

```

6 n_eta = sigma_eta*randn([m 1]) + mean_eta;
7 n_eta = n_eta/norm(n_eta); % Normalizzazione unitaria del rumore
8 norm_n_eta = 0.5; % Settaggio della norma del rumore
9 n_eta = norm_n_eta*n_eta;

```

Successivamente, potremo calcolare la misurazione y , definita, secondo il CS, come $y = \Phi x$, e aggiungere il rumore.

```

1 function y = calcolaY(phi_mat,x,n_eta,norm_n_eta)
2 % Definiamo la misurazione y
3 y = phi_mat*x + n_eta;
4
5 % Rappresentiamo y
6 figure(2);
7 stem(real(y));
8 title(['Il segnale misurato y, ' '||rumore||_2 = ' num2str(norm_n_eta)]);
9 xlabel('indice');
10 ylabel('grandezza');

```

In Figura 3.6, rappresentiamo anche il grafico del segnale misurato.

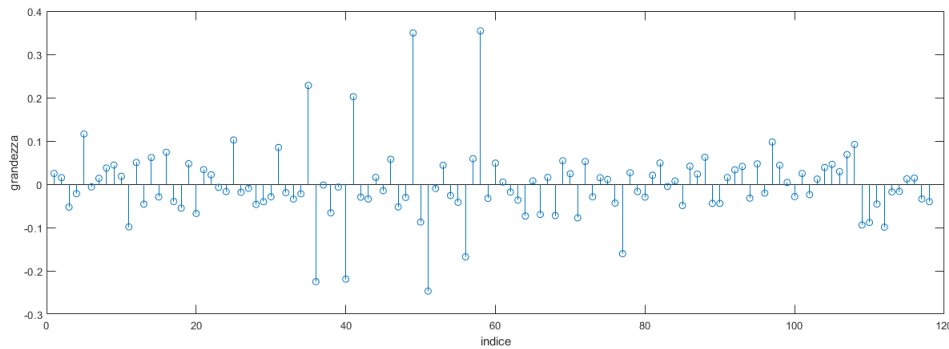


Figura 3.6: Rappresentazione del segnale di misurazione y , con $\| \text{rumore} \|_2 = 0,5$

3.4.3 LASSO

Presentiamo di seguito l'algoritmo risolutore LASSO per il recupero del segnale sparso x .

```

1 function [x,status,history] = l1_ls(A,varargin)
2 %
3 % l1-Regularized Least Squares
4 % l1_ls risolve il problema nella seguente forma:
5 % minimize ||A*x-y||^2 + lambda*sum|x_i|,
6 % dove A e y sono gli input e x e' il segnale campionato.
7
8         [ . . . ]
9
10 %           Ciclo principale
11 cput0 = cputime;
12
13 for ntiter = 0:MAX_NT_ITER
14
15     z = A*x-y;
16
17     %           Calcolo del Duality Gap
18     nu = 2*z;
19     maxAnu = norm(A*nu,inf);
20
21     if (maxAnu > lambda)

```



```

22     nu = nu*lambda/maxAnu;
23 end
24
25 pobj = z'*z+lambda*norm(x,1);
26 dobj = max(-0.25*nu'*nu-nu'*y,dobj);
27 gap = pobj - dobj;
28
29 pobjs = [pobjs pobj]; dobj = [dobj dobj]; sts = [sts s];
30 pflgs = [pflgs pflg]; pitrs = [pitrs pitr];
31 cputs = [cputs cputime - cput0];
32
33 % Criterio di fermata
34 if (~quiet) disp(sprintf('%4d %12.2e %15.5e %15.5e %11.1e %8d',...
35     ntiter, gap, pobj, dobj, s, pitr)); end
36
37 if (gap/dobj < reltol)
38     status = 'Solved';
39     history = [pobjs-dobj; pobjs; dobj; sts; pitrs; pflgs; cputs];
40     if (~quiet) disp('Absolute tolerance reached.');
```

```

41 end
42 return;
43 end
44
45 % Aggiornamento t
46
47 if (s >= 0.5)
48     t = max(min(2*n*MU/gap, MU*t), t);
49 end
50
51 % Calcoliamo lo step del metodo di Newton
52 q1 = 1./(u+x); q2 = 1./(u-x);
53 d1 = (q1.^2+q2.^2)/t; d2 = (q1.^2-q2.^2)/t;
54
55 % Calcolo del gradiente
56 gradphi = [At*(z*2)-(q1-q2)/t; lambda*ones(n,1)-(q1+q2)/t];
57
58 % Calcolo dei vettori usati per la predizione
59 prb = diagtx+d1;
60 prs = prb.*d1-(d2.^2);
61
62 % Settaggio della tolleranza
63 normg = norm(gradphi);
64 pcgtol = min(1e-1,eta*gap/min(1,normg));
65
66 if (ntiter ~= 0 && pitr == 0) pcgtol = pcgtol*0.1; end
67 [dxu,pflg,prelres,pitr,presvec] = ...
68     pcg(@AXfunc_l1_ls,-gradphi,pcgtol,pcgmaxi,@Mfunc_l1_ls,...
69     [],dxu,A,At,d1,d2,d1./prs,d2./prs,prb./prs);
70 if (pflg == 1) pitr = pcgmaxi; end
71
72 dx = dxu(1:n);
73 du = dxu(n+1:end);
74
75 % Backtracking line search
76 phi = z'*z+lambda*sum(u)-sum(log(-f))/t;
77 s = 1.0;
78 gdx = gradphi'*dx;
79
80 for lsiter = 1:MAX_LS_ITER
81     newx = x+s*dx; newu = u+s*du;
82     newf = [newx-newu;-newx-newu];
83
84     if (max(newf) < 0)
85         newz = A*newx-y;
86         newphi = newz'*newz+lambda*sum(newu)-sum(log(-newf))/t;
87
88         if (newphi-phi <= ALPHA*s*gdx)
89             break;
90         end
91     end
92     s = BETA*s;
93 end
94

```

```

95     if (lsiter == MAX_LS_ITER) break; end % uscita
96
97     x = newx; u = newu; f = newf;
98 end
99 history = [pobj-s-dobj-s; pobj-s; dobj-s; sts; pitrs; pflgs; cputs];
100 return;
101
102 [ . . . ]

```

3.4.4 OMP

Mostriamo di seguito l'algoritmo greedy OMP per il recupero del segnale sparso x .

```

1 function [x_est] = OMP(y,norm_n_eta,phi_mat, dim_x)
2
3 %% L'algoritmo OMP
4
5 % Settaggio dei parametri iniziali per l'OMP
6 gamma_curr = y; % gamma_0, residuo iniziale
7 I_set = []; % insieme di colonne che contribuiscono a y
8 col_set = []; % indice delle colonne
9 max_iter = 100; % numero massimo di iterazioni
10 iter_no = 1; % inizializzazione della variabile di iterazione
11 error_iter = Inf; % errore iniziale
12
13 % Soglia di arresto per errore
14 if norm_n_eta > 0
15     thresh = norm_n_eta;
16 else
17     thresh = 10^-10;
18 end
19
20 % Inizializziamo la norma del residuo
21 norm_gamma = [norm(gamma_curr)];
22 % Verifichiamo i criteri di arresto
23 while((iter_no < max_iter+1) && error_iter > thresh)
24     % numero iterazioni
25     max_ip = 0;
26
27     % Troviamo la colonna che contribuisce maggiormente
28     for col_no = 1:size(phi_mat,2)
29         curr_ip = abs(gamma_curr'*phi_mat(:,col_no));
30         if curr_ip > max_ip
31             max_ip = curr_ip;
32             curr_col = col_no;
33         end
34     end
35     col_set = [col_set curr_col]; % Unione degli indici
36     I_set = [I_set phi_mat(:,curr_col)]; % Troviamo la base
37
38     % Troviamo la proiezione del segnale misurato sulla base,
39     % usiamo la funzione apposita
40     z_curr = projection_meg(y, I_set);
41
42     % Aggiornamento del residuo
43     gamma_curr = y - z_curr;
44
45     % Verifica dei criteri di arresto
46     if norm(gamma_curr) > norm_gamma(end)
47         break
48     else
49         norm_gamma = [norm_gamma norm(gamma_curr)];
50     end
51
52     % aggiornamento errore e indice di iterazione
53     error_iter = norm(gamma_curr);
54     iter_no = iter_no+1;
55 end
56
57 % Troviamo la stima finale di x

```

```

58 col_set = unique(col_set);
59 I_set = [];
60
61 % Troviamo x da I_set
62 for col=1:length(col_set)
63     I_set = [I_set phi_mat(:, col_set(col))];
64 end
65
66 % Risoluzione del problema nella dimensione ridotta di I_set
67 y_reduced_dim = zeros(length(col_set),1);
68 phi_reduced_dim = zeros(length(col_set),length(col_set));
69 x_reduced_dim = zeros(length(col_set),1);
70
71 for col = 1:length(col_set)
72     y_reduced_dim(col) = y'*I_set(:, col);
73 end
74
75 for row = 1:length(col_set)
76     for col = 1:length(col_set)
77         phi_reduced_dim(row, col) = I_set(:, row)'\*I_set(:, col);
78     end
79 end
80
81 % Troviamo la versione ridotta di x
82 x_reduced_dim = pinv(phi_reduced_dim)*y_reduced_dim;
83
84 % Riempiamo il vettore sparso x con gli elementi
85 % della versione ridotta di x
86 x_est = zeros(dim_x);
87 for pos = 1:length(col_set)
88     x_est(col_set(pos)) = x_reduced_dim(pos);
89 end

```

Capitolo 4

Simulazioni: Test e valutazione

In questa sezione discuteremo dei test e delle conseguenti misure di valutazione delle performance risolutive delle tecniche di recupero precedentemente enunciate.

4.1 Modello e misure di valutazione

4.1.1 Scelta del modello di test

Si è scelto di effettuare i test sul funzionamento del CS di entrambe le tecniche considerate, OMP e LASSO, su diversi modelli di PN dello standard IEEE. In particolare, si è scelto di effettuare i test sulle strutture a 30, 118 e 300 Bus IEEE, mostrate in Figura 4.1.1.

4.1.2 Scelta delle misure di valutazione

Abbiamo individuato diversi parametri quantitativi e qualitativi per la valutazione delle performance. Ovvero: *sparsità* del segnale, scelta del λ (LASSO), *tempo* di esecuzione e tasso di *errore e correlazione*. A seconda del rapporto e dei risultati ottenuti al cambio di una o più variabili considerate forniremo delle considerazioni.

4.1.2.1 Sparsità

Effettueremo una valutazione del grado recupero di una tecnica di CS in relazione al valore di sparsità del segnale campionato. Come descritto precedentemente le SG, anche di grandi dimensioni, non richiedono, nella fase di *recovery*, un numero di valori sparsi molto grande (circa da 3 a 9 elementi sparsi). Nelle nostre simulazioni proporremo misure di sparsità realistiche (3,5,7 e 9 elementi) e irrealistiche (15 e 20 elementi) con lo scopo di constatare l'influenza di questo fattore nelle performance del CS.

4.1.2.2 Scelta del λ

Nel metodo convesso LASSO il parametro λ costituisce un importante fattore influente nelle performance quanto nel recupero del segnale. Sceglieremo diversi valori di λ , ovvero $\lambda = 0.1, 0.01, 0.001$, al fine di valutare quello ottimale in un dato contesto.

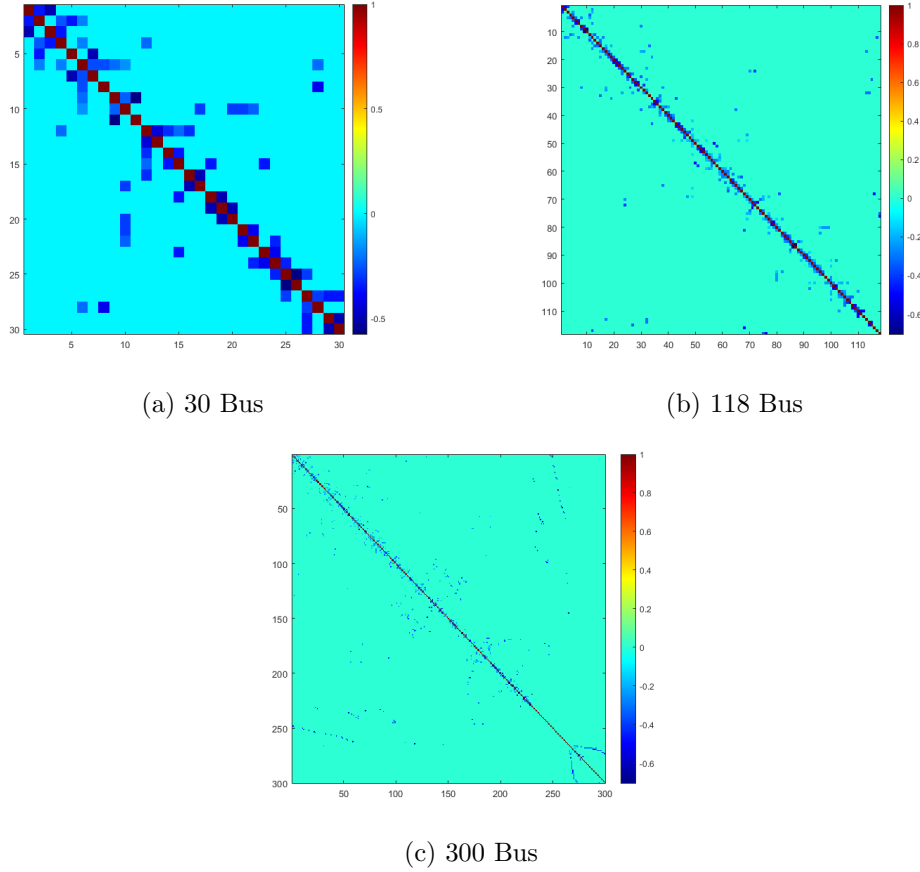


Figura 4.1: Matrice Laplaciana rappresentante gli Standard IEEE

4.1.2.3 Tempo

Stabilita la sparsità e, nel LASSO un λ , effettueremo diverse misurazioni temporali per verificare come e quanto il recupero di un segnale risenta della variazione del valore di sparsità nel segnale e/o della scelta del λ . Abbiamo usato a questo proposito la funzione *tic-toc* fornita da MATLAB. La misurazione temporale fornita sarà relativa all'esclusiva tecnica in esame, quindi escludendo da ciò la creazione della matrice, del segnale sparso e della misurazione.

4.1.2.4 Errore e correlazione

Per la valutazione effettiva del grado di recupero del segnale sparso x adottiamo tre misure: la radice dell'errore quadratico medio (RMSE, *Root Mean Squared Error*), l'indice di correlazione di Pearson e la somma assoluta dell'errore.

Lo *RMSE* della stima \hat{x} rispetto al parametro stimato x è definito come:

$$RMSE(\hat{x}) = \sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n}}$$

ovvero la radice dell'errore quadratico medio. Questa ci dà una misura per giudicare la *qualità* dello stimatore in termini della sua variazione e della sua distorsione.

Definiamo invece l'*indice di correlazione di Pearson* come:

$$r_{x,\hat{x}} = \frac{\sum_{i=1}^n (x_i - \bar{x})(\hat{x}_i - \bar{\hat{x}})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (\hat{x}_i - \bar{\hat{x}})^2}}$$

ovvero, date due variabili x e \hat{x} e la loro media \bar{x} e $\bar{\hat{x}}$, esso è definito come la covarianza divisa per il prodotto delle deviazioni standard delle due variabili. Il valore $r_{x,\hat{x}}$ risultante, compreso fra $[-1, 1]$, esprime la relazione di linearità tra le due variabili. Ciò ci garantirà una valutazione ancora più completa, che consideri la presenza del rumore, quantificando, quindi, quanto quest'ultimo influisca nella predizione.

Infine, la somma assoluta dell'errore, ovvero:

$$err = \sum_{i=1}^n |\hat{x}_i - x_i|$$

sarà la somma in valore assoluto della differenza del segnale predetto \hat{x} e il segnale originario x . Essa fornirà una misurazione quantitativa della presenza di errore nel segnale stimato.

Al termine di ogni esecuzione è proposto un grafico descritto in Figura 4.2, che rappresenta la differenza tra $\hat{x} - x$, ovvero tra il segnale stimato e quello originale. Questo ci garantirà una ulteriore valutazione fattiva e visiva delle prestazioni di recupero del segnale, ciò anche in relazione al variare del λ (nel LASSO) e della sparsità del vettore x .

4.2 Metodologia

Il metodo con cui abbiamo organizzato i test si basa su una struttura composta da diversi casi. Ogni caso e i suoi risultati saranno frutto della media di *cinque* misurazioni. Abbiamo associato un caso ad ogni grado di sparsità K , e cioè:

- Caso 1 - $K = 3$
- Caso 2 - $K = 5$
- Caso 3 - $K = 7$
- Caso 4 - $K = 9$
- Caso 5 - $K = 15$
- Caso 6 - $K = 20$

Ogni caso sarà oggetto di misurazioni per entrambe le tecniche oggetto di studio, ottenendo così un caso per ogni tecnica. La tecnica LASSO, come detto precedentemente, aggiunge un ulteriore parametro da considerare nelle misurazioni, cioè la scelta del λ . Per questa ragione abbiamo scomposto ogni caso LASSO in un sotto-caso associato ad un certo valore λ . Otterremo così, per ogni caso e per il solo LASSO quattro sotto-casi. Per chiarezza forniamo la struttura della misurazione del *Caso 1* (quindi con sparsità $K = 3$), per entrambe le tecniche, ottenendo così:

- OMP - Caso 1 - $K = 3$;
- LASSO - Caso 1A - $K = 3$, $\lambda = 1$;
- LASSO - Caso 1B - $K = 3$, $\lambda = 0.1$;
- LASSO - Caso 1C - $K = 3$, $\lambda = 0.01$;
- LASSO - Caso 1D - $K = 3$, $\lambda = 0.001$;

Avremo infine un totale di 150 misurazioni. Questa procedura verrà adottata per ogni standard IEEE a 30,118 e 300 Bus, arrivando ad un totale di 450 misurazioni.

4.3 Misurazione

Presentiamo di seguito i codici usati per eseguire il test. In particolare, il codice per il test della tecnica convessa LASSO:

```

1 function [end_alg, RMSE , PEARSON ,errore] = TestLASSO(K,1)
2
3 %% Genero la matrice IEEE
4 [phi_mat,m] = creaMatrice()
5 % Definiamo le dimensioni della matrice phi = m x n
6 n = m;
7 %% Genero il segnale sparso x
8 [x,s0] = creaSegnaleSparso(K,n);
9
10 %% Genero il rumore e la misurazione y
11 % Generiamo l'errore n_eta e la norma del rumore
12 [n_eta, norm_n_eta] = generaErrore(m);
13
14 % Calcoliamo la misurazione y
15 y = calcolaY(phi_mat,x,n_eta,norm_n_eta);
16
17 %% L1 Regulation o LASSO
18 % Richiamo la funzione lasso
19 % diamo per parametri, la matrice, la misurazione e lambda
20 lambda = 1
21
22 % Uso la funzione tic-toc di Matlab per misurare i tempi di esecuzione
23 tic
24 z = l1_ls(phi_mat,y,lambda);
25 end_alg = toc
26
27 % Plottiamo per confrontare il segnale effettivo e il segnale stimato x_est
28 figure(3);
29 subplot(3,1,1);
30 stem(real(x))
31 title(['Segnale sorgente, s = ' num2str(s0)]);
32 xlabel('indice');
33 ylabel('grandezza');
34
35 subplot(3,1,2);
36 stem(real(z));
37 title(['Segnale stimato, ' 'con Lamda = ' num2str(lambda)]);
38 xlabel('indice');
39 ylabel('grandezza');
40
41 q = (z - x)
42
43 subplot(3,1,3);
44 stem(real(q));
45 title(['Errore, ']);
46 xlabel('indice');
47 ylabel('grandezza');
48

```

```

49 %% Calcolo le misure di errore e correlazione
50 RMSE = (sqrt(mean( z - x).^2));
51 ERR= sum(abs(q))
52 PEARSON = corrcoef(x-z)
53
54 return

```

Segue il codice per il test dell'algoritmo OMP:

```

1 function [end_alg, RMSE , PEARSON , ERR] = TestOMP(K)
2 %% Genero la matrice IEEE
3 [phi_mat,m] = creaMatrice()
4
5 % Definiamo le dimensioni della matrice phi = m x n
6 n = m;
7 %% Genero il segnale sparso x
8 [x,s0] = creaSegnaleSparso(K,n);
9
10 %% Genero il rumore e la misurazione y
11 % Generiamo l'errore n_eta e la norma del rumore
12 [n_eta, norm_n_eta] = generaErrore(m);
13 % Calcolo la misurazione y
14 y = calcolaY(phi_mat,x,n_eta,norm_n_eta);
15
16 %% Richiamo l'Algoritmo OMP
17 % Uso la funzione tic-toc di Matlab per misurare i tempi di esecuzione
18 tic
19 x_est = OMP(y,norm_n_eta,phi_mat, dim_x)
20 end_alg = toc
21
22 %% Plotto i risultati per confrontare il segnale effettivo
23 %% e il segnale stimato x_est
24 figure(3);
25 subplot(3,1,1);
26 stem(real(x))
27 title(['Segnale sorgente, x = ' num2str(s0)]);
28 xlabel('indice');
29 ylabel('grandezza');
30
31 subplot(3,1,2);
32 stem(real(x_est));
33 title(['Segnale stimato, ' '||rumore||_2 = ' num2str(norm_n_eta)]);
34 xlabel('indice');
35 ylabel('grandezza');
36
37 subplot(3,1,3);
38 j = x_est-x
39 stem(real(j))
40 title(['Errore,']);
41 xlabel('indice');
42 ylabel('grandezza');
43
44 %% Calcolo le misure di errore e correlazione
45 RMSE = real(sqrt(mean( x_est-x).^2));
46 ERR = sum(abs(j))
47 PEARSON = corrcoef(x-x_est)
48 return

```

Avremo per output la Figura 4.2 e i valori delle misurazioni precedentemente descritte.

4.4 Risultati e valutazione

Per brevità e completezza riportiamo i risultati ottenuti dalle misurazioni degli standard a 30,118 e 300 Bus nelle rispettive appendici A, B e C. Forniremo qui un'analisi dei risultati ottenuti del solo standard IEEE a 300 Bus. Quest'ultimo, infatti, si è

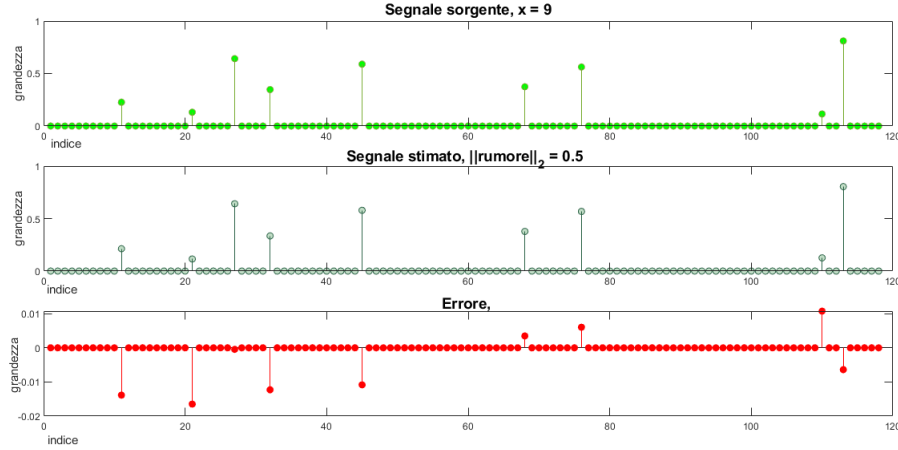


Figura 4.2: Rappresentazione dei risultati del test per uno Standard IEEE a 118 Bus

rilevato essere il più rappresentativo e onnicomprensivo degli standard, evidenziando pregi, difetti, peculiarità e limiti di ogni tecnica e caso specifico. Inoltre, è risultato essere il modello IEEE più *stressante* dei tre e ciò ha permesso di marcare maggiormente le differenze delle tecniche oggetto di studio. Effettueremo prima un'analisi del caso più specifico e quindi del LASSO, per poi passare ad una valutazione della tecnica algoritmica OMP.

4.4.1 Valutazione LASSO

Inizieremo la valutazione della tecnica LASSO partendo dalla prima scelta del parametro λ a seguito del quale considereremo il parametro di *sparsità*.

4.4.1.1 Scelta del λ

Stabiliremo il migliore valore λ , ovvero quello in grado di fornire il miglior compromesso tra tempo e basso tasso di errore.

Considerando il solo fattore temporale i valori λ più stabili, come è evidente in Figura 4.3, sono $\lambda = 0.1$ e $\lambda = 0.001$. Infatti, risultano essere preferibili rispetto agli altri in quanto mantengono una certa regolarità anche al variare di K .

Considerando il solo fattore RMSE, Figura 4.4, la distorsione del segnale predetto è pressoché nulla nei valori $\lambda = 0.1$ e $\lambda = 0.001$, soprattutto in relazione ai valori di K fondamentali (3,5,7 e 9). Inoltre, esaminando i risultati delle misurazioni in relazione all'indice di Pearson e ai valori λ , Figura 4.5, avremo una ulteriore conferma nei valori $\lambda = 0.1$ in particolare, e $\lambda = 0.001$. Analizzando invece la somma dell'errore assoluto, Figura 4.6, le migliori performance sono ottenute da $\lambda = 0.1$ seguito da $\lambda = 0.001$.

Considerando l'unione dei fattori di errore e correlazione, il compromesso migliore risulta essere $\lambda = 0.1$ seguito da $\lambda = 0.001$.

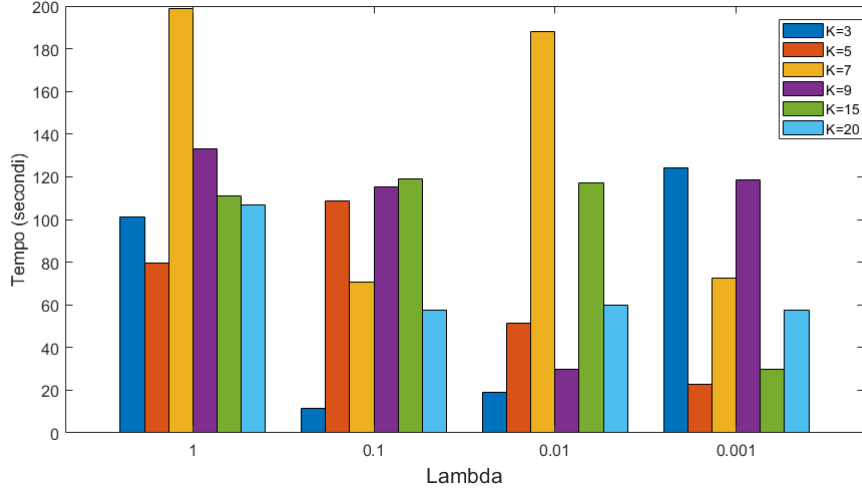


Figura 4.3: Valori λ in relazione al tempo per uno Standard IEEE 300 Bus

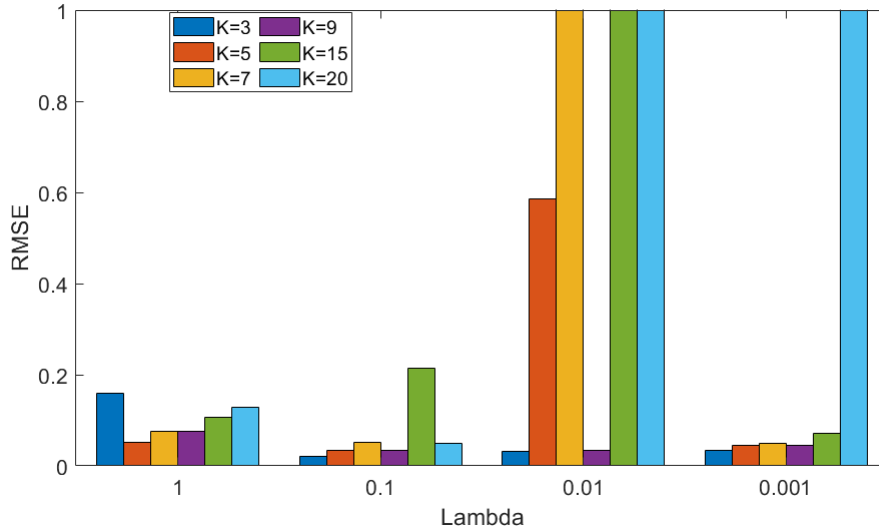


Figura 4.4: Valori λ in relazione al RMSE per uno Standard IEEE 300 Bus

4.4.1.2 Sparsità

In questo paragrafo faremo un'osservazione relativa alla sparsità valutando le performance del LASSO al variare di K . Considereremo in questo caso solo nel complesso i valori λ associati ad ogni caso.

Analizzando il fattore temporale, all'aumentare della sparsità la tecnica sembra reagire meglio, richiedendo mediamente tempistiche più brevi, come evidenziabile in Figura 4.7. Prendendo in considerazione i valori λ invece noteremo come le osservazioni proposte precedentemente siano in parte coerenti, purché mitigate da una certa variabilità di caso in caso.

Esaminando la misura RMSE in relazione alla sparsità, Figura 4.8, noteremo un medio e progressivo aumento di distorsione e varianza all'aumentare di K . Osservando λ e i valori precedente scelti (0.1 e 0.001), noteremo come in genere quest'ultimi performino meglio degli altri.

Considerando invece l'indice di Pearson rispetto la sparsità, Figura 4.9, noteremo

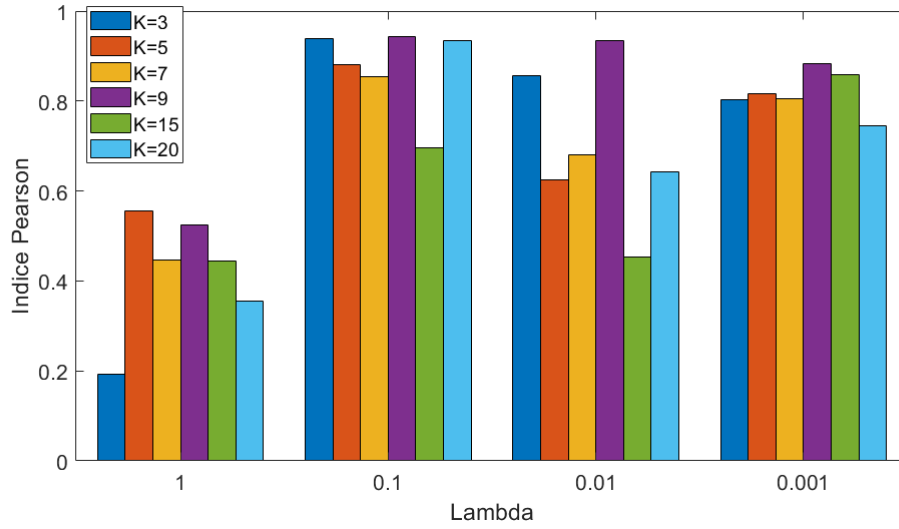


Figura 4.5: Valori λ in relazione all'indice di Pearson per uno Standard IEEE 300 Bus

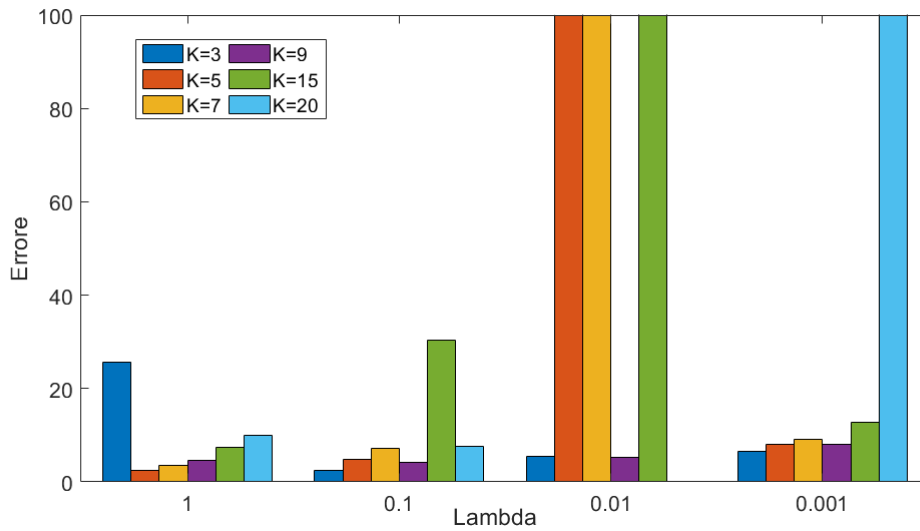


Figura 4.6: Valori λ in relazione alla somma dell'errore per uno Standard IEEE 300 Bus

come, all'aumentare di K , in media la tecnica tende a performare peggio. Anche in questo caso i λ scelti tendono a dare performance migliori delle restanti.

Infine, osservando la sommatoria dell'errore assoluto, Figura 4.10, noteremo un medio aumento progressivo dell'errore all'aumento di K .

Concludiamo l'analisi sulla sparsità prendendo in esame tutti i fattori coinvolti. Constatiamo un lieve peggioramento delle performance al variare di K , ma tale variazione è fortemente dipendente dalla scelta del valore λ . Quest'ultimo è in grado di influire pesantemente sui tempi e le performance mettendo di fatto in secondo piano il problema della sparsità.

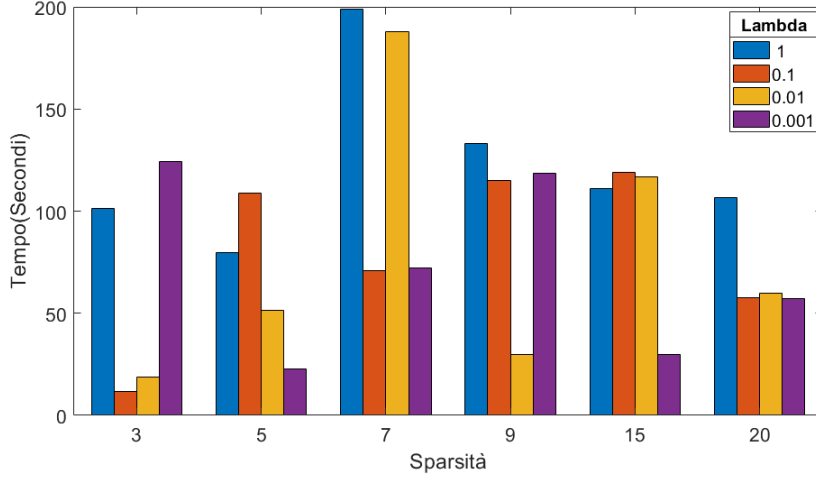


Figura 4.7: Sparsità K in relazione al tempo per uno Standard IEEE 300 Bus

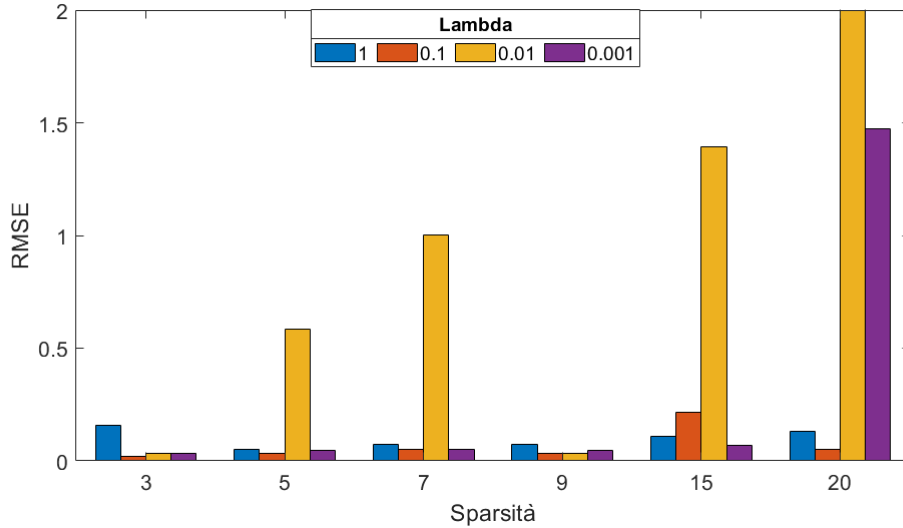


Figura 4.8: Sparsità K in relazione al RMSE per uno Standard IEEE 300 Bus

4.4.2 Valutazione OMP

Effettueremo la valutazione dell'algoritmo OMP considereremo il solo parametro di *sparsità*.

Analizzando il fattore temporale, Figura 4.11, notiamo un generico aumento delle tempistiche (salvo il *Caso 2* con $K = 5$) all'aumentare della sparsità.

Prendendo in esame la misura RMSE, Figura 4.12, si evidenzia un netto peggioramento delle performance nei Casi 5 e 6, i quali ricordiamo costituire dei valori irrealistici in relazione al contesto.

Osservando, invece, la sommatoria dell'errore assoluto, Figura 4.13, evidenziamo un aumento dell'errore progressivo all'aumentare di K .

Si rende doveroso effettuare due importanti osservazioni. Per primo, possiamo notare la marcata differenza nelle dimensioni delle misure rispetto al LASSO. Infatti, emerge la natura *greedy* dell'algoritmo proprio dall'alto grado di precisione evidenziabile nei tempi (frazioni di secondo), nello RMSE (fattore di scala $\times 10^{-4}$)

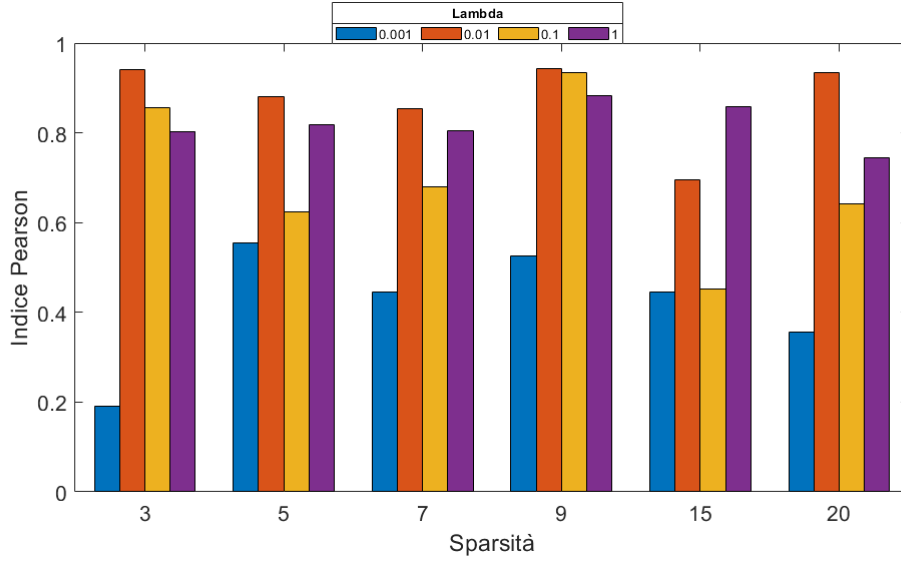


Figura 4.9: Sparsità K in relazione all'indice di Pearson per uno Standard IEEE 300 Bus

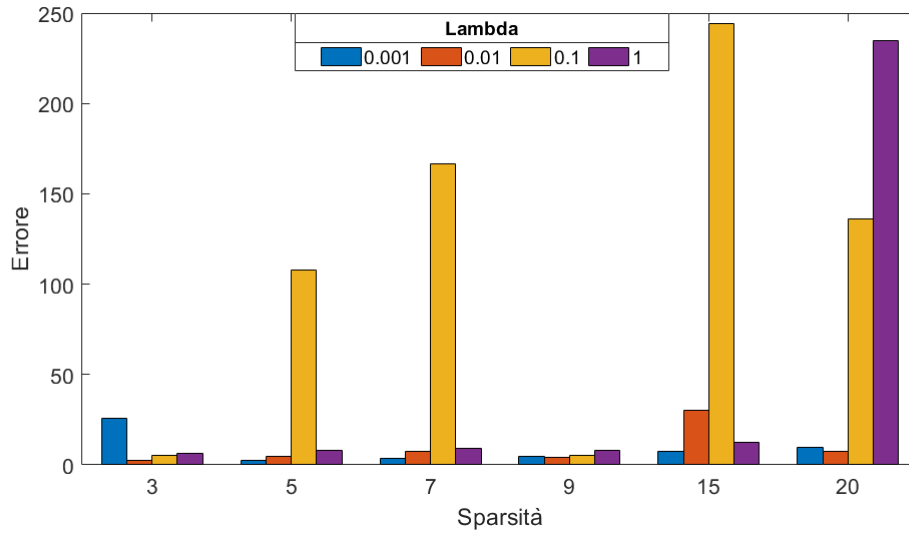


Figura 4.10: Sparsità K in relazione alla somma dell'errore per uno Standard IEEE 300 Bus

e nell'errore (valori < 1). Per secondo, come osservabile, non si è reso necessario rappresentare l'indice di Pearson in rapporto alla sparsità. Infatti, in tutte le misurazioni il parametro $r_{x,\hat{x}}$ eguagliava sempre 1, garantendo una correlazione del 100% in ogni caso proposto. Possiamo affermare in definitiva che, benché risulti un lieve peggioramento delle performance all'aumentare di K , quest'ultimo fattore sia *de facto* irrilevante e impercettibile.

4.4.3 Confronto LASSO vs OMP

Benché l'analisi valutativa delle due tecniche appena proposta risulti decisamente esaustiva nell'evidenziare le nette differenze, proponiamo un rapido confronto delle tecniche oggetto di studio. Infatti, se confrontiamo i tempi medi di esecuzione,

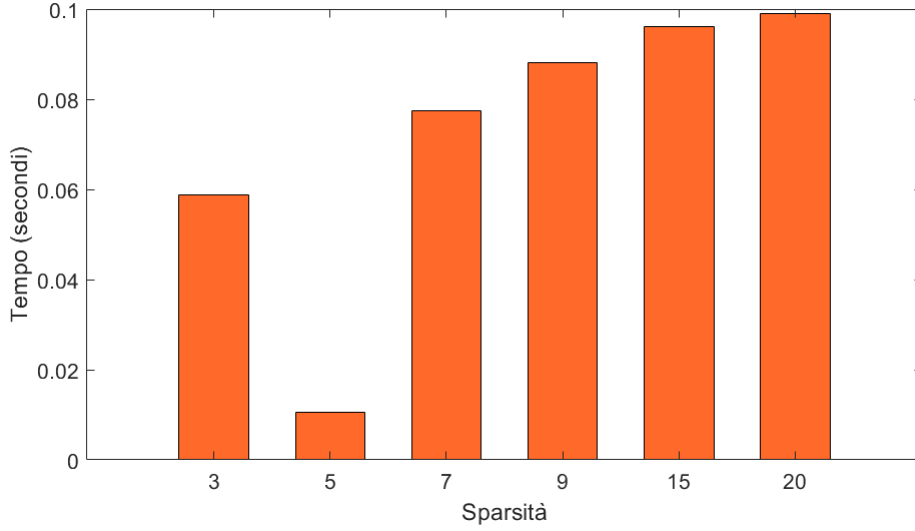


Figura 4.11: Sparsità K in relazione al tempo per uno Standard IEEE 300 Bus

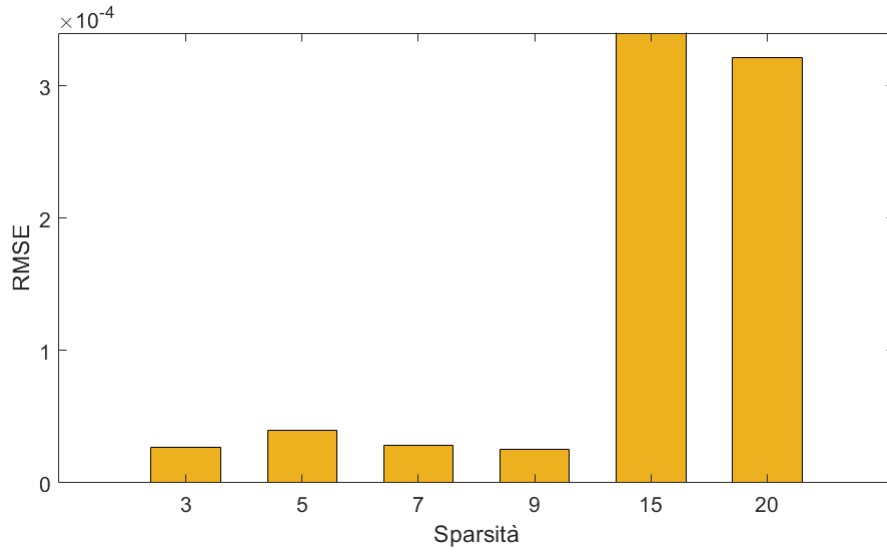


Figura 4.12: Sparsità K in relazione allo RMSE per uno Standard IEEE 300 Bus

il LASSO tocca picchi di circa 110 secondi, a fronte dell'OMP che nel caso peggiore arriva a 0.8 secondi. Considerando lo RMSE, il LASSO ha mediamente un valore RMSE di 0.5 rispetto l'OMP con valori in media vicini allo 0.002. Come anticipato l'OMP consegue un grado di correlazione di Pearson sempre pari a 1, mentre nel LASSO questo dato è fortemente condizionato e variabile, avvicinandoci mediamente e nel migliore dei casi a 0.85. Per ultimo l'errore, nel LASSO assume valori mediamente vicini al 4, nell'OMP e sempre inferiore a 1. Per quanto emerso l'OMP costituisce la migliore soluzione al vaglio di tutti i fattori da noi considerati, garantendo stabilità, tempestività, robustezza e ottime performance.

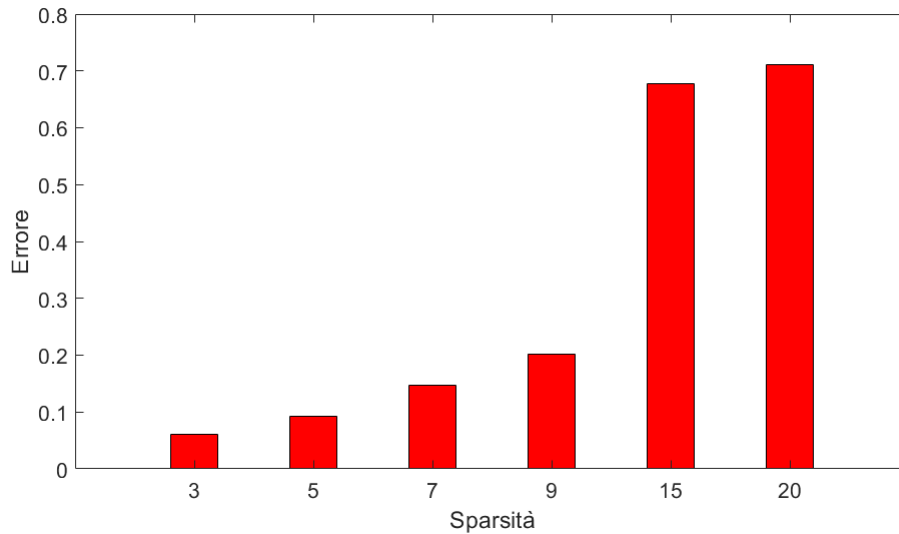


Figura 4.13: Sparsità K in relazione alla somma dell'errore per uno Standard IEEE 300 Bus

4.5 Conclusioni

In questa tesi abbiamo analizzato le *Smart Grid*, osservando le grandi sfide e le conseguenti opportunità proposte. Abbiamo esaminato come la teoria *Compressive Sensing* costituisca una soluzione al problema della sparsità, garantendo una piena osservabilità dei dati, se pur proveniente da un piccolo campione. Definendo matematicamente il problema e studiandone le tecniche principali, abbiamo implementato due delle tecniche fondamentali del CS-SR e ne abbiamo analizzato i contenuti. Infine, tramite la definizione di un modello di simulazione, abbiamo toccato con mano e valutato l'effettivo ed enorme potenziale di queste tecniche. L'applicazione delle tecniche CS nelle Smart Grid e in altri contesti costituisce la risorsa che oggi, e nel prossimo futuro, garantirà di raggiungere e varcare limiti fino a qualche decennio fa insuperabili per l'uomo, aprendo nuove porte alla ricerca e garantendo nuove opportunità alla conoscenza umana.

Ringraziamenti

Sono così tanti i mentori, i colleghi, gli amici e i familiari che hanno contribuito durante gli anni al raggiungimento di questo obiettivo, che citarne alcuni in particolare è al contempo doveroso e ingiusto. La mia gratitudine, e le mie scuse, vanno pertanto a tutti coloro che pur non venendo menzionati hanno comunque svolto un ruolo fondamentale (e voi sapete chi siete!). Consapevole che non sono le cose che possediamo a fare la differenza, ma coloro che fanno parte della nostra vita, volgo un dovuto e sentito ringraziamento :

Alla Prof.ssa Giuseppina Settanni, relatrice di questa tesi, per avermi guidato e supportato nella fase più importante del mio percorso accademico, fornendomi dritte indispensabili e la propria professionalità nella realizzazione di ogni capitolo della mia tesi.

Ai miei genitori. Prima o poi tutte le ansie e le preoccupazioni che vi ho gentilmente donato dovevano essere ripagate! A voi devo molto se non tutto. Vi ringrazio per il supporto, la fiducia e l'amore dimostratomi ogni giorno. Mi avete cresciuto all'insegna della cultura, dell'amore, della sincerità e dell'onestà, principi e valori in cui credo e crederò per il resto della mia vita. Grazie infinite per tutto questo.

A mia sorella Silvia, per avermi dimostrato la sua vicinanza e la sua volontà a capirmi, quando io stesso non ero disposto a farlo. Grazie per essere stata mia guida, quando ero privo d'orientamento, e adesso mia irrinunciabile compagna di viaggio. Grazie per esserci ed esserci sempre stata.

Alla mia fidanzata Federica, per avermi fatto conoscere cos'è l'Amore e scoprire la bellezza di sentirsi amati. Con te ho condiviso i momenti più duri della mia carriera universitaria, e celebrato le gioie e i successi. Grazie per avermi supportato (e sopportato) ogni giorno ed aver mantenuto la tua promessa. Grazie per la fiducia, l'amore, il duro lavoro d'incoraggiamento, la complicità e tutta la bellezza che hai portato nella mia vita

Al mio amico Francesco, per aver dato luce a ciò che era buio ed avermi aiutato nel riscoprire le mie potenzialità. Sei stato partecipe nel cambiamento e leale compagno di avventura. Grazie per le infinite ore passate insieme a parlare e dietro uno strumento, ore spensierate e di crescita.

Agli amici di sempre, Gianluca, Marina, Florenzo, Vera e Nicola. Ognuno di voi ha lasciato ricordi indelebili e grande ricchezza, senza le quali non sarei la persona che sono oggi. Grazie per i profondi spunti di crescita, critica e di ricerca che mi hanno aiutato ad approfondire e crescere la mia personalità negli anni.

Agli amici (e solo dopo colleghi) Sarah, Vanessa, Andrea De Rinaldis , Andrea Pichierri, Andrea Paone, Roberto, Vincenzo, Giuseppe e Daniele, con i quali ho condiviso la quotidianità universitaria ed extra-universitaria. Grazie per le infinite risate, la sincera compagnia e il supporto durante questi anni.

Infine, a tutti coloro che hanno incrociato la loro vita con la mia lasciandomi qualcosa di buono. Grazie per essere stati miei complici, ognuno a suo modo, in questo percorso intenso ed entusiasmante, nel bene e nel male. Sono così tanti i ricordi che mi passano per la testa che è impossibile trovare le parole giuste per onorarli. A farlo saranno le mie emozioni, i miei sorrisi e le mie lacrime che insieme si mescolano in un bagaglio di affetto sincero e gratitudine per tutti voi.

Grazie per aver reso il mio traguardo davvero speciale!

Appendici

Appendice A

Test IEEE a 30 Bus

Caso:	<i>30.1</i>	Tecnica :	<i>OMP</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>0.00091 sec</i>	
Sparsità K		<i>3</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0,0105</i>	
$r_{x,\hat{x}}$		<i>1</i>	
Errore		<i>0.4487</i>	
% Correlazione		<i>100</i>	

Caso:	<i>30.1.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>3.6931 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.16</i>	
$r_{x,\hat{x}}$		<i>0.3</i>	
Errore		<i>1.89</i>	
% Correlazione		<i>30</i>	
Caso:	<i>30.1.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>3.287 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1179</i>	
$r_{x,\hat{x}}$		<i>0.847</i>	
Errore		<i>2.657</i>	
% Correlazione		<i>85</i>	
Caso:	<i>30.1.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>3.937 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1014</i>	
$r_{x,\hat{x}}$		<i>0.8756</i>	
Errore		<i>2.4934</i>	
% Correlazione		<i>87.5</i>	
Caso:	<i>30.1.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>37.7148 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>1.0509</i>	
$r_{x,\hat{x}}$		<i>0.4538</i>	
Errore		<i>23.9418</i>	
% Correlazione		<i>45</i>	

Caso:	30.2	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0014 sec	
Sparsità K		5	
Misura		Valore	
RMSE(\hat{x})		0.0529	
$r_{x,\hat{x}}$		1	
Errore		1.8084	
% Correlazione		100	

Caso:	30.2.a	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		1.5211 sec	
Sparsità K		5	
Valore λ		1	
Misura		Valore	
RMSE(\hat{x})		0.1398	
$r_{x,\hat{x}}$		0.753	
Errore		2.0981	
% Correlazione		75	
Caso:	30.2.b	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		3.716 sec	
Sparsità K		5	
Valore λ		0.1	
Misura		Valore	
RMSE(\hat{x})		0.1605	
$r_{x,\hat{x}}$		0.7829	
Errore		3.3415	
% Correlazione		78	
Caso:	30.2.c	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		5.703 sec	
Sparsità K		5	
Valore λ		0.01	
Misura		Valore	
RMSE(\hat{x})		0.1098	
$r_{x,\hat{x}}$		0.8875	
Errore		2.5345	
% Correlazione		89	
Caso:	30.2.d	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		4.6393 sec	
Sparsità K		5	
Valore λ		0.001	
Misura		Valore	
RMSE(\hat{x})		0.2335	
$r_{x,\hat{x}}$		0.811	
Errore		4.523	
% Correlazione		81	

Caso:	30.3	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0045 sec	
Sparsità K		7	
Misura		Valore	
RMSE(\hat{x})		0.0553	
$r_{x,\hat{x}}$		1	
Errore		2.1747	
% Correlazione		100	

Caso:	<i>30.3.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>7.665 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.2434</i>	
$r_{x,\hat{x}}$		<i>0.508</i>	
Errore		<i>4.796</i>	
% Correlazione		<i>50</i>	
Caso:	<i>30.3.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>0.923 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.107</i>	
$r_{x,\hat{x}}$		<i>0.899</i>	
Errore		<i>2.324</i>	
% Correlazione		<i>90</i>	
Caso:	<i>30.3.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>4.252 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.23</i>	
$r_{x,\hat{x}}$		<i>0.778</i>	
Errore		<i>5.276</i>	
% Correlazione		<i>78</i>	
Caso:	<i>30.3.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>4.092 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.172</i>	
$r_{x,\hat{x}}$		<i>0.865</i>	
Errore		<i>3.805</i>	
% Correlazione		<i>86.5</i>	

Caso:	30.4	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0011 sec	
Sparsità K		9	
Misura		Valore	
RMSE(\hat{x})		0.0361	
$r_{x,\hat{x}}$		1	
Errore		1.6936	
% Correlazione		100	

Caso:	<i>30.4.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>9.723 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.307</i>	
$r_{x,\hat{x}}$		<i>0.254</i>	
Errore		<i>5.2763</i>	
% Correlazione		<i>25</i>	
Caso:	<i>30.4.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>9.61 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.318</i>	
$r_{x,\hat{x}}$		<i>0.867</i>	
Errore		<i>7.874</i>	
% Correlazione		<i>87</i>	
Caso:	<i>30.4.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>1.651 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1314</i>	
$r_{x,\hat{x}}$		<i>0.902</i>	
Errore		<i>2.971</i>	
% Correlazione		<i>90</i>	
Caso:	<i>30.4.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>11.414 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1916</i>	
$r_{x,\hat{x}}$		<i>0.81</i>	
Errore		<i>4.349</i>	
% Correlazione		<i>81</i>	

Caso:	30.5	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0016 sec	
Sparsità K		15	
Misura		Valore	
RMSE(\hat{x})		0.161	
$r_{x,\hat{x}}$		1	
Errore		5.4091	
% Correlazione		100	

Caso:	<i>30.5.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>6.402 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.3624</i>	
$r_{x,\hat{x}}$		<i>0.058</i>	
Errore		<i>7.61</i>	
% Correlazione		<i>0</i>	
Caso:	<i>30.5.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>1.733 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.2181</i>	
$r_{x,\hat{x}}$		<i>0.81</i>	
Errore		<i>5.035</i>	
% Correlazione		<i>81</i>	
Caso:	<i>30.5.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>1.312 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.207</i>	
$r_{x,\hat{x}}$		<i>0.839</i>	
Errore		<i>4.663</i>	
% Correlazione		<i>84</i>	
Caso:	<i>30.5.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>9.465 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.173</i>	
$r_{x,\hat{x}}$		<i>0.865</i>	
Errore		<i>4.145</i>	
% Correlazione		<i>86.5</i>	

Caso:	30.6	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0025 sec	
Sparsità K		20	
Misura		Valore	
RMSE(\hat{x})		0.233	
$r_{x,\hat{x}}$		1	
Errore		8.3088	
% Correlazione		100	

Caso:	<i>30.6.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>3.25 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.394</i>	
$r_{x,\hat{x}}$		<i>0.508</i>	
Errore		<i>9.5469</i>	
% Correlazione		<i>51</i>	
Caso:	<i>30.6.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>9.467 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.298</i>	
$r_{x,\hat{x}}$		<i>0.803</i>	
Errore		<i>7.408</i>	
% Correlazione		<i>80</i>	
Caso:	<i>30.6.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>6.231 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.27</i>	
$r_{x,\hat{x}}$		<i>0.835</i>	
Errore		<i>6.229</i>	
% Correlazione		<i>83.5</i>	
Caso:	<i>30.6.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>11.519 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.237</i>	
$r_{x,\hat{x}}$		<i>0.832</i>	
Errore		<i>5.185</i>	
% Correlazione		<i>83</i>	

Appendice B

Test IEEE a 118 Bus

Caso:	<i>118.1</i>	Tecnica :	<i>OMP</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>0.0047 sec</i>	
Sparsità K		<i>3</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.00011</i>	
$r_{x,\hat{x}}$		<i>1</i>	
Errore		<i>0.0838</i>	
% Correlazione		<i>100</i>	

Caso:	<i>118.1.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>20.552</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0654</i>	
$r_{x,\hat{x}}$		<i>0.525</i>	
Errore		<i>1.6437</i>	
% Correlazione		<i>52.5</i>	
Caso:	<i>118.1.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>3.300 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0.049</i>	
$r_{x,\hat{x}}$		<i>0.7463</i>	
Errore		<i>3.0441</i>	
% Correlazione		<i>75</i>	
Caso:	<i>118.1.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>8.2446 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0415</i>	
$r_{x,\hat{x}}$		<i>0.8799</i>	
Errore		<i>3.5857</i>	
% Correlazione		<i>88</i>	
Caso:	<i>118.1.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>25.0383 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0859</i>	
$r_{x,\hat{x}}$		<i>0.7011</i>	
Errore		<i>7.0307</i>	
% Correlazione		<i>70</i>	

Caso:	118.2	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0023 sec	
Sparsità K		5	
Misura		Valore	
RMSE(\hat{x})		0.0654	
$r_{x,\hat{x}}$		1	
Errore		0.2789	
% Correlazione		100	

Caso:	<i>118.2.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>29.3710 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.2006</i>	
$r_{x,\hat{x}}$		<i>0.2976</i>	
Errore		<i>12.909</i>	
% Correlazione		<i>30</i>	
Caso:	<i>118.2.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>19.915 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0441</i>	
$r_{x,\hat{x}}$		<i>0.9164</i>	
Errore		<i>3.4284</i>	
% Correlazione		<i>92</i>	
Caso:	<i>118.2.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>10.226 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0363</i>	
$r_{x,\hat{x}}$		<i>0.9446</i>	
Errore		<i>2.9399</i>	
% Correlazione		<i>94</i>	
Caso:	<i>118.2.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>24.235 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1163</i>	
$r_{x,\hat{x}}$		<i>0.7548</i>	
Errore		<i>9.7363</i>	
% Correlazione		<i>75.5</i>	

Caso:	118.3	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0019 sec	
Sparsità K		7	
Misura		Valore	
RMSE(\hat{x})		0.003	
$r_{x,\hat{x}}$		1	
Errore		0.648	
% Correlazione		100	

Caso:	<i>118.3.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>15.651 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1624</i>	
$r_{x,\hat{x}}$		<i>0.3701</i>	
Errore		<i>6.6863</i>	
% Correlazione		<i>37</i>	
Caso:	<i>118.3.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>24.9239 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0565</i>	
$r_{x,\hat{x}}$		<i>0.8702</i>	
Errore		<i>3.87</i>	
% Correlazione		<i>87</i>	
Caso:	<i>118.3.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>23.878 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0638</i>	
$r_{x,\hat{x}}$		<i>0.9033</i>	
Errore		<i>5.1865</i>	
% Correlazione		<i>90</i>	
Caso:	<i>118.3.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>23.0681 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1158</i>	
$r_{x,\hat{x}}$		<i>0.6716</i>	
Errore		<i>8.9255</i>	
% Correlazione		<i>67</i>	

Caso:	118.4	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0023 sec	
Sparsità K		9	
Misura		Valore	
RMSE(\hat{x})		0.0040	
$r_{x,\hat{x}}$		1	
Errore		0.788	
% Correlazione		100	

Caso:	118.4.a	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		28.845 sec	
Sparsità K		9	
Valore λ		1	
Misura		Valore	
RMSE(\hat{x})		0.1759	
$r_{x,\hat{x}}$		0.338	
Errore		6.4246	
% Correlazione		34	
Caso:	118.4.b	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		23.888 sec	
Sparsità K		9	
Valore λ		0.1	
Misura		Valore	
RMSE(\hat{x})		0.2129	
$r_{x,\hat{x}}$		0.8596	
Errore		15.794	
% Correlazione		86	
Caso:	118.4.c	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		22.896 sec	
Sparsità K		9	
Valore λ		0.01	
Misura		Valore	
RMSE(\hat{x})		0.8708	
$r_{x,\hat{x}}$		0.8277	
Errore		70.412	
% Correlazione		83	
Caso:	118.4.d	Tecnica :	LASSO
Nr. esecuzioni		5	
Tempo medio per esecuzione		40.1353 sec	
Sparsità K		9	
Valore λ		0.001	
Misura		Valore	
RMSE(\hat{x})		0.1445	
$r_{x,\hat{x}}$		0.7126	
Errore		10.2874	
% Correlazione		71	

Caso:	118.5	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0024 sec	
Sparsità K		15	
Misura		Valore	
RMSE(\hat{x})		0.0065	
$r_{x,\hat{x}}$		1	
Errore		1.099	
% Correlazione		100	

Caso:	<i>118.5.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>18.2561 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1368</i>	
$r_{x,\hat{x}}$		<i>0.4600</i>	
Errore		<i>8.5576</i>	
% Correlazione		<i>46</i>	
Caso:	<i>118.5.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>9.939 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>13.463</i>	
$r_{x,\hat{x}}$		<i>0.754</i>	
Errore		<i>1190.23</i>	
% Correlazione		<i>1</i>	
Caso:	<i>118.5.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>22.32 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0391</i>	
$r_{x,\hat{x}}$		<i>0.97</i>	
Errore		<i>3.47</i>	
% Correlazione		<i>97</i>	
Caso:	<i>118.5.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>22.434 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.085</i>	
$r_{x,\hat{x}}$		<i>0.835</i>	
Errore		<i>7.125</i>	
% Correlazione		<i>86.5</i>	

Caso:	118.6	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0023 sec	
Sparsità K		20	
Misura		Valore	
RMSE(\hat{x})		0.00063	
$r_{x,\hat{x}}$		1	
Errore		0.361	
% Correlazione		100	

Caso:	<i>118.6.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>17.0177 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.125</i>	
$r_{x,\hat{x}}$		<i>0.573</i>	
Errore		<i>4.8371</i>	
% Correlazione		<i>57</i>	
Caso:	<i>118.6.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>19.524 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0492</i>	
$r_{x,\hat{x}}$		<i>0.929</i>	
Errore		<i>3.529</i>	
% Correlazione		<i>93</i>	
Caso:	<i>118.6.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>10.265 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.062</i>	
$r_{x,\hat{x}}$		<i>0.918</i>	
Errore		<i>4.684</i>	
% Correlazione		<i>92</i>	
Caso:	<i>118.6.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>4.512 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.082</i>	
$r_{x,\hat{x}}$		<i>0.834</i>	
Errore		<i>7.167</i>	
% Correlazione		<i>83</i>	

Appendice C

Test IEEE a 300 Bus

Caso:	300.1	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.058 sec	
Sparsità K		3	
Misura		Valore	
RMSE(\hat{x})		0.000027	
$r_{x,\hat{x}}$		1	
Errore		0.0605	
% Correlazione		100	

Caso:	<i>300.1.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>101.146</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.159</i>	
$r_{x,\hat{x}}$		<i>0.1917</i>	
Errore		<i>25.687</i>	
% Correlazione		<i>19</i>	
Caso:	<i>300.1.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>11.6403 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.021</i>	
$r_{x,\hat{x}}$		<i>0.939</i>	
Errore		<i>2.567</i>	
% Correlazione		<i>93</i>	
Caso:	<i>300.1.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>18.9620 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0312</i>	
$r_{x,\hat{x}}$		<i>0.856</i>	
Errore		<i>5.421</i>	
% Correlazione		<i>85.5</i>	
Caso:	<i>300.1.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>124.366 sec</i>	
Sparsità K		<i>3</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0349</i>	
$r_{x,\hat{x}}$		<i>0.8035</i>	
Errore		<i>6.541</i>	
% Correlazione		<i>80</i>	

Caso:	300.2	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0105 sec	
Sparsità K		5	
Misura		Valore	
RMSE(\hat{x})		0.000040	
$r_{x,\hat{x}}$		1	
Errore		0.092858	
% Correlazione		100	

Caso:	<i>300.2.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>79.558 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0525</i>	
$r_{x,\hat{x}}$		<i>0.554702</i>	
Errore		<i>2.5127</i>	
% Correlazione		<i>55.5</i>	
Caso:	<i>300.2.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>108.852 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0342</i>	
$r_{x,\hat{x}}$		<i>0.8802</i>	
Errore		<i>4.7514</i>	
% Correlazione		<i>88</i>	
Caso:	<i>300.2.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>51.29431 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.585</i>	
$r_{x,\hat{x}}$		<i>0.6247</i>	
Errore		<i>107.830</i>	
% Correlazione		<i>62.5</i>	
Caso:	<i>300.2.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>22.889 sec</i>	
Sparsità K		<i>5</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0449</i>	
$r_{x,\hat{x}}$		<i>0.817</i>	
Errore		<i>8.0249</i>	
% Correlazione		<i>82</i>	

Caso:	300.3	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0774 sec	
Sparsità K		7	
Misura		Valore	
RMSE(\hat{x})		0.000028	
$r_{x,\hat{x}}$		1	
Errore		0.146	
% Correlazione		100	

Caso:	<i>300.3.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>198.981 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0752</i>	
$r_{x,\hat{x}}$		<i>0.445</i>	
Errore		<i>3.623</i>	
% Correlazione		<i>44.5</i>	
Caso:	<i>300.3.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>70.744 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0506</i>	
$r_{x,\hat{x}}$		<i>0.8545</i>	
Errore		<i>7.2316</i>	
% Correlazione		<i>85</i>	
Caso:	<i>300.3.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>188.0547 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>1.0001</i>	
$r_{x,\hat{x}}$		<i>0.68</i>	
Errore		<i>166.57</i>	
% Correlazione		<i>68</i>	
Caso:	<i>300.3.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>72.363 sec</i>	
Sparsità K		<i>7</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0505</i>	
$r_{x,\hat{x}}$		<i>0.804</i>	
Errore		<i>9.0716</i>	
% Correlazione		<i>80</i>	

Caso:	300.4	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.088 sec	
Sparsità K		9	
Misura		Valore	
RMSE(\hat{x})		0.000025	
$r_{x,\hat{x}}$		1	
Errore		0.2025	
% Correlazione		100	

Caso:	<i>300.4.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>133.293 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0751</i>	
$r_{x,\hat{x}}$		<i>0.5246</i>	
Errore		<i>4.569</i>	
% Correlazione		<i>52</i>	
Caso:	<i>300.4.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>115.047 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0336</i>	
$r_{x,\hat{x}}$		<i>0.9425</i>	
Errore		<i>4.18329</i>	
% Correlazione		<i>94</i>	
Caso:	<i>300.4.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>29.597 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0338</i>	
$r_{x,\hat{x}}$		<i>0.9345</i>	
Errore		<i>5.2598</i>	
% Correlazione		<i>93</i>	
Caso:	<i>300.4.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>118.490 sec</i>	
Sparsità K		<i>9</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.046</i>	
$r_{x,\hat{x}}$		<i>0.8827</i>	
Errore		<i>8.021</i>	
% Correlazione		<i>88</i>	

Caso:	300.5	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0960 sec	
Sparsità K		15	
Misura		Valore	
RMSE(\hat{x})		0.00101	
$r_{x,\hat{x}}$		1	
Errore		0.6773	
% Correlazione		100	

Caso:	<i>300.5.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>110.998 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.1073</i>	
$r_{x,\hat{x}}$		<i>0.4447</i>	
Errore		<i>7.331</i>	
% Correlazione		<i>46</i>	
Caso:	<i>300.5.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>119.155 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.213</i>	
$r_{x,\hat{x}}$		<i>0.694</i>	
Errore		<i>30.323</i>	
% Correlazione		<i>69</i>	
Caso:	<i>300.5.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>116.94 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>1.393</i>	
$r_{x,\hat{x}}$		<i>0.452</i>	
Errore		<i>244.22</i>	
% Correlazione		<i>45</i>	
Caso:	<i>300.5.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>29.9472 sec</i>	
Sparsità K		<i>15</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0708</i>	
$r_{x,\hat{x}}$		<i>0.858</i>	
Errore		<i>12.6682</i>	
% Correlazione		<i>86</i>	

Caso:	300.6	Tecnica :	OMP
Nr. esecuzioni		5	
Tempo medio per esecuzione		0.0989 sec	
Sparsità K		20	
Misura		Valore	
RMSE(\hat{x})		0.00032	
$r_{x,\hat{x}}$		1	
Errore		0.7102	
% Correlazione		100	

Caso:	<i>300.6.a</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>106.697 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.129</i>	
$r_{x,\hat{x}}$		<i>0.355</i>	
Errore		<i>9.948</i>	
% Correlazione		<i>35.5</i>	
Caso:	<i>300.6.b</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>57.497 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.1</i>	
Misura		Valore	
RMSE(\hat{x})		<i>0.0489</i>	
$r_{x,\hat{x}}$		<i>0.9347</i>	
Errore		<i>7.616</i>	
% Correlazione		<i>93.5</i>	
Caso:	<i>300.6.c</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>59.9473 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.01</i>	
Misura		Valore	
RMSE(\hat{x})		<i>9.093</i>	
$r_{x,\hat{x}}$		<i>0.642</i>	
Errore		<i>1363.157</i>	
% Correlazione		<i>64.2</i>	
Caso:	<i>300.6.d</i>	Tecnica :	<i>LASSO</i>
Nr. esecuzioni		<i>5</i>	
Tempo medio per esecuzione		<i>57.377 sec</i>	
Sparsità K		<i>20</i>	
Valore λ		<i>0.001</i>	
Misura		Valore	
RMSE(\hat{x})		<i>1.473</i>	
$r_{x,\hat{x}}$		<i>0.745</i>	
Errore		<i>234.560</i>	
% Correlazione		<i>74.5</i>	

Bibliografia

- [1] Y. Avonds, Y. Liu, S. Van Huffel, *Simultaneous greedy analysis pursuit for compressive sensing of multi-channel ECG signals*. In *Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Chicago, IL, USA, pp. 6385–6388, 6–30 August 2014.
- [2] M Babakmehr, MG Simoes, MB Wakin, A Durra, F Harirchi. "Smart grid topology identification using sparse recovery". Presented in Industry Applications Society Annual Meeting, to be reviewed in IEEE Trans of Industrial Applications, Oct 2015.
- [3] Baron, D.; Duarte, M.F.; Wakin, M.B.; Sarvotham, S.; Baraniuk, R.G. *Distributed compressive sensing*. arXiv, arXiv:0901.3403, 2009.
- [4] R. G. Baraniuk, M. Davenport, R. DeVore, and M. B. Wakin, *The Johnson-Lindenstrauss lemma meets compressed sensing*, 2006.
- [5] A. Beurling. *Sur les intégrales de Fourier absolument convergentes et leur application à une transformation fonctionnelle*. In Proc Scandi Math Congr, Helsinki, Finland, 1938.
- [6] J.D. Blanchard, J. Tanner, *Performance comparisons of greedy algorithms in compressed sensing*. Numer. Linear Algebra Appl., 22:254–282, 2015.
- [7] Y. Bresler and P. Feng. *Spectrum-blind minimum-rate sampling and reconstruction of 2-D multiband signals*. Proc IEEE Int Conf Image Proc (ICIP), Zurich, Switzerland, 1996.
- [8] Y. Bresler and P. Feng. *Spectrum-blind minimum-rate sampling and reconstruction of multiband signals*. Proc IEEE Int Conf Acoust, Speech, Sig Proc (ICASSP), Atlanta, GA, May 1996.
- [9] E. Candès. *Compressive sampling*. Proc Int Congre Math, Madrid, Spain, 2006.
- [10] E. Candes, *The restricted isometry property and its implications for compressed sensing*. Comptes Rendus Math., 346:589–592, 2008.
- [11] E. Candès e J. Romberg. *Quantitative robust uncertainty principles and optimally sparse decompositions*. Found Comput Math, 6(2):227–254, 2006.
- [12] E. Candès, J. Romberg, e T. Tao. *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*. IEEE Trans Inform Theory, 52(2):489–509, 2006.

- [13] E. Candès, J. Romberg, and T. Tao. *Stable signal recovery from incomplete and inaccurate measurements*. Comm Pure Appl Math, 59(8):1207–1223, 2006.
- [14] E. Candes and T. Tao, *Decoding by Linear Programming*. IEEE Transactions on Information Theory, 51:4203–4215, 2005.
- [15] E. Candès and T. Tao. *Near optimal signal recovery from random projections: Universal encoding strategies?* IEEE Trans Inform Theory, 52(12):5406–5425, 2006.
- [16] E. Candes, T. Tao, *The Dantzig selector: Statistical estimation when p is much larger than n* . Ann. Stat., 35:2313–2351, 2007.
- [17] C. Carathéodory. *Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen*. Math Ann, 64:95–115, 1907.
- [18] C. Carathéodory. *Über den Variabilitätsbereich der Fourierschen Konstanten von positiven harmonischen Funktionen*. Rend Circ Mat Palermo, 32:193–217, 1911.
- [19] C. Cartis, A. Thompson, *A new and improved quantitative recovery analysis for iterative hard thresholding algorithms in compressed sensing*. IEEE Trans. Inf. Theory, 61:2019–2042, 2015.
- [20] Chen, Donoho, Saunders, M.A. *Atomic decomposition by basis pursuit*. SIAM J. Sci. Comput., 20:33–61, 1998.
- [21] W.Dai, Milenkovic, O. *Subspace pursuit for compressive sensing signal reconstruction*. IEEE Trans. Inf. Theory, 55:2230–2249, 2009.
- [22] D. Donoho. *Compressed sensing*. IEEE Trans Inform Theory, 52(4):1289–1306, 2006.
- [23] D.L. Donoho, Y. Tsaig, I. Drori, J.L. Starck, *Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit*. IEEE Trans. Inf. Theory, 58:1094–1121, 2012.
- [24] A. Fannjiang, W. Liao. "Coherence Pattern–Guided Compressive Sensing with Unresolved Grids". SIAM J. IMAGING SCIENCES, vol. 5, no. 1, pp. 179–202, 2012.
- [25] P. Feng. *Universal spectrum blind minimum rate sampling and reconstruction of multiband signals*. PhD thesis, University of Illinois at Urbana-Champaign, Mar. 1997.
- [26] M.A.Figueiredo, R.D. Nowak, S.J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*. IEEE J. Sel. Top. Signal Process., 1:586–597, 2007.
- [27] S. Foucart, H. Rauhut, *A Mathematical Introduction to Compressive Sensing*; Birkhäuser: Basel, Switzerland, Volume 1, 2013.

- [28] I. Gorodnitsky, J. George, and B. Rao. *Neuromagnetic source imaging with FOCUSS: A recursive weighted minimum norm algorithm*. Electroen Clin Neuro, 95(4):231–251, 1995.
- [29] I. Gorodnitsky, J. George and B. Rao. *Source localization in magnetoencephalography using an iterative weighted minimum norm algorithm*. Proc Asilomar Conf Sig, Syst and Comput, Pacific Grove, CA, Oct. 1992.
- [30] I. Gorodnitsky and B. Rao. *Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm*. IEEE Trans Sig Proc, 45(3):600–616, 1997.
- [31] M. He and J. Zhang. *A dependency graph approach for fault detection and localization towards secure smart grid*. IEEE Transactions on Smart Grid, no. 2, pp. 342–351, June 2011.
- [32] V. Kotelnikov *On the carrying capacity of the ether and wire in telecommunications.*, Izd Red Upr Svyazi RKKA, Moscow, Russia, 1933.
- [33] S. Krstulovic, R. Gribonval, *MPTK: Matching pursuit made tractable*. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toulouse, France, 14–19 May 2006; IEEE: Piscataway, NJ, USA, Volume 3, p. III, 2006.
- [34] E. Liu, V.N. Temlyakov, *The orthogonal super greedy algorithm and applications in compressed sensing*. IEEE Trans. Inf. Theory, 58:2040–2047, 2012.
- [35] S.G. Mallat, Z. Zhang, *Matching pursuits with time-frequency dictionaries*. IEEE Trans. Signal Process., 41:3397–3415, 1993.
- [36] Muthukrishnan, S. *Data Streams: Algorithms and Applications*; Now Publishers Inc.: Delft, The Netherlands, 2005.
- [37] D. Needell, J.A. Tropp, *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*. Appl. Comput. Harmon. Anal., 26:301–321, 2009.
- [38] D. Needell, R. Vershynin, *Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit*. Found. Comput. Math. , 9:317–334, 2009.
- [39] H. Nyquist , *Certain topics in telegraph transmission theory*. Trans AIEE, 47:617–644, 1928.
- [40] R. Prony. *Essai expérimental et analytique sur les lois de la Dilatabilité des fluides élastiques et sur celles de la Force expansive de la vapeur de l’eau et de la vapeur de l’alkool, à différentes températures*. J de l’École Polytechnique, Floréal et Prairial III, 1(2):24–76, 1795.
- [41] B. Rao. *Signal processing with the sparseness constraint*. Proc IEEE Int Conf Acoust, Speech, Sig Proc (ICASSP), Seattle, WA, 1998.
- [42] A. Schellenberg, W. Rosehart, and J. Aguado, *“Cumulant-based probabilistic optimal power flow (P-OPF) with Gaussian and Gamma distributions,”* IEEE Trans. on Power Systems, vol. 20, pp. 773–781, May 2005.

- [43] H. Sedghi, E Jonckheere. *Information and Control in Networks*. In Giacomo Como, Bo Bernhardsson, Anders Rantzer, editors, Lecture Notes in Control and Information Sciences, vol. 450, Ch. 9, pp 277-297, 2014.
- [44] C. Shannon. *Communication in the presence of noise*. Proc Inst Radio Engineers, 37(1):10–21, 1949.
- [45] J.A. Tropp, A.C. Gilbert, M.J. Strauss, *Algorithms for simultaneous sparse approximation. Part II: Convex relaxation*. Signal Process., 86:589–602, 2006.
- [46] J.A. Tropp, A.C. Gilbert, M.J. Strauss, *Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit*. Signal Process., 86:572–588, 2006.
- [47] J.A Tropp, A.C. Gilbert , *A.C. Signal recovery from random measurements via orthogonal matching pursuit*. IEEE Trans. Inf. Theory, 53:4655–4666 2007.
- [48] S. Vasanawala, M. Alley, R. Barth, et al. *Faster pediatric MRI via compressed sensing*. Proc Ann Meeting Soc Pediatric Radiology (SPR), Carlsbad, CA, 2009.
- [49] R. Venkataramani and Y. Bresler. *Further results on spectrum blind sampling of 2-D signals*. Proc IEEE Int Conf Image Proc (ICIP), Chicago, IL, 1998.
- [50] M. Vetterli, P. Marziliano, and T. Blu. *Sampling signals with finite rate of innovation*. IEEE Trans Sig Proc, 50(6):1417–1428, 2002
- [51] D. B. West, *Introduction to Graph Theory*, 2nd ed., Prentice Hall, 2001.
- [52] E. Whittaker. *On the functions which are represented by the expansions of the interpolation theory*. Proc Roy Soc Edin Sec A, 35:181–194, 1915.
- [53] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, 2nd ed. New York: Wiley, 1996.
- [54] H. Zhu and G. B. Giannakis, *Sparse Overcomplete Representations for Efficient Identification of Power Line Outages* IEEE Trans. on Power Systems, vol. 23, pp. 1644-1652, Oct. 2012.