



UNIVERSITÀ DEGLI STUDI DI TRIESTE
XXVII CICLO DEL DOTTORATO DI RICERCA IN
GEOSCIENZE

SVILUPPO E SPERIMENTAZIONE DI METODI INNOVATIVI
PER L'ANALISI DEL SEGNALE CON APPLICAZIONI ALLA
GEOFISICA ED AI CONTROLLI NON DISTRUTTIVI.

Dottorando:
Stefano Spagnul

Coordinatore:
Prof. Giovanni Costa

Supervisore di Tesi:
Prof. Michele Pipan

Riassunto

La produzione dei semilavorati di acciaio è realizzata attraverso fasi diverse, che comprendono la fusione, la solidificazione e la deformazione a caldo. La fase di solidificazione avviene normalmente nella macchina di colata continua. In questo processo l'acciaio liquido, posto in un grande contenitore chiamato siviera, viene travasato in un polmone, chiamato paniera, che lo distribuisce alle linee di colata. Nella siviera l'acciaio è ricoperto di scoria liquida per proteggerlo dall'ossidazione, per ridurre la dispersione termica e per catturare le impurità flottanti. Quando lo svuotamento sta per completarsi, aumenta il rischio di trascinarsi verso la paniera della scoria. Attualmente l'operatore umano attraverso l'uso dei propri sensi decide autonomamente di interrompere il flusso. Sarebbe invece utile disporre di sistemi automatici in grado di sostituirsi all'uomo in modo affidabile. Questa ricerca si pone l'obiettivo di esplorare sperimentalmente la possibilità di applicare tecniche di analisi comunemente applicate in geofisica per identificare l'evento catastrofico rappresentato dal passaggio della scoria. Per comprendere in maggior dettaglio gli aspetti fisici coinvolti nel fenomeno è stato intrapreso anche lo studio di tecniche computazionali tipiche della CFD. L'attività di ricerca si è articolata nella realizzazione di un sistema hardware e software in tempo reale che è stato impiegato sia in impianti reali, sia in un modello in scala ad acqua. I risultati ottenuti, applicando la trasformata *wavelet* e il metodo degli attributi istantanei mediante la trasformata di Hilbert, hanno dimostrato la validità dell'approccio e aprono la strada a futuri approfondimenti.

Indice

Indice	iv
1 Introduzione	1
1.1 Il processo di colata continua	1
1.2 Definizione del problema	6
1.3 Caratterizzazione fluidodinamica del fenomeno	15
1.4 Obiettivi dell'attività di ricerca	21
2 Tecniche classiche di stima dello spettro di potenza	25
2.1 Introduzione storica	25
2.2 Definizione di ESD e PSD per segnali discreti	28
2.3 Periodogramma e Correlogramma	30
2.4 Metodo di Bartlett	32
2.5 Metodo di Blackman-Tuckey	33
2.6 Periodogramma di Welch	34
2.7 Periodogramma di Burg	35
2.8 Modelli AR, MA e ARMA.	37
2.8.1 Caso AR	39
2.8.2 Algoritmo di Levinson-Durbin per modelli AR	43
2.8.3 Caso MA	45
2.8.4 Caso ARMA	46
2.9 Filtro di Wiener-Kolmogorov	47
2.10 Modelli di rumore	49
3 Trasformata Wavelet discreta	51
3.1 Introduzione storica	51
3.2 Analisi <i>Wavelet</i> e banco di filtri	52
3.3 Proprietà dei filtri ortonormali	60
3.4 Filtri di Daubechies o <i>Maxflat Filters</i>	63
3.5 Analisi <i>Multi-risoluzione</i>	68
3.6 Wavelet Packets	72
4 Trasformata di Hilbert	75
4.1 Definizione	75
4.2 Segnali analitici, attributi e proprietà	78
4.3 Trasformata di Hilbert-Huang	82
4.4 Modalità di calcolo	84

5	Algoritmi in C++	87
5.1	Descrizione	88
5.1.1	Classe TDWT	89
5.1.2	Classe TClassicPowerSpectrum	105
5.1.3	Classe TFFT	108
5.1.4	Classe TFiltri	113
5.1.5	Classe TFILTROMEDIANO	122
5.1.6	Classe TBufferCircolare	125
5.1.7	Classe THilbert	126
5.2	Interfaccia grafica	129
6	Simulazione numerica con acqua	143
6.1	Introduzione	143
6.2	Caso bifase	146
6.2.1	Modello geometrico e <i>mesh</i>	146
6.2.2	Modello in <i>openFoam</i>	150
6.3	Caso multifase	158
6.3.1	Modello in OpenFOAM	158
6.4	Analisi scalabilità sistema	159
6.5	Risultati numerici – Drain Sink	161
6.5.1	Altezza critica in funzione dell'altezza della colonna di fluido	161
6.5.2	Altezza critica e tempi di svuotamento in funzione del diametro dell'ugello	162
6.5.3	Confronto numerico – sperimentale	164
6.5.4	Analisi profilo superficie libera	166
6.5.5	Analisi campo di moto superficiale	166
6.5.6	Analisi del numero di Reynolds	168
	Calcolo teorico del numero di Reynolds	168
	Calcolo del numero di Reynolds da OpenFOAM	169
7	Apparato sperimentale con acqua	171
7.1	Descrizione dell'apparato sperimentale	171
7.1.1	Descrizione dei sensori	172
7.1.2	Descrizione del sistema di acquisizione	172
7.2	Descrizione degli esperimenti	173
7.3	Analisi dei risultati	176
7.3.1	Scarico libero senza vortici	176
7.3.2	Scarico libero con vortici	177
7.3.3	Scarico in vasca senza vortici	178
7.3.4	Scarico in vasca con vortici	179
7.3.5	Analisi dei tempi di svuotamento	179
8	Misure in impianto industriale	221
8.1	Apparato sperimentale in campo	221
8.2	Analisi mediante metodi di stima spettrale	222
8.3	Analisi mediante i metodi <i>Wavelet</i> ed Hilbert	233
8.4	Analisi degli effetti dovuti alla movimentazione del cassetto della siviera	237

9	Conclusioni	239
A	Costruzione di filtri IIR	243
A.1	Filtro - passa basso	244
A.2	Filtro passa alto	245
A.3	Filtro passa banda	247
A.4	Filtro di Butterworth	248
A.5	Filtro di Chebyshev	249
A.6	Filtro di Chebyshev inverso	249
A.7	Filtro ellittico	250
A.8	Confronto con MATLAB	251
	Bibliografia	253

Capitolo 1

Introduzione

1.1 Il processo di colata continua

L'idea del colaggio continuo dei metalli risale ai primi dell'Ottocento, mentre l'applicazione industriale è iniziata almeno un secolo dopo. Tra gli anni Sessanta e Settanta avviene la crescita significativa del numero di impianti di colata continua, fino a coprire oltre il 90% della produzione mondiale di acciaio. La progettazione degli impianti è continuamente progredita determinando un aumento impressionante della produttività, giustificando in questo modo il continuo aumento degli investimenti. Uno dei fattori che ha determinato questa maggiore produttività è rappresentato dall'aumento graduale della velocità di colata, che, a sua volta, ha permesso di ridurre il numero di linee per una specifica esigenza produttiva. La riduzione delle linee ha determinato una diminuzione non solo dei costi di capitale, ma anche di quelli operativi. Il processo di colata continua, essendo un'operazione di trasformazione dalla fase liquida alla fase solida, è fondamentale per una produzione di acciaio di buona qualità. Nella progettazione di una macchina di colata continua le varie discipline dell'ingegneria (metallurgica, meccanica, elettrica, controllo automatico, fluidodinamica,...) devono operare sinergicamente per garantire buoni risultati. Le macchine di colata attuali incorporano molte sofisticate tecnologie. I procedimenti utilizzati per produrre acciaio sono essenzialmente due (figura 1.1),

- ciclo integrato, dove l'acciaio viene ottenuto per riduzione dei minerali di ferro;
- ciclo da forno elettrico, dove l'acciaio viene ottenuto per fusione dei rottami ferrosi.

La produzione di acciaio in un ciclo integrato è realizzata attraverso una serie di processi correlati tra loro di cui i principali sono:

- la produzione del coke metallurgico;

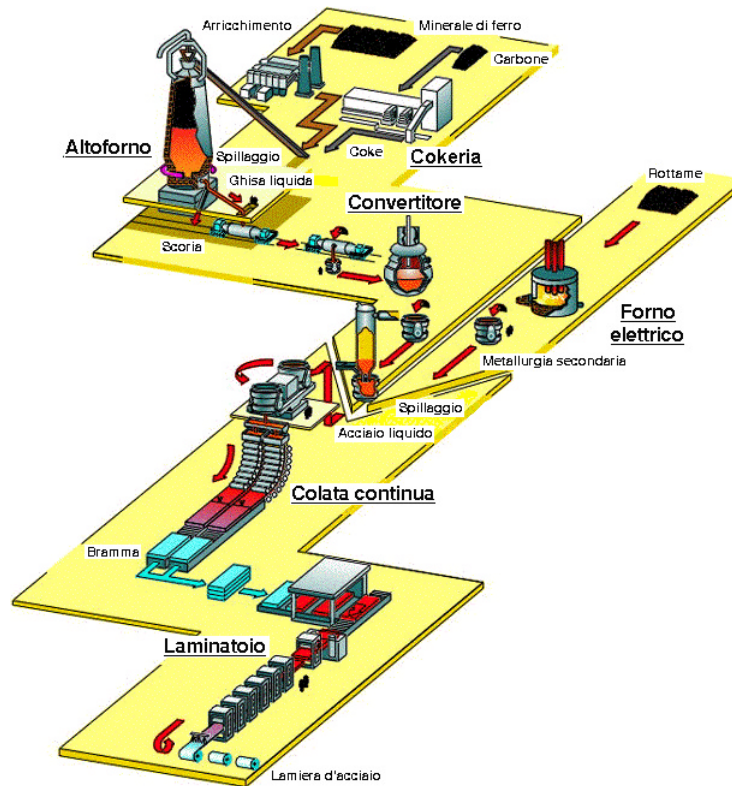


FIGURA 1.1: Cicli per la produzione dell'acciaio.

- la produzione di agglomerato di minerali di ferro;
- la produzione di ghisa;
- la produzione di acciaio con relativa colata continua;

a cui possono seguire dei processi di trasformazione dell'acciaio, nonché la produzione di energia elettrica e vapore attraverso anche l'utilizzo di gas siderurgici (gas di cokeria, gas di altoforno, gas di acciaieria). La produzione di acciaio al forno elettrico è più semplice rispetto al ciclo integrato, in quanto la materia prima è costituita da rottame di ferro. La produzione è essenzialmente realizzata per fusione del rottame di ferro con relativo colaggio in continuo dell'acciaio, a cui possono seguire dei processi di trasformazione dell'acciaio. Fra le materie prime, i minerali di ferro ed i carboni fossili provenienti dai Paesi in cui sono presenti le relative miniere vengono depositati in appositi parchi per poi essere ripresi ed inviati ai processi produttivi. I fondenti invece, essenzialmente costituiti da calcare e dolomite, provengono dalla estrazione in cave locali. La ghisa viene prodotta dagli altoforni con una carica costituita da coke metallurgico, agglomerato, minerali di ferro in pezzatura e fondenti. Il coke metallurgico svolge le seguenti principali funzioni:

- sviluppa il gas riducente necessario alla trasformazione degli ossidi di ferro in ferro metallico;

- fornisce il carbonio necessario per la carburazione della ghisa e per la riduzione di alcuni elementi di lega;
- sostiene il peso del materiale caricato fino alla parte bassa dell'altoforno;
- fornisce il calore necessario alla fusione dei minerali.

Per la produzione di coke metallurgico vengono utilizzate delle miscele di carbone, preparate in appositi impianti con operazioni di vagliatura e dosatura e stoccate in *silos* situati nei pressi dei forni di distillazione. Il caricamento dei forni è effettuato per mezzo delle macchine cariatrici sfruttando la gravità. Nelle celle il carbon fossile distilla ad elevata temperatura (1000-1100°C) ed in assenza di aria, libera le materie volatili generando coke metallurgico con caratteristiche di porosità e di resistenza necessarie per la carica negli altoforni. Ogni cella si presenta come una camera chiusa, con una serie di aperture per le varie operazioni di insufflazione. Il riscaldamento del carbon fossile avviene mediante la combustione di gas di cokeria (o gas di altoforno miscelato con gas di cokeria), nelle camere di combustione adiacenti. La miscela gassosa, che si sviluppa durante la distillazione del carbon fossile, viene raccolta e trattata ed il gas combustibile recuperato, sottoposto ad un processo di depurazione, può essere immesso nella rete di distribuzione alle varie utenze termiche di stabilimento. Il coke metallurgico incandescente, raccolto in un apposito recipiente, viene portato allo spegnimento, effettuato con forti getti di acqua sotto torri dedicate. Successivamente il coke spento viene inviato agli impianti di vagliatura per ottenere la pezzatura idonea alla carica in altoforno. I minerali di ferro fini, per il loro impiego nel processo di produzione della ghisa in altoforno, vengono avviati a fasi di lavorazione intermedie, quali l'omogeneizzazione e la sinterizzazione, nelle quali si realizza l'ottimizzazione dell'analisi chimica e la preparazione di un prodotto agglomerato che ha le caratteristiche chimico-fisiche ottimali per l'impiego in altoforno. Il processo di sinterizzazione dei minerali di ferro viene effettuato negli impianti di agglomerazione, secondo le tre fasi principali di lavorazione: preparazione della miscela, produzione dell'agglomerato e trattamento dell'agglomerato. Per la preparazione della miscela, le materie prime (omogeneizzato fine, coke fine, calcare, calce, prodotti di riciclo) vengono inviate ai mescolatori, dove con l'aggiunta di opportune quantità di acqua e materiale umido di recupero, raggiungono l'umidità desiderata. La miscela umida in uscita dai tamburi mescolatori viene distribuita uniformemente sul nastro di agglomerazione e inviata al processo di sinterizzazione. L'agglomerato, in uscita dal raffreddatore, viene frantumato e vagliato a freddo per ottenere la pezzatura idonea alla carica in altoforno che ha la funzione di trasformare in ghisa i ferriferi della carica. Dalla parte superiore vengono periodicamente introdotti a strati alterni il minerale, già in pezzatura idonea, l'agglomerato, il coke ed i fondenti. Durante la lenta discesa della carica avvengono le reazioni di ossido riduzione degli ossidi di ferro ad opera del gas riducente, che attraversa la carica dal basso verso l'alto. A livello tubiere viene insufflata aria preriscaldata e arricchita con ossigeno, la quale reagisce con il carbonio del coke per dare origine alla miscela gassosa che esercita

l'azione riducente sui minerali di ferro. Negli altoforni vengono anche iniettati a livello tubiere carbon fossile polverizzato, combustibili, oli usati. La tecnica di iniezione di carbon fossile polverizzato rappresenta un'evoluzione dell'ultimo decennio, e consente di ridurre il fabbisogno di coke nel processo di fabbricazione della ghisa con evidenti vantaggi sotto il profilo energetico ed ambientale, in quanto viene in parte ridimensionato il ciclo di produzione del coke metallurgico. Dalla parte alta viene recuperato il gas di altoforno che, dopo opportuna depurazione, viene immesso nella rete di stabilimento per essere utilizzato come combustibile di recupero in varie utenze termiche. Nella parte bassa dell'altoforno, dove più alte sono le temperature, avviene la fusione della carica con la formazione di ghisa e della ganga dei minerali; questa, unitamente alle ceneri di coke ed ai fondenti, determina la scoria, nota come loppa di altoforno. L'evacuazione dei prodotti della riduzione avviene attraverso un apposito foro di colata, situato nella parte bassa dell'altoforno e aperto mediante macchina perforatrice. I prodotti fusi vengono raccolti in un canale principale di colaggio, rivestito con materiale refrattario, ove ghisa e loppa stratificano per effetto dei differenti pesi specifici. Una barriera a sifone posta all'estremità separa i due flussi e determina il loro convogliamento in due canali di colaggio differenti. La ghisa, caricata in appositi carri ferroviari (carri siluro), viene trasferita nelle acciaierie per essere affinata ad acciaio. La trasformazione della ghisa in acciaio (processo di affinazione), avviene riducendo il contenuto di carbonio nella ghisa e con il trasferimento dei prodotti indesiderati nella scoria. Tale processo di affinazione avviene nei convertitori ad ossigeno. La carica nei convertitori è costituita da rottami di ferro, ghisa granulata e da una ghisa fusa. Nel processo riveste un ruolo importante anche l'utilizzo di calce per la formazione della scoria, nella quale vengono inglobate una serie di impurezze dell'acciaio. Il processo di affinazione avviene per insufflazione di ossigeno nel convertitore; reagendo principalmente con il carbonio della ghisa, l'ossigeno produce una fase gassosa che può essere combusta o recuperata per essere riutilizzata come combustibile di recupero. L'acciaio viene spillato dai convertitori in appositi contenitori (siviere) e, successivamente, colato negli impianti di colata continua, dove si trasforma in bramme, blumi, billette (semiprodotto a sezione rettangolare di larghezza ed altezza variabili). Grande importanza hanno assunto gli impianti di trattamento dell'acciaio, a valle dei convertitori, che permettono di ridurre sempre più le impurezze ancora presenti nel bagno e di conferire al prodotto, con l'aggiunta di idonee ferroleghe, particolari caratteristiche meccaniche. L'acciaio solido così prodotto può essere avviato ai successivi processi di trasformazione, quali la laminazione a caldo e a freddo. I gas siderurgici prodotti (gas di cokeria, gas di altoforno, gas di acciaieria), dopo recupero e depurazione, in parte vengono riutilizzati sempre nell'ambito dei suddetti processi produttivi ed in parte per la produzione di energia elettrica e/o vapore in centrali termoelettriche.

Il processo di produzione secondaria da rottame d'acciaio al carbonio e legato, prevede il seguente flusso operativo:

- carica del forno fusorio (EAF);
- fusione nel Forno Elettrico ad Arco (EAF) dell'acciaio e affinazione;
- scorifica e spillaggio;
- trasporto con siviere;
- trattamenti di metallurgia secondaria;
- colata continua o in fossa dell'acciaio.

Il forno EAF fonde la carica e l'acciaio liquido ottenuto viene versato in una secchia chiamata siviera, la cui capacità è equivalente a quella del forno. La siviera è un recipiente in lamiera d'acciaio saldata e internamente rivestito in refrattario. È trasportata da un carroponete di colata e alimenta le macchine di colata continua 1.2.

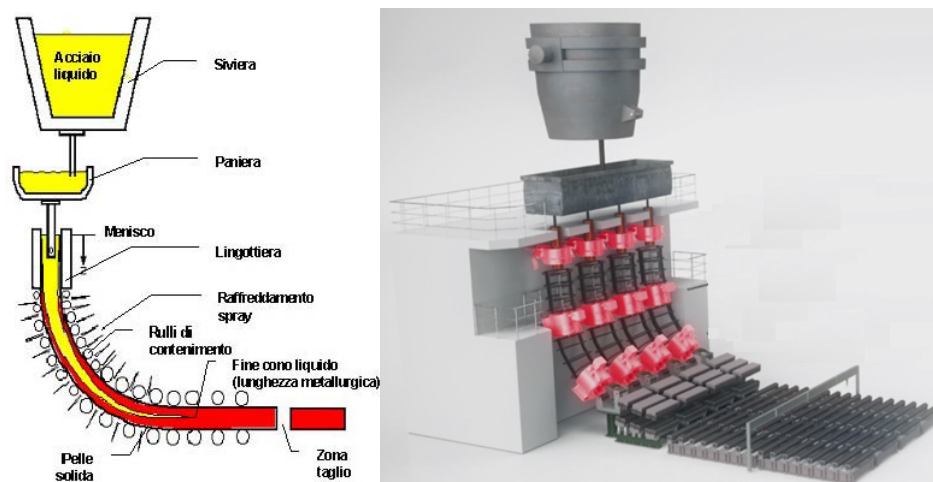


FIGURA 1.2: Macchina di colata continua.

Il principio sul quale si basano i processi di colata continua sono concettualmente semplici: l'acciaio liquido esce verticalmente con continuità e senza interruzioni da un polmone chiamato paniera (figura 1.3). Esso viene appunto alimentato dalle sequenze di siviere che escono dal forno fusorio o dalle stazioni di affinazione. Le linee di colata sono composte da una lingottiera (figura 1.4) contenente uno scambiatore di calore in rame aperto sul fondo, raffreddato ad acqua e della forma desiderata e da un sistema di contenimento basato su rulli. Il calore trasferito al rame causa l'immediata solidificazione dello strato superficiale dell'acciaio che così forma una pellicola solida che incrementa di spessore lungo la lingottiera. La lingottiera ha una lunghezza utile per la formazione di una pelle di acciaio sufficiente a contenere il cuore liquido. Una lunghezza maggiore non sarebbe efficiente poiché l'acciaio ritirandosi riduce drasticamente il calore ceduto al rame. È molto più efficace usare un getto d'acqua diretto tramite ugelli ad alta pressione. La pellicola ancora calda non può sostenere tutto il carico dell'acciaio liquido

e quindi viene fatta scorrere sui rulli. Alla base della colata il prodotto completamente colato viene tirato da rulli estrattori e contemporaneamente raddrizzato per essere successivamente tagliato nelle lunghezze previste.

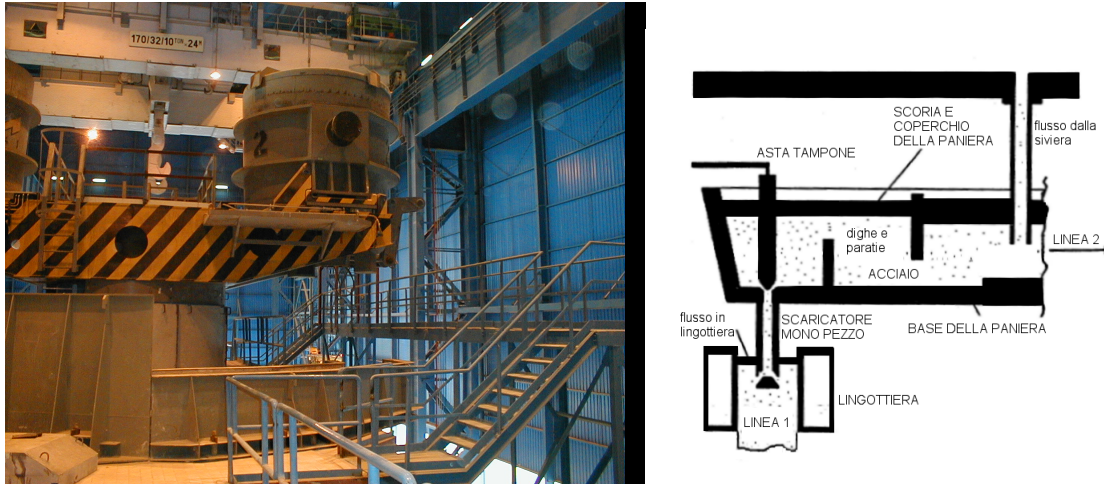


FIGURA 1.3: Siviera e paniera.

La colata dell'acciaio in lingottiera si basa su una complessa tecnologia determinante per il successo qualitativo della produzione. L'obiettivo fondamentale è l'ottenimento di un prodotto che, dopo solidificazione, sia omogeneo chimicamente e che presenti una determinata struttura dei grani. La tecnologia utilizzata deve tener conto dei fenomeni fisici che si svolgono nel corso della solidificazione come il ritiro di solidificazione, le variazioni della struttura cristallina, la formazione delle dendriti, la segregazione chimica, la precipitazione di inclusioni non metalliche e la formazione di soffiature per sviluppo di gas in soluzione nel metallo liquido. Chiaramente nel corso del colaggio dell'acciaio liquido possono formarsi alcuni difetti, che generalmente si manifestano come discontinuità fisiche del materiale metallico, sia superficiali che interne. La tecnologia di colata viene continuamente migliorata per ridurre l'incidenza dei difetti inaccettabili. In effetti il colaggio dell'acciaio è sostanzialmente svincolato dai processi per la produzione dell'acciaio liquido, la macchina di colata è piuttosto condizionata dai tipi di acciai e di prodotti da fabbricare e dalle caratteristiche degli impianti dove vengono svolte le lavorazioni successive del prodotto di acciaieria.

1.2 Definizione del problema

Nel processo di colata continua l'acciaio liquido fuoriesce dalla siviera ed entra nella paniera attraverso un orificio disposto sulla base inferiore. Generalmente esso si trova in una posizione eccentrica. Il fondo della siviera inoltre può essere piatto od inclinato. Sopra la superficie libera dell'acciaio viene depositato uno strato di scoria liquida che previene le perdite termiche e raccoglie le particelle flottanti incluse nell'acciaio. Durante la fase di svuotamento la scoria

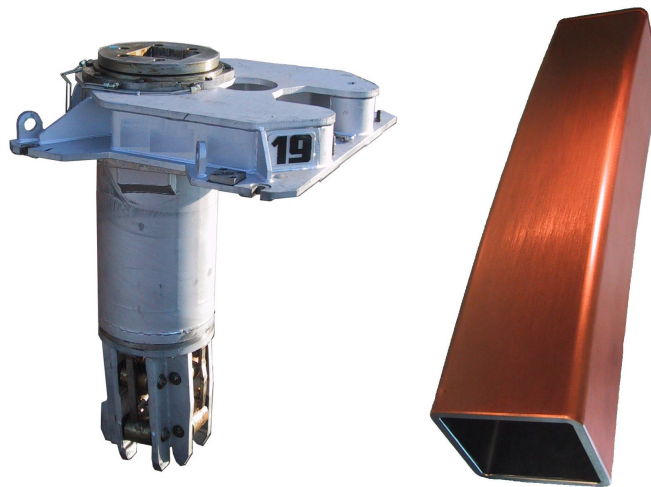
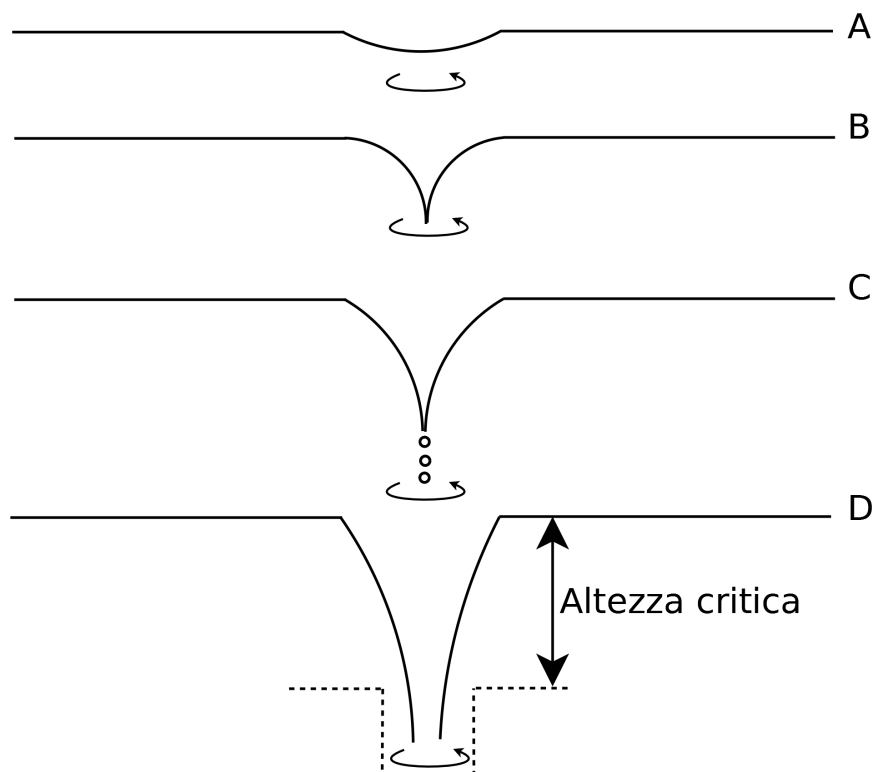


FIGURA 1.4: Lingottiera.

può essere trasportata nella panierina con conseguente perdita della qualità dell'acciaio prodotto. Per i produttori di acciaio questo fenomeno è molto importante e quindi nel corso degli anni si sono svolti diversi studi con l'obiettivo di comprendere la dinamica dello svuotamento [47] [39] [49] [40] [46]. Sono stati individuati due meccanismi: nel primo si instaura un vortice simile a quello che si osserva quando si svuota un lavandino riempito d'acqua: in letteratura è indicato come *funnel vortex*. Il secondo meccanismo prevede uno svuotamento progressivo senza la formazione vortice rotante ed è chiamato *drain sink*. Gli studi sono stati condotti con modelli ad acqua al fine di determinare le caratteristiche di questi processi. La formazione del vortice [51] [54] si sviluppa attraverso diversi stati cui tutti gli studiosi sostanzialmente concordano. Nella prima fase, figura 1.5 - (A), si incontra la formazione di una depressione che appare come una leggera incurvatura della superficie libera. Successivamente, fase (B), essa si trasforma in un vortice di superficie. Il processo si alimenta fino a sfociare in (C), in cui del gas comincia ad essere intrappolato dal vortice fino a raggiungere il foro di uscita. Infine in prossimità dell'altezza critica, H_{cr} , si instaura il vortice completo in cui una colonna di aria viene trascinata stabilmente fino al foro di uscita, fase (D). Dal punto di vista fluidodinamico un vortice è una regione di spazio con un fluido che ruota attorno ad un asse istantaneo. I vortici sono una delle maggiori componenti del flusso turbolento ed in assenza di forze esterne la viscosità tende ad organizzare il fluido in una serie di vortici *irrotazionali*. In questo tipo di vortici la velocità è massima vicino all'asse e decresce in modo inversamente proporzionale alla distanza. Definita quindi la vorticosità come

$$\omega_{\text{vorticosità}} = \nabla \times \mathbf{v}, \quad (1.1)$$

essa è alta nella regione centrale e praticamente nulla in periferia. Il vortice può muoversi e deformarsi trasportando momento angolare, quantità di moto, energia e massa. Si possono distinguere due casi ideali particolarmente importanti:


 FIGURA 1.5: Fasi dello sviluppo di un vortice (*funnel vortex*).

- la velocità aumenta proporzionalmente alla distanza (fluido come corpo rigido);
- la velocità diminuisce con la distanza come $1/r$ (fluido irrotazionale).

Applicando direttamente la definizione di vorticosità (1.1) si ricava che nel primo caso vale,

$$\omega_{\text{vorticosità}} = \frac{v_{\theta}}{r} + \frac{dv_{\theta}}{dr} = 2\omega_{\text{velocità angolare}}, \quad (1.2)$$

mentre nel secondo,

$$\omega_{\text{vorticosità}} = \frac{v_{\theta}}{r} + \frac{dv_{\theta}}{dr} = 0. \quad (1.3)$$

Poiché in assenza di forze esterne un vortice evolve in un vortice irrotazionale, per questa ragione viene chiamato anche *free vortex* o vortice libero. La circuitazione $\Gamma = \oint J \cdot d\mathbf{l}$, $J = \rho\mathbf{v}$, risulta essere costante per ogni percorso che include l'asse di rotazione e nulla per ogni percorso che non lo include. Questo risultato apparentemente paradossale spiega infatti il motivo per cui un corpo immerso in questo tipo di fluido, ruotando attorno al centro non cambia orientazione. Ciò infatti non accade in un corpo rigido. Il valore di Γ è proporzionale al momento angolare, pertanto se esso si conserva anche il valore dell'integrale $\oint J \cdot d\mathbf{l}$ rimane costante. Sotto queste ipotesi la componente tangenziale è data da,

$$v_{\theta} = \frac{\Gamma}{2\pi r}, \quad (1.4)$$

come ipotizzato nella definizione di moto irrotazionale. In effetti questo spiega perché nei vortici osserviamo un incremento della velocità quando la particella si avvicina al centro. Dalla espressione (1.4) si deduce che un moto completamente irrotazionale è fisicamente irrealizzabile, infatti all'asse la velocità cresce all'infinito. Sperimentalmente invece in quella regione si nota che la velocità non cresce più e anzi tende a decrescere progressivamente verso zero. Sulla base di queste osservazioni il più semplice modello di vortice, detto di *Rankine*, assume un comportamento a corpo rigido all'interno di una regione r_0 e diventa invece irrotazionale all'esterno:

$$v_\theta = \begin{cases} \omega r & r \leq r_0 \\ \frac{\omega r_0^2}{r} & r > r_0 \\ v_r = 0 & v_z = 0. \end{cases} \quad (1.5)$$

Nel modello *Lamb-Oseen* risolvendo analiticamente le equazioni del moto nel caso a simmetria cilindrica si trova,

$$v_\theta = (1 - e^{-\frac{r^2}{4\nu t}}) \frac{\Gamma}{2\pi r}. \quad (1.6)$$

In questo caso l'annullamento della velocità all'asse rientra in modo naturale nell'espressione di v_θ . Il movimento del fluido crea dei gradienti di pressione dinamica in modo tale che la pressione è minima nella regione centrale. In un vortice irrotazionale essa assume una dipendenza funzionale del tipo $A - C/r^2 - \rho g z$ per $r > r_0$ e $B + Dr^2 - \rho g z$ per $r \leq r_0$. A questo risultato si giunge risolvendo direttamente le equazioni di Eulero trascurando i termini viscosi. Si comprende inoltre che nel modello proposto la forma della superficie libera seguirà le seguenti leggi,

$$z \sim r^2 \quad r \leq r_0 \\ z \sim \frac{1}{r^2} \quad r > r_0.$$

L'altezza critica, corrispondente allo sviluppo completo del vortice e dipende in modo proporzionale dall'intensità del moto di rotazione iniziale. Effettivamente maggiore è la velocità tangenziale più rapidamente si osserva il vortice. L'eccentricità del foro di scarico invece riduce drammaticamente il valore di H_{cr} [23] [45] e quindi a parità di condizioni al contorno il vortice tende a formarsi ad altezze più basse. Gli studi di Sankaranarayanan e Guthri [49] dimostrano inoltre che le dimensioni e la forma del contenitore contribuiscono in modo non rilevante alla dinamica di formazione del vortice. Il diametro del foro di scarico invece ha un effetto diretto sull'altezza del battente critico: maggiore la sua dimensione maggiore il valore di H_{cr} . Nella figura 1.6 si può osservare l'andamento tipico di H_{cr} in funzione della rotazione iniziale (v_{θ_0}). In particolare se il foro si trova disposto assialmente all'orifizio di scarico l'altezza critica aumenta rapidamente con v_{θ_0} fino a tendere ad un valore asintotico. L'eccentricità invece tende a comprimere l'altezza critica in modo significativo pur mantenendo la stessa dipendenza funzionale. Applicando il semplice modello introdotto in precedenza ed ipotizzando un fluido non viscoso si

può ricavare [46] l'andamento della deformazione della superficie durante la fase di svuotamento sino al raggiungimento dell'altezza critica. Partendo infatti dall'equazione della vorticosità che ne descrive il suo sviluppo temporale e spaziale e applicando il modello di *Rankine*, si ricava che la deformazione h della superficie del vortice segue questo andamento funzionale,

$$h(t) = h_0 e^{f(v_{\theta 0})t}. \quad (1.7)$$

L'andamento globale del livello invece come è noto segue una legge quadratica del tipo $H(t) = H_0(1 - \frac{t}{\tau})^2$, con τ tempo totale di svuotamento. Questo fatto suggerisce come da una condizione iniziale relativamente piatta la superficie evolva rapidamente verso la formazione del vortice completo. Si può inoltre osservare che con l'annullarsi della velocità di rotazione il valore di H_{cr} non tende a zero, in effetti in questa regime si instaura un altro meccanismo di svuotamento. Quando $v_{\theta 0}$ tende a zero, in prossimità di un valore del battente pari a quello del diametro del

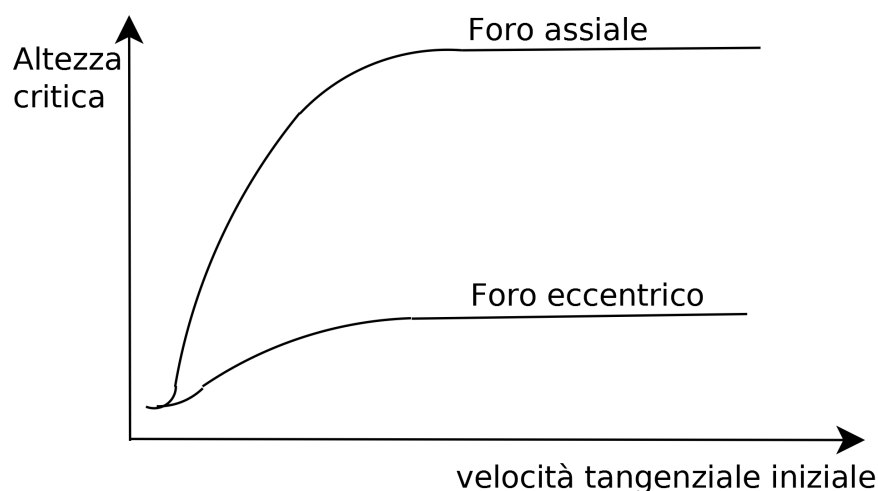
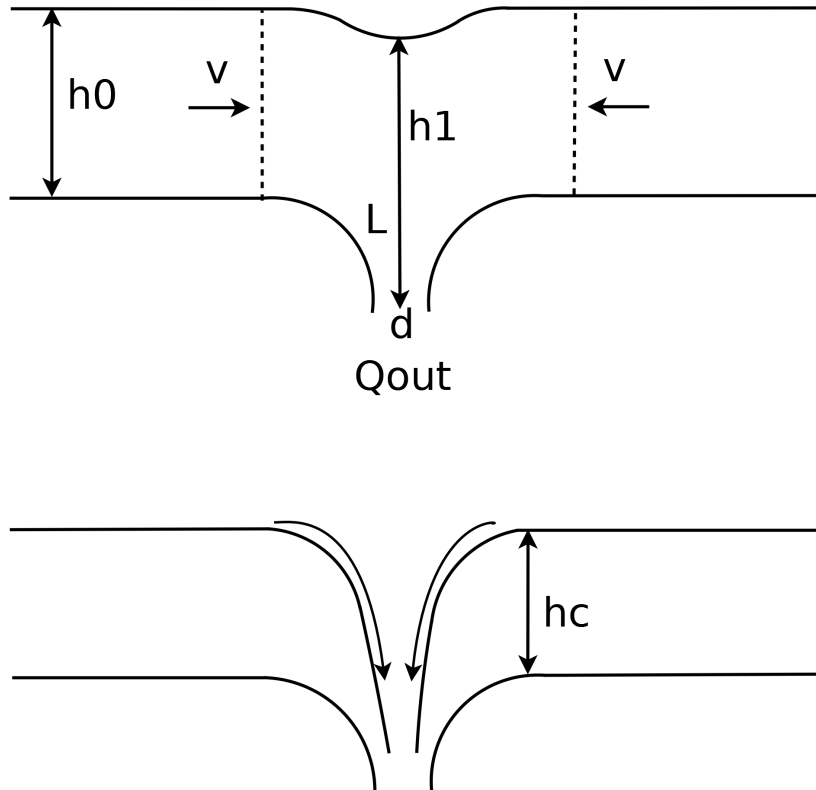


FIGURA 1.6: Altezza critica, H_{cr} , in funzione della velocità tangenziale iniziale $v_{\theta 0}$ al variare dell'eccentricità del foro di uscita.

foro di scarico, il flusso di uscita Q_{out} è alimentato dal fluido con velocità sostanzialmente parallela al fondo (figura 1.7). Quando questo flusso entrante non riesce più a compensare la richiesta potenzialmente possibile dallo scarico, dovuta al battente h_1 , la superficie si curva progressivamente fino a crollare. In letteratura questo fenomeno viene chiamato *drain-sink*. A seguito del collasso della superficie la scoria in superficie viene risucchiata nello scarico e passa nella panna. Gli studi effettuati [30] dimostrano che questo fenomeno è pressoché indipendente dalla posizione dello scarico, dall'altezza iniziale, dal diametro del contenitore e dalla sua geometria. In effetti l'altezza critica sembra essere una funzione diretta del diametro dello scarico. Anche la presenza di un'altra fase disposta sulla superficie libera dell'acciaio liquido aumenta la profondità di formazione del vortice. In letteratura sono riportati molti modelli analitici dedicati alla comprensione del fenomeno [31] [3] [34]. L'idea base consiste nell'applicare un volume di controllo in prossimità del livello critico in cui avviene il collasso della superficie. Esso può


 FIGURA 1.7: Meccanismo di *drain-sink*.

avere forma cilindrica o sferica ed inoltre può essere inclusa una fase aggiuntiva sopra la superficie del liquido. Uguagliando il flusso attraverso la superficie di volume con quello attraverso lo scarico di uscita si giunge ad una equazione che può essere risolta per un determinato valore H definito proprio come altezza critica. Il modello originale di *Lubin* e *Springer* definisce l'altezza critica secondo questa espressione,

$$H_{cr} = \left(\frac{Q_{cr}^2}{2\pi^2 g \left(1 - \frac{\rho_2}{\rho_1}\right)} \right), \quad (1.8)$$

dove Q_{cr} rappresenta la portata critica, ρ_1 e ρ_2 rispettivamente le densità della fase liquida e della fase galleggiante. L'idea di fondo di questi modelli analitici consiste nell'assumere che la velocità di ingresso nel volume di controllo sia normale alla sua superficie e numericamente uguale alla velocità che si ottiene applicando il teorema di Bernoulli tra un punto sulla superficie della sfera ed uno all'uscita dell'orifizio. La pressione in alto è data dalla pressione statica creata dalla fase a densità inferiore. Il flusso entrante quindi viene calcolato come il prodotto tra l'area della sfera e la velocità calcolata con Bernoulli. Chiaramente questo valore è zero sia quando il raggio della sfera coincide con l'altezza del bagno (H) sia quando tende a zero, pertanto esisterà una soluzione intermedia dell'equazione (1.8). Al di sotto di un H critico però non sarà possibile trovare un valore che soddisfi la (1.8), quindi *Lubin* e *Springer* definiscono altezza critica il valore tale che $dH/dr = 0$. La velocità ipotizzata applicando Bernoulli sarà

più alta di quella reale che entra nel volume di controllo, inoltre al scendere di H la superficie sferica diventa meno appropriata. In alternativa può essere costruito un modello con superficie cilindrica [3], anche se in questo modo vengono introdotti degli spigoli vivi. In ogni caso si tratta di modelli semi-quantitativi con lo scopo principale di comprendere qualitativamente il contributo delle variabili fisiche coinvolte. Un'ipotesi comune a tutti è che si tratta di fluidi newtoniani non viscosi, poiché si ritiene ([49]) che sia più influente il rapporto ρ_2/ρ_1 rispetto a quello μ_2/μ_1 . La scoria è realizzata unendo insieme principalmente CaO, Al₂O₃ e SiO₂. Le concentrazioni relative tra i vari ossidi ne determinano le proprietà. Generalmente la densità è dell'ordine di 2.5 kg/m³ mentre la viscosità può variare notevolmente passando da 3.5 P a 200 P. Considerando che la densità dell'acciaio liquido è pari a 7000 kg/m³, mentre la viscosità è dell'ordine di 0.07 P, otteniamo i seguenti dati:

- $\rho_2/\rho_1 \sim 0.3$;
- $\mu_2/\mu_1 = 50 - 3000$.

Generalmente vengono utilizzati dei modelli ad acqua per simulare l'acciaio liquido, mentre per la scoria sono stati usati dei silicani o n-pentano. Se eseguiamo ora la stessa analisi con queste sostanze otteniamo,

- $\rho_2/\rho_1 = 0.6 - 0.96$;
- $\mu_2/\mu_1 = 1 - 100$.

Altri modelli introducono direttamente la tensione superficiale, ad esempio *Koyola* [31] ricava con la stessa tecnica, nel caso di liquido-gas, la seguente espressione:

$$\frac{4}{3} \left[\left(H + \frac{2\sigma}{\rho g r} \right)^{\frac{3}{2}} - \left(\frac{2\sigma}{\rho g r} \right)^{\frac{3}{2}} \right] = r \sqrt{H + l}, \quad (1.9)$$

con r raggio dell'orifizio di uscita, l lunghezza del tuffante e σ tensione superficiale. Se consideriamo due sistemi aria-gas e acciaio-argon, osserviamo che le rispettive tensioni sono, $\sigma_1 = 0.072$ N/m e $\sigma_2 = 1.85$ N/m, quindi apparentemente ci si deve attendere un contributo significativo all'altezza critica. Nella realtà industriale l'acciaio è ricoperto da scoria e misure dirette (ad esempio [41]), mostrano che il campo di variabilità di σ è compreso tra 0.6 e 2.5 N/m. Applicando la (1.9) si ricava che data una σ l'altezza critica aumenta all'aumentare della densità del liquido, questo accade perché le forze di superficie devono sostenere una maggiore pressione quando il liquido ha un'elevata densità poiché la sua energia cinetica è maggiore. Il raggio dello scarico di uscita invece riduce l'effetto di σ poiché la forze superficiale agendo sul bordo della deformazione della superficie cresce con r , mentre quella dovuta alla pressione cresce con r^2 . Tuttavia a parità di tensione superficiale l'altezza critica diminuisce al decrescere r e per valori

σ non elevati l'andamento è lineare. Risulta quindi chiaro che i risultati analitici e sperimentali, ottenuti in laboratorio, devono essere considerati indicativi ma ugualmente fondamentali per circoscrivere accuratamente il problema.

Infine, secondo *Mazzaferro* [39], le condizioni industriali tipiche non consentono la formazione di una velocità tangenziale sufficiente da creare un vortice, pertanto il processo potrebbe essere esclusivamente governato dal fenomeno *drain-sink*. Su questo punto tuttavia non c'è accordo unanime. Egli inoltre sostiene che anche in presenza di vortice comunque la parte finale dello svuotamento rimane governata dal collasso della superficie senza vortice. Osservando la figura

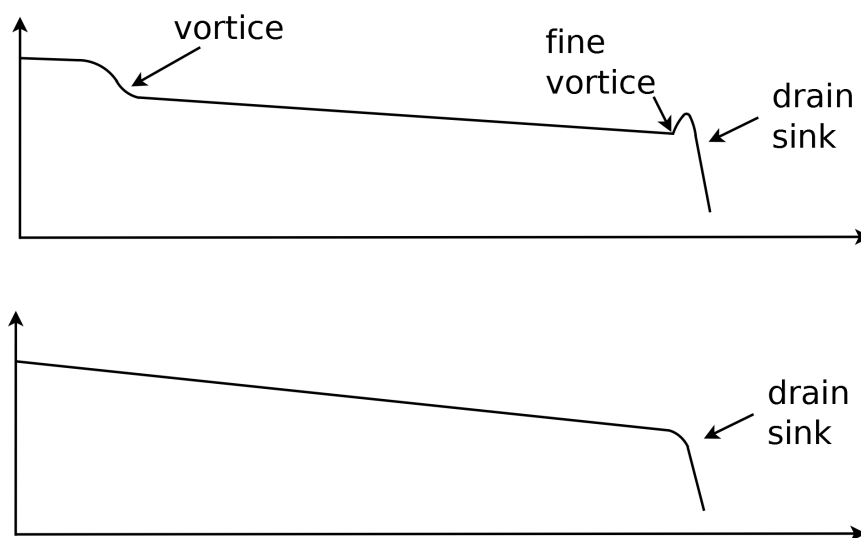


FIGURA 1.8: Meccanismo di svuotamento secondo G. Mazzaferro.

1.8 anche se c'è sufficiente velocità tangenziale con conseguente formazione del vortice, in prossimità della fine la circolazione non è sufficiente a sostenere il vortice stesso e per un breve istante la superficie ritorna piatta ed il foro di scarico nuovamente si riempie. Ciò permette un istantaneo aumento della portata, con successivo innesco del fenomeno di *drain-sink*.

Evitare il passaggio di scoria rappresenta quindi un problema che i produttori di acciaio hanno tentato di affrontare in vari modi. Poiché i modelli suggeriscono che l'altezza critica viene ridotta riducendo il diametro di uscita, si potrebbe suggerire che per ridurre il problema del passaggio scoria sia sufficiente parzializzare lo scarico durante la fase terminale del processo di svuotamento. Effettivamente questo può essere fatto ma non risulta una soluzione praticabile perché il peso in paniera deve rimanere entro determinati valori altrimenti non c'è sufficiente acciaio per alimentare le linee di colata. Una tale condizione imporrebbe quanto meno la riduzione della velocità di colata, ma non eliminerebbe il rischio di cadute di temperature con il probabile congelamento degli scarichi della paniera. Inoltre un basso livello in paniera riduce la rimozione delle inclusioni ad opera dello strato di scoria presente in superficie ed inoltre aumenta drammaticamente la probabilità di trasporto di ossidi altofondenti in lingottiera con conseguente rischio di *breakout*, ovvero di rottura della superficie solidificata del prodotto colato e successivo

svuotamento del cuore liquido. Quindi metodi per risolvere questo problema sono fortemente ricercati. Allo stato attuale **non esistono metodi consolidati ed universalmente accettati**. Da un punto di vista operativo l'operazione di chiusura della siviera viene demandata all'operatore, che attraverso l'uso contemporaneo di più sensi riesce ad identificare il momento di passaggio della scoria. Generalmente poiché la scoria ha una densità inferiore a quella dell'acciaio quando essa travasa nella paniera è immediato osservarla emergere in superficie in modo repentino e violento. I costruttori di siviere stanno cercando di introdurre dei profili della superficie di base tali da minimizzare l'altezza critica e ridurre la probabilità di formazione di vortici. Lo studio di queste soluzioni al momento non ha portato a risultati definitivi. L'altra strada seguita consiste nell'utilizzo di opportuni sensori nel tentativo di identificare il passaggio della scoria. Allo stato attuale l'unica soluzione disponibile a livello industriale si basa sul principio di induzione elettromagnetica [27]. Due spire conduttive, posizionate sul fondo siviera, vengono poste assialmente al foro di scarico. La variazione di conducibilità elettrica, indotta dal passaggio scoria, produce una variazione del campo elettromagnetico indotto dalla prima spira e ricevuto dalla seconda. Oltre alle problematiche intrinseche di identificazione di cui soffre anche l'utilizzo di questo principio di misura, esiste tuttavia una difficoltà critica legata all'installazione. Le spire infatti devono essere protette e poste in un ambiente estremo, questa difficoltà rende il dispositivo estremamente costoso e poco affidabile in termini di resistenza ai danneggiamenti. Un'altra tecnologia che è stata proposta riguarda l'uso di un sensore ad infrarossi per determinare eventuali variazioni di temperatura dovute al passaggio della scoria. Essa tuttavia non si è dimostrata affidabile e ripetibile (figura

Un'altra tecnica di rilievo consiste nell'uso di accelerometri posti sul braccio porta-tuffante [53]. Come si può osservare dalla figura 1.9 il flusso di acciaio proveniente dall'apertura inferiore della siviera verso la paniera sottostante non è a diretto contatto con l'atmosfera. Infatti per proteggere il getto dall'ossidazione viene inserito un tuffante (*ladle shroud*, figura 1.10), ossia un tubo ceramico. Esso viene messo dopo il posizionamento della siviera sulla torretta di sostegno e deve essere rimosso alla fine dello svuotamento. Il braccio porta-tuffante assolve proprio a questa funzione e può essere realizzato in due modalità. La prima, molto diffusa, consiste in un'asta di acciaio con un perno posizionato circa a metà. All'estremità opposta del tuffante ci sono dei pesi che spingono il tuffante verso il foro di scarico della siviera dotato di un ugello che si inserisce nel tubo di protezione (figura 1.11). L'altra modalità utilizza invece un braccio pneumatico (figura 1.12), in questo modo l'operatore esegue le operazioni di aggancio e sgancio in modo più agevole. In entrambi i casi su queste aste viene installato un accelerometro che rileva le vibrazioni dell'asta dovute al passaggio del getto di acciaio, il quale entrando nel tuffante lo scuote in quantità proporzionale alla velocità di transito. In prossimità del completo svuotamento, nel caso di ingresso scoria, l'intensità della vibrazione diminuisce. Sicuramente la densità inferiore della scoria contribuisce a rendere meno vigorosi gli sforzi sul tuffante. Questo evento finale si sviluppa in massimo due secondi ed entro questo intervallo il foro di uscita deve essere



FIGURA 1.9: Asta porta-tuffante.

chiuso altrimenti si ha un travaso eccessivo di scoria tra siviera e paniera. Il tubo si riempie rapidamente di scoria abbassando drammaticamente il livello di vibrazione, quindi è importante riuscire ad identificare la transizione sin dai primi istanti.

1.3 Caratterizzazione fluidodinamica del fenomeno

La simulazione dei flussi interni alla siviera, alla paniera ed alla lingottiera possono essere simulati convenientemente con modelli ad acqua [42]. Questa tecnica rappresenta un modo conveniente per investigare i fenomeni fluidodinamici che avvengono alle alte temperature. In funzione del tipo di analisi che si vuol fare possono essere utilizzati sia i modelli non isotermi che isotermi. Prendendo in considerazione questo ultimo caso che rappresenta la condizione più generale, possiamo scrivere le seguenti equazioni di *Navier-Stokes* nel caso di fluidi newtoniani (1.10),

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i v_i)}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\mu_{eff} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] - \frac{\partial p}{\partial x_i} - \rho_0 \beta \Delta T g_i, \quad (1.10)$$

con $i = 1 \dots 3$, $j = 1 \dots 3$, utilizzando l'approssimazione di *Boussineq* per le forze di galleggiamento. Chiaramente nel caso isoterma il termine additivo $\rho_0 \beta \Delta T g_i$ non compare. Alla (1.10) va aggiunta l'equazione di scambio termico (1.11).

$$\frac{\partial(\rho CT)}{\partial t} + \frac{\partial(\rho u_j CT)}{\partial x_j} = \frac{\partial}{\partial x_j} \left(\lambda_{eff} \frac{\partial T}{\partial x_j} \right). \quad (1.11)$$



FIGURA 1.10: Tuffanti per protezione contro l'ossidazione dell'acciaio.

In accordo alla teoria della similitudine per sistemi geometricamente simili nei quali il flusso è regolato dalle stesse equazioni matematiche, due sistemi possono essere considerati *simili* sotto opportuni criteri noti come gruppi adimensionali [32]. Scrivendo in forma adimensionale le equazioni (1.10) e (1.11) otteniamo,

$$\frac{\partial(\rho^* u_i^*)}{\partial t^*} + \frac{\partial(\rho^* u_i^* v_i^*)}{\partial x_j^*} = \frac{\partial}{\partial x_j^*} \left[\frac{\mu_{eff}}{\rho_0 U L} \left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} \right) \right] - \frac{\partial p^*}{\partial x_i^*} - \frac{g_i L}{U^2} \beta \Delta T \quad (1.12)$$

e

$$\frac{\partial(\rho^* T^*)}{\partial t^*} + \frac{\partial(\rho^* u_j^* T^*)}{\partial x_j^*} = \frac{\partial}{\partial x_j^*} \left(\frac{\lambda_{eff}}{\rho_0 C U L} \frac{\partial T^*}{\partial x_j^*} \right). \quad (1.13)$$

Nelle (1.12) e (1.13) le variabili con asterisco assumono la seguente forma,



FIGURA 1.11: Tipico braccio porta tuffante, con contrappesi.



FIGURA 1.12: Braccio porta tuffante idraulico.

$$\begin{aligned}\rho^* &= \frac{\rho}{\rho_0}, \\ x^* &= \frac{x}{L}, \\ u^* &= \frac{u}{U}, \\ t^* &= \frac{Ut}{L}, \\ T^* &= \frac{T}{T_{ref}}, \\ p^* &= \frac{p - p_0}{\rho_0}.\end{aligned}$$

I parametri ρ , L , U , T_{ref} e p_0 sono valori di riferimento o caratteristici. Le equazioni (1.12) ed (1.13) sotto opportune condizioni adimensionali iniziali ed al contorno hanno le stesse soluzioni sia per il modello ad acqua che per quello con acciaio liquido. Tali condizioni sono

$$\left(\frac{\mu_{\text{eff}}}{\rho_0 U L} \right)_{\text{acqua}} = \left(\frac{\mu_{\text{eff}}}{\rho_0 U L} \right)_{\text{acciaio}}, \quad (1.14)$$

$$\left(\frac{\lambda_{\text{eff}}}{\rho_0 C U L} \right)_{\text{acqua}} = \left(\frac{\lambda_{\text{eff}}}{\rho_0 C U L} \right)_{\text{acciaio}}, \quad (1.15)$$

$$\left(\frac{g_i}{L U^2} \beta \Delta T \right)_{\text{acqua}} = \left(\frac{g_i}{L U^2} \beta \Delta T \right)_{\text{acciaio}}. \quad (1.16)$$

Se le similitudini (1.14), (1.15) e (1.16) vengono rispettate allora i due sistemi sono equivalenti. Notiamo che le espressioni (1.14) e (1.15) rappresentano rispettivamente il numero di Reynolds e di Peclet. Tuttavia nel caso specifico le condizioni (1.14) e (1.15) non possono essere rispettate simultaneamente, infatti se $\text{Re}_{\text{acqua}} = \text{Re}_{\text{acciaio}}$ allora otteniamo che $(UL)_{\text{acqua}} \approx (UL)_{\text{acciaio}}$ essendo infatti le viscosità cinematiche dell'acciaio e dell'acqua simili ($\nu_{\text{acqua}} = 1 \text{ m}^2/\text{s}$ e $\nu_{\text{acciaio}} = 0.8 \text{ m}^2/\text{s}$ a 1600°C). Negli impianti industriali le misure di temperatura in siviera e paniera sono una pratica quotidiana, la temperatura infatti è un parametro critico che deve essere costantemente monitorato (es. [22]). Particolare importanza riveste il valore del *superheat*, ovvero l'eccedenza rispetto alla temperatura di solidus dell'acciaio colato. Valori bassi portano al rischio di congelamento delle linee, valori alti invece aumentano la probabilità di *breakouts*. I modelli isotermi quindi sono importanti quando si vuole studiare gli effetti termici durante le fasi di svuotamento della siviera. Tuttavia *Morales* [16] riporta che la presenza di gradienti termici può influenzare l'altezza critica, ma per gli scopi relativi all'identificazione del passaggio scoria i modelli non isotermi non risultano essere fondamentali ed introducono solo una complicazione nei calcoli o nelle simulazioni in laboratorio. L'equazione (1.16) diventa quindi l'unica relazione utilizzabile [44] [15] e rappresenta l'effetto dei termini di galleggiamento. E' tuttavia conveniente reinterpretare l'espressione al modo seguente,

$$\frac{gL}{U^2} \beta \Delta T = \frac{\beta \Delta T}{Fr},$$

con Fr numero di Froude,

$$Fr = \frac{U^2}{gL}. \quad (1.17)$$

In base alle precedenti definizioni possiamo riscrivere la similitudine in questa forma,

$$\left(\frac{\beta \Delta T}{Fr} \right)_{\text{acqua}} = \left(\frac{\beta \Delta T}{Fr} \right)_{\text{acciaio}}, \quad (1.18)$$

e dalla (1.18) otteniamo,

$$Fr_{\text{acqua}} = Fr_{\text{acciaio}}, \quad (1.19)$$

$$(\beta\Delta T)_{\text{acqua}} = (\beta\Delta T)_{\text{acciaio}}, \quad (1.20)$$

Il numero di *Froude* risulta essere quindi un ottimo candidato per applicare i criteri della similitudine. A questo risultato si può pervenire anche attraverso percorsi più empirici, infatti l'altezza critica, considerando sia il caso *funnel vortex* che *drain sink*, sperimentalmente risulta essere potenzialmente una funzione di,

- velocità tangenziale iniziale (v_{θ_i});
- altezza iniziale (H_i);
- diametro dello scarico (d);
- diametro contenitore (D);
- densità del fluido (ρ);
- viscosità del fluido (μ);
- gravità (g).

Applicando quindi il teorema di *Buckingham* è possibile costruire il seguente gruppo di numeri adimensionali,

- $\pi_1 = \frac{H_{CR}}{H_i}$;
- $\pi_2 = \frac{d}{D}$;
- $\pi_3 = \frac{v_{\theta_i}}{\sqrt{gH_i}} \simeq \frac{v_{\theta_i}}{v_{out_i}}$, numero di *Froude*;
- $\pi_4 = \frac{\rho H_i \sqrt{gH_i}}{\mu} \simeq \frac{\rho v_{out_i} H_i}{\mu}$, numero di *Reynolds*;

Infine si può applicare un approccio ancora più elementare [54]: le cause che governano lo svuotamento di contenitori di liquido e la conseguente formazioni di vortici sono la forza gravitazionale, la tensione superficiale, gli effetti inerziali e la viscosità. Secondo quanto riportato da *Ramamurthi* [54] gli unici contributi vengono dati dalla gravità e dall'inerzia. In particolare l'inerzia può essere sia assiale che gravitazionale da cui,

- $Fr_{\text{assiale}} = \frac{v_{\text{assiale}}}{\sqrt{gD}}$;
- $Fr_{\text{rotazionale}} = \frac{v_{\text{angolare}}}{\sqrt{gH}}$.

Ma l'unico parametro critico per esprimere gli effetti rotazionali è evidentemente H , quindi il numero di *Froude* rotazionale è il criterio da considerare.

Dalla conservazione del numero di *Froude* possiamo ricavare la legge di scala temporale,

$$\left(\frac{U_{\text{acqua}} t_{\text{acqua}}}{L_{\text{acqua}}} \right) = \left(\frac{U_{\text{acciaio}} t_{\text{acciaio}}}{L_{\text{acciaio}}} \right)$$

$$\frac{t_{\text{acqua}}}{t_{\text{acciaio}}} = \sqrt{\frac{L_{\text{acqua}}}{L_{\text{acciaio}}}}$$

Per quanto riguarda invece il fattore di scala geometrico, la sola condizione sul numero di *Froude* non impone particolari restrizioni.

NOTA SULLE FORZE DI CORIOLIS

La formazione di vortici dovute alla forza di Coriolis è stata studiata sperimentalmente da vari ricercatori [50] [2] [55] e su piccola scala l'effetto è considerevolmente piccolo. Come è noto l'accelerazione di Coriolis è data da

$$a_c = 2\boldsymbol{\Omega} \wedge \mathbf{v},$$

con \mathbf{v} velocità del corpo in movimento rispetto al sistema di riferimento terrestre. Facendo

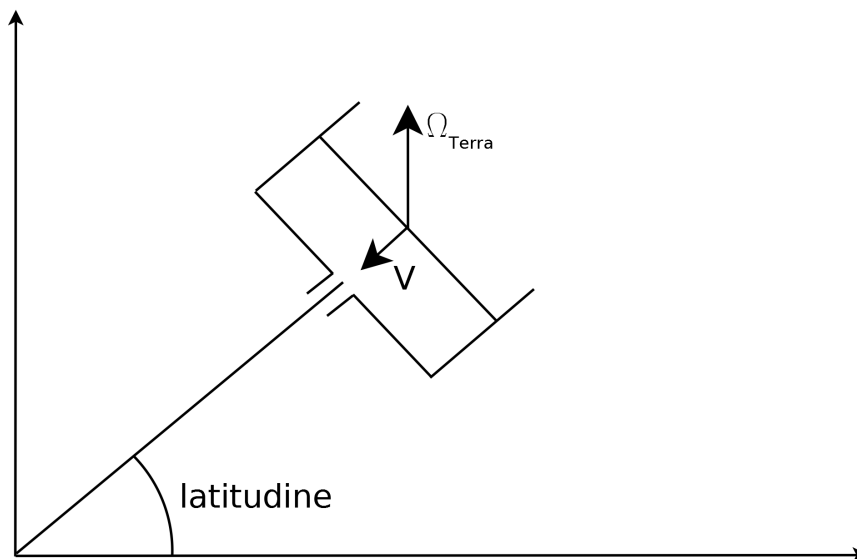


FIGURA 1.13: accelerazione di Coriolis.

riferimento alla figura 1.13, supponiamo una velocità di svuotamento di 15 cm/minuto per un recipiente cilindrico di dimensioni contenute alla latitudine del nord Italia. In queste condizioni

il rapporto con l'accelerazione di gravità risulta essere molto piccolo, infatti

$$\frac{a_c}{g} \sim 10^{-8}.$$

Da questa semplice osservazione si può dedurre che l'effetto seppur presente risulta facilmente mascherato da altre fonti di disturbo. In effetti, come evidenziano gli studi effettuati, le condizioni sperimentali devono essere particolarmente curate, in quanto anche il moto convettivo dovuto alla stratificazione termica all'interno del recipiente può alterare il risultato.

1.4 Obiettivi dell'attività di ricerca

I principali obiettivi della presente attività di ricerca sono due:

- valutare le tecniche avanzate di analisi del segnale per individuare l'evento catastrofico corrispondente al passaggio della scoria in siviera;
- applicare gli algoritmi in un sistema hardware e software real time.

Su questo argomento tuttavia la letteratura è praticamente inesistente e quindi la descrizione fisica del fenomeno è carente. Inoltre il processo reale avviene con acciaio liquido e non è possibile cambiare alcun parametro operativo o strutturale nel processo. In particolare l'operatore umano o il controllore automatico interagiscono costantemente regolando il flusso di acciaio e questo chiaramente cambia le condizioni operative in modo non riproducibile, inoltre la decisione di interrompere il processo avviene a discrezione dell'uomo il quale quindi ricorrendo alla propria esperienza decide di chiudere il cassetto della siviera. Poiché i test in impianto non possono essere numerosi, quest'ultimo fatto complica notevolmente il problema. Alla luce di tutti questi vincoli risulta evidente come possa rivelarsi arduo valutare efficacemente la validità di una tecnica di analisi. Un considerevole contributo alla risoluzione di queste problematiche può venire dall'utilizzo di un modello in scala ad acqua. La disponibilità di un simile prototipo permette di eseguire un numero elevato di esperimenti. I vantaggi tuttavia sono molteplici, essendo carente la descrizione fisica del problema, con questo modello si possono variare sia i parametri operativi che strutturali al fine di comprendere quali possono essere le variabili più importanti. Naturalmente l'utilizzo del modello ad acqua presuppone pur sempre una approssimazione della realtà. L'utilizzo di metodi di simulazione fluidodinamica permette di studiare quello che avviene all'interno della siviera. In quest'ottica il prototipo in scala serve anche per tarare i modelli di calcolo ed in particolare le approssimazioni geometriche e le condizioni al contorno. Da un punto di vista più ampio quindi gli obiettivi della ricerca riguardano non solo l'applicabilità in tempo reale delle tecniche avanzate di analisi del segnale, ma anche la comprensione fisica del fenomeno in modo da interpretare correttamente il segnale registrato dall'accelerometro. In

questo modo l'attività di ricerca deve dimostrare se l'approccio che ho delineato possa essere considerato valido per poterlo impiegare in analisi successive più focalizzate. Ad esempio sarà possibile studiare configurazioni di trasduttori alternative al fine di migliorare la capacità predittiva degli algoritmi. Per quanto riguarda l'attività di simulazione fluidodinamica, l'idea di fondo è che se queste tecniche si dimostrano efficaci allora si può ipotizzare di studiare qualche configurazione modificata della siviera in modo da poter indurre un segnale di identificazione più intenso. Nel dettaglio il programma di ricerca sarà composto dai seguenti punti:

- selezione delle tecniche di analisi da applicare;
- sviluppo degli algoritmi di calcolo in tempo reale;
- implementazione degli algoritmi in C++;
- sviluppo di una architettura hardware funzionante in tempo reale usabile sia in laboratorio che in campo;
- sviluppo di un sistema in scala con acqua;
- test in campo;
- sviluppo di un modello fluidodinamico tridimensionale in transitorio e con superficie libera per compararlo con il prototipo in acqua.

La selezione delle tecniche di analisi tiene in considerazione il fatto che il fenomeno è transitorio. Per questo motivo si è deciso di considerare le trasformate *Wavelet* e Hilbert. Essendo il segnale basato su elementi risonanti dovuti alla particolare struttura meccanica su cui è fissato l'accelerometro, si è considerata la trasformata *Wavelet Packets*, la quale offre la possibilità di costruire un banco di filtri aventi la caratteristica di conservare l'energia. La medesima caratteristica del segnale suggerisce anche l'impiego dell'attributo frequenza istantanea: è ragionevole infatti attendersi che modifiche delle strutture risonanti possano essere efficacemente intercettate da questo attributo. Le tecniche classiche, essenzialmente basate sull'idea di spettrogramma, sono ugualmente impiegate; esse sicuramente sono un importante termine di confronto e permettono certamente di integrare l'osservazione e l'analisi del segnale evidenziandone caratteristiche diverse. L'accelerometro sarà connesso ad una unità di acquisizioni dati ethernet che a sua volta comunicherà con un computer. Questa soluzione offre sufficiente versatilità pur mantenendo i requisiti di un sistema in tempo reale. Per quanto riguarda il software invece si è deciso di scegliere il linguaggio C++ poiché in futuro gli algoritmi sviluppati potranno essere direttamente implementati in un sistema dedicato come un DSP. I test in campo saranno condotti in alcuni impianti che hanno dato la disponibilità ad installare l'attrezzatura, che sebbene sperimentale deve comunque basarsi su soluzioni industriali sufficientemente robuste visto l'ambiente aggressivo.

I test come anticipato sono di breve durata e non possono interagire con il processo. L'obiettivo della ricerca chiaramente non è quello di riprodurre il miglior prototipo in scala possibile, l'idea di fondo è di simulare alcune caratteristiche per poter eseguire un numero sufficiente di misure con cui valutare le caratteristiche degli algoritmi. Per questo motivo la scala geometrica viene spinta a rapporti elevati, dell'ordine di 1:15. Le simulazioni fluidodinamiche precedono l'utilizzo del software *openFoam* per la sua facilità d'uso e per la scalabilità in ambienti HPC. E' chiaro quindi che la ricerca intrapresa rappresenta il primo fondamentale passo verso questo più ampio obiettivo. Dimostrare la reale fattibilità del metodo garantirà la possibilità di studiare in modo più dettagliato singoli aspetti di questo complesso argomento.

Capitolo 2

Tecniche classiche di stima dello spettro di potenza

In questo capitolo si riporta una breve introduzione teorica dei metodi classici per la stima della potenza del segnale. La letteratura su questo argomento è ovviamente estremamente vasta, ma utili riferimenti si possono trovare ad esempio in [20] [21] [52] [57] [58] [58] [64]. L'introduzione iniziale ha lo scopo di porre il problema secondo una prospettiva storica che aiuta a comprendere la sequenza di passaggi chiave che hanno permesso lo sviluppo di questo settore dell'analisi del segnale. I metodi esposti sono stati implementati nel software *real-time* descritto nel capitolo 5. Essi inoltre sono stati utilizzati anche nei capitoli 8 e 7 per l'analisi dei dati raccolti sia dal dispositivo sperimentale in laboratorio sia per quelli provenienti direttamente dal campo.

2.1 Introduzione storica

Lo spettro di potenza rivela l'esistenza di schemi ripetitivi e di strutture di correlazione presenti in un processo [48]. Il metodo più comune per la stima spettrale si basa sulla trasformata di Fourier e per molte applicazioni produce risultati sufficientemente buoni. Da un punto di vista generale le tecniche di stima si dividono in classiche o non parametriche e moderne o parametriche. Chiaramente le seconde offrono migliori prestazioni se la sequenza di dati rispetta il modello assunto, altrimenti i metodi non parametrici risultano migliori. Le origini dell'analisi spettrale possono essere ricondotte a Newton (1643-1727), ma è sicuramente con Jean Baptiste Joseph de Fourier (1768-1830) che questa branca della scienze riceve un primo impulso fondamentale. Egli affermò che una funzione arbitraria definita su un intervallo finito può essere rappresentata da una somma infinita di seni e coseni. La comunità matematica del tempo reagì rigettando l'idea poiché riteneva che solo le funzioni analitiche potessero essere rappresentate

in questo modo in quanto infinitamente differenziabili. Sul finire del Diciannovesimo secolo sir Arthur Schuster (1851-1934) introdusse il concetto di periodogramma $P(\omega)$ definito come,

$$P(\omega) = \frac{1}{N} |x(1)e^{-j\omega} + x(2)e^{-j\omega^2} + \dots + x(N)e^{-j\omega N}|^2. \quad (2.1)$$

Fino al lavoro di Yule (1871-1951) nel 1927, il metodo di Schuster era l'unico approccio all'analisi dello spettro di potenza. Molte serie temporali osservate in natura tuttavia non mostravano picchi dominanti. Questo portò Yule a sviluppare il primo metodo basata su un modello parametrico finito noto come processo autoregressivo (AR). Nel 1930 Norbert Wiener (1894-1964) pubblicò un articolo fondamentale, *Generalized Harmonic Analysis* [62][63], in cui definì in modo rigoroso le relazioni tra lo spettro di potenza e l'autocovarianza. Per gli studiosi della prima parte del Ventesimo secolo, il periodogramma di Schuster era l'unico metodo disponibile e nel caso di processi stazionari composti da sinusoidi sovrapposte a rumore bianco, esso risulta molto efficace. Tuttavia per un processo stazionario non deterministico esso si rivela molto grezzo e di difficile interpretazione. Molte serie temporali osservate in natura non potevano essere analizzate con questa tecnica. Fu Wiener a gettare le fondamenta per applicare la trasformata di Fourier di un processo caotico stazionario. Egli introdusse il concetto di processo stocastico continuo caratterizzando la variabile aleatoria W_t con le seguenti tre proprietà:

1. $W_0 = 0$;
2. W_t è sicuramente continua;
3. W_t ha incrementi indipendenti tali che $W_t - W_s$ sono normalmente distribuiti.

Questo processo si dice anche martingale e markoviano. Un'ulteriore caratterizzazione di W_t si basa sul fatto che la sua rappresentazione spettrale consiste in una serie di seni i cui coefficienti sono delle variabili normali indipendenti. Un esempio di processo di Wiener è il rumore bianco. In molti processi naturali ed industriali l'ingresso non è predicibile; posto pertanto $x(n)$ l'output di un sistema avente $b(n)$ come risposta all'impulso e utilizzando il concetto di processo stocastico si può scrivere,

$$x(n) = \sum_{k=0}^{\infty} b(k)\epsilon(n-k), \quad (2.2)$$

mentre nel dominio delle frequenze si ha,

$$X(\omega) = B(\omega)E(\omega), \quad (2.3)$$

con $E(\omega)$ trasformata di Fourier del rumore bianco $\epsilon(n)$. Essendo $E(\omega)$ molto irregolare e $B(\omega)$ al contrario regolare, risulta che $X(\omega)$ non avrà un comportamento ben definito in frequenza. Dato quindi

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega)e^{j\omega n} d\omega, \quad (2.4)$$

l'espressione

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} B(\omega)E(\omega)e^{j\omega n} d\omega, \quad (2.5)$$

rappresenta proprio l'analisi armonica generalizzata di Wiener ed esprime la rappresentazione spettrale di un processo stazionario non deterministico. Applicando il periodogramma di Schuster (2.1) si comprende immediatamente dalla (2.3) che vale

$$P(\omega) = \frac{1}{N} |X(\omega)|^2. \quad (2.6)$$

Si trova quindi una chiara giustificazione teorica alla causa della sua irregolarità. Wiener in alternativa per serie molto lunghe propose il calcolo dell'autocovarianza, $\gamma_{xx}(\tau)$ (2.7), e definì la densità spettrale di energia del segnale come la sua trasformata di Fourier (2.8).

$$\gamma_{xx}(t) = \int_{-\infty}^{\infty} x^*(\tau)x(t + \tau)d\tau \quad (2.7)$$

$$\gamma_{xx}(\omega) = \int_{-\infty}^{\infty} \gamma_{xx}(t)e^{-j\omega t} dt \quad (2.8)$$

$$\gamma_{xx}(t) = \int_{-\infty}^{\infty} |X(\omega)|^2 e^{-j\omega t} d\omega \quad (2.9)$$

L'espressione (2.9) prende attualmente il nome di teorema di Wiener-Khinchin. Il contributo essenziale di questa ricerca è che anche se la trasformata di Fourier di un segnale non esiste tuttavia se il processo è stazionario in senso lato allora è comunque calcolabile la decomposizione spettrale dell'autocorrelazione. Ad esempio in un sistema lineare tempo-invariante con input ed output non quadrato-integrabili, si può comunque affermare che la trasformata di Fourier dell'autocorrelazione dell'output è il prodotto di quella dell'input con la trasformata di Fourier della risposta all'impulso del sistema. Questo risultato sta alla base dei metodi parametri di stima della potenza. Il problema della stima si pone tuttavia in modo stringente quando effettivamente si dispone di una sequenza di lunghezza ridotta. Nel 1949 Tukey (1915-2000) [56] diede un impulso decisivo con i suoi studi sull'approccio statistico al problema. Un processo può essere diviso in:

1. *noise-like*;
2. *signal-like*;
3. *signal plus noise-like*.

Il secondo caso sebbene irrealizzabile costituisce comunque un'utile idealizzazione dell'ultimo caso. E' chiaro comunque che il primo processo produrrà delle serie temporali molto diverse da quelle del caso 3). Il punto cruciale è che la regolarità di una serie *noise-like* non sta nelle

forme del segnale ma nella struttura statistica che la caratterizza. Chiaramente in queste condizioni l'autocovarianza delle realizzazioni di un segmento non sarà uguale a quella di un altro. Brillinger e Tukey sottolinearono che spesso si hanno solo distinte realizzazioni con le quali realizzare valide ipotesi sulla popolazione sottostante. Quindi è assolutamente necessario pensare in termini statistici. Nel 1967 Burg [12] diede un altro contributo fondamentale. Infatti i metodi convenzionali per il calcolo dello spettro di potenza assumono che l'autocovarianza sia nulla per ritardi sufficientemente grandi. Effettivamente dato un limitato insieme di sequenze di autocovarianza con la sola ipotesi che $\gamma_{xx}(\omega) \geq 0$ risulta che un numero infinito di spettri di potenza siano in accordo con questa informazione. Sono quindi necessarie altre ipotesi. Burg nel tentativo di risolvere questo problema introdusse il concetto di entropia. L'analisi spettrale basata sulla massima entropia richiede che la serie più caotica e non predicibile abbia autocovarianza coincidente con un determinato insieme di realizzazioni. Dopo questi lavori la stima spettrale si è continuamente arricchita di nuovi metodi e strategie.

2.2 Definizione di ESD e PSD per segnali discreti

Sia $x(n)$ una sequenza discreta e deterministica a media nulla e ipotizziamo che valga la (2.10) allora possiamo assumere che esista la DTFT di $x(n)$ (2.11) [4] [58] [60][52].

$$\sum_{n=-\infty}^{+\infty} |x(n)|^2 < \infty \quad (2.10)$$

$$X(\omega) = \sum_{n=-\infty}^{+\infty} x(n)e^{-i\omega n} \quad (2.11)$$

Sotto queste ipotesi la densità di energia spettrale risulta essere semplicemente,

$$E(\omega) = |X(\omega)|^2 \quad (2.12)$$

Inoltre definita l'autocorrelazione come

$$r_x(k) = \sum_{n=-\infty}^{+\infty} x(n)x(n-k) \quad (2.13)$$

allora mediante la sostituzione della (2.11) nella trasformata di Fourier della (2.13) otteniamo,

$$\begin{aligned}
 R(\omega) &= \sum_{k=-\infty}^{+\infty} x(k)e^{-i\omega k} \\
 &= \sum_{k=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} x(n)x^*(n-k)e^{-i\omega n}e^{-i\omega(n-k)} \\
 &= |X(\omega)|^2 \\
 &= E(\omega)
 \end{aligned} \tag{2.14}$$

Molti segnali che si incontrano nelle applicazioni reali al pari delle serie naturali sono tali che in seguito alle loro variazioni stocastiche non è possibile prevedere una realizzazione futura con certezza. Poiché in generale si può osservare solo una realizzazione del segnale si potrebbe pensare che per esso esista la DTFT. Per un processo stazionario caotico tuttavia la (2.12) non è garantita ed in generale si può parlare di potenza media finita che può essere caratterizzata dalla densità spettrale di potenza (PSD). Definendo quindi il segnale discreto $x(n)$ come una realizzazione di una variabile casuale avente valore medio nullo, si introduce la funzione di autocorrelazione come il valore atteso,

$$r_x(k) = E[x(n)x^*(n-k)], \tag{2.15}$$

e si assume che (2.15) dipenda solo dalla distanza tra due eventi e non dall'indice assoluto. Un processo si dice stazionario (o stazionario in senso stretto) se le probabilità congiunte non cambiano nel tempo, questo significa che tutti i momenti rimangono invariati. Si dice debolmente stazionario se almeno la media e la varianza rimangono indipendenti. Il teorema di Wiener-Khinchin afferma che per ogni processo debolmente stazionario esiste

$$R(\omega) = \int_{-\infty}^{+\infty} r(\tau)e^{-i\omega\tau} d\tau, \tag{2.16}$$

e può essere interpretata come la densità di potenza spettrale.

Per una sequenza discreta posto $g_k(m) = x_m x_{m-k}^*$ allora la stima di autocorrelazione consiste nella media di $g_k(m)$:

$$\hat{r}^{(1)}(k) = \frac{1}{N-k} \sum_{m=k+1}^N x(m)x^*(m-k) \quad 0 \leq k \leq N-1 \tag{2.17}$$

e

$$\hat{r}^{(2)}(k) = \frac{1}{N} \sum_{m=k+1}^N x(m)x^*(m-k) \quad 0 \leq k \leq N-1. \tag{2.18}$$

Lo stimatore (2.17) non è polarizzato mentre quello relativo alla definizione (2.18) lo è. Generalmente è utilizzata quest'ultima poiché quando l'autocorrelazione decade rapidamente la stima (2.17) tende ad essere molto variabile per grandi valori di k . Mentre per piccoli valori di k si assume che abbiamo un comportamento simile. Infatti,

$$\begin{aligned} E[\hat{r}_{xx}^{(2)}(k)] &= E\left[\frac{N-|k|}{N}\hat{r}_{xx}^{(1)}(k)\right] \\ &= \frac{N-|k|}{N}E[r_{xx}^{(1)}(k)] \\ &= \frac{N-|k|}{N}r_{xx}(k). \end{aligned} \quad (2.19)$$

In base a quanto introdotto si possono dare due definizioni di PSD per una serie discreta. La prima è

$$P(\omega) = \sum_{k=-\infty}^{+\infty} r_{xx}(k)e^{-i\omega k}, \quad (2.20)$$

mentre la seconda è

$$P(\omega) = \lim_{N \rightarrow \infty} E\left[\frac{1}{N}\left|\sum_{k=0}^{N-1} x(n)e^{i\omega n}\right|^2\right]. \quad (2.21)$$

Le due definizioni sono in effetti equivalenti,

$$\begin{aligned} &\lim_{N \rightarrow \infty} E\left[\frac{1}{N}\left|\sum_{k=0}^{N-1} x(n)e^{-i\omega n}\right|^2\right] = \\ &\frac{1}{N} \lim_{N \rightarrow \infty} E\left[\sum_{n=0}^{N-1} \sum_{s=0}^{N-1} x(n)x^*(s)e^{-i\omega(n-s)}\right] = \\ &\frac{1}{N} \lim_{N \rightarrow \infty} E\left[\sum_{n=0}^{N-1} \sum_{s=0}^{N-1} x(n)x^*(n-k)e^{-i\omega k}\right] = \\ &\lim_{N \rightarrow \infty} \sum_{k=-(N-1)}^{N-1} E\left[\frac{1}{N} \sum_{n=0}^{N-1-|k|} x(n)x^*(n-|k|)e^{-i\omega k}\right] = \\ &\lim_{N \rightarrow \infty} \sum_{k=-(N-1)}^{N-1} \left(1 - \frac{|k|}{N}\right) r_{xx}(k)e^{-i\omega k} = \sum_{k=-\infty}^{\infty} r_{xx}(k)e^{-i\omega k}. \end{aligned}$$

2.3 Periodogramma e Correlogramma

Il periodogramma introdotto da Shuster discende proprio dalla definizione (2.21).

$$\hat{P}_N^{(p)}(\omega) = \frac{1}{N}|X(\omega)|^2. \quad (2.22)$$

mentre il correlogramma discende dalla definizione (2.20),

$$\hat{P}_N^{(c)}(\omega) = \sum_{m=-(N-1)}^{N-1} r_{xx}(m)e^{-j\omega m}. \quad (2.23)$$

Interpretando nella (2.19) il termine $1 - |k|/N$, come una *finestra* e definendo,

$$w_B(k) = \begin{cases} \frac{N-|k|}{N}, & |k| < N \\ 0, & |k| \geq N, \end{cases} \quad (2.24)$$

si può reinterpretare il valore atteso dello spettro di potenza in questo modo,

$$E[\hat{P}(\omega)] = \sum_{k=-\infty}^{\infty} w_B(k)r_{xx}(k)e^{-i\omega k}. \quad (2.25)$$

L'espressione (2.25) rappresenta la DTFT del prodotto di due sequenze e quindi essa può anche essere scritta come la convoluzione delle rispettive trasformate di Fourier.

$$E[\hat{P}(\omega)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} R_{xx}(\psi)W_B(\omega - \psi)d\psi. \quad (2.26)$$

Un calcolo diretto dimostra che la DTFT di $w_B(r)$ è,

$$W_B(\omega) = \frac{1}{N} \left| \frac{\sin(\frac{\omega N}{2})}{\sin(\frac{\omega}{2})} \right|^2. \quad (2.27)$$

L'espressione (2.27) viene chiamata anche *Fejer kernel*. Definendo,

$$w_R(k) = \begin{cases} 1, & |k| < N \\ 0, & |k| \geq N. \end{cases} \quad (2.28)$$

allora dalla (2.20) si può in modo simile alla (2.26) definire il valore atteso dello spettro di potenza come,

$$E[\hat{P}(\omega)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} R_{xx}(\psi)W_R(\omega - \psi)d\psi. \quad (2.29)$$

con $W_R(\omega)$ *kernel di Dirichlet*,

$$W_R(\omega) = \frac{\sin[\omega(2N-1)/2]}{\sin[\omega/2]}, \quad (2.30)$$

In effetti affinché $E[\hat{P}(\omega)]$ coincida con $P(\omega)$ la funzione $W_B(\omega)$ dovrebbe approssimare la delta di Dirac. Il lobo principale ha una larghezza in frequenza pari a circa $1/N$ e quindi per valori contenuti di N non è certamente una buona approssimazione. L'effetto principale del lobo centrale è quello di smussare lo spettro stimato, infatti se due picchi sono distanti meno di $1/N$

allora non vengono risolti e per questo motivo questo valore viene proprio chiamato limite della risoluzione spettrale del periodogramma. I lobi laterali invece trasferiscono potenza dalle bande dove essa è concentrata a quelle dove ne è contenuta meno. Questo fenomeno viene appunto chiamato *leakage*. Da questa introduzione segue che gli aspetti negativi del periodogramma o correlogramma, sono particolarmente critici per spettri caratterizzati da picchi intensi. Nel caso invece di spettri più smussati il problema risulta più contenuto. In realtà il problema principale di questo metodo risiede nella varianza. In generale non è possibile ottenere un'espressione semplice per la varianza del periodogramma, tuttavia se si considera una sequenza campione di un processo reale, bianco e gaussiano allora si può giungere a questa espressione [4],

$$\text{var}[\hat{P}_N(\omega)] = \sigma_x^4 \left\{ 1 + \left(\frac{\sin[\omega N]}{N \sin[\omega]} \right)^2 \right\}. \quad (2.31)$$

Dalla formula (2.31) si comprende che la varianza non tende a zero quando N tende all'infinito anzi indipendentemente da N esso rimane dell'ordine di σ_x^4 . Il periodogramma quindi non è una stima consistente.

2.4 Metodo di Bartlett

Un metodo classico per ridurre la varianza delle stime è quello di fare la media di numerose stime indipendenti secondo quanto suggerito da Bartlett [6]. Una sequenza di dati $x(n)$, $0 \leq n \leq N-1$ viene suddivisa in K segmenti di M campioni ciascuno ($N = KM$) e si costruiscono le seguenti sottosequenze,

$$x^{(i)}(n) = x(n + iM - M), \quad 0 \leq n \leq M-1, \quad 1 \leq i \leq K \quad (2.32)$$

di cui si calcolano i K periodogrammi $P_M^{(i)}(\omega)$ secondo la definizione (2.21). Se $r_{xx}(m)$ è piccola per $m > M$ allora ogni $P_M^{(i)}(\omega)$ si può ritenere indipendente e quindi si può costruire la stima dello spettro di potenza come

$$\hat{P}_{xx}^{(B)}(\omega) = \frac{1}{K} \sum_{i=1}^K \hat{P}_M^{(i)}(\omega) \quad (2.33)$$

il cui valore atteso è

$$E[\hat{P}_{xx}^{(B)}(\omega)] = \frac{1}{2\pi M} \int_{-\pi}^{\pi} P_{xx}(\theta) \left(\frac{\sin[(\omega - \theta)M/2]}{\sin[(\omega - \theta)/2]} \right)^2 d\theta \quad (2.34)$$

Dalla (2.34) si evince che la stima di Bartlett è la convoluzione dello spettro vero $P_{xx}(\omega)$ con la trasformata della finestra rettangolare relativa ad M campioni. È chiaro quindi che lo stimatore

è asintoticamente polarizzato. Se assumiamo che i periodogrammi siano indipendenti allora

$$\text{var}[\hat{P}_{xx}^{(B)}(\omega)] \simeq \frac{1}{K} P_{xx}^2(\omega) \left[1 + \left(\frac{\sin(\omega M)}{M \sin \omega} \right)^2 \right] \quad (2.35)$$

e quindi al crescere di K la varianza tende a zero. In base a questi risultati si può affermare che la stima di Bartlett è consistente. Osserviamo inoltre che per una lunghezza fissa della sequenza di dati, la varianza diminuisce al crescere del numero di periodogrammi, ma contemporaneamente diminuisce anche M , ovvero la risoluzione dello spettro. La scelta sarà sempre un compromesso tra polarizzazione e varianza e dipenderà dal tipo di segnale che si sta analizzando.

2.5 Metodo di Blackman-Tuckey

Estendendo le idee di Bartlett si può definire in modo generale uno stimatore dello spettro secondo la seguente definizione,

$$\hat{P}_{xx}^{(BT)}(\omega) = \sum_{k=-\infty}^{\infty} w(k) \hat{r}_{xx}(k), \quad (2.36)$$

dove $w(k)$ è una funzione pari tale che $w(0) = 1$ e $w(k) = 0$ per $|k| \geq M$ con la caratteristica di decadere in modo smussato verso lo zero. Poiché $w(k)$ pesa i ritardi della sequenza di autocorrelazione allora è chiamata anche *lag window*. La stima dello spettro [9] può essere fatta quindi facendo una convoluzione tra il periodogramma e una opportuna finestra spettrale $W(\omega)$,

$$\hat{P}_{xx}^{(BT)}(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{P}_N^{(p)}(\theta) W(\omega - \theta) d\theta. \quad (2.37)$$

Poiché per una sequenza $x(n)$ reale deve essere $\hat{P}_{xx}(\omega) \geq 0$ allora anche $W(\omega - \theta) \geq 0$, da cui si deduce che non tutte le finestre possono essere utilizzate. Il valore di aspettazione della (2.36) è

$$E[\hat{P}_{xx}^{(BT)}(\omega)] \approx \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{xx}(\theta) W(\omega - \theta) d\theta \quad (2.38)$$

mentre la varianza vale

$$\text{var}[\hat{P}_{xx}^{(BT)}(\omega)] \approx \left(\frac{1}{N} \sum_{m=-(M-1)}^{M-1} w^2(m) \right) P_{xx}(\omega)^2. \quad (2.39)$$

Le formule (2.38) e (2.39) valgono nel caso in cui la lunghezza $(2M - 1)$ della finestra $w(m)$ sia tale per cui $W(\omega - \theta)$ risulti a banda stretta rispetto alle variazioni di $P_{xx}(\omega)$ e a banda larga rispetto a $(\sin(\omega N/2)/\sin(\omega/2))^2$, ricordando infatti che $\hat{P}_N^{(p)}(\theta)$ è la convoluzione di $P_{xx}(\omega)$ con il *kernel* di Fejer. Il problema principale del periodogramma sono proprio le grandi variazioni

della funzione $P_{xx}(\omega)$, in questa logica l'uso delle finestra permette di smussare le fluttuazioni. Lo svantaggio però consiste in una perdita di risoluzione. Posta infatti la lunghezza M della finestra, minore sarà il suo valore maggiore sarà la riduzione della varianza e peggiore sarà la risoluzione spettrale. In effetti la risoluzione dello stimatore è dell'ordine di $1/M$ mentre la sua varianza è dell'ordine di M/N . E' chiaro quindi che le proprietà dello stimatore di Blackman e Tuckey dipende direttamente dalle proprietà della finestra. Per molte finestre di tipo segnale, con un picco sia nel dominio delle frequenze che nel dominio del tempo si definiscono un *tempo equivalente* ed un *larghezza di banda equivalente* in questo modo,

$$N_e = \frac{\sum_{k=-(M-1)}^{M-1} w(k)}{w(0)}, \quad (2.40)$$

e

$$\beta_e = \frac{\frac{1}{2\pi} \int_{-\pi}^{\pi} W(\omega) d\omega}{W(0)} \quad (2.41)$$

da cui si deduce che,

$$N_e \beta_e = 1. \quad (2.42)$$

Dalla (2.42) si capisce che essendo N_e proporzionale alla lunghezza della finestra, β_e sarà invece inversamente proporzionale ad essa. Per segnali di tipo più generale, la (2.42) rappresenta un limite inferiore e si modifica in

$$N_e \beta_e \geq 1. \quad (2.43)$$

2.6 Periodogramma di Welch

Il metodo di Welch [61] consiste nella modifica di quello di Bartlett sotto due aspetti:

- i segmenti si possono sovrapporre;
- ai dati viene applicata la finestra prima di calcolare il periodogramma.

Consideriamo il generico segmento definito come $x_j(n) = x((j-1)K + n)$, con $n = 1, \dots, M$ e $j = 1, \dots, S$. Se $K = M$ le sequenze non si sovrappongono e risultano essere N/M . Ad ogni segmento viene applicata la finestra secondo la formula seguente,

$$\hat{P}_M^i(\omega) = \frac{1}{MU} \left| \sum_{n=0}^{M-1} x^{(i)}(n) w(n) e^{-j\omega n} \right|^2, \quad (2.44)$$

con

$$U = \frac{1}{M} \sum_{n=0}^{M-1} w^2(n) \quad (2.45)$$

e la stima dello spettro diventa

$$\hat{P}^{(W)}(\omega) = \frac{1}{K} \sum_{i=1}^K P_M^i(\omega). \quad (2.46)$$

Il valore atteso della (2.46) risulta essere

$$E[\hat{P}^{(W)}(\omega)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} P_{xx}(\theta) \frac{1}{MU} \left| \sum_{n=0}^{M-1} w(n) e^{-j\omega n} \right|^2 d\theta. \quad (2.47)$$

Se le sotto-sequenze sono disgiunte allora la varianza risulta essere

$$\text{var}[\hat{P}^{(W)}(\omega)] \approx \frac{1}{K} P_{xx}^2(\omega). \quad (2.48)$$

Permettendo di sovrapporre le sotto-sequenze si aumentano il numero di periodogrammi, riducendo la varianza. In questo modo però si aumenta la correlazione tra i segmenti, tuttavia la finestra riduce i valori agli estremi del supporto e allieva questo effetto indesiderato. Lo stimatore di Welch può essere calcolato in modo efficiente con l'algoritmo della FFT e per questo motivo è uno dei metodi maggiormente utilizzati.

2.7 Periodogramma di Burg

Secondo la definizione (2.20) la densità spettrale di potenza è definita in base alla trasformata di Fourier dell'autocorrelazione. Tuttavia sebbene gli indici si estendano a $\pm\infty$, nei metodi non parametrici precedentemente elencati, gli stimatori $\hat{r}_{xx}(k)$ assumono che l'autocorrelazione si annulli per $|k| \geq N$. Questa ipotesi ovviamente del tutto arbitraria, introduce *leakage* e perdita di risoluzione spettrale. Il metodo di Burg [11] [12] assume invece che la stima di $r_{xx}(k)$ si basi sul segnale maggiormente caotico la cui autocorrelazione coincida con quella misurata per tutti $|k| < N$. In virtù di questa definizione è chiamato anche metodo della massima entropia. Per un dato segnale $x(t)$ l'entropia è definita come,

$$H[P_{xx}(\omega)] = \int_{-\pi}^{\pi} \ln(P_{xx}(\omega)) d\omega. \quad (2.49)$$

Imponendo la condizione di massimo, otteniamo

$$\frac{\partial H[P_{xx}(\omega)]}{\partial r_{xx}(k)} = \int_{-\pi}^{\pi} \frac{\partial \ln(P_{xx}(\omega))}{\partial r_{xx}(k)}(\omega) d\omega = 0. \quad (2.50)$$

Dalla (2.20) si ricava immediatamente che,

$$\frac{\partial P_{xx}(\omega)}{\partial r_{xx}(k)}(\omega) = e^{-i\omega k}, \quad (2.51)$$

da cui,

$$\frac{\partial \ln(P_{xx}(\omega))}{\partial r_{xx}(k)}(\omega) = \frac{1}{P_{xx}(\omega)} e^{-i\omega k}. \quad (2.52)$$

Sostituendo l'equazione (2.52) nella (2.50) otteniamo,

$$\int_{-\pi}^{\pi} P_{xx}^{-1}(\omega) e^{-i\omega k} d\omega = 0 \quad |k| > N. \quad (2.53)$$

Assumendo che il termine $P_{xx}^{-1}(\omega)$ sia integrabile, allora può essere associato ad una funzione di autocorrelazione $c(m)$, così definita,

$$P_{xx}^{-1}(\omega) = \sum_{m=-\infty}^{\infty} c(m) e^{-i\omega m}, \quad (2.54)$$

e

$$c(m) = \int_{-\pi}^{\pi} P_{xx}^{-1}(\omega) e^{i\omega m} d\omega. \quad (2.55)$$

Ma proprio in virtù della condizione (2.53) si ottiene che $c(m) = 0$ per $|m| \geq N$, pertanto l'inverso dello spettro di potenza può essere ottenuto come trasformata di Fourier di una serie finita di autocorrelazione,

$$P_{xx}^{-1}(\omega) = \sum_{m=-N}^N c(m) e^{-i\omega m}. \quad (2.56)$$

Lo stimatore della densità di potenza nel metodo di Burg risulta quindi essere,

$$\hat{P}_{xx}^{ME}(\omega) = \frac{1}{\sum_{m=-N}^N c(m) e^{-i\omega m}}. \quad (2.57)$$

Utilizzando la trasformata z e osservando che il denominatore della (2.57) è simmetrico, si può scrivere la seguente uguaglianza,

$$\sum_{m=-N}^N c(m) z^{-m} = \frac{\sigma^2}{A(z)A(z^{-1})} \quad (2.58)$$

con $A(z)$ polinomio di ordine P , $A(z) = 1 + a_1 z^{-1} + \dots + a_P z^{-P}$. Lo stimatore (2.57) diventa,

$$\hat{P}_{xx}^{ME}(\omega) = \frac{\sigma^2}{A(z)A(z^{-1})}. \quad (2.59)$$

E' possibile identificare l'espressione (2.59) con un modello autoregressivo (AR).

2.8 Modelli AR, MA e ARMA.

Utilizzando il teorema di Weierstrass si può affermare che qualsiasi funzione continua di densità di potenza spettrale può essere approssimata da una espressione razionale del tipo [8][43],

$$\tilde{P}_{xx}(\omega) = \frac{\sum_{k=-m}^m \gamma_k e^{-i\omega k}}{\sum_{k=-n}^n \rho_k e^{-i\omega k}}, \quad (2.60)$$

con $\gamma_{-k} = \gamma_k^*$, $\rho_{-k} = \rho_k^*$ e m, n di grado opportuno. La funzione (2.60), grazie al teorema della fattorizzazione spettrale, può essere posta anche nella forma,

$$\tilde{P}_{xx}(\omega) = \left| \frac{B(\omega)}{A(\omega)} \right|^2 \sigma^2, \quad (2.61)$$

con $\sigma > 0$ e,

$$\begin{aligned} A(\omega) &= 1 + a_1 e^{-i\omega} + \dots + a_n e^{-in\omega}, \\ B(\omega) &= 1 + b_1 e^{-i\omega} + \dots + b_m e^{-im\omega}. \end{aligned}$$

Utilizzando la trasformata z l'espressione (2.61) assume la forma,

$$\tilde{P}_{xx}(z) = \sigma^2 \frac{B(z)B^*(1/z^*)}{A(z)A^*(1/z^*)}. \quad (2.62)$$

Definito ora un arbitrario sistema lineare asintoticamente stabile avente funzione di trasferimento $H(t)$, la relazione tra ingresso ed uscita è data dalla convoluzione $x(t) = H(t) \star e(t)$. Sotto queste ipotesi vale la seguente importante formula,

$$P_{xx}(\omega) = |H(\omega)|^2 P_{ee}(\omega). \quad (2.63)$$

Osservando la (2.61) e (2.63) possiamo concludere che un arbitrario spettro di potenza può essere associato al segnale ottenuto filtrando un rumore bianco avente potenza σ^2 attraverso un filtro razionale con funzione di trasferimento $H(\omega) = B(\omega)/A(\omega)$. Per un sistema discreto possiamo identificare i seguenti tre modelli:

- ARMA modello autoregressivo a media mobile,

$$H(z) = \frac{\sum_{k=0}^m b_k z^{-k}}{1 + \sum_{k=1}^n a_k z^{-k}}$$

- AR modello autoregressivo,

$$H(z) = \frac{1}{1 + \sum_{k=1}^n a_k z^{-k}}$$

- MA modello a media mobile,

$$H(z) = \sum_{k=0}^m b_k z^{-k}.$$

Per ipotesi $P_{xx}(\omega)$ è finita per tutti i valori di ω , quindi $A(z) = 1 + \sum_{k=1}^n a_k z^{-k}$ non può avere zeri sul cerchio unitario. Inoltre poiché dalla equazione (2.62) gli zeri ed i poli di $P_{xx}(z)$ compaiono in coppie reciproche allora è sempre possibile scegliere $A(z)$ in modo che abbia gli zeri dentro il cerchio unitario. Il modello quindi risulta stabile. Inoltre se analogamente scegliamo gli zeri di $B(z)$ dentro il cerchio unitario allora il modello è anche a fase minima. Ciò significa che qualsiasi altro polinomio $C(z)$ avente modulo uguale a $|B(z)|$ per ogni z , il ritardo di fase è maggiore od uguale a quello di $B(z)$. Scriviamo ora esplicitamente le formule per un processo ARMA:

$$x(t) + \sum_{i=1}^n a_i x(t-i) = \sum_{j=1}^m b_j e(t-j), \quad b_0 = 1, \quad t = 1, \dots, N. \quad (2.64)$$

Moltiplichiamo la (2.64) per $x^*(t-k)$ e prendiamo il valore atteso,

$$E[x(t)x^*(t-k)] + \sum_{i=1}^n a_i E[x(t-i)x^*(t-k)] = \sum_{j=1}^m b_j E[e(t-j)x^*(t-k)]. \quad (2.65)$$

Poiché valgono le seguenti relazioni,

$$\begin{aligned} E[x(t)x^*(t-k)] &= r_{xx}(k) \\ E[x(t-i)x^*(t-k)] &= r_{xx}(k-i), \end{aligned}$$

la (2.65) si riscrive così,

$$r_{xx}(k) + \sum_{i=1}^n a_i r_{xx}(k-i) = \sum_{j=1}^m b_j E[e(t-j)x^*(t-k)]. \quad (2.66)$$

Ricordando ora la definizione di filtro stabile e causale per $H(z)$, abbiamo che

$$x(t) = \sum_{s=0}^{\infty} h_s e(t-s), \quad (2.67)$$

e sostituendo la (2.67) nella (2.66), otteniamo

$$r_{xx}(k) + \sum_{i=1}^n a_i r_{xx}(k-i) = \sum_{j=1}^m b_j \sum_{s=0}^{\infty} h_s^* E[e(t-j)e^*(t-k-s)] = \sigma^2 \sum_{j=1}^m b_j \sum_{s=0}^{\infty} h_s^* \delta_{j,k+s}, \quad (2.68)$$

ed infine,

$$r_{xx}(k) + \sum_{i=1}^n a_i r_{xx}(k-i) = \sigma^2 \sum_{j=1}^m b_j h_{j-k}^*. \quad (2.69)$$

Osserviamo che poiché $h_s = 0$ per $s < 0$ la (2.69) si riduce alla seguente espressione semplificata,

$$r_{xx}(k) + \sum_{i=1}^n a_i r_{xx}(k-i) = 0, \quad k > m. \quad (2.70)$$

2.8.1 Caso AR

Per un modello AR, in cui $m = 0$ e $B(z) = 1$, l'equazione (2.69) diventa,

$$r_{xx}(0) + \sum_{i=1}^n a_i r_{xx}(-i) = \sigma^2 \quad (2.71)$$

$$r_{xx}(k) + \sum_{i=1}^n a_i r_{xx}(k-i) = 0, \quad k > 0. \quad (2.72)$$

Le equazioni (2.71) e (2.72) sono chiamate anche equazioni di *Yule-Walker*. Poste in forma matriciale esse diventano,

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(-1) & \cdots & r_{xx}(-n) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(-n+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(n) & r_{xx}(n-1) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (2.73)$$

oppure

$$\begin{pmatrix} r_{xx}(1) \\ \vdots \\ r_{xx}(n) \end{pmatrix} + \begin{pmatrix} r_{xx}(0) & \cdots & r_{xx}(-n+1) \\ \vdots & \ddots & \vdots \\ r_{xx}(n-1) & \cdots & r_{xx}(0) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (2.74)$$

Definendo inoltre,

$$R_n = \begin{pmatrix} r_{xx}(0) & \cdots & r_{xx}(-n+1) \\ \vdots & \ddots & \vdots \\ & r_{xx}(n-1) & \cdots & r_{xx}(0) \end{pmatrix},$$

$$r_n = \begin{pmatrix} r_{xx}(1) \\ \vdots \\ r_{xx}(n) \end{pmatrix},$$

$$\theta = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}$$

Le equazioni (2.73) e (2.74) possono essere poste nella forma simbolica,

$$R_{n+1} \begin{pmatrix} 1 \\ \theta_n \end{pmatrix} = \begin{pmatrix} \sigma^2 \\ 0 \end{pmatrix}, \quad (2.75)$$

$$r_n + R_n \theta_n = 0. \quad (2.76)$$

Se le sequenze di autocorrelazione $\{r_{xx}(k)\}_{k=0}^n$ sono note allora il sistema (2.76) si risolve facilmente e il vettore dei coefficienti ignoti risulta essere,

$$\theta_n = -R_n r_n \quad (2.77)$$

$$\sigma^2 = r_{xx}(0) + \sum_{i=1}^n a_i r_{xx}(-i). \quad (2.78)$$

In pratica le autocorrelazioni vanno sostituite con le rispettive stime introdotte nei paragrafi precedenti. E' interessante utilizzare anche un altro approccio legato al problema della predizione lineare. Consideriamo il modello di filtro

$$y_t = f_t \star x_t, \quad (2.79)$$

e confrontiamo con un segnale d_t . Ci chiediamo quali sono i coefficienti che minimizzano l'errore quadratico medio,

$$L = \sum_t (d_t - y_t)^2. \quad (2.80)$$

E' chiaro che si tratta di un problema ai minimi quadrati che può essere risolto valutando le derivate di L rispetto al vettore dei parametri $(f_0, f_1, \dots, f_{n-1})$. Procediamo sostituendo la (2.79) nella (2.80),

$$L = \sum_t \left(d_t - \sum_{\tau} f_{\tau} x_{t-\tau} \right)^2, \quad (2.81)$$

e valutiamo le derivate parziali,

$$\frac{\partial L}{\partial f_i} = 0, \quad i = 0, 1, 2, \dots, (n-1). \quad (2.82)$$

Espandendo i termini quadratici della (2.81) si ha,

$$L = \sum_t d_t^2 - 2 \sum_t \sum_\tau f_\tau x_{t-\tau} + \sum_t \left(\sum_\tau f_\tau x_{t-\tau} \right)^2 = 0, \quad (2.83)$$

e applicando la condizione di minimo,

$$\frac{\partial L}{\partial f_i} = -2 \sum_t d_t x_{t-\tau} + 2 \sum_t \left(\sum_\tau f_\tau x_{t-\tau} \right) x_{t-i} = 0, \quad (2.84)$$

$$\sum_\tau f_\tau \sum_t x_{t-\tau} x_{t-i} = \sum_t d_t x_{t-\tau}. \quad (2.85)$$

Sostituendo nella (2.85) le seguenti equivalenze,

$$\sum_t x_{t-\tau} x_{t-i} = r_{i-\tau} \quad (2.86)$$

e

$$\sum_t d_t x_{t-i} = g_i, \quad (2.87)$$

si ottiene

$$\sum_\tau f_\tau r_{i-\tau} = g_i, \quad i = 0, 1, 2, \dots, (n-1). \quad (2.88)$$

L'equazione (2.88) può essere posta in forma matriciale,

$$\begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ r_1 & r_0 & r_1 & \cdots & r_{n-2} \\ \vdots & \vdots & r_1 & \ddots & \vdots \\ r_{n-1} & r_{n-2} & r_{n-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix}, \quad (2.89)$$

dove r_t è l'autocorrelazione dell'input e g_t è la correlazione tra output e input. Possiamo quindi riscrivere la (2.81) al modo seguente,

$$L_{\min} = \sum_t d_t^2 - \sum_\tau f_\tau g_\tau. \quad (2.90)$$

Le espressioni (2.89) sono chiamate equazioni di *Wiener-Hopf*. Supponiamo che il l'output desiderato sia un valore nel futuro, $d(t) = x(t + \alpha)$, allora vogliamo progettare il filtro $f(t)$ in modo tale che predica $x(t + \alpha)$ a partire dai valori passati. In questo caso la funzione di correlazione diventa,

$$g_\tau = \sum_{\tau} d - \tau x_{t-\tau} = \sum_t x_{t+\alpha} x_{t-\tau} = \sum_t x_t x_{t-(\alpha+\tau)} = r_{\alpha+\tau} \quad (2.91)$$

Sostituendo la (2.91) nella (2.89) si ottiene

$$\begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ r_1 & r_0 & r_1 & \cdots & r_{n-2} \\ \vdots & \vdots & r_1 & \ddots & \vdots \\ r_{n-1} & r_{n-2} & r_{n-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} r_\alpha \\ r_{\alpha+1} \\ \vdots \\ r_{\alpha+n-a} \end{pmatrix}. \quad (2.92)$$

Se consideriamo ora un ritardo pari a $\alpha = 1$ allora la (2.92) diviene,

$$\begin{pmatrix} r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ r_1 & r_0 & r_1 & \cdots & r_{n-2} \\ \vdots & \vdots & r_1 & \ddots & \vdots \\ r_{n-1} & r_{n-2} & r_{n-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}. \quad (2.93)$$

Modificando i termini della (2.93) otteniamo,

$$\begin{pmatrix} -r_1 & r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ -r_2 & r_1 & r_0 & r_1 & \cdots & r_{n-2} \\ \vdots & \vdots & \vdots & r_1 & \ddots & \vdots \\ -r_n & r_{n-1} & r_{n-2} & r_{n-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} 1 \\ f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (2.94)$$

Se ora aggiungiamo un'altra riga e cambiamo i segni

$$\begin{pmatrix} r_0 & r_1 & r_2 & r_2 & \cdots & r_n \\ r_1 & r_0 & r_1 & r_2 & \cdots & r_{n-1} \\ r_2 & r_1 & r_0 & r_1 & \cdots & r_{n-2} \\ \vdots & \vdots & \vdots & r_1 & \ddots & \vdots \\ r_n & r_{n-1} & r_{n-2} & r_{n-3} & \cdots & r_0 \end{pmatrix} \begin{pmatrix} 1 \\ -f_0 \\ -f_1 \\ \vdots \\ -f_{n-1} \end{pmatrix} = \begin{pmatrix} L \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (2.95)$$

otteniamo un sistema di $n + 1$ equazioni in $n + 1$ incognite $(f_0, f_1, \dots, f_{n-1}, L)$:

$$L = r_0 - r_1 f_0 - \dots - r_n f_{n-1}. \quad (2.96)$$

Introducendo nella (2.83) la predizione di un passo

$$d_t = x_{t+1}, \quad (2.97)$$

$$\sum_t d_t^2 = r_0, \quad (2.98)$$

$$g_t = r_{t+1}, \quad (2.99)$$

otteniamo

$$L_{\min} = r_0 - (r_1 f_0 + r_2 f_1 + \dots + r_n f_{n-1}). \quad (2.100)$$

La (2.100) è identica alla (2.96), pertanto risolvere il sistema (2.95) equivale a calcolare i coefficienti del filtro con il minimo errore. Le equazioni (2.95) sono equivalenti a quelle (2.75), quindi trovare i coefficienti di un modello AR equivale ad un problema di predizione lineare. Infine lo spettro di potenza del modello autoregressivo sarà

$$\hat{P}_{xx}^{AR}(\omega) = \frac{\sigma^2}{\left|1 - \sum_{k=1}^P a_k e^{-j\omega k}\right|^2}. \quad (2.101)$$

2.8.2 Algoritmo di Levinson-Durbin per modelli AR

La risoluzione delle equazioni di *Yule-Walker* o *Wiener-Hopf* all'aumentare dell'ordine del modello AR diventa particolarmente oneroso dal punto di vista computazionale. Per superare questa difficoltà Levinson e Durbin [17] hanno proposto un algoritmo ricorsivo particolarmente efficiente. Consideriamo un modello del primo ordine,

$$\hat{x}(n) = a_1^{(1)} x(n-1), \quad (2.102)$$

e valutiamo l'errore di predizione,

$$e^{(1)}(n) = x(n) - \hat{x}(n) = x(n) - a_1^{(1)} x(n-1). \quad (2.103)$$

Per quanto abbiamo visto nel precedente paragrafo esso deve essere ortogonale ai dati utilizzati per la stima, quindi dovrà essere,

$$E[e^{(1)}(n)x(n-1)] = E[(x(n) - a_1^{(1)}x(n-1))x(n-1)] = 0, \quad (2.104)$$

da cui si ricava immediatamente il valore del primo coefficiente,

$$a_1^{(1)} = \frac{E[x(n)x(n-1)]}{E[x^2(n-1)]} = \frac{r_{xx}(1)}{r_{xx}(0)}. \quad (2.105)$$

In generale per predire il valore del campione all'istante n , con un filtro di ordine N , bisogna utilizzare gli N campioni precedenti a $x(n)$,

$$\hat{x}^{(N)}(n) = \sum_{k=1}^N a_k^{(N)} x(n-k). \quad (2.106)$$

In effetti sfruttando il fatto che il processo è stazionario possiamo procedere al contrario e cioè stimare anche il campione $x(n-N-1)$ con lo stesso modello,

$$\hat{x}^{(N)}(n-N-1) = \sum_{k=1}^N a_k^{(N)} x(n-N-1+k), \quad (2.107)$$

a cui sarà associato un errore di predizione,

$$e_{(N)}^B = x(n-N-1) - \hat{x}^{(N)}(n-N-1). \quad (2.108)$$

Il vantaggio di considerare l'equazione (2.108) è che il campione $x(n-N-1)$ è già stato acquisito quindi può essere utilizzato per valutare la bontà della stima. Se il processo $x(N)$ è autoregressivo significa che il generico campione è correlato solo agli N campioni precedenti o successivi. Un predittore di ordine N quindi estrae tutta l'informazione possibile e quindi non avrebbe senso utilizzare $x(n)$ o $x(n-N-1)$ essendo appunto non correlati. Tuttavia se a priori non è noto l'ordine del processo autoregressivo non è nemmeno possibile stabilire N , inoltre come eventualmente si può considerare il predittore di ordine $N+1$ avendo appena determinato quello di ordine N ? In effetti parte dell'informazione è già stata estratta, quindi $x(n-N-1)$ deve essere utilizzato solo per la nuova informazione che porta e cioè l'*innovazione*. Riconosciamo immediatamente che (2.108) risolve proprio questo problema. Quindi ha senso scrivere la seguente uguaglianza,

$$\hat{x}^{(N+1)}(n) = \hat{x}^{(N)}(n) + c_{N+1} e_{(N)}^B(n-N-1). \quad (2.109)$$

Utilizzando la (2.109) esplicitiamo l'errore in avanti,

$$e_{(N+1)}^F(n) = x(n) - \hat{x}^{(N+1)}(n) = e_{(N)}^F(n) - c_{N+1} e_{(N)}^B(n-N-1), \quad (2.110)$$

e osserviamo che il criterio di scelta del coefficiente c_{N+1} è quello di minimizzare l'errore atteso della (2.110). Posto quindi

$$\frac{\partial E[(e_{(N+1)}^F(n))^2]}{\partial c_{N+1}} = 0, \quad (2.111)$$

ricaviamo,

$$c_{N+1} = \frac{E[e_{(N)}^F(n)e_{(N)}^B(n-N-1)]}{E[(e_{(N)}^B(n-N-1))^2]}. \quad (2.112)$$

Nell'equazione (2.112) il termine $E[(e_{(N)}^B(n-N-1))^2]$ può essere valutato ricorsivamente, essendo valida la relazione $e_{(N)}^B(n-N-1) = (1-c_N^2)e_{(N-1)}^B(n-N-1)$. In definitiva quindi l'esposizione precedente può essere riassunta nel seguente algoritmo ricorsivo:

- $E^{(0)} = r_{xx}(0)$;
- per $i = 1, \dots, N$

$$\Delta^{(i-1)} = r_{xx}(i) - \sum_{k=1}^{i-1} a_k^{(i-1)} r_{xx}(i-k),$$

$$c_i = \frac{\Delta^{(i-1)}}{E^{(i-1)}},$$

$$a^{(i)} = c_i,$$

$$a_j^i = a_j^{(i-1)} - c_i a_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1,$$

$$E^{(i)} = (1 - c_i^2)E^{(i-1)}.$$

La quantità c_i ha un significato rilevante dal punto di vista statistico, esso infatti misura la correlazione mutua tra l'errore di predizione in avanti e quello indietro dopo aver eliminato tutte le dipendenze tra il campione $x(n)$ e quello $x(n-N-1)$, che appunto dipendono dai campioni intermedi. Essa quindi viene chiamata *coefficiente di correlazione parziale* ed il suo valore numerico scende bruscamente per modelli AR.

2.8.3 Caso MA

Il modello MA è descritto dall'equazione (2.113) o equivalentemente dalla (2.114)

$$H(z) = \sum_{k=0}^m b_k z^{-k} \quad (2.113)$$

$$x(n) = \sum_{k=0}^m b_k e(n-k), \quad (2.114)$$

e quindi l'espressione (2.69) diventa

$$r_{xx}(k) = \sigma^2 \sum_{j=1}^m b_j b_{j-k}. \quad (2.115)$$

L'espressione (2.115) rappresenta un sistema di m equazioni non lineari nelle m incognite b_j e assumono la seguente forma,

$$\begin{cases} r(0) = \sigma^2 \sum_{j=1}^m b_j^2 \\ r(1) = \sigma^2 \sum_{j=1}^m b_j b_{j-1} \\ \dots \\ r(m) = \sigma^2 \sum_{j=1}^m b_j b_{j-m} \end{cases} .$$

Il sistema da risolvere assume una forma quadratica. Una procedura per risolvere questo problema si basa sul metodo di *Durbin*. Poiché sappiamo dalla (2.62) che ad ogni radice $z = \alpha$ esiste anche la reciproca $z = 1/\alpha$, quindi per ogni coefficiente del filtro b_k si hanno due possibili soluzioni. In totale quindi esistono $2m$ soluzioni che generano 2^m combinazioni di filtri possibili. Se si sceglie tuttavia un filtro a *fase minima* allora tutte le radici di $B(z^{-1})$ dovranno stare dentro il cerchio unitario e viceversa per $B(z)$. In sostanza il modello MA ipotizza un andamento non nullo per la funzione di autocorrelazione solo nell'intervallo $|k| \leq m$, quindi la sua PSD assume la forma,

$$P_{xx}(\omega) = \sum_{k=-m}^{k=m} r(k) e^{-i\omega t} .$$

Questo equivale ad applicare una finestra rettangolare alla funzione di autocorrelazione effettiva. In effetti la procedura risulta essere uguale a quella del metodo di *Blackman-Turkey*.

2.8.4 Caso ARMA

Quando uno spettro presenta sia picchi che estese valli e non può essere rappresentato da modelli AR o MA di ordine ragionevole allora occorre utilizzare il modello generale ARMA, rappresentato dall'equazione seguente,

$$H(z) = \frac{\sum_{k=0}^m b_k z^{-k}}{1 + \sum_{k=1}^n a_k z^{-k}} . \quad (2.116)$$

Esplicitando l'equazione (2.69) otteniamo,

$$r_{xx}(k) = \begin{cases} -\sum_{j=1}^n a_j r_{xx}(k-j) & n > m \\ -\sum_{j=1}^n a_j r_{xx}(k-j) + \sigma^2 \sum_{j=1}^m b_j h_{j-k}^* & 0 \leq n \leq m \\ -r_{xx}^*(-k) & n < 0 \end{cases} .$$

Si vengono quindi a creare un sistema di $n + m$ equazioni basate sulle stime dei campioni di $r_{xx}(k)$. Il modello ARMA può essere visto come la serie di un modello AR (solo poli) e di uno MA (solo zeri),

$$H_{ARMA}(z) = H_{MA}(z) H_{AR}(z) . \quad (2.117)$$

Poiché per il calcolo di $H_{MA}(z)$ è necessario considerare solo i primi m valori allora possiamo stimare separatamente $H_{AR}(z)$ considerando solo i ritardi maggiori di m . Mediante questa

tecnica possiamo applicare le equazioni di *Yule-Walker* al sistema,

$$\begin{pmatrix} r_{xx}(m) & r_{xx}(m-1) & \cdots & r_{xx}(m-n+1) \\ r_{xx}(m+1) & r_{xx}(m) & \cdots & r_{xx}(m-n) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(m+n-1) & r_{xx}(m+n-2) & \cdots & r_{xx}(m) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} r_{xx}(m+1) \\ r_{xx}(m+2) \\ \vdots \\ r_{xx}(m+n) \end{pmatrix}. \quad (2.118)$$

Dopo aver risolto il sistema (2.118) si considerano i ritardi minori di m e si calcolano i coefficienti di H_{AR} . Lo svantaggio di questa procedura è che il modello AR viene calcolato a ritardi elevati dove la varianza tende ad aumentare.

2.9 Filtro di Wiener-Kolmogorov

Norbert Wiener e indipendentemente Andrey Nikolaevich Kolmogorov formularono nel dominio del tempo continuo la stima ai minimi quadrati dei coefficienti del filtro che calcola la correlazione incrociata a partire dall'autocorrelazione di un segnale misurato dato il segnale di riferimento e il rumore del processo. L'estensione nel tempo discreto risulta più conveniente per l'implementazione in un sistema digitale. Considerati $\{w_i\}_{i=0}^{P-1}$ i coefficienti del filtro, la stima del segnale desiderato a partire dal segnale misurato è

$$\hat{x}(m) = \sum_{k=0}^{P-1} w_k y(m-k), \quad (2.119)$$

che in notazione matriciale diventa,

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(N-1) \end{pmatrix} = \begin{pmatrix} y(0) & y(-1) & \cdots & y(-P+1) \\ y(1) & y(0) & \cdots & y(-P+2) \\ \vdots & \vdots & \ddots & \vdots \\ y(N-1) & y(N-2) & \cdots & y(N-P) \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{P-1} \end{pmatrix}. \quad (2.120)$$

Secondo la procedura ai minimi quadrati deve essere minimizzato l'errore $e(m) = x(m) - \hat{x}(m)$ rispetto al set di coefficienti $\{w_i\}$. Come anticipato nel paragrafo relativo ai modelli AR, la soluzione è analoga alle equazioni di *Yule-Walker* e risulta essere,

$$\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{P-1} \end{pmatrix} = \begin{pmatrix} r_{yy}(0) & r_{yy}(-1) & \cdots & r_{yy}(P-1) \\ r_{yy}(1) & r_{yy}(0) & \cdots & r_{yy}(P-2) \\ \vdots & \vdots & \ddots & \vdots \\ r_{yy}(P-1) & r_{yy}(P-2) & \cdots & r_{yy}(0) \end{pmatrix}^{-1} \begin{pmatrix} r_{yx}(0) \\ r_{yx}(1) \\ \vdots \\ r_{yx}(P-1) \end{pmatrix}. \quad (2.121)$$

Dalla (2.121) si deduce che per la stima dei coefficienti è necessario calcolare la correlazione tra il vettore del segnale di input e il segnale desiderato. Questa informazione non è sempre disponibile inoltre spesso i segnali non sono stazionari e quindi è necessario segmentare i vettori in modo da ottenere dei processi quasi-stazionari. Il filtro di Wiener può essere anche interpretato come proiezione in uno spazio vettoriale. In effetti l'equazione (2.119) può essere scritta anche in questi termini,

$$\begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(N-1) \end{pmatrix} = w_0 \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{pmatrix} + w_1 \begin{pmatrix} y(-1) \\ y(0) \\ \vdots \\ y(N-2) \end{pmatrix} + \cdots + w_{P-1} \begin{pmatrix} y(1-P) \\ y(2-P) \\ \vdots \\ y(N-P) \end{pmatrix}. \quad (2.122)$$

E' chiaro che se interpretiamo $[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{P-1}]$ come la base di un sottospazio, allora la stima $\hat{\mathbf{x}}$ sarà interamente contenuta in tale sottospazio. Poiché in genere il segnale di ingresso sarà composto da N elementi con $N > P$ allora avremo $\hat{\mathbf{x}} = \mathbf{x}$ solo se il segnale desiderato è interamente contenuto nel sottospazio altrimenti la miglior stima di \mathbf{x} sarà uguale alla proiezione di \mathbf{x} nel sottospazio in esame.

Il filtro di Wiener può essere espresso anche nel dominio delle frequenze, in questo caso l'espressione sarà,

$$\hat{X}(\omega) = W(\omega)Y(\omega), \quad (2.123)$$

e l'errore di stima sarà dato da

$$E(\omega) = X(\omega) - \hat{X}(\omega) = X(\omega) - W(\omega)Y(\omega). \quad (2.124)$$

L'errore quadratico medio atteso risulta quindi essere

$$E[|E(\omega)|^2] = E[(X(\omega) - W(\omega)Y(\omega))^*(X(\omega) - W(\omega)Y(\omega))]. \quad (2.125)$$

Minimizzando l'equazione (2.125) rispetto a $W(\omega)$ otteniamo,

$$\begin{aligned} \frac{\partial E[|E(\omega)|^2]}{\partial W(\omega)} &= E[-Y^*(\omega)X(\omega) - X^*(\omega)Y(\omega) + W(\omega)|Y(\omega)|^2] \\ &= 2W(\omega)P_{YY}(\omega) - 2P_{XY}(\omega). \end{aligned} \quad (2.126)$$

Eguagliando a zero l'equazione (2.126) si ottiene la soluzione

$$W(\omega) = \frac{P_{XY}(\omega)}{P_{YY}(\omega)}. \quad (2.127)$$

Un caso notevole di utilizzo del filtro di Wiener è quando si è in presenza di un segnale rumoroso, $y(m) = x(m) + n(m)$ con $n(m)$ rumore bianco. In questo caso si può ipotizzare che $x(m)$ e

$n(m)$ siano non correlati e quindi la relativa matrice di correlazione sarà nulla. Risulta quindi che $Y(\omega) = X(\omega) + N(\omega)$ e pertanto la (2.127) si semplifica in

$$W(\omega) = \frac{P_{XX}(\omega)}{P_{XX}(\omega) + P_{NN}(\omega)}. \quad (2.128)$$

Poiché inoltre $SNR(\omega) = P_{XX}(\omega)/P_{NN}(\omega)$ la (2.128) si può riscrivere in questo modo

$$W(\omega) = \frac{SNR(\omega)}{SNR(\omega) + 1}. \quad (2.129)$$

Le prestazioni del filtro di Wiener sono limitate da alcuni fattori ed in particolare dal decrescere del fattore SNR. Il filtro inoltre è lineare e pertanto qualsiasi distorsione ne degrada il risultato.

2.10 Modelli di rumore

Le varie forme di rumore possono essere classificate in un certo numero di categorie che richiamano l'origine fisica e comunemente sono:

- disturbi acustici. Ad esempio rumori causati da movimento, vibrazioni o collisioni oppure da echi e riflessioni.
- Rumore dei dispositivi elettrici. In questa categoria rientrano i seguenti tipi di rumore:
 - rumore termico, dovuto al movimento caotico dei portatori di carica.
 - Rumore *flicker*, dovuto a vari contributi, come ad esempio le impurità. Il suo spettro varia come $1/f$.
 - Rumore *shot*, dovuto a variazioni casuali della corrente.
 - Rumore *burst*, dovuto ad improvvisi incrementi del livello di tensione elettrica.
- Rumore elettromagnetico. Si tratta di un disturbo presente a tutte le frequenze ed in particolare nella banda radio.
- Rumore elettrostatico. Esso è generato dalla presenza di tensione elettrica, ad esempio le lampade a fluorescenza sono una tipica sorgente.

A queste categorie standard si possono aggiungere anche le distorsioni o i *crosstalk* nei canali di comunicazione, oppure la mancanza di campioni o di pacchetti digitali ed infine gli errori di quantizzazione. Il rumore inoltre si può classificare anche in base alle sue caratteristiche in frequenza e temporali. In base a questa logica si individuano i seguenti modelli di rumore:

- rumore bianco, si tratta di segnali caotici impulsivi non correlati. In principio esso contiene tutte le frequenze allo stesso livello di potenza.

- Rumore bianco limitato in banda.
- Rumore a banda stretta. Un tipico esempio è il segnale dovuto ai 50 Hz dell'alimentazione elettrica.
- Rumore colorato. Si tratta di segnale caotico che non ha uno spettro piatto in frequenza.
- Rumore impulsivo. Si tratta di impulsi di breve durata.
- Rumore ad impulsi transitori. Si tratta di impulsi relativamente lunghi come possono essere i *clicks* o i *burst*.

Sicuramente il rumore bianco costituisce un concetto teorico fondamentale. Il suo spettro si estende a tutte le frequenze e per questo viene detto piatto. Nel tempo è rappresentato da una delta di Dirac. La sua funzione di autocorrelazione è data da

$$r_{nn}(\tau) = E[n(t)n(t + \tau)] = \sigma_n^2 \delta(\tau), \quad (2.130)$$

e la densità di potenza è

$$P_{NN}(\omega) = \int_{-\infty}^{\infty} r_{nn}(t) e^{-i\omega t} dt = \sigma_n^2. \quad (2.131)$$

Chiaramente il processo bianco è una pura idealizzazione, tuttavia un concetto molto più pratico consiste nell'introdurre una banda B ,

$$P_{NN}(\omega) = \begin{cases} \sigma & |f| \leq B \\ 0 & |f| > B \end{cases}. \quad (2.132)$$

In questo caso la potenza totale sarà $2B\sigma^2$ e la sua funzione di autocorrelazione sarà data una funzione del tipo $\sin x/x$. Un altro concetto molto utile consiste nella definizione di rumore colorato, cioè di un segnale caotico a larga banda il cui spettro non è piatto. Due esempi classici sono il rumore rosa e quello marrone.

Capitolo 3

Trasformata Wavelet discreta

In questo capitolo si traccia una breve introduzione teorica dell'analisi Wavelet discreta. La letteratura che tratta questo argomento è vasta e utili riferimenti possono ad esempio essere trovati in [1] [36] [24]. L'attività di ricerca prevede lo sviluppo di un software in *real-time* che sia in grado eseguire la DWT nei tempi di ciclo previsti. L'esposizione teorica riportata ha lo scopo di gettare le basi per la realizzazione di un algoritmo in C++ (descritto nel capitolo 5) agevolmente implementabile ed i cui risultati siano comparabili con quello che si possono ottenere dal *wavelet toolbox* di matlab. Il concetto chiave da discutere è il banco di filtri con la nozione di ricostruzione perfetta. Compreso questo elemento fondamentale si può agevolmente introdurre la teoria della multi-risoluzione, che prevede appunto l'applicazione iterativa del concetto di banco di filtri. Sebbene esistano diverse realizzazioni pratiche di filtri, sicuramente i filtri di Daubeuchies sono da considerarsi tra i più importanti. Per questo motivo si discute anche il metodo algoritmico con cui è possibile ricavare i coefficienti. Gli aspetti più prettamente numerici, come ad esempio la finitezza del numero dei campioni acquisiti, saranno discussi nel capitolo 5.

3.1 Introduzione storica

Le *wavelet* sono funzioni che soddisfano certe condizioni matematiche e sono usate per rappresentare dati o altre funzioni. Questa idea naturalmente non è nuova, ma risale a Fourier, tuttavia nell'analisi *wavelet* assume un ruolo chiave il concetto di scala o risoluzione. L'idea di base è di guardare un segnale con differenti "finestre" ognuna delle quali è adatta a cogliere diversi dettagli, che possono agire su scale completamente differenti. Le funzioni seno e coseno sono per definizione non locali ed infatti non si prestano molto bene ad analizzare ad esempio impulsi o repentine variazioni del segnale. L'analisi *wavelet* prevede invece l'utilizzo di una funzione prototipo localizzata, chiamata *mother wavelet*. Essa viene contratta e dilatata in modo da poter analizzare tutte le scale. Dal punto di vista storico lo sviluppo di questa tecnica mostra diverse

origini, nel senso che vari settori del mondo scientifico si sono indipendentemente interessati a questi concetti. La prima citazione delle *wavelet* avvenne nella tesi di laurea di Haar, che introdusse delle basi ortonormali, che tuttavia avevano la caratteristica di non essere continuamente differenziabili. Negli anni Trenta, molti gruppi di ricercatori in modo indipendente lavorarono sul concetto di basi di funzioni a scala variabile. Ulteriori sviluppi si ebbero tra gli anni Sessanta e Ottanta grazie ai lavori di Guido Weiss, Ronald R. Coifman e Morlet. Ma sicuramente fu nel 1985 che con il suo lavoro Stephan Mallat diede un impulso fondamentale a questa tecnica scoprendo alcune relazioni chiave tra i filtri in quadratura, gli algoritmi piramidali e le basi *wavelet* ortonormali. Inspirato da questo lavoro Y. Meyer costruì il primo esempio non banale di *wavelet*. Contrariamente alle basi di Haar quelle di Meyer sono continuamente differenziabili anche se non hanno un supporto compatto. Alcuni anni dopo usando il lavoro di Mallat, Ingrid Debeuchies costruì un set di basi ortonormali che sicuramente rappresentano una pietra miliare. A partire da quegli anni l'analisi *wavelet* ha assunto un ruolo sempre più rilevante nell'analisi del segnale, continuamente arricchendosi di nuovi strumenti teorici ed algoritmici.

Se una funzione non è periodica la somma di seni e coseni non la rappresenta accuratamente. Per arginare questo problema si fa ricorso alla trasformata di Fourier finestrata (WFT), ma come abbiamo visto nel capitolo sull'analisi classica della potenza, l'intervento non è esente da controindicazioni e certamente il *leakage* è uno dei noti aspetti negativi. Ma sicuramente la caratteristica più evidente riguarda la risoluzione costante in frequenza. Se nel piano tempo-frequenza tracciamo appunto il legame tra Δt e Δf osserviamo che esso è rappresentato da un reticolo regolare di rettangoli tutti uguali. Questa rigidità viene spezzata dalle *wavelet* le quali grazie al concetto di scala variabile hanno funzioni di base sia per le alte frequenze che per quelle basse. La relazione tempo-frequenza assume quindi un legame variabile, e se la rappresentiamo nello stesso piano di quella di Fourier osserviamo che a basse risoluzioni Δt risulta maggiore rispetto a quello associato alle alte frequenze. In sostanza quindi fissata una scala la funzione in *quella determinata scala* viene scomposta spostando la *wavelet* lungo l'asse temporale. Questa operazione genera un set di dati che sono proprio i coefficienti della proiezione. Ripetendo il procedimento *per ogni* scala si ottiene una rappresentazione adeguata in tutto il piano tempo-frequenza. Da questa breve descrizione risulta evidente l'importanza della località della funzione *wavelet* in opposizione al seno e coseno. La scomposizione di Fourier non può contemplare il concetto di "scorrimento" ed infatti produce un solo coefficiente globale per ogni frequenza, contrariamente alle *wavelet* che ne generano un intero insieme.

3.2 **Analisi Wavelet e banco di filtri**

Un approccio comune alla DWT è legato al concetto di banco di filtri a due canali. Un filtro è una funzione lineare che lega un segnale ad energia finita con uno sempre ad energia finita. Nel

dominio del tempo è dato dalla convoluzione del segnale con un dato vettore \mathbf{h} . Per preservare la proprietà relativa all'energia assumiamo che la trasformata z di questo vettore, $H(z)$, sia limitata sul cerchio unitario. La convoluzione con \mathbf{h} è chiamata filtraggio, mentre il vettore \mathbf{h} è chiamato risposta all'impulso del filtro e $H(e^{j\omega})$ rappresenta la sua funzione di trasferimento. Un banco a due filtri, inizia con due filtri di analisi, che indichiamo con \mathbf{h}_0 e \mathbf{h}_1 e prosegue con due filtri di sintesi, \mathbf{g}_0 e \mathbf{g}_1 . Assumiamo che tutti siano filtri FIR a coefficienti reali. In genere il filtro con indice 0 è un filtro passa basso, mentre quello con indice 1 è passa alto. Nel primo caso il suo valore è prossimo all'unità per $|\omega| \leq \pi/2$ e vicino allo zero per $\pi/2 < \omega < \pi$, con π frequenza di *Nyquist*. L'insieme è chiamato banco di filtri ed è rappresentato nella figura (3.1).

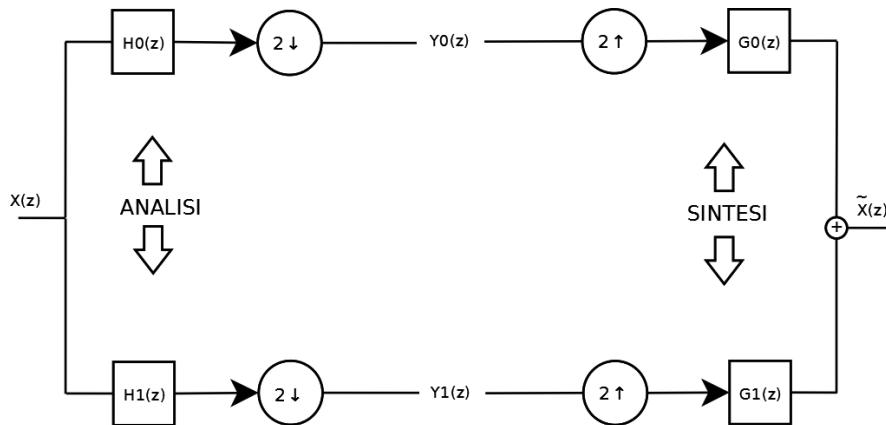


FIGURA 3.1: Schema concettuale di un banco di filtri.

Consideriamo ora lo schema di filtraggio dal punto di vista della trasformata z . L'analisi trasforma l'ingresso $X(z)$ in due uscite $Y_0(z)$ e $Y_1(z)$, mentre la sintesi le trasforma in $\tilde{X}(z)$. Il banco dei filtri si dice che gode della proprietà di *ricostruzione perfetta* se $X(z) = \tilde{X}(z)$ per tutte le $X(z)$ ad energia finita. Analizziamo le condizioni richieste ai quattro filtri affinché sia soddisfatta tale proprietà. Per procedere analizziamo l'operatore di sottocampionamento $2 \downarrow$ che nel dominio del tempo è definito al modo seguente:

$$x_{2\downarrow}[n] = x[2n] \quad n \in \mathbb{Z}, \quad (3.1)$$

in altre parole gli indici dispari vengono soppressi. La trasformata z di tale operatore sarà

$$\begin{aligned}
X_{2\downarrow}(z) &= \sum_n x[n]z^{-n} = \sum_n x[n+n]z^{-n} \\
&= \sum_n x[n+n]z^{-(n+n)+n} = \sum_n x[2n]z^{-2n}z^n \\
&= \sum_{k \text{ pari}} x[k]z^{-k}z^{k/2} = \sum_{k \text{ pari}} x[k]z^{-\frac{k}{2}}, \\
&= \sum_k x[k] \frac{z^{-k/2} + z^{-k/2}(-1)^{-k}}{2} \\
&= \frac{1}{2} \sum_k x[k](\sqrt{z})^{-k} + \frac{1}{2} \sum_k x[k](-\sqrt{z})^{-k} \\
&= \frac{1}{2} \left(X(z^{1/2}) + X(-z^{1/2}) \right). \tag{3.2}
\end{aligned}$$

Applicando la (3.2) al blocco di analisi otteniamo,

$$\begin{aligned}
Y_0(z) &= \frac{1}{2} (H_0(\sqrt{z})X(\sqrt{z}) + H_0(-\sqrt{z})X(-\sqrt{z})) \\
Y_1(z) &= \frac{1}{2} (H_1(\sqrt{z})X(\sqrt{z}) + H_1(-\sqrt{z})X(-\sqrt{z})). \tag{3.3}
\end{aligned}$$

Calcoliamo ora la trasformata z dell'operatore di sovracampionamento $2 \uparrow$,

$$\begin{aligned}
Y_{2\uparrow}(z) &= \sum_n y_{2\uparrow}[n]z^{-n} = \sum_{n \text{ pari}} y[\frac{n}{2}]z^{-n} \\
&= \sum_k y[k]z^{-2k} = Y(z^2), \tag{3.4}
\end{aligned}$$

con

$$y_{2\uparrow}[n] = \begin{cases} 0 & n \text{ dispari} \\ y[\frac{n}{2}] & n \text{ pari.} \end{cases}$$

Applicando ora il precedente risultato al blocco di sintesi otteniamo,

$$\begin{aligned}
\tilde{X}(z) &= G_0(z)Y(z^2) + G_1(z)Y_1(z^2) \\
&= \frac{1}{2} G_0(z) (H_0(z)X(z) + H_0(-z)X(-z)) + \\
&\quad \frac{1}{2} G_1(z) (H_1(z)X(z) + H_1(-z)X(-z)) \\
&= \frac{1}{2} (G_0(z)H_0(z) + G_1(z)H_1(z)) X(z) + \\
&\quad \frac{1}{2} (G_0(z)H_0(-z) + G_1(z)H_1(-z)) X(-z). \tag{3.5}
\end{aligned}$$

Infine dalla condizione di ricostruzione perfetta otteniamo,

$$G_0(z)H_0(z) + G_1(z)H_1(z) = 2 \quad (3.6a)$$

$$G_0(z)H_0(-z) + G_1(z)H_1(-z) = 0. \quad (3.6b)$$

Le (3.6a) e (3.6b) possono essere posta in forma matriciale,

$$\begin{pmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{pmatrix} \begin{pmatrix} G_0(z) \\ G_1(z) \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \quad (3.7)$$

Affinché il sistema di equazioni della (3.7) abbia soluzioni la matrice deve essere invertibile, posto quindi,

$$d(z) = \begin{vmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{vmatrix} \quad (3.8)$$

si ha $d(-z) = -d(z)$. Poiché $H(z)$ è un polinomio di *Laurent* si può dimostrare che $d(z)$ deve essere un monomio di *Laurent* con potenza dispari di z , e cioè

$$d(z) = 2c^{-1}z^{-2k-1}$$

con c costante da definire. Utilizzando la regola di *Cramer* il sistema (3.7) può essere risolto:

$$\begin{aligned} G_0(z) &= \frac{\begin{vmatrix} 2 & H_1(z) \\ 0 & H_1(-z) \end{vmatrix}}{d(z)} = cz^{2k+1}H_1(z) \\ G_1(z) &= \frac{\begin{vmatrix} H_0(z) & 2 \\ H_0(-z) & 0 \end{vmatrix}}{d(z)} = -cz^{2k+1}H_0(-z) \end{aligned} \quad (3.9)$$

Il banco di filtri con la condizione di ricostruzione perfetta può essere anche introdotto nel dominio del tempo utilizzando la nozione di convoluzione, a tal fine è necessario introdurre il concetto di basi ortogonali e biortogonali. Nello spazio dei segnali ad energia finita $l^2(\mathbb{Z})$ si definisce prodotto scalare, per ogni coppia $\mathbf{x}, \mathbf{y} \in l^2(\mathbb{Z})$, mediante la seguente formula

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{n \in \mathbb{Z}} \overline{x_n} y_n,$$

da cui la norma $\|x\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. Due vettori \mathbf{x}, \mathbf{y} sono detti appunto ortogonali se $\langle \mathbf{x}, \mathbf{y} \rangle = 0$. Una sequenza $\mathbf{e}_n, n \in \mathbb{Z}$, di vettori di $l^2(\mathbb{Z})$ è detta una base ortonormale se

$$1) \quad \forall m, n \in \mathbb{Z} \rightarrow \langle \mathbf{e}_n, \mathbf{e}_m \rangle = \delta[m - n];$$

$$2) \quad \text{se } \mathbf{x} \in \mathbb{Z} \text{ e } \langle \mathbf{x}, \mathbf{e}_n \rangle = 0 \quad \forall n \in \mathbb{Z} \rightarrow \mathbf{x} = 0.$$

La base canonica $\mathbf{e}_n[k] = \delta[k - n]$ è un esempio di base ortonormale. Tuttavia si può estendere questo concetto. Consideriamo due sequenze $\mathbf{f}_n, n \in \mathbb{Z}$, e $\tilde{\mathbf{f}}_m, m \in \mathbb{Z}$, tali che:

- 1) $\forall m, n \in \mathbb{Z} \rightarrow \langle \mathbf{f}_n, \tilde{\mathbf{f}}_m \rangle = \delta[m - n]$;
- 2) $\exists A, B, \tilde{A}$ e \tilde{B} , positive tali che $\forall x \in l^2(\mathbb{Z})$,

$$A\|\mathbf{x}\|^2 \leq \sum_n \|\langle \mathbf{f}_n, \mathbf{x} \rangle\|^2 \leq B\|\mathbf{x}\|^2$$

$$\tilde{A}\|\mathbf{x}\|^2 \leq \sum_n \|\langle \tilde{\mathbf{f}}_n, \mathbf{x} \rangle\|^2 \leq \tilde{B}\|\mathbf{x}\|^2$$

allora esse costituiscono una coppia di basi biortogonali. L'espansione canonica classica $\mathbf{x} = \sum_n \langle \mathbf{e}_n, \mathbf{x} \rangle \mathbf{e}_n$ viene sostituita dalle seguenti:

$$\mathbf{x} = \sum_n \langle \mathbf{f}_n, \mathbf{x} \rangle \tilde{\mathbf{f}}_n$$

$$\mathbf{x} = \sum_n \langle \tilde{\mathbf{f}}_n, \mathbf{x} \rangle \mathbf{f}_n$$

mentre il prodotto scalare diventa

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_n \langle \mathbf{x}, \mathbf{f}_n \rangle \langle \tilde{\mathbf{f}}_n, \mathbf{y} \rangle .$$

Le basi ortonormali diventano un caso particolare di quelle biortonormali quando $\tilde{\mathbf{f}}_n = \mathbf{f}_n$. Consideriamo ora i banchi di filtri relativi alla figura (3.1) e scriviamo le equazioni nel dominio del tempo,

$$y_0[k] = \sum_n h_0[2k - n]x[n] = \sum_n h_0[n]x[2k - n]$$

$$y_1[k] = \sum_n h_1[2k - n]x[n] = \sum_n h_1[n]x[2k - n], \quad (3.10)$$

dove l'indice $2k$ emerge dall'operazione di *downsample*. Proseguendo con lo schema (3.1) ed assumendo la condizione di ricostruzione perfetta otteniamo,

$$x[k] = \sum_n (g_0[k - sn]y_0[n] + g_1[k - 2n]y_1[n]). \quad (3.11)$$

Richiamiamo ora le condizioni espresse nel sistema (3.6) riscrivendole in modo diverso utilizzando le (3.9),

$$\begin{aligned}
 G_0(-z)H_0(-z) &= G_0(-z) \left(-c^{-1}z^{-(2k+1)}G_1(z) \right) = \\
 &= -cz^{2k+1}H_1(z) \left(-c^{-1}z^{-(2k+1)}G_1(z) \right) = \\
 &= H_1(z)G_1(z).
 \end{aligned} \tag{3.12}$$

Con l'ausilio della (3.12) si può riscrivere l'equazione (3.6a) ,

$$G_0(z)H_0(z) + G_0(-z)H_0(-z) = 2 \tag{3.13}$$

Notando inoltre che $H_0(-z)$ è la trasformata z di $\{(-1)^n h_0[n]\}$ e 2 è quella di $2\delta[n]$, la prima equazione possiamo riscriverla nel dominio del tempo al modo seguente,

$$\sum_k g_0[k]h_0[n-k] + (-1)^n \sum_k g_0[k]h_0[n-k] = 2\delta[n]. \tag{3.14}$$

Nella (3.14) i termini dispari si cancellano ed essa si semplifica in,

$$\sum_k g_0[k]h_0[2n-k] = \delta[n] \quad \forall n \in \mathbb{Z}. \tag{3.15}$$

E' necessario ora trovare delle relazioni simili per tutte le combinazioni h_0, h_1, g_0, g_1 . Utilizzando sempre la (3.12), la (3.6a) può essere posta in questa ulteriore forma,

$$G_1(z)H_1(z) + G_1(-z)H_1(-z) = 2, \tag{3.16}$$

e sfruttando ora il fatto che $H_1(-z)$ è la trasformata z di $(-1)^n h_1[n]$, otteniamo,

$$\sum_k g_1[k]h_1[n-k] + \sum_k g_1[k]h_1[n-k] = 2\delta[n], \tag{3.17}$$

da cui infine,

$$\sum_k g_1[k]h_1[2n-k] = \delta[n] \quad \forall n \in \mathbb{Z}. \tag{3.18}$$

Per i termini misti procediamo in modo del tutto analogo, quindi usando le (3.9) otteniamo,

$$\begin{aligned}
 G_0(z)H_1(z) &= -c^{-1}z^{-(2k+1)}G_0(z)G_0(-z) \\
 G_0(-z)H_1(-z) &= c^{-1}z^{-(2k+1)}G_0(z)G_0(-z),
 \end{aligned}$$

che sommate danno,

$$G_0(z)H_1(z) + G_0(-z)H_1(-z) = 0. \tag{3.19}$$

Trasportando ora la (3.19) nel dominio del tempo abbiamo,

$$\sum_k g_0[k]h_1[n-k] + (-1)^n \sum_k g_0[k]h_1[n-k] = 0,$$

da cui

$$\sum_k g_0[k]h_1[2n-k] = 0 \quad \forall n \in \mathbb{Z}. \quad (3.20)$$

Infine da

$$\begin{aligned} G_1(z)H_0(z) &= -cz^{2k+1}H_0(z)H_0(-z) \\ G_1(-z)H_0(-z) &= cz^{2k+1}H_0(z)H_0(-z) \end{aligned}$$

si ottiene,

$$G_1(z)H_0(z) + G_1(-z)H_0(-z) = 0, \quad (3.21)$$

e ripetendo lo stesso schema,

$$\sum_k g_1[k]h_0[n-k] + (-1)^n \sum_k g_1[k]h_0[n-k] = 0,$$

da cui

$$\sum_k g_1[k]h_0[2n-k] = 0 \quad \forall n \in \mathbb{Z}. \quad (3.22)$$

Ponendo ora,

$$\begin{aligned} f_{2n}[k] &= h_0[2n-k] \\ \tilde{f}_{2n}[k] &= g_0[2n-k] \\ f_{2n+1}[k] &= h_1[2n-k] \\ \tilde{f}_{2n+1}[k] &= g_1[2n-k] \end{aligned} \quad (3.23)$$

le relazioni (3.15),(3.18),(3.20) e (3.22) si possono esprimere in questa forma,

$$\begin{aligned} \langle \tilde{\mathbf{f}}_0, \mathbf{f}_{2n} \rangle &= \delta[n] \\ \langle \tilde{\mathbf{f}}_0, \mathbf{f}_{2n+1} \rangle &= 0 \\ \langle \tilde{\mathbf{f}}_1, \mathbf{f}_{2n} \rangle &= 0 \\ \langle \tilde{\mathbf{f}}_1, \mathbf{f}_{2n+1} \rangle &= \delta[n] \end{aligned} \quad (3.24)$$

L'estensione delle (3.24) porta alla condizione di biortogonalità:

$$\langle \tilde{\mathbf{f}}_i, \tilde{\mathbf{f}}_{i'} \rangle = \delta[i - i'] \quad (3.25)$$

Verifichiamo la (3.25) direttamente, partendo dal primo caso i, i' pari.

$$\begin{aligned} \langle \tilde{\mathbf{f}}_{2m}, \tilde{\mathbf{f}}_{2n} \rangle &= \sum_k g_0[k-2m]h_0[2n-k] \\ &= \sum_k g_0[k']h_0[2n-2m-k'] \\ &= \delta[n-m]. \end{aligned}$$

Consideriamo ora il caso dispari,

$$\begin{aligned} \langle \tilde{\mathbf{f}}_{2m+1}, \tilde{\mathbf{f}}_{2n+1} \rangle &= \sum_k g_1[k-2m]h_1[2n-k] \\ &= \sum_k g_1[k']h_1[2n-2m-k'] \\ &= \delta[n-m], \end{aligned}$$

ed infine i casi intermedi,

$$\begin{aligned} \langle \tilde{\mathbf{f}}_{2m+1}, \tilde{\mathbf{f}}_{2n} \rangle &= \sum_k g_1[k-2m]h_0[2n-k] \\ &= \sum_k g_1[k']h_0[2n-2m-k'] \\ &= 0, \end{aligned}$$

e

$$\begin{aligned} \langle \tilde{\mathbf{f}}_{2m}, \tilde{\mathbf{f}}_{2n+1} \rangle &= \sum_k g_0[k-2m]h_1[2n-k] \\ &= \sum_k g_0[k']h_1[2n-2m-k'] \\ &= 0. \end{aligned}$$

Per affermare che i vettori $\tilde{\mathbf{f}}_n$ e \mathbf{f}_m sono effettivamente delle basi biortonormali occorre verificare anche la condizione di stabilità, come riportato in [1]. E' inoltre importante sottolineare che dalle equazioni (3.23), oppure con quattro filtri che soddisfano le equazioni (3.15),(3.18),(3.20) e (3.22), è possibile dimostrare che le rispettive trasformate z , $H_0(z), H_1(z), G_0(z)$ e $G_1(z)$ soddisfano la condizione di perfetta ricostruzione (3.6). In altre parole le condizioni esaminate sono necessarie e sufficienti.

3.3 Proprietà dei filtri ortonormali

Un importante set di filtri che soddisfano le condizioni di ricostruzione perfetta sono i **filtri ortogonali**, che sono definiti dalla condizione

$$\tilde{\mathbf{f}}_n = \mathbf{f}_n \quad \forall n \in \mathbb{Z}, \quad (3.26)$$

che si esplicita in

$$\begin{aligned} g_0[k] &= h_0[-k] \quad \forall k \in \mathbb{Z} \\ g_1[k] &= h_1[-k] \quad \forall k \in \mathbb{Z}. \end{aligned} \quad (3.27)$$

Dalla (3.27) e dalle (3.9) si ricava che è necessario specificare un solo filtro. A tal fine ipotizziamo che la lunghezza sia $L > 2$ e riportiamo alcune importanti proprietà.

(A) La lunghezza di \mathbf{h}_0 è pari e vale $L = 2K$

Dalle (3.27) si ha

$$\begin{aligned} \sum_k g_0[k] h_0[2n - k] &= \delta[n] \\ &= \sum_k h_0[-k] h_0[2n - k] \\ &= \sum_k h_0[k] h_0[2n + k]. \end{aligned} \quad (3.28)$$

Ipotizzando per assurdo che $L = 2N + 1$ ed assumendo che il filtro abbia la seguente forma,

$$\mathbf{h}_0 = [0, \dots, 0, h_0[0], \dots, h_0[2N], 0, \dots]$$

con $h_0[0] \neq 0$ e $h_0[2N] \neq 0$ si ricava che $h_0[0]h_0[2N] \neq 0$ in contrasto con la (3.28).

(B) $H_1(z)$ può essere espresso come $H_1(z) = -z^{-2K-1}H_0(-z^{-1})$, e se $h_0[1] \neq 0, \dots, h_0[2K] \neq 0$ allora $h_1[k] = (-1)^k h_0[2K + 1 - k]$ con $k = 1, 2, \dots, 2K$.

L'assunzione $g_0[k] = h_0[-k]$ implica $G_0(z) = H_0(z^{-1})$, usando ora la (3.9) otteniamo,

$$H_1(z) = -c^{-1}z^{-2k-1}G_0(-z) = -c^{-1}z^{-2k-1}H_0(-z^{-1}).$$

Ponendo $c = 1$ e $k = K$ otteniamo proprio l'espressione voluta per $H_1(z)$, che automaticamente si esprime nel dominio del tempo come ipotizzato.

(C) Vale l'equazione complementare della potenza: $|H_0(e^{j\omega})|^2 + |H_1(e^{j\omega})|^2 = 2 \forall \omega \in \mathbb{R}$.

Consideriamo la (3.6a) e imponiamo la condizione di ortogonalità assieme a $z = e^{j\omega}$, otteniamo

$$H_0(e^{j\omega})H_0(e^{-j\omega}) + H_0(-e^{j\omega})H_0(-e^{-j\omega}) = 2. \quad (3.29)$$

Se nella (3.29) consideriamo il fatto che i coefficienti del filtro sono reali possiamo appunto scrivere

$$H_0(e^{j\omega})H_0(e^{-j\omega}) = |H_0(e^{j\omega})|^2.$$

Dalla proprietà **B** ricaviamo che $H_0(-z^{-1}) = -z^{2K+1}H_1(z)$ e cambiando z con z^{-1} otteniamo $H_0(-z) = -z^{-2K-1}H_1(z^{-1})$, da cui infine

$$H_0(-z^{-1})H_0(-z) = H_1(z)H_1(z^{-1}).$$

Inoltre posto $z = 1$ si ha $H_0(1)H_0(1) + H_0(-1)H_0(-1) = 2$ e ipotizzando $H_0(-1) = 0$ si ricava $H_0(1) = \sqrt{2}$.

(D) Vale la conservazione dell'energia $\|x\|^2 = \|y_0\|^2 + \|y_1\|^2$.

Consideriamo l'equazione (3.19) ed utilizziamo la condizione di ortogonalità $G_0(z) = H_0(z^{-1})$ per ricavare la seguente espressione,

$$H_0(z^{-1})H_1(z) + H_0(-z^{-1})H_1(-z) = 0 \quad (3.30)$$

e sostituendo z con z^{-1} , si ricava,

$$H_0(z)H_1(z^{-1}) + H_0(-z)H_1(-z^{-1}) = 0. \quad (3.31)$$

Imponiamo ora nella (3.31) la sostituzione $z = e^{j\omega}$ e ricaviamo

$$H_0(e^{j\omega})\overline{H_1(e^{j\omega})} + H_0(e^{j(\omega+\pi)})\overline{H_1(e^{j(\omega+\pi)})} = 0, \quad (3.32)$$

dove si sono utilizzate le seguenti espressioni $z^{-1} = \overline{e^{j\omega}}$ e $-z = e^{j(\omega+\pi)}$. Analogamente partendo dalla relazione $H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 2$ (eq. (3.29)) e usando la

proprietà **(B)** otteniamo

$$H_1(z)H_1(z^{-1}) + H_1(-z)H_1(-z^{-1}) = 0, \quad (3.33)$$

da cui

$$H_1(e^{j\omega})\overline{H_1(e^{j\omega})} + H_1(e^{j(\omega+\pi)})\overline{H_1(e^{j(\omega+\pi)})} = 2. \quad (3.34)$$

Utilizzando le equazioni (3.32) e (3.34) verifichiamo che la seguente matrice è unitaria,

$$U(e^{j\omega}) = \frac{1}{2} \begin{pmatrix} H_0(e^{j\omega}) & H_0(e^{j(\omega+\pi)}) \\ H_1(e^{j\omega}) & H_1(e^{j(\omega+\pi)}) \end{pmatrix}, \quad (3.35)$$

infatti il valore assoluto del determinante è

$$\begin{aligned} |\det U(e^{j\omega})| &= \frac{1}{2} \left| H_0(e^{j\omega})H_1(e^{j(\omega+\pi)}) - H_0(e^{j(\omega+\pi)})H_1(e^{j\omega}) \right| \\ &= \frac{1}{2} \left| H_0(e^{j\omega})H_1(e^{j(\omega+\pi)}) + H_0(e^{j\omega})\frac{H_1(e^{j\omega})\overline{H_1(e^{j\omega})}}{H_1(e^{j(\omega+\pi)})} \right| \\ &= \left| \frac{H_0(e^{j\omega})}{H_1(e^{j(\omega+\pi)})} \right| = \left| e^{-j\omega(2K+1)} \right| = 1. \end{aligned}$$

Riscriviamo ora le equazioni (3.3) con la sostituzione $z = e^{j\omega}$,

$$\begin{aligned} Y_0(e^{j\omega}) &= \frac{1}{2} \left(H_0(e^{j\frac{\omega}{2}})X(e^{j\frac{\omega}{2}}) + H_0(e^{j(\frac{\omega}{2}+\pi)})X(e^{j(\frac{\omega}{2}+\pi)}) \right) \\ Y_1(e^{j\omega}) &= \frac{1}{2} \left(H_1(e^{j\frac{\omega}{2}})X(e^{j\frac{\omega}{2}}) + H_1(e^{j(\frac{\omega}{2}+\pi)})X(e^{j(\frac{\omega}{2}+\pi)}) \right), \end{aligned} \quad (3.36)$$

e introduciamo la matrice (3.37),

$$\begin{pmatrix} Y_0(e^{j\omega}) \\ Y_1(e^{j\omega}) \end{pmatrix} = U(e^{j\frac{\omega}{2}}) \begin{pmatrix} \frac{1}{\sqrt{2}}X(e^{j\frac{\omega}{2}}) \\ \frac{1}{\sqrt{2}}X(e^{j(\frac{\omega}{2}+\pi)}) \end{pmatrix}. \quad (3.37)$$

Poiché la trasformazione unitaria preserva la norma otteniamo che

$$|Y_0(e^{j\omega})|^2 + |Y_1(e^{j\omega})|^2 = \frac{1}{2} \left(|X(e^{j\frac{\omega}{2}})|^2 + |X(e^{j(\frac{\omega}{2}+\pi)})|^2 \right). \quad (3.38)$$

Per ogni sequenza $\mathbf{x} \in l^2(\mathbb{Z})$ vale l'equazione di Parseval, quindi possiamo scrivere le seguenti relazioni,

$$\begin{aligned} \|y_0\|^2 + \|y_1\|^2 &= \frac{1}{2\pi} \int_0^{2\pi} (|Y_0(e^{j\omega})|^2 + |Y_1(e^{j\omega})|^2) d\omega \\ &= \frac{1}{4\pi} \int_0^{2\pi} \left(|X(e^{j\frac{\omega}{2}})|^2 + |X(e^{j(\frac{\omega}{2}+\pi)})|^2 \right) d\omega, \end{aligned}$$

che con la sostituzione $\frac{\omega}{2} = \omega'$, infine diventa

$$\begin{aligned} \|y_0\|^2 + \|y_1\|^2 &= \frac{1}{2\pi} \int_0^\pi \left(|X(e^{j\frac{\omega'}{2}})|^2 + |X(e^{j(\frac{\omega'}{2}+\pi)})|^2 \right) d\omega' \\ &= \frac{1}{2\pi} \int_0^{2\pi} (|X(e^{j\omega'})|^2) d\omega' \\ &= \|x\|^2. \end{aligned}$$

3.4 Filtri di Daubechies o *Maxflat Filters*

I filtri di Daubechies costituiscono una classe molto importante di basi ortonormali e si basano appunto su due proprietà chiave:

- ortogonalità;
- massima piatezza in $\omega = 0$ e $\omega = \pi$.

Assegniamo ora alla quantità $|H_0(e^{j\omega})|^2$ il simbolo $P(\omega)$ che identifica la risposta dello spettro di potenza, esso è simmetrico essendo $P(\omega) = H_0(e^{j\omega})\overline{H_0(e^{j\omega})}$, ed è esprimibile al modo seguente,

$$P(\omega) = \sum_{n=-N}^{+N} p[n]e^{-jn\omega} = \left(\sum_{n=0}^N h_0[n]e^{-jk\omega} \right) \left(\sum_{n=0}^N h_0[n]e^{jk\omega} \right), \quad (3.39)$$

con $p(n) = p(-n)$. Ricordiamo che in base alla proprietà **(A)** del paragrafo precedente il numero di coefficienti è $2p$, quindi $N = 2p - 1$. Essendo inoltre $p[n]$ una sequenza di autocorrelazione vale $p[n] = \sum_n h_0[n]h_0[k-n]$ e pertanto sarà $p[0] = \sum_n h_0[n]^2$. Richiamiamo ora la condizione (3.15) assieme alla (3.27) e ricaviamo la seguente relazione

$$\sum_n h_0[n]h_0[n-2k] = \delta[k], \quad (3.40)$$

da cui

$$\sum_n h_0[n]^2 = 1 \rightarrow p[0] = 1. \quad (3.41)$$

L'equazione (3.40) se posta in questa forma, $p[2k] = \sum_n h_0[n]h_0[n-2k] = \delta[k]$ ci suggerisce che tutti gli elementi pari di $p[n]$ sono nulli. Quindi dalla condizione di ortogonalità ricaviamo p condizioni così esprimibili,

$$p(0) = 1, \quad p(2) = p(4) = \dots = p(2p-2) = 0 \quad (3.42)$$

Se consideriamo ora la relazione,

$$P(\omega) + P(\omega + \pi) = 2 \quad (3.43)$$

ed esplicitiamo i termini,

$$P(\omega + \pi) = \sum_{n=-N}^N p[n]e^{-jn(\omega+\pi)} = \sum_{n=-N}^N (-1)^n p[n]e^{-jn\omega}$$

otteniamo

$$P(\omega) + P(\omega + \pi) = \sum_n [1 + (-1)^n] p[n]e^{-jn\omega}. \quad (3.44)$$

Dall'espressione (3.44) si deduce che i termini dispari si elidono ma ciò non aggiunge alcuna informazione. Esaminiamo ora la condizione di massima piatezza e ricordiamo che per un polinomio $\tilde{H}(y)$ di grado $2p - 1$ essa è così definita,

- $\tilde{H}^{(k)}(0) = \delta[k]$ per $0 \leq k < p$;
- $\tilde{H}^{(k)}(1) = 0$ per $0 \leq k < p$.

Il secondo vincolo impone che la forma sia del tipo $\tilde{H}^{(k)}(y) = (1-y)^p Q(y)$, mentre le p condizioni in $y = 0$ determinano la funzione $Q(y)$. In pratica quindi i $2p$ numeri necessari a formare il filtro vengono forniti equamente dalle p condizioni dell'ortogonalità e dalle p condizioni della piatezza. Le relazioni tra y e le variabili ω e z sono,

$$\begin{aligned} y &= \frac{1 - \cos \omega}{2} = \frac{1}{4}(2 - z - z^{-1}) = -z \left(\frac{1 - z^{-1}}{2} \right)^2 \\ 1 - y &= \frac{1 + \cos \omega}{2} = \frac{1}{4}(2 + z - z^{-1}) = -z \left(\frac{1 + z^{-1}}{2} \right)^2, \end{aligned} \quad (3.45)$$

pertanto le condizioni al contorno saranno,

$$\begin{aligned} y = 1 & \quad \omega = \pi & \quad z = -1 \\ y = 0 & \quad \omega = 0 & \quad z = 1. \end{aligned} \quad (3.46)$$

Applichiamo ora questi risultati al polinomio H_0 , cominciando dalla condizione sugli zeri di ordine p :

$$H_0(e^{j\pi}) = H_0^{(1)}(e^{j\pi}) = \dots = H_0^{(p-1)}(e^{j\pi}) = 0 \quad (3.47)$$

$$H_0(e^{j\pi}) = 0 \quad \rightarrow \quad \sum_n h_0[n](-1)^n = 0 \quad \rightarrow \quad \sum_{\text{PARI}} h_0[n] = \sum_{\text{DISPARI}} h_0[n]. \quad (3.48)$$

Inoltre poiché $|H_0(e^{j\pi})|^2 = 0$ allora anche $P(\omega) = 0$ da cui segue che $\sum_{\text{PARI}} p[n] = \sum_{\text{DISPARI}} p[n]$. Però ricordando le condizioni derivanti dall'ipotesi di ortogonalità, abbiamo la seguente relazione,

$$\sum_{\text{PARI}} p[n] = p[0] = 1 \quad \rightarrow \quad \sum_{\text{DISPARI}} p[n] = 1, \quad (3.49)$$

e dalla (3.49) segue immediatamente che

$$P(0) = \sum_n p[n] = \sum_{\text{PARI}} p[n] + \sum_{\text{DISPARI}} p[n] = 2, \quad (3.50)$$

e quindi

$$H_0(1) = \sqrt{2} \quad (3.51)$$

come annunciato nella proposizione (C) del paragrafo 3.3. Inoltre in accordo con quanto svolto la forma del polinomio sarà del tipo,

$$H_0(e^{j\pi}) = \left(\frac{1 + e^{-j\pi}}{2} \right)^p R(e^{j\omega}), \quad (3.52)$$

da cui segue che $P(\omega)$ deve contenere un fattore del tipo $\left(\frac{1 + \cos \omega}{2} \right)^p$. Consideriamo nuovamente $\tilde{P}(y) = 2(1 - y)^p B_p(y)$ ed esaminiamo l'idea base di Ingrid Debauchies, che consiste nella scelta di una opportuna approssimazione del polinomio $B_p(y)$,

$$\tilde{P}(y) = 2(1 - y)^p B_p(y) = 2(1 - y)^p ((1 - y)^{-p} + O(y^p)). \quad (3.53)$$

Osserviamo infatti che sicuramente $\tilde{P}(0) = 2$ e che tutte le sue derivate sono nulle come richiesto dalla condizione di massima piatezza. Inoltre esso soddisfa anche le seguenti relazioni

$$\tilde{P}(y) + \tilde{P}(1 - y) = 2 \quad (3.54)$$

e

$$\tilde{P}'(y) = -C y^{p-1} (1 - y)^{p-1} \quad C \text{ costante.} \quad (3.55)$$

La (3.55) è assicurata dal fatto che $\tilde{P}'(y)$ possiede $p - 1$ zeri agli estremi ed è un polinomio di grado $2p - 2$. Se ora trasliamo queste espressione nel dominio ω otteniamo,

$$P(\omega) = 2 \left(\frac{1 + \cos \omega}{2} \right)^{2p-1} \sum_{k=0}^{p-1} \binom{p+k-1}{k} \left(\frac{1 - \cos \omega}{2} \right)^k, \quad (3.56)$$

e

$$P'(\omega) = -c (\sin \omega)^{2p-1} \quad (3.57)$$

Come atteso la (3.56) soddisfa la relazione (3.43). Osserviamo inoltre che poiché la derivata di $P(\omega)$ è sempre negativa e che $P(0) = 2$ e $P(\pi) = 0$ allora $P(\omega) \geq 0$ come atteso dalla (3.40).

Inoltre $P(\pi/2) = 1/2$ e quindi effettivamente $P(\omega)$ rappresenta la decomposizione spettrale di un filtro ortonormale a mezza banda.

L'ultimo passo consiste nel passare alla variabile z per calcolare i coefficienti del filtro FIR. Quindi applicando le (3.45) otteniamo,

$$P(z) = 2\left(\frac{1+z}{2}\right)^p \left(\frac{1+z^{-1}}{2}\right)^p \sum_{k=0}^{p-1} \binom{p+k-1}{k} \left(\frac{1-z}{2}\right)^k \left(\frac{1-z^{-1}}{2}\right)^k. \quad (3.58)$$

Il polinomio (3.58) è centrato sullo zero quindi per renderlo causale è necessario eseguire uno shift, ottenendo $P_0(z) = z^{-N}P(z) = z^{1-2p}P(z)$. Per la costruzione dei polinomi $H_0(z)$ e $H_1(z)$ quindi occorre proseguire con i seguenti passi:

- (1) trovare i $p-1$ zeri dell'espressione $\sum_{k=0}^{p-1} \binom{p+k-1}{k} \left(\frac{1-z}{2}\right)^k \left(\frac{1-z^{-1}}{2}\right)^k$ relativi a $|z| < 1$, in quanto il filtro deve essere a fase minima (gli zeri devono stare all'interno del cerchio unitario);
- (2) inserire p zeri in $z = -1$ mediante il termine $\left(\frac{1+z^{-1}}{2}\right)^p$;
- (3) inserire una costante moltiplicativa α e determinarla mediante la condizione di normalizzazione $H_0(1) = \sqrt{2}$;
- (4) determinare i coefficienti di $H_1(z)$ mediante la proprietà **(B)**.

Il primo punto si basa sulla fattorizzazione classica di *Fejer-Riesz*, che appunto prevede il calcolo degli zeri del polinomio $P(z)$, in particolare nel caso di coefficienti reali simmetrici vale $P(z) = P(1/z)$ e quindi se z_i è uno zero lo è anche $1/z_i$, tuttavia nel caso in esame i valori entro il cerchio unitario generano un filtro a fase minima. Pertanto posti gli z_i tali che $|z_i| < 1$ allora segue che,

$$H_0(z) = \alpha \left(\frac{1+z^{-1}}{2}\right)^p \prod_{i=1}^{p-1} (1 - z^{-1}z_i) \quad (3.59)$$

con

$$\alpha = \frac{\sqrt{2}}{\prod_{i=1}^{p-1} (1 - z_i)} \quad (3.60)$$

Per verifica proviamo a calcolare i coefficienti dei filtri di Daubechies D4 ed a compararli con quelli calcolati dalla *toolbox wavelet* di *matlab*. Consideriamo quindi il caso $p = 2$, quindi il polinomio $H_0(z)$ deve avere $2p - 1 = 3$ zeri. Seguendo quindi lo schema delineato, otteniamo

il seguente polinomio

$$\sum_{k=0}^{p-1} \binom{p+k-1}{k} \left(\frac{1-z}{2}\right)^k \left(\frac{1-z^{-1}}{2}\right)^k = \frac{-z+4-z^{-1}}{2} = \frac{-z^2+4z-1}{2z}$$

che ha due zeri in $z = 2 \pm \sqrt{3}$. Poiché deve essere $|z_i| < 1$ allora consideriamo solo $z_i = 2 - \sqrt{3}$ e costruiamo l'espressione (3.59),

$$\begin{aligned} H_0(z) &= \alpha \left(\frac{1+z^{-1}}{2}\right)^2 (1 - (2 - \sqrt{3})z^{-1}) \\ &= \alpha \left(1 + \sqrt{3}z^{-1} + (-3 + 2\sqrt{3})z^{-2} + (-2 + \sqrt{3})z^{-3}\right), \end{aligned}$$

con $\alpha = \frac{\sqrt{2}}{-4+4\sqrt{3}} = \frac{1+\sqrt{3}}{4\sqrt{2}}$. In definitiva l'espressione diventa,

$$H_0(z) = \frac{1 + \sqrt{3}}{4\sqrt{2}} + \frac{3 + \sqrt{3}}{4\sqrt{2}}z^{-1} + \frac{3 - \sqrt{3}}{4\sqrt{2}}z^{-2} + \frac{1 - \sqrt{3}}{4\sqrt{2}}z^{-3}. \quad (3.61)$$

Per completare lo schema è necessario applicare la proprietà **(B)** e ricavare il filtro passa alto $H_1(z)$. Posto quindi,

$$\begin{aligned} h_0(0) &= \frac{1 + \sqrt{3}}{4\sqrt{2}} \\ h_0(1) &= \frac{3 + \sqrt{3}}{4\sqrt{2}} \\ h_0(2) &= \frac{3 - \sqrt{3}}{4\sqrt{2}} \\ h_0(3) &= \frac{1 - \sqrt{3}}{4\sqrt{2}}, \end{aligned}$$

secondo la regola dell'alternanza otteniamo,

$$\begin{aligned} h_1(0) &= (-1)^0 h_0(3) = \frac{1 - \sqrt{3}}{4\sqrt{2}} \\ h_1(1) &= (-1)^1 h_0(2) = -\frac{3 - \sqrt{3}}{4\sqrt{2}} \\ h_1(2) &= (-1)^2 h_0(1) = \frac{3 + \sqrt{3}}{4\sqrt{2}} \\ h_1(3) &= (-1)^3 h_0(0) = -\frac{1 + \sqrt{3}}{4\sqrt{2}}, \end{aligned}$$

da cui ricaviamo l'espressione per $H_1(z)$,

$$H_1(z) = \frac{1 - \sqrt{3}}{4\sqrt{2}} - \frac{3 - \sqrt{3}}{4\sqrt{2}}z^{-1} + \frac{3 + \sqrt{3}}{4\sqrt{2}}z^{-2} - \frac{1 + \sqrt{3}}{4\sqrt{2}}z^{-3}. \quad (3.62)$$

Il risultato ottenuto coincide con quello di *matlab* [59]. Nella figura (3.2) è riportata la risposta in frequenza di $H_0(\omega)$ e $H_1(\omega)$. Come si può osservare i filtri presentano un andamento perfettamente simmetrico essi vengono anche chiamati filtri a mezza banda. Come è intuibile, salendo

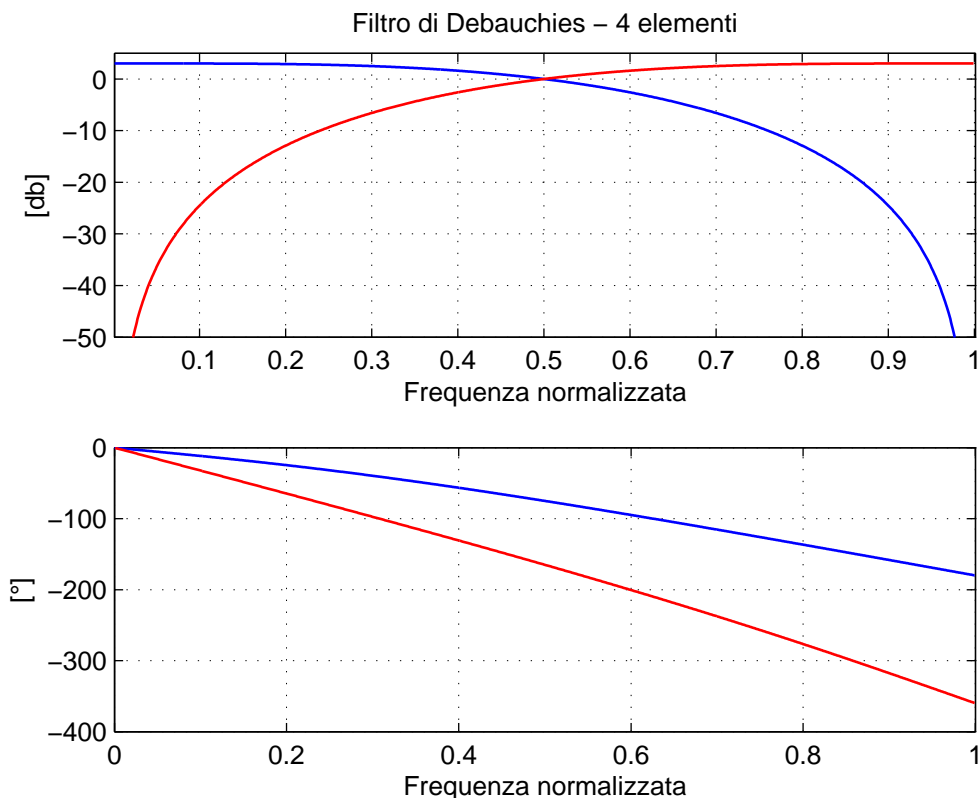


FIGURA 3.2: Risposta in frequenza.

con l'ordine del filtro (figura 3.3) la transizione diventa più evidente.

3.5 Analisi Multi-risoluzione

L'idea introdotta nel paragrafo precedente si presta ad una continua iterazione, come infatti possiamo notare dallo schema (3.4), il banco di filtri può essere applicato ad $Y_0(z)$ e a $Y_1(z)$. Chiaramente il processo può essere iterato ancora fino alla disponibilità di elementi numerici su cui far agire il filtro.

Da un punto di vista matematico l'idea della multi-risoluzione si basa sul concetto che una funzione possa essere decomposta in spazi con scale progressivamente più fini. Sia quindi V_j un

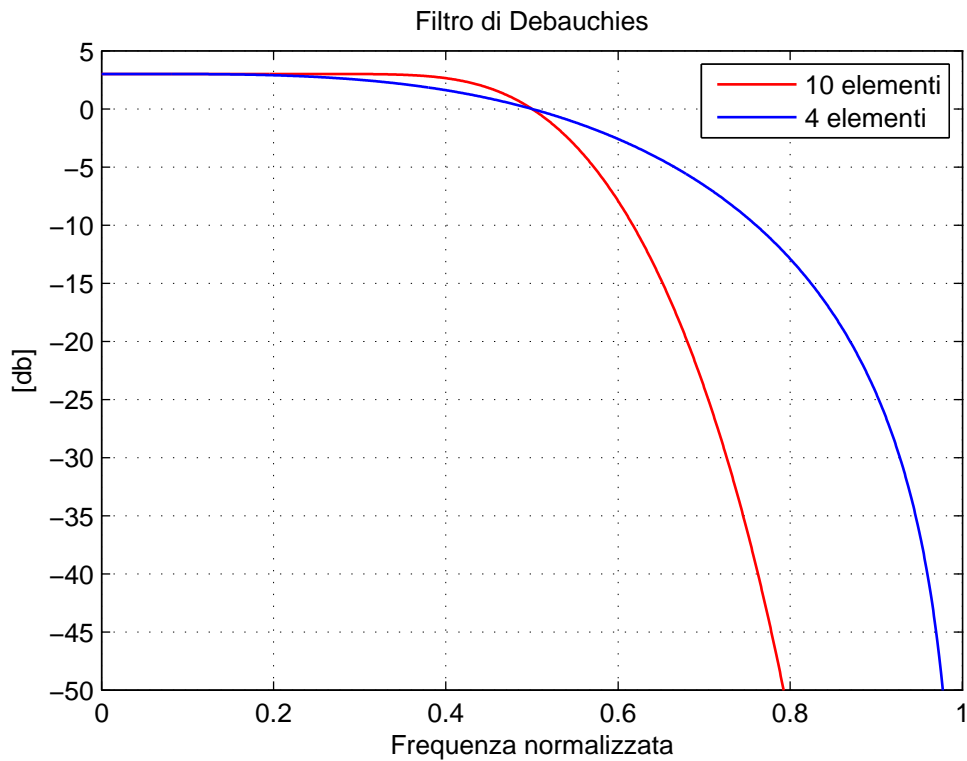


FIGURA 3.3: Confronto tra filtri di Debuachies.

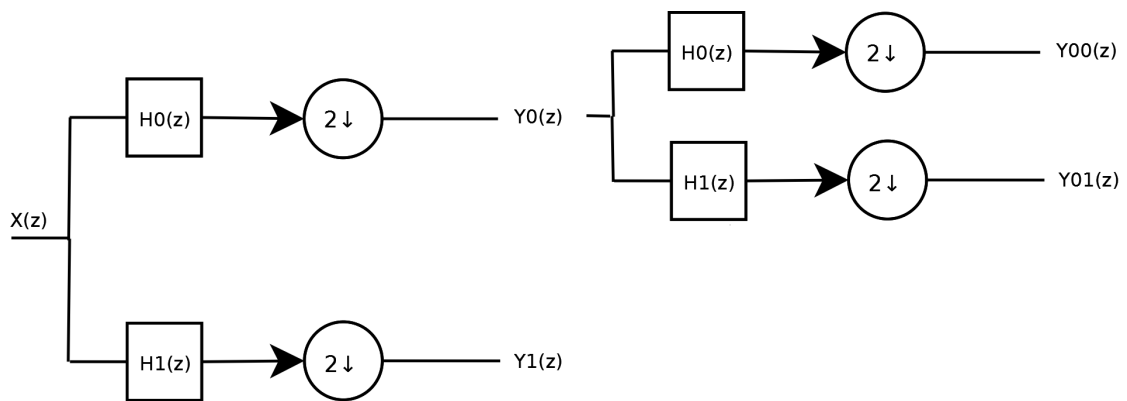


FIGURA 3.4: Iterazione logaritmica del banco di filtri.

tale sottospazio allora V_{j+1} rappresenta un sottospazio ad una scala più fine. La differenza tra V_j e V_{j+1} è data da W_j . In sostanza quindi se una funzione sta in un sottospazio allora essa sarà contenuta in tutti i sottospazi con risoluzione maggiore, pertanto possiamo supporre l'esistenza di una sequenza di V_j tale che

$$V_0 \subset V_1 \subset V_2 \subset \dots \subset V_j \subset \dots$$

Occorrerà inoltre richiedere che la sequenza sia compatta nel senso che le singole proiezioni

convergono alla funzione $f(t)$,

$$\lim_{j \rightarrow \infty} f_j(t) = f(t)$$

come nel caso dello sviluppo in serie di Fourier. L'altra famiglia di sottospazi sarà data dalle differenze $f_{j+1}(t) = f_j(t) + \Delta f_j(t)$ e $\Delta f_j(t)$ dettaglio alla scala j . Avremo pertanto $V_j \oplus W_j = V_{j+1}$ Seguendo questa strategia possiamo scrivere

$$V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \cdots \oplus W_j = V_{j+1} \quad (3.63)$$

e dal punto di vista delle funzioni otteniamo

$$f_0(t) + \Delta f_0(t) + \Delta f_1(t) + \cdots + \Delta f_j(t) = f_{j+1}(t). \quad (3.64)$$

La condizione di ortogonalità tra V_j e W_j è conveniente ma non è essenziale, come infatti anticipato nel caso delle basi *biortogonali*, l'importante è che siano verificate le seguenti condizioni,

$$\begin{aligned} V_j \cap W_j &= 0 \\ V_j + W_j &= V_{j+1}. \end{aligned}$$

Possiamo quindi riassumere le proprietà dei sottospazi V_j

- $V_j \subset V_{j+1}, \bigcup V_j = \{0\}$ e $\overline{\bigcap V_j} = L^2$;
- invarianza di scala: $f(t) \in V_j \iff f(2t) \in V_{j+1}$;
- invarianza di traslazione: $f(t) \in V_0 \iff f(t - k) \in V_0$;
- invarianza di traslazione per le basi: in V_0 esiste una base di *Riesz* $\{\phi(t - k)\}$.

Data quindi la base $\{\phi(t - k)\}$ in V_0 allora in V_j sarà $\phi_{jk}(t) = 2^{j/2}\phi(2^j t - k)$ e la proiezione di $f(t)$ in questo sottospazio sarà $f_j(t) = \sum_{k=-\infty}^{\infty} a_{jk}\phi_{jk}(t)$. Una diretta conseguenza di queste ipotesi è la **equazione di scala** o **dilation equation**, infatti poiché $V_0 \in V_1$, segue che

$$\phi(t) = \sqrt{2} \sum_k c(k)\phi(2t - k). \quad (3.65)$$

L'idea che suggerisce la (3.65) è che $\phi(t)$ venga risolta con miglior dettaglio ad una scala inferiore. I coefficienti $c(n)$ si calcoleranno al modo seguente,

$$\int_{-\infty}^{+\infty} \phi(t)\sqrt{2}\phi(2t - n)dt = c(n). \quad (3.66)$$

Si ricava inoltre che le funzioni $\phi(2^j t - k)$ sono ortogonali in V_j ma non in V_k , con $k \neq j$. Come anticipato gli spazi delle *wavelet* sono ortogonali a V_j ed analogamente alla (3.65) (poiché $W_0 \in V_1$) varrà la seguente relazione,

$$w(t) = \sqrt{2} \sum_k d(k) \phi(2t - k). \quad (3.67)$$

La (3.67) prende il nome di **wavelet equation**. Inoltre poiché $W_j \perp V_1$ per $j \geq 1$ allora sarà valida anche questa relazione

$$\int_{-\infty}^{+\infty} w(t) w(2^j t - k) dt = 0. \quad (3.68)$$

Se ora consideriamo la proiezione $f_1(t)$ e la esplicitiamo dal punto di vista della multi-risoluzione, $V_1 = V_0 \oplus W_0$, otteniamo

$$\begin{aligned} \sum a_{1k} \phi_{1k}(t) &= \sum a_{0k} \phi_{0k}(t) + \sum b_{0k} w_{0k} \\ &= \sum a_{0k} \phi(t - k) + \sum b_{0k} w(t - k). \end{aligned} \quad (3.69)$$

La (3.69) suggerisce che si possano appunto calcolare i coefficienti a_0 e b_0 in funzione di a_1 . Consideriamo per semplicità il caso di basi ortonormali: se si fa il prodotto scalare tra $f_1(t)$ e $\phi(t - k)$ e $w(t - k)$ usando rispettivamente la eq. (3.65) e la eq. (3.67) si ottengono le seguenti equazioni ricorsive,

$$\begin{aligned} a_{0k} &= \sum c(l - 2k) a_{1l} \\ b_{0k} &= \sum d(l - 2k) a_{1l} \end{aligned}$$

Esse coincidono con le equazioni (3.10) del banco di filtri. In generale si può dimostrare che vale

$$\begin{aligned} a_{jk} &= \sum c(l - 2k) a_{j+1,l} \\ b_{jk} &= \sum d(l - 2k) a_{j+1,l}, \end{aligned}$$

e questo schema è equivalente a quello rappresentato in figura 3.4. Naturalmente vale anche il processo inverso (*sintesi*),

$$\begin{aligned} a_{j+1,l} &= \sum c(2k - 1) a_{j,k} \\ b_{j+1,l} &= \sum d(2k - l) a_{j,k}. \end{aligned}$$

E' possibile dimostrare che un banco di filtri genera lo spazio V_j con le caratteristiche richieste e viceversa.

3.6 Wavelet Packets

L'applicazione naturale dell'approccio *multirisoluzione* consiste nella costruzione di una iterazione del solo ramo passa basso. Tuttavia il processo può essere anche applicato contemporaneamente al ramo passa alto come visualizzato in figura 3.5. Chiaramente si genera una sovrabbondanza di informazioni, più generalmente tale decomposizione include molte rappresentazioni equivalenti del segnale ed ognuna di essa corrisponde ad una scelta delle basi. Consideriamo una decomposizione contenente $j + 1$ livelli e chiamiamo A_j il numero di basi (o rappresentazioni). Ogni volta che scegliamo una rappresentazione nel ramo di sinistra della figura 3.6 possiamo sceglierne un'altra in quello di destra. In totale quindi avremo A_j^2 set di basi ma a queste dobbiamo aggiungere anche il segnale originale, quindi

$$A_{j+1} = A_j^2 + 1. \quad (3.70)$$

La (3.70) mostra come le rappresentazioni crescano in modo drammatico con i livelli. Posto ora

$$\begin{aligned} B_{j+1} &= B_j^2 \quad B_1 = 2 \\ C_{j+1} &= B_{j+1}^2 \quad C_1 = 2, \end{aligned}$$

si ha

$$B_j \leq A_j \leq C_j.$$

E' possibile tuttavia esplicitare B_j e C_j , infatti

$$\begin{aligned} B_j &= B_{j-1}^2 = (B_{j-2}^2)^2 = \dots = 2^{2^{j-1}} \\ C_j &= (B_j)^2 = \dots = 2^{2^j}, \end{aligned}$$

da cui ricaviamo i limiti per il numero di rappresentazioni,

$$2^{2^{j-1}} \leq A_j \leq 2^{2^j}.$$

E' chiaro che occorre stabilire un criterio per decidere qual è la miglior rappresentazione del segnale e generalmente questo problema si risolve introducendo una funzione costo K . A livello teorico è possibile indicare una procedura generale per risolvere il problema. Come primo passo si calcola il costo di tutti gli elementi della decomposizione fino al livello prescritto J , poi si eseguono i seguenti passaggi:

- 1 marcare tutti gli elementi al livello più basso J ;
- 2 porre $j=J$;
- 3 porre $k=0$;

4 Comparare il costo c_1 dell'elemento k al livello $j - 1$, a partire dall'alto nello schema 3.5 con la somma c_2 del costo degli elementi $2k$ e $2k + 1$ al livello j :

[4.1] se $c_1 \leq c_2$ allora spostare la marcatura sull'elemento k al livello $j - 1$;

[4.2] se $c_1 > c_2$ allora per l'elemento k in $j - 1$ porre $c_2 = c_1$ (questo serve per propagare l'informazione su c_2 ai livelli superiori);

5 incrementare k e ritornare al punto 4 altrimenti passare al punto successivo;

6 decrementare j e ritornare al punto 3.

Il procedimento precedente prevede di scegliere come base di riferimento l'ultimo livello di decomposizione e successivamente risalire, comparando i costi, verso il segnale originale. Il cuore della procedura consiste nel confrontare il costo di un elemento con i due immediatamente sottostanti nel ramo della decomposizione: se la somma del costo di questi elementi è superiore allora l'elemento superiore è considerato come membro della nuova base. Le funzioni costo utili sono quelle che in senso lato misurano la concentrazione come ad esempio la norma o l'entropia, questo perché in effetti un basso costo implica un'alta concentrazione e questo in sostanza significa che ci sono pochi numeri di valore elevato e molti di piccola quantità. Una funzione costo elementare è basata sulla soglia, ovvero sul conteggio di quanti numeri superano un determinato limite. In alternativa misure più sofisticate possono essere:

- $K_{L^p}(a) = \sum_n |a[n]|^p$;
- $K_{Shannon}(a) = - \sum_n a[n]^2 \log(a[n]^2)$.

La proprietà di additività richiesta per le funzioni costo assicura una rapida convergenza mentre la complessità dell'algorithmo è $O(J \log J)$.

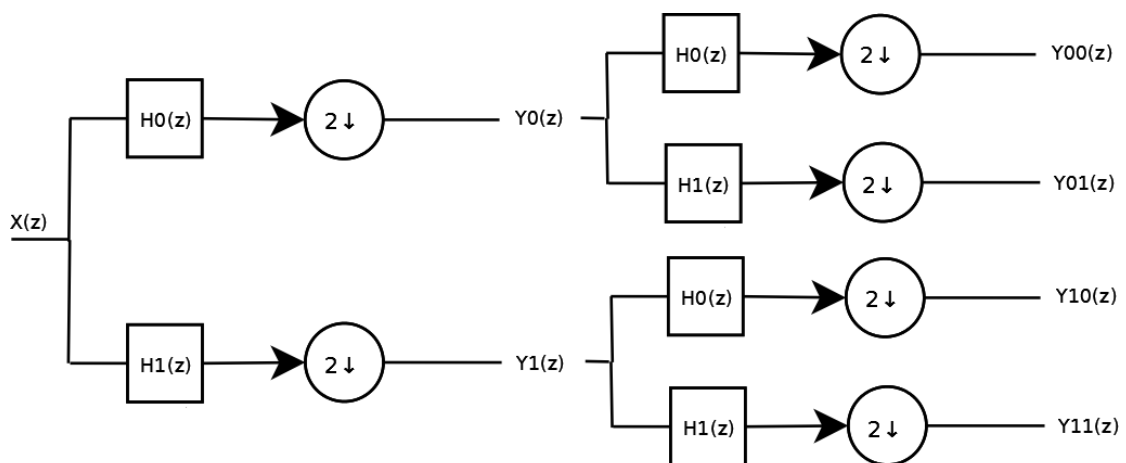


FIGURA 3.5: Iterazione completa (Walsh) del banco di filtri.

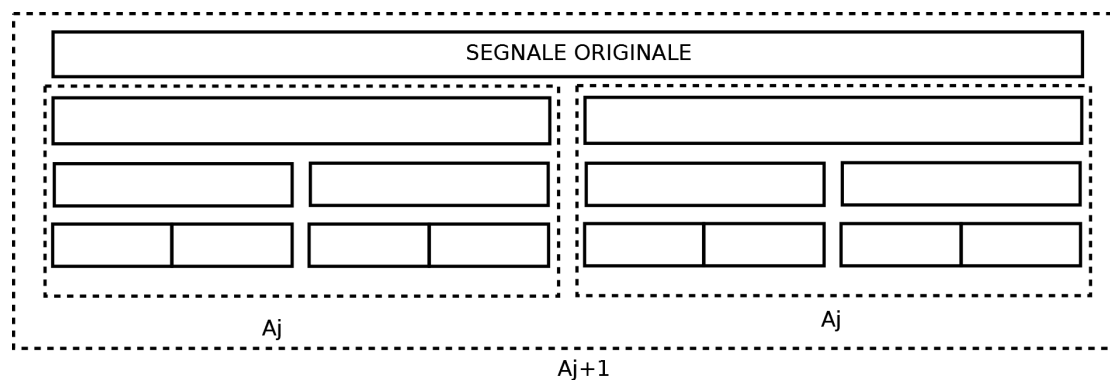


FIGURA 3.6: Numero di rappresentazioni nella scomposizione *Wavelet Packets*.

Capitolo 4

Trasformata di Hilbert

Nel presente capitolo si riporta la teoria della trasformata di Hilbert. La letteratura su questo argomento è varia e utili riferimenti possono essere trovati in [28][29][35][37][7][10][25].

4.1 Definizione

Molte comuni trasformazioni integrali [28] possono essere scritte nella seguente forma:

$$g(x) = \int_a^b k(x, y) dy, \quad (4.1)$$

dove $k(x, y)$ rappresenta la funzione *kernel*. La trasformata di Hilbert [35] è un operatore lineare così definito,

$$H[x(t)] = x_H(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau, \quad (4.2)$$

con $x(t)$ segnale nel tempo. L'equazione (4.2) rappresenta una trasformazione integrale che non chiarisce immediatamente le proprietà dell'operatore. Normalmente non è possibile calcolare l'integrale improprio a causa della singolarità in $t = \tau$, pertanto la (4.2) è da intendersi come valore principale di Cauchy. In realtà questa trasformazione rivela informazioni su particolari attributi istantanei chiamati ampiezza e fase.

Storicamente l'espressione (4.2) non è stata definita da David Hilbert, il quale lavorando nell'area delle equazioni integrali arrivò a formulare una serie di relazioni tra la parte reale ed immaginaria delle funzioni analitiche. La prima definizione in realtà si deve al matematico inglese G. H. Hardy, che conì il nome attualmente in uso. La maggior parte degli sviluppi fu fatta proprio da Hardy, da Titchmarsh e da Young. Sebbene l'espressione (4.2) sia la più usata esistono tuttavia delle definizioni alternative nelle quali viene omissa il termine π oppure il *kernel*

viene riscritto in questo modo, $1/\pi(y-x)$. Infine esiste una definizione in cui l'unità immaginaria viene inclusa rimpiazzando π^{-1} con $(i\pi)^{-1}$. Affinchè l'integrale della (4.3) esista, ci sono diversi gradi di rigore, per quanto riguarda le applicazioni fisiche generalmente l'esistenza dell'integrale di Riemann della funzione $|x|^2$ è condizione sufficiente affinché esista la trasformata di Hilbert.

E' importante comprendere gli effetti della trasformata di Hilbert dal punto di vista di quella di Fourier. Richiamando le proprietà sulla convoluzione la (4.2) può essere riscritta in questo modo,

$$H[x(t)] = \frac{1}{\pi} x(t) \star \frac{1}{t}. \quad (4.3)$$

Si può riconoscere che il termine $1/t$ possiede una trasformata di Fourier nota, infatti,

$$F\left[-\frac{1}{i\pi t}\right] = \text{sgn}(\omega), \quad (4.4)$$

dove $\text{sgn}(\omega)$ rappresenta la funzione segno, così definita,

$$\text{sgn}(\omega) = \begin{cases} +1 & \omega \geq 0 \\ -1 & \omega < 0 \end{cases}. \quad (4.5)$$

Eseguendo quindi la trasformata di Fourier della (4.2) otteniamo,

$$X_H(\omega) = F[x_H(t)] = X(\omega)(-i\text{sgn}(\omega)) = X_{\perp}(\omega). \quad (4.6)$$

Nel dominio delle frequenze la trasformata di Hilbert corrisponde ad un cambio di fase pari a -90° per frequenze positive e 90° per quelle negative. E' chiaro quindi che eseguendo l'antitrasformata della (4.6) si ottiene proprio o la $x_H(t)$. La figura 4.1 mostra in modo figurato l'equivalenza dei due approcci e implicitamente suggerisce un modo pratico per calcolare la trasformata di Hilbert. L'equazione (4.6) suggerisce anche che l'operatore $H[\]$ può essere visto come un filtro in quadratura con funzione di trasferimento:

$$H(\omega) = \begin{cases} i & \omega > 0 \\ 0 & \omega = 0 \\ -i & \omega < 0. \end{cases}. \quad (4.7)$$

La trasformata inversa di Hilbert è definita in modo analogo alla (4.2),

$$H^{-1}[x_H(t)] = x(t) = -\frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x_H(\tau)}{t-\tau} d\tau. \quad (4.8)$$

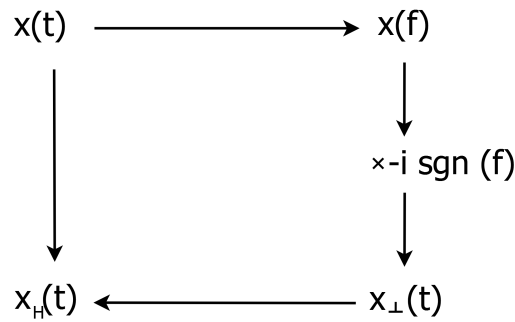


FIGURA 4.1: Modalità di calcolo della trasformata di Hilbert nel dominio delle frequenze.

Ovviamente varrà anche la seguente relazione

$$X(\omega) = X_{\perp}(\omega)(i \operatorname{sgn}(\omega)), \tag{4.9}$$

da cui si comprende che l'operatore inverso di Hilbert corrisponde ad uno sfasamento di 90° in direzione opposta alla trasformazione diretta.

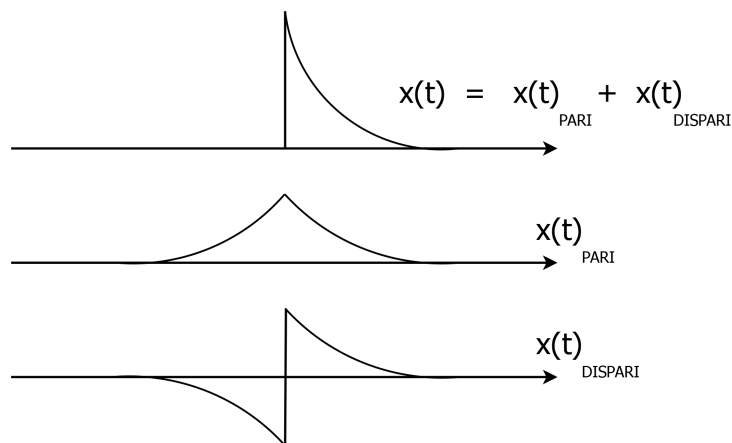


FIGURA 4.2: Scomposizione di un segnale causale nella somma di due segnali non causali uno pari e l'altro dispari.

Un segnale causale può essere scomposto nella somma di due segnali non causali, figura 4.2, tra di loro legati dalla funzione segno,

$$\begin{aligned} x(t) &= x_{\text{pari}}(t) + x_{\text{dispari}}(t) \\ x_{\text{pari}}(t) &= x_{\text{dispari}}(t) \operatorname{sgn}(t). \end{aligned} \tag{4.10}$$

Nel dominio della frequenza avremo,

$$\begin{aligned} F[x(t)] &= X_R(\omega) + iX_I(\omega) = \dots \\ &\quad X_{\text{pari}}(\omega) + iX_{\text{dispari}}(\omega), \\ F[x_R(t)] &= iX_{\text{dispari}}(\omega) \star \frac{1}{i\pi f} = \dots \\ &= H[X_{\text{dispari}}(\omega)] = H[X_I(\omega)], \end{aligned}$$

da cui segue immediatamente la seguente relazione,

$$X_R(\omega) = H[X_I(\omega)]. \quad (4.11)$$

Per i segnali causali quindi la rispettiva trasformata di Fourier della parte reale ed immaginaria è legata dall'operatore di Hilbert [4]. Osservando inoltre le equazioni (4.7) e (4.8) comprendiamo immediatamente che il segnale originale dovrà avere media nulla affinché possa essere completamente recuperato, altrimenti l'informazione in $X(\omega)|_{\omega=0}$ andrà perduta.

4.2 Segnali analitici, attributi e proprietà

Dato un segnale $x(t)$ e la sua trasformata di Hilbert si definisce *segnale analitico* la seguente espressione,

$$x_a(t) = x(t) + ix_H(t). \quad (4.12)$$

Il modulo della (4.12) viene chiamato *ampiezza istantanea* e rappresenta l'involuppo del segnale [35]. La *fase istantanea* è definita invece come,

$$\theta(t) = \text{tg}^{-1} \frac{x_H(t)}{x(t)}, \quad (4.13)$$

mentre la derivata della (4.13) rappresenta la *frequenza istantanea*,

$$f(t) = \frac{1}{2\pi} \frac{d\theta(t)}{dt}, \quad (4.14)$$

Consideriamo un segnale armonico del tipo $x(t) = \cos 2\pi ft$, allora la trasformata di Hilbert sarà, $x_H(t) \sin 2\pi ft$, come si può osservare dalla figura 4.3.

La trasformata di Fourier del segnale analitico sarà,

$$X_a(\omega) = X(\omega) + iX_H\omega,$$

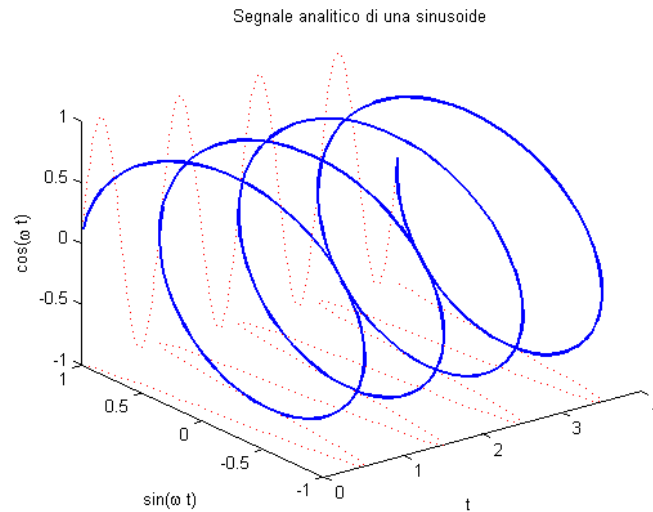


FIGURA 4.3: Esempio di segnale analitico di una senoide.

che in base alla (4.8) diventa,

$$X_a(\omega) = \begin{cases} 0 & \omega < 0 \\ X(\omega) & \omega = 0 \\ 2X(\omega) & \omega > 0. \end{cases} \quad (4.15)$$

È chiaro che la (4.15) rappresenta lo spettro di una funzione *one-side*. Come si può osservare dalla figura 4.4 il segnale analitico contiene solo frequenze positive, in altre parole i vettori ruotano in na sola direzione nel piano complesso. Con la sua introduzione si eliminano le frequenze negative, ma si introducono i segnali complessi nel tempo. Emergono tuttavia due utili proprietà, o attributi, l'involuppo e la fase istantanea.

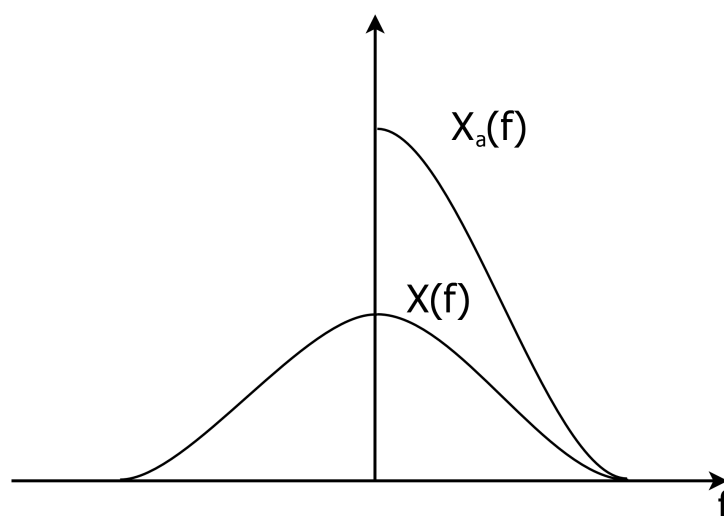


FIGURA 4.4: Spettro del segnale reale $X(f)$ ed analitico $X_a(f)$.

La risposta all'impulso $h(t)$ di un sistema causale deve essere nulla per $t < 0$, pertanto sfruttando l'equazione (4.12) otteniamo,

$$H(\omega) = H_R(\omega) + iH_I(\omega) = H[H_I(\omega)] + iH_I(\omega) = A(\omega)e^{i\phi(t)(\omega)}. \quad (4.16)$$

Una relazione simile alla (4.12) vale per $A(\omega)$ e $\phi(\omega)$ in misura ridotta. Consideriamo infatti $h(t)$ nel dominio s ,

$$H(s) = k \frac{(s - a_1)(s - a_2) \dots (s - a_n)}{(s - b_1)(s - b_2) \dots (s - b_n)}, \quad (4.17)$$

con a_i zeri e b_i poli del sistema. La funzione di trasferimento di un sistema lineare può essere sempre scomposta nel prodotto seguente,

$$H(s) = H(s)_{min} H_{ap}(s). \quad (4.18)$$

$H_{min}(s)$ è chiamata porzione a fase minima, in cui tutti i poli e gli zeri sono posti nella parte sinistra (reale) del piano s , e rappresenta il minimo ritardo tra input ed output. $H_{ap}(s)$ rappresenta invece la parte passa tutto in cui i poli e gli zeri sono coniugati negativi l'uno dell'altro: la forma del singolo fattore sarà $\frac{s-a}{s+a^*}$. L'ampiezza caratteristica di $H_{ap}(s)$ è unitaria e quindi solo la sua fase influenzerà la risposta del sistema. In definitiva quindi

$$H(f) = A_{min}(f)A_{ap}(f)e^{i(\phi_{min}(f)+\phi_{ap}(f))} = A_{min}(f)e^{i(\phi_{min}(f)+\phi_{ap}(f))} \quad (4.19)$$

e per H_{min} vale,

$$\ln A_{min}(f) = H[\phi_{min}(f)]. \quad (4.20)$$

L'equazione (4.20) indica che per un dato spettro dell'ampiezza istantanea, la trasformata di Hilbert può essere usata per trovare la corrispondente componente a fase minima.

Se consideriamo due segnali $x(t)$ e $y(t)$ con spettro non sovrapposto, allora la trasformata di Hilbert del prodotto gode di una notevole proprietà [7] [10]. Infatti supponendo che il segnale $y(t)$ un contenuto armonico superiore a quello di $x(t)$, si avrà

$$H[x(t)y(t)] = x(t)H[y(t)]. \quad (4.21)$$

Questa proprietà è molto utile con segnali modulati. Introduciamo infatti il concetto di *involuppo complesso* $\hat{x}(t)$ come un fasore per il quale il modulo e la fase siano funzioni del tempo,

$$\hat{x}(t) = a(t)e^{i\phi(t)} = a(t) \cos \phi(t) + a(t) \sin \phi(t). \quad (4.22)$$

Ad esso possiamo associare un segnale reale

$$x(t) = \Re\{\hat{x}(t)e^{i\omega_0 t}\} = a(t) \cos(\omega_0 t + \phi(t)). \quad (4.23)$$

Ponendo ora

$$\begin{aligned}x_c(t) &= a(t) \cos \phi(t) \\x_s(t) &= a(t) \sin \phi(t),\end{aligned}$$

si ottiene,

$$x(t) = x_c(t) \cos 2\pi f_0 t - x_s(t) \sin 2\pi f_0 t. \quad (4.24)$$

I termini $x_c(t)$ e $x_s(t)$ sono definiti componenti analogiche di bassa frequenza. Osservando ora l'equazione (4.23) notiamo che se $\phi(t) = 0$ allora otteniamo $x(t) = a(t) \cos \omega_0 t$ che corrisponde ad una modulazione di ampiezza. Se al contrario imponiamo $a(t) = 1$ allora otteniamo $x(t) = \cos(2\pi f_0 t + \phi(t))$, ovvero una modulazione di fase. Posto $\psi(t) = 2\pi f_0 t + \phi(t)$, la frequenza istantanea (4.14) sarà,

$$f_{\text{istantanea}} = \frac{1}{2\pi} \frac{d}{dt} \psi(t) = f_0 + \frac{1}{2\pi} \phi'(t). \quad (4.25)$$

Un generico segnale modulato può quindi essere rappresentato dall'involuppo complesso (4.22). Particolarmente interessante è il caso in cui $x_c(t)$ e $x_s(t)$ risultano a banda limitata ($2W$) attorno ad una frequenza centrale f_0 . Poiché $W < f_0$ possiamo usare il risultato del teorema (4.21) e scrivere il seguente sistema di equazioni,

$$\begin{cases} x(t) = x_c(t) \cos \omega_0 t - x_s(t) \sin \omega_0 t \\ x_H(t) = x_c(t) \sin \omega_0 t + x_s(t) \cos \omega_0 t, \end{cases} \quad (4.26)$$

che può essere agevolmente invertito, ottenendo

$$\begin{cases} x_c(t) = x(t) \cos \omega_0 t + x_H(t) \sin \omega_0 t \\ x_s(t) = x_H(t) \cos \omega_0 t - x(t) \sin \omega_0 t. \end{cases} \quad (4.27)$$

Il sistema (4.27) è particolarmente interessante perché permette di ottenere le componenti a bassa frequenza (demodulazione) a partire dal segnale e dalla sua trasformata di Hilbert. Infatti immediatamente deriviamo l'involuppo complesso,

$$\begin{aligned}a(t) &= \sqrt{x_c^2(t) + x_s^2(t)} \\ \phi(t) &= \arctan \frac{x_s(t)}{x_c(t)}.\end{aligned}$$

Nel caso in cui $W \ll f_0$ il metodo risulta particolarmente efficiente.

4.3 Trasformata di Hilbert-Huang

In alternativa alla trasformata di Fourier, Norbert Huang [29] ha proposto un'interessante metodo per estrarre i modi o armoniche presenti in un segnale in condizioni non lineari e non stazionarie. Il metodo viene chiamato anche *Empirical Mode Decomposition* e si basa sull'idea di ridurre qualsiasi serie di dati in una somma di funzioni modali proprie (*IMF - intrinsic mode function*), ognuna delle quali rappresenta un particolare modo intrinseco di oscillare del sistema. Seguendo questa idea egli propose il seguente algoritmo.

1. Individuare ciascun massimo e costruire un curva di inviluppo che connette i massimi mediante interpolazione di una *spline* cubica.
2. Individuare ciascun minimo e costruire un curva di inviluppo che connette i minimi mediante interpolazione di una *spline* cubica.
3. Costruire la media ($m_{kl}(t)$) dei due inviluppi. Il primo indice si riferisce al particolare IMF, mentre il secondo indica il ciclo o iterazione.
4. Costruire la differenza tra la serie di dati e m_{1l} , $h_{1l}(t) = h_{1(l-1)}(t) - m_{1l}$, con $h_{10} \equiv x(t)$.
5. Considerare $h_{1l}(t)$ come un nuovo input e ripetere i passi precedenti fino a soddisfare un criterio di convergenza. Al termine $h_{1l}(t)$ sarà considerato il primo IMF.
6. Porre $h_{20}(t) = x(t) - h_{1l}(t)$ e ripetere la procedura.
7. Al termine porre $c_i(t) = h_{ik}(t)$. Questa particolare azione viene chiamata *sifsting*.

Identificando i termini $c_i(t)$ come i modi intrinseci (IMF), si può scrivere la seguente espressione,

$$X(t) = \sum_{i=1}^n c_i(t) + r_n(t), \quad (4.28)$$

con $r_n(t)$ residuo. L'equazione (4.28) rappresenta proprio la scomposizione in IMF. Il metodo può essere applicato fino a quando sono soddisfatte le seguenti condizioni:

- il numero di massimi e di passaggi per lo zero differisce per non più di una unità;
- il valore medio calcolato dagli inviluppi deve essere nullo.

La procedura delineata tuttavia pur essendo empirica non introduce un criterio per definire il numero di IMF in cui decomporre il segnale. Tra le varie possibilità può essere definita la seguente quantità,

$$D_w = \sum_{i=0}^T \frac{[h_{i(k-1)}(t) - h_{ik}(t)]^2}{h_{i(k-1)}^2(t)}, \quad (4.29)$$

con T tempo totale dell'intervallo in analisi. Nella somma (4.29) gli zeri devono essere esclusi. Quando D_w soddisfa un determinato limite il processo di *sisfting* può essere interrotto. La connessione con il dominio delle frequenze avviene considerando proprio la trasformata di Hilbert. Trascurando infatti il residuo $r_n(t)$ avremo

$$x_h(t) = H[x(t)] = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau = \frac{1}{\pi} \sum_{i=1}^n P \int_{-\infty}^{\infty} \frac{c_i(\tau)}{t - \tau} d\tau. \quad (4.30)$$

Sfruttando ora la nozione di segnale analitico,

$$y(t) = x(t) + ix_H(t) = a(t)e^{i\theta(t)}, \quad (4.31)$$

si ottiene,

$$y(t) = \sum_{k=0}^N \left(c_k(t) + \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{c_k(\tau)}{t - \tau} d\tau \right) = \sum_{k=1}^N a_k(t) e^{i\theta_k(t)}, \quad (4.32)$$

oppure in termini della frequenza istantanea, posto $\theta(t) = \theta_0 + \int_0^t \omega(\tau) d\tau$,

$$y(t) = \sum_{k=1}^N a_k(t) e^{i \int \omega_k(\tau) d\tau}. \quad (4.33)$$

Per confronto scriviamo la scomposizione del segnale reale mediante la trasformata di Hilbert-Huang e quella di Fourier,

$$x(t) = \Re \left\{ \sum_{k=1}^N a_k(t) e^{i \int \omega_k(\tau) d\tau} \right\} \quad (4.34)$$

$$x(t) = \Re \left\{ \sum_{k=1}^{\infty} a_k e^{i\omega_k t} \right\} \quad (4.35)$$

Risulta evidente dalla (4.34) che la forma della trasformata di Hilbert-Huang si presta naturalmente ad analizzare situazioni non stazionarie. La rappresentazione frequenza-tempo delle ampiezze viene chiamata spettro di Hilbert,

$$H(\omega_k, t) = a_k(t). \quad (4.36)$$

La misura dell'ampiezza totale di ogni singola frequenza si chiama spettro marginale ed è data da,

$$H(\omega_k) = \int_0^T H(\omega_k, t) dt, \quad (4.37)$$

o equivalentemente dallo spettro marginale medio,

$$n(\omega_k) = \frac{1}{T} \int_0^T H(\omega_k, t) dt, \quad (4.38)$$

Analogamente si definisce l'energia istantanea al modo seguente,

$$IE(t) = \frac{1}{T} \int_0^{\omega_N} H^2(\omega, t) d\omega. \quad (4.39)$$

Sfruttando i precedenti risultati si può introdurre il grado di stazionarietà di una serie temporale, utilizzando la seguente espressione,

$$DS(\omega) = \frac{1}{T} \int_0^T \left(1 - \frac{H(\omega, t)}{n(\omega)} \right)^2 dt. \quad (4.40)$$

Per certe frequenze $DS(\omega)$ può essere non stazionario per altre può al contrario rimanere stazionario. Nel caso di un processo puramente stazionario $DS(\omega) \equiv 0$ poiché sotto questa ipotesi $n(\omega) \equiv H(\omega, t)$.

4.4 Modalità di calcolo

Il calcolo della trasformata di Hilbert in termini pratici può essere affrontato sia nel dominio del tempo che in quello della frequenza. Nel primo caso supponendo $x_k = x(k) = x(k\Delta t)$, con $k = 1 \dots N$, un segnale campionato uniformemente nel tempo con intervallo Δt la (4.2) può essere riscritta nel modo seguente,

$$x_H^{(k)} = x_H(k\Delta t) = \frac{1}{\pi} \int_{\Delta t}^{N\Delta t} \frac{x(\tau)}{k\Delta t - \tau} d\tau. \quad (4.41)$$

Assumendo nella (4.41) un intervallo Δt tale che $x(\tau)$ possa essere discretizzato, si può giungere ad una forma agevole per l'integrazione diretta, il cui costo computazionale è tuttavia sempre dell'ordine di N^2 . Un metodo diretto molto conveniente può essere dedotto interpretando la trasformata di Hilbert come un filtro [59],

$$x_H(t) = h(t) \star x(t). \quad (4.42)$$

La risposta all'impulso del filtro di Hilbert risulta essere [4],

$$h(n) = \begin{cases} \frac{2}{n\pi} \sin^2\left(\frac{n\pi}{2}\right) & \text{se } n \neq 0 \\ 0 & \text{se } n = 0. \end{cases} \quad (4.43)$$

L'equazione (4.43) non è assolutamente sommabile pertanto la risposta in frequenza del filtro di Hilbert è ideale. In effetti esso è anche non causale essendo non nulla la risposta all'impulso

per $t < 0$, quindi così com'è non può essere applicato ad un segnale tempo dipendente in *real-time*. Si possono utilizzare due tipologie di filtri, gli IIR o i FIR. I filtri IIR di Hilbert conservano un errore di fase e di ampiezza approssimando la risposta ideale. Essi possono essere progettati a partire da due filtri passa tutto aventi una risposta in fase che differisce di circa $\pi/2$. Prendendo come segnale analitico la parte reale e complessa dell'uscita dei due filtri si nota che lo spettro del segnale è praticamente zero nelle frequenze negative. I filtri FIR di Hilbert hanno esattamente $\pi/2$ di sfasamento con in aggiunta una fase lineare richiesta per rendere causale il filtro. Dall'equazione (4.43) inoltre si deduce che il filtro è antisimmetrico. Questo significa che il filtro può essere solo di Tipo-III o Tipo-IV, i quali hanno ampiezza nulla per $\omega = 0$ e nel secondo caso anche in $\omega = \pi$. L'approssimazione quindi è accettabile in una banda che esclude gli estremi. La possibilità di sfasare esattamente di $\pi/2$ e la maggior stabilità rendono i filtri FIR un'attrattiva migliore rispetto a quelli IIR. La progettazione dei filtri FIR si può basare sulle seguenti strategie,

1. approssimazione nel dominio delle frequenze;
2. approssimazione mediante funzioni finestra;
3. approssimazione massimizzando la piattezza della risposta;
4. approssimazione per ottimizzazione.

Nel metodo (1) la funzione di trasferimento desiderata del filtro viene campionata in modo da avere esattamente il numero di coefficienti voluto, in questo modo si ottiene la risposta all'impulso desiderata. In pratica bisogna risolvere la seguente equazione,

$$A(k)e^{i\theta(k)} = D\left(\frac{k\omega_s}{N}\right), \quad k = 1, \dots, N-1. \quad (4.44)$$

Nel secondo metodo invece si parte dall'osservazione che i filtri ideali (passa passo, passa alto, ecc.) hanno una risposta infinita e quindi non sono realizzabili nello schema FIR. Tuttavia se ad essi viene applicata una opportuna finestra allora si possono ottenere delle utili approssimazioni poiché è possibile troncare la sequenza di coefficienti. Nel terzo metodo invece la piattezza viene raggiunta imponendo l'annullamento delle derivate della funzione di trasferimento del filtro nei punti estremi ($\omega = 0$ e $\omega = \pi$). Nell'ultimo caso possono essere utilizzati diversi criteri per la progettazione, ad esempio si può considerare l'ottimizzazione rispetto alla risposta lineare attesa della fase oppure si possono applicare criteri energetici.

Per quanto tuttavia è stato introdotto nel capitolo precedente, la trasformata di Hilbert può essere calcolata anche nel dominio delle frequenze. Infatti dalla relazione (4.8) si ricava la seguente relazione, che può essere agevolmente implementata per via numerica [37],

$$x_H(k) = F^{-1}[-i \operatorname{sgn}(\omega_k) F(x(k))]. \quad (4.45)$$

Formalmente un segnale analitico è una funzione complessa della variabile continua tempo. Una sequenza di numeri complessi ad intervalli temporali discreti, $z(n)$, non può essere considerata in senso formale una funzione analitica. Tuttavia poiché essa condivide molte proprietà della rispettiva controparte continua si può ugualmente parlare di segnale discreto analitico. In termini pratici la segnale analitico discreto si calcola utilizzando la seguente procedura, assumendo per semplicità un numero N pari di campioni:

1. calcolare la DTFT $X(m)$ dei campioni reali.
2. definire la trasformata del segnale analitico $Z(m)$ in questo modo,

$$Z(m) = \begin{cases} X(0) & \text{se } m = 0 \\ 2X(m) & \text{se } 1 \leq m \leq \frac{N}{2} - 1 \\ X(\frac{N}{2}) & \text{se } m = \frac{N}{2} \\ 0 & \text{se } \frac{N}{2} + 1 \leq m \leq N - 1. \end{cases} \quad (4.46)$$

3. calcolare la DTFT inversa per ottenere il segnale analitico completo,

$$z(n) = \frac{1}{NT} \sum_{m=0}^{N-1} Z(m) e^{\frac{i2\pi mn}{2}}. \quad (4.47)$$

Capitolo 5

Algoritmi in C++

I metodi di analisi descritti nei precedenti capitoli sono stati implementati in un software *real-time* in grado di eseguire molteplici funzioni, come:

- ricezione dei dati dall'accelerometro;
- applicazione degli algoritmi di stima;
- visualizzazione dei risultati dei calcoli;
- salvataggio dei dati su disco;
- Visualizzazione dei dati registrati (*playback*).

L'architettura prevede che una scheda di acquisizione, appositamente costruita, campioni il segnale dell'accelerometro e lo trasmetta via ethernet utilizzando il protocollo UDP. Considerando che il programma compilato viene eseguito su un computer standard con sistema operativo della Microsoft, l'invio dei dati è suddiviso in pacchetti con cadenza di 100 ms. Il software pertanto si occuperà di riassemblare ogni singolo pacchetto o buffer in modo da ottenere un segnale continuo. La frequenza di trasmissione di 10 Hz risulta un buon compromesso tra i tempi di risposta richiesti, la quantità di dati campionati, la complessità degli algoritmi di stima e le prestazioni delle schede hardware. Per facilitare i test in campo è previsto un modulo hardware e software ausiliario. Esso ha il compito di interfacciarsi con l'automazione tipica degli ambienti industriali. Oltre al motore di calcolo il software è dotato di una interfaccia grafica che permette di visualizzare il risultato di tutti i calcoli. Sebbene il kernel sia stato scritto in standard C++, per facilitare lo sviluppo della parte grafica ho utilizzato l'ambiente di sviluppo della Embarcadero. Nei paragrafi successivi descriverò gli oggetti più importanti delle routine di calcolo che traducono in termini computazionali pratici gli algoritmi teorici descritti nei capitoli precedenti. Per ogni classe riporterò i metodi più importanti descrivendo in dettaglio l'algoritmo implementato.

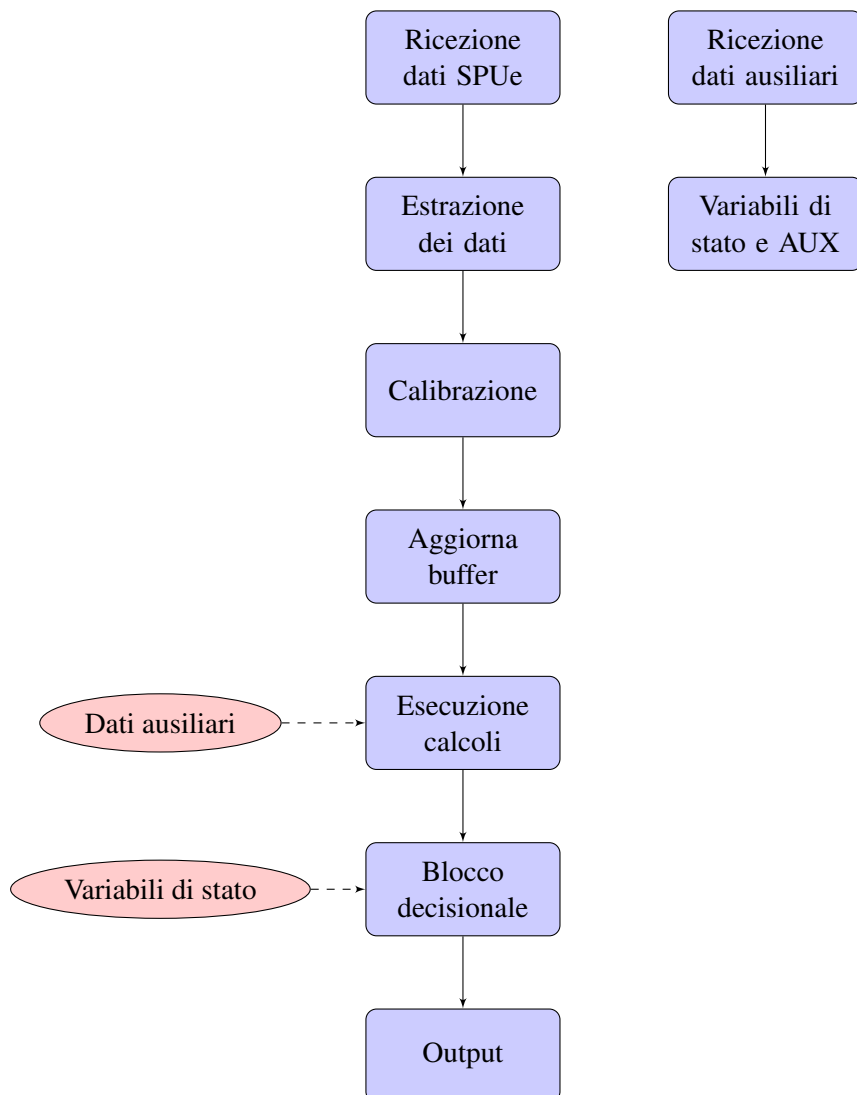


FIGURA 5.1: Diagramma di flusso del ciclo principale.

5.1 Descrizione

L'algoritmo di calcolo è composto da un ciclo principale che viene attivato alla ricezione dei dati dell'accelerometro e da un ciclo minore che viene attivato alla ricezione dei dati ausiliari (figura 5.1).

Dal punto di vista dell'analisi del segnale la sequenza di operazioni è illustrata in figura 5.2, il nuovo buffer di dati viene filtrato e ridotto. Successivamente viene applicato un filtro passa alto per eliminare la componente continua. A questo punto i dati sono pronti per essere inviati alle classi che eseguono i calcoli specifici.

Il software è organizzato in classi C++, che svolgono operazioni specifiche. Per quanto riguarda l'analisi del segnale le classi principali sono:

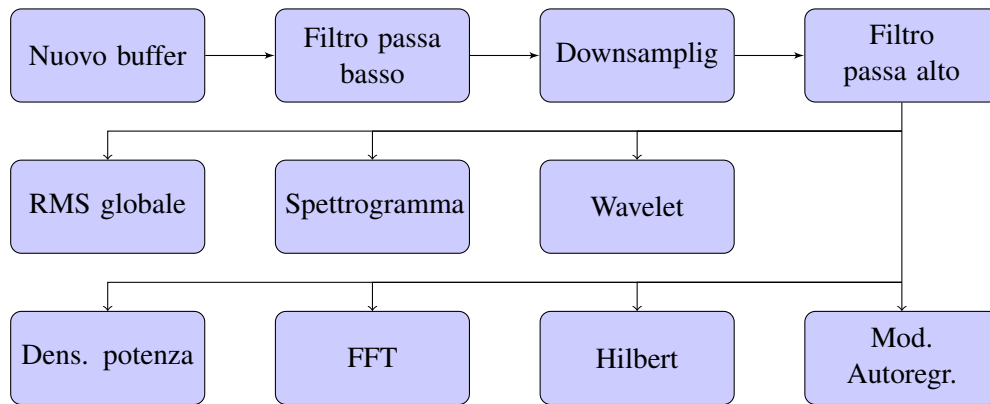


FIGURA 5.2: Sequenza logica delle operazioni di analisi.

- oggetto *TVSD_CALC*, gestione calcoli;
- oggetto *TRMS*, *Root Mean Square*;
- oggetto *TDWT*, analisi *Wavelet* e *Wavelet Packet*;
- oggetto *TClassicPowerSpectrum*, calcolo della potenza con il metodo classico.;
- oggetto *TAR*, modello autoregressivo AR ;
- oggetto *THilbert*, trasformata di Hilbert.

Esiste una serie di classi ausiliarie usate in modo trasversale come,

- *TFFT*. *Fast Fourier Transform*;
- *TFFTW*. *Fast Fourier Transform* utilizzando le librerie reperibili presso il sito <http://www.fftw.org/>;
- *TFiltri*. Implementazione dei filtri IIR e FIR;
- *TBufferCircolari*.

Chiaramente risulta impossibile riportare in dettaglio tutte le classi sviluppate, pertanto verranno considerati solo gli elementi essenziali, in modo da comprendere la filosofia utilizzata e le tecniche computazionali utilizzate.

5.1.1 Classe TDWT

Questa classe si occupa del calcolo sia della trasformata *Wavelet* e sia della *Wavelet Packet*. Essa contiene le classi *TWODEC* e *TWP_RIC* che si occupano rispettivamente della decomposizione e ricostruzione del segnale. Tutti i vettori e le matrici vengono preallocate per velocizzare i tempi

di calcolo. Questa classe viene utilizzata sia per la trasformata Wavelet che Wavelet Packet, ed è implementata basandosi sul concetto di albero il cui singolo nodo è chiamato atomo (figura 5.3). Ogni atomo è caratterizzato dal livello, dalla posizione a quel dato livello e dal tipo di decomposizione (filtro passa alto o filtro passa basso) e contiene al suo interno i coefficienti della decomposizione. Questo concetto è implementato mediante la classe TWATOMO di cui riporto il codice C++ del costruttore. L'etichetta *A* indica i coefficienti approssimati, mentre la *B* indica quelli di dettaglio.

```

1 TDWT::TDWT(stDWT_conf *pst)
  {   int i,j,n,nc;
3  pBuff=pst->pb;
  NpBuffer=pst->np;
5  NLivelli=pst->liv;
  Fs = pst->fs;
7  TipoConvoluzione = pst->tipoConvoluzione;
  TFFTW::stFFTW_conf stFFTW;
9  ///////////////////////////////////////////////////
  switch (pst->tipo)
11 {
  case TDWT::FILTRO_DB10:
13     TipoFiltro=pst->tipo;
     NpFiltroID = 20;
15     NpFiltroR = 20;
     pLo_D = new double[NpFiltroID];
17     pHi_D = new double[NpFiltroID];
     pLo_R = new double[NpFiltroR];
19     pHi_R = new double[NpFiltroR];
     Filtri_db10(pLo_D,pHi_D,pLo_R,pHi_R,NpFiltroID,NpFiltroR);
21     break;
  default:
23     TipoFiltro=pst->tipo;
     NpFiltroID = 2;
25     NpFiltroR = 2;
     pLo_D = new double[NpFiltroID];
27     pHi_D = new double[NpFiltroID];
     pLo_R = new double[NpFiltroR];
29     pHi_R = new double[NpFiltroR];
     Filtri_haar(pLo_D,pHi_D,pLo_R,pHi_R,NpFiltroID,NpFiltroR);
31     break;
  }
33 ///////////////////////////////////////////////////
  // COSTRUZIONE OGGETTO WAVELET DECOMPOSTO
35 pTWDEC = new TWODEC(NpBuffer,NLivelli,NpFiltroID,NpFiltroR,Fs);
  //
37 ///////////////////////////////////////////////////
  // Costruzioni vettori per output esterno tipo dwt matlab
39 //pDec = {App[NLivelli-1] Det[NLivelli-1] Det[NLivelli-2]...Det[0]}
  //          |                               |           |           |
41 //pCoeff = {   NpApp           NpDet           NpDet   ...           NpDet
               NpBuffer}
  //
43 NCoeff = NLivelli+2;

```

```

pCoeff = new int[NCoeff];           // Numero di coefficienti nel vettore di output (
    come matlab)
45 NpOffsetsDec = NCoeff-1;
    pOffsetsDec = new int[NpOffsetsDec];
47 n=NpBuffer;
    pCoeff[NCoeff-1]=NpBuffer;
49 //NpmaxBuffInt=(n*2)*2; //con upsample
    for(i=0;i<NLivelli;i++){
51         pCoeff[NCoeff-1-1-i] = pTWDEC->NumPuntiDec(i);
    }
53 pCoeff[0]=pTWDEC->NumPuntiDec(NLivelli-1);
    pOffsetsDec[0]=0;
55 for(i=1;i<NpOffsetsDec;i++){
        pOffsetsDec[i]=pOffsetsDec[i-1]+pCoeff[i-1];
57 }
    NpDec = pOffsetsDec[NpOffsetsDec-1] + pCoeff[NCoeff-2];
59 pDec = new double[NpDec];
    //////////////////////////////////////
61 //Per dimensionare la massima estensione nelle condizioni al contorno
    NpmaxBuffInt=(NpBuffer*2)*2; //con upsample
63 for(i=0;i<NLivelli;i++){
        //nc = int((n-1)/2+NpFiltriD/2);
65         nc=pTWDEC->NumPuntiDec(i);
        //n=nc;
67         if(NpmaxBuffInt<(nc*4))NpmaxBuffInt=nc*4;
    }
69

71 TipoBordi = pst->tipoBordi;
    NpBufferEsteso = NpBuffer*2; //Dimensionamento massimo
73 NpDummy1 = NpBuffer*2; //Dimensionamento massimo
    NpDummy2 = NpBuffer*2; //Dimensionamento massimo
75 pBuffEsteso = new double[NpBufferEsteso];
    pDummy1 = new double[NpDummy1];
77 pDummy2 = new double[NpDummy2];

79 //////////////////////////////////////

81 NcA = NpBufferEsteso;
    NcD = NcA;
83 pcA = new double[NcA];
    pcD = new double[NcD];
85

    pLp1 = new double[NpmaxBuffInt];
87 pLp2 = new double[NpmaxBuffInt];
    pLp3 = new double[NpmaxBuffInt];
89 pLp4 = new double[NpmaxBuffInt];
    pLp5 = new double[NpmaxBuffInt];
91 pLp6 = new double[NpmaxBuffInt];

93 pHp1 = new double[NpmaxBuffInt];
    pHp2 = new double[NpmaxBuffInt];
95 pHp3 = new double[NpmaxBuffInt];
    pHp4 = new double[NpmaxBuffInt];
97 pHp5 = new double[NpmaxBuffInt];

```

```

pHp6 = new double[NpmaxBuffInt];
99 pXeR = new double[NpmaxBuffInt];
pXlp = new double[NpmaxBuffInt];
101 pXhp = new double[NpmaxBuffInt];
pXR = new double[NpmaxBuffInt]; NpxR=NpBuffer; //Segnale ricostruito
103 pXZeros = new double[NpmaxBuffInt];
for (i = 0; i < NpmaxBuffInt; i++)pXZeros[i]=0.0;
105
////////////////////////////////////////////////////
107 // livello 0 => primo livello di decomposizione
//PREALLOCAZIONE MATRICI DI TOEPLITZ PER DECOMPOSIZIONE
109 // Livello=0 --> primo step di decomposizione --> NumRigheToeplitz0 = NCoeff[(
    NLivelli-1)-0]*2
// Livello=1 --> secondo step di decomposizione --> NumRigheToeplitz1 = NCoeff[(
    NLivelli-1)-1]*2
111 // ...
//PREALLOCAZIONE MATRICI DI TOEPLITZ PER RICOSTRUZIONE
113 // Livello=0 --> ultimo step di decomposizione --> NumRigheToeplitz0 = NCoeff[(
    NLivelli-1)-1]*4
// Livello=1 --> penul. step di decomposizione --> NumRigheToeplitz1 = NCoeff[(
    NLivelli-1)-2]*4
115 // ...
// Livello=n-1 --> primo step di decomposizione --> NumRigheToeplitzn-1 = NCoeff
    [1]*4
117
if(TipoConvoluzione==T_Tipo_CONV_DWT::CONV_TEMPO){
119     NRigheHToepMax=0;
    vNRigheHToepD = new int[NLivelli];
121     vNRigheHToepR = new int[NLivelli];
    for(j=0;j<NLivelli;j++){
123         vNRigheHToepD[j]=pCoeff[(NCoeff-1)-j]*2;
        vNRigheHToepR[j]=pCoeff[1+j]*4;
125         if(NRigheHToepMax<vNRigheHToepD[j])NRigheHToepMax=vNRigheHToepD[j];
        if(NRigheHToepMax<vNRigheHToepR[j])NRigheHToepMax=vNRigheHToepR[j];
127     }
    //NRigheHToepMax= NpmaxBuffInt;
129     pHToepV1 = new double[NRigheHToepMax];
    pHToepV2 = new double[NRigheHToepMax];
131     pHToepV3 = new double[NRigheHToepMax];
    pHToep_Lo_D = new double**[NLivelli];
133     pHToep_Hi_D = new double**[NLivelli];
    HToep_Tot_Bytes=0;
135     for(j=0;j<NLivelli;j++){
        // vNRigheHToepD[j]=pCoeff[(NLivelli-1)-j]*2;
137         //NRigheHToep=pCoeff[(NLivelli-1)-j]*2;
        // vNRigheHToepD[j]=pCoeff[(NCoeff-1)-j]*2;
139         pHToep_Lo_D[j] = new double*[vNRigheHToepD[j]];
        pHToep_Hi_D[j] = new double*[vNRigheHToepD[j]];
141         for(i=0;i< vNRigheHToepD[j];i++){
            pHToep_Lo_D[j][i] = new double[vNRigheHToepD[j]];
143             pHToep_Hi_D[j][i] = new double[vNRigheHToepD[j]];
            HToep_Tot_Bytes+=2*vNRigheHToepD[j];
145         }
        CostruzioneMatriceToeplitz(pLo_D,NpFilttriD,pHToep_Lo_D[j],
vNRigheHToepD[j]);

```

```

147         CostruzioneMatriceToeplitz (pHi_D, NpFiltriD, pHToep_Hi_D[j],
vNRigheHToepD[j]);
    }
149
    pHToep_Lo_R = new double**[NLivelli];
151    pHToep_Hi_R = new double**[NLivelli];
    for (j=0; j<NLivelli; j++){
153        //      vNRigheHToepR[j]=pCoeff[1+j]*4;
        //      NRigheHToep=pCoeff[1+j]*4;
155        pHToep_Lo_R[j] = new double*[vNRigheHToepR[j]];
        pHToep_Hi_R[j] = new double*[vNRigheHToepR[j]];
157        for (i=0; i<vNRigheHToepR[j]; i++){
            pHToep_Lo_R[j][i] = new double[vNRigheHToepR[j]];
159            pHToep_Hi_R[j][i] = new double[vNRigheHToepR[j]];
            HToep_Tot_Bytes+=2*vNRigheHToepR[j];
161        }
        CostruzioneMatriceToeplitz (pLo_R, NpFiltriR, pHToep_Lo_R[j],
vNRigheHToepR[j]);
163        CostruzioneMatriceToeplitz (pHi_R, NpFiltriR, pHToep_Hi_R[j],
vNRigheHToepR[j]);
    }
165    HToep_Tot_Bytes*=sizeof(double);
}
167
////////////////////////////////////////////////////
169 // COSTRUZIONE OGGETTI PER CONVOLUZIONE CON FOURIER
//PREALLOCAZIONE VETTORE FILTRI PER DECOMPOSIZIONE
171 // Livello=0 --> primo step di decomposizione --> nexD0 = NCoeff[(NLivelli-1)-0]*2
// Livello=1 --> secondo step di decomposizione --> nexD1 = NCoeff[(NLivelli-1)-1]*2
173 // ...
//PREALLOCAZIONE VETTORE FILTRI PER RICOSTRUZIONE
175 // Livello=0 --> ultimo step di decomposizione --> nexR0 = NCoeff[(NLivelli-1)-1]*4
// Livello=1 --> penul. step di decomposizione --> nexR1 = NCoeff[(NLivelli-1)-2]*4
177 // ...
// Livello=n-1 --> primo step di decomposizione --> nexRn-1 = NCoeff[1]*4
179 if (TipoConvoluzione==T_Tipo_CONV_DWT::CONV_FREQUENZA) {
    pFourier_Lo_D = new double*[NLivelli];
181    pFourier_Hi_D = new double*[NLivelli];
    pFourier_Lo_R = new double*[NLivelli];
183    pFourier_Hi_R = new double*[NLivelli];
    pTFFTWCNV_Lo_D = new TFFTWCNV*[NLivelli];
185    pTFFTWCNV_Hi_D = new TFFTWCNV*[NLivelli];
    pTFFTWCNV_Lo_R = new TFFTWCNV*[NLivelli];
187    pTFFTWCNV_Hi_R = new TFFTWCNV*[NLivelli];

189    int nexD, nexR;
    stFFTWCNV.freqCampionamento = 1;
191    stFFTWCNV.npFFT = 0;
    stFFTWCNV.tipoFinestra = TFinestre::RECT;
193    stFFTWCNV.pOpt[0] = 0;
    stFFTWCNV.nOpt = 0;
195    stFFTWCNV.numMedie = 1;
    for (int i=0; i<NLivelli; i++){
197        nexD=pCoeff[(NCoeff-1)-i]*2;
        pFourier_Lo_D[i]= new double[NpmaxBuffInt];

```

```

199         for (int j = 0; j < NpFiltriD; j++) pHFourier_Lo_D[i][j]=pLo_D[j];
        for (int j = 0; j < (nexD-NpFiltriD); j++) pHFourier_Lo_D[i][j+
NpFiltriD]=0.0;
201         stFFT.W.pB = pBuffEsteso;
        stFFT.W.nP = nexD;
203         stFFT.W.nP2 = nexD;
        stFFT.W.pB2 = pHFourier_Lo_D[i];
205         pTFFT.WCONV_Lo_D[i] = new TFFT(W(&stFFT.W));

207         pHFourier_Hi_D[i]= new double[NpmaxBuffInt];
        for (int j = 0; j < NpFiltriD; j++) pHFourier_Hi_D[i][j]=pHi_D[j];
209         for (int j = 0; j < (nexD-NpFiltriD); j++) pHFourier_Hi_D[i][j+
NpFiltriD]=0.0;
        stFFT.W.pB = pBuffEsteso;
211         stFFT.W.nP = nexD;
        stFFT.W.nP2 = nexD;
213         stFFT.W.pB2 = pHFourier_Hi_D[i];
        pTFFT.WCONV_Hi_D[i] = new TFFT(W(&stFFT.W));
215

        nexR=pCoeff[1+i]*4;
217         pHFourier_Lo_R[i]= new double[NpmaxBuffInt];
        for (int j = 0; j < NpFiltriR; j++) pHFourier_Lo_R[i][j]=pLo_R[j];
219         for (int j = 0; j < (nexR-NpFiltriR); j++) pHFourier_Lo_R[i][j+
NpFiltriR]=0.0;
        stFFT.W.pB = pLp5;
221         stFFT.W.nP = nexR;
        stFFT.W.nP2 = nexR;
223         stFFT.W.pB2 = pHFourier_Lo_R[i];
        pTFFT.WCONV_Lo_R[i] = new TFFT(W(&stFFT.W));
225

        pHFourier_Hi_R[i]= new double[NpmaxBuffInt];
227         for (int j = 0; j < NpFiltriR; j++) pHFourier_Hi_R[i][j]=pHi_R[j];
        for (int j = 0; j < (nexR-NpFiltriR); j++) pHFourier_Hi_R[i][j+
NpFiltriR]=0.0;
229         stFFT.W.pB = pHp5;
        stFFT.W.nP = nexR;
231         stFFT.W.nP2 = nexR;
        stFFT.W.pB2 = pHFourier_Hi_R[i];
233         pTFFT.WCONV_Hi_R[i] = new TFFT(W(&stFFT.W));

235     } // end for
} //end if
237
////////////////////////////////////
239
pTWP_RIC = new TWP_RIC(NpBuffer,1<<NLivelli);
241

////////////////////////////////////
243 NnodiMax = pTWP_RIC->NumNodi(); //2^Livello
WP_RMS = new TRMS*[NnodiMax];
245 double **matr;
matr = pTWP_RIC->GetPtr();
247
for (int i = 0; i < NnodiMax; i++){
249     WP_RMS[i] = new TRMS(matr[i],pTWP_RIC->NumPunti());

```

```

        WP_RMS[i]->SetFiltroExp(pst->RMS_FiltroExpAttenuazione,0);
251 }

253
WP_RMS_Out = new double[NnodiMax];
255 WP_RMS_Out_Ordered = new double[NnodiMax];
pNodoToFreqIndex = new int[NnodiMax];
257 for (int i = 0; i < pTWP_RIC->NumNodi(); i++){
        pNodoToFreqIndex[i] = pTWDEC->WP_GetNodoToFreqIndex(i);
259 }
//////////
261 //Costruzione della matrice WP di uscita dei coeff decomposti;

263 WP_NumPuntiLastDec = pTWDEC->NumPuntiDec(NLivelli-1);
WP_LastDecEnergiaMedia = new double[NnodiMax];
265 WP_LastDecFreqs = pTWDEC->WP_GetFreqs();

267 //////////
// Tempo di calcolo
269 pTempo = new TTempiPC();
AbilitaCalcoloTempo = false;
271 W_TempoCalcolo_Dec_ms = 0.0;
W_TempoCalcolo_Ric_ms = 0.0;
273 WP_TempoCalcolo_Dec_ms = 0.0;
WP_TempoCalcolo_Ric_ms = 0.0;
275 }
\lstset {language=C++}
277 \begin{lstlisting}
TWATOMO::TWATOMO(int np,int liv,int nodo){
279     int i;
        Npunti = np;
281     Livello = liv;
        Nodo = nodo;
283     if ( (Nodo%2)==0) Tipo = T_Tipo::A;
        else Tipo = T_Tipo::D;
285     pCoeff = new double[Npunti];
        for (i = 0; i < Npunti; i++)pCoeff[i]=0.0;
287 }

```

L'intero albero viene creato e gestito mediante la classe TWODEC il cui costruttore è il seguente.

```

1 TWODEC::TWODEC(int npBufferINI,int nlivelli,int npFiltro_D,int npFiltro_R,double fs){
        NLivelli = nlivelli;
3         npFiltro_D = npFiltro_D;
        npFiltro_R = npFiltro_R;
5         npBufferIniziale = npBufferINI;
        Fs=fs;
7
        int i,j,n,nc;
9         NpCoeffDec_Np = NLivelli;
        pCoeffDec_Np = new int[NpCoeffDec_Np];
11
        n = npBufferIniziale;
13         for(i=0;i<NLivelli;i++){
                n = int((n-1)/2+npFiltro_D/2); // punti extra dovuti al filtro

```

```

15         pCoeffDec_Np[i] = n;
16     }
17
18     // COSTRUZIONE ALBERO COMPLETO DEI NODI
19     mATOMI = new TWATOMO**[NLivelli]; //Righe
20     for (i = 0; i < NLivelli; i++)
21     {
22         n=1<<(i+1); //2^Livello
23         mATOMI[i] = new TWATOMO*[n];
24         for (j = 0; j < n ; j++) mATOMI[i][j] = new TWATOMO(pCoeffDec_Np[i],i,
25             j); //2^i
26     }
27
28     WP_NpFreq = (1<<NLivelli);
29     WP_pFreqCentrale_NaturalOrder = new double[WP_NpFreq];
30     WP_pFreqInf_NaturalOrder      = new double[WP_NpFreq];
31     WP_pFreqSup_NaturalOrder      = new double[WP_NpFreq];
32
33     WP_FreqSpan = Fs/((1<<(NLivelli+1))*1.0);
34     for (int i = 0; i < WP_NpFreq; i++)
35     {
36         WP_pFreqCentrale_NaturalOrder[i] = WP_FreqSpan*(i*1.0+0.5);
37         WP_pFreqInf_NaturalOrder[i]      = WP_FreqSpan*(i*1.0);
38         WP_pFreqSup_NaturalOrder[i]      = WP_FreqSpan*(i*1.0+1.0);
39     }

```

Notiamo dal costruttore TWODEC (contenuto a sua volta nel costruttore TDWDT) come il numero di punti ad ogni livello deve tener conto anche dei punti aggiuntivi dovuti al filtro che sono necessari per il calcolo della convoluzione. Al suo interno un ciclo for si occupa della creazione dell'albero. Successivamente vengono calcolate le frequenze centrali di ogni singola banda. La decomposizione ordina i nodi secondo quello che viene chiamato *ordine del banco di filtri*, che non corrisponde a quello crescente in frequenza. Per comprendere meglio questo fenomeno consideriamo che la trasformata di Fourier di un segnale real $X_T(\omega)$ è una funzione simmetrica e periodica, questo significa che lo spettro nell'intervallo $[\omega_N, \omega_s]$ è l'immagine speculare dell'intervallo $[0, \omega_N]$, dove ω_N è la frequenza di Nyquist e ω_s è quella di campionamento. Per questa ragione bisogna prestare particolare attenzione nell'interpretazione delle frequenze nel caso dell'analisi *Wavelet Packet*. Supponiamo un segnale s_j con intervallo spettrale $[0, \omega_N^{(j)}]$ e decomponiamolo in s_{j-1} e d_{j-1} . Nel caso di applicazione di un filtro ideale ai due segnali decomposti vengono assegnati gli intervalli $[0, \omega_N^{(j)}/2]$ e $[\omega_N^{(j)}/2, \omega_s^{(j)}/2]$. Nel secondo passaggio di decomposizione le due componenti subiscono l'operazione di *downsampling* e l'intervallo spettrale del segnale viene dimezzato. Quindi i filtri passa alto e passa basso presuppongono l'intervallo $[0, \omega_N^{(j)}/2] = [0, \omega_N^{(j-1)}]$; questa ipotesi è soddisfatta solo nel caso della decomposizione *Wavelet* in quanto il processo viene iterativamente applicato solo al segnale s_{j-1} . Nel caso invece di decomposizione *Wavelet Packet* si applica il processo anche alla componente d_{j-1} , ma essa in seguito al *downsampling* avrà nell'intervallo base l'immagine speculare dell'intervallo

n	GC ->	-> IGC
0=00	00	00
1=01	01	01
2=10	11	10
3=11	10	11

TABELLA 5.1: Tabella codici Gray.

originale $[\omega_N^{(j)}/2, \omega_s^{(j)}/2]$. A titolo di esempio supponiamo di avere una banda iniziale di $[0, 64]$ Hz, allora al primo passo otteniamo $[0, 32]$ Hz per i coefficienti approssimati e $[32, 64]$ Hz per i coefficienti di dettaglio. Dopo il *downsampling* il secondo intervallo verrà traslato nella banda base in modo speculare e indicheremo con $[32, 0]$ Hz questa operazione come rappresentato in figura 5.3. Lo schema introdotto introduce un ordinamento dei nodi che non è equivalente a quello in frequenza, infatti avremo:

- nodo (2,1) - banda 0-16 Hz;
- nodo (2,2) - banda 16-32 Hz;
- nodo (2,3) - banda 48-64 Hz;
- nodo (2,4) - banda 32-48 Hz;

Pertanto l'ordine in frequenza sarà 1,2,4,3. Se ora ridefiniamo la numerazione partendo da zero e scriviamo i numeri in formato binario otteniamo 000, 001, 011, 010. Notiamo che questa sequenza si ottiene cambiando esattamente una cifra per ottenere il numero successivo e tale sequenza viene chiamata *codice Gray permutato*. In generale dato un numero in formato binario il codice Gray equivalente è dato dalla seguente formula

$$\text{Gray Code} = GC(n)_i = n_i + n_{i+1} \text{ mod } 2, \quad (5.1)$$

con inversa

$$\text{Gray Code inverso} = IGC(n)_i = \sum_{k \geq i} n_k + n_{i+1} \text{ mod } 2. \quad (5.2)$$

Riprendendo l'esempio precedente, se applichiamo la procedura ai numeri 0,1,2,3 otterremo la codifica riportata nella tabella 5.1.

Applicando la mappa inversa alla numerazione dei nodi otteniamo l'ordine in frequenza, come rappresentato nella tabella 5.2.

La decomposizione avviene applicando in modo iterativo i filtri, che possono essere implementati sia nel dominio del tempo che nel dominio della frequenza. La differenza è critica in quanto

Nodi =	IGC->	Freq.=	Nodi
0=00=	00	00=	0
1=01=	01	01=	1
2=10=	10	11=	3
3=11=	11	10=	2

TABELLA 5.2: Modalità di passaggio dalla sequenza di nodi crescente alla sequenza di nodi crescente secondo la frequenza

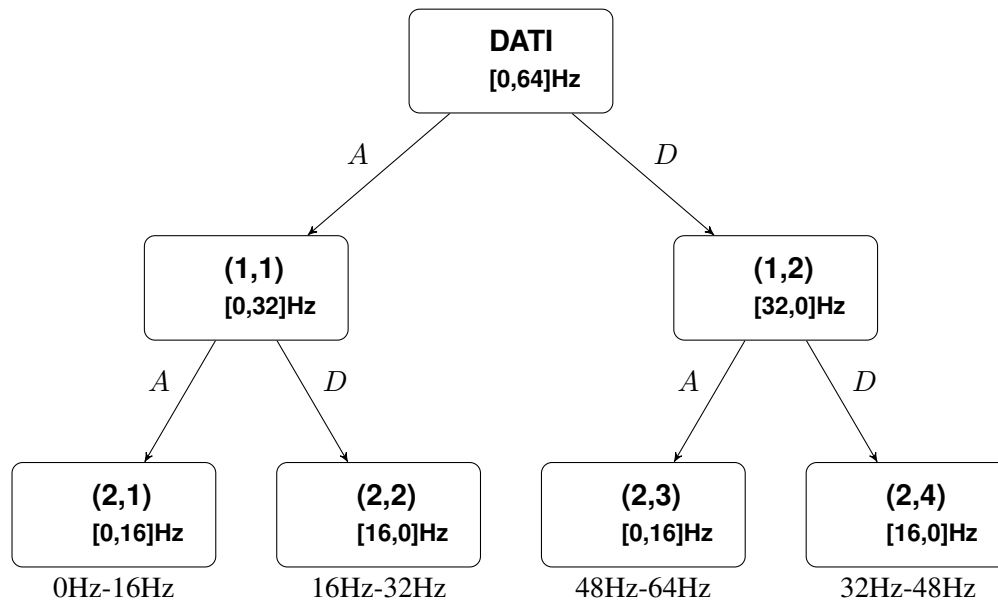


FIGURA 5.3: Struttura ad albero della decomposizione *Wavelet Packet*.

il numero di operazioni nel primo caso cresce come N^2 , mentre nel secondo come $N \log N$. Il seguente codice esegue il passo di decomposizione.

```

1 void TDWT::DecUnLivello(int liv, double *pin, int np, double *pcA, double *pcD, int &nput
  ) {
    int nest, ndec, i, knorm;
3   double *pout1, *pout2;
    SimmetricPaddingHalfPoint(pin, pBuffEsteso, np, nest);
5   switch (TipoConvoluzione)
    {
7   case TDWT::CONV_TEMPO:
        ConvoluzioneCircolareConToeplitz(pBuffEsteso, pLp1, nest, pHToep_Lo_D[
liv]);
9        ConvoluzioneCircolareConToeplitz(pBuffEsteso, pHp1, nest, pHToep_Hi_D[
liv]);
        break;
11  case TDWT::CONV_FREQUENZA:
        pTFFTWCNV_Lo_D[liv]->CalcolaConvConFFT(pout1);
13        pTFFTWCNV_Hi_D[liv]->CalcolaConvConFFT(pout2);
        knorm=nest;
15        for (i = 0; i < nest; i++)
            {pLp1[i]=pout1[i]/knorm;
    
```

```

17         pHp1[i]=pout2[i]/knorm;
18         }
19         break;
20     default:
21         break;
22     }
23     Downsample2(pLp1,nest, pLp2,ndec,1);
24     Downsample2(pHp1,nest, pHp2,ndec,1);
25     npout=pTWDEC->NumPuntiDec(liv);
26     for(i=0;i<npout;i++){
27         pcA[i]=pLp2[i];
28         pcD[i]=pHp2[i];
29     }
30 }

```

Il codice seguente invece implementa il passo di ricostruzione.

```

void TDWT::RicUnLivello(int liv,double *pl,int npl,double *ph,int np, double *pout,
    int &npout){
2     int nest,nup,off,q2,i,livello;
3     double *pout1,*pout2,knorm;
4     livello = NLivelli-1-liv;
5     SimmetricPaddingHalfPoint(pl ,pLp4 ,npl,nest);
6     SimmetricPaddingHalfPoint(ph ,pHp4 ,npl,nest);
7     Upsample2(pLp4,nest, pLp5,nup,0);
8     Upsample2(pHp4,nest, pHp5,nup,0);
9     switch (TipoConvoluzione)
10    {
11    case TDWT::CONV_TEMPO:
12        ConvoluzioneCircolareConToeplitz(pLp5,pLp6,nup,pHToep_Lo_R[livello]);
13        ConvoluzioneCircolareConToeplitz(pHp5, pHp6,nup,pHToep_Hi_R[livello]);
14        break;
15    case TDWT::CONV_FREQUENZA:
16
17        pTFFTWCONV_Lo_R[livello]->CalcolaConvConFFT(pout1);
18        pTFFTWCONV_Hi_R[livello]->CalcolaConvConFFT(pout2);
19        knorm=nup;
20        for (i = 0; i < nup; i++)
21            {pLp6[i]=pout1[i]/knorm;
22            pHp6[i]=pout2[i]/knorm;
23            }
24        break;
25    default:
26        break;
27    }
28    for(i=0;i<nup;i++)pXeR[i]=pLp6[i]+pHp6[i];
29    off=NpFilttriR-1-1;
30    q2=np1-(int)(NpFilttriR/2.0-1.0);
31    q2=q2*2;
32    for(i=0;i<q2;i++)pout[i]=pXeR[i+off];
33    npout=q2;
34 }

```

Le funzioni TDWT::DecUnLivello e TDWT::RicUnLivello, dopo aver applicato le condizioni ai bordi eseguono il filtraggio utilizzando le matrici di Toeplitz oppure la FFT. Un punto cruciale del processo è l'applicazione delle corrette condizioni ai bordi. Infatti nel caso di segnali finiti per garantire la ricostruzione perfetta è necessario avere i dati anche fuori dall'intervallo di analisi. Per risolvere questo problema si possono adottare varie soluzioni, alcune delle quali consistono nell'estendere con zeri, nel rendere l'intervallo simmetrico o periodico. in questo caso discuterò l'estensione simmetrica. Essa può essere realizzata in due modi: data una sequenza $x(0) \dots x(N-1)$, si può imporre $x(-1) = x(1)$ oppure $x(-1) = x(0)$. Nel primo caso si parla di simmetrizzazione completa (**W** extension e il punto di simmetria diventa $x(0)$). Nel secondo caso si ottiene una simmetria a metà intervallo (**H** extension poiché il punto sta idealmente a $x(-1/2)$). La scelta dipende dalla lunghezza del filtro, nel caso di filtri simmetrici pari si usa l'estensione (**H**).

$$\hat{x}(n) = \begin{cases} x(n) & 0 \leq n \leq N - 1 \\ x(2N - n - 1) & N \leq n \leq 2N - 1 \\ 0 & \text{altrove,} \end{cases} \quad (5.3)$$

mentre per filtri *dispari*

$$\hat{x}(n) = \begin{cases} x(n) & 0 \leq n \leq N - 1 \\ x(2N - n - 2) & N \leq n \leq 2N - 3 \\ 0 & \text{altrove.} \end{cases} \quad (5.4)$$

Nel codice è stata implementata la seconda soluzione poiché i filtri utilizzati sono pari.

```
void TDWT::SymmetricPaddingHalfPoint(double *pin, double *pout, int nin, int &nout){
2     int i;
    nout = nin*2;
4     for(i=0;i<nin;i++)pout[i]=pin[i];
    for(i=0;i<nin;i++)pout[i+nin]=pin[nin-1-i];
6 }
```

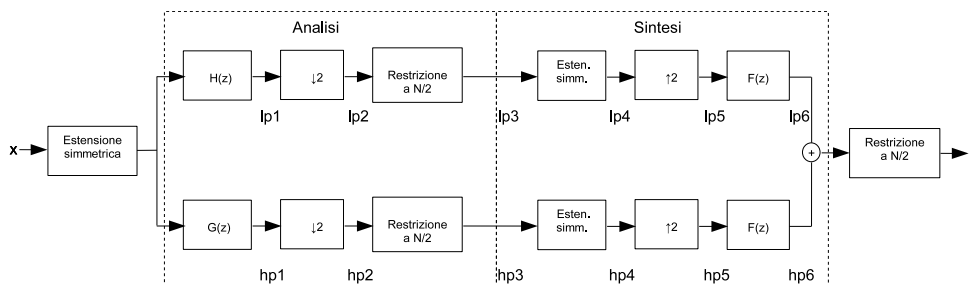


FIGURA 5.4: Schema di simmetrizzazione.

Dalla funzione `TDWT::SymmetricPaddingHalfPoint` si evince che l'operatore raddoppia il numero di elementi del segnale iniziale. La procedura di estensione va applicata sia in fase di decomposizione che in fase di ricostruzione come raffigurato nello schema 5.4. Consideriamo prima il caso di un segnale con un numero di campioni (N) pari ed estendiamolo secondo la procedura introdotta. Nelle figure (5.5a) e (5.5b) è rappresentato il segnale originale e quello esteso considerando di applicare un filtro pari con 20 coefficienti (db10). Applicando la convoluzione ciclica otteniamo i segnali `lp1` e `hp1` (figure 5.5c e 5.5d). Successivamente si esegue il `downsampling` con passo 2 (figure 5.5e e 5.5f). La riduzione del numero di campioni ad N termina la fase di analisi (figure 5.5g e 5.5h). Anche nella fase di sintesi è necessario simmetrizzare il segnale (figure 5.6a e 5.6b) decomposto. Prima del filtraggio (figura 5.6e e 5.6f) si esegue la procedura di `upsampling` (figure 5.6c e 5.6d). Il segnale ricostruito (figura 5.6g) deve essere ridotto al numero di campioni originali (figura 5.6h). È necessario notare alcuni punti importanti analizzando la formula della convoluzione ciclica (5.5).

$$y(k) = \sum_{j=j_0}^{j_1} x(j)h(k-j+1) \quad (5.5)$$

La presenza di $h(k-j+1)$ impone $k-j+1 \geq 0$ da cui si ottengono dei limiti per j_1 per ogni k , operando le sostituzioni osserviamo che nel segnale ricostruito $M-2$ elementi iniziali della convoluzione ciclica devono essere scartati dal processo di restrizione del segnale. Riferendoci al caso specifico in esame il numero di punti dopo la simmetrizzazione è pari a 200 mentre all'uscita dei filtri due rami del filtro di sintesi dopo il `downsampling` i punti sono 100. Nel processo di restrizione è necessario includere anche il contributo dovuto alla lunghezza del filtro, quindi i punti diventano $\text{intero}((N-1)/2+M/2)$ e poiché $N = 100$ e $M = 20$ otteniamo $50+9 = 59$ coefficienti approssimati e 59 coefficienti di dettaglio. Durante la fase di analisi operiamo allo stesso modo, i coefficienti vengo raddoppiati dalla simmetrizzazione e quindi campioni diventano 118 e raddoppiano ulteriormente in seguito al sovra campionamento. L'applicazione dei filtri e la somma degli `output` non altera la lunghezza, tuttavia come si può osservare dalla figura 5.6g si hanno 18 campioni extra all'inizio e 18 campioni extra alla fine.

Approfondiamo ora il metodo con cui applicare i filtri. Un filtro ordinario tempo invariante in forma matriciale assume la configurazione di una matrice di *Toeplitz*, mentre se il filtro è circolare allora la matrice di *Toeplitz* diventa circolante. I blocchi base per la costruzione dei filtri sono le matrici di spostamento (\mathbf{S} e \mathbf{S}^{-1}) che applicate ad una serie temporale di dati causano un ritardo o fanno avanzare di un passo. Per i filtri circolari le matrici saranno spostamenti ciclici. Ad esempio consideriamo la matrice \mathbf{S}_3 applicata al vettore \mathbf{x} ((5.6)), allora potremo dire anche che l'operazione è una permutazione ciclica e notando che ha le stesse righe della matrice identità ma con ordine diverso, possiamo affermare che è una matrice ortogonale. Una generica matrice \mathbf{S}_N ha N autovalori pari alle radici di 1 nel piano complesso. Si indicano con

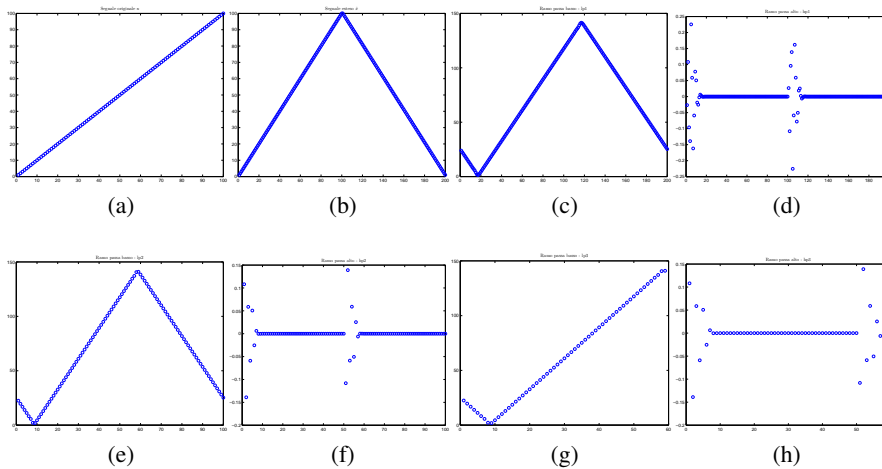


FIGURA 5.5: Fase di analisi: decomposizione DWT.

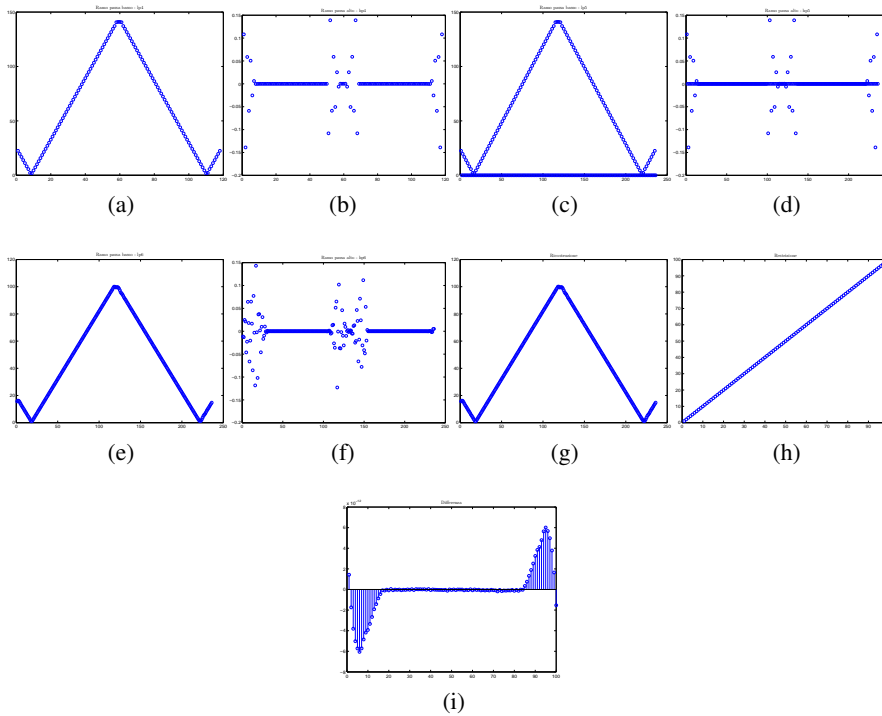


FIGURA 5.6: Fase di sintesi: ricostruzione DWT. La figura 5.6i rappresenta l'errore di ricostruzione.

$W = e^{2\pi i/N}$. Inoltre vale $\bar{W} = W^{-1} = e^{-2\pi i/N}$. Elencando gli autovalori in senso orario ($1, \bar{W}, \bar{W}^2, \dots$) osserviamo che il generico termine $\bar{W}^k = e^{-2\pi i k/N}$ rappresenta la base sui cui viene sviluppata la DFT. Infatti la trasformata di Fourier di \mathbf{x} che indico con \hat{x} è data proprio dalla (5.7).

$$\mathbf{S}_3 \mathbf{x} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \end{pmatrix} = \begin{pmatrix} x(2) \\ x(0) \\ x(1) \end{pmatrix} \quad (5.6)$$

$$\mathbf{x} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \bar{W} & \bar{W}^2 \\ 1 & \bar{W}^2 & \bar{W}^4 \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \end{pmatrix} \quad (5.7)$$

Ad un dato autovettore \bar{W}^k l'autovettore corrispondente è $x = (1, \bar{W}^k, \bar{W}^{2k}, \dots)$ che forma le colonne della matrice DFT nella (5.7). Indicando con \mathbf{F}_N tale matrice possiamo scrivere l'espressione per la DFT di \mathbf{x} (5.8).

$$\hat{\mathbf{x}} = \bar{\mathbf{F}}_N \mathbf{x} \quad (5.8a)$$

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) \bar{W}^{nk} = \sum_{n=0}^{N-1} x(n) e^{-2\pi i nk/N}, \quad (5.8b)$$

dove l'indice k corrisponde alla frequenza e n al tempo. La trasformazione inversa sarà quindi

$$\mathbf{x} = \bar{\mathbf{F}}_N^{-1} \hat{\mathbf{x}} = \frac{1}{N} \mathbf{F}_N \hat{\mathbf{x}} \quad (5.9a)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}(k) W^{nk} = \sum_{k=0}^{N-1} \hat{x}(k) e^{2\pi i nk/N}. \quad (5.9b)$$

Con queste definizioni facciamo la trasformata di Fourier del segnale filtrato $\mathbf{y} = \mathbf{H}\mathbf{x}$.

$$\hat{\mathbf{F}}\mathbf{y} = (\mathbf{F}^{-1}\mathbf{H}\mathbf{F}) \hat{\mathbf{F}}\mathbf{x} = \hat{\mathbf{H}}\hat{\mathbf{x}}. \quad (5.10)$$

L'osservazione cruciale è che l'operazione $(\mathbf{F}^{-1}\mathbf{H}\mathbf{F})$ diagonalizza la matrice \mathbf{H} . Infatti sappiamo che per un filtro circolare abbiamo $H_N = \sum_{n=0}^{N-1} h(n) S_N^n$ ma le potenze di S_N generano la stessa combinazione di potenze di \bar{W}^{kn} , nel senso che se \mathbf{x} è un autovettore allora $\mathbf{S}\mathbf{x} = \bar{W}\mathbf{x}$. Quindi poiché le colonne della matrice \mathbf{F} sono autovettori risulta chiaro che

$$\mathbf{S}\mathbf{F} = \mathbf{F}\mathbf{D}, \quad (5.11)$$

con \mathbf{D} matrice diagonale pari a $diag = (1, \bar{W}, \bar{W}^2, \dots)$ da cui si ricava immediatamente che $\mathbf{S}^n \mathbf{F} = \mathbf{F} \mathbf{D}^n$ e di conseguenza $\mathbf{F}^{-1} \mathbf{S}^n \mathbf{F} = \mathbf{D}^n$. Applicando quanto ricavato alla (5.10), otteniamo

$$\hat{\mathbf{F}} \mathbf{y} = (\mathbf{F}^{-1} \sum_n h(n) \mathbf{S}^n \mathbf{F}) \hat{\mathbf{F}} \mathbf{x} = \left(\sum_n h(n) \mathbf{D}^n \right) \hat{\mathbf{F}} \mathbf{x}. \quad (5.12)$$

Analizziamo ora il termine $\sum_n h(n) \mathbf{D}^n = (h(0) \mathbf{D}^0 + h(1) \mathbf{D}^1 + \dots)$. Essendo infatti tutte matrici diagonali, la loro somma sarà ancora una matrice diagonale:

$$\sum_n h(n) \mathbf{D}^n = \begin{pmatrix} h(0)+h(1)+h(2)+h(3)+\dots & 0 & 0 & \dots \\ 0 & h(0)+h(1)\bar{W}+h(2)\bar{W}^2+h(3)\bar{W}^3+\dots & 0 & \dots \\ 0 & 0 & h(0)+h(1)\bar{W}^2+h(2)\bar{W}^4+h(3)\bar{W}^6\dots & \dots \\ \vdots & \vdots & \vdots & \dots \end{pmatrix} \quad (5.13)$$

Nella (5.13) il generico termine della diagonale è scrivibile come $\hat{h}(k) = \sum_n h(n) \bar{W}^{nk}$, che corrisponde proprio alla trasformata di Fourier del filtro. In conclusione per le matrici circolanti vale la regola secondo cui la convoluzione nel tempo corrisponde alla moltiplicazione nel dominio delle frequenze, infatti:

$$\begin{aligned} \hat{\mathbf{F}} \mathbf{y} &= \begin{pmatrix} \hat{h}(0) & 0 & 0 & \dots \\ 0 & \hat{h}(1) & 0 & \dots \\ 0 & 0 & \hat{h}(2) & \dots \\ \vdots & \vdots & \vdots & \dots \end{pmatrix} \hat{\mathbf{F}} \mathbf{x} & (5.14a) \\ &= \begin{pmatrix} \hat{h}(0) \hat{x}(0) \\ \hat{h}(1) \hat{x}(1) \\ \hat{h}(2) \hat{x}(2) \\ \vdots \end{pmatrix} \\ &= \hat{\mathbf{H}} \hat{\mathbf{x}}, \end{aligned}$$

dove in $\hat{\mathbf{H}} \hat{\mathbf{x}}$ si intende moltiplicazione punto a punto. La routine C++ che implementa quanto descritto si trova nell'oggetto *TFFTW*. Per eseguire la convoluzione nel tempo si costruisce la matrice di *Toeplitz* (5.15) del filtro utilizzando il seguente codice C++.

$$\mathbf{H} = \begin{pmatrix} h(0) & h(N-1) & h(N-2) & \dots \\ h(1) & h(0) & h(N-1) & \dots \\ h(2) & h(1) & h(0) & \dots \\ \vdots & \vdots & \vdots & \dots \end{pmatrix} \quad (5.15)$$

```

void TDWT::CostruzioneMatriceToeplitz(double *pfiltro, int npfiltro, double **matrice,
    int nest) {
2     int i, j;

4     for(i=0; i<npfiltro; i++) pHToepV1[i]=pfiltro[i];
    for(i=0; i<(nest-npfiltro); i++) pHToepV1[i+npfiltro]=0.0;
6     pHToepV2[0]=pHToepV1[0];
    for(i=1; i<nest; i++) {
8         pHToepV2[i]=pHToepV1[(nest-1)-i+1];
    }
10    for(j=0; j<nest; j++) {
        matrice[0][j]=pHToepV2[j];
12    }
    for(j=1; j<nest; j++) {
14        pHToepV2[i]=pHToepV1[(nest-1)-i+1];
        for(i=1; i<nest; i++) pHToepV3[i]=pHToepV2[i-1];
16        pHToepV3[0]=pHToepV2[nest-1];
        for(i=0; i<nest; i++) pHToepV2[i]=pHToepV3[i];
18        for(i=0; i<nest; i++) {
            matrice[j][i]=pHToepV2[i];
20        }
    }
22 }

```

Il numero degli elementi del filtro è minore del numero di elementi del vettore temporale quindi \mathbf{h} viene costruito aggiungendo zeri fino al raggiungimento della stessa lunghezza N del vettore \mathbf{x} : $\mathbf{h} = (h_f(0), h_f(1), h_f(2), \dots, h_f(N_h), 0, \dots, 0)$, dove \mathbf{h}_f sono i coefficienti del filtro, N_h è il numero di coefficienti con $N_h < N$. A titolo di esempio consideriamo il caso del filtro di analisi Daubechies passa basso avente lunghezza 4; applicando l'algoritmo precedente nel caso $N = 6$ otteniamo la seguente matrice,

$$\mathbf{H} = \begin{pmatrix} -0.1294 & 0 & 0 & 0.4830 & 0.8365 & 0.2241 \\ 0.2241 & -0.1294 & 0 & 0 & 0.4830 & 0.8365 \\ 0.8365 & 0.2241 & -0.1294 & 0 & 0 & 0.4830 \\ 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 & 0 \\ 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 & 0 \\ 0 & 0 & 0.4830 & 0.8365 & 0.2241 & -0.1294 \end{pmatrix}$$

con $h_f = (-0.1294, 0.2241, 0.8365, 0.4830)$.

5.1.2 Classe TClassicPowerSpectrum

La classe *TClassicPowerSpectrum* si occupa di calcolare lo spettro di potenza del segnale secondo vari metodi, tuttavia verrà analizzato l'algoritmo di stima di *Welch*. Nel costruttore della classe si può osservare come il numero di punti $NpFFT$ su cui calcolare la FFT venga fissato in

base alla potenza di 2 minore od uguale al numero di punti del vettore dati. La classe utilizzata per la trasformata di Fourier è la *TFFT*, descritta in seguito, che incorpora il codice completo per il calcolo della FFT senza l'ausilio di librerie esterne. Facendo riferimento alla figura 5.7, notiamo che nella variabile *WELCH_NpFFT_seg* viene definito il numero L di campioni del segmento su cui si calcola la FFT, in *WELCH_Sovrapposizione_FFT* si definisce il numero di punti di sovrapposizione L-D e in *WELCH_NSegmenti_FFT* il numero di segmenti in cui è diviso il vettore di dati lungo *NpFFT*. Consideriamo ad esempio un buffer dati di 2002 punti su cui calcolare la FFT a 512 punti e ignoriamo la sovrapposizione allora il numero di segmenti sarà $\text{int}(2002/512) = 3$. Se al contrario consideriamo ad esempio 12 punti di sovrapposizione allora i segmenti diventano 4.

```

TClassicPowerSpectrum::TClassicPowerSpectrum(stClassicPS_conf *pst) {
2     TFFT::stFFT_conf    stFFT;
    int i;
4     double *pb1;
    NPuntiBuffer = pst->npBuffer;
6     pBufferDati = pst->pbuff;
    TipoCalcolo = pst->tipoCalcolo;
8     NOptsCalcolo = pst->noptsCalc;
    TipoFinestra = pst->tipoFinestra;
10    NOptsFinestra = pst->noptsFin;
    FreqCamp = pst->freqCamp;
12
    pOptsCalcolo = new double[NOptsCalcolo];
14    pOptsFinestra = new double[NOptsFinestra];
    for(i=0;i<NOptsCalcolo;i++) pOptsCalcolo[i]=pst->poptsCalc[i];
16    for(i=0;i<NOptsFinestra;i++) pOptsFinestra[i]=pst->poptsFin[i];

18    NpFFT = (int) pow(2.0, int (log(NPuntiBuffer*1.0)/log(2*1.0))*1.0);

20    WELCH_Sovrapposizione_FFT=0;
    switch(pst->tipoCalcolo) {
22    case WELCH :
        WELCH_NpFFT_seg = (int) pOptsCalcolo[0];
24        if ((int)pOptsCalcolo[1]>=0) WELCH_Sovrapposizione_FFT = (int)
pOptsCalcolo[1];
        WELCH_NpFFT_Delta_seg = WELCH_NpFFT_seg-WELCH_Sovrapposizione_FFT;
26        WELCH_NSegmenti_FFT = (int)NPuntiBuffer/WELCH_NpFFT_Delta_seg;
        TipoCalcolo = WELCH;
28
        stFFT.nP = NPuntiBuffer;
30        stFFT.pB = pBufferDati;
        stFFT.freqCampionamento = FreqCamp;
32        stFFT.npFFT = WELCH_NpFFT_seg;
        stFFT.tipoFinestra = TipoFinestra;
34        stFFT.nOpt = NOptsFinestra;
        for(int i=0;i<stFFT.nOpt;i++) stFFT.pOpt[i] = pOptsFinestra[i];
36        stFFT.tipoAllineamento = TFFT_ALLINEAMENTO_LEFT;
        stFFT.numMedie = 1;
38
        pTFFT = new TFFT(&stFFT);
40        WELCH_Somma2Finestra=pTFFT->GetSomma2Finestra();

```

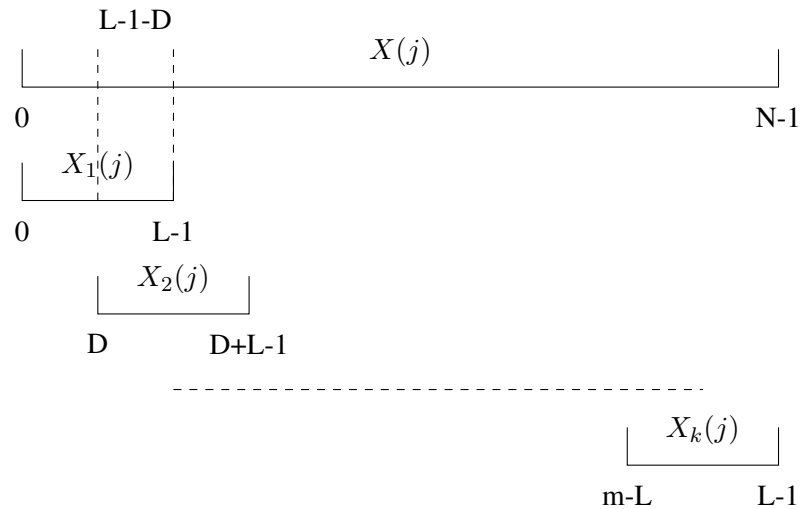


FIGURA 5.7: Segmentazione di Welch. Un vettore di dati lungo N , può essere diviso in sotto-elementi lunghi L e sfasati di D passi.

```

WELCH_NPuntiOut = pTFFT->GetNPuntiModuliFFT();
42  WELCH_pBufferOut      = new double[WELCH_NPuntiOut];
WELCH_pBufferOutdBm = new double[WELCH_NPuntiOut];
44  WELCH_pBufferOut_FREQ = new double[WELCH_NPuntiOut];
pbl = pTFFT->GetFreqs();
46  for (i=0; i<pTFFT->GetNPuntiModuliFFT(); i++) {
        WELCH_pBufferOut_FREQ[i] = pbl[i];
48      WELCH_pBufferOut[i]      = 0.0;
    }
50  WELCH_TempoCalcolo_ms = 0.0;
    break;
52  default: // NON DEFINITA
        TipoCalcolo = NON_DEF;
54  }

56
}

```

Di seguito riporto il metodo esplicito per il calcolo della potenza come descritto nel capitolo 2. Nel calcolo del logaritmo è inserito un valore infinitesimale per evitare instabilità numeriche.

```

1
void TClassicPowerSpectrum::METODO_WELCH(void) {
3  int i, j, np, indice;
    double *pb, k_norma;
5  np= pTFFT->GetNPuntiModuliFFT();
    for (j=0; j<np; j++) WELCH_pBufferOut[j]=0.0;
7
    for (i=0; i<WELCH_NSegmenti_FFT; i++) {
9        indice = IndiceInizialeBuffer+WELCH_NpFFT_Delta_seg*i;
        if (indice<NPuntiBuffer) pTFFT->CalcolaFFT(indice);
11       else pTFFT->CalcolaFFT(indice-NPuntiBuffer);
    }
}

```

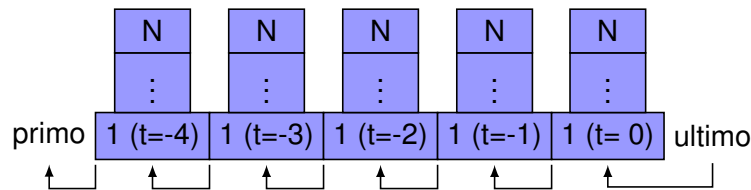


FIGURA 5.8: Schema di buffer circolare bidimensionale. L'inserimento di una nuova colonna di dati al tempo $t = 0$ porta all'espulsione di quella al tempo $t=-4$. Tutte le altre vengono traslate di una unità temporale.

```

13         pb=pTFFT->GetModuli_NO_NORM();
           for (j=0; j<np; j++) {
15             WELCH_pBufferOut [j] +=pb[j]*pb[j];
           }
17     }
           k_norma  = 1.0/(2*PI);
19     k_norma *= PI/(FreqCamp/2.0);
           k_norma *= 1.0/WELCH_Somma2Finestra;
21     k_norma *= 1/2.0;
           for (j=0; j<np; j++) {
23         WELCH_pBufferOut [j]  = WELCH_pBufferOut [j]/WELCH_NSegmenti_FFT*
           k_norma;
           WELCH_pBufferOutdBm[j] = 10*log(WELCH_pBufferOut [j]/DBM_REFERENCE +
           LOG_OFF);
25     }
}

```

Strettamente connessa a questa classe è quella relativa allo spettrogramma. Per ogni buffer acquisito viene implementato il calcolo della potenza il cui risultato è un vettore nello spazio delle frequenze. Ad ogni vettore è associato un preciso intervallo temporale in modo tale che sia possibile costruire una matrice in cui una dimensione rappresenta il tempo e l'altra la frequenza. Definito un intervallo temporale ad ogni nuovo aggiornamento i dati relativi all'intervallo meno recente devono essere cancellati e il nuovo vettore deve essere inserito nella corretta sequenzialità temporale. Questa procedura può essere realizzata mediante il concetto di buffer circolare bidimensionale, la cui rappresentazione grafica si trova in 5.8.

5.1.3 Classe TFFT

Nella classe *TFFT* è implementata la procedura per il calcolo della *Fast Fourier Transform* con l'algoritmo classico derivato dal lavoro di *Cooley-Tukey* [14][4] che costituì una pietra miliare per lo sviluppo di implementazioni sempre più efficienti. L'intera classe di algoritmi FFT si basa sul principio del calcolo della DFT su sequenze via via decrescenti in lunghezza e sicuramente due classi fondamentali si basano sul concetto della decimazione in tempo o in frequenza. Consideriamo un numero di punti $N = 2^p$ e scriviamo la DFT di un segnale periodico $x(m)$ con

$m = 0 \dots N - 1$,

$$F(k) = \sum_{m=0}^{N-1} x(m)W_N^{km}, \quad W_N^{km} = e^{-i\frac{2\pi}{N}km}. \quad (5.16)$$

La equazione (5.16) può essere riscritta al modo seguente,

$$F(k) = \sum_{m=0}^{N-1} A_{km}x(m), \Rightarrow X = Ax, \quad (5.17)$$

con A_{km} matrice quadrata $N \times N$ uguale a $W : N^{km}$. E' chiaro tuttavia che il costo computazionale è pari a N^2 . Per comprendere i passi decisivi della FFT applichiamo il lemma di *Danielson-Lanczos*, da cui la (5.16) può essere riscritta al modo seguente,

$$\begin{aligned} F(k) &= \sum_{m=0}^{\frac{N}{2}-1} x(2m)W_{\frac{N}{2}}^{km} + W_N^k \sum_{m=0}^{\frac{N}{2}-1} x(2m+1)W_{\frac{N}{2}}^{km} \\ &= G(k) + W_N^k H(k). \end{aligned} \quad (5.18)$$

Inoltre poiché $W_{\frac{N}{2}}^{km} = W_N^{(k+\frac{N}{2})m}$ otteniamo l'importante proprietà che $G(k + N/2) = G(k)$ e $H(k + N/2) = H(k)$. Osservando la (5.18) notiamo che abbiamo la somma di due termini uno ottenuto dalla serie iniziale con indici pari e l'altro con indici dispari. Ogni termine è inoltre interpretabile come la DFT di una sequenza avente $N/2$ elementi. Il termine W_N^k viene chiamato *twiddle factor*. Il processo può essere iterato, ad esempio se facciamo un ulteriore passo otteniamo,

$$\begin{aligned} F(k) &= \sum_{m=0}^{\frac{N}{4}-1} x(4m)W_{\frac{N}{4}}^{km} + W_N^k \sum_{m=0}^{\frac{N}{4}-1} x(4m+2)W_{\frac{N}{4}}^{km} + \\ &W_N^k \left(\sum_{m=0}^{\frac{N}{4}-1} x(4m+1)W_{\frac{N}{4}}^{km} + W_N^k \sum_{m=0}^{\frac{N}{4}-1} x(4m+3)W_{\frac{N}{4}}^{km} \right). \end{aligned} \quad (5.19)$$

Consideriamo il caso di 4 elementi, la (5.19) si semplifica in,

$$F(k) = x(0) + W_2^k x(2) + W_4^k (x(1) + W_2^k x(3)), \quad (5.20)$$

che esplicitata per ogni k porta al seguente insieme di equazioni,

$$\begin{aligned} F(0) &= x(0) + W_2^0 x(2) + W_4^0 (x(1) + W_2^0 x(3)), \\ F(1) &= x(0) + W_2^1 x(2) + W_4^1 (x(1) + W_2^1 x(3)), \\ F(2) &= x(0) + W_2^2 x(2) + W_4^2 (x(1) + W_2^2 x(3)), \\ F(3) &= x(0) + W_2^3 x(2) + W_4^3 (x(1) + W_2^3 x(3)). \end{aligned}$$

La ridondanza dei fattori W_N^k permette di scrivere che $W_2^0 = W_2^2 = \dots$ e $W_2^1 = W_2^3 = \dots$, quindi sostituendo abbiamo,

$$\begin{aligned} F(0) &= x(0) + W_2^0 x(2) + W_4^0 (x(1) + W_2^0 x(3)), \\ F(1) &= x(0) + W_2^1 x(2) + W_4^1 (x(1) + W_2^1 x(3)), \\ F(2) &= x(0) + W_2^0 x(2) - W_4^0 (x(1) + W_2^0 x(3)), \\ F(3) &= x(0) + W_2^1 x(2) - W_4^1 (x(1) + W_2^1 x(3)), \end{aligned}$$

e ancora ulteriormente semplificando,

$$\begin{aligned} F(0) &= x(0) + W_2^0 x(2) + W_4^0 (x(1) + W_2^0 x(3)), \\ F(1) &= x(0) - W_2^0 x(2) + W_4^1 (x(1) - W_2^0 x(3)), \\ F(2) &= x(0) + W_2^0 x(2) - W_4^0 (x(1) + W_2^0 x(3)), \\ F(3) &= x(0) - W_2^0 x(2) - W_4^1 (x(1) - W_2^0 x(3)), \end{aligned}$$

Le precedenti equazioni possono essere scritte in forma grafica introducendo il concetto di cella a farfalla (figura 5.9). Infatti ogni ad esempio il termine $F(0)$ può essere visto come la composizione di una cella a farfalla i cui ingressi sono a loro volta l'uscita di due celle simili. In termini analitici essa può essere scritta con il seguente sistema di equazioni,

$$\begin{aligned} X_m(i) &= X_{m-1}(i) + W_N^r X_{m-1}(j), \\ X_m(j) &= X_{m-1}(i) - W_N^r X_{m-1}(j). \end{aligned}$$

Questa forma permette la scrittura di un codice ricorsivo strutturato secondo tre cicli principali. Il più esterno si sviluppa sul numero di passi uguale alla potenza di 2 che da il numero di elementi della serie, nel caso di $N = 4$ avremo 2 passi. Il ciclo intermedio itera sul numero di gruppi di DFT elementari che si hanno ad ogni passo del ciclo principale. Infine il ciclo più interno esegue il numero di celle a farfalla per ogni DFT elementare. Questa tecnica si chiam *in place*. Il codice C++ riportato in seguito implementa la logica di calcolo descritta. Osserviamo inoltre che non è necessario istanziare un altro vettore in memoria ma per la natura ricorsiva dell'algoritmo si può utilizzare lo stesso vettore dati iniziale. In definitiva per ogni cella a farfalla si hanno solo due moltiplicazioni e per ogni ciclo esterno si hanno $N/2$ celle, pertanto considerando che il numero di passi esterno è il logaritmo in base 2 di N , abbiamo che il numero totale di moltiplicazione scende da N^2 a $N \log_2 N$.

```
void TMatematica::ffft1_interna(const int n, double *x_re, double *x_im, const double
    *w_re, const double *w_im)
2 { double w_re, w_im, A_re, A_im, B_re, B_im;
    int s, h, d;
4     int st, k;
```

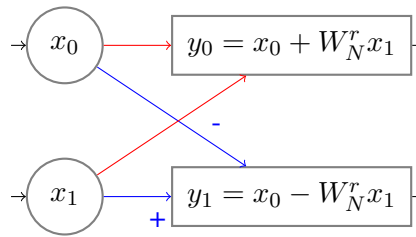


FIGURA 5.9: Cella FFT *butterfly*

```

6  for (s=n; s>=2; s=s>>1)
    {
8      st = s >> 1;
        for (h=0,k=0; h<n; h+=s,k++)
10     {
        w_re = W_re[k];
12     w_im = W_im[k];

14     for (d=h; d<(h+st); d++)
        {
16         B_re = w_re * x_re[d+st] - w_im * x_im[d+st];
        B_im = w_re * x_im[d+st] + w_im * x_re[d+st];
18         A_re = x_re[d];
        A_im = x_im[d];
20         x_re[d] = A_re + B_re;
        x_im[d] = A_im + B_im;
22         x_re[d+st] = A_re - B_re;
            x_im[d+st] = A_im - B_im;
24     }
        }
26 }
};

```

Supponiamo sempre il caso di $N = 4$ e scriviamo esplicitamente la serie di indici che emergono dai cicli *for*.

- $s=4, st=2$

h=0

$[d=0, d+st=2]$ e $[d=1, d+st=3]$

prima farfalla

$$x[0] + W \cdot x[2] \rightarrow x[0];$$

$$x[0] - W \cdot x[2] \rightarrow x[2];$$

seconda farfalla

$$x[1] + W \cdot x[3] \rightarrow x[1];$$

$$x[1] - W \cdot x[3] \rightarrow x[3];$$

$x[0]$	$x[00]$	$x[00]$	$x[0] \rightarrow F[0]$
$x[1]$	$x[01]$	$x[10]$	$x[2] \rightarrow F[1]$
$x[2]$	$x[10]$	$x[01]$	$x[1] \rightarrow F[2]$
$x[3]$	$x[11]$	$x[11]$	$x[3] \rightarrow F[3]$

TABELLA 5.3: Inversione dei bit

- $s=2, st=1$

h=0

[d=0,d+st=1]

prima farfalla

$$x[0] + W \cdot x[1] \rightarrow x[0];$$

$$x[0] - W \cdot x[1] \rightarrow x[1];$$

h=2

[d=2,d+st=3]

seconda farfalla

$$x[2] + W \cdot x[3] \rightarrow x[2];$$

$$x[2] - W \cdot x[3] \rightarrow x[3];$$

h=2

Secondo l'algoritmo precedente quindi otteniamo il seguente risultato,

- $x[0] + W \cdot x[2] + W \cdot (x[1] + W \cdot x[3]) \rightarrow x[0] = F[0];$
- $x[0] + W \cdot x[2] - W \cdot (x[1] + W \cdot x[3]) \rightarrow x[1] = F[2];$
- $x[0] - W \cdot x[2] + W \cdot (x[1] - W \cdot x[3]) \rightarrow x[2] = F[1];$
- $x[0] - W \cdot x[2] - W \cdot (x[1] - W \cdot x[3]) \rightarrow x[3] = F[3].$

Risulta quindi evidente rispetto alla sequenza delle trasformate attese si ha una inversione dei termini $F[2]$ e $F[1]$: questo precisamente è il fenomeno di *bit reversal* che si ha applicando gli algoritmi *in place*. Quindi per ottenere la sequenza corretta è necessario esplicitare gli indici nel sistema binario e invertirli, in questo modo la nuova sequenza di indici ottenuta è quella corretta (5.3). Esistono due modalità per realizzare la tecnica descritta: nella prima la sequenza dati viene elaborata senza inversione allora in questo caso alla serie DFT risultante devono essere invertiti gli indici. Nella seconda si può procedere in modo opposto e cioè applicare l'inversione alla serie temporale, in questo modo la DFT finale risulta ordinata in modo corretto. Nel caso

tuttavia della convoluzione wavelet, non è possibile garantire un buffer con un numero di punti coincidente con una potenza di 2. Ho quindi utilizzato la tecnica del riempimento non zeri per ottenere n vettore di lunghezza opportuna. I risultati sperimentali in termini di velocità di calcolo hanno dimostrato che su un processore intel i7 la routine di calcolo aumenta drammaticamente pur rispettando il tempo di 100 ms. Per questo specifico caso ho quindi utilizzato una libreria pubblica (www.fftw.org) che generalizza il concetto di fattorizzazione intrinseco nel metodo classico della FFT.

5.1.4 Classe TFilteri

La classe dei filtri permette di generare i coefficienti dei filtri discreti Butterworth, Chebyshev ed ellittici. Il metodo utilizzato si basa sulla tecnica standard [33] che prevede l'approssimazione del filtro ideale in termini del filtro analogico, il calcolo dei coefficienti ed infine, mediante la trasformazione bilineare, l'estrazione dei coefficienti nel dominio discreto (5.10). L'espressione riporta la forma generale del filtro passa basso nella sua forma normalizzata, cioè con frequenza di taglio della banda passante pari a 1 rad/sec, per un ordine dato n :

$$H(S) = G \frac{\prod_m (S^2 + A_{1m}S + A_{2m})}{\prod_m (S^2 + B_{1m}S + B_{2m})} \quad (5.21a)$$

$m = 0, \dots, n/2$ n pari

$$H(S) = G \frac{A_1^{(L)}S + A_2^{(L)}}{B_1^{(L)}S + B_2^{(L)}} \frac{\prod_m (S^2 + A_{1m}S + A_{2m})}{\prod_m (S^2 + B_{1m}S + B_{2m})} \quad (5.21b)$$

$m = 0, \dots, n/2$ n dispari

Nelle equazioni (5.21), i polinomi sono espressi in forma quadrata in modo da evidenziare che le radici sono coniugate e rappresentano coppie di poli o zeri, simmetriche rispetto l'asse reale. A questo punto è necessario passare alla forma non normalizzata secondo le trasformazioni (5.22).

$$S \rightarrow \frac{s}{\omega_0} \quad \text{passa basso} \quad (5.22a)$$

$$S \rightarrow \frac{\omega_0}{s} \quad \text{passa alto} \quad (5.22b)$$

$$S \rightarrow \frac{s^2 + \omega_0}{BW \cdot s} \quad \text{passa banda} \quad (5.22c)$$

Il successivo passo consiste nell'applicare la trasformazione bilineare, la quale tuttavia impone delle condizioni sulla frequenza. Per comprendere meglio questa affermazione scriviamo tale

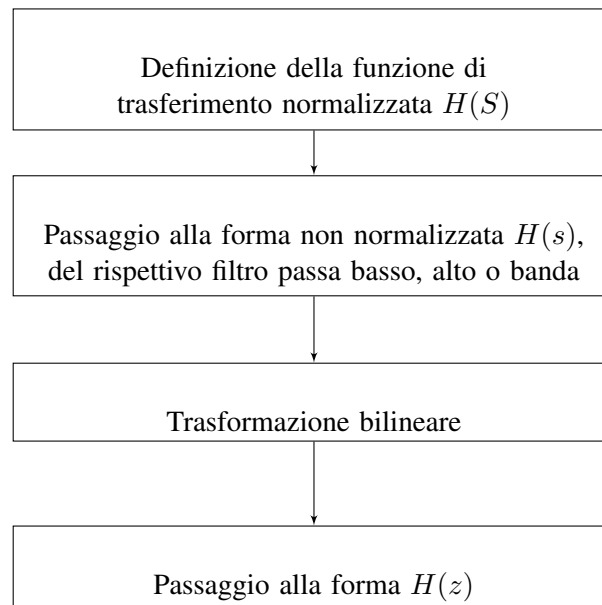


FIGURA 5.10: Procedura di calcolo dei filtri digitali

espressione in forma trigonometrica,

$$\begin{aligned}
 s &= \frac{2}{T} \left(\frac{z-1}{z+1} \right)_{z=Re^{j\Omega}} = \\
 &= \frac{2}{T} \frac{Re^{j\Omega} - 1}{Re^{j\Omega} + 1} \\
 &= \frac{2}{T} \left(\frac{R^2 - 1}{R^2 + 2R \cos \Omega + 1} + j \frac{2R \sin \Omega}{R^2 + 2R \cos \Omega + 1} \right). \quad (5.23)
 \end{aligned}$$

Se nella (5.23) poniamo $R = 1$ allora otteniamo,

$$s = j\omega = \frac{2}{T} j \frac{2 \sin \Omega}{1 + \cos \Omega} = \frac{2}{T} j \tan \frac{\Omega}{2} \quad (5.24)$$

ed invertendo si ha infine,

$$\Omega = 2 \tan \left(\frac{\omega T}{2} \right). \quad (5.25)$$

Questa trasformazione viene definita *warping* nel senso che modifica opportunamente le alte frequenze per mapparle sul circolo unitario. Essa è molto importante perché le specifiche vengono date per il filtro digitale, ma esso viene costruito a partire da quello analogico: è pertanto necessario modificare opportunamente le frequenze nel dominio s affinché combacino con quelle richieste nel dominio z . Il costruttore della classe *TFiltri* permette di scegliere tra i vari tipi di filtri, specificando l'ordine e la banda passante.

```

1  TFiltro::TFiltro(double fpass, double fstop, double banda, double apass, double astop,
   double Tc, T_Tipo_Filtro tipofiltro, int ordine, int npbuffer) {
   TFiltroInterna(fpass, fstop, banda, apass, astop, Tc, tipofiltro, ordine, npbuffer);
3 }

```

```
5 void TFilter::TFilterInterna(double fpass, double fstop, double banda, double apass,
   double astop, double Tc, T_Tipo_Filtro tipofiltro, int ordine, int npbuffer){

7     pMAT = new TMatematica();
   Tipo_Filter=tipofiltro;
9     NPuntiBuffer=npbuffer;
   if (fpass==0) fpass=1.0;
11    if (fstop==0) fstop=1.0;

13

   switch(tipofiltro) {
15

   case BUTTER_LOW:
17         Pord = ordine;
           Pa = new double[Pord+1];
19         Pb = new double[Pord+1];
           Psi = new double[Pord+1];
21         Psu = new double[Pord+1];
           IIRButter_PassaBasso_calcola_coeff(Pord, fpass, apass, Tc, Pa, Pb);
23         break;

25     case BUTTER_HIGH:
           Pord = ordine;
27         Pa = new double[Pord+1];
           Pb = new double[Pord+1];
29         Psi = new double[Pord+1];
           Psu = new double[Pord+1];
31         IIRButter_PassaAlto_calcola_coeff(Pord, fpass, apass, Tc, Pa, Pb);
           break;

33

   case BUTTER_BAND:
35         Pord = ordine*2;
           Pa = new double[(Pord+1)];
37         Pb = new double[(Pord+1)];
           Psi = new double[Pord+1];
39         Psu = new double[Pord+1];
           IIRButter_PassaBanda_calcola_coeff(ordine, fpass, apass, banda, Tc, Pa,
41         Pb);
           break;

43     case CHEBYI_LOW:
           Pord = ordine;
45         Pa = new double[Pord+1];
           Pb = new double[Pord+1];
47         Psi = new double[Pord+1];
           Psu = new double[Pord+1];
49         IIRChebyI_PassaBasso_calcola_coeff(Pord, fpass, apass, Tc, Pa, Pb);
           break;

51

   case CHEBYI_HIGH:
53         Pord = ordine;
           Pa = new double[Pord+1];
55         Pb = new double[Pord+1];
           Psi = new double[Pord+1];
57         Psu = new double[Pord+1];
```

```
        IIRChebyI_PassaAlto_calcola_coeff(Pord, fpass, apass, Tc, Pa, Pb);
59         break;

61     case CHEBYI_BAND:
        Pord = ordine*2;
63         Pa = new double[Pord+1];
        Pb = new double[Pord+1];
65         Psi = new double[Pord+1];
        Psu = new double[Pord+1];
67         IIRChebyI_PassaBanda_calcola_coeff(ordine, fpass, banda, apass, Tc, Pa,
Pb);
        break;

69     case CHEBYII_LOW:
71         Pord = ordine;
        Pa = new double[Pord+1];
73         Pb = new double[Pord+1];
        Psi = new double[Pord+1];
75         Psu = new double[Pord+1];
        IIRChebyII_PassaBasso_calcola_coeff(ordine, fstop, astop, Tc, Pa, Pb);
77         break;

79     case CHEBYII_HIGH:
        Pord = ordine;
81         Pa = new double[Pord+1];
        Pb = new double[Pord+1];
83         Psi = new double[Pord+1];
        Psu = new double[Pord+1];
85         IIRChebyII_PassaAlto_calcola_coeff(Pord, fstop, astop, Tc, Pa, Pb);
        break;

87     case CHEBYII_BAND:
89         Pord = ordine*2;
        Pa = new double[Pord+1];
91         Pb = new double[Pord+1];
        Psi = new double[Pord+1];
93         Psu = new double[Pord+1];
        IIRChebyII_PassaBanda_calcola_coeff(ordine, fstop, banda, astop, Tc, Pa,
Pb);
95         break;

97     case ELLIPTIC_LOW:
        Pord = ordine;
99         Pa = new double[Pord+1];
        Pb = new double[Pord+1];
101        Psi = new double[Pord+1];
        Psu = new double[Pord+1];
103        IIREllittico_PassaBasso_calcola_coeff(ordine, fpass, fstop, apass, astop,
Tc, Pa, Pb);
        break;

105     case ELLIPTIC_HIGH:
107         Pord = ordine;
        Pa = new double[Pord+1];
109         Pb = new double[Pord+1];
```

```

        Psi = new double[Pord+1];
111     Psu = new double[Pord+1];
        IIREllittico_PassaAlto_calcola_coeff(ordine, fpass, fstop, apass,
        astop, Tc, Pa, Pb);
113     break;

115     case ELLIPTIC_BAND:
        Pord = ordine*2;
117     Pa = new double[Pord+1];
        Pb = new double[Pord+1];
119     Psi = new double[Pord+1];
        Psu = new double[Pord+1];
121     IIREllittico_PassaBanda_calcola_coeff(ordine, fpass, fstop, banda,
        apass, astop, Tc, Pa, Pb);
        break;

123
        default: ;
125     }
    }

```

Poiché lo schema è ripetitivo riporto esplicitamente solo l'implementazione del filtro di Butterworth. In questo caso l'equazione nel dominio s è.

$$H(S) = G \frac{\prod_m B_{2m}}{\prod_m (S^2 + B_{1m}S + B_{2m})} \quad (5.26a)$$

$m = 0, \dots, n/2$ n pari

$$H(S) = G \frac{R}{S + R} \frac{\prod_m B_{2m}}{\prod_m (S^2 + B_{1m}S + B_{2m})} \quad (5.26b)$$

$m = 0, \dots, n/2$ n dispari,

che dopo la sostituzione $s = S/\omega_0$, diventa,

$$H(s) = \frac{\prod_m B_{2m}}{\prod_m (s^2 + B_{1m}\omega_0 s + B_{2m}\omega_0^2)} \quad (5.27a)$$

$m = 0, \dots, n/2$ n pari

$$H(s) = \frac{R\omega_0}{S + R\omega_0} \frac{\prod_m B_{2m}}{\prod_m (s^2 + B_{1m}\omega_0 s + B_{2m}\omega_0^2)} \quad (5.27b)$$

$m = 0, \dots, n/2$ n dispari,

ed infine dopo la sostituzione bilineare,

$$H(z) = \frac{\prod_m G_{0m}(1 + 2z^{-1} + z^{-2})}{\prod_m (1 + \beta_{1m}z^{-1} + \beta_{2m}z^{-2})} \quad (5.28a)$$

$m = 0, \dots, n/2$ n pari

$$H(z) = \alpha \frac{1 + \alpha_0 z^{-1}}{1 + \alpha_1 z^{-1}} \frac{\prod_m G_{0m}(1 + 2z^{-1} + z^{-2})}{\prod_m (1 + \beta_{1m}z^{-1} + \beta_{2m}z^{-2})} \quad (5.28b)$$

$m = 0, \dots, n/2$ n dispari,

con,

$$\alpha = \frac{\omega_N R}{R\omega_N + \frac{1}{\pi}},$$

$$\alpha_0 = 1,$$

$$\alpha_1 = \frac{\omega_N R - \frac{1}{\pi}}{\omega_N R + \frac{1}{\pi}},$$

$$\beta_{1m} = 2 \frac{B_{2m}\omega_N^2 - \frac{1}{\pi^2}}{B_{2m}\omega_N^2 + \frac{\omega_N}{\pi}B_{1m} + \frac{1}{\pi^2}},$$

$$\beta_{2m} = \frac{B_{2m}\omega_N^2 - \frac{\omega_N}{\pi}B_{1m} + \frac{1}{\pi^2}}{B_{2m}\omega_N^2 + \frac{\omega_N}{\pi}B_{1m} + \frac{1}{\pi^2}},$$

$$G_{0m} = \frac{\omega_N^2 B_{2m}}{B_{2m}\omega_N^2 + \frac{\omega_N}{\pi}B_{1m} + \frac{1}{\pi^2}},$$

$$B_{2m} = R^2,$$

$$B_{1m} = -2R \cos(\theta_m),$$

$$\theta_m = \pi \frac{2m + n + 1}{2n},$$

$$\omega_N = \frac{\omega_0}{\omega_{\text{sampling}}}.$$

In appendice A riporto il calcolo completo dei coefficienti di tutti i filtri. Il codice seguente calcola i coefficienti secondo le formule precedenti. La funzione `IIRButter_PassaBasso_calcola_coeff`, indipendentemente dall'ordine del filtro, calcola i coefficienti solo del termine lineare e quadratico; successivamente attraverso la funzione `molt_poly` li moltiplica.

```
int TFiltro::IIRButter_PassaBasso_calcola_coeff(int ord, double f0, double Tc, double
    *a, double *b)
2 {
    int i, j, l, m;
4     long double ai[MAX_TERM_POLY], bi[MAX_TERM_POLY], ai0[MAX_TERM_POLY], bi0[
MAX_TERM_POLY];
    long double nz1[MAX_TERM_POLY], dz1[MAX_TERM_POLY], nz2[MAX_TERM_POLY], dz2[
MAX_TERM_POLY];
6     long double w0=2*M_PI*f0; // w0 centrale - media
    aritmetica
    long double wN, wN2, pA[5], pB[5];
8     long double thetam, B1m, B2m, R, G0m, G0, G0mtot, L0;
    long double pi, pi2;
10
12     for(i=0; i<MAX_TERM_POLY; i++) ai[i]=bi[i]=ai0[i]=bi0[i]=nz1[i]=dz1[i]=nz2[i]=
dz2[i]=0.0;
    for(i=0; i<=ord; i++) a[i]=b[i]=0; // azzero polinomi a, b
14
    pi=M_PI;
16     pi2=pi*pi;
18     R=CircleRadius(-3.01, ord);
    w0 = prewarping(w0, Tc);
```

```

20     wN=w0/(2*M_PI)*Tc;
21     wN2=wN*wN;
22
23     G0          =          1.0;
24     G0mtot    =          1.0;
25
26
27     for (int i = 0; i < 2; i++) nz2[i]=dz2[i]=1.0;
28
29     //TERMINE QUADRATICO
30     l=ord/2-1;
31     for (m=0;m<=l;m++) { // calcolo funzioni trasferimento quadratiche
32         thetam = M_PI*(2.0*m+ord+1.0)/(2.0*ord);
33         B1m    = -2.0*R*cos(thetam);
34         B2m    = R*R;
35
36         //Numeratore
37         G0      =          G0*B2m;
38         pA[0] = 0.0; pA[1] = 0.0; pA[2] = 1.0;
39         pB[0] = 1.0; pB[1] = B1m; pB[2] = B2m;
40         //Denominatore
41         TermineQuadraticoGeneralePassaBasso(wN,pA,pB,nz2,dz2,G0m);
42         G0mtot =          G0mtot*G0m;
43         molt_poly(2*m+1,3,ai,bi,ai0,bi0,nz2,dz2);
44     }
45
46     //Guadagno totale termine quadratico
47     for (int i = 0; i < MAX_TERM_POLY; i++){bi[i]=bi[i]*G0*G0mtot;bi0[i]=bi[i];}
48
49     //TERMINE LINEARE
50     if (ord &1) {
51         //Parametri termine Lineare
52         pA[0] = 0.0;          pA[1] = R;
53         pB[0] = 1.0;          pB[1] = R;
54         TermineLineareGeneralePassaBasso(wN,pA,pB,nz1,dz1,L0);
55         molt_poly(ord+1,2,ai,bi,ai0,bi0,nz1,dz1);
56         //Guadagno totale termine lineare
57         for (int i = 0; i < MAX_TERM_POLY; i++){bi[i]=bi[i]*L0;ai0[i]=ai[i];
58         bi0[i]=bi[i];}
59     }
60
61     for(i=0;i<=ord;i++){ a[i]=ai[i]; b[i]=bi[i]; };
62     return 0;
63 }
64 void TFiltro::TermineQuadraticoGeneralePassaBasso(long double wN,long double *pA,long
65 double *pB,long double *pNum, long double *pDen,long double &Gain){
66     long double wN2,bwN2,pi,pi2;
67     long double a0,a1,a2,a3,a4;
68     long double b0,b1,b2,b3,b4;
69     long double A0,A1,A2,A3,A4;
70     long double B0,B1,B2,B3,B4;
71     long double N0,N1,N2,N3,N4;
72     long double D0,D1,D2,D3,D4;

```

```

74      //
75      //          N0  1 + N1/N0*z^-1 + N2/N0*z^-2
76      //  H2(z) = -- * -----
77      //          D0  1 + D1/D0*z^-1 + D2/D0*z^-2
78      //
79
80      A0=pA[0];
81      A1=pA[1];
82      A2=pA[2];
83      B0=pB[0];
84      B1=pB[1];
85      B2=pB[2];
86
87      pi          =      M_PI;
88      pi2         =      pi*pi;
89      wN2         =      wN*wN;
90
91      a0          =  A0/wN2/pi2;
92      a1          =  A1/wN/pi;
93      a2          =  A2;
94
95      b0          =  B0/wN2/pi2;
96      b1          =  B1/wN/pi;
97      b2          =  B2;
98
99      N0          =  a0+a1+a2;
100     N1          =  -2.0*a0+2.0*a2;
101     N2          =  a0-a1+a2;
102
103     D0          =  b0+b1+b2;
104     D1          =  -2.0*b0+2.0*b2;
105     D2          =  b0-b1+b2;
106
107     pNum[0]     =      1.0;
108     pNum[1]     =      N1/N0;
109     pNum[2]     =      N2/N0;
110
111     pDen[0]     =      1.0;
112     pDen[1]     =      D1/D0;
113     pDen[2]     =      D2/D0;
114
115     Gain        =      N0/D0;
116 }
117 void TFiltro::TermineLineareGeneralePassaBasso(long double wN,long double *pA,long
118 double *pB,long double *pNum, long double *pDen,long double &Gain){
119     long double wN2,bwN2,pi,pi2;
120     long double a0,a1,a2,a3,a4;
121     long double b0,b1,b2,b3,b4;
122     long double A0,A1,A2,A3,A4;
123     long double B0,B1,B2,B3,B4;
124     long double N0,N1,N2,N3,N4;
125     long double D0,D1,D2,D3,D4;
126
127     //          A1*S + A2          |

```



```

128 // H(S) = ----- | s
//          B1*S + B2 | S = --
//
//
130 //
//
132 // H1(s) | 2 (z-1)
//          | z = -----
//          |          Tc(z+1)
//
134 //
//
136 // H2(z) = -- * -----
//          D0 1 + D1/D0*z^-1
138 //
//
140 A1=pA[0];
A2=pA[1];
142 B1=pB[0];
B2=pB[1];
144
146 pi      = M_PI;
pi2      = pi*pi;
wN2      = wN*wN;
148
150 a1      = A1/pi/wN;
a2      = A2;
152
154 b1      = B1/pi/wN;
b2      = B2;
156
158 N0      = a1+a2;
N1      = -a1+a2;
160
162 D0      = b1+b2;
D1      = -b1+b2;
164
166 pNum[0] = 1.0;
pNum[1] = N1/N0;
168
164 pDen[0] = 1.0;
pDen[1] = D1/D0;
166
Gain     = N0/D0;
168
}

```

In effetti dopo aver calcolato i coefficienti è necessario affrontare il problema della moltiplicazione dei polinomi. Per comprendere la tecnica consideriamo due polinomi di secondo grado $a[0] + a[1]z^{-1} + a[2]z^{-2}$, $b[0] + b[1]z^{-1} + b[2]z^{-2}$ e procediamo alla moltiplicazione,

$$a[0]b[0] + (a[0]b[1] + a[1]b[0])z^{-1} + (a[0]b[2] + a[1]b[1] + a[2]b[0])z^{-2} + (a[1]b[2] + a[2]b[1])z^{-3} + a[2]b[2]z^{-4}.$$

Possiamo osservare dalla precedente espressione che vengono sommati i coefficienti i cui indici hanno somma pari alla potenza della variabile z^{-1} , in termini formali possiamo scrivere

$$(a[0] + a[1]z^{-1} + a[2]z^{-2})(b[0] + b[1]z^{-1} + b[2]z^{-2}) = \sum_{i+j=0}^4 (a[i]b[j]z^{-(i+j)}). \quad (5.29)$$

Generalizzando la 5.29, otteniamo

$$(a[0] + a[1]z^{-1} + \dots + a[m]z^{-m})(b[0] + b[1]z^{-1} + \dots + b[n]z^{-n}) = \sum_{i+j=0}^{m+n} (a[i]b[j]z^{-(i+j)}),$$

la cui implementazione si trova nella funzione `TFiltro::molt_poly`.

```

1 void TFiltro::molt_poly(int numtermini1, int numtermini2, long double *ai, long double *
    bi, long double *ai0, long double *bi0, long double *nz, long double *dz) {
    for(int i=0; i<MAX_TERM_POLY; i++) ai[i]=bi[i]=0.0;
3     if(numtermini1==1) for (int i = 0; i < MAX_TERM_POLY; i++) {ai0[i]=dz[i];bi0[i]
        ]=nz[i];}
        for (int j = 0; j < numtermini1; j++)for (int i = 0; i < numtermini2; i++) bi[
i+j]=bi[i+j]+bi0[j]*nz[i];
5         for (int j = 0; j < numtermini1; j++)for (int i = 0; i < numtermini2; i++) ai[
i+j]=ai[i+j]+ai0[j]*dz[i];
        for (int i = 0; i < MAX_TERM_POLY; i++) {ai0[i]=ai[i];bi0[i]=bi[i];}
7     }

```

5.1.5 Classe `TFILTROMEDIANO`

Durante le normali operazioni di colaggio l'accelerometro può registrare degli impulsi dovuti alla movimentazione del cassetto di chiusura della siviera o al posizionamento della nuova siviera. Questa perturbazione altera in misura significativa la stima della potenza, pertanto è opportuno limitarne gli effetti. Le tecniche di filtraggio lineare in genere tendono a smorzare lentamente il rumore impulsivo, pertanto ho implementato una tecnica non lineare che si basa sul concetto di mediana [64]. L'idea base di questo metodo consiste nell'applicare una finestra al segnale e rimpiazzare il valore del dato centrale con la mediana. Il processo viene replicato ad ogni avanzamento della finestra. Consideriamo la seguente sequenza di numeri 1 2 9 4 5, allora se li ordiniamo in senso crescente abbiamo 1 2 4 5 9, il cui valore mediano è 4. Tale valore sarà sostituito al posto del numero 9. L'applicazione di queste tecniche richiede una valutazione comunque empirica in quanto esse tendono a rimuovere anche informazione relativa al segnale oltre che al rumore. Di seguito riporto il codice C++ per l'implementazione del filtro.

```
//
```

```

2 TFiltroMediano::TFiltroMediano(double* pb, int nPunti, int nPMediana, bool
    mantieniDatiPrecedenti)

```

```

{
4     int i;
    NpExt=nPunti;
6     NpMediana = nPMediana;
    if ((NpMediana%2) > 0){
8         NpPre = (NpMediana-1)>>1;
        NpPost= NpPre;
10    }
    else {
12        NpPre = NpMediana>>1;
        NpPost= (NpMediana>>1) - 1;
14    }

16    pBufExt  = pb;
    pBufPre  = new double[NpPre];
18    pBufPost = new double[NpPost];
    pBufMediana = new double[NpMediana];
20    MantieniDatiPrecedenti=mantieniDatiPrecedenti;

22    for(i=0;i<NpMediana;i++) pBufMediana[i]=0.0;
    for(i=0;i<NpPre;i++) pBufPre[i]=0.0;
24    for(i=0;i<NpPost;i++) pBufPost[i]=0.0;
}

26 void TFiltroMediano::FiltraMediano(double *pOut)
28 {int i, j, k, ipost;
    for(i=0;i<NpExt;i++){
30        ipost=0;
        for(j=0;j<NpMediana;j++){
32            k=i-NpPre +j;
            if(k<0){
34                pBufMediana[j]=pBufPre[j];
            }
            else if(k>= NpExt){
36                pBufMediana[j]=pBufPost[ipost++];
            }
            else{
38                pBufMediana[j]=pBufExt[k];
            }
40        }
42        Sort(pBufMediana, NpMediana); //Ordinamento in senso crescente
44        pOut[i]=Mediana(pBufMediana, NpMediana);
    }
46 if(MantieniDatiPrecedenti) for(i=0;i<NpPre;i++)pBufPre[i]=pBufExt[NpExt-NpPre+i];
}

48 void TFiltroMediano::Sort(double * pb, int np)
{
50     int b;
    double temp;
52     for (int a=0; a <= np-2; ++a)
    {
54         b = a;
        for (int c=a+1; c <= np-1; ++c)
56         {
            if (pb[c] < pb[b])

```

```

58             b = c;
                }
60         temp = pb[b];
        pb[b] = pb[a];
62         pb[a] = temp;
    }
64 }

66 double TFiltroMediano::Mediana(double *pb, int np)
{   double y;
68 if((np%2)>0){
        y=pb[(np-1)/2]; // se dispari   l'elemento centrale
70 }
    else{
72         y=(pb[np/2-1]+pb[np/2])/2.0; // se pari   la media dei due centrali
    }
74 return y;
}

```

Quando la finestra arriva al termine del buffer, per procedere occorrerebbero dei dati non ancora acquisiti, in questo caso si crea un'estensione del segnale riempiendola con valori nulli. Per comprendere meglio l'effetto del filtro consideriamo un segnale sintetico con frequenza 550Hz al quale è stato aggiunto un impulso della durata di 2 campioni. Proviamo ad applicare il filtro con una finestra di 5 punti. Come possiamo osservare dalla figura 5.11 l'effetto è notevole, tuttavia se confrontiamo i valori efficaci (figura 5.12) notiamo che anche una porzione di segnale utile è stata soppressa. Nella figura 5.13 invece è stato aumentato il numero di punti della finestra e come si può notare l'effetto sulla sinusoidale è ancora più vistoso. Per questo motivo l'applicazione del filtro deve essere eseguita con particolare attenzione. Nella figura 5.11 è riportato anche l'output della funzione *medfilt1* di matlab, come si può osservare il risultato è identico.

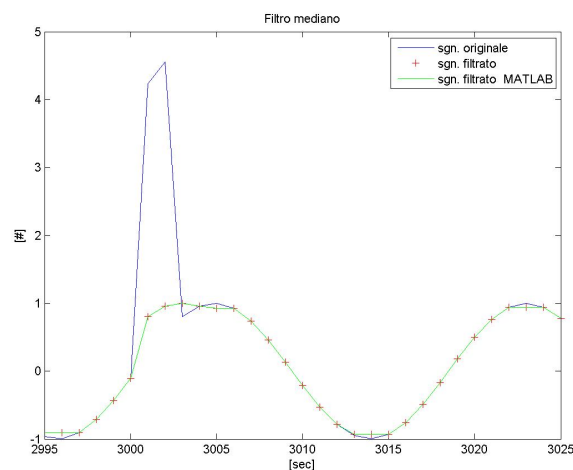


FIGURA 5.11: Filtro mediano applicato ad un segnale sinusoidale con sovrapposto un impulso.

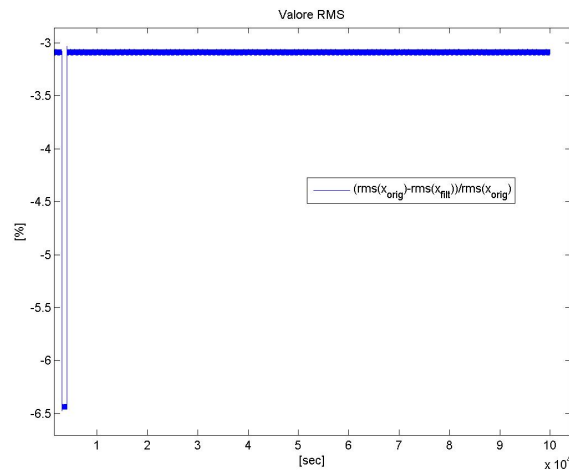


FIGURA 5.12: Effetto del filtro mediano sul valore efficace del segnale con finestra di 3 punti.

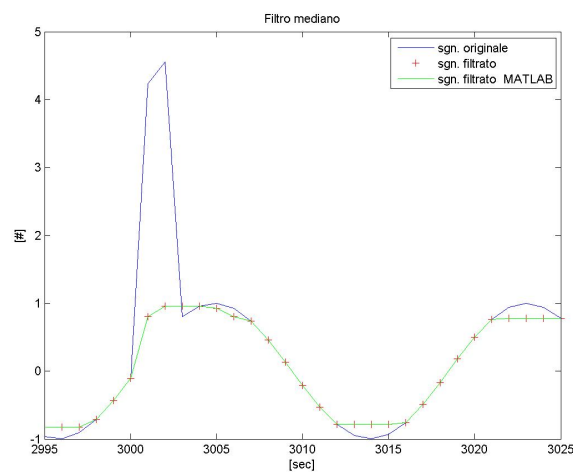


FIGURA 5.13: Effetto del filtro mediano sul valore efficace del segnale con finestra di 5 punti.

5.1.6 Classe TBufferCircolare

Le acquisizioni in tempo reale producono un continuo flusso di dati diviso in pacchetti chiamati buffer, che seguono i tempi di ciclo relativi al sistema di acquisizione. Tuttavia le operazioni matematiche non possono essere vincolate alla lunghezza di questi pacchetti ma devono potersi svolgere su un set di dati avente lunghezza appropriata. Per risolvere questo problema uno dei modi più efficienti è introdurre il concetto di buffer circolare. La parola stessa indica che si tratta di una zona di memoria continuamente riempita. L'idea che ho implementato si basa sulla creazione di una porzione di memoria avente dimensioni multiple rispetto ad un buffer elementare di riferimento, come si può osservare dalla definizione del costruttore.

```

1 TBufferCircolare::TBufferCircolare(int npBufferElementare,int nBufferElementari)
  {   int i;
3       Pn   = npBufferElementare; //Numero di elementi del buffer elementare

```

```

    Pnp  = nBufferElementari; //Numero di buffer elementari che costituiranno
il buffer circolare
5    Pp   = 0;                //Indice dell'ultimo elemento inserito nel buffer
    circolare
    Ppmax = Pn*Pnp;
7    Pbuf  = new double[Ppmax];
    PbufNormalizzato = new double[Ppmax];
9    PCumSum = new double[Ppmax];
    for(i=0;i<Ppmax;i++){Pbuf[i]=0.0;PbufNormalizzato[i]=0.0;}
11 }

```

Per poter funzionare correttamente è necessario definire un puntatore all'*ultimo elemento inserito* e aggiornarlo ad ogni nuovo inserimento mediante il metodo `AggiungiBuffer`.

```

1 void TBufferCircolare::AggiungiBuffer(double *a) {
    int i;
3    for (i=0; i < Pn; i++) Pbuf[Pp+i] = a[i];
    Pp=Pp+Pn;
5    if (Pp>=Ppmax) Pp=0;
}

```

Nel caso ad esempio di spettrogrammi la classe `TBufferCircolare` non è sufficiente in quanto in questo caso il buffer risulta bidimensionale. I concetti precedenti possono tuttavia essere agevolmente estesi; infatti nel costruttore della classe `TBufferCircolareXY` osserviamo che il puntatore al buffer diventa un puntatore ad una matrice di dati.

```

TBufferCircolareXY::TBufferCircolareXY(int npX,int npY)
2 {   int i,j;
    NX   = npX;           //Numero di punti nel tempo per un dato parametro
4    NY   = npY;           //Numero di parametri
    Pp   = 0;             //Indice dell'ultimo elemento inserito nel buffer circolare
6    Pbuf = new double*[NY];
    PbufNormalizzato = new double*[NY];
8    for(i=0;i<NY;i++){
        Pbuf[i] = new double[NX];
10       PbufNormalizzato[i] = new double[NX];
    }
12 for(i=0;i<NY;i++) for(j=0;j<NX;j++){Pbuf[i][j]=0.0;PbufNormalizzato[i][j]=0.0;}
}

```

Questa tecnica si è dimostrata estremamente efficiente e versatile. Risulta anche chiaro che a questa classe si possono agevolmente aggiungere dei metodi che possono eseguire operazioni matematiche sul set di dati.

5.1.7 Classe THilbert

L'algoritmo per la trasformata di Hilbert è stato scritto secondo quanto esposto nel capitolo 4. In particolare il metodo più efficiente consiste nell'utilizzare la trasformata di Fourier. Il costruttore della classe `THilbertTransform()`, crea appunto i vettori necessari per eseguire l'algoritmo

appoggiandosi alla libreria FFTW.

```

1  THilbertTransform::THilbertTransform(double *pb, int npb, int nptrasformata,
    T_Tipo_Hilbert_Calcolo tipo){

3     pBufferIN = pb;
    ModoCalcolo = tipo;
5     NpBufferIN = npb;
    NpTrasformata = nptrasformata;
7     if (nptrasformata > npb) NpTrasformata = npb;
    NpX = NpTrasformata;
9     NpHilbert = NpTrasformata;
    pX_Real = new double[NpX];
11    pX_Imag = new double[NpX];
    pHilbert_Real = new double[NpX];
13    pHilbert_Imag = new double[NpX];

15    pBufferIN_TEST = new double[npb];
    Phit = new double[npb];
17    At = new double[npb];
    tempo = new double[npb];    for (int i = 0; i < npb; i++)tempo[i
        ]=i;
19

21    switch (ModoCalcolo) {
    case T_Tipo_Hilbert_Calcolo::FREQ_FFTW:
23        pFFTWin = (fftw_complex*)fftw_malloc ( sizeof(fftw_complex)
        * NpTrasformata);
        pFFTWout = (fftw_complex*)fftw_malloc ( sizeof(fftw_complex) *
        NpTrasformata);
25        pFFTWHilbert = (fftw_complex*)fftw_malloc ( sizeof(fftw_complex) *
        NpTrasformata);
        FFTWplan_forward = fftw_plan_dft_1d(NpTrasformata, pFFTWin, pFFTWout,
        FFTW_FORWARD, FFTW_MEASURE );
27        FFTWplan_backward = fftw_plan_dft_1d(NpTrasformata, pFFTWout,
        pFFTWHilbert, FFTW_BACKWARD, FFTW_MEASURE );
        break;
29
    default:
31        ;
        }
33 }

```

Il metodo CalcolaHilbert() implementa la seguente procedura,

1. calcolare la DTFT $X(m)$ dei campioni reali.
2. definire la trasformata del segnale analitico $Z(m)$ in questo modo,

$$Z(m) = \begin{cases} X(0) & \text{se } m = 0 \\ 2X(m) & \text{se } 1 \leq m \leq \frac{N}{2} - 1 \\ X(\frac{N}{2}) & \text{se } m = \frac{N}{2} \\ 0 & \text{se } \frac{N}{2} + 1 \leq m \leq N - 1. \end{cases} \quad (5.30)$$

3. calcolare la DTFT inversa per ottenere il segnale analitico completo,

$$z(n) = \frac{1}{NT} \sum_{m=0}^{N-1} Z(m) e^{\frac{i2\pi mn}{2}}. \quad (5.31)$$

Nel codice si trovano commentate le istruzioni che eseguono i passi dell'algoritmo. Il calcolo viene eseguito su ogni singolo buffer acquisito. Al termine si estraggono i valori istantanei di ampiezza e fase rispettivamente con le istruzioni $At[i] = \text{sqrt}(pHilbert_Real[i] * pHilbert_Real[i] + pHilbert_Imag[i] * pHilbert_Imag[i])$ e $Phit[i] = \text{this} \rightarrow \text{atan2}(pHilbert_Imag[i], pHilbert_Real[i])$. In realtà il valore di maggiore interesse è la frequenza istantanea, tuttavia poiché essa è un valore altamente oscillante viene calcolato il valore medio sul singolo buffer con l'istruzione $\text{this} \rightarrow \text{FitLineareCoef}(\text{tempo}, Phit, NpTrasformata, OmegaM, q)$. Si è osservato che per questo tipo di processo il valore di $d\Phi/dt$ è effettivamente costante su un singolo buffer, quindi la procedura è stabile e fornisce un valore attendibile. Per completezza si calcola anche la media dell'ampiezza istantanea per poter fare un confronto con il valore *rms*.

```

1 //-----
void THilbertTransform::CalcolaHilbert(double *pb_real, double *pb_imag, int &np) {
3     Calcola(pBufferIN, pb_real, pb_imag, np);
4 }
5
void THilbertTransform::Calcola(double* pin, double *pb_real, double *pb_imag, int &np)
6 {
7     int i, np_2a, np_2b;
8
9     // PASSO 1 - CALCOLO FFT CAMPIONI REALI
10    for (int i = 0; i < NpTrasformata; i++){
11        pFFTWIn[i][0] = pin[i];
12        pFFTWIn[i][1] = 0.0;
13        pX_Real[i] = pFFTWout[i][0];
14        pX_Imag[i] = pFFTWout[i][1];
15    }
16    fftw_execute ( FFTWplan_forward );
17
18    // PASSO 2 - TRASFORMATA DEL SEGNALE ANALITICO
19    if (NpTrasformata &1 ) {
20        np_2a = (NpTrasformata>>1) + 1;
21        np_2b = np_2a;
22    } else {
23        np_2a = (NpTrasformata>>1);
24        np_2b = np_2a+1;
25    }

```



```

27     for (int i = 1; i < np_2a; i++)
        { pFFTWout[i][0] = 2.0*pFFTWout[i][0];
29           pFFTWout[i][1] = 2.0*pFFTWout[i][1];
        }
31     for (int i = np_2b; i < NpTrasformata; i++)
        { pFFTWout[i][0] = 0.0;
33           pFFTWout[i][1] = 0.0;
        }
35 // PASSO 3 - CALCOLO DELLA FFT INVERSA
    fftw_execute(FFTWplan_backward);
37     for (int i = 0; i < NpTrasformata; i++){
        pHilbert_Real[i] = pFFTWHilbert[i][0]/(NpTrasformata*1.0);
39         pHilbert_Imag[i] = pFFTWHilbert[i][1]/(NpTrasformata*1.0);
        }
41 // FINE ALGORITMO
    //
43     pb_real = pHilbert_Real;
    pb_imag = pHilbert_Imag;
45     np      = NpTrasformata;

47 // CALCOLO ATTRIBUTI Istantanei
    double x,y;
49     for (int i = 0; i < NpTrasformata; i++) {
        y          = pHilbert_Imag[i];
51         x          = pHilbert_Real[i];
        Phit[i] = this->atan2(pHilbert_Imag[i],pHilbert_Real[i]);
53         At[i]    = sqrt(pHilbert_Real[i]*pHilbert_Real[i] + pHilbert_Imag[i]*
            pHilbert_Imag[i]);
        }
55     this->unwrap(Phit,NpTrasformata,Phit);

57 // MEDIA SU SINGOLO BUFFER DEI VALORI Istantanei
    double q;
59     this->FitLineareCoef(tempo, Phit, NpTrasformata, OmegaM, q);
    AmpiezzaM = this->Media(At,NpTrasformata);
61 }

```

La validità della procedura di calcolo è stata valutata con la funzione *hilbert* di matlab. In figura 5.14a possiamo osservare un segnale sintetico realizzato con una serie di seni a frequenze diverse, mentre nelle figure 5.14b e 5.14c ho riportato la differenza di numerica ottenuta tra la routine C++ e matlab.

5.2 Interfaccia grafica

Il software si basa su un'interfaccia grafica che può essere utilizzata in tempo reale e che ha diverse funzioni o caratteristiche; le principali sono:

- comandi di avvio e fermo delle funzioni;

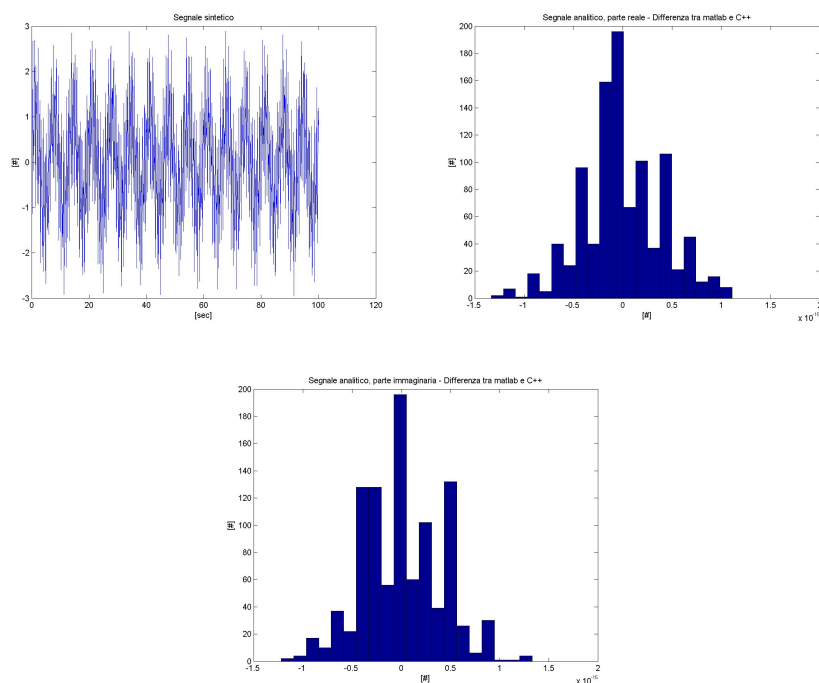


FIGURA 5.14: (a) Segnale sintetico costruito con una somma di sinusoidi a frequenza diversa. (b) e (c) Differenza tra la parte reale ed immaginaria del segnale analitico ottenuto dalla funzione *hilbert* di matlab e la routine C++.

- comandi per la registrazione dei dati;
- pagine dedicate per la visualizzazione dei dati secondo le diverse tecniche di analisi;
- comandi per la gestione dei parametri di configurazione;
- comandi per la visualizzazione *off-line* dei dati registrati.

Lo sviluppo dell'applicazione è stato fatto utilizzando l'ambiente Embarcadero C++ Builder, figura 5.15, che offre un approccio RAD. In questo modo è relativamente semplice costruire elementi grafici a cui associare informazioni o *callback*.

All'avvio l'applicazione si presenta come mostrato nella figura 5.16, la zona centrale è riservata ad un oggetto *TTabSheet* che permette di organizzare le informazioni da visualizzare in pagine selezionabili dalla barra superiore. La zona laterale sinistra è riservata a comandi e visualizzazioni sempre disponibili indipendentemente dalla pagina visualizzata. In alto si trovano i led di segnalazione. Il dispositivo può essere attivato attraverso un interruttore (capitolo 8) ed in questo caso la segnalazione *ACTIVE* diventa verde. L'utilizzo di un interruttore esterno si rende necessaria perché la durata di una siviera è di circa un'ora, mentre i segnali utili si ottengono solo negli ultimi istanti. Quando l'operatore ritiene opportuno attiva il dispositivo. Il led *SLAG* invece segnala l'avvenuta identificazione di un evento scoria. Contemporaneamente vengono attivate le segnalazioni acustica e luminosa presso l'operatore di colata. Il led *GATE* segnala

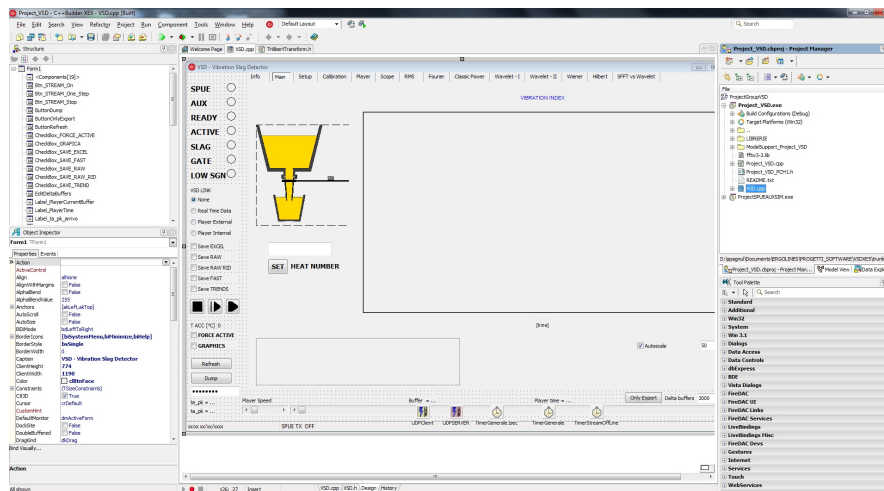


FIGURA 5.15: Ambiente di sviluppo Embarcadero C++ Builder.

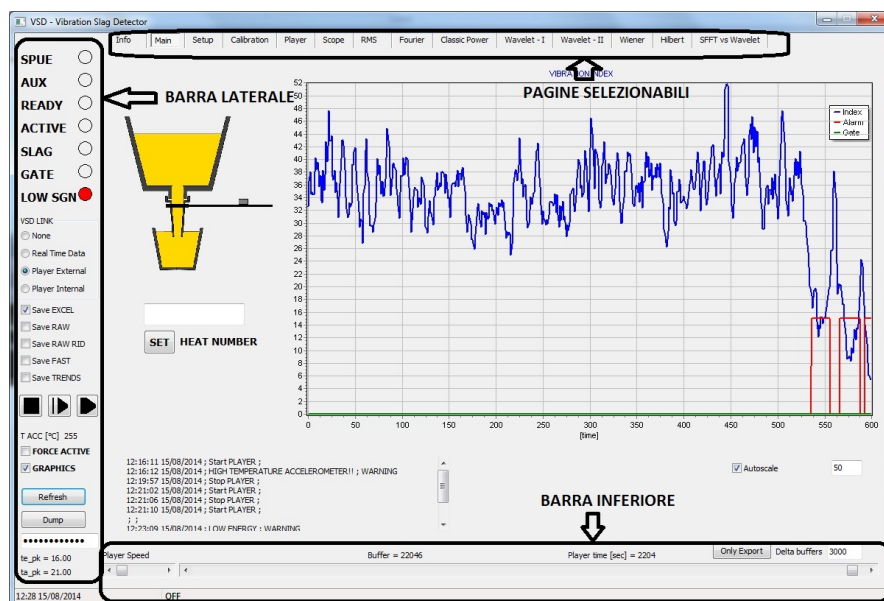


FIGURA 5.16: Organizzazione dell'interfaccia grafica.

invece la movimentazione del cassetto che naturalmente inibisce le routine di calcolo. Sotto le segnalazioni si trovano due gruppi di selettori, i primi permettono di impostare il programma su una specifica modalità mentre i secondi attivano o disattivano la registrazione dei dati. Le modalità sono sostanzialmente due: una in tempo reale, l'altra fuori linea. Sotto questi selettori si trovano tre pulsanti che permettono di avviare o fermare il software in funzione della modalità selezionata. In particolare se lo stato è *real time* allora è possibile attivare o disattivare l'acquisizione dei dati. Nella zona inferiore troviamo due oggetti *TScrollBar*: il primo permette di selezionare la velocità di riproduzione del *player* (modalità fuori linea), mentre con il secondo si può scorrere l'intera sequenza di dati e posizionarsi nella zona desiderata. La pagina *MAIN* contiene un grafico predefinito e selezionabile che rappresenta una delle possibili variabili ottenute dagli algoritmi di calcolo. Inoltre per tracciare i singoli test è possibile inserire il numero

di colata che caratterizza la singola siviera (*heat number*). L'avvio dell'acquisizione *real time* avviene attivando il *thread* che gestisce il servizio UDP: ad ogni ricezione di un nuovo pacchetto dati dall'unità SPUE, viene chiamata una funzione che svolge i calcoli, registra i dati ed infine aggiorna la grafica.

```

1 void __fastcall TForm1::UDPSERVERUDPRead(TIdUDPListenerThread *AThread, const TIdBytes
    AData,
        TIdSocketHandle *ABinding)
3 {
    BYTE BUFFER[20000];
5     if (CntLedLink == 0) CntLedLink = 2;
    if (ABinding->Port == VSD_CONF.SPUETXDATA_PORT) && (ABinding->PeerIP == VSD_CONF.
        SPUERXDATA_IP ) ){
7
            if(VSD_CONF.SPUE_NumBytes_FROM != (AData.Length )) return;
9             BytesToRaw(AData, (BYTE*) BUFFER, AData.Length);
11
                SPUE_Timeout = SPUE_TIMEOUT;
                SPUE->CaricaStream( BUFFER,AData.Length,TSPUEDATA::T_Tipo_SPUE::RAW );
13                GestioneStream();
                if (VSD_ST_INT.GRAFICA_Mostra)AggiornaGrafici();
15
            return;
17     }
19     if( (ABinding->Port == VSD_CONF.LIOETXDATA_PORT) && (ABinding->PeerIP == VSD_CONF.
        LIOERXDATA_IP) ){
                BytesToRaw(AData, (BYTE*) BUFFER, AData.Length);
21                AUX->CaricaStream(BUFFER,AData.Length,VSD_CONF.AUX_VERSIONE,true);
                if(AUX->ErroreLunghezzaPkt() ) {return; };
23                AUX_Timeout = AUX_TIMEOUT;
                VSD_STATI.IsActive = AUX->GetInMarcia();
25            }
    }

```

Come si può osservare dal codice c++ di riferimento, il servizio UDP chiama la funzione `UDPSERVERUDPRead()` al cui interno vengono analizzati gli indirizzi IP. Se il pacchetto proviene dall'unità SPUE allora si procede con il calcolo e la visualizzazione grafica, se i dati provengono dalla unità AUX allora significa che sono segnali digitali provenienti dal PLC di automazione dell'impianto o da interruttori azionati dall'operatore di colata. Di particolare interesse è l'azione che attiva o disattiva l'apparecchiatura. Quando infatti viene attivata tutte le segnalazioni sono disponibili altrimenti il programma si limita ad eseguire i calcoli senza segnalare l'evento scoria. Un oggetto *TTimer* ad intervalli regolari attiva una funzione che si occupa di gestire le segnalazioni luminose come il lampeggiamento dei led che informano sullo stato della comunicazione ethernet. Una serie di pagine dedicate permettono di visualizzare il risultato delle elaborazioni ottenute con gli algoritmi descritti in precedenza. Nella pagina *SCOPE*, figura 5.17, sono visibili i dati grezzi provenienti dall'accelerometro, essi sono visualizzati alla ricezione di ogni nuovo buffer. Nella pagina *FFT*, figura 5.18, si osserva il modulo della trasformata di Fourier relativa

ai buffer acquisiti. Il grafico del valore *rms* globale è riportato nella pagina denominata *RMS*, figura 5.19. Nello stesso grafico è possibile riportare anche il valore mediato con un filtro FIR avente una costante di tempo elevata (traccia nera centrale) e i due limiti ottenuti sommando e sottraendo al valore filtrato la deviazione standard del segnale *rms*. Eventualmente è possibile riportare anche il grafico relativo al segnale che identifica l'evento scoria. Nella pagina *CLASSIC POWER*, figura 5.20, osserviamo superiormente il grafico istantaneo della densità di potenza, mentre inferiormente troviamo raffigurati gli andamenti della densità di potenza anche ad intervalli precedenti. L'asse verticale rappresenta il tempo e le curve in frequenza sono riportate una sopra l'altra, con la possibilità di cambiare il fattore di guadagno (controllo in basso a destra denominato *gain*), amplificando in questo modo ogni singola traccia. La pagina denominata *WAVELET-I*, figura 5.21 riporta la sintesi dei risultati della trasformata wavelet packet discreta. In particolare sono riportati i valori *rms* ottenuti dal banco di filtri. Le singole tracce sono verticalmente spostate in modo tale che la banda bassa sia associata alla traccia inferiore e quella alta alla superiore. Si ottiene in questo modo un'efficace visualizzazione tempo frequenza. La tabella in alto a sinistra riporta i calcoli relativi ai valori energetici per verificare se la somma delle singole bande coincide con il valore globale, ovvero serve a controllare che la condizione di ricostruzione perfetta venga rispettata. Nella pagina *WAVELET-II*, figura 5.22, sono riportate le trasformate inverse delle trasformate *wavelet packet* relative ad ogni singolo ramo del banco di filtri. La funzione di questa rappresentazione è essenzialmente di verifica del corretto funzionamento degli algoritmi, essendo chiaramente inutilizzabile in tempo reale. Nella pagina denominata *HILBERT*, figura 5.23, sono riportate l'ampiezza e la frequenza istantanea ottenute dall'algoritmo della trasformata di Hilbert in tempo reale. Oltre che per ogni canale la trasformata di Hilbert può essere applicata anche ad ogni singolo ramo del banco di filtri *wavelet*. Nella figura 5.24 si può osservare lo schema di calcolo applicato. Nella pagina *SFFT vs Wavelet*, figura 5.25, sono raffigurate due immagini aggiornate in tempo reale. Quella inferiore rappresenta una visione a falsi colori dello spettrogramma relativo ad un predeterminato numero di buffer. Il grafico superiore rappresenta il risultato analogo ottenibile dalla trasformata *wavelet*. Le due immagini sono state create secondo la seguente procedura:

- calcolo della densità di potenza o della trasformata *wavelet*;
- aggiornamento di un buffer bidimensionale rappresentante lo spazio tempo-frequenza;
- normalizzazione dei valori;
- creazione di una *palette* grafica con una predefinita selezione cromatica;
- costruzione dell'immagine jpeg;
- visualizzazione dell'immagine.

Mentre lo spettrogramma può efficacemente dare informazioni sulle bande risonanti, osserviamo che l'equivalente ottenuto con i valori della trasformata *wavelet* non si rivela altrettanto valido, contrariamente alla figura 5.21 che invece mostra una capacità informativa nettamente superiore. La pagina *CONFIG*, figura 5.26, mostra la configurazione del funzionamento del software direttamente da interfaccia grafica. Ogni elemento del calcolo è stato associato ad una variabile che può essere modificata attraverso un file di configurazione che viene letto all'avvio del programma. Alcuni parametri però sono accessibili anche attraverso questa pagina, in modo da poterli cambiare in tempo reale senza dover riavviare il software.

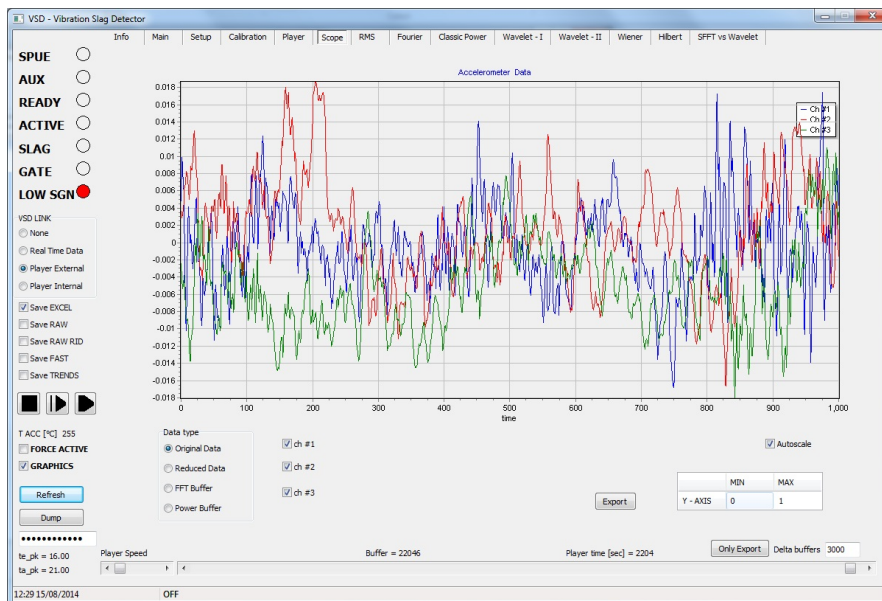


FIGURA 5.17: La pagina *SCOPE* mostra in tempo reale i dati acquisiti dall'accelerometro.

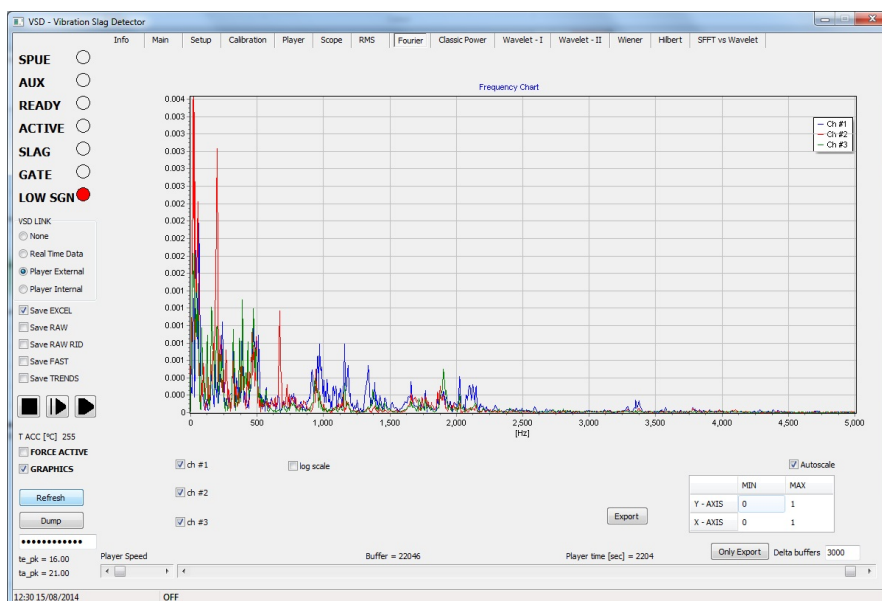


FIGURA 5.18: La pagina *FFT* mostra in tempo reale la trasformata di Fourier del singolo buffer, o di una serie predefinita di buffer contigui.

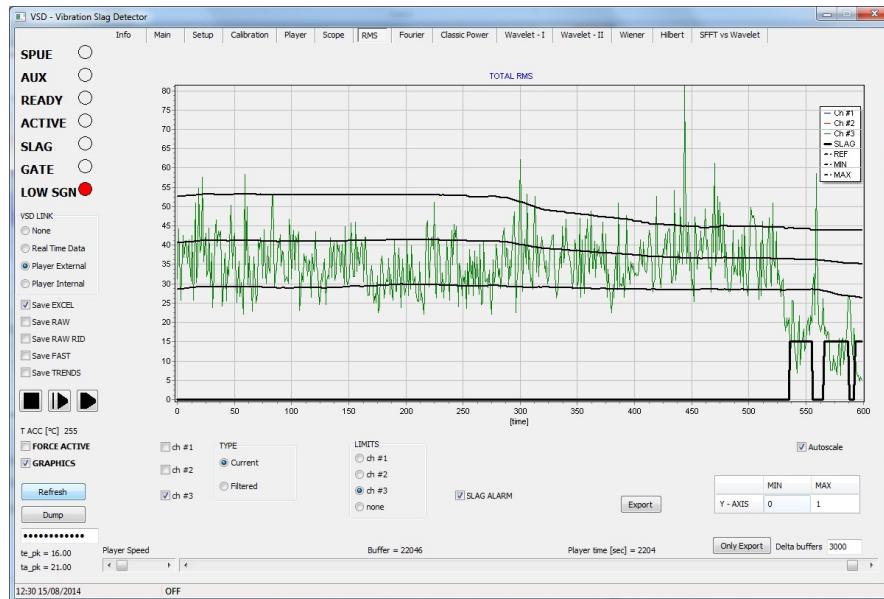


FIGURA 5.19: La pagina *RMS* mostra in tempo reale l'andamento del valore *rms* del singolo buffer. Le tracce nere rappresentano rispettivamente il valore filtrato, il valore filtrato sommato la deviazione standard, il valore filtrato con sottratta la deviazione standard.

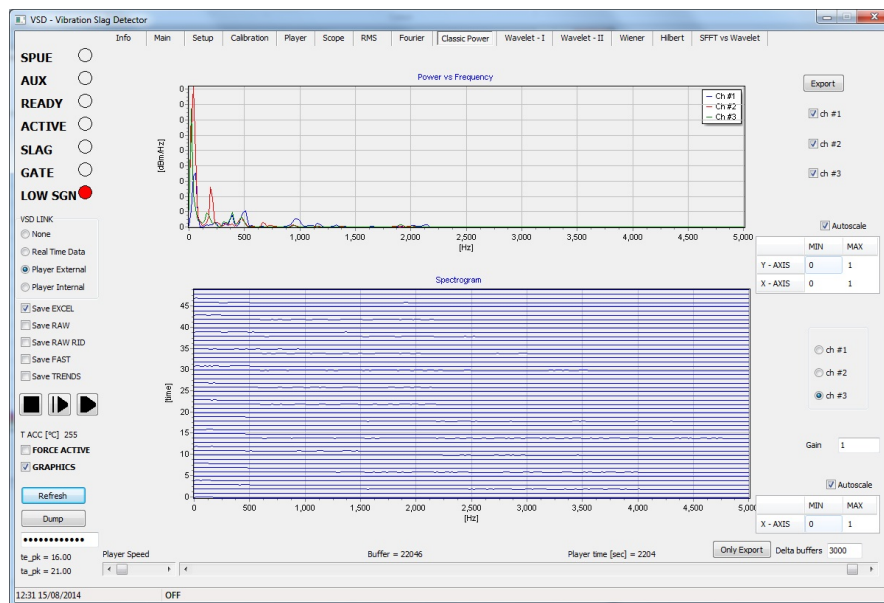


FIGURA 5.20: La pagina *CLASSIC POWER* mostra in tempo reale la densità di potenza secondo il metodo di Welch nel grafico superiore, mentre in quello inferiore è possibile osservare una serie predeterminata di curve di densità di potenza che rappresentano uno spettrogramma

Infine nella pagina *PLAYER* sono visibili i comandi per caricare in memoria i dati registrati e visualizzarli in modalità *off-line*. Selezionando una data nel calendario vengono mostrati tutti i file registrati in quella giornata. Alcuni possono essere selezionati e caricati. L'oggetto che si occupa della gestione della riproduzione dei file registrati è stato chiamato *TPlayer* il cui costruttore è riportato nel codice seguente.

```
1 TPLAYER::TPLAYER(stVSD_POD_Config *pVSD_CONF, TBufferAccelerometri3D *pACC3, TSPUEDATA
    *pSPUE, TAUXDATA *pAUX) {
```

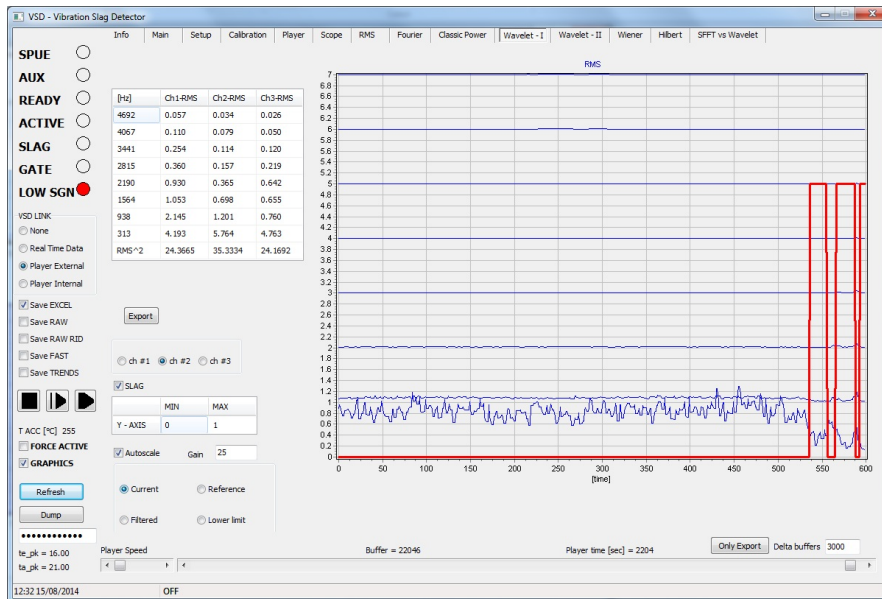


FIGURA 5.21: La pagina WAVELET-I mostra in tempo reale l'esito dell'analisi mediante la trasformata wavelet packet discreta.

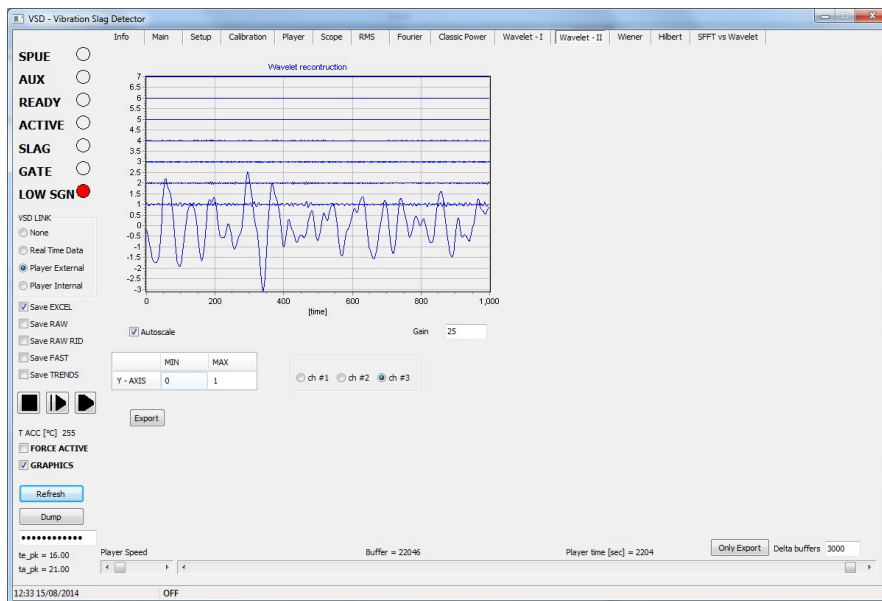


FIGURA 5.22: La pagina WAVELET-II mostra in tempo reale le trasformate inverse delle trasformate wavelet packet relative ad ogni singolo ramo del banco di filtri.

```

3  VSD_CONF = pVSD_CONF;

5  ACC3D = pACC3;
   SPUE = pSPUE;
7  AUX = pAUX;

9  NCicloStreamOffLine=0;
   Npk_analitico =0;
11 NPuntiBuffer = VSD_CONF->NPuntiBufferOriginale;
   pCh1 = new double[NPuntiBuffer];

```




FIGURA 5.23: La pagina *HILBERT* mostra in tempo reale l'andamento della frequenza ed ampiezza istantanea ottenuti dall'applicazione della trasformata di Hilbert.

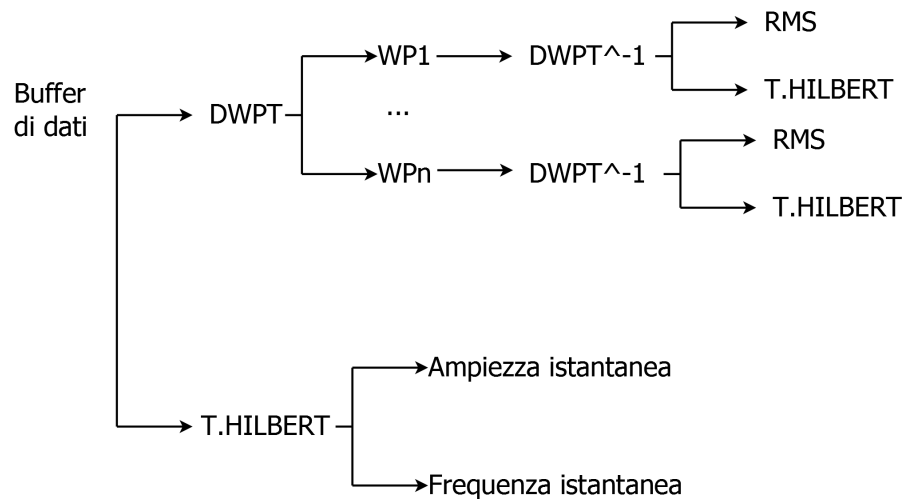


FIGURA 5.24: Modalità di applicazione delle trasformate Hilbert e wavelet.

```

13  pCh2 = new double[NPuntiBuffer];
    pCh3 = new double[NPuntiBuffer];
15  FreqOrig = VSD_CONF->FreqCampSpue;
    TipoPlayer = T_TipoPLAYER::INTERNO_FREQ_ORIG;
17
    STREAM_FILES_DATA = new TSTREAMDAFILE (VSD_CONF->SPUE_NumBytes_FROM, VSD_CONF->
        PLAYER_FILES_IN_MEMORY);
19  iBufferInFiles = 0;
    MaxBufferInFiles = 0;
21  ErroreCaricamentoFiles = false;
    DirAttuale = "";
23  StreamCaricato = false;
    CurrentVersionRAWRID = 0;
25  }
  
```

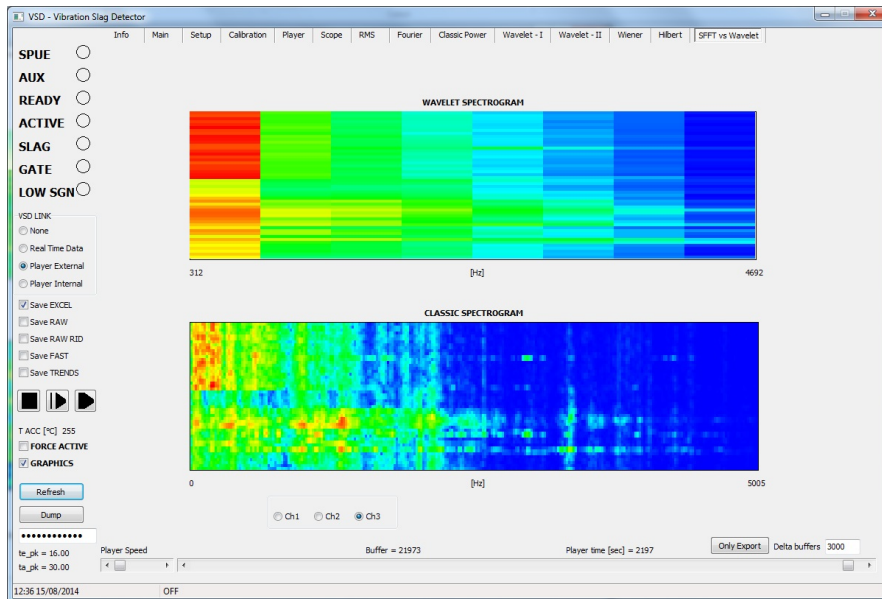


FIGURA 5.25: La figura mostra la pagina dello spettrogramma di Welch (in basso) e l'equivalente ottenuto con la trasformata *wavelet Packets*.

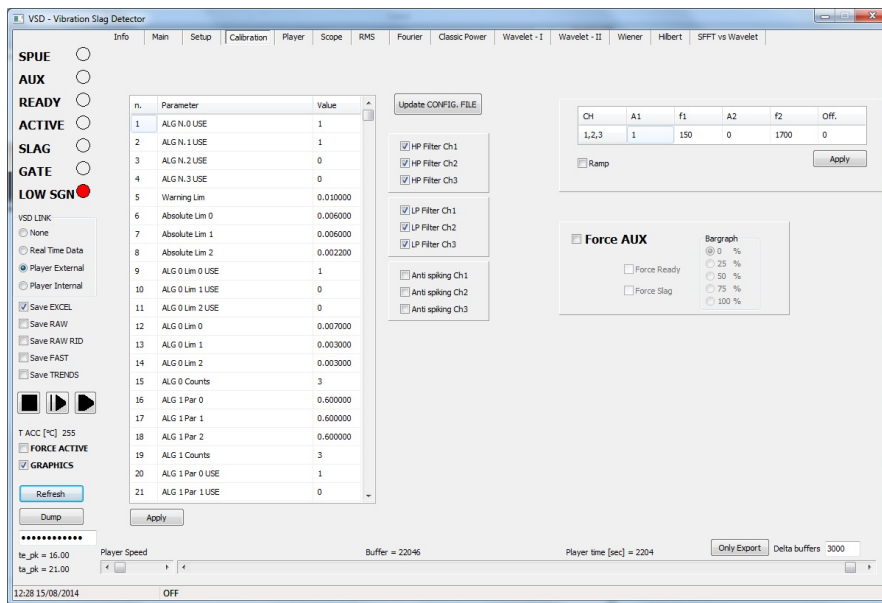


FIGURA 5.26: La figura mostra la pagina in cui è possibile modificare i parametri più importanti degli algoritmi.

Esso contiene al suo interno un altro oggetto fondamentale che permette di costruire un flusso binario di dati a partire da un elenco di file. La classe `TSTREAMDAFILE` permette di creare un elenco di oggetti `TFILEBINARI` che costituiscono l'interfaccia al file fisico.

```

1 TSTREAMDAFILE::TSTREAMDAFILE(int numBytesBuffer, bool fileInMemory){
2
3     FILES_BINARI = new TFILEBINARI*[MAX_FILES_OPEN ];
4     for (int i = 0; i < MAX_FILES_OPEN; i++) FILES_BINARI[i]=NULL;
5     FileInMemory = fileInMemory;
6     NumBytesPerBuffer = numBytesBuffer;

```

```

7     NumFilesOpen = 0;
      CurrentFile  = -1;
9     NumBuffers   = 0;
      BuffersPerFile = 0;
11    TotBytes     = 0;
    }
13    TFILIBINARI::TFILIBINARI(int numBytesBuffer, AnsiString nomefile, bool scrittura, bool
      fileinmemory){

15        div_t  x;

17        NumBytesPerBuffer = numBytesBuffer;
      NumBuffers           = 0;
19        NumBytesTot      = 0;
      PuntatoreByte       = 0;
21        PuntatoreBuffer  = 0;
      pFile                = NULL;
23        Errore           = true;
      Scrittura            = scrittura;
25        if(NumBytesPerBuffer<1) return;
      FileInMemory        = fileinmemory;
27        BufferAll         = NULL;
      Buffer                = NULL;
29        if (scrittura) {
            pFile = fopen(nomefile.c_str(), "wb");
31            if (!pFile){ Errore = false; Scrittura = true;}
        } else {
33            pFile = fopen(nomefile.c_str(), "rb");
            if (pFile) {
35                fseek(pFile,0,SEEK_END);
                NumBytesTot = ftell(pFile);
37                x = div(NumBytesTot,NumBytesPerBuffer);
                if (x.rem > 0) {return;} //se non un multiplo segnala l'
errore
39                NumBuffers = x.quot;
                rewind(pFile);
41                Buffer      = new UINT8[NumBytesPerBuffer];
                if (FileInMemory && (NumBytesTot<MAX_BYTES_FILE_IN_MEMORY)){
43                    BufferAll = new UINT8[NumBytesTot];

45                    fread(BufferAll, sizeof(UINT8), NumBytesTot,
pFile);

                    fclose(pFile);

47                    } else {
49                        rewind(pFile);
                        FileInMemory = false;
51                    }
                    Errore      = false;
53                }
            }
55    }

```

Quando il *Player* viene attivato la sequenza fondamentale di codice che viene eseguita è riportata di seguito.

```

1 // All'interno della funzione chiamata dal timer che gestisce l'oggetto PLAYER
  .
3 .
  .
5 m = PLAYER->GeneraStream();
  .
7 .
  .
9 // All'interno di "GeneraStream()"
  .
11 .
  .
13 STREAM_FILES_DATA->GetBuffer(iBufferInFiles,pb,nB)
  .
15 .
  .
17 // All'interno di "GetBuffer()"
  int TSTREAMDATAFILE::GetBuffer(int ibuffer,UINT8* &pbuf, int &nB){
19   if (NumFilesOpen<1) return -1;

21   div_t  x;
     int ib,ibj;
23   ib=ibuffer;
     if (ibuffer<0) ib = 0;
25   if (ibuffer>NumBuffers) ib = NumBuffers-1;

27   x = div(ib,BuffersPerFile);
     CurrentFile = x.quot;
29   ibj = x.rem;
     if (FILES_BINARI[CurrentFile]->GetBuffer(ibj,pbuf,nB)<0) return -1;
31   return CurrentFile;
  //
33 int TFILEBINARI::GetBuffer(int ibuffer,UINT8* &pbuf, int &nB){
     if (Errore) return -1;
35   if (SetBufferPointer(ibuffer)<0) return -1;

37   if (FileInMemory) {
     pbuf = &BufferAll[PuntatoreByte];
39   } else {
     fseek(pFile,PuntatoreByte,SEEK_SET);
41     fread(Buffer,sizeof(UINT8),NumBytesPerBuffer,pFile);
     pbuf = Buffer;
43   }
     nB = NumBytesPerBuffer;
45
     return 0;
47   }

```

Come si può osservare data la posizione del buffer, *iBufferInFiles*, nell'intero flusso di dati attraverso il meccanismo descritto si risale al singolo file e alla posizione del buffer all'interno

di esso. Cambiando la posizione della barra di scorrimento inferiore si cambia semplicemente l'indice del buffer (funzione ScrollBar_PlayerTimeChange).

```

1 void __fastcall TForm1::ScrollBar_PlayerTimeChange(TObject *Sender)
  {
2     int m;
3     if(VSD_STATI.IsPlayerON) return;
4     PLAYER->SetBufferInFiles(ScrollBar_PlayerTime->Position*10);
5     m = 10*ScrollBar_PlayerTime->Position;
6     Label_PlayerTime->Caption = AnsiString("Player time [sec] = ") + IntToStr(m
7     /10);
8     Label_PlayerCurrentBuffer->Caption = AnsiString("Buffer = ") + m;
9
10  }

```

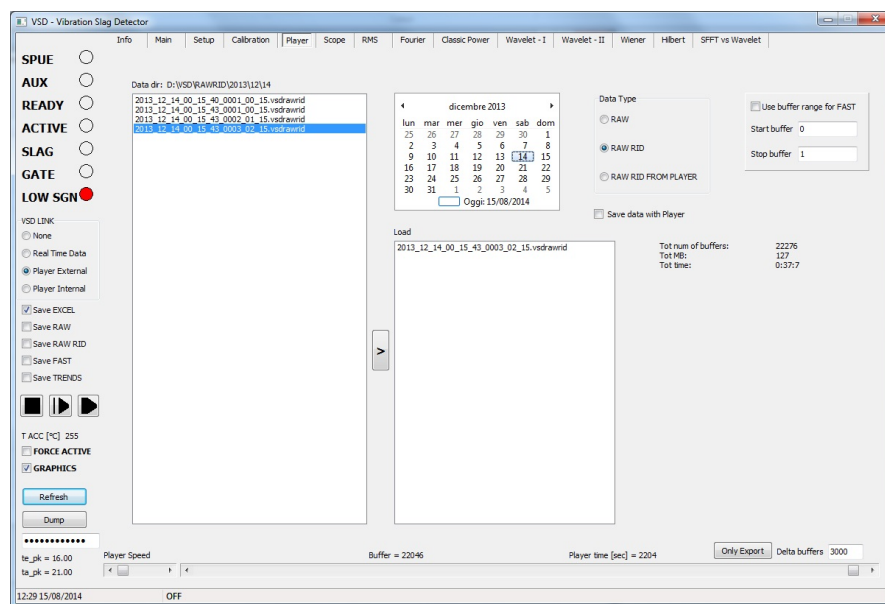


FIGURA 5.27: La pagina del PLAYER mostra un calendario ed alcuni box in cui sono visibili i file. Attraverso il tasto centrale è possibile caricarli in memoria.

Capitolo 6

Simulazione numerica con acqua

6.1 Introduzione

Il problema fluidodinamico dello svuotamento della siviera consiste nella soluzione delle equazioni di *Naviers-Stokes* nell'approssimazione di fluidi newtoniani incomprimibili con proprietà termofisiche costanti.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (6.1a)$$

$$\frac{\partial}{\partial t}(\rho \mathbf{U}) + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p + \nabla \cdot (\mu \nabla \mathbf{U}) \quad (6.1b)$$

Le equazioni (6.1) descrivono in modo corretto sia il comportamento del fluido in condizioni laminari che turbolente, pertanto le equazioni inglobano la natura caotica della turbolenza. In linea di principio soluzioni numeriche dirette sufficientemente accurate potrebbe risolvere il problema, tuttavia vista l'enorme richiesta di potenza di calcolo nel corso del tempo sono stati sviluppati dei modelli che caratterizzano la turbolenza ed in particolare descrivono il trasferimento di energia dalle grandi alle piccole scale. In ultima analisi quindi applicare un modello di turbolenza significa rinunciare allo studio dei dettagli del comportamento del fluido introducendo un fluido equivalente descritto da opportune equazioni costitutive. Certamente la teoria della turbolenza sviluppata da *Kolmogorov* nel 1930 costituisce la base dei modelli attuali. Due sono gli approcci fondamentali alla modellazione della turbolenza:

- modelli LES, *Large Eddy Simulation*. Si basano su un processo di filtraggio spaziale;
- modelli RANS, *Reynolds Averaged Navier Stokes*. Si basano su un processo di filtraggio temporale.

I LES si basano sul fatto che le strutture a più grande scala non si prestano ad una modellazione generale e al contempo sono responsabili del trasporto convettivo della quantità di moto e dell'energia turbolenta. Al contrario le strutture di piccola scala sono sostanzialmente dissipative ed universali, quindi indipendenti dal problema specifico. Esse quindi si prestano ad una modellazione generale e semplificata. Nei modelli RANS invece viene introdotto un filtro rispetto al tempo, in pratica una media a lungo periodo. E' necessario quindi che il moto sia globalmente stazionario, quindi gli unici casi in cui un modello di turbolenza RANS può essere usato sono quelli in cui le scale temporali del transitorio sono completamente diverse da quelle propriamente turbolente. In questa categoria i modelli più noti sono:

- $k - \epsilon$ (Jones-Launder, 1972);
- $k - \omega$ (Wilcox, 1988);
- SST (Menter, 1993).

Il primo è un modello a due equazioni basato sulla scelta dell'energia turbolenta per unità di massa (k) e del suo tasso di dissipazione (ϵ) come grandezze caratterizzanti della turbolenza. Il secondo è sostanzialmente equivalente solo che utilizza $\omega = \epsilon/k$ come dissipazione turbolenta specifica. Il terzo modello invece tenta di coniugare i benefici dei precedenti. Esso vuole sfruttare la buona accuratezza e robustezza del modello $k - \omega$ vicino alla parete con l'indipendenza dalle condizioni al contorno di quello $k - \epsilon$ in zone di flusso libero. Un altro modello che rientra in questa categoria è chiamato $v^2 - f$ in cui si introduce un termine non lineare (del secondo ordine) per il calcolo della velocità turbolenta nell'approccio $k - \epsilon$ ed in questo modo si ottengono alcuni vantaggi ai bassi numeri di Reynolds. Per completezza, esistono anche i modelli *RMS* che sono delle evoluzioni dei modelli storici $k - \epsilon$.

Osservando la figura (6.1) notiamo sia la presenza di una superficie libera sia di un numero di fasi maggiore di uno. L'ipotesi semplificata è che si tratta di fasi immiscibili. In generale la procedura che deve tracciare l'andamento della superficie libera, oltre alla risoluzione delle equazioni (6.1a) e (6.1b) deve appunto calcolare la forma, la sua evoluzione temporale tenendo in debita considerazione vari fattori, come ad esempio la tensione superficiale. Il trattamento dei flussi multifase in cui la superficie libera viene appunto inserita si dividono in due grandi gruppi:

- metodi di superficie. In questo caso l'interfaccia viene inserita con dei markers oppure viene deformata opportunamente la mesh.
- metodi di volume. Uno dei fluidi viene marcato con delle particelle senza massa o con una funzione indicatrice.

Poiché il software scelto per le simulazioni è *openFoam*, l'approccio alla superficie libera si basa sul volume di fluido (VOF - *Volume of Fluid*). Questo metodo prevede la definizione di una funzione scalare compresa nell'intervallo $[0, 1]$ con la quale distinguere il tipo di fluido. Da un punto di vista formale essa è definita così,

$$\alpha^k = \lim_{V \rightarrow \infty} \frac{1}{V} \int \int \int \chi^k(x, y, z, t) dV \quad (6.2)$$

dove χ^k è identicamente nulla e diventa unitaria se il punto (x, y, z) è nella fase k all'istante t . Chiaramente nel volume discretizzato un valore di α^k intermedio indica la proporzione tra le varie fasi. Per descrivere l'evoluzione della frontiera α deve soddisfare l'equazione della conservazione di massa,

$$\frac{\partial \rho^k \alpha^k}{\partial t} + \nabla \cdot (\rho^k \alpha^k \mathbf{U}^k) = \Gamma^k \quad (6.3)$$

con Γ^k flusso di massa relativo alla k -esima fase. All'interfaccia le proprietà del fluido vengono ottenute attraverso la media pesata delle proprietà delle singole fasi.

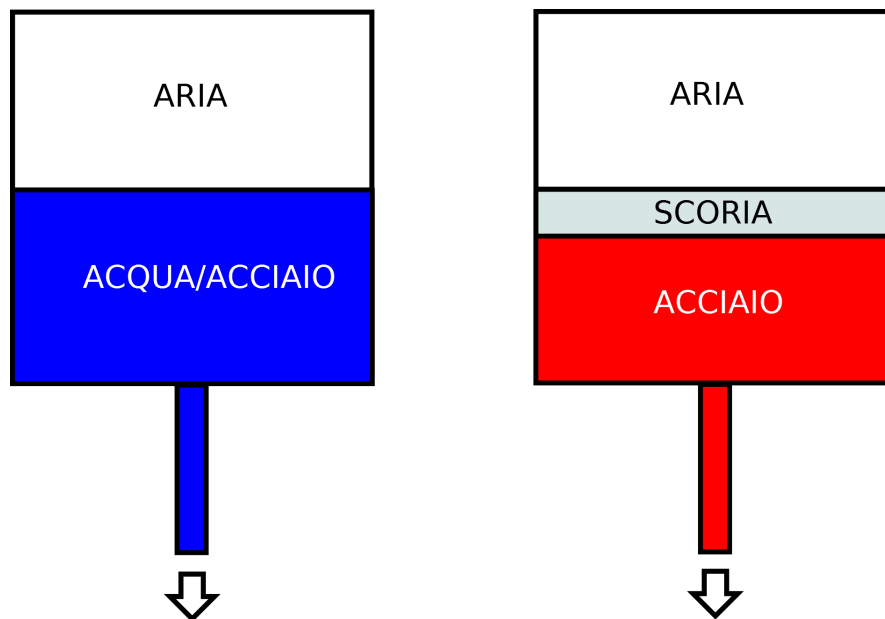


FIGURA 6.1: Schemi fisici del problema fluidodinamico.

Come accennato in precedenza il software utilizzato per la risoluzione del problema fluidodinamico è *openFoam*, rilasciato sotto la licenza GNU in ambiente linux e nasce nel 1993. Esso è un insieme di librerie C++ , per la soluzione di equazioni a derivate parziali mediante la tecnica dei volumi finiti. Essendo scritto in linguaggio C++ il software in questione fa proprie alcune caratteristiche della programmazione *object-oriented* che consente di modificare parti del programma od introdurne di nuove con relativa semplicità. Il codice prevede uno spazio utente suddiviso in due distinte sottocartelle. La prima, relativa alle *applications*, contiene gli algoritmi di soluzione e le utilità di *preprocessing* e *postprocessing* creati dall'utente e compilati con il

comando *wmake*. Nella seconda cartella denominata *run* si trovano invece i problemi affrontati. Ogni problema a sua volta ha una struttura molto precisa di directory:

- directory *0*;
- directory *constant*;
- directory *system*.

In *0* saranno poste le condizioni al contorno e quelle iniziali. In *constant* si troverà la *mesh* e parametri relativi alla scelta del tipo di simulazione e modello di turbolenza. In *system* infine si troveranno informazioni quali ad esempio il tipo di *solver*, il passo temporale, gli schemi di approssimazione per le derivate.

6.2 Caso bifase

6.2.1 Modello geometrico e *mesh*

Il modello geometrico utilizzato è relativamente semplice e ripropone il modello ad acqua che verrà utilizzato per la verifica dei risultati teorici. Il punto critico naturalmente è la realizzazione della *mesh*. Sebbene sia possibile realizzare una discretizzazione utilizzando dei tetraedri, per massimizzare l'accuratezza del calcolo è opportuno realizzare una *mesh* di tipo strutturato, ovvero utilizzando esaedri. A tal fine si è utilizzato il pacchetto software incluso in *openFoam* chiamato *blockMesh*. Per la particolare geometria del problema è necessario utilizzare la tecnica *o-grid*, infatti sull'asse del sistema la realizzazione degli esaedri presenta delle difficoltà. Introducendo invece un dominio fittizio a forma di parallelepipedo è possibile collegare gli esaedri con superficie curvilinea che nascono dalla periferie con quelli centrali che invece non hanno superfici curvilinee. Come si può infatti osservare dalla figura 6.2, i nodi 24,25,26,27,20,21,22,23,28,29,30 e 31 definiscono un parallelepipedo centrale incluso nel cilindro definito dai nodi 4,5,6,7,0,1,2,3,16,17,18 e 19. La parte superiore di questo cilindro rappresenta appunto la prosecuzione dell'ugello di scarico inferiore. Lo *script* che definisce la suddetta geometria è di seguito riportato.

```

/*-----* C++ *-----*/
2 | ===== |
| \\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
4 | \\ / O p e r a t i o n | Version: 2.1.0 |
| \\ / A n d | Web: www.OpenFOAM.org |
6 | \\ / M a n i p u l a t i o n | |
/*-----*/
8 FoamFile
{

```

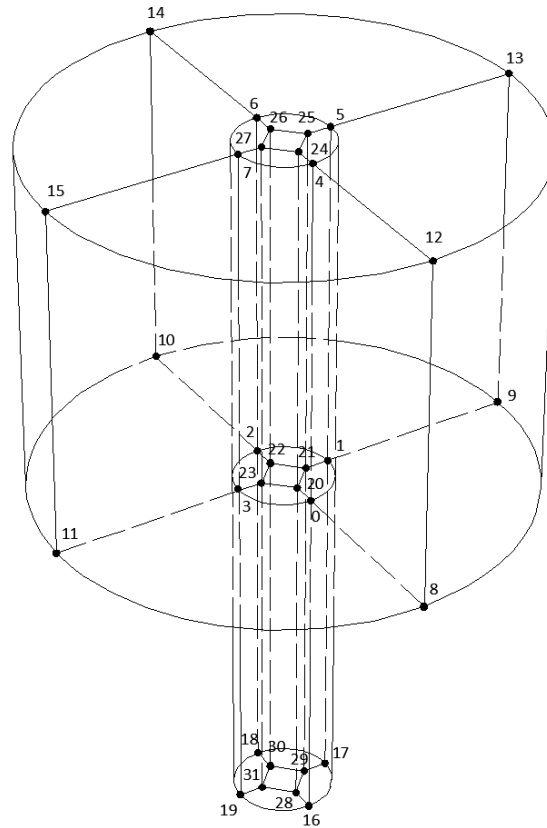


FIGURA 6.2: Schema dei blocchi per la realizzazione della *mesh* strutturata.

```

10     version      2.0;
        format      ascii;
12     class        dictionary;
        object       blockMeshDict;
14 }
    // * * * * * //
16
    convertToMeters 0.001;
18
    vertices
20 (
        (0 -5 50)           //0
22     (5 0 50)             //1
        (0 5 50)            //2
24     (-5 0 50)           //3
        (0 -5 105)          //4
26     (5 0 105)           //5
        (0 5 105)           //6
28     (-5 0 105)         //7
        (0 -105 50)         //8
30     (105 0 50)          //9
        (0 105 50)          //10
32     (-105 0 50)        //11
        (0 -105 105)        //12
34     (105 0 105)         //13
        (0 105 105)         //14
    )
    
```

```

36     (-105 0 105)           //15
      (0 -5 00)             //16
38     (5 0 00)             //17
      (0 5 00)             //18
40     (-5 0 00)           //19
      (0 -3.536 50)        //20
42     (3.536 0 50)        //21
      (0 3.536 50)        //22
44     (-3.536 0 50)       //23
      (0 -3.536 105)       //24
46     (3.536 0 105)       //25
      (0 3.536 105)       //26
48     (-3.536 0 105)     //27
      (0 -3.536 00)        //28
50     (3.536 0 00)        //29
      (0 3.536 00)        //30
52     (-3.536 0 00)      //31
);

54 blocks
56 (
      hex (0 1 21 20 4 5 25 24) (12 4 110) simpleGrading (1 0.8 1)
58     hex (21 1 2 22 25 5 6 26) (4 12 110) simpleGrading (1.25 1 1)
      hex (22 2 3 23 26 6 7 27) (4 12 110) simpleGrading (1.25 1 1)
60     hex (23 3 0 20 27 7 4 24) (4 12 110) simpleGrading (1.25 1 1)
      hex (20 21 22 23 24 25 26 27) (12 12 110) simpleGrading (1 1 1)
62     hex (8 9 1 0 12 13 5 4) (12 100 110) simpleGrading (1 0.25 1)
      hex (1 9 10 2 5 13 14 6) (100 12 110) simpleGrading (4 1 1)
64     hex (2 10 11 3 6 14 15 7) (100 12 110) simpleGrading (4 1 1)
      hex (3 11 8 0 7 15 12 4) (100 12 110) simpleGrading (4 1 1)
66
      hex (16 17 29 28 0 1 21 20) (12 4 100) simpleGrading (1 0.8 1)
68     hex (17 18 30 29 1 2 22 21) (12 4 100) simpleGrading (1 0.8 1)
      hex (18 19 31 30 2 3 23 22) (12 4 100) simpleGrading (1 0.8 1)
70     hex (19 16 28 31 3 0 20 23) (12 4 100) simpleGrading (1 0.8 1)
      hex (28 29 30 31 20 21 22 23) (12 12 100) simpleGrading (1 1 1)
72 );

74 edges
   (
76     arc 0 1 (3.536 -3.536 50.00)
      arc 1 2 (3.536 3.536 50.00)
78     arc 2 3 (-3.536 3.536 50.00)
      arc 3 0 (-3.536 -3.536 50.00)
80     arc 4 5 (3.536 -3.536 105.00)
      arc 5 6 (3.536 3.536 105.00)
82     arc 6 7 (-3.536 3.536 105.00)
      arc 7 4 (-3.536 -3.536 105.00)
84     arc 8 9 (74.246 -74.246 50.00)
      arc 9 10 (74.246 74.246 50.00)
86     arc 10 11 (-74.246 74.246 50.00)
      arc 11 8 (-74.246 -74.246 50.00)
88     arc 12 13 (74.246 -74.246 105.00)
      arc 13 14 (74.246 74.246 105.00)
90     arc 14 15 (-74.246 74.246 105.00)

```

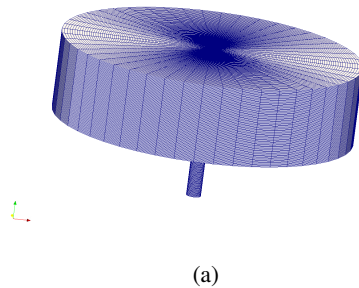
```
    arc 15 12 (-74.246 -74.246 105.00)
92   arc 16 17 (3.536 -3.536 00.00)
    arc 17 18 (3.536 3.536 00.00)
94   arc 18 19 (-3.536 3.536 00.00)
    arc 19 16 (-3.536 -3.536 00.00)
96   );

98   boundary
    (
100   outlet
    {
102     type patch;
    faces
104     (
        (29 17 16 28)
106     (30 18 17 29)
        (31 19 18 30)
108     (28 16 19 31)
        (31 30 29 28)
110     );
    }
112
    atmo
114   {
    type patch;
116   faces
    (
118     (4 5 25 24)
        (5 6 26 25)
120     (6 7 27 26)
        (7 4 24 27)
122     (24 25 26 27)
        (12 13 5 4)
124     (5 13 14 6)
        (6 14 15 7)
126     (15 12 4 7)
    );
128   }

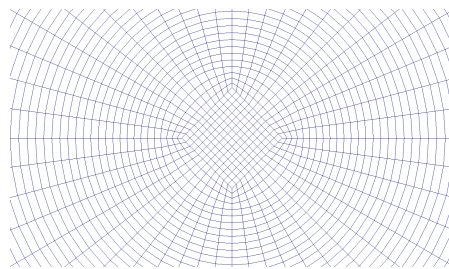
130   defaultFaces
    {
132     type wall;
    faces ();
134   }
    );
136
mergePatchPairs
138 (
    );
140 // ***** //
```

Nella prima parte (*vertices*), vengono definiti i nodi della geometria. In *blocks* vengono definiti i blocchi ed il numero di elementi per singolo blocco mentre in *edges* ed in *boundary* vengono

definiti rispettivamente gli spigoli curvi e le superfici esterne del dominio di calcolo. Si nota che il numero di nodi in cui dividere la *mesh* viene stabilito mediante il comando `hex (...)` ($n1\ n2\ n3$) `simpleGrading (ax ay az)` che appunto organizza la suddivisione dei lati di uno dei blocchi rappresentato in figure 6.2. Le superfici su cui verranno definite le condizioni al contorno sono quella superiore *atmo* e quella inferiore *outlet* dove fuoriesce il liquido. Tutte le rimanenti superfici esterne sono definite come *wall* ovvero come pareti.



(a)

FIGURA 6.3: Griglia (*mesh*) strutturata.

(a)

FIGURA 6.4: Particolare della tecnica *o-grid*.

6.2.2 Modello in *openFoam*

Prima di descrivere in dettaglio i parametri del modello, è opportuno illustrare il *solver InterFoam* con cui si risolve attraverso il metodo VOF il caso di flussi bifasici con fasi isoterme, non comprimibili ed immiscibili. Di seguito riporto il codice C++.

```

/*-----*\
2  ===== |
   \\      / F i e l d       | OpenFOAM: The Open Source CFD Toolbox
4   \\      / O p e r a t i o n |
   \\      / A n d           | Copyright (C) 2011 OpenFOAM Foundation
6   \\/      M a n i p u l a t i o n |
-----*/

8 License
   This file is part of OpenFOAM.

10
   OpenFOAM is free software: you can redistribute it and/or modify it
12   under the terms of the GNU General Public License as published by
   the Free Software Foundation, either version 3 of the License, or
14   (at your option) any later version.

```

```

16     OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
17     ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
18     FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
19     for more details.
20
21     You should have received a copy of the GNU General Public License
22     along with OpenFOAM.  If not, see <http://www.gnu.org/licenses/>.
23
24 Application
25     interFoam
26
27 Description
28     Solver for 2 incompressible, isothermal immiscible fluids using a VOF
29     (volume of fluid) phase-fraction based interface capturing approach.
30
31     The momentum and other fluid properties are of the "mixture" and a single
32     momentum equation is solved.
33
34     Turbulence modelling is generic, i.e. laminar, RAS or LES may be selected.
35
36     For a two-fluid approach see twoPhaseEulerFoam.
37
38 \*-----*/
39
40 #include "fvCFD.H"
41 #include "MULES.H"
42 #include "subCycle.H"
43 #include "interfaceProperties.H"
44 #include "twoPhaseMixture.H"
45 #include "turbulenceModel.H"
46 #include "interpolationTable.H"
47 #include "pimpleControl.H"
48
49 // * * * * *
50
51 int main(int argc, char *argv[])
52 {
53     #include "setRootCase.H"
54     #include "createTime.H"
55     #include "createMesh.H"
56
57     pimpleControl pimple(mesh);
58
59     #include "initContinuityErrs.H"
60     #include "createFields.H"
61     #include "readTimeControls.H"
62     #include "correctPhi.H"
63     #include "CourantNo.H"
64     #include "setInitialDeltaT.H"
65
66     // * * * * *
67
68     Info<< "\nStarting time loop\n" << endl;

```

```

70     while (runTime.run())
71     {
72         #include "readTimeControls.H"
73         #include "CourantNo.H"
74         #include "alphaCourantNo.H"
75         #include "setDeltaT.H"
76
77         runTime++;
78
79         Info<< "Time = " << runTime.timeName() << nl << endl;
80
81         twoPhaseProperties.correct();
82
83         #include "alphaEqnSubCycle.H"
84
85         // --- Pressure-velocity PIMPLE corrector loop
86         while (pimple.loop())
87         {
88             #include "UEqn.H"
89
90             // --- Pressure corrector loop
91             while (pimple.correct())
92             {
93                 #include "pEqn.H"
94             }
95
96             if (pimple.turbCorr())
97             {
98                 turbulence->correct();
99             }
100         }
101
102         runTime.write();
103
104         Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
105             << " ClockTime = " << runTime.elapsedClockTime() << " s"
106             << nl << endl;
107     }
108
109     Info<< "End\n" << endl;
110
111     return 0;
112 }
113
114 // *****

```

Sostanzialmente il ciclo di una singola iterazione è così organizzato:

- risoluzione dell'equazione di trasporto della frazione di volume α (alphaEqnSubCycle.H) ed aggiornamento del valore della densità delle celle della *mesh*;

- risoluzione del campo di moto (UEqn.H), delle pressioni (pEqn.H) e aggiornamento delle grandezze turbolente secondo lo schema dell'algoritmo PIMPLE

L'algoritmo PIMPLE è la fusione del metodo PISO e SIMPLE. La prima azione consiste nell'uso di SIMPLE il quale calcola la pressione dalle componenti di velocità applicando una procedura iterativa accoppiata con le equazioni di *Navier-Stokes* (Ferziger and Peric, 1999) poi utilizza l'algoritmo PISO per le correzioni della pressioni ed infine corregge esplicitamente sia la pressione che la velocità. Al termine dell'iterazione viene scritta la soluzione nella directory *run*. Osserviamo inoltre che ad ogni iterazione viene calcolato il numero di *Courant* (*Courant-No.H*). Esso riflette la porzione di cella che è attraversata per trasporto in un passo temporale ed è definito nel caso tridimensionale come:

$$C = \frac{u_x \Delta t}{\Delta x} + \frac{u_y \Delta t}{\Delta y} + \frac{u_z \Delta t}{\Delta z}. \quad (6.4)$$

La condizione $C < C_{\max}$ è necessaria ma non sufficiente per la convergenza dell'algoritmo numerico di soluzione delle equazioni alle derivate parziali. Un numero di *Courant* sufficientemente basso assicura che il flusso sia rimasto sufficienti passi iterativi nella cella spaziale. E' opportuno inoltre sottolineare che rispetto al metodo VOF introdotto nel capitolo precedente, in *openFOAM* l'equazione è modificata con l'aggiunta di un termine di compressione artificiale dell'interfaccia

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{U}) + \nabla \cdot (\alpha(1 - \alpha)U_r) = 0, \quad (6.5)$$

dove U_r rappresenta il campo di velocità normale all'interfaccia ed è diverso da zero solo dove $0 < \alpha < 1$, in questo modo si evita la diffusione numerica della frazione di volume nel dominio. Con questa strategia la zona di transizione viene mantenuta il più possibile fedele alla condizione fisica reale.

Per completare le impostazioni è necessario specializzare i file inclusi nelle directory *constant*, *system* e *0*. Nella prima definiremo il modello di turbolenza mediante le istruzioni riportate di seguito.

```

1 simulationType RASModel; (NEL FILE turbulenceProperties )
3
RASModel          kOmegaSST; (NEL FILE RASProperties )

```

In seguito è necessario definire le proprietà di trasporto nel file *transportProperties*.

```

1 twoPhase
  {
3     transportModel twoPhase;
     phase1          phase1;
5     phase2          phase2;
  }
7

```

```

phase1
9 {
    transportModel Newtonian;
11    nu          nu [ 0 2 -1 0 0 0 0 ] 1e-06;
    rho          rho [ 1 -3 0 0 0 0 0 ] 1000;
13    CrossPowerLawCoeffs
    {
15        nu0          nu0 [ 0 2 -1 0 0 0 0 ] 1e-06;
        nuInf        nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
17        m          m [ 0 0 1 0 0 0 0 ] 1;
        n          n [ 0 0 0 0 0 0 0 ] 0;
19    }

21    BirdCarreauCoeffs
    {
23        nu0          nu0 [ 0 2 -1 0 0 0 0 ] 0.0142515;
        nuInf        nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
25        k          k [ 0 0 1 0 0 0 0 ] 99.6;
        n          n [ 0 0 0 0 0 0 0 ] 0.1003;
27    }
}
29
phase2
31 {
    transportModel Newtonian;
33    nu          nu [ 0 2 -1 0 0 0 0 ] 1.48e-05;
    rho          rho [ 1 -3 0 0 0 0 0 ] 1;
35    CrossPowerLawCoeffs
    {
37        nu0          nu0 [ 0 2 -1 0 0 0 0 ] 1e-06;
        nuInf        nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
39        m          m [ 0 0 1 0 0 0 0 ] 1;
        n          n [ 0 0 0 0 0 0 0 ] 0;
41    }

43    BirdCarreauCoeffs
    {
45        nu0          nu0 [ 0 2 -1 0 0 0 0 ] 0.0142515;
        nuInf        nuInf [ 0 2 -1 0 0 0 0 ] 1e-06;
47        k          k [ 0 0 1 0 0 0 0 ] 99.6;
        n          n [ 0 0 0 0 0 0 0 ] 0.1003;
49    }
}
51 sigma          sigma [ 1 0 -2 0 0 0 0 ] 0.07;

```

Per ogni fase si definisce il modello, che nel caso in esame è *newtoniano*, e il valore delle proprietà fisiche che lo caratterizzano. Tra le due fasi inoltre bisogna specificare la tensione superficiale *sigma*. Nella directory *system* si stabiliscono i parametri della simulazione. Nel file *fvSchemes* vengono associati agli operatori differenziali gli schemi numerici utilizzati per approssimarli.

```

1 ddtSchemes
{

```

```

3     default      Euler;
    }
5 gradSchemes
  {
7     default      Gauss linear;
    }
9 divSchemes
  {
11    div(rho*phi,U)  Gauss linear;
    div(phi,alpha)  Gauss vanLeer;
13    div(phi*rb,alpha) Gauss interfaceCompression;
    div(phi,k)      Gauss upwind;
15    div(phi,omega) Gauss upwind;
    div(phi,R)      Gauss upwind;
17    div(R)          Gauss linear;
    div(phi,nuTilda) Gauss upwind;
19  }
  laplacianSchemes
21  {
    default      Gauss linear corrected;
23  }
  interpolationSchemes
25  {
    default      linear;
27  }
  snGradSchemes
29  {
    default      corrected;
31  }
  fluxRequired
33  {
    default      no;
35    p_rgh;
    pcorr;
37    alpha;
  }

```

In *controlDict* si stabiliscono il solver, il passo temporale, il numero di *Courant* e la durata della simulazione.

```

application      interFoam;
2 startFrom      latestTime;
  startTime      0;
4 stopAt         endTime;
  endTime        100;
6 deltaT         2e-06;
  adjustTimeStep off;
8 maxCo          0.2;
  maxAlphaCo     0.2;
10 maxDeltaT     1;

```

In *fvSolution* si stabiliscono gli algoritmi di soluzione delle equazioni discretizzate .

```

solvers

```

```
2 {
3     pcorr
4     {
5         solver          PCG;
6         preconditioner  DIC;
7         tolerance       1e-10;
8         relTol          0;
9     }
10
11     p_rgh
12     {
13         solver          PCG;
14         preconditioner  DIC;
15         tolerance       1e-07;
16         relTol          0.05;
17     }
18
19     p_rghFinal
20     {
21         solver          PCG;
22         preconditioner  DIC;
23         tolerance       1e-07;
24         relTol          0;
25     }
26
27     "(U|k|omega)"
28     {
29         solver          PBiCG;
30         preconditioner  DILU;
31         tolerance       1e-06;
32         relTol          0;
33     }
34
35     "(U|k|omega)Final"
36     {
37         solver          PBiCG;
38         preconditioner  DILU;
39         tolerance       1e-08;
40         relTol          0;
41     }
42 }
43
44 PIMPLE
45 {
46     momentumPredictor yes;
47     nCorrectors          3;
48     nNonOrthogonalCorrectors 0;
49     nAlphaCorr           1;
50     nAlphaSubCycles      4;
51     cAlpha                0.8;
52 }
```

In questo caso sono stati provati vari schemi e nel caso in esame quello più stabile si è rivelato l'algoritmo *Preconditioned (bi-)conjugate gradient* (PCG/PBiCG). Nel file *setFieldsDict* è possibile stabilire la porzione di geometria in cui la fase diversa dall'aria assume un valore diverso da 1 all'istante iniziale.

```

defaultFieldValues
2 (
    volScalarFieldValue alpha1 0
4 );

6 regions
  (
8   boxToCell
    {
10    box (-0.30 -0.30 -0.050) (0.30 0.30 0.100);
        fieldValues
12    (
            volScalarFieldValue alpha1 1
14    );
    }
16 );

```

Per concludere è possibile anche parallelizzare il calcolo modificando i parametri del file *decomposeParDict*.

```

numberOfSubdomains 3;
2 method          simple;
  simpleCoeffs
4 {
    n              ( 3 1 1 );
6   delta          0.001;
  }
8 hierarchicalCoeffs
  {
10    n              ( 1 1 1 );
        delta          0.001;
12    order          xyz;
  }
14 manualCoeffs
  {
16    dataFile       "";
  }
18 distributed     no;
  roots            ( );

```

Nella directory *0* infine si definiranno le condizioni al contorno relative alla velocità (file *U*) e alla fase liquida (file *alpha1*).

6.3 Caso multifase

Per i calcoli effettuati utilizzando il modello multifase (due fasi liquide e l'aria), è stato usato sia lo stesso modello geometrico che la stessa mesh.

6.3.1 Modello in OpenFOAM

Al fine di rendere le simulazioni con due o più fasi confrontabili, i due gruppi di simulazioni sono stati calcolati con lo stesso solver *multiphaseInterFoam*. L'unica differenza nell'impostazione della simulazione, consiste nella definizione ed inizializzazione di una terza fase. La definizione della fase viene fatta sempre nel file *transportProperties* presente nella cartella *constant*. In tale file andranno aggiunte le proprietà della terza fase, insieme alla tensione superficiale relative alle altre due fasi. Di seguito viene riportato il file modificato per includere nella simulazione una terza fase, che in questo caso si tratta di olio, posto sopra all'acqua.

```
1 phases
2   (
3     water
4     {
5       transportModel Newtonian;
6       nu nu [ 0 2 -1 0 0 0 0 ] 1e-06;
7       rho rho [ 1 -3 0 0 0 0 0 ] 1000;
8     }
9
10    oil
11    {
12      transportModel Newtonian;
13      nu nu [ 0 2 -1 0 0 0 0 ] 1e-06;
14      rho rho [ 1 -3 0 0 0 0 0 ] 500;
15    }
16
17    air
18    {
19      transportModel Newtonian;
20      nu nu [ 0 2 -1 0 0 0 0 ] 1.48e-05;
21      rho rho [ 1 -3 0 0 0 0 0 ] 1;
22    }
23  );
24
25  refPhase      air;
26
27  sigmas
28  (
29    (air water) 0.07
30    (air oil) 0.07
31    (water oil) 0.07
32  )
33  );
```

Per quanto riguarda l'inizializzazione della terza fase, questa verrà fatta sempre con l'utility *setFields*, aggiungendo al file mostrato in precedenza una zona (ad esempio tramite il comando *boxToCell*) dove settare il valore di *alphaoil* pari a uno.

6.4 Analisi scalabilità sistema

Come prima cosa, è stata analizzata la scalabilità del sistema, vale a dire i miglioramenti dei tempi di calcolo parallelizzando la simulazione su un numero crescente di core. Tale analisi è stata fatta utilizzando due diverse mesh, una da 350k elementi ed una da 600k elementi. Sono state effettuate diverse simulazioni per un tempo fisico totale simulato di 2 secondi, in regime laminare e considerando due fasi, parallelizzando il calcolo su un diverso numero di core e suddividendo il dominio di calcolo secondo diversi schemi. Da un'analisi preliminare si è visto come, a parità di numero di sottodomini (e quindi di core usati per il calcolo), i tempi di calcolo variassero usando diversi schemi di suddivisione. Nelle due tabelle seguenti vengono riportati i tempi di calcolo, in secondi, ottenuti discretizzando il dominio in vari modi, sia per quanto riguarda il numero di sottodomini, sia per quanto riguarda lo schema di discretizzazione. Le due tabelle si riferiscono ai valori ottenuti con le mesh di 350k e 600k elementi.

# core	Sudd 1	Sudd 2	Sudd 3	Tempo 1 [s]	Tempo 2 [s]	Tempo 3 [s]
6	2x3x1	-	2x1x3	65233	-	61101
12	-	2x2x3	2x1x6	-	61471	55379
18	3x3x2	-	2x1x9	44007	-	36226
24	4x2x3	2x2x6	2x1x12	22969	25032	24516
30	2x3x5	-	2x1x15	20318	-	18434
36	3x3x4	2x2x9	2x1x18	23017	15062	14825
48	4x4x3	2x2x12	2x1x24	10344	7435	11266

# core	Sudd 1	Sudd 2	Sudd 3	Tempo 1 [s]	Tempo 2 [s]	Tempo 3 [s]
12	-	2x2x3	2x1x6	-	281600	274039
18	3x3x2	-	2x1x9	198008	-	175605
24	4x2x3	2x2x6	2x1x12	130280	122628	124798
30	2x3x5	-	2x1x15	97298	-	94747
36	3x3x4	2x2x9	2x1x18	108963	76402	76227
48	4x4x3	2x2x12	2x1x24	62464	53032	54292
60	-	2x2x15	2x1x30	-	41092	43188
72	-	2x2x18	2x1x36	-	34412	37606
84	-	2x2x21	2x1x42	-	31429	34623
96	4x4x6	2x2x24	2x1x48	38380	29591	33006

Dall'analisi di queste due tabelle, si vede come le discretizzazioni del tipo $2 \times 2 \times N/2$ o $2 \times 1 \times N$ (cioè prediligendo le suddivisioni lungo l'asse z , coincidente con l'asse dell'ugello di uscita) siano più convenienti al fine di velocizzare i tempi di calcolo. Infatti in questo modo vengono ottimizzati sia il numero di celle per sottodominio che il numero superfici in comune tra due o più sottodomini.

Per visualizzare l'incremento della velocità del calcolo dovuto all'utilizzo di un numero maggiore di sottodomini (e quindi di core), vengono riportati nei due grafici seguenti gli speed-up relativi alle simulazioni con discretizzazioni del tipo $2 \times 1 \times N$. Lo speed-up per la ennesima simulazione, viene calcolato dividendo il tempo impiegato dalla ennesima simulazione per quello impiegato dalla simulazione con il minor numero di core. In questo modo, se ad esempio il tempo di calcolo viene dimezzato, si otterrà un valore di speed-up pari a 2.

Dall'analisi dei grafici riportati, si vede come lo speed-up sia maggiore rispetto a quello teorico. Infatti, a parità di altre condizioni, raddoppiando il numero di core utilizzati, posso al massimo ottenere un dimezzamento del tempo di calcolo. Nel nostro caso invece, si ottiene un vantaggio maggiore rispetto a quello teorico: molto probabilmente ciò è dovuto ad effetti secondari dovuti al numero di facce in comune tra i processori. Nel grafico relativo alla mesh di 600k elementi, si vede come la curva dello speed-up si allontani dalla linearità: ciò è dovuto al raggiungimento dei benefici massimi ottenibili dalla parallelizzazione con questo numero di elementi. Infatti, suddividendo il dominio di calcolo in un numero sempre maggiore di sottodomini, il numero di elementi per core sarebbe troppo basso per ottenere qualsiasi giovamento dall'uso di maggiori capacità di calcolo. Dall'analisi di questa curva, si può risalire quindi al numero ottimale di core per una determinata discretizzazione spaziale del problema, al fine di ottimizzare le risorse di calcolo utilizzate ed i tempi di calcolo.

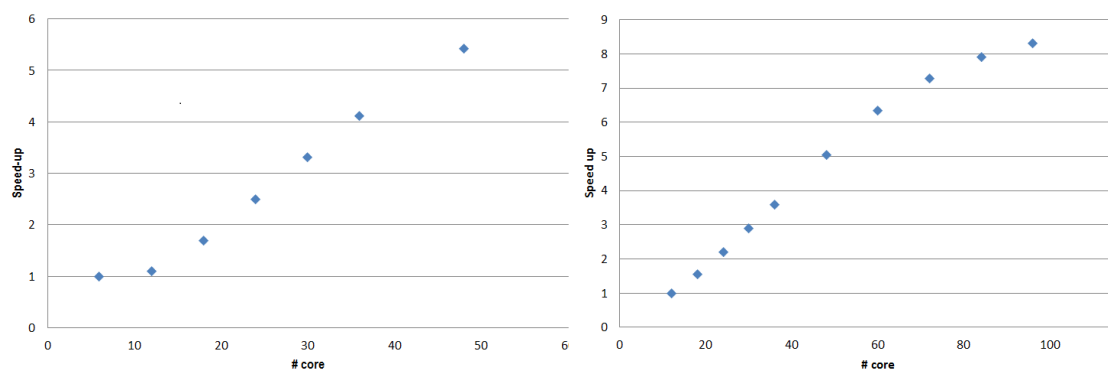


FIGURA 6.5: Speed-up mesh da 350k elementi e 600k elementi.

6.5 Risultati numerici – Drain Sink

Nel seguente capitolo verranno presentati i risultati, numerici e sperimentali, ottenuti con i modelli ad acqua con il fluido in quiete come condizione iniziale.

6.5.1 Altezza critica in funzione dell'altezza della colonna di fluido

La prima analisi eseguita è quella relativa all'altezza critica in funzione del livello iniziale di acqua nel modello. Tutti i modelli teorici presenti in letteratura riportano come l'altezza critica non dipenda dal livello del liquido presente nel recipiente, ma solamente dal diametro del foro di uscita. Come modello di riferimento è stato preso il modello di Koyola [31], la cui equazione è

$$\frac{4}{3} \left[\left(H + \frac{2\sigma}{\rho g r} \right)^{\frac{3}{2}} - \left(\frac{2\sigma}{\rho g r} \right)^{\frac{3}{2}} \right] = r \sqrt{H + l}, \quad (6.6)$$

con r raggio dell'orifizio di uscita, l lunghezza del tuffante e σ tensione superficiale.

Anche il modello di Lubin Springer [34] è stato considerato come controllo.

Al fine di verificare questa teoria, sono state fatte delle prove sia numeriche che sperimentali, con livello iniziale variabile.

I test sperimentali sono stati effettuati con altezza iniziale di 30 mm, 50 mm e 90 mm calcolati dal livello del foro al pelo libero. Sono stati presi tre valori molto diversi al fine di massimizzare eventuali differenze dovute al livello iniziale. Il livello all'interno del recipiente è stato monitorato utilizzando la cella di carico posta al di sotto del recipiente stesso. In questo modo, dopo un'accurata calibrazione della cella, è possibile correlare il segnale con il livello del recipiente. Nella figura 6.6 vengono mostrati i risultati ottenuti per i tre diversi casi. Si vede come la quota alla quale avviene il drain sink sia sempre la stessa, indipendentemente dal livello iniziale, così come ci si aspettava. Per quanto riguarda i risultati numerici, l'istante in cui avviene il drain sink viene univocamente identificato dall'analisi nel tempo delle sezioni del dominio. Nella figura 6.7 viene illustrato un esempio di sezione durante il drain sink.

Per quanto riguarda gli esperimenti numerici, i livelli iniziali di acqua considerati sono stati: 20 mm, 30 mm e 40 mm. Nella tabella seguente vengono illustrati i valori del drain sink per i tre diversi livelli iniziali calcolati con OpenFOAM.

Livello iniziale [mm]	H_{cr} [mm]
20	7.3
30	7.2
40	7.2

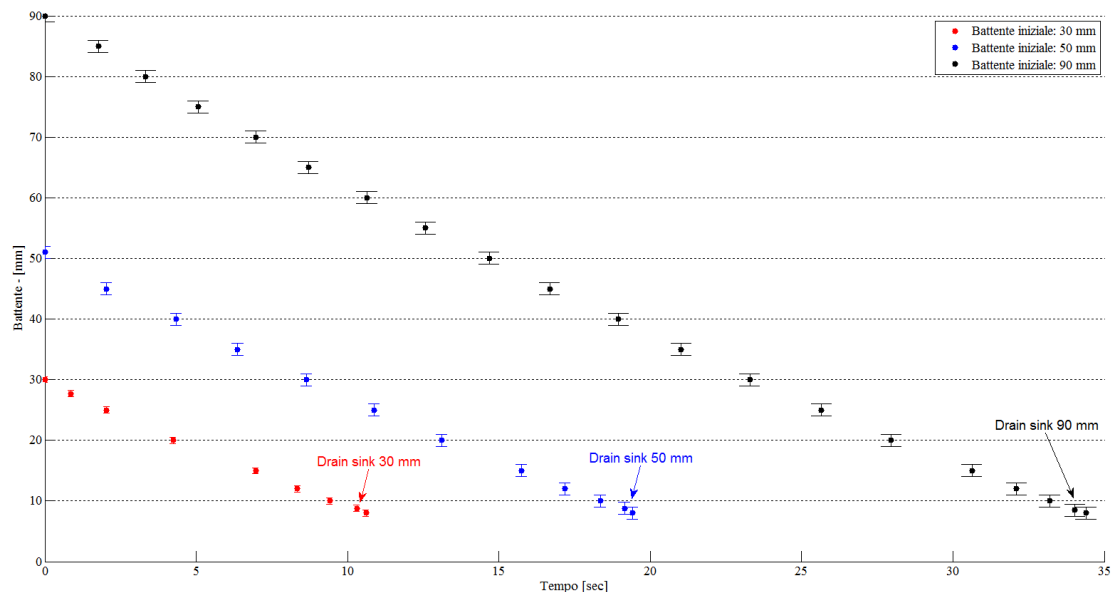


FIGURA 6.6: Altezza del battente in funzione del tempo con livello iniziale diverso.

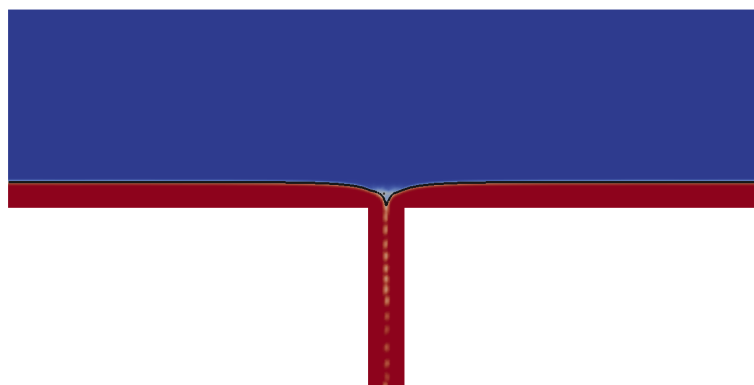


FIGURA 6.7: Istante immediatamente precedente al drain sink in OpenFOAM.

Come si può vedere dai valori della tabella, l'altezza critica non cambia al variare del livello iniziale dell'acqua, così come predetto dal modello di Koyola e così come verificato dalle prove sperimentali. La differenza di 0.1 mm tra la prova a 20 mm e quelle a 30 e 40 mm, è dovuta solamente ad errori di tipo numerico.

6.5.2 Altezza critica e tempi di svuotamento in funzione del diametro dell'ugello

La seconda assunzione del modello teorico è che l'altezza critica nel caso del drain sink dipenda dal raggio del foro di uscita. Per verificare questa ipotesi, sono state fatte alcune simulazioni numeriche con diametro di uscita variabile, in particolar modo sono stati fatti i calcoli con diametro D pari a 10 mm, 11 mm e 20 mm (configurazione A di figura 6.8). Sono inoltre state fatte anche delle simulazioni con l'ugello di uscita composto, come nel caso reale, da due

cilindri con diametro decrescente (configurazione B di figura 6.8). Il diametro maggiore è stato considerato di 11.5 mm, mentre per quello minore sono stati considerati due diversi valori, 10 e 9.5 mm.

Nella figura 6.9 si vede come i valori dell'altezza critica ottenuti con OpenFOAM sono in

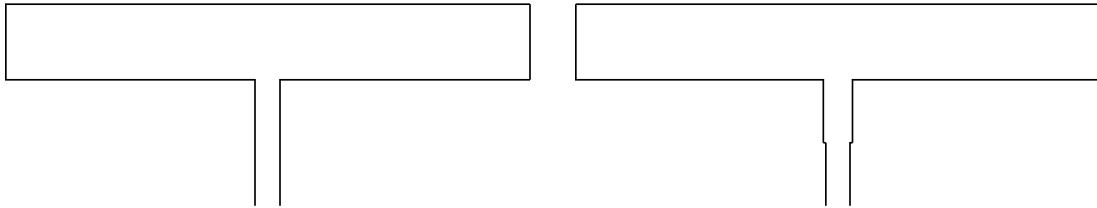


FIGURA 6.8: Configurazione (A) e (B) del modello geometrico.

perfetto accordo con quelli predetti dai modelli teorici. Per quanto riguarda il confronto tra risultati numerici e sperimentali, si rimanda alla prossima sezione. Si può vedere come le altezze critiche ottenute dai calcoli numerici siano in posizione intermedia rispetto a quella predetta dai due modelli considerati (Koyola e Lubin Springer).

Nella prossima tabella, vengono riportati i valori delle altezze critiche ottenute al variare del

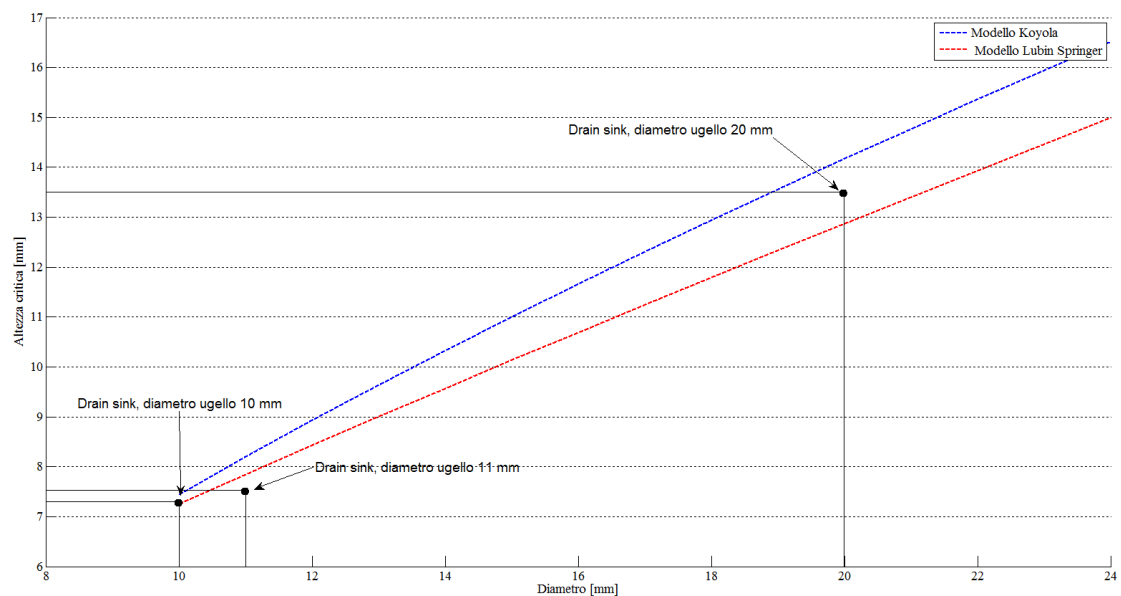


FIGURA 6.9: Confronto tra i modelli teorici ed i risultati fluidodinamici.

diametro dell'ugello di uscita, insieme ai tempi di svuotamento, definiti come il tempo in cui per la prima volta la fase gassosa entra all'interno dell'ugello. Vengono riportati sia i dati relativi alle simulazioni con ugello di 10, 11 e 20 mm, insieme alle configurazioni con l'ugello composto da due cilindri. Nella prima configurazione i due cilindri hanno diametro 11.5 e 10 mm andando dall'altro verso il basso, mentre nella seconda configurazione i due diametri misurano 11.5 e 9.5 mm rispettivamente.

Diametro ugello [mm]	H_{cr} [mm]	Tempo svuotamento [s]
10	7.2	12.2
11	7.5	9.8
20	13.5	2.3
11.5 / 10	7.8	10.1
11.5 / 9.5	7.1	12.0

Dai dati riportati nella tabella si possono fare alcune considerazioni.

Innanzitutto, per quanto riguarda le prime tre righe, si vede come all'aumentare del diametro diminuisca il tempo di svuotamento e aumenti l'altezza critica, così come già visto nel grafico precedente, dalle prove sperimentali e teorizzato da Koyola. Considerazioni analoghe possono essere fatte anche per le ultime due righe in quanto, al diminuire del diametro finale dell'ugello, diminuisce l'altezza critica, ed ovviamente aumenta il tempo di svuotamento.

Se però si confrontano i risultati riportati nella prima e nella quarta riga (ugello con diametro costante di 10 mm ed ugello con due diametri di 11.5 e 10 mm rispettivamente), si nota come, a fronte di un'altezza critica pressoché invariata, il tempo di svuotamento si riduca. Inoltre si vede come la presenza di una zona di imbocco prima della parte finale dell'ugello non sembra modificare l'altezza critica, come invece teorizzato da Koyola. Infatti nei due casi con il doppio diametro, pur avendo la zona di imbocco un diametro di 11.5 mm, l'altezza critica è più bassa rispetto alla simulazione con diametro di 11 mm.

Per concludere questa analisi, le correlazioni teoriche come quella proposta da Koyola possono aiutare ad individuare l'ordine di grandezza dell'altezza critica ma, soprattutto nel caso di geometrie più complesse di un semplice ugello cilindrico, non sono sufficienti ad individuare il valore dell'altezza critica stessa.

Le simulazioni fluidodinamiche, e le prove sperimentali, si rivelano quindi fondamentali per lo studio di queste relazioni.

6.5.3 Confronto numerico – sperimentale

Al fine di validare i modelli numerici utilizzati per la simulazione dello svuotamento di un recipiente, vanno confrontati i risultati ottenuti con OpenFOAM con quelli ottenuti, partendo dalle stesse condizioni iniziali, con le prove sperimentali. È infatti buona norma validare i modelli numerici su prove sperimentali facilmente replicabili con geometrie semplificate, in modo da acquisire sensibilità nell'analisi delle simulazioni, dei risultati e delle criticità che saranno utili quando si andranno ad analizzare configurazioni non replicabili in laboratorio. Il caso della sfera piena di acciaio liquido è un caso tipico dove non si possono fare prove sperimentali per evidenti motivi legati alla presenza dell'acciaio fuso e dove quindi si avranno a disposizione

solamente i dati relativi alle simulazioni numeriche. L'apparato sperimentale è descritto nel capitolo 7

In questo capitolo verranno confrontati i risultati ottenuti con l'apparato sperimentale e con le simulazioni fluidodinamiche i risultati relativi alla prova con altezza iniziale del liquido di 30 mm. Per quanto riguarda la parte numerica, è stato considerato il risultato ottenuto con la geometria dell'ugello composta da due cilindri di diametro 11.5 e 10 mm, in quanto la più simile alla reale geometria presente nell'apparato sperimentale.

Per il confronto dei risultati, sono stati utilizzati più parametri. Per la prova sperimentale sono stati raccolti il segnale proveniente dalla cella di carico, proporzionale al peso dell'acqua all'interno del recipiente, e quindi al suo livello medio, ed il livello dell'acqua analizzato dai frame provenienti dalla ripresa fatta con la telecamera ad alta definizione. Per quanto riguarda la simulazione numerica invece, sono stati raccolti i dati relativi alla frazione di volume (è stato preso come riferimento il valore $\alpha = 0.5$ per evitare errori dovuti a fluttuazioni numeriche), su una linea verticale posizionata esattamente al centro dell'ugello di uscita, in modo da conoscere con esattezza la posizione del drain sink.

Dall'analisi del grafico riportato in figura 6.10 si può notare come, innanzi tutto i dati ottenuti

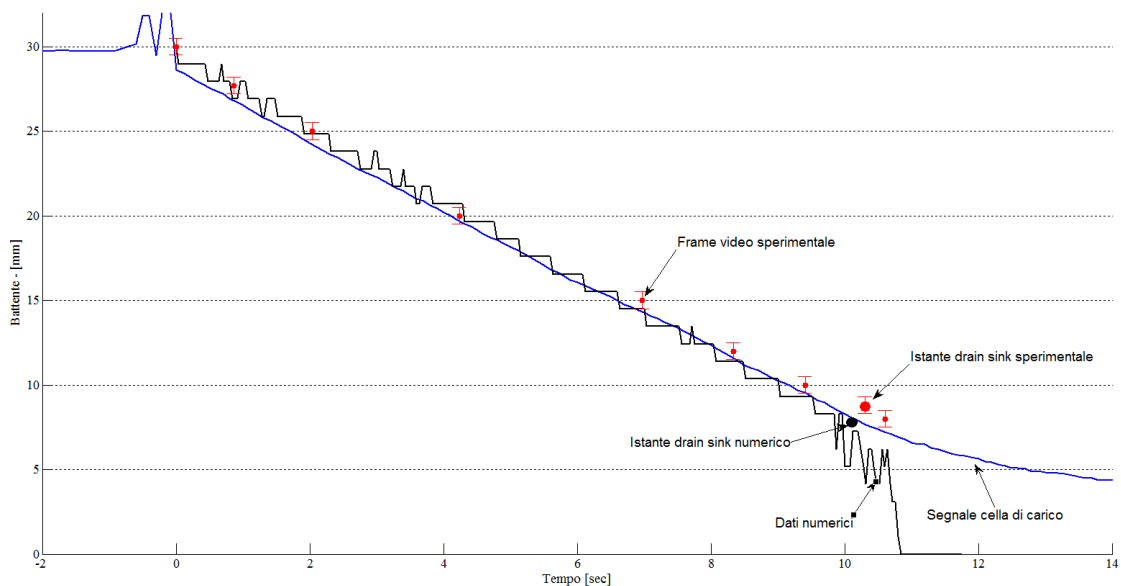


FIGURA 6.10: Confronto tra risultati sperimentali e numerici con ugello di diametro 11.5/10 mm.

con la cella di carico e dall'analisi dei frame del video coincidono, a riprova della corretta calibrazione della cella.

Confrontando invece i risultati sperimentali con quelli numerici, si vede come la simulazione con OpenFOAM riesca a predire con buona precisione sia l'altezza critica che l'istante in cui il drain sink entra all'interno dell'ugello.

Se all'interno dello stesso grafico mettiamo anche i dati relativi alle simulazioni effettuate con geometria dell'ugello costante, di diametro 10, 11 e 20 mm, vediamo come le differenze nel

tipo di geometria e nella misura dei diametri comportino diverse altezze critiche e diversi tempi di svuotamento, così come già spiegato nella sezione precedente. Il calcolo con l'ugello con il

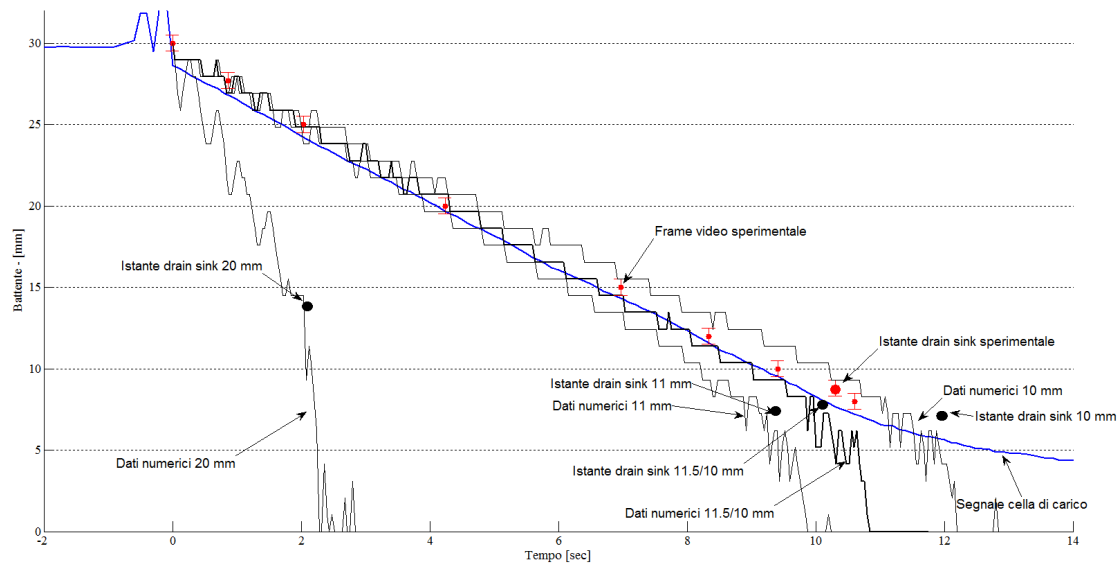


FIGURA 6.11: Confronto tra risultati sperimentali e numerici con ugello di vari diametri.

doppio diametro da 11.5 e 9.5 mm si è rivelato in grado di prevedere adeguatamente il fenomeno del drain sink in questa configurazione. Ulteriori ottimizzazioni della geometria dell'ugello chiaramente miglioreranno il livello di accuratezza.

6.5.4 Analisi profilo superficie libera

Un'ulteriore analisi eseguita, riguarda il profilo della superficie libera predetto da OpenFOAM. Così come ci si aspettava dalle prove sperimentali, anche nelle simulazioni numeriche la forma della superficie libera rimane orizzontale per tutta la durata dello svuotamento, ad eccezione della depressione causata dal drain sink negli istanti finali. Nella figura 6.12 viene riportata la forma della superficie libera al variare del tempo. Per la definizione di superficie libera è stato preso il contour dei punti del dominio per il valore di α pari a 0.5. Si vede come la superficie rimanga sempre orizzontale ad eccezione degli istanti finali e della sola zona centrale. Negli ultimi contour, i punti sono posizionati in modo più caotico in quanto, essendo nei momenti finali dello svuotamento, la frazione di volume è ormai collassata e la fase gassosa ha cominciato ad entrare all'interno dell'ugello.

6.5.5 Analisi campo di moto superficiale

Oltre all'analisi della forma della superficie libera, è stato analizzato anche il campo di moto superficiale, in quanto ha un ruolo decisivo per la previsione del comportamento della superficie

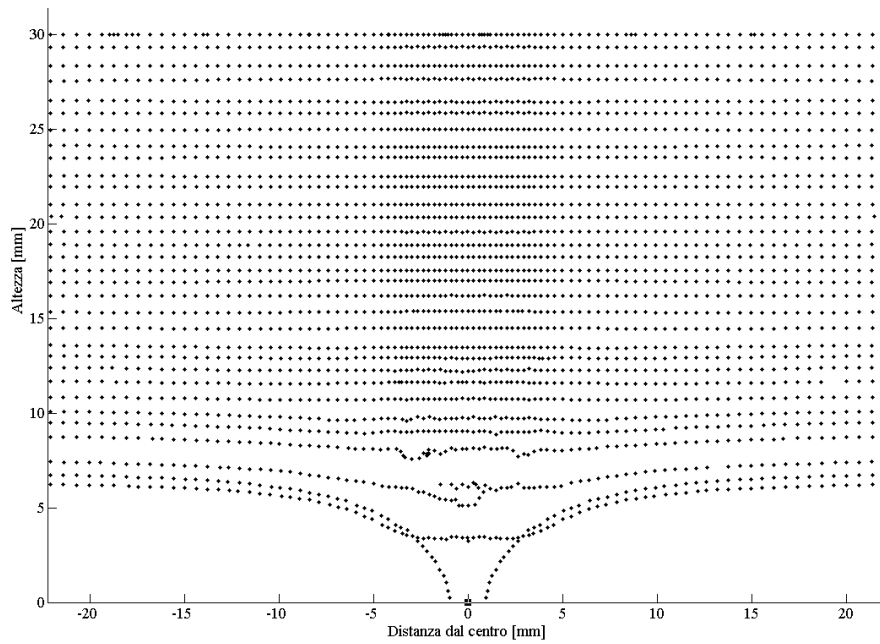


FIGURA 6.12: Andamento nel tempo della superficie libera.

libera stessa. Tutti i modelli teorici indicano come da un campo di moto uniformemente verticale per tutta la superficie si passi ad un campo di moto radiale negli ultimi istanti dello svuotamento. Nella figura 6.13 vengono rappresentati i vettori velocità disegnati sulla superficie. Si vede come il loro andamento rispetti quello teorico, passando da una distribuzione completamente verticale ad una sempre più radiale. Anche in questo caso sono presenti delle singolarità nella zona centrale negli ultimi istanti, dovuti al fatto che anche l'aria, venendo risucchiata verso l'ugello, va a modificare la forma e quindi il moto della superficie libera. Anche per quanto riguarda

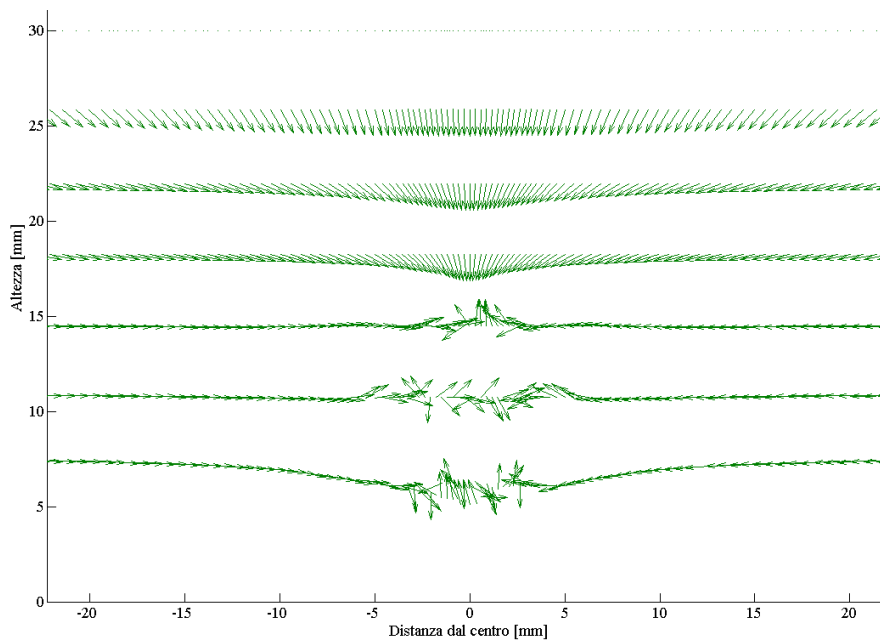


FIGURA 6.13: Andamento della velocità nella superficie libera.

questa analisi, OpenFOAM ha fornito risultati affidabili e coerenti con la teoria.

6.5.6 Analisi del numero di Reynolds

Tutti i risultati presentati sinora sono stati ottenuti con delle simulazioni di tipo laminare. Non è stato utilizzato alcun modello per il trattamento della turbolenza in quanto, dalle prime stime teoriche, il calcolo non lo richiedeva. Il fatto di non utilizzare modelli di turbolenza permette di velocizzare il calcolo di una quantità che va dal 30 al 50%, valore stimato sulla base di alcune prove effettuate.

Calcolo teorico del numero di Reynolds Al fine di verificare la correttezza dell'assunzione di calcolo laminare per le simulazioni in OpenFOAM nel caso del Drain Sink, è stato innanzi tutto definito un numero di Reynolds limite. Si ricorda che il numero di Reynolds è definito come:

$$Re = \frac{\rho v D}{\mu} \quad (6.7)$$

dove ρ è la densità del fluido, v è la sua velocità, D è una lunghezza caratteristica e μ è la viscosità dinamica.

Trattandosi di un moto all'interno di condotti cilindrici (il recipiente oppure l'ugello), i numeri di Reynolds limite possono essere così riassunti:

- regime laminare: $Re < 2300$;
- regime di transizione: $2300 < Re < 10000$;
- regime turbolento: $Re > 10000$.

Poiché il fluido è acqua i valori di densità e di viscosità dinamica saranno rispettivamente 1000 kg/m^3 e $1e-3 \text{ Pa} \cdot \text{s}$. La scelta della grandezza caratteristica della geometria dipende da alcune considerazioni. Innanzi tutto va detto che nella zona del recipiente indisturbata dalla presenza dell'ugello, come grandezza caratteristica di riferimento è stato scelto il diametro del recipiente stesso. All'interno dell'ugello invece, trattandosi di un flusso in un condotto chiuso, la grandezza caratteristica sarà il diametro dell'ugello (o il suo diametro medio, che comunque non si discosta dal valore nominale). La zona più critica per quanto riguarda la definizione del numero di Reynolds è la zona del recipiente immediatamente adiacente all'imbocco dell'ugello. Infatti, pur essendo contenuta all'interno del recipiente con diametro maggiore, questa zona di liquido viene influenzata solamente dalla dimensione del diametro dell'ugello. Per questo motivo anche in questa zona è stata presa come grandezza caratteristica il diametro dell'ugello.

Come velocità di prima approssimazione, è stata presa quella relativa alla caduta libera per gravità:

$$v = \sqrt{2 \cdot 9.81 \cdot H} \quad (6.8)$$

dove H è il salto geodetico che fa il fluido. In questo caso, come H totale è stata considerato la distanza tra il livello iniziale dell'acqua e l'imbocco dell'ugello (30 mm). Inserendo questi dati si ottiene una velocità di 0.75 m/s, a cui corrisponde, all'interno di un ugello di diametro 10 mm, un numero di Reynolds pari a 7500.

Il numero così calcolato rientra all'interno del regime di transizione, ma va ricordato che la velocità all'interno dell'ugello sarà molto minore a causa degli attriti e delle perdite di carico all'imbocco dell'ugello stesso. La velocità così calcolata, inoltre, si riferisce all'istante iniziale, quando le velocità di uscita sono maggiori, per poi diminuire con l'abbassarsi del livello di liquido all'interno del recipiente. La supposizione di regime laminare può quindi considerarsi valida all'interno dell'ugello.

Per quanto riguarda invece il numero di Reynolds del contenitore ($D = 0.210$ mm), si possono solamente stimare le velocità che interessano il fluido in questa zona. Innanzi tutto, dagli esperimenti di laboratorio, si è visto che il fluido si abbassa di 25 mm in circa 10 secondi, con una velocità che quindi è pari a 0.0025 m/s. Per quanto riguarda la componente radiale, nelle zone più lontane dall'ugello questa sarà dello stesso ordine di grandezza. Il numero di Reynolds all'interno del recipiente sarà quindi di circa 500 – 1000, ampiamente sotto la condizione di regime laminare.

I calcoli teorici danno quindi ragione del fatto che la supposizione di regime laminare sia corretta.

Calcolo del numero di Reynolds da OpenFOAM Per assicurarsi della bontà delle considerazioni iniziali, è stato analizzato il numero di Reynolds locale prendendo, in ogni punto del dominio, il valore del vettore velocità e mettendolo all'interno della formula per il calcolo del numero di Reynolds.

Dall'analisi del grafico riportato in figura 6.14, che riporta il valore del numero di Reynolds su una linea orizzontale a 5 mm dal fondo in funzione del tempo, si nota come nella zona più vicina all'ugello il numero di Reynolds raggiunga al massimo il valore di 2600 negli istanti iniziali. Successivamente il numero di Reynolds raggiunge valori inferiori ai 2000, in particolare nella zona interessata dal drain sink. Tali valori sono perfettamente compatibili con l'ipotesi di moto laminare già verificata dai calcoli teorici. Anche dall'analisi dei valori ottenuti con OpenFOAM, si può quindi validare l'ipotesi iniziale di fenomeno laminare.

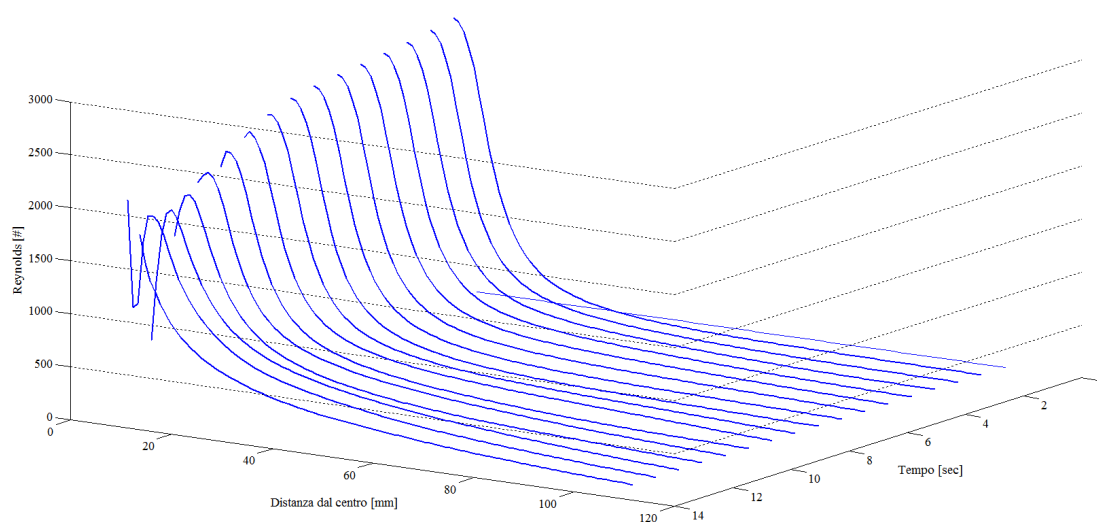


FIGURA 6.14: Andamento numero di Reynolds su una linea orizzontale a 5 mm dal fondo in funzione del tempo.

Capitolo 7

Apparato sperimentale con acqua

7.1 Descrizione dell'apparato sperimentale

Il dispositivo di sperimentazione (figura 7.1) comprende un contenitore cilindrico in plexiglass del diametro di 21 cm e di altezza 11 cm. Sul fondo sono praticati due fori con relativi ugelli di diametro 14 mm, uno al centro e uno distante 88 mm. Esso poggia su una struttura in alluminio con quattro punti di appoggio di cui uno è realizzato con una cella di carico. Alla struttura inoltre è fissata un'asta libera di ruotare sul piano sagittale; ad una estremità sono appese delle masse mentre all'altra è agganciato un tronco di tubo conico che spinto dai pesi si incastra nell'ugello del contenitore in plexiglass. Sull'asta sono fissati gli accelerometri. L'intera struttura poggia su una base in legno a cui sono fissati dei supporti antivibranti. Infine un contenitore in plastica raccoglie l'acqua di scarico. L'apparato sperimentale è realizzato in modo da poter variare alcuni parametri fondamentali:

1. la posizione di scarico dell'ugello;
2. il materiale del tubo conico;
3. il materiale dell'asta;
4. la posizione degli accelerometri.

Lo scopo della cella di carico è sia di permettere l'identificazione temporale della completa uscita dell'acqua sia di misurare il tempo di svuotamento.

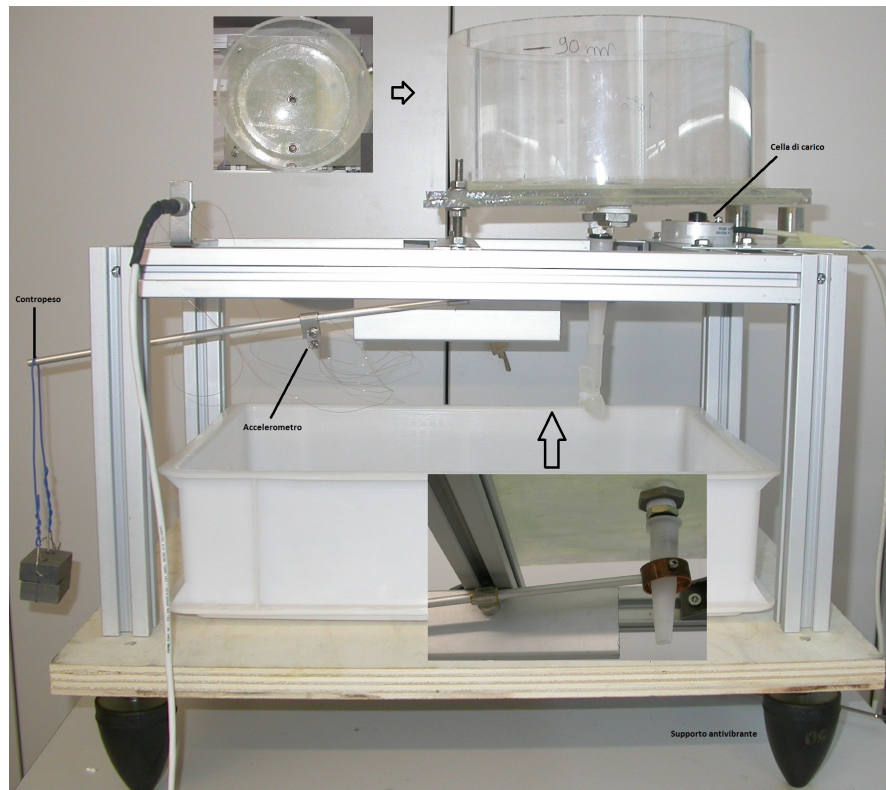


FIGURA 7.1: Apparato sperimentale.

7.1.1 Descrizione dei sensori

Per l'acquisizione dell'accelerazione ho utilizzato il sensore ADXL203 della Analog Device di cui riporto le caratteristiche tecniche:

- Numero di canali: 2.
- Sensibilità: 1000 mV/g.
- Tensione a 0 g: 2.5 V.
- Frequenza di risonanza: 5.5 kHz.

Per l'acquisizione del peso ho utilizzato la cella di carico 8523-500 della Burst avente un intervallo di misura pari 0 ··· 500 [N].

7.1.2 Descrizione del sistema di acquisizione

Il sistema di acquisizione dati (figura 7.2) è composto da una scheda elettronica a cui sono connessi i segnali dei trasduttori. La scheda è dotata di tre canali di ingresso differenziale $\pm 5V$ con un filtro *antialiasing*. I segnali vengono campionati da un DAQ a 16 bit con frequenza

programmabile. I dati digitalizzati vengono inviati ad uno specifico indirizzo IP mediante protocollo UDP con frequenza di $10Hz$. I pacchetti dati sono composti da interi a 16 bit che corrispondono esattamente ai valori di quantizzazione.

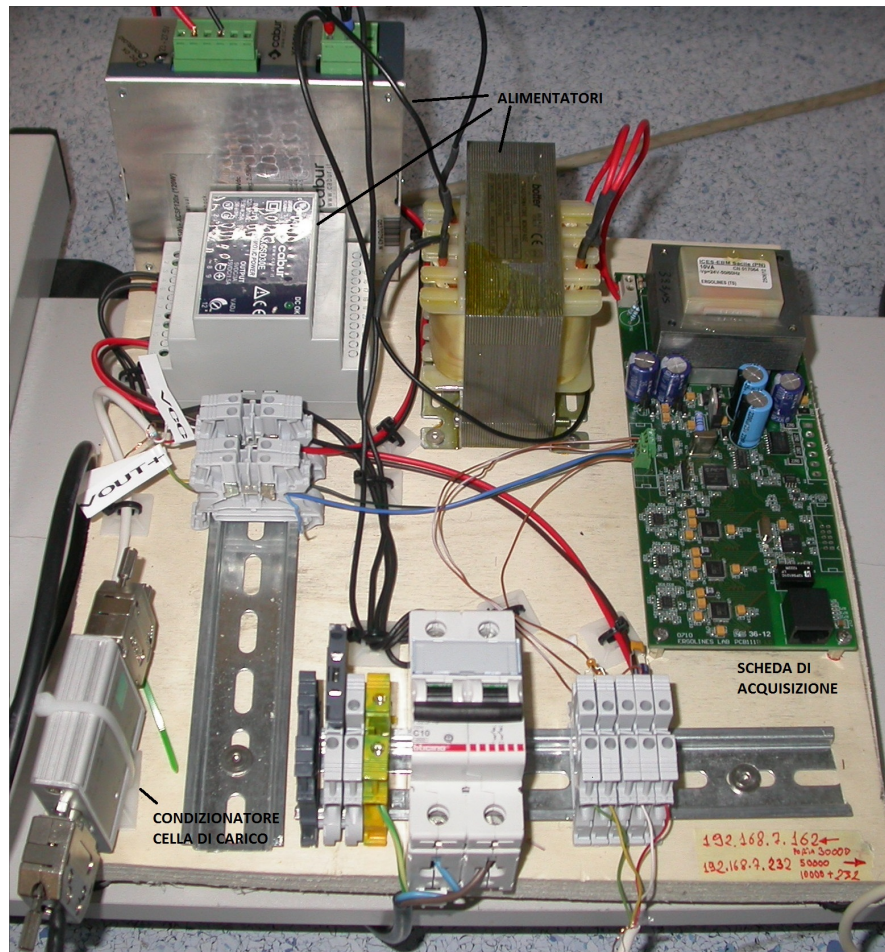


FIGURA 7.2: Sistema di acquisizione dati.

7.2 Descrizione degli esperimenti

Il criterio di scelta con cui sono stati organizzati gli esperimenti si basa sull'idea di riprodurre in scala le tipiche condizioni operative che si possono incontrare in impianto. Si è quindi ipotizzato un possibile numero di variabili che in principio in impianto possono condizionare l'esito della sperimentazione:

- posizione dell'accelerometro;
- presenza della componente tangenziale della velocità nel fluido contenuto nel cilindro di plexiglass;
- presenza della vasca sotto il tuffante.

Si potrebbe supporre che la posizione dell'accelerometro possa influenzare il segnale che rappresenta la dinamica del sistema. In ogni caso in impianto non è sempre possibile mettere l'accelerometro nella posizione desiderata pertanto è utile indagare questa eventualità. Una delle principali incognite nello svuotamento della siviera è proprio la presenza del vortice, come ampiamente descritto nel capitolo introduttivo non è noto con certezza se esso si sviluppa durante il processo. Infine sebbene in tutti gli impianti sia previsto che il tuffante entri nella paniera, è necessario verificare quale impatto ha il fluido scaricato sul tuffante stesso. E' lecito infatti attendersi che esso possa alterare la dinamica rispetto al caso con scarico in aria: in effetti si potrebbe supporre che il liquido esterno assorba parte dell'energia vibrazionale del tubo. Naturalmente il numero di variabili può rapidamente crescere, ad esempio il tipo di materiale con cui viene realizzata l'asta, la modalità di fissaggio del tubo e la posizione del foro di scarico rispetto all'asse di simmetria del contenitore sono alcuni dei parametri che potrebbero essere analizzati. Come evidenziato nel capitolo introduttivo lo scopo dell'attività di ricerca consiste principalmente nel valutare le potenzialità delle tecniche di analisi considerate nel caso specifico del passaggio scoria. E' chiaro che un'analisi di tutti i parametri porterebbe oltre gli obiettivi iniziali. In sintonia con questa impostazione si è definito il *setup* sperimentale secondo lo schema riportato in tabella 7.1. Gli esperimenti sono stati divisi quindi in gruppi come riportato nella seguente lista:

TIPO	DESCRIZIONE
S001A	Accelerometro in posizione a 21.5 cm dal fondo dell'asta. Asse X verticale. Asse Y lungo l'asta. Asta in alluminio lunga 41.5 con diametro esterno 5.80 mm e foro interno da 4 mm. Scarico libero con tuffante in plastica. Battente acqua 90 mm.
S001B	Accelerometro in posizione a 21.5 cm dal fondo dell'asta. Asse X verticale. Asse Y lungo l'asta. Asta in alluminio lunga 41.5 con diametro esterno 5.80 mm e foro interno da 4 mm. Scarico in vasca con tuffante in plastica. Battente acqua 90 mm.
S001C	Accelerometro in posizione a 4 cm dal fondo dell'asta. Asse X trasversale. Asse Y lungo l'asta. Asta in alluminio lunga 41.5 con diametro esterno 5.80 mm e foro interno da 4 mm. Scarico libero con tuffante in plastica. Battente acqua 90 mm.
S001D	Accelerometro in posizione a 4 cm dal fondo dell'asta. Asse X trasversale. Asse Y lungo l'asta. Asta in alluminio lunga 41.5 con diametro esterno 5.80 mm e foro interno da 4 mm. Scarico in vasca con tuffante in plastica. Battente acqua 90 mm.
S001E	Accelerometro in posizione a 4 cm dal fondo dell'asta. Asse X verticale. Asse Y lungo l'asta. Asta in alluminio lunga 41.5 con diametro esterno 5.80 mm e foro interno da 4 mm. Scarico libero con tuffante in plastica. Battente acqua 90 mm.

TABELLA 7.1: Schema *SETUP* esperimento.

- **GRUPPO 001, SETUP S001A**

1. 001A -Senza rotazione indotta con polistirolo in superficie.
 2. 002A -Senza rotazione indotta.
 3. 003A -Con rotazione indotta.
 4. 004A -Con rotazione indotta e polistirolo.
- **GRUPPO 002, SETUP S001A**
 1. 005A -Con rotazione indotta.
 2. 006A -Con rotazione indotta.
 3. 007A -Con rotazione indotta.
 4. 008A -Senza rotazione indotta.
 5. 009A -Senza rotazione indotta.
 6. 010A -Senza rotazione indotta.
 - **GRUPPO 002, SETUP S001B**
 1. 011A -Senza rotazione indotta.
 2. 012A -Con modica rotazione indotta.
 3. 013A -Con elevata rotazione indotta.
 - **GRUPPO 003, SETUP S001C**
 1. 014A -Senza rotazione indotta e scarico libero.
 2. 015A -Con rotazione e scarico libero.
 - **GRUPPO 003, SETUP S001D**
 1. 016A -Senza rotazione indotta e scarico in vasca.
 2. 017A -Con rotazione e scarico e scarico in vasca.
 3. 018A -Con rotazione e scarico e scarico in vasca.
 - **GRUPPO 004, SETUP S001E**
 1. 019A -Senza rotazione indotta e scarico libero.
 2. 020A -Con rotazione e scarico e scarico libero.
 3. 021A -Con rotazione e scarico e scarico libero.
 4. 022A -Senza rotazione e scarico e scarico libero.
 5. 023A -Con rotazione e scarico e scarico libero.

La suddivisione ripercorre il percorso originale con cui si è cercato di esplorare lo spazio dei parametri e per questo motivo si spiega l'apparente incongruenza tra i primi due gruppi. L'intera sequenza è stata ripetuta per tre volte, tuttavia essendo i risultati sostanzialmente simili, nel paragrafo seguente si riporta l'analisi di una singola sequenza.

7.3 Analisi dei risultati

Analogamente al capitolo relativo ai dati provenienti dagli impianti l'approccio segue uno schema prefissato. Prima vengono esaminate alcune caratteristiche globali del segnale come il valore *rms* e la media della densità spettrale di potenza. L'analisi prosegue quindi con l'analisi classica della potenza nello spazio del tempo e della frequenza (spetrogramma), con la trasformata discreta *Wavelet Packets* e con la trasformata di Hilbert. L'obiettivo principale di questa attività consiste nel valutare come queste tecniche permettano di discriminare i vari *setup* sperimentali. Essendo emerso che la posizione dell'accelerometro non costituisce una variabile chiave i risultati sono stati raggruppati in quattro gruppi:

- scarico libero senza vortici;
- scarico libero con vortici;
- scarico in vasca senza vortici;
- scarico in vasca con vortici;

I risultati dei modelli autoregressivi non sono riportati poiché come nel caso dei dati dall'impianto non offrono alcuna sostanziale informazione aggiuntiva.

7.3.1 Scarico libero senza vortici

Dai grafici del valore *rms* riportati nella figura 7.3 appare evidente che all'apertura del getto la potenza del segnale cresce per poi attenuarsi con il progressivo svuotamento. L'andamento tuttavia non è perfettamente ripetibile anche se lo schema comunque risulta chiaro. In questa modalità di svuotamento è prevista la formazione del fenomeno di *drain-sink* come discusso nel capitolo introduttivo. Effettivamente prima del completo svuotamento si ha un brusco aumento della potenza. Fa eccezione la figura 7.3e nella quale è presente una potenza maggiore ed un andamento meno conforme ai precedenti. Al termine di questa prima analisi sarà possibile interpretare meglio questo risultato. La densità di potenza media relativa al canale CH1 (figura 7.4) mostra una struttura a bande sì simile a quella del canale CH2 (figura 7.5). Su questo asse tuttavia la potenza si spinge a frequenze superiori. L'analisi degli spetrogrammi (figura 7.6) del canale CH1 rivelano chiaramente uno schema basato su bande concentrate principalmente a basse frequenze. I grafici mostrano come l'energia si attenua nel tempo anche se le frequenze maggiori hanno un comportamento più complesso. In effetti esse, in alcuni casi, si eccitano dopo un determinato tempo e non all'apertura del getto. L'analisi mostra inoltre come gli schemi siano sostanzialmente ripetibili. La posizione dell'accelerometro non appare essere una variabile il cui effetto è dominante. Le considerazioni espresse per il canale CH1 valgono inalterate

anche per il canale CH2 (figura 7.7). In questo caso tuttavia si noti come la potenza sia dilatata su un intervallo di frequenze più ampio. Notiamo infine come il fenomeno del *drain-sink* sia chiaramente identificabile in prossimità del periodo di svuotamento. Anche in questo caso si osserva come il caso descritto dalle figure 7.6e e 7.7e sia sostanzialmente diverso dagli altri. In effetti le bande a più alta frequenza pur eccitandosi in ritardo si estendono fino al termine. Negli altri casi è palese come si smorzano in anticipo. L'analisi *Wavelet packet* (figura 7.8 e 7.9) mette in luce in modo chiaro l'andamento della banda più bassa. I contributi di quelle superiori sono in effetti molto più bassi nel caso del canale CH1 (figura 7.8): infatti si nota che nel canale CH2 le bande superiori sono più energetiche. In entrambe i casi il fenomeno del *drain-sink* risulta perfettamente identificabile. Rispetto agli spettrogrammi si ha una minore risoluzione in frequenza ma si riesce ad apprezzare molto meglio il contenuto energetico relativo delle varie banda e soprattutto la dinamica del segnale risulta molto più definita. Anche con questa metodologia il comportamento del test 10, riportato nelle figure 7.8e e 7.9e è chiaramente in minor accordo con lo schema identificabile dagli altri test. E' tuttavia innegabile che anche in questo caso la tecnica wavelet permette di isolare con chiarezza l'evento *drain-sink* rispetto al calcolo del valore *rms* globale e allo spettrogramma. L'analisi mediante la trasformata di Hilbert rivela un andamento molto interessante. Le figure relative al canale CH1 7.10 riportano l'ampiezza istantanea e la frequenza istantanea. La prima in effetti risulta perfettamente congruente con l'andamento del valore *rms*. Per la seconda invece risulta che all'apertura del getto avviene un brusco salto passando da circa 1000 Hz a circa 200 Hz. Successivamente all'esaurirsi della potenza il valore risale verso il punto iniziale. In prossimità del *drain-sink* risulta evidente la brusca variazione verso il basso. Notiamo infine come il test relativo al grafico 7.10e presenti chiaramente un andamento diverso. I grafici relativi al canale CH2 (figura 7.11) rivelano uno schema simile anche se la risalita avviene più rapidamente. In effetti le figure 7.11c ed 7.11f sono più confuse. Anche in questo caso tuttavia l'evento *drain-sink* è identificabile. Tutti i metodi impiegati risultano utili perché permettono di apprezzare aspetti differenti del segnale registrato.

7.3.2 Scarico libero con vortici

In questa modalità l'analisi dei grafici del valore efficace (figura 7.12) mostra immediatamente come l'intensità della potenza sia notevolmente maggiore. Sicuramente la caratteristica più evidente è l'aumento dell'energia dopo un determinato intervallo dall'apertura del getto. In effetti questo evento coincide con la formazione delle varie fasi del vortice, come descritto nel capitolo introduttivo. Il canale CH1 risulta generalmente più energetico e con maggiore variabilità del valore. In questa serie di test non è stata controllata la velocità iniziale pertanto è lecito attendersi una certa variabilità degli schemi. Anche in questo caso la posizione dell'accelerometro non sembra essere una variabile significativa. Notiamo infine come in prossimità della fase finale la potenza decresca senza mostrare alcun fenomeno simile al *drain-sink*. Le figure 7.15 e

7.14 relative alla densità di potenza media indicano una struttura a bande che però risulta più ricca rispetto al caso senza vortici. Anche in questo caso il canale CH2 contiene un contributo maggiore alle frequenze più alte. Analogamente al caso precedente le immagini 7.15 e 7.16 che riportano lo spettrogramma rivelano l'usuale struttura. Anche in questo caso l'eccitazione delle bande superiori risulta spesso in ritardo rispetto all'apertura del getto. Inoltre diversamente dal caso senza vortici, esse temporalmente si estinguono come le bande inferiori. Nessun segnale impulsivo compare in prossimità dello svuotamento finale, a conferma che fenomeni tipo *drain-sink* non sono presenti. L'analisi *Wavelet Packets* relativo al CH1 (figure 7.17) indica come il fenomeno sia quantitativamente descritto dalle basse frequenze. Risulta interessante notare invece come le bande superiori in prossimità dello svuotamento abbiano un comportamento leggermente diverso, infatti prima di diminuire il livello di potenza presentano un piccolo incremento. Le figure 7.18 invece relative al canale CH2 mostrano invece come ci sia un contributo importante anche dalle bande superiori. Nel complesso comunque non emerge alcun evidente andamento tipo *drain-sink*. Infine si ribadisce come questo tipo di analisi permetta di apprezzare molto bene la dinamica delle varie bande pur mantenendo una risoluzione in frequenza chiaramente inferiore rispetto allo spettrogramma. La trasformata di Hilbert del canale CH1, riportata nei grafici 7.19 mostra un comportamento molto interessante. All'apertura del getto la frequenza istantanea cade bruscamente e rimane costante fino alla fine. In prossimità dello svuotamento finale il suo valore risale, non mostrando alcun evento catastrofico. L'unico grafico in disaccordo risulta essere quello riportato in figura 7.20b. Anche nelle figure 7.20 relative al canale CH2 si ripresenta uno schema simile. Risulta effettivamente molto evidente come la risalita della frequenza istantanea coincida con la fase di svuotamento finale. Infine sottolineiamo che anche in questo caso l'ampiezza istantanea coincide con l'andamento del valore efficace.

7.3.3 Scarico in vasca senza vortici

Cambiando la modalità di scarico e cioè inserendo una vasca sotto l'ugello, i grafici del valore efficace 7.21 cambiano in modo rilevante. Per prima cosa si nota che gli andamenti dei suoi assi sono molto simili. La potenza del segnale inoltre scende notevolmente rispetto al caso dello scarico libero. Sovrapposti al segnale compaiono molti eventi impulsivi. Infine l'evento *drain-sink* appare poco identificabile. I grafici della densità di potenza media 7.22 evidenziano una struttura a bande con una apparente estensione del canale CH2 verso le alte frequenze. In effetti le immagini 7.23 dello spettrogramma mostrano uno schema abbastanza simile ai casi precedenti. Infatti il canale CH2 (figure 7.23c e 7.23d) contiene un contributo in frequenza più esteso. I grafici però rivelano che non si ha una situazione simile a quella dello scarico libero, infatti nessun evento catastrofico è facilmente identificabile al termine dello svuotamento e le bande in frequenza tendono a mantenersi attive fino alla fine. Anche dai risultati dell'analisi *Wavelet*

Packets, rappresentati nelle figure 7.24, si possono ricavare le stesse informazioni. Colpisce infatti che rispetto al caso con scarico libero, la potenza del segnale si distribuisce su più bande e con una dinamica che ricorda più quella della formazione del vortice. Assolutamente particolare risulta invece l'analisi della trasformata di Hilbert, rappresentata nelle figure 7.25. L'immagine 7.25a rivela che la frequenza istantanea dopo la brusca caduta iniziale tende a rimanere costante, mentre nel caso 7.25b essa risale verso il valore iniziale. In questo caso al termine compare una brusca caduta che ricorda molto il fenomeno del *drain-sink*. Il grafico 7.25a ricorda invece il comportamento in presenza di vortice. I grafici relativi al canale 2 sono invece diversi e non permettono di apprezzare andamenti interessanti.

7.3.4 Scarico in vasca con vortici

La comparsa di un vortice visibile modifica la potenza del valore efficace. Come si può notare dai grafici 7.26, i livelli del segnale sono superiori a quelli dei grafici 7.21. Permane invece l'evidente uguaglianza di comportamento tra i due canali. Le densità di potenza media (figura 7.27) rivelano la ormai nota struttura, con il canale CH2 che contiene un contributo in frequenza maggiore. Gli spettrogrammi (figura 7.28) mostrano che l'andamento è simile al caso di getto libero con vortice: le frequenze più alte tendono ad eccitarsi dopo. Il fenomeno può essere apprezzato anche mediante l'analisi *Wavelet Packets* riportata nelle immagini 7.29. Per quanto riguarda l'analisi di Hilbert i grafici 7.30 rivelano che il comportamento è ben definito solo nel caso riportato in figura 7.30b.

7.3.5 Analisi dei tempi di svuotamento

Dall'esame congiunto dei dati della cella di carico e del livello di potenza del segnale, è possibile analizzare i tempi di svuotamento. I dati relativi alla serie di test con assenza di rotazione indotta sono riportati nella tabella 7.2. Osserviamo che i valori sono abbastanza uniformi. Il fenomeno del *drain-sink* mostra una durata dell'ordine dei 5 – 6sec. I dati relativi alla serie con scarico libero ma velocità indotta sono invece riportati in tabella 7.3. In questo caso i tempi di svuotamento sono molto variabili, anche se appare chiaro che il vortice tende ad allungare il tempo di svuotamento.

TEST	$t_{svuotamento}$	$t_{drain-sink}$
001A	$60 \pm 1sec$	$6 \pm 1sec$
002A	$61 \pm 1sec$	$5 \pm 1sec$
008A	$62 \pm 1sec$	$6 \pm 1sec$
009A	$61 \pm 1sec$	$6 \pm 1sec$
010A	$63 \pm 1sec$	$5 \pm 1sec$
014A	$62 \pm 1sec$	$6 \pm 1sec$
019A	$62 \pm 1sec$	$5 \pm 1sec$
022A	$64 \pm 1sec$	$6 \pm 1sec$

TABELLA 7.2: Tempi di svuotamento e durata del fenomeno di *drain-sink* nel caso della serie di test relativi allo scarico libero senza rotazione indotta.

TEST	$t_{svuotamento}$
003A	$65 \pm 1sec$
004A	$64 \pm 1sec$
005A	$84 \pm 1sec$
006A	$73 \pm 1sec$
007A	$84 \pm 1sec$
015A	$74 \pm 1sec$
020A	$60 \pm 1sec$
021A	$67 \pm 1sec$
023A	$72 \pm 1sec$

TABELLA 7.3: Tempi di svuotamento nel caso della serie di test relativi allo scarico libero con rotazione indotta.

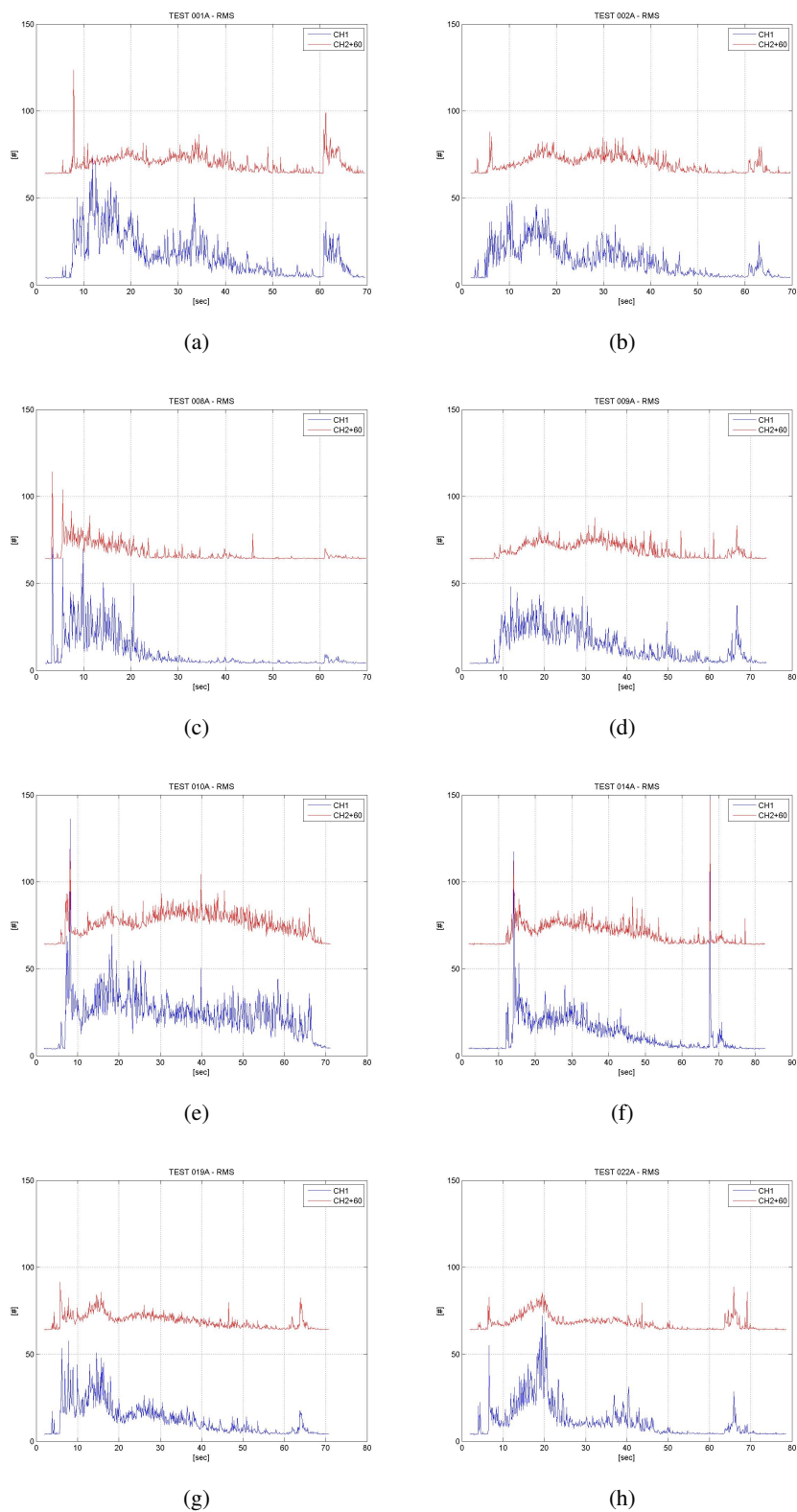


FIGURA 7.3: Svuotamento senza vortice. Grafici con l'andamento del valore *rms*. Dopo il salto dovuto all'apertura del getto l'energia progressivamente diminuisce. Al termine si registra un impulso dovuto al fenomeno di *drain-sink*. La posizione dell'accelerometro non influenza i risultati.

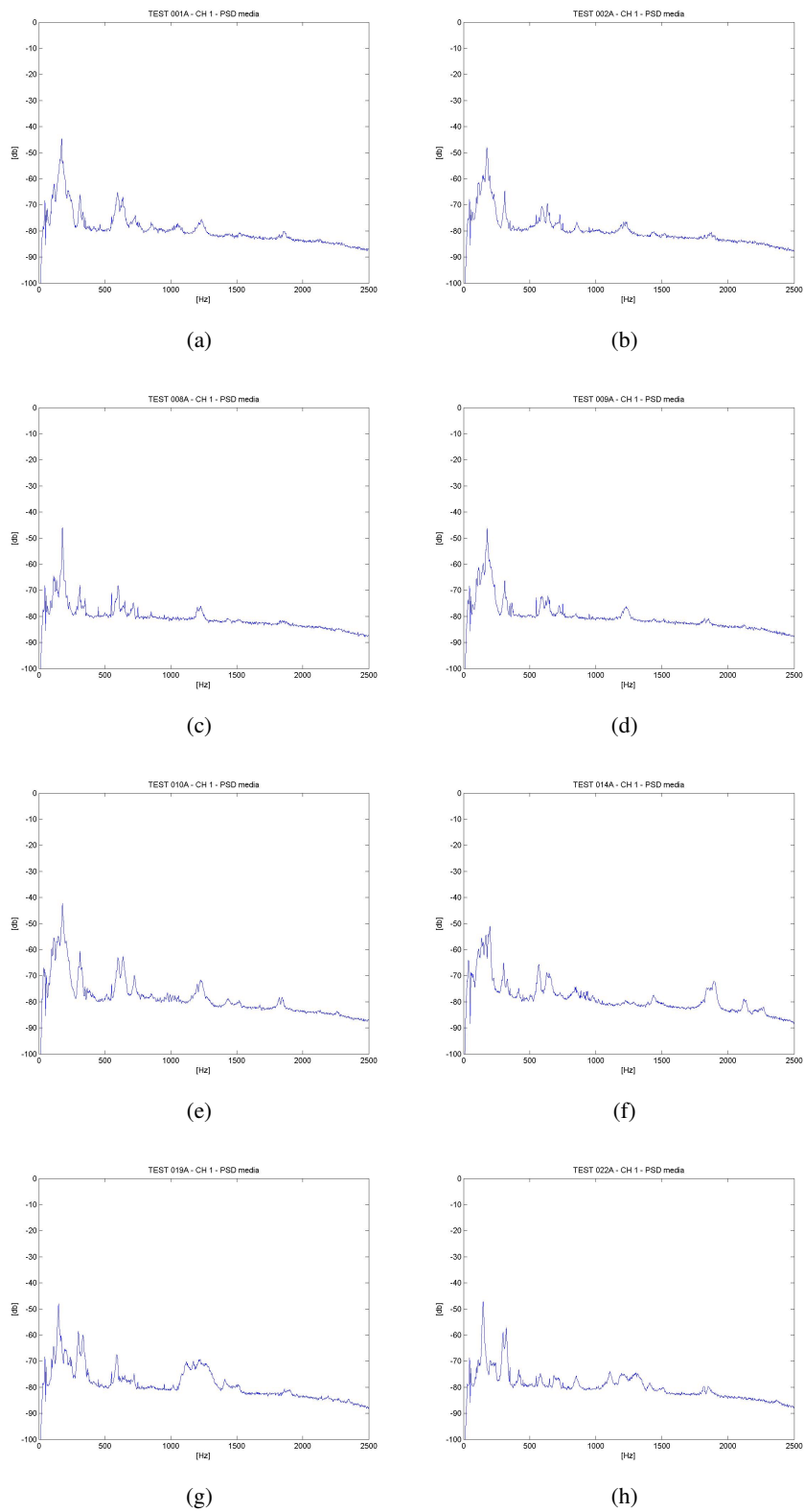


FIGURA 7.4: Svuotamento senza vortice. Grafici con la densità di potenza media del canale CH1 in funzione della frequenza. Svuotamento senza vortice.

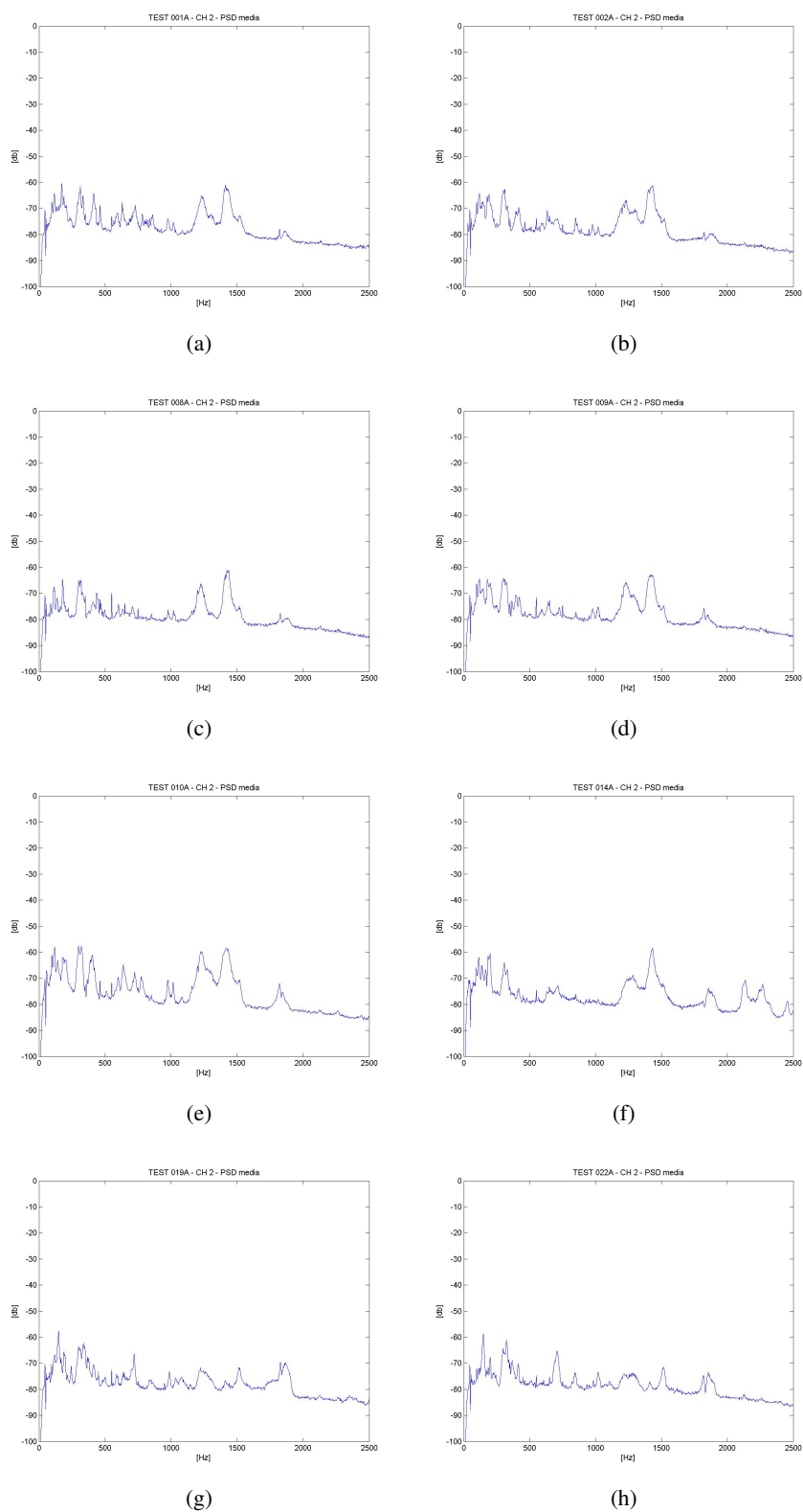


FIGURA 7.5: Svuotamento senza vortice. Grafici con la densità di potenza media del canale CH2 in funzione della frequenza. Si evidenzia una struttura a bande simile al canale CH1 con una maggiore estensione verso le alte frequenze.

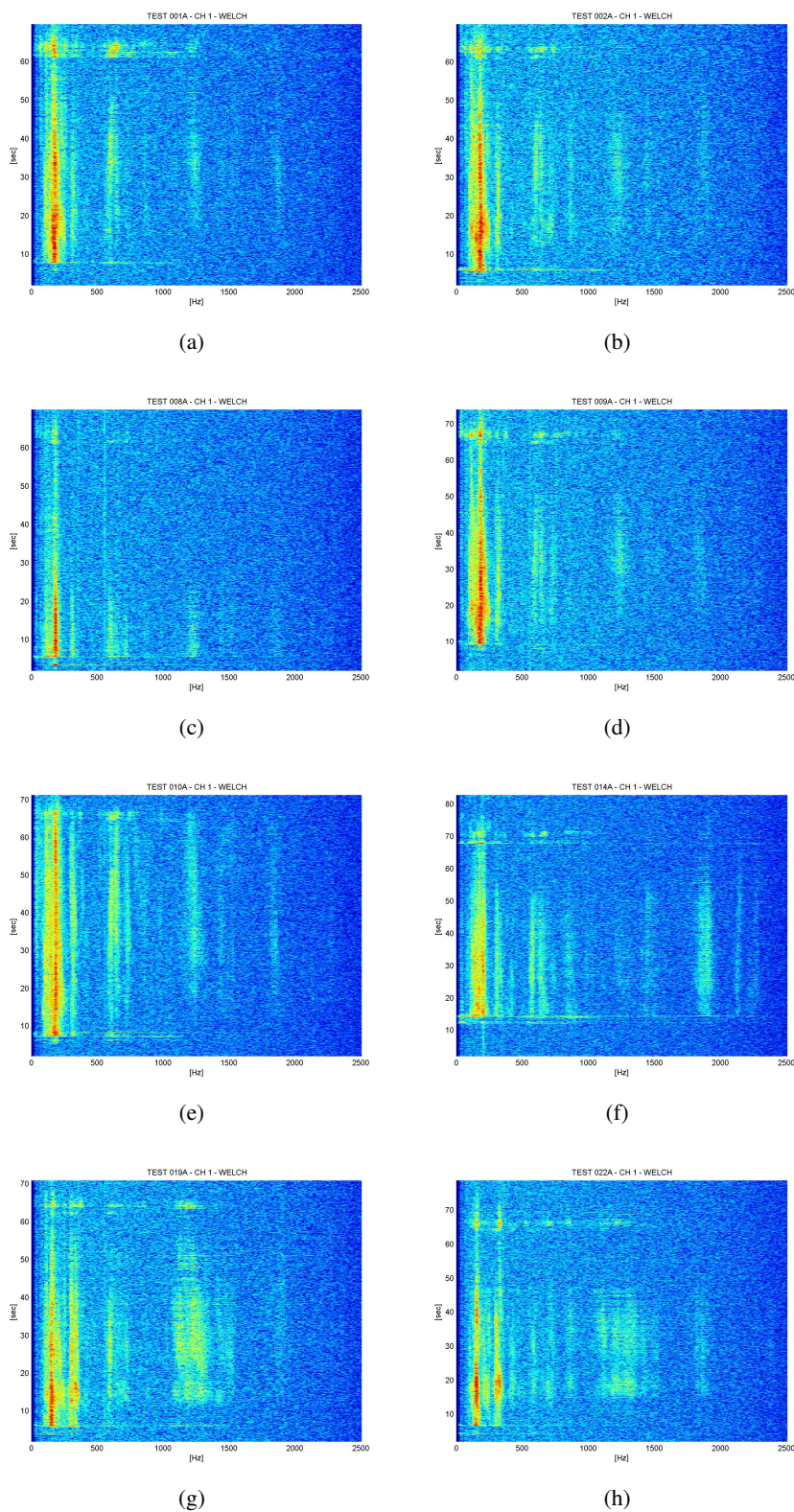


FIGURA 7.6: Svuotamento senza vortice. Spettrogramma relativo al canale CH1. La posizione dell'accelerometro non modifica in modo apprezzabile lo schema. Tutte le bande a frequenze maggiori si smorzano prima del termine, tranne nel caso (e).

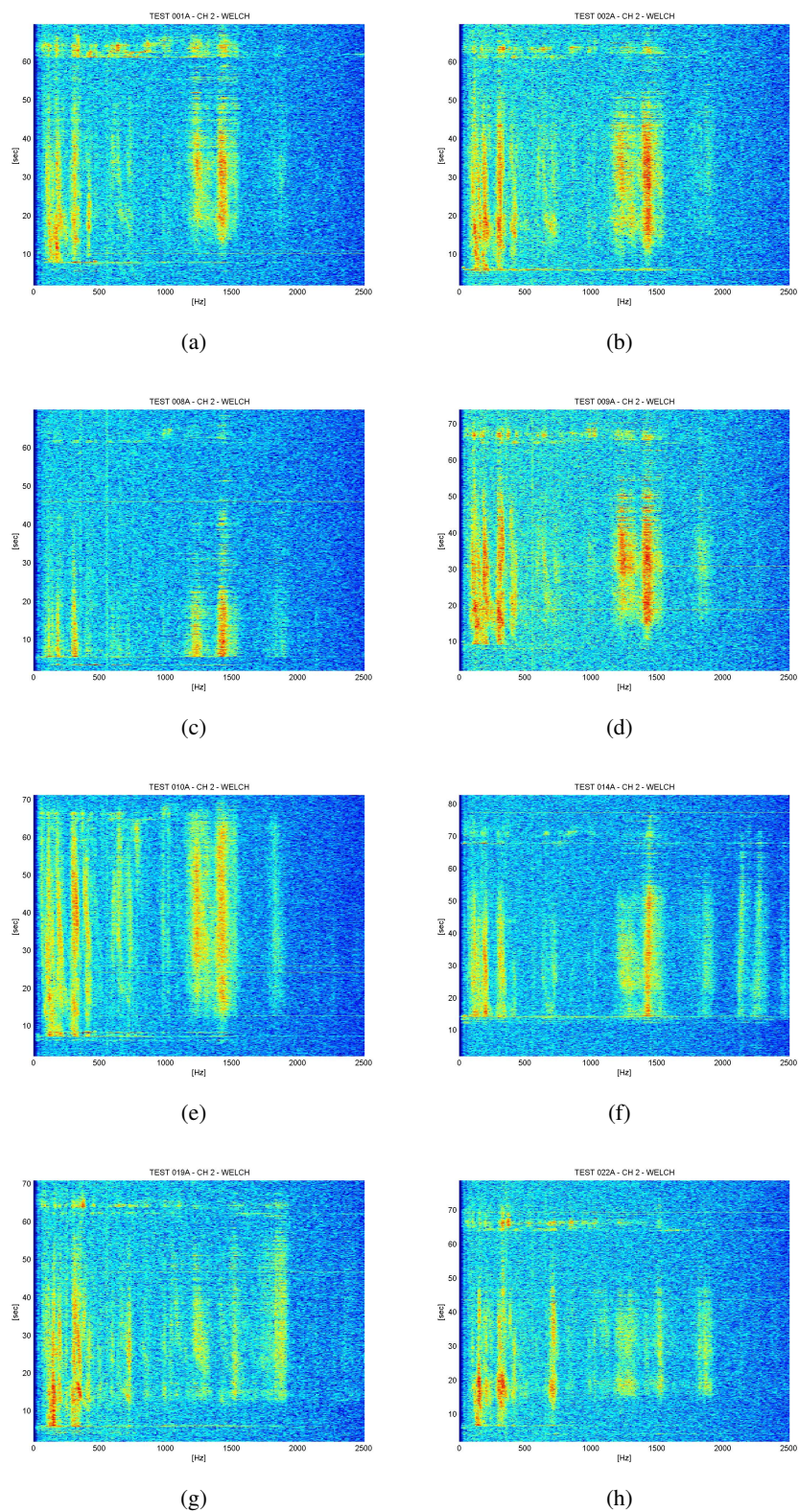


FIGURA 7.7: Svuotamento senza vortice. Spettrogramma relativo al canale CH2. I risultati sono simili a quelli del canale CH1, le bande a frequenza superiore si estendono maggiormente e appaiono più energetiche. Nella figura (e) nessuna banda si attenua in anticipo.

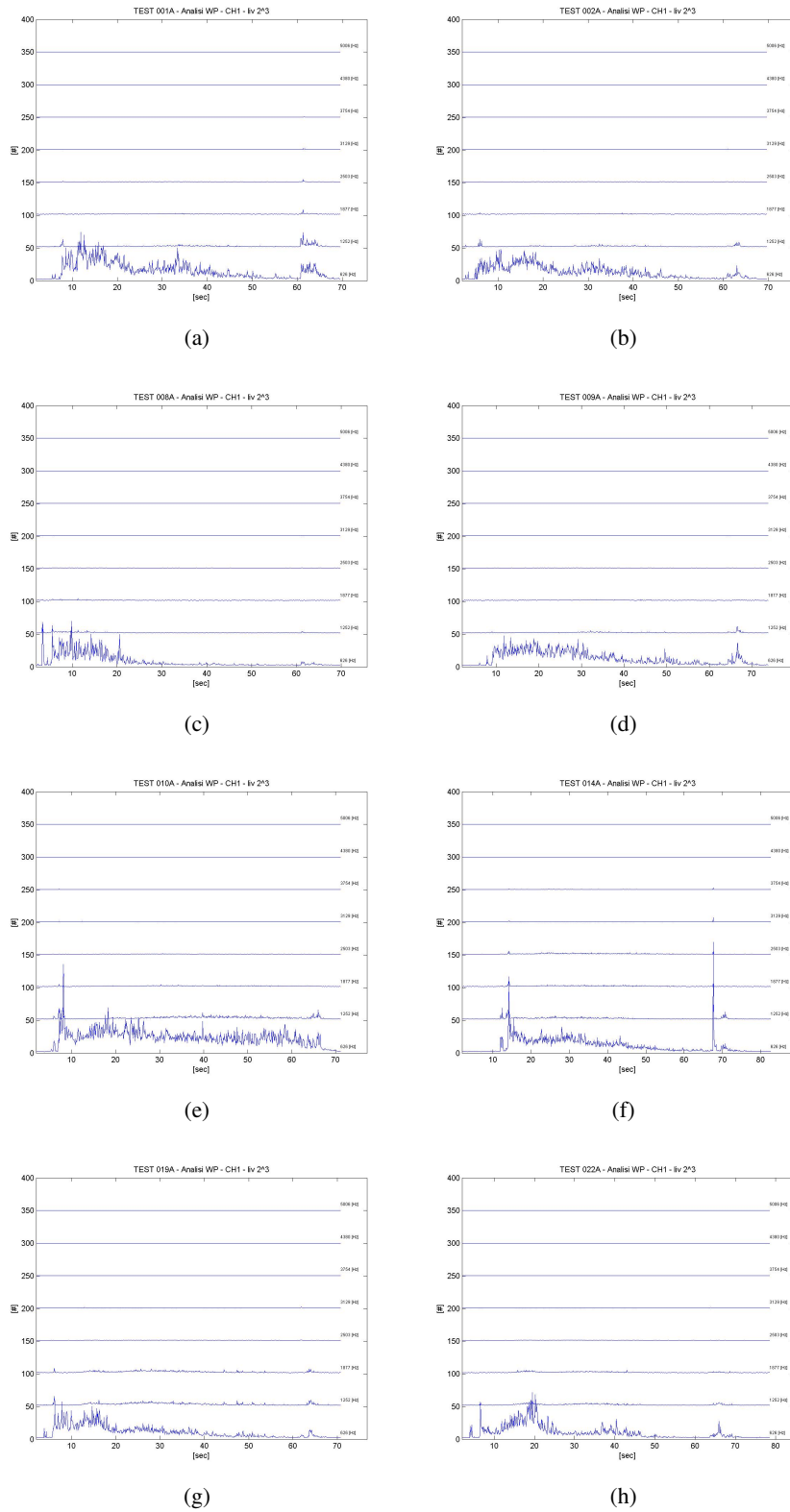


FIGURA 7.8: Svuotamento senza vortice. Analisi Wavelet Packet relativa al canale CH1. Il contenuto energetico è tutto incorporato nella prima banda. Si osserva che nel caso (e) l'evento *drain-sink* è perfettamente identificabile.

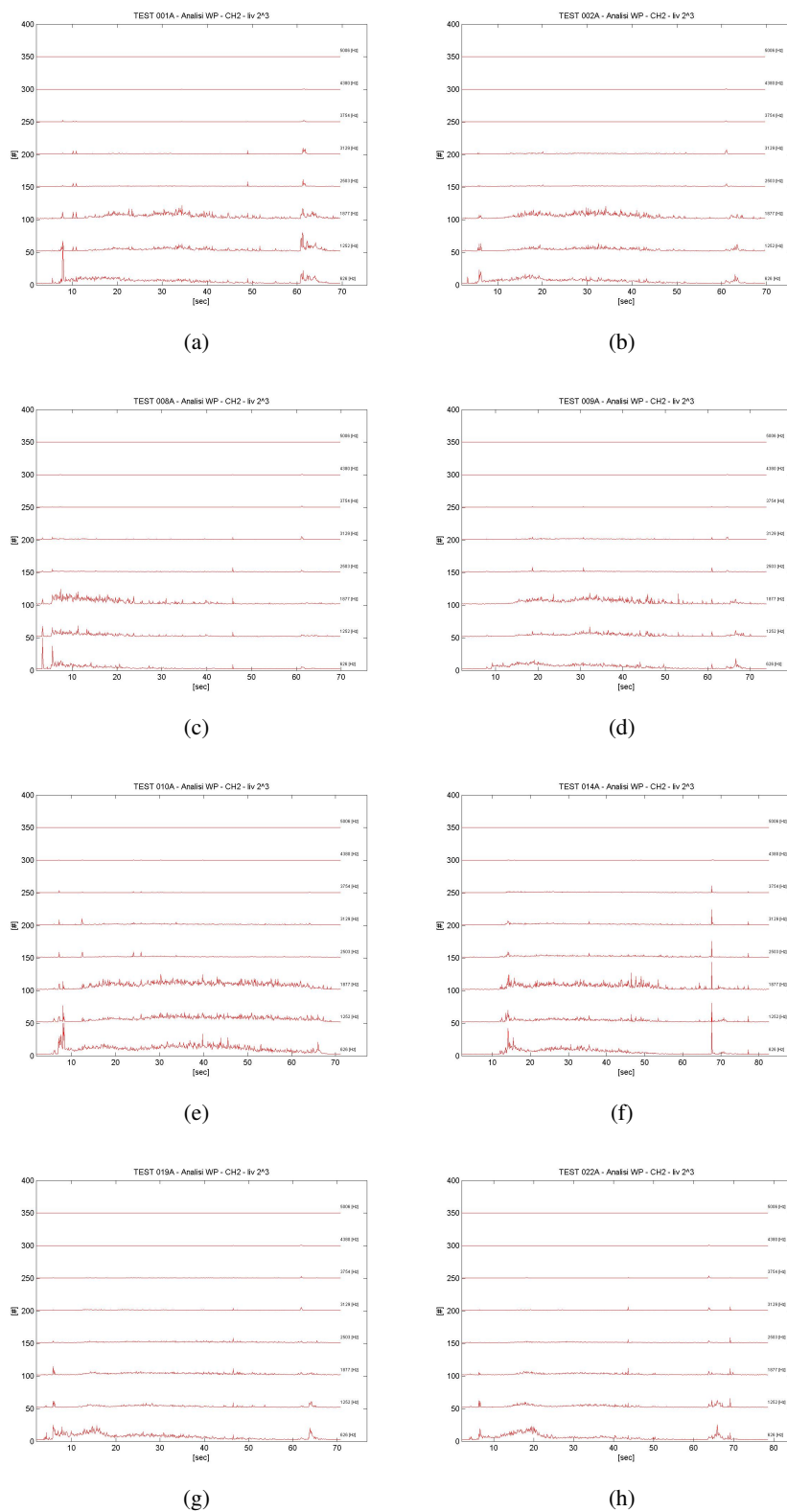


FIGURA 7.9: Svuotamento senza vortice. Analisi Wavelet Packet relativa al canale CH2. Rispetto al canale CH1 le bande superiori sono più energetiche. L'evento *drain-sink* è perfettamente identificabile.

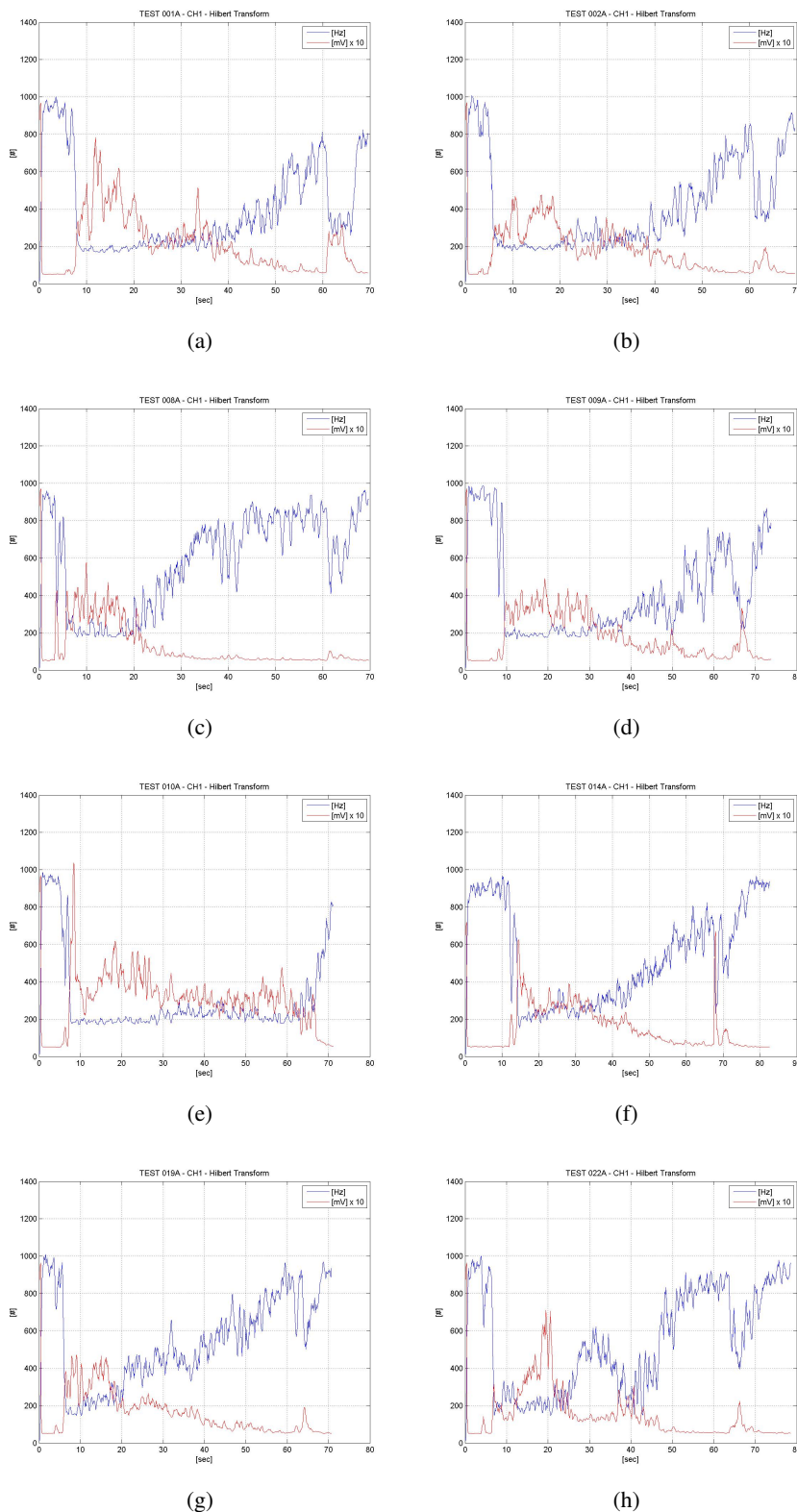


FIGURA 7.10: Svuotamento senza vortice. Trasformata di Hilbert relativa al canale CH1. L'ampiezza istantanea ricalca l'andamento del valore *rms* globale. La frequenza istantanea dopo una brusca variazione iniziale tende progressivamente a risalire al variare dell'ampiezza istantanea. La figura (e) mostra invece un andamento diverso. Il fenomeno *drain-sink* è perfettamente identificabile.

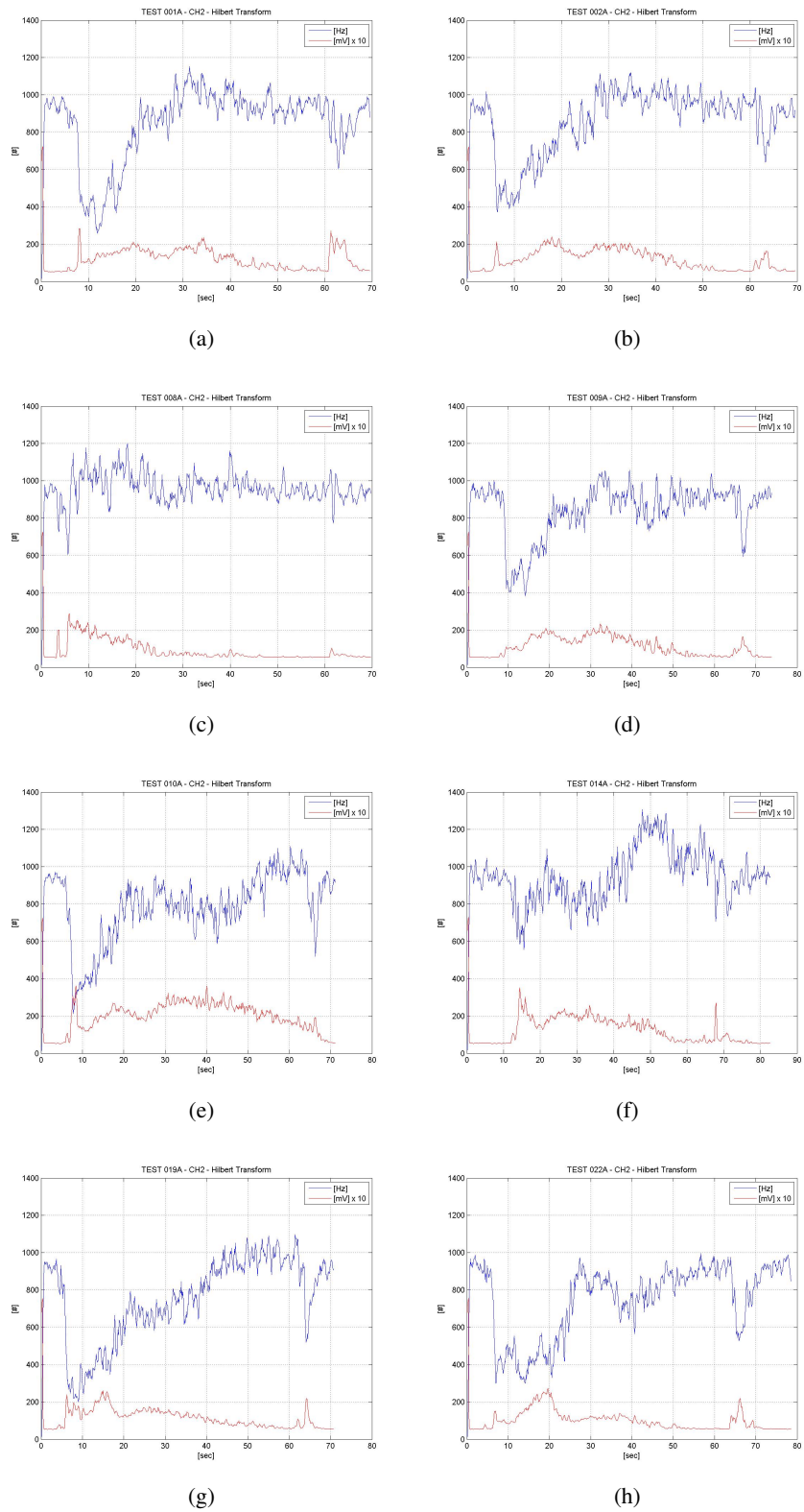
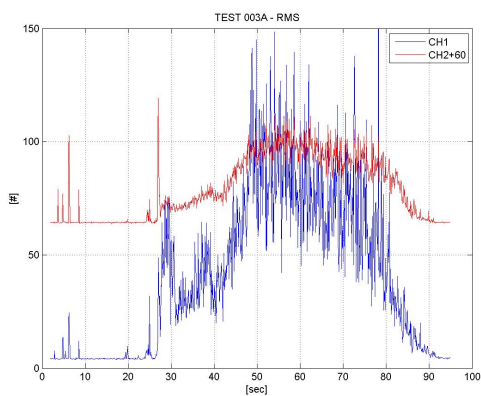
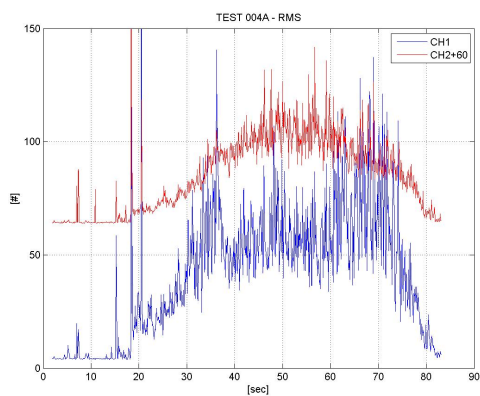


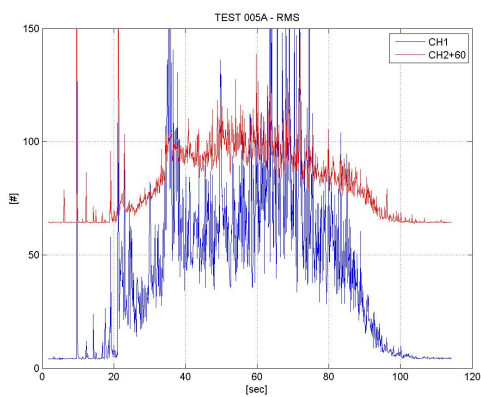
FIGURA 7.11: Svuotamento senza vortice. Trasformata di Hilbert relativa al canale CH2. L'ampiezza istantanea riproduce l'andamento del valore *rms* globale. La frequenza istantanea analogamente al CH1 presenta una brusca variazione iniziale ma con una risalita diversa. Il fenomeno *drain-sink* è identificabile.



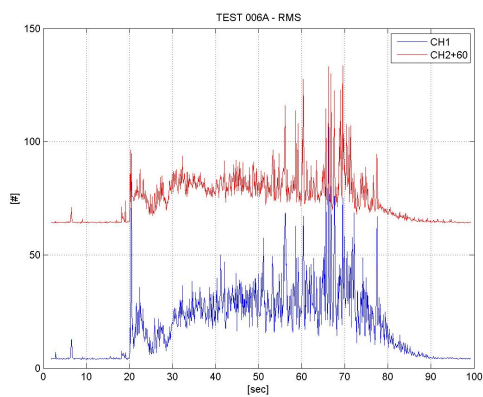
(a)



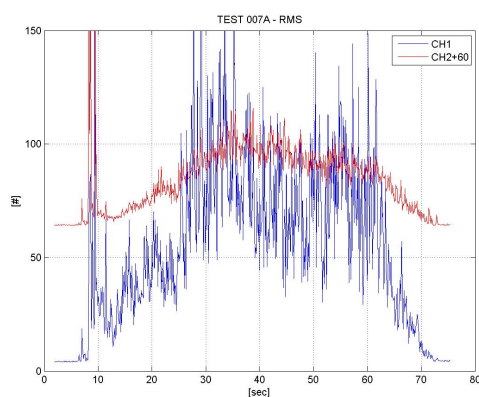
(b)



(c)



(d)



(e)

FIGURA 7.12: Svuotamento con vortice. Grafici con l'andamento del valore *rms*. Il livello di potenza risulta nettamente superiore al caso con *drain-sink*

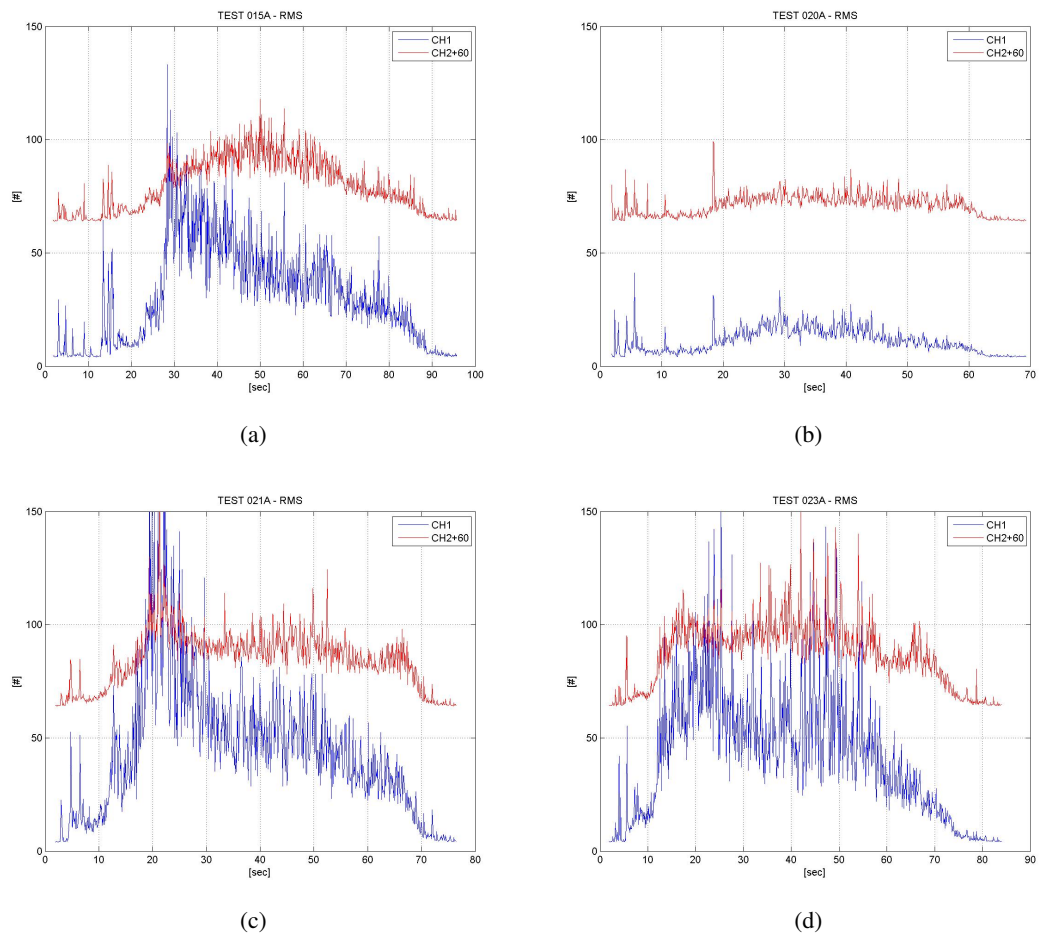


FIGURA 7.12: Svuotamento con vortice. Grafici con l'andamento del valore *rms*. Il livello di potenza risulta nettamente superiore al caso con *drain-sink*

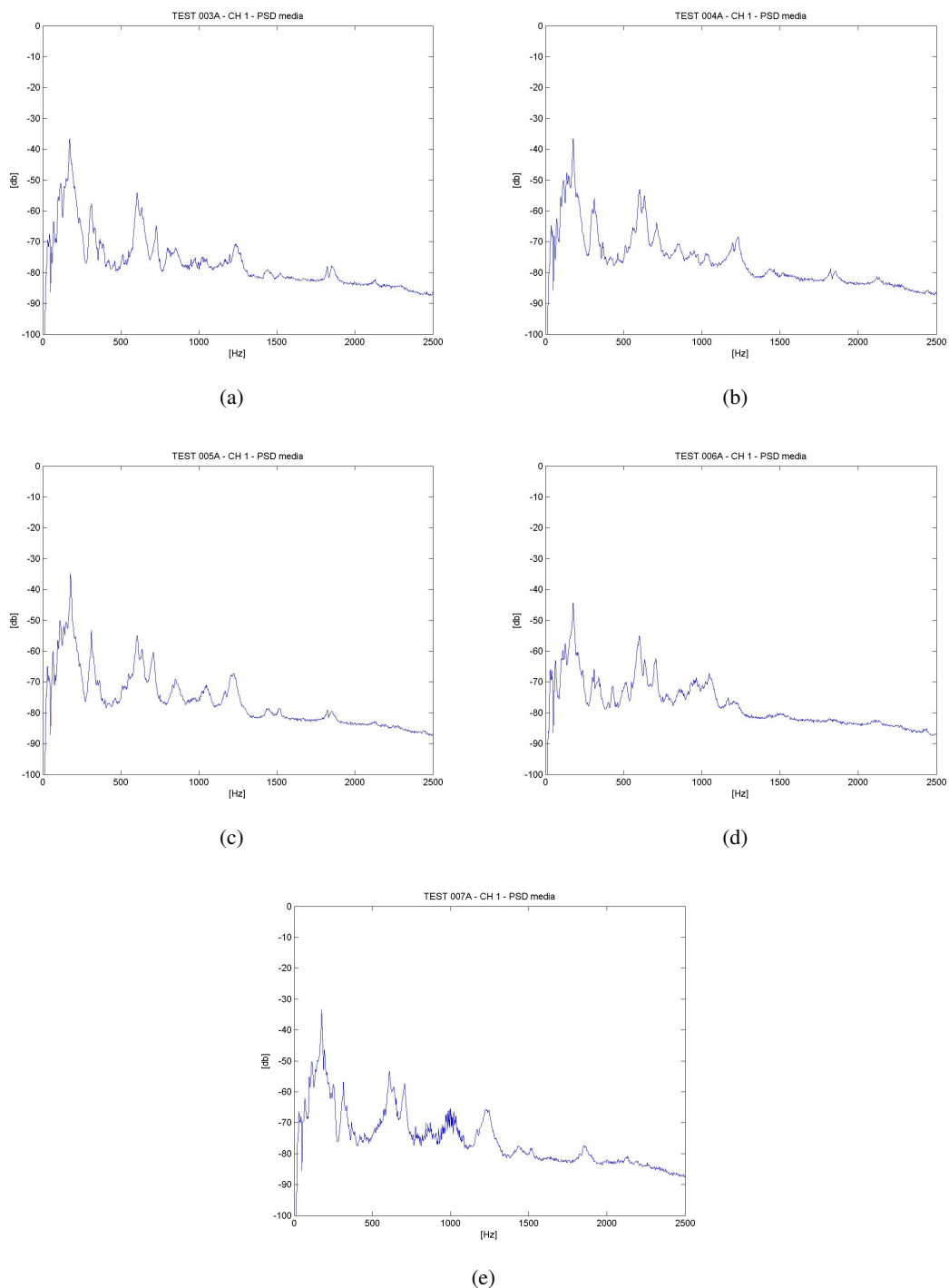


FIGURA 7.13: Svuotamento con vortice. Grafici con la densità di potenza media del canale CH1 in funzione della frequenza. Lo schema a bande è simile al caso senza vortice.

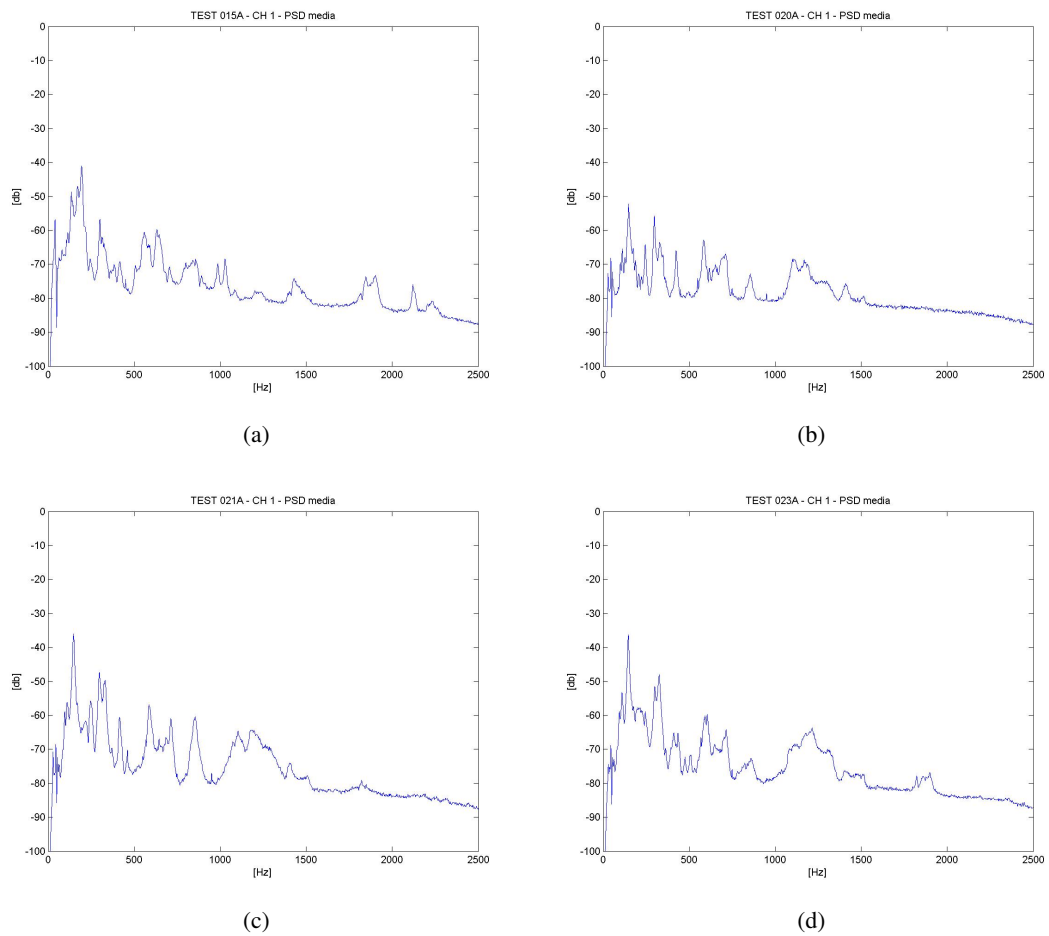
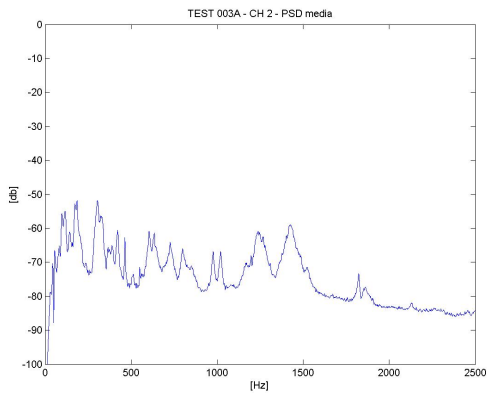
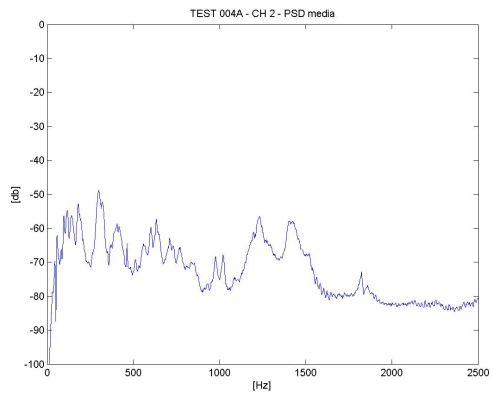


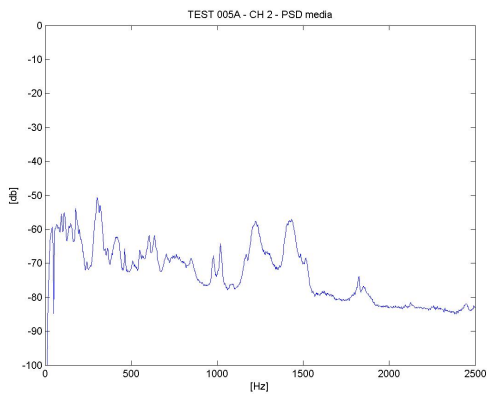
FIGURA 7.13: Svuotamento con vortice. Grafici con la densità di potenza media del canale CH1 in funzione della frequenza. Lo schema a bande è simile al caso senza vortice.



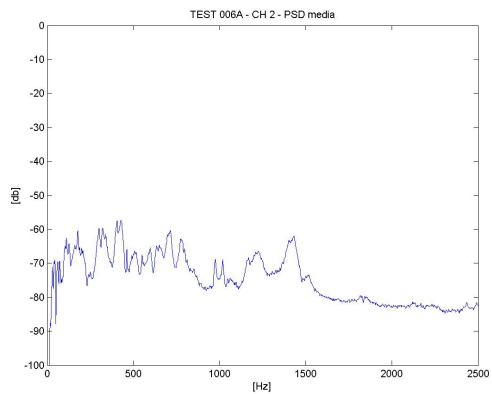
(a)



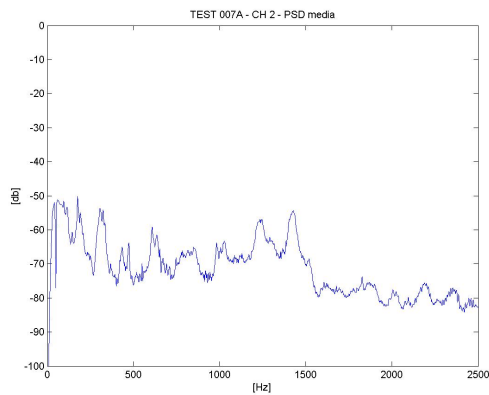
(b)



(c)



(d)



(e)

FIGURA 7.14: Svuotamento con vortice. Grafici con la densità di potenza media del canale CH2 in funzione della frequenza. Lo schema è simile al canale CH1 con un contributo maggiore delle frequenze più alte.

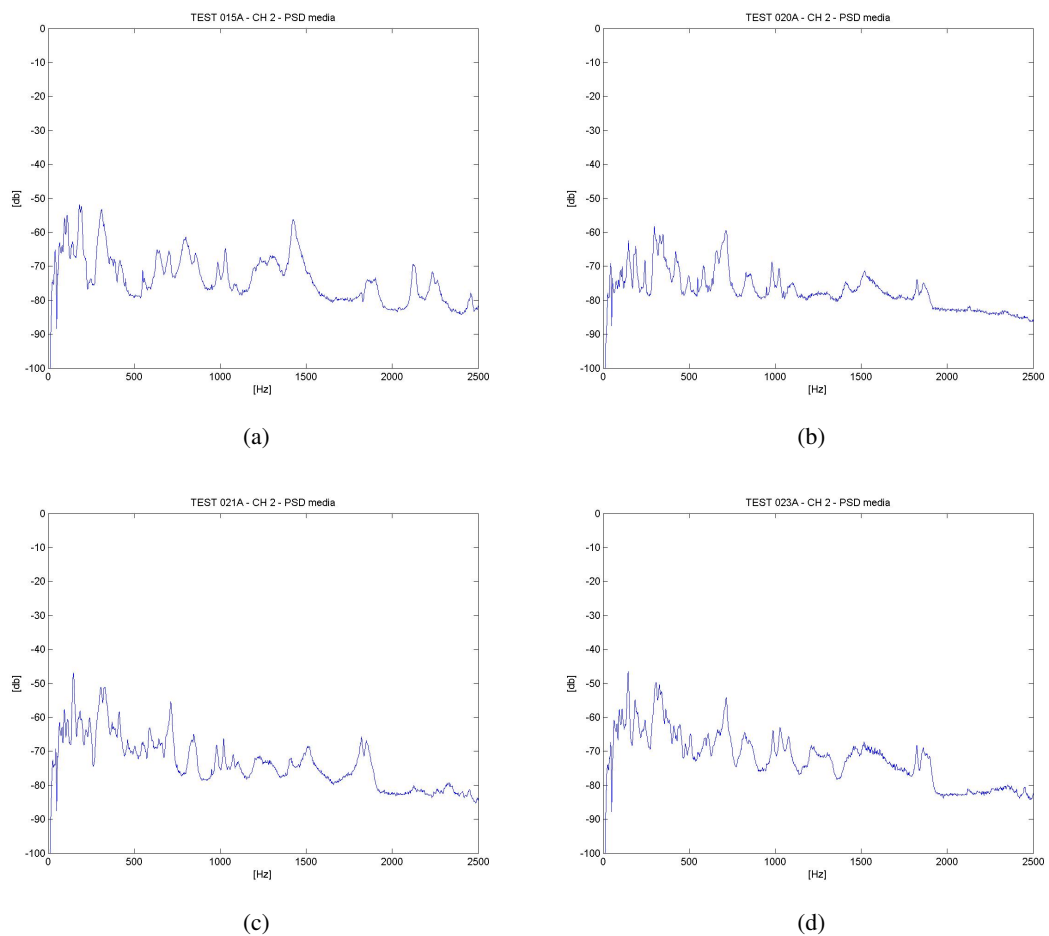
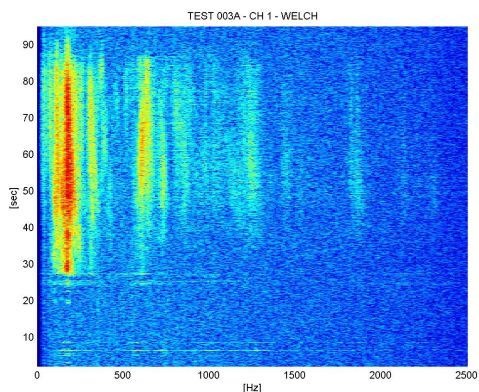
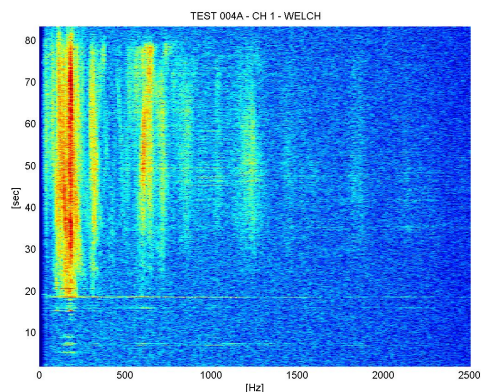


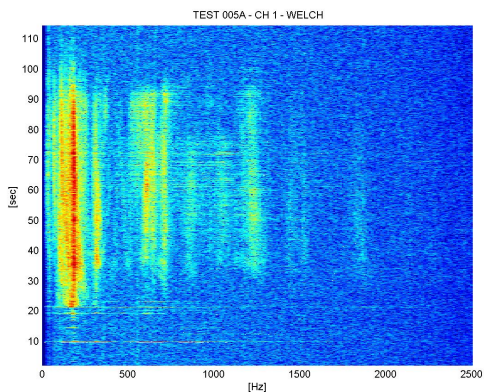
FIGURA 7.14: Svuotamento con vortice. Grafici con la densità di potenza media del canale CH2 in funzione della frequenza. Lo schema è simile al canale CH1 con un contributo maggiore delle frequenze più alte.



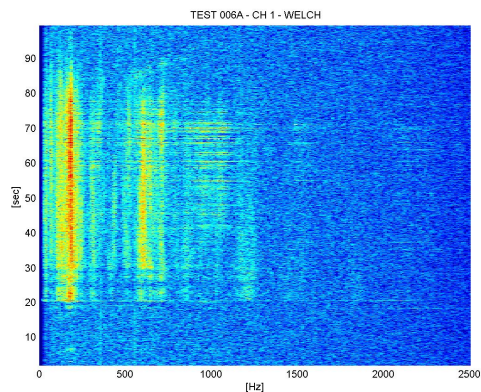
(a)



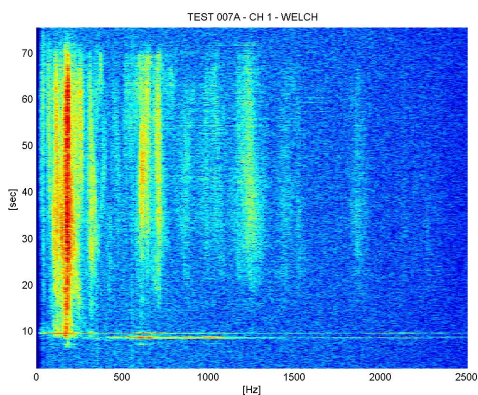
(b)



(c)



(d)



(e)

FIGURA 7.15: Svuotamento con vortice. Spettrogramma relativo al canale CH1. Risulta evidente la struttura a bande. E' immediato notare come i contributi ad alta frequenza fino alla fine del processo contrariamente al caso con *drain-sink*.

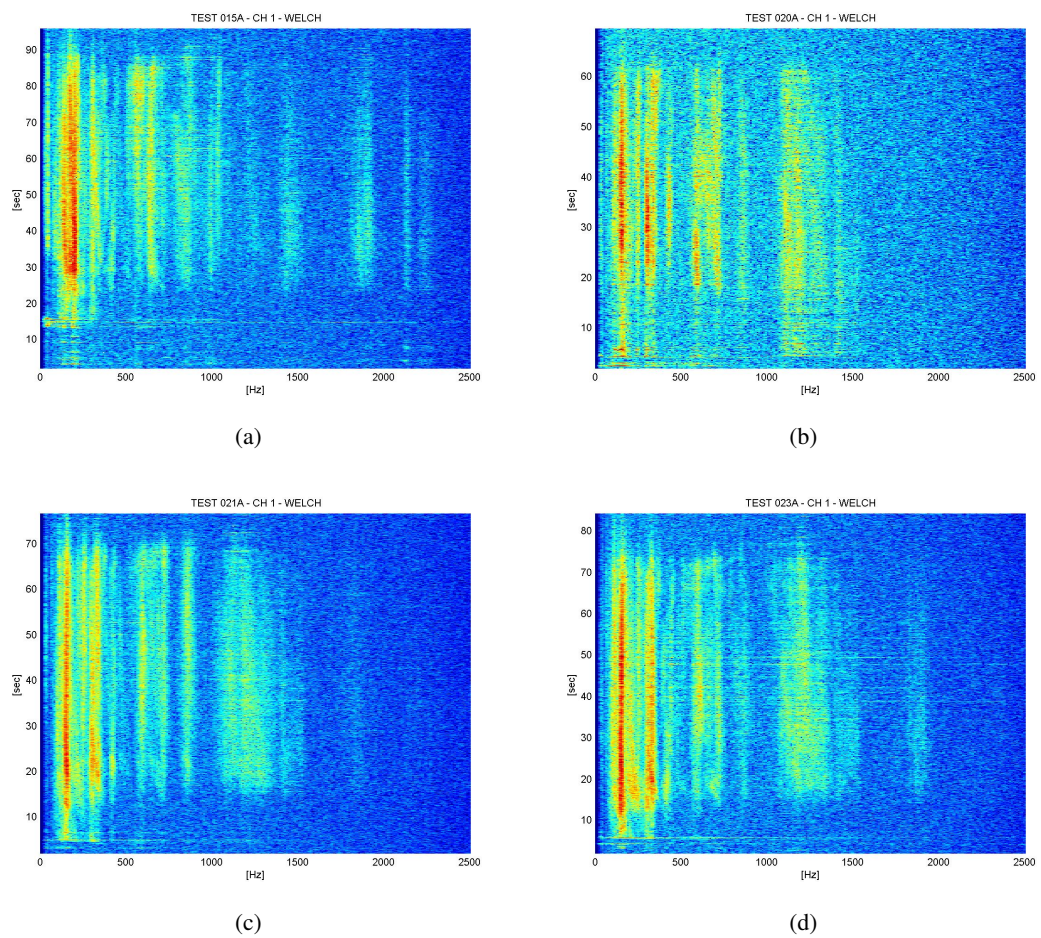


FIGURA 7.15: Svuotamento con vortice. Spettrogramma relativo al canale CH1. Risulta evidente la struttura a bande. E' immediato notare come i contributi ad alta frequenza fino alla fine del processo contrariamente al caso con *drain-sink*.

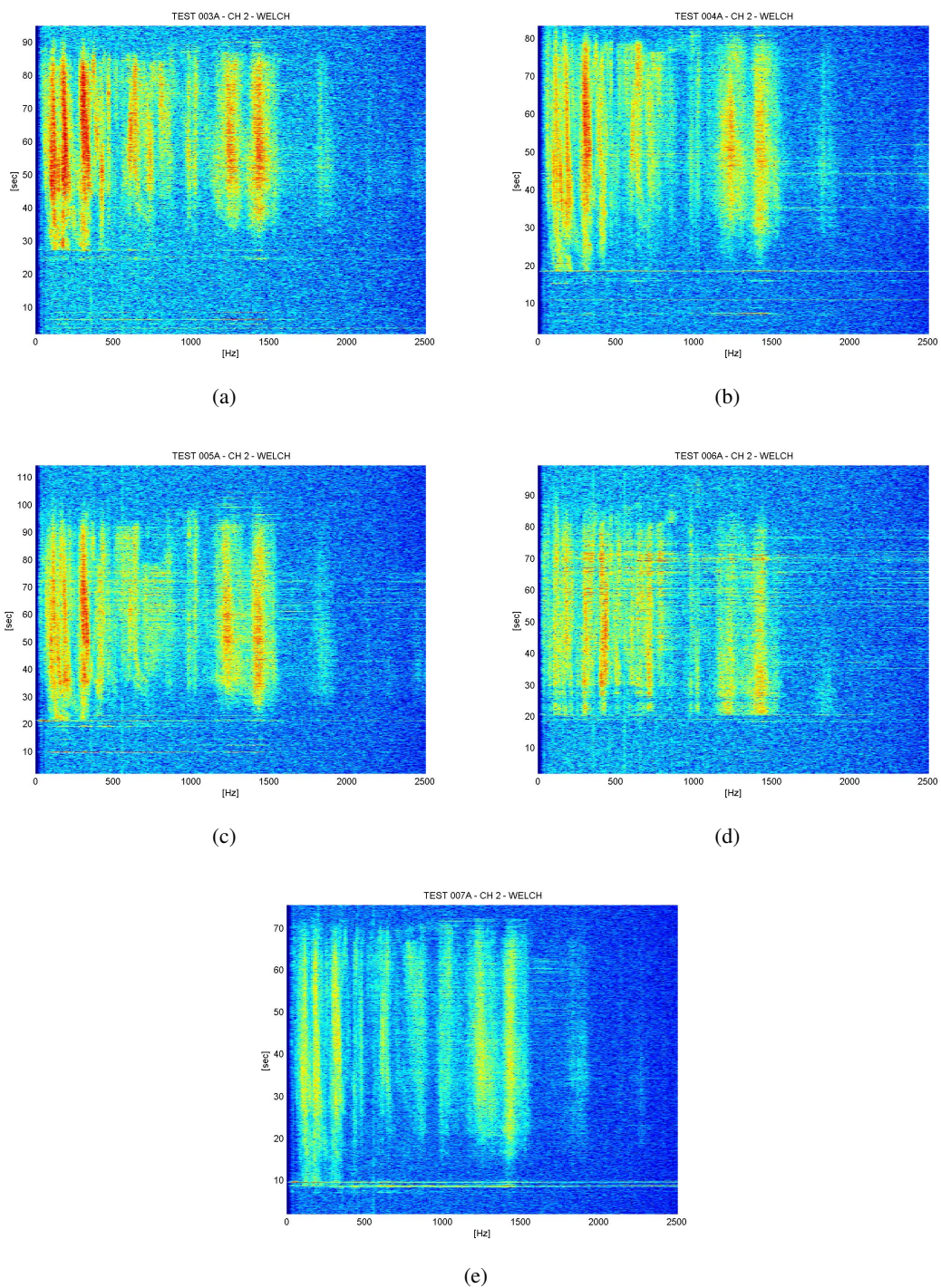


FIGURA 7.16: Svuotamento con vortice. Spettrogramma relativo al canale CH2. La struttura è simile al caso del canale CH1 ma con un maggior contributo delle frequenze più alte.

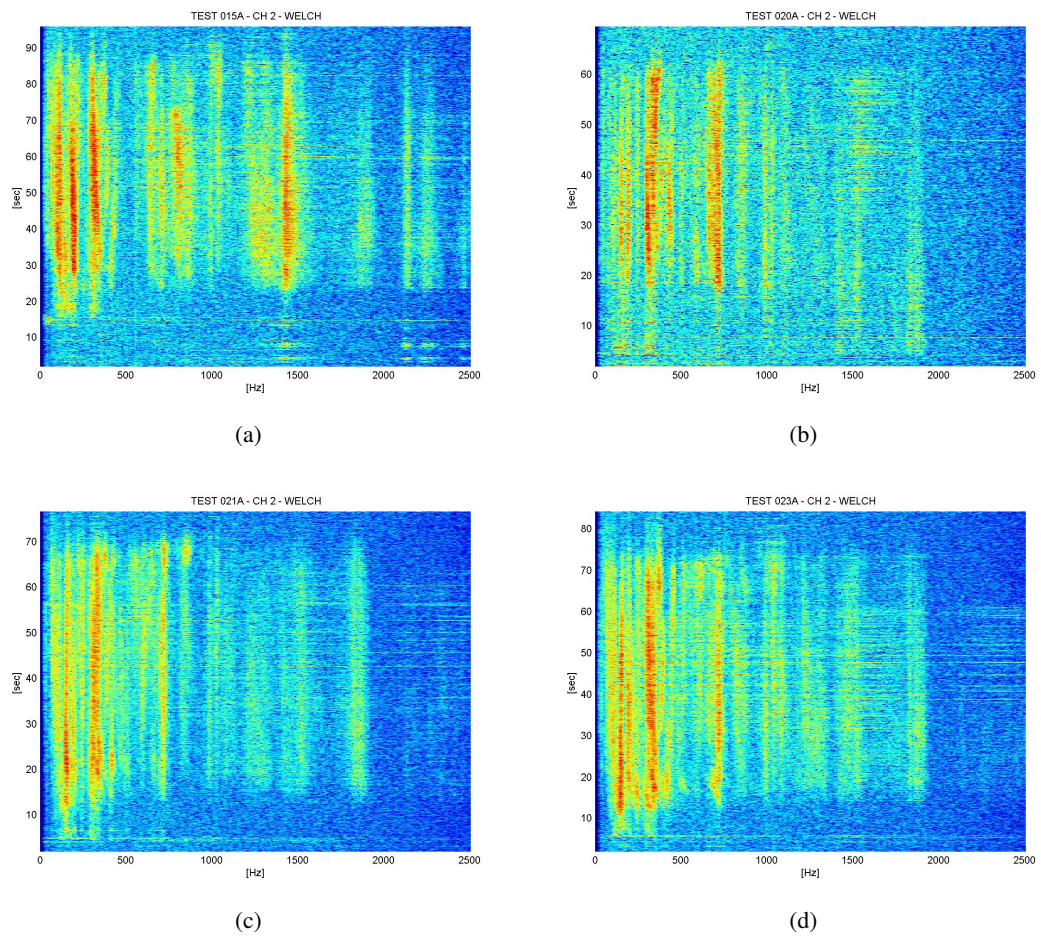


FIGURA 7.16: Svuotamento con vortice. Spetrogramma relativo al canale CH2. La struttura è simile al caso del canale CH1 ma con un maggior contributo delle frequenze più alte.

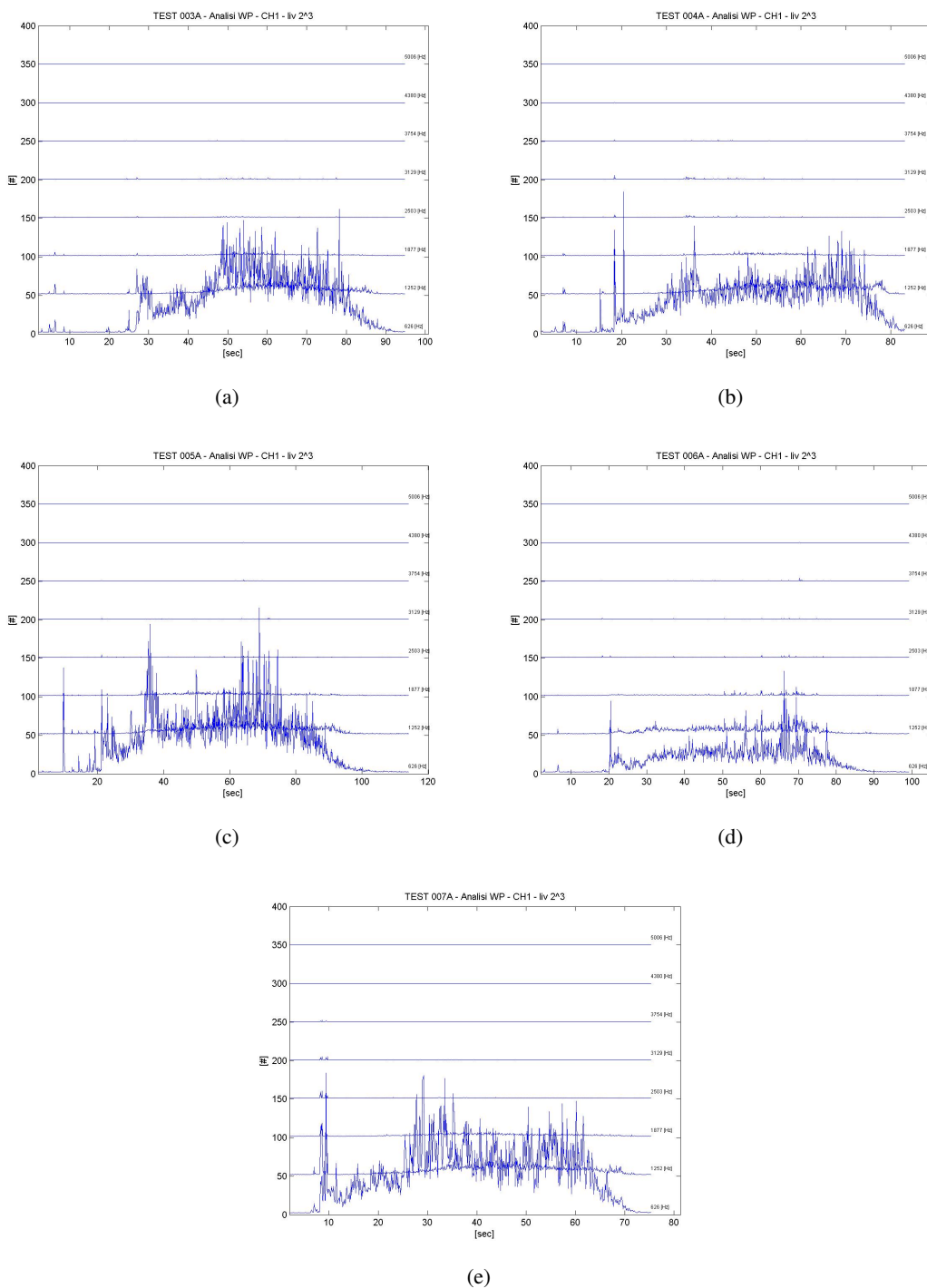


FIGURA 7.17: Svuotamento con vortice. Analisi Wavelet Packet relativa al canale CH1. Analogamente al caso del *drain-sink* la banda inferiore contiene il maggior contributo energetico. Le bande superiori mostrano una tendenza ad aumentare l'energia in prossimità dello svuotamento finale.

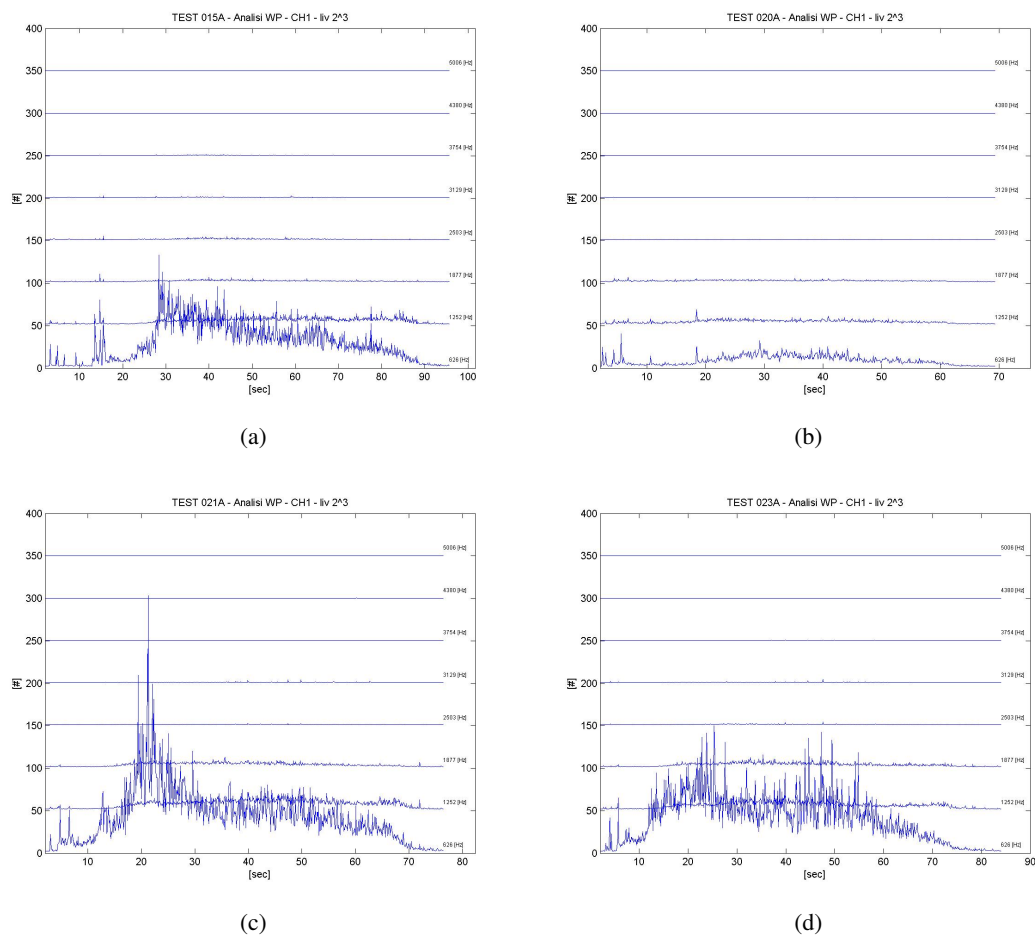


FIGURA 7.17: Svuotamento con vortice. Analisi Wavelet Packet relativa al canale CH1. Analogamente al caso del *drain-sink* la banda inferiore contiene il maggior contributo energetico. Le bande superiori mostrano una tendenza ad aumentare l'energia in prossimità dello svuotamento finale.

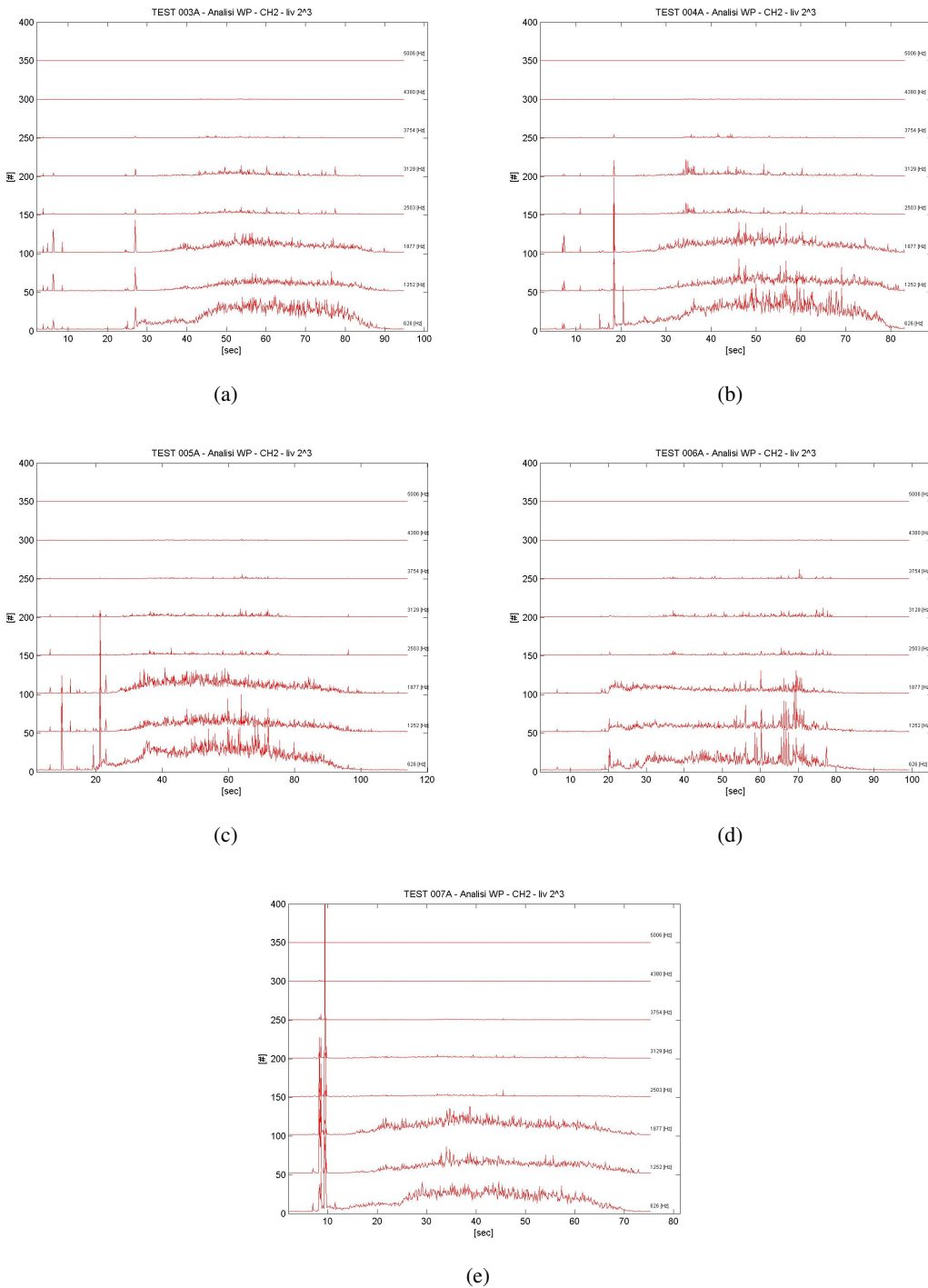


FIGURA 7.18: Svuotamento con vortice. Analisi Wavelet Packet relativa al canale CH2. Risulta evidente il maggior contributo delle bande superiori rispetto al caso CH1.

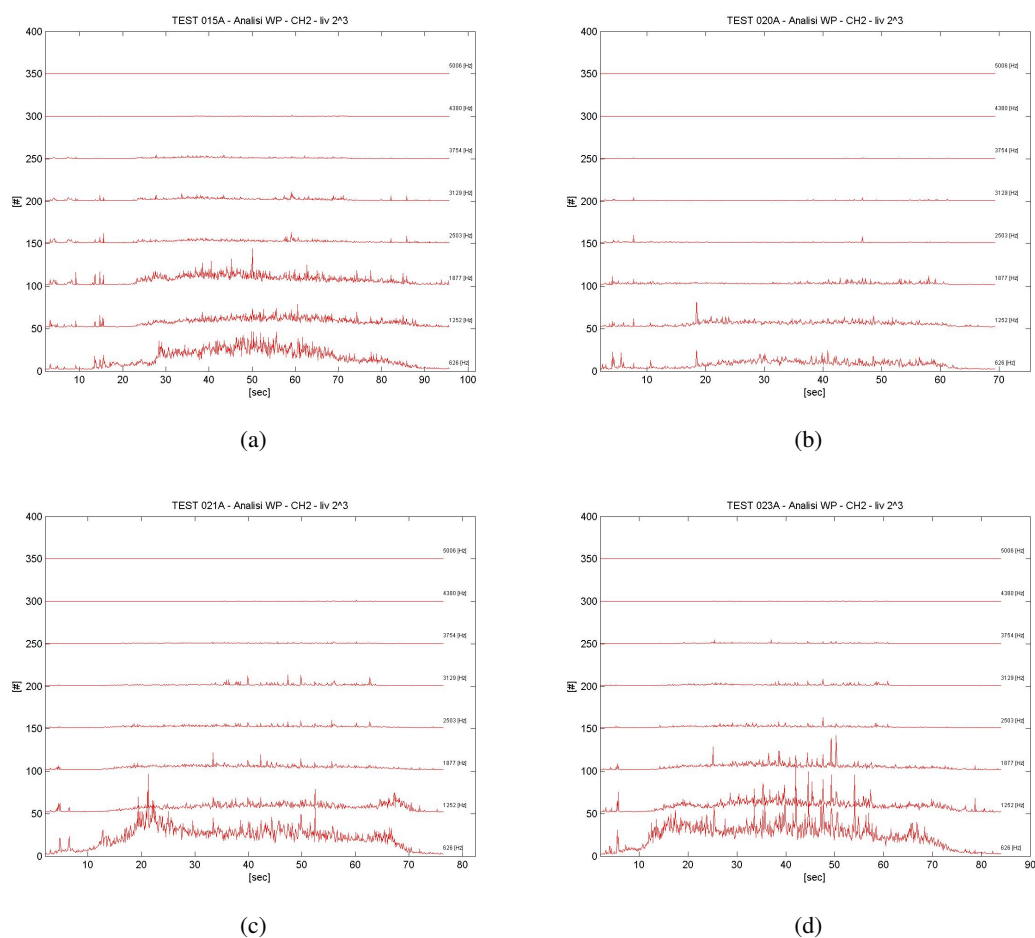
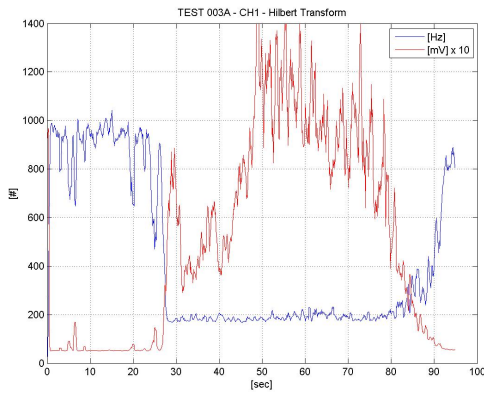
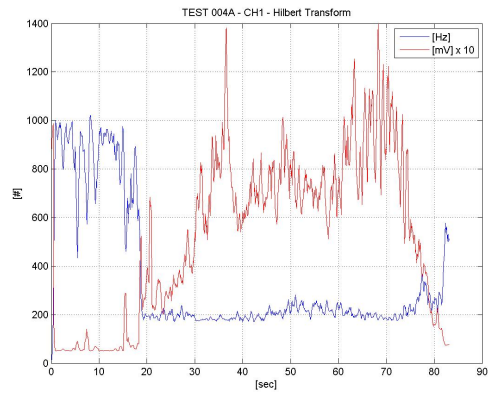


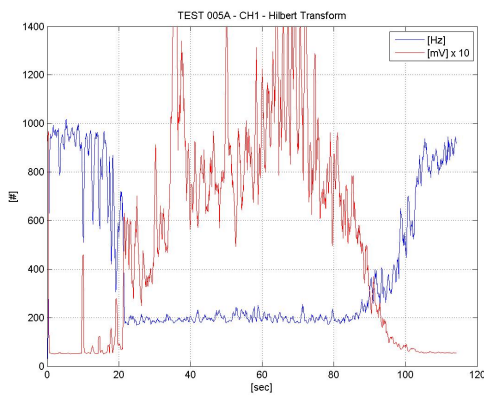
FIGURA 7.18: Svuotamento con vortice. Analisi Wavelet Packet relativa al canale CH2. Risulta evidente il maggior contributo delle bande superiori rispetto al caso CH1.



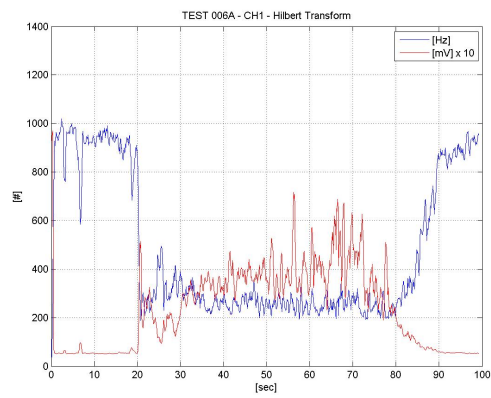
(a)



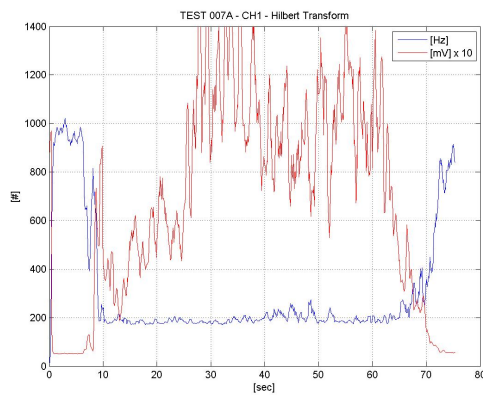
(b)



(c)



(d)



(e)

FIGURA 7.19: Svuotamento con vortice. Trasformata di Hilbert relativa al canale CH1. L'ampiezza istantanea mostra lo stesso andamento del valore *rms* globale. La frequenza istantanea dopo una brusca variazione tende a rimanere costante fino allo svuotamento finale mostrando un comportamento molto differente rispetto al caso senza vortice.

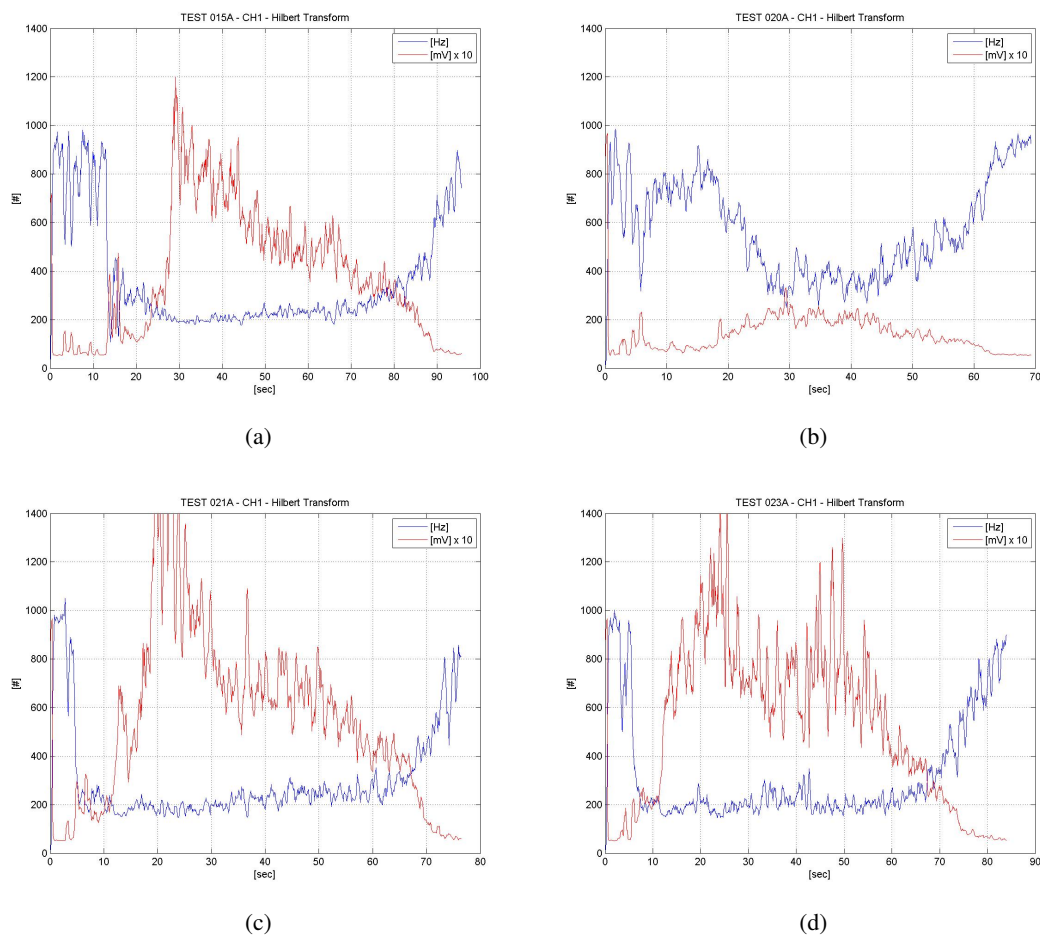
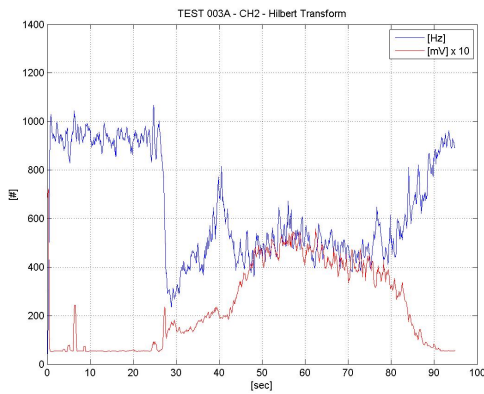
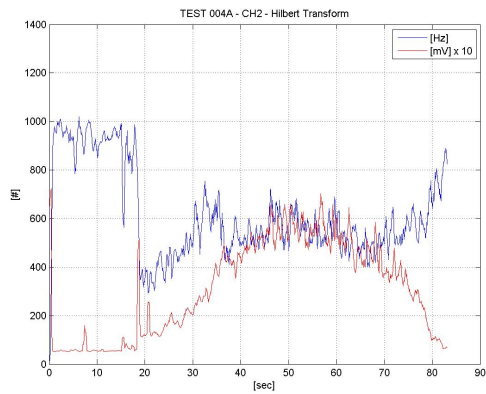


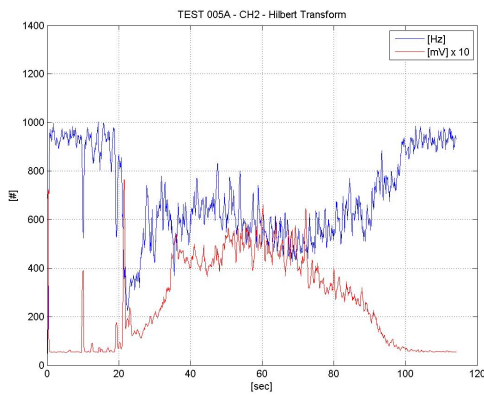
FIGURA 7.19: Svuotamento con vortice. Trasformata di Hilbert relativa al canale CH1. L'ampiezza istantanea mostra lo stesso andamento del valore *rms* globale. La frequenza istantanea dopo una brusca variazione tende a rimanere costante fino allo svuotamento finale mostrando un comportamento molto differente rispetto al caso senza vortice.



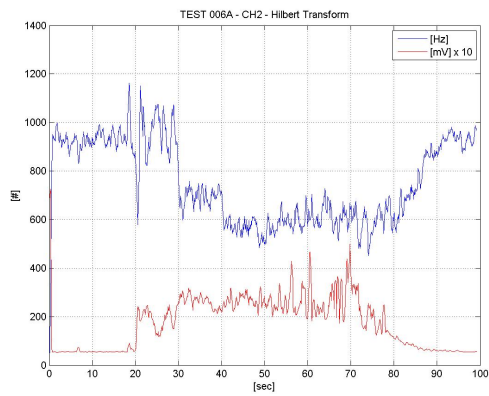
(a)



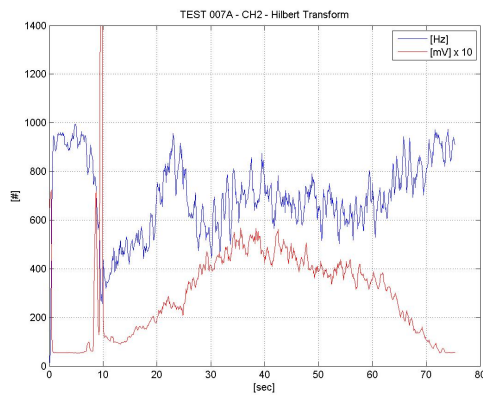
(b)



(c)



(d)



(e)

FIGURA 7.20: Svuotamento con vortice. Trasformata di Hilbert relativa al canale CH2. L'ampiezza istantanea mostra lo stesso andamento del valore *rms* globale. L'andamento della frequenza istantanea è simile a quello del canale CH1.

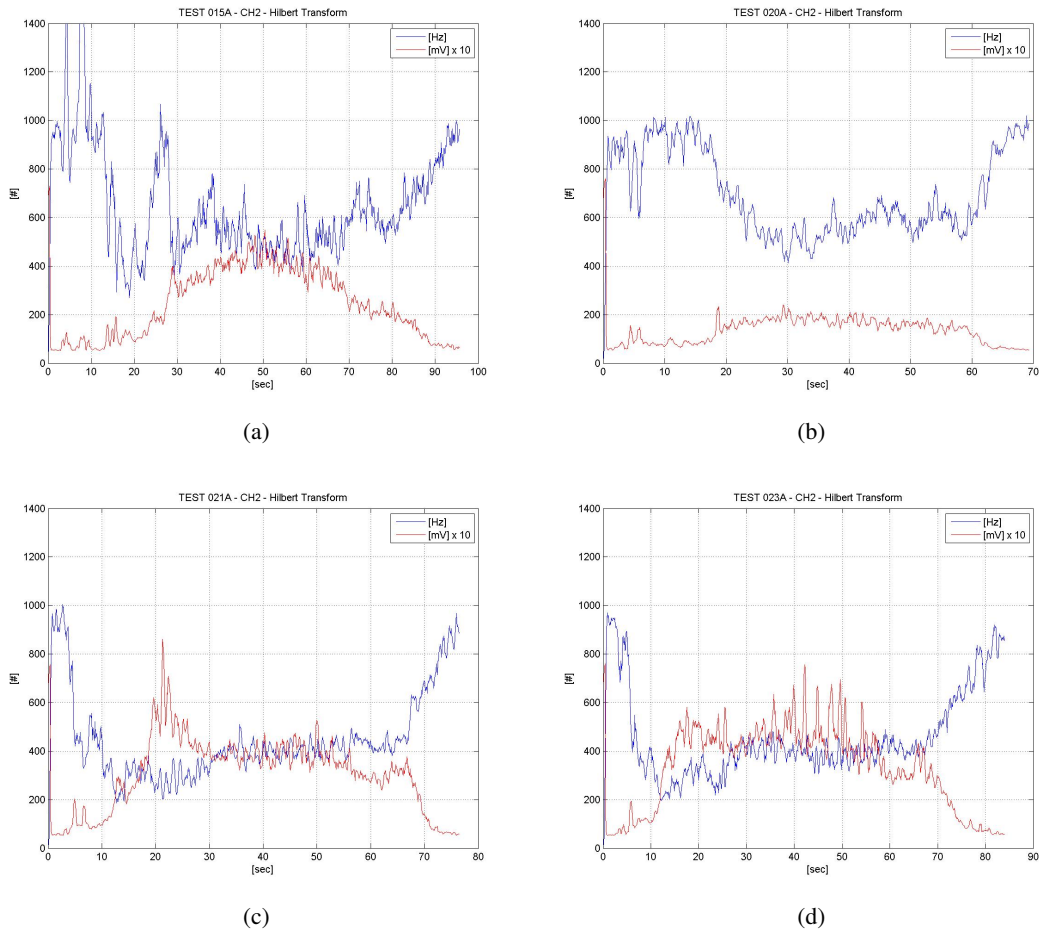


FIGURA 7.20: Svotamento con vortice. Trasformata di Hilbert relativa al canale CH2. L'ampiezza istantanea mostra lo stesso andamento del valore *rms* globale. L'andamento della frequenza istantanea è simile a quello del canale CH1.

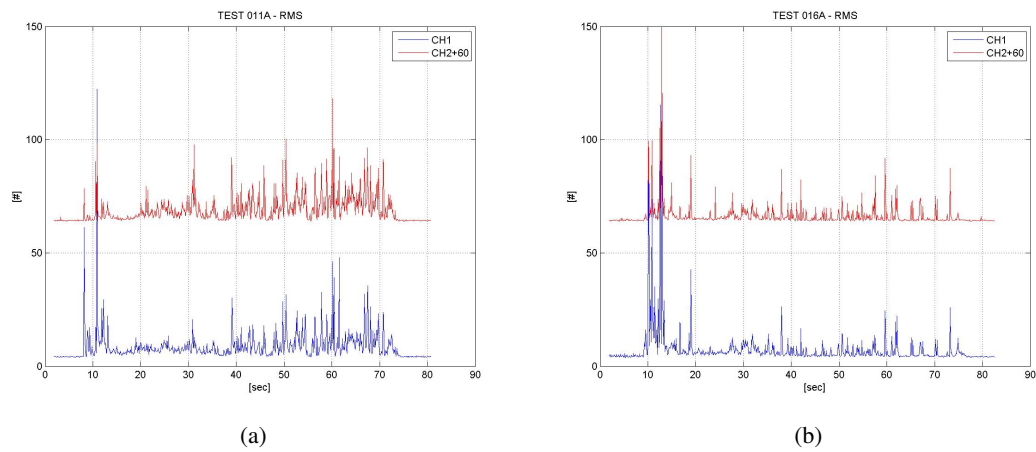
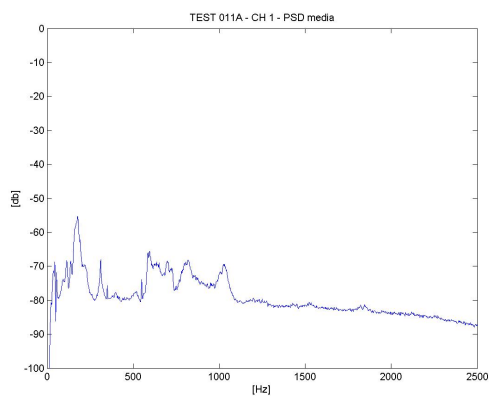
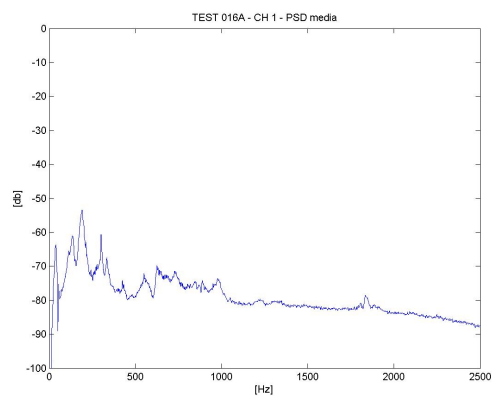


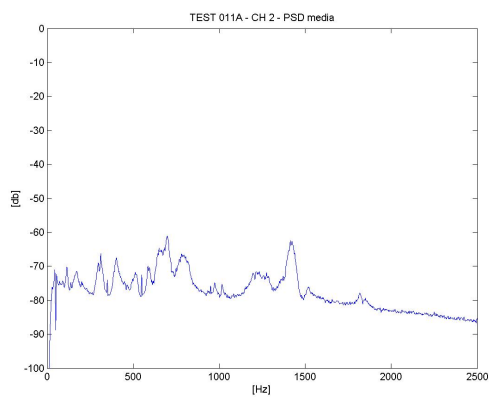
FIGURA 7.21: Svotamento senza vortice in vasca. Grafici con l'andamento del valore *rms*. Il tipico effetto del *drain-sink* non è identificabile.



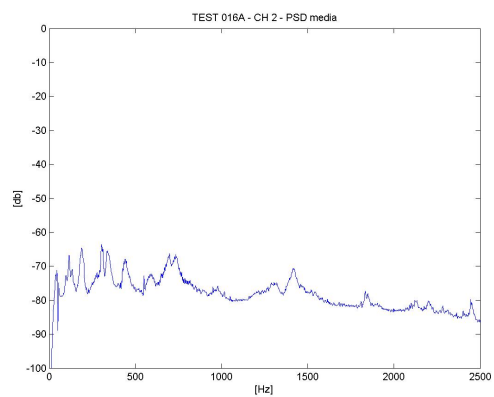
(a)



(b)



(c)



(d)

FIGURA 7.22: Svuotamento senza vortice in vasca. Grafici con la densità di potenza media in funzione della frequenza. La struttura è sempre a banda con un maggior contributo delle bande superiori per quanto riguarda il canale CH2.

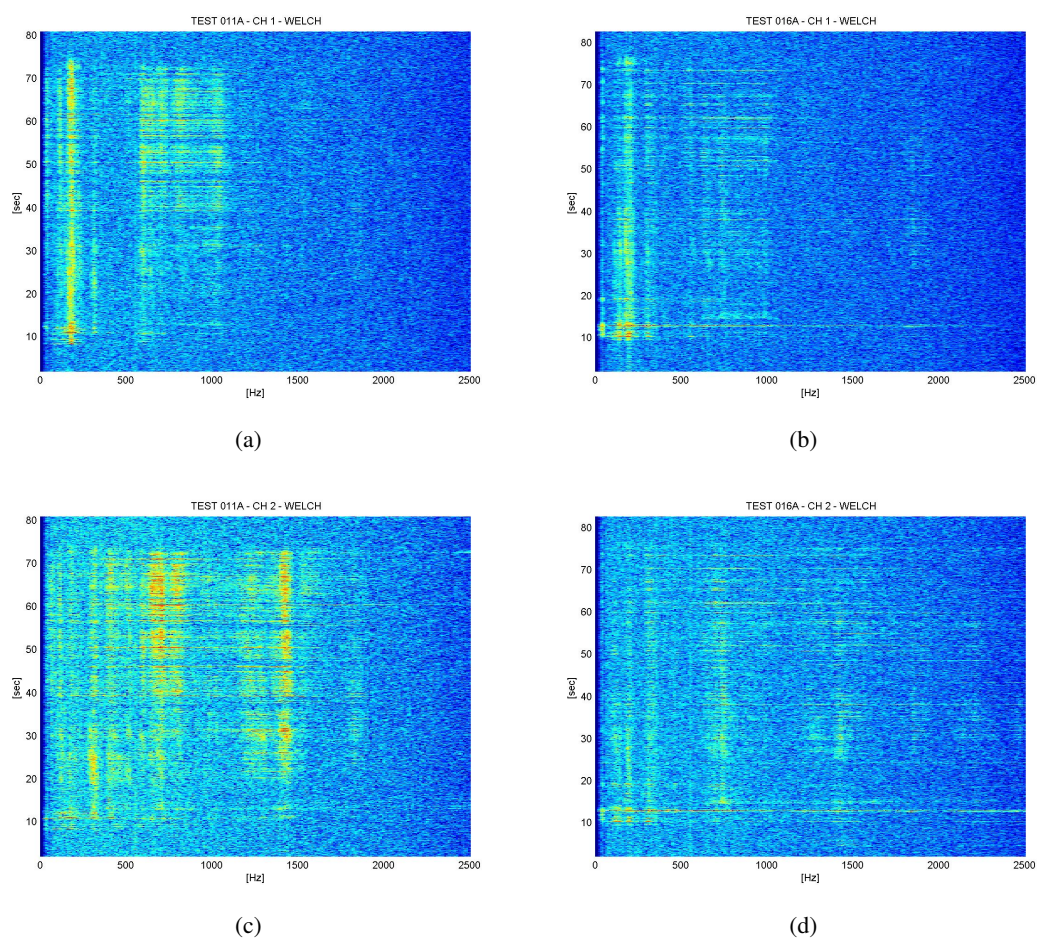


FIGURA 7.23: Svuotamento senza vortice in vasca. Grafici relativi allo spettrogramma. Nessun evento catastrofico è identificabile.

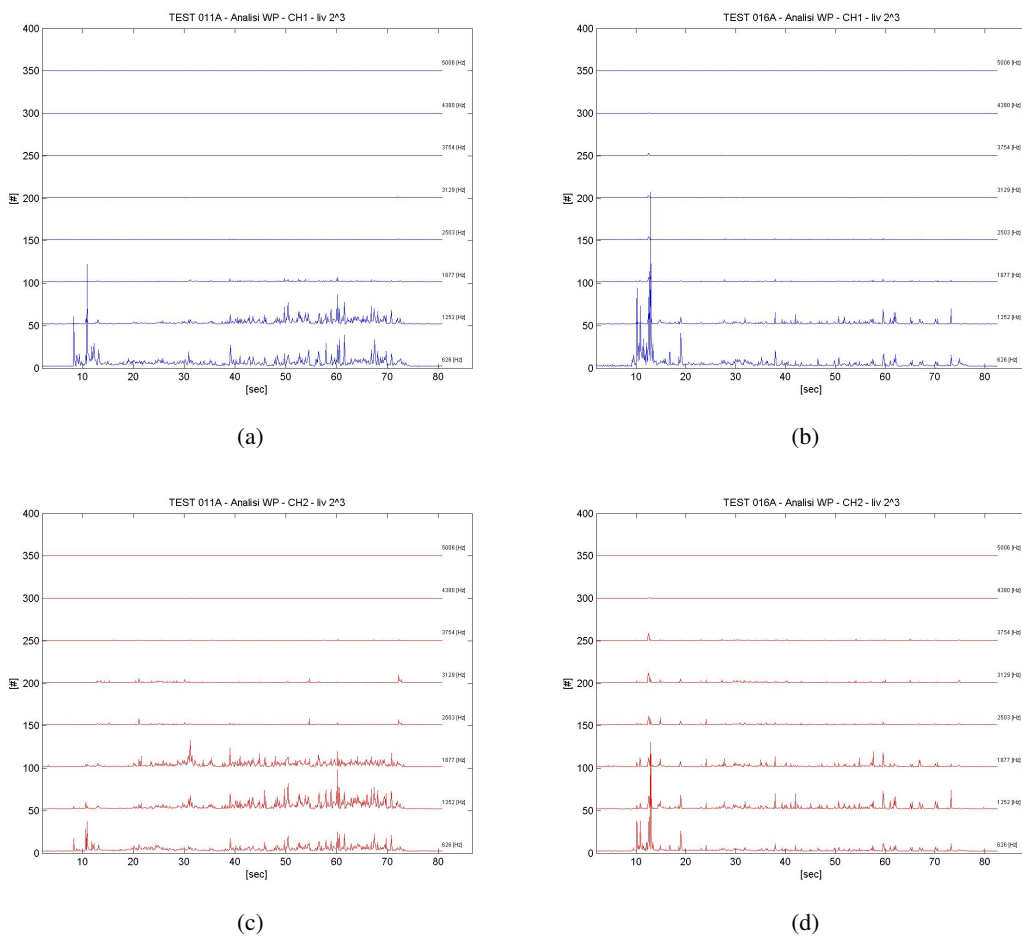


FIGURA 7.24: Svuotamento senza vortice in vasca. Analisi *Wavelet Packets*. Si osserva che si ripete lo schema in cui il canale CH2 mostra un maggior contenuto in frequenza. I grafici (a) e (c) mostrano che l'energia aumenta dopo un certo periodo. Nei grafici (b) e (d) lo stesso comportamento non è identificabile.

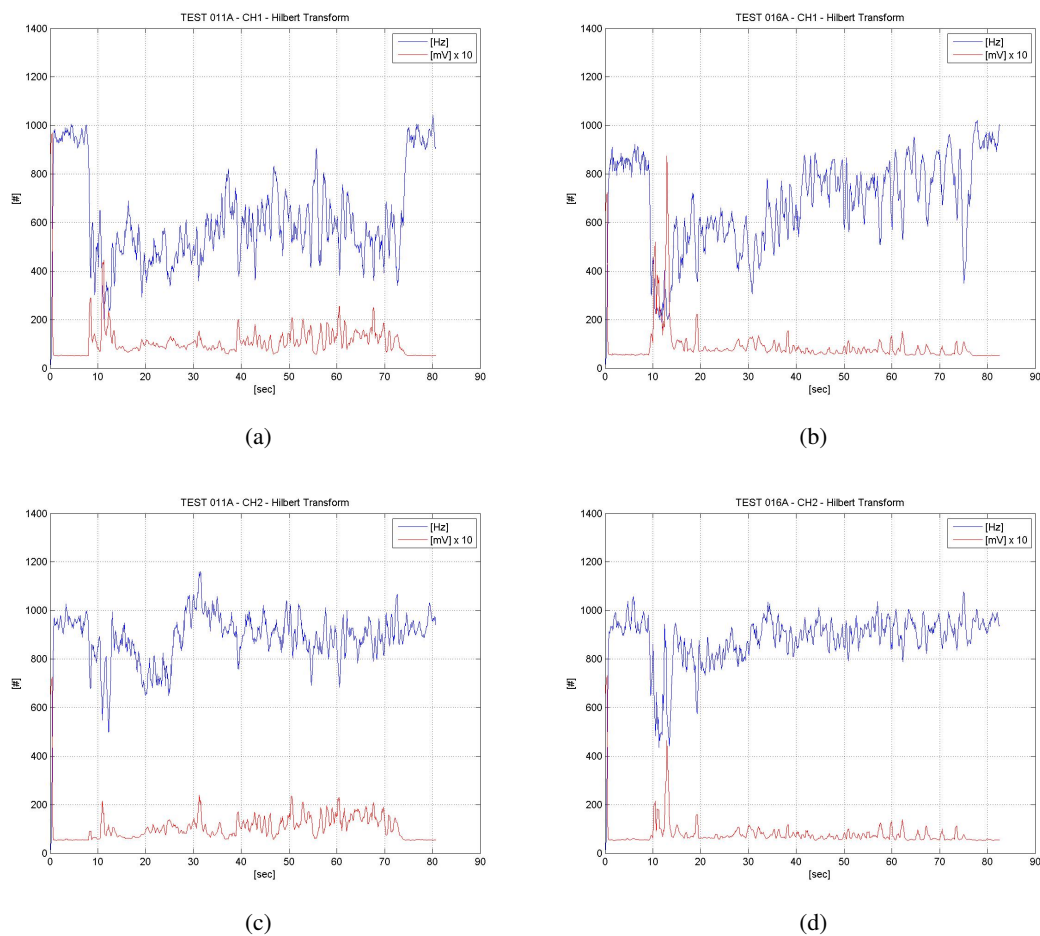
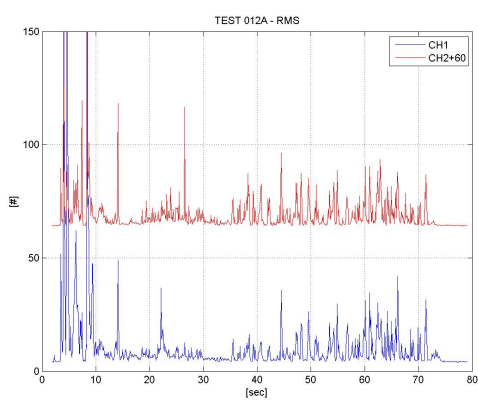
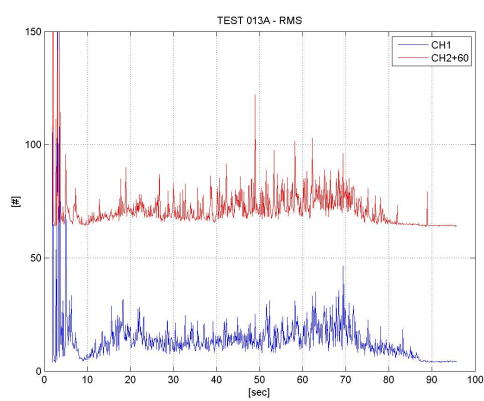


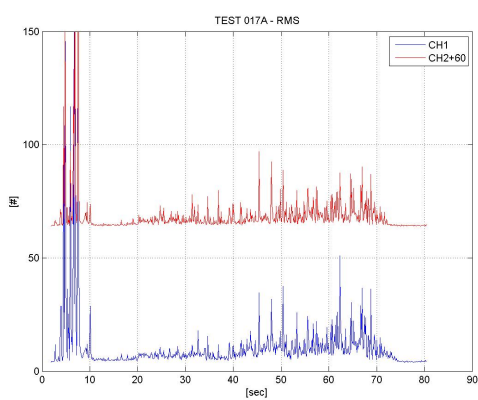
FIGURA 7.25: Svuotamento senza vortice in vasca. Grafici relativi alla trasformata di Hilbert. Nella figura (a) sebbene nessun vortice sia stato registrato l'andamento della frequenza istantanea mostra uno schema simile al caso con vortice. Contrariamente nella figura (b) in questo caso la frequenza istantanea mostra il tipo andamento registrato nelle figure 7.10.



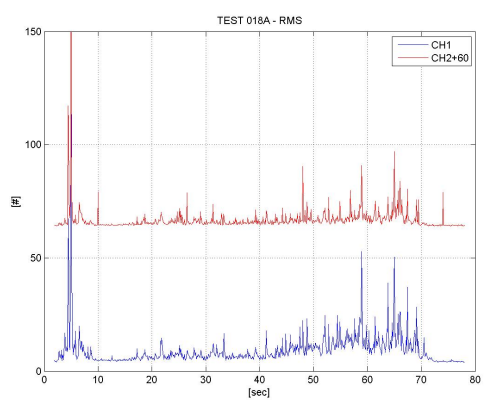
(a)



(b)



(c)



(d)

FIGURA 7.26: Svuotamento con vortice in vasca. Grafici con l'andamento del valore *rms* globale. Le tracce riportate mostrano un andamento simile a quello delle figure 7.12.

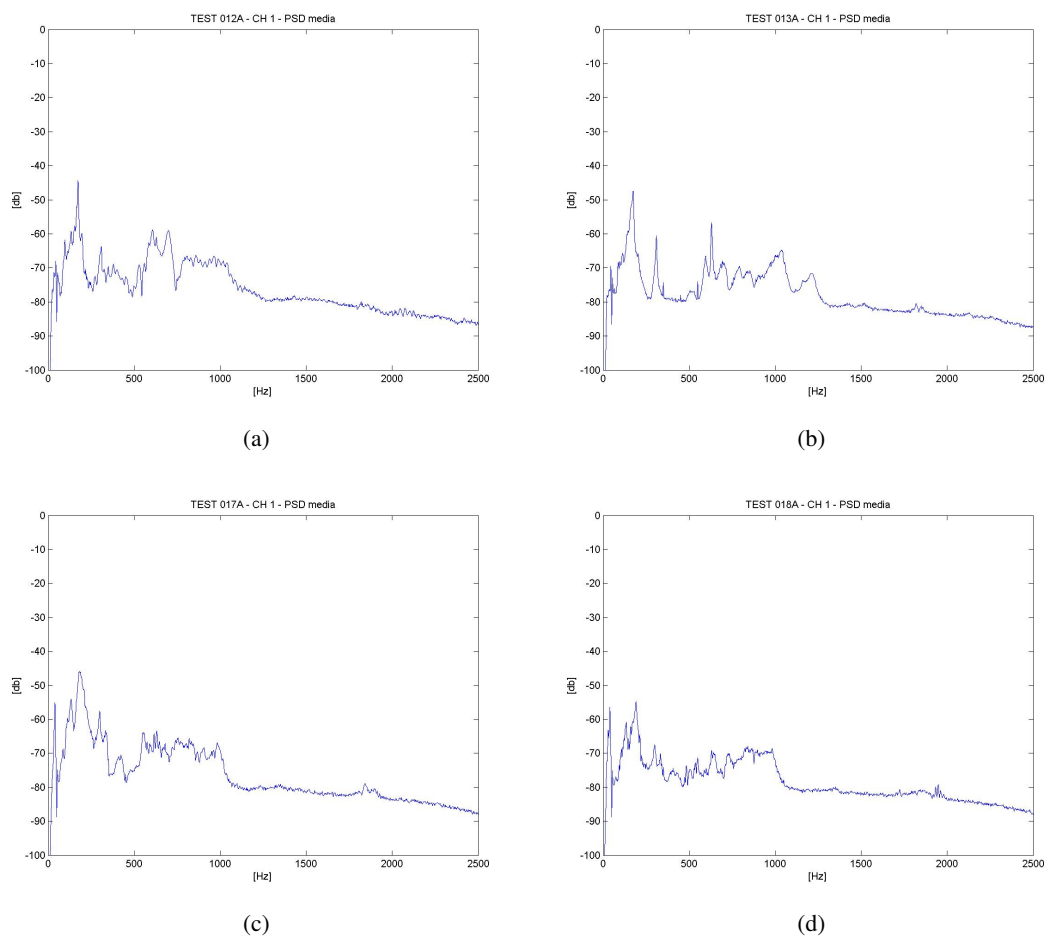
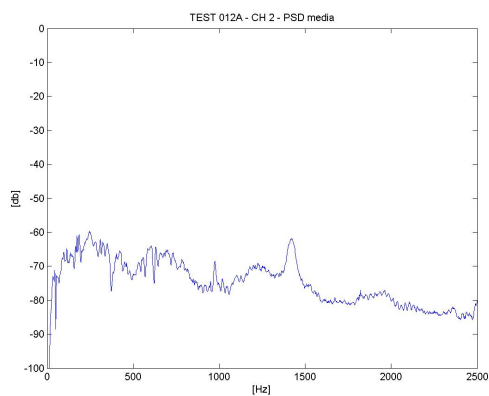


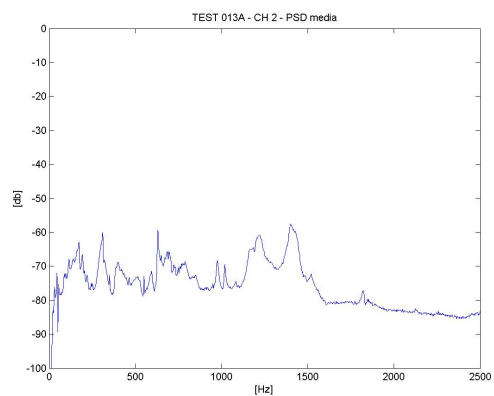
FIGURA 7.27: Svuotamento con vortice in vasca. Grafici con la densità di potenza media in funzione della frequenza. E' presente la tipica struttura a bande con un maggior contributo delle frequenze più alte per il canale CH2.

OSSERVAZIONE SUI VORTICI

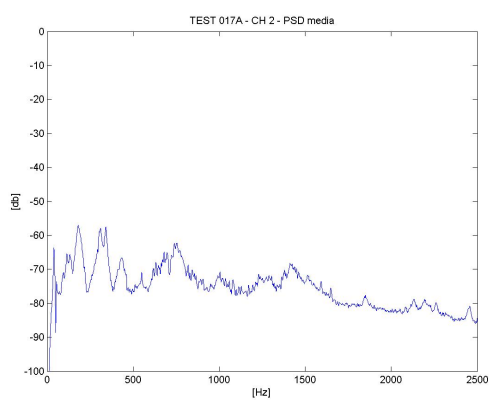
Osservando la superficie del contenitore mentre si sviluppa il vortice, si possono identificare tre fasi. Durante la formazione (fase A) non si notano in modo apprezzabile onde superficiali. Nella fase B invece, la superficie risulta increspata e l'energia del segnale aumenta. Infine durante lo svuotamento nuovamente le onde superficiali scompaiono.



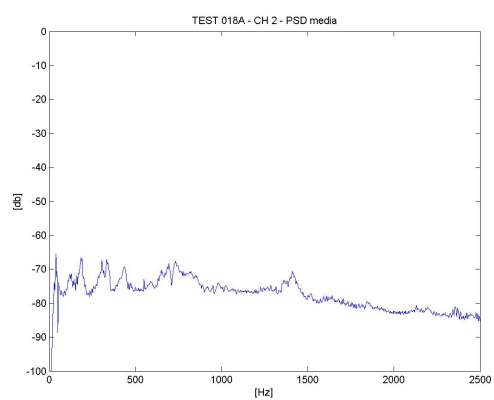
(a)



(b)



(c)



(d)

FIGURA 7.27: Svuotamento con vortice in vasca. Grafici con la densità di potenza media in funzione della frequenza. E' presente la tipica struttura a bande con un maggior contributo delle frequenze più alte per il canale CH2.

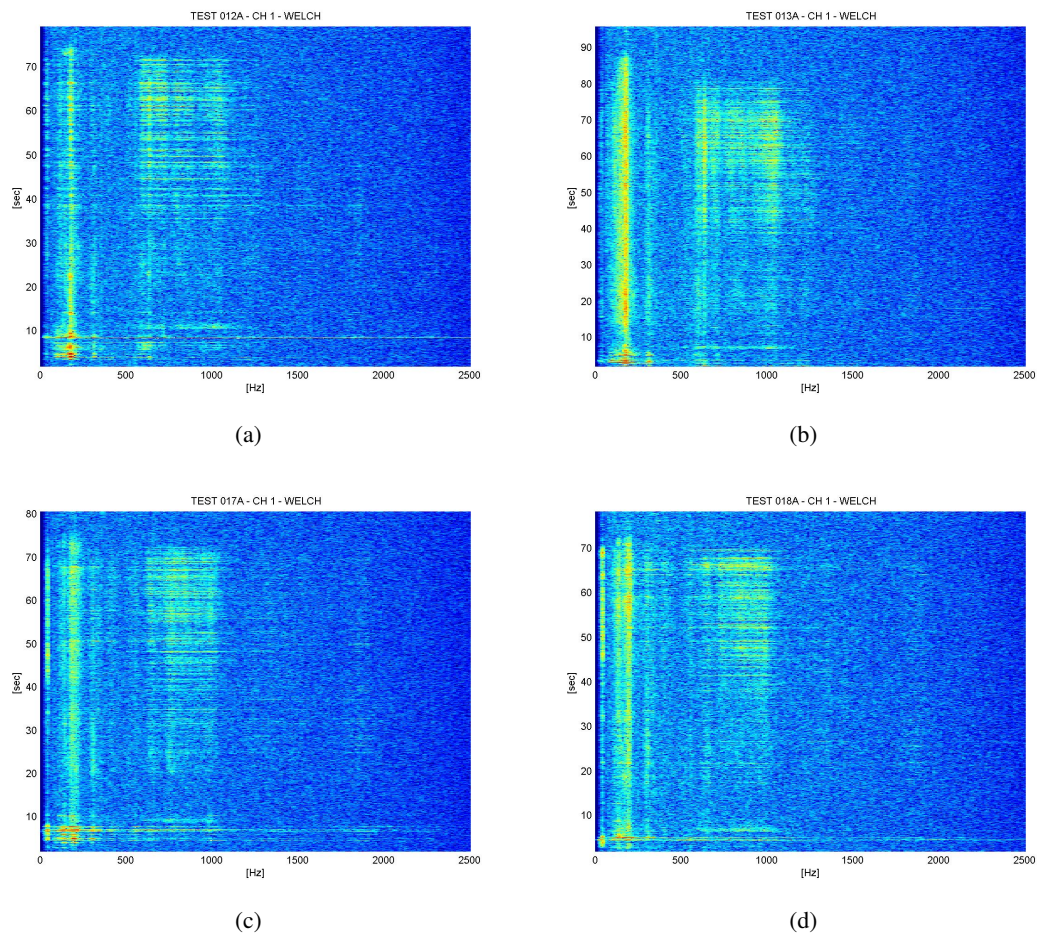


FIGURA 7.28: Svuotamento con vortice in vasca. Grafici relativi allo spettrogramma. Nessun evento catastrofico è visibile. Gli andamenti sono simili al caso con scarico libero.

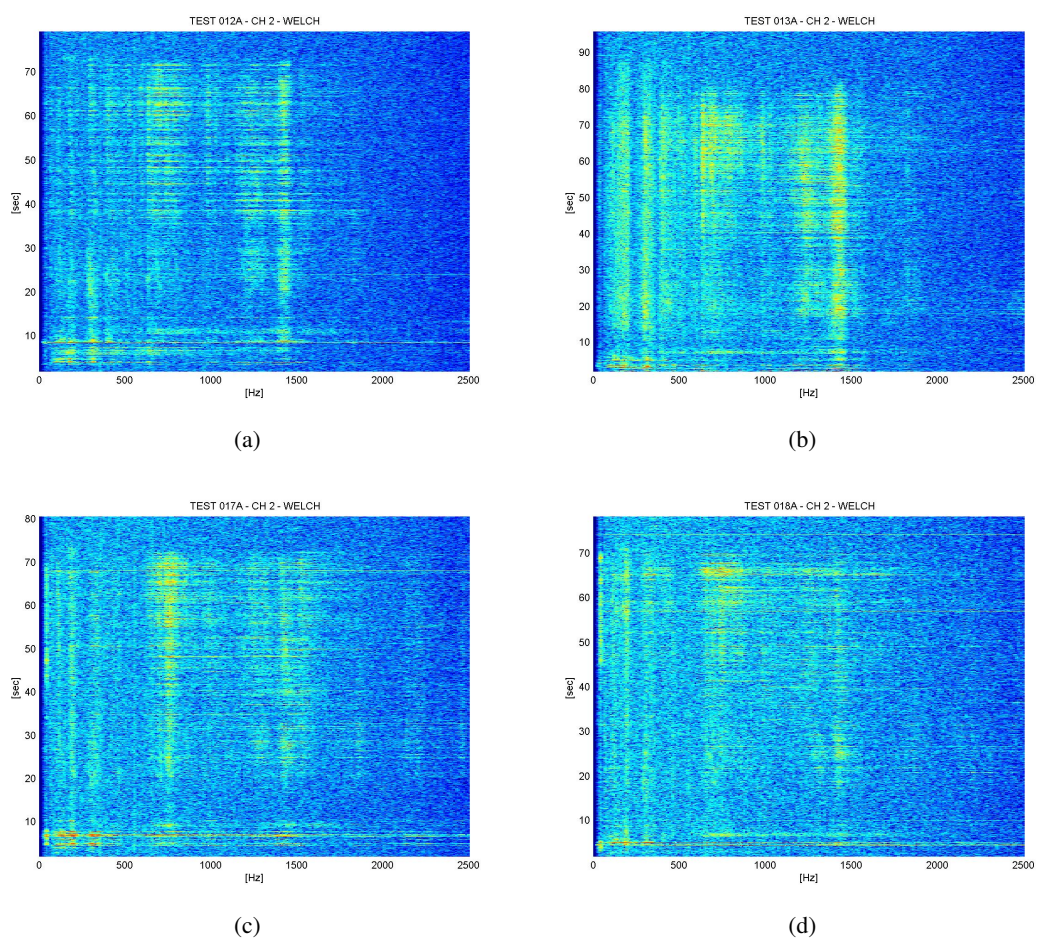


FIGURA 7.28: Svuotamento con vortice in vasca. Grafici relativi allo spettrogramma. Nessun evento catastrofico è visibile. Gli andamenti sono simili al caso con scarico libero.

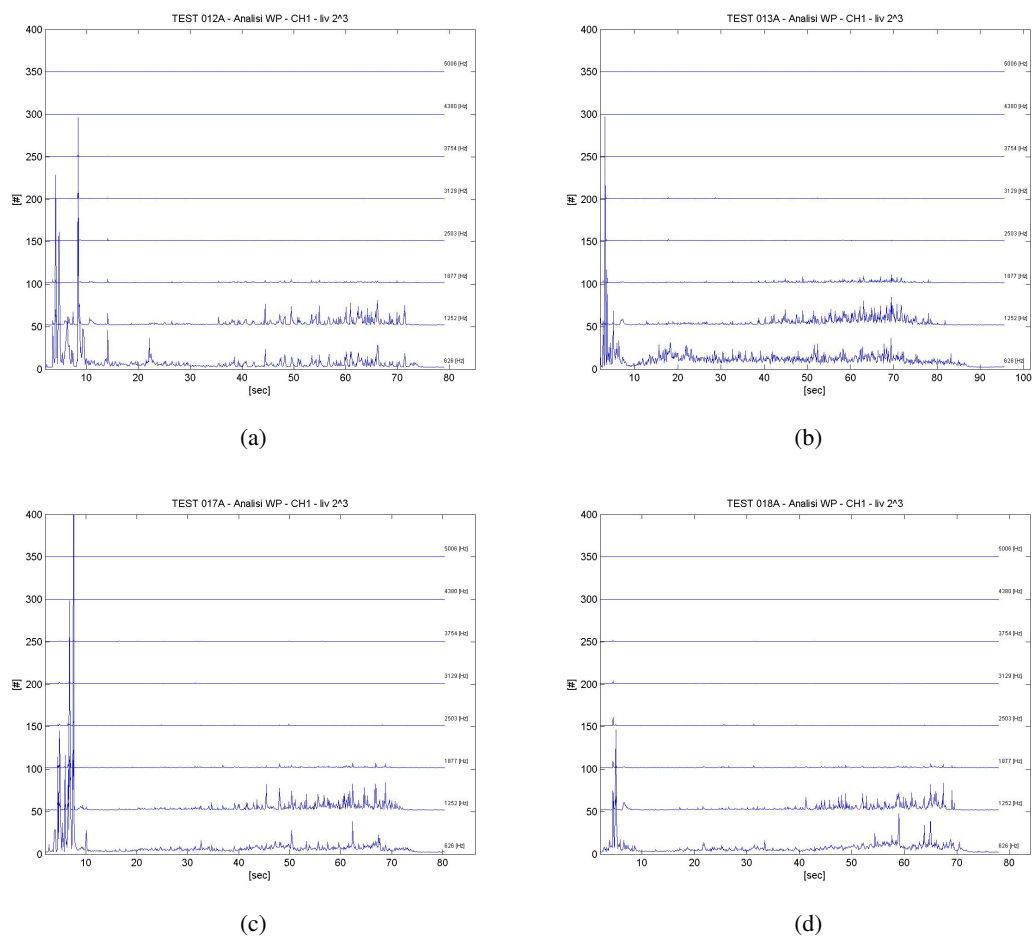


FIGURA 7.29: Svuotamento con vortice in vasca. Analisi Wavelet Packet. E' presente il tipico schema riportato nelle figure 7.17 e 7.18.

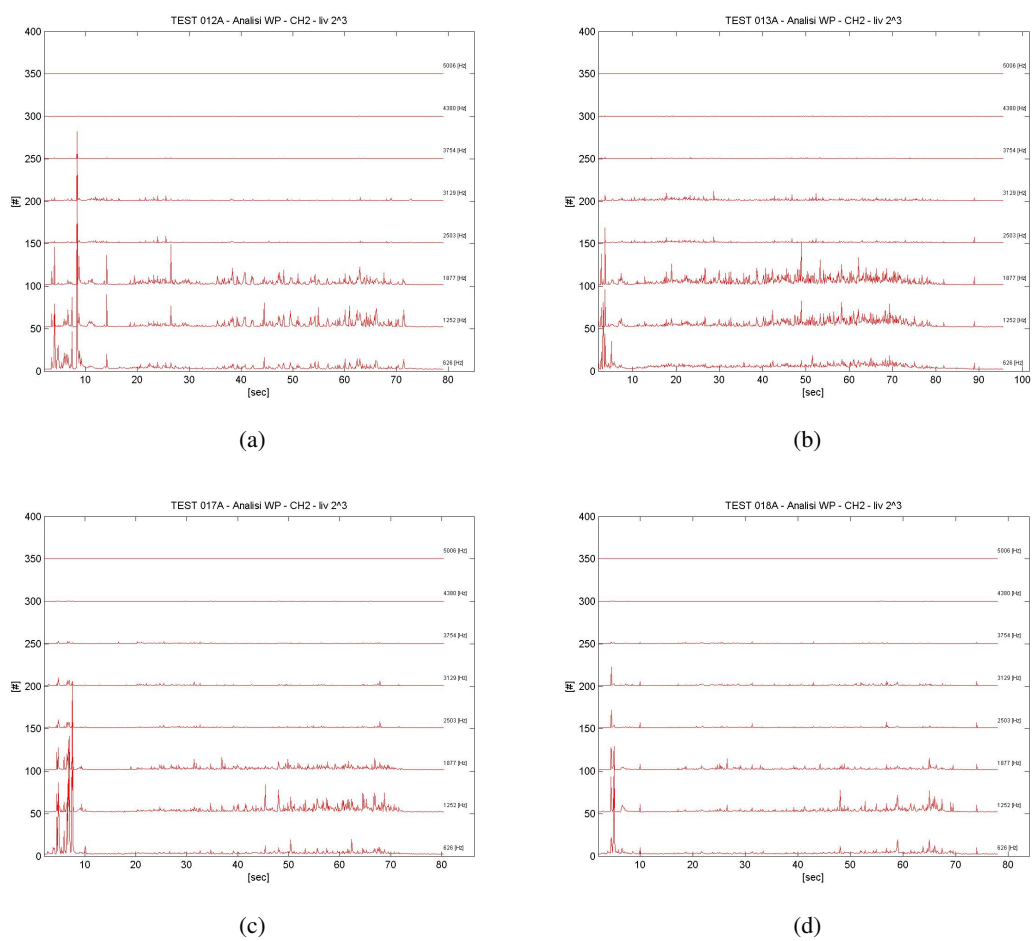


FIGURA 7.29: Svuotamento con vortice in vasca. Analisi Wavelet Packet. E' presente il tipico schema riportato nelle figure 7.17 e 7.18.

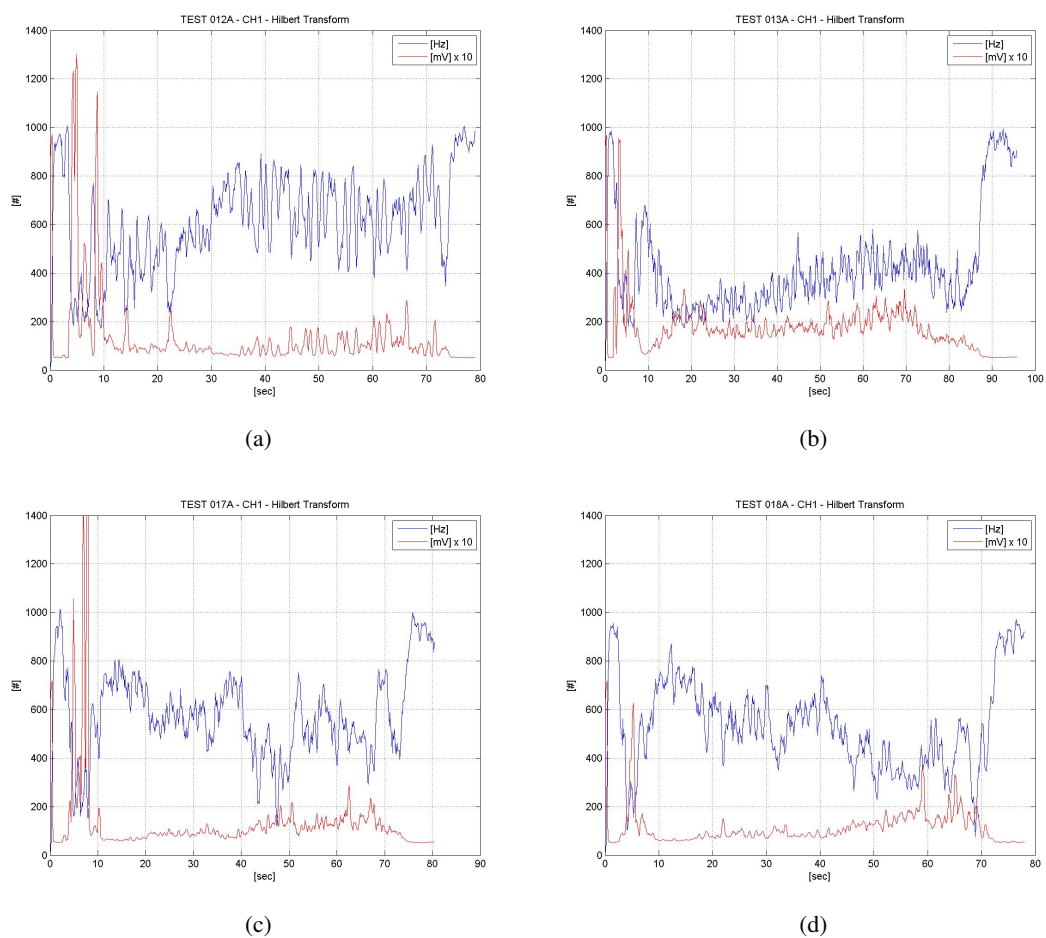


FIGURA 7.30: Svuotamento con vortice in vasca. Trasformata di Hilbert. L'introduzione della vasca perturba lo schema identificato nelle figure 7.19, rendendo meno chiaro il comportamento del canale CH1.

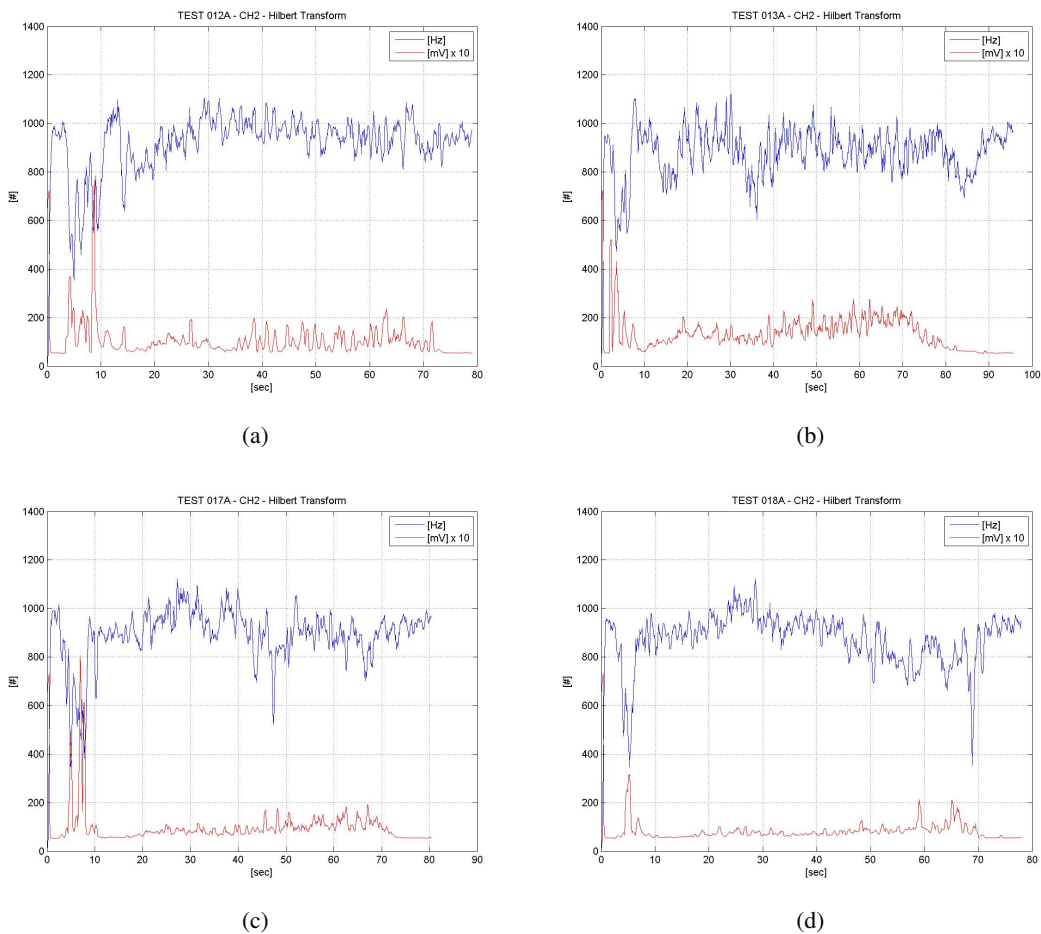


FIGURA 7.30: Svuotamento con vortice in vasca. Trasformata di Hilbert. L'introduzione della vasca perturba lo schema identificato nelle figure 7.19, rendendo meno chiaro il comportamento del canale CH1.

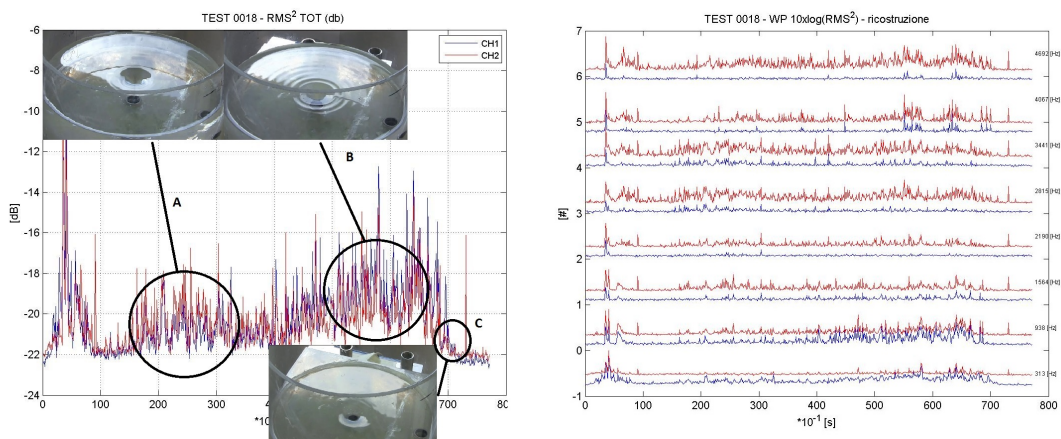


FIGURA 7.31: Correlazione tra potenza del segnale e onde di superficie durante il vortice.

Capitolo 8

Misure in impianto industriale

Oltre ai test in laboratorio su un prototipo in scala ad acqua sono stati eseguiti dei test presso alcuni impianti industriali con l'obiettivo di applicare gli algoritmi in condizioni reali e valutarne l'efficacia rispetto all'azione umana. Chiaramente a differenza degli esperimenti in laboratorio l'attrezzatura in campo deve essere particolarmente robusta. Ho potuto eseguire tre diverse campagne di misura in altrettanti impianti aventi tutti una struttura simile del braccio porta - tuffante. I risultati ottenuti sono in effetti del tutto equivalenti.

8.1 Apparato sperimentale in campo.

L'apparecchiatura utilizzata sebbene finalizzata ad un test è composta da elementi appositamente progettati per l'ambiente industriale. Come si può osservare nella figura 8.1a l'accelerometro è posto all'interno di una struttura in acciaio inox, nella quale è inserito anche un misuratore di temperatura. Il sensore viene solidamente installato sul braccio porta tuffante (figura 8.1b) e collegato attraverso un cavo opportunamente protetto ad un quadro elettrico (figura 8.2a). All'interno del quadro (figura 8.2b) è inserita la scheda elettronica per la digitalizzazione del segnale e la scheda di gestione dei segnali ausiliari. A questa unità è connessa anche una interfaccia hardware (figura 8.2c) comprendente una sirena ed un lampeggiante per facilitare il confronto con l'azione dell'operatore. Infine attraverso un cavo ethernet le unità elettroniche sono interfacciate ad un computer su cui è installato il software descritto nel capitolo dedicato.

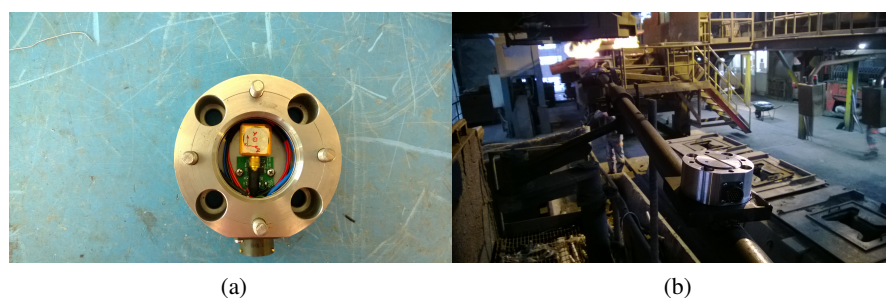


FIGURA 8.1: Installazione.

Parametro	Valore
Numero di assi	3
Sensibilità	1000 mV/g
Range di frequenze	0.3-5000 Hz
Frequenza di risonanza	> 20 kHz
Densità spettrale di rumore	0.4 -11 μ g/ \sqrt{Hz}

TABELLA 8.1: Principali caratteristiche dell'accelerometro.

8.2 Analisi mediante metodi di stima spettrale

L'analisi del segnale registrato dagli accelerometri deve basarsi su due vincoli fondamentali. Il primo riguarda gli aspetti hardware. Come è stato descritto nel paragrafo precedente l'acquisizione dei sensori avviene attraverso un dispositivo che invia su ethernet i pacchetti dati ad un computer. Un software dedicato dovrà leggere il pacchetto, salvarlo su disco rigido, elaborarlo ed infine visualizzarlo. Il tempo di ciclo ottimale per un'attrezzatura di questo dovrà essere dell'ordine del 100 ms. L'altro vincolo deriva invece dal fenomeno fisico. L'obiettivo dell'analisi infatti deve essere l'individuazione del momento in cui avviene il passaggio scoria. Questo fenomeno si sviluppa con tempi di qualche secondo. Occorre quindi valutare se gli algoritmi impiegati sono tali da poter essere applicati in questo contesto. Come accennato in precedenza tra diversi impianti non si notano significative differenze. All'interno di uno stesso ciclo produttivo si possono invece apprezzare situazioni diverse, dovute al momento a variabili non identificate. In effetti possono sicuramente intervenire fattori come il tipo di acciaio, l'usura del materiale refrattario e la forma del fondo siviera. In totale sono state fatte i seguenti test:

- Impianto N°1: 9 test.
- Impianto N°2: 6 test.
- Impianto N°3: 17 test.

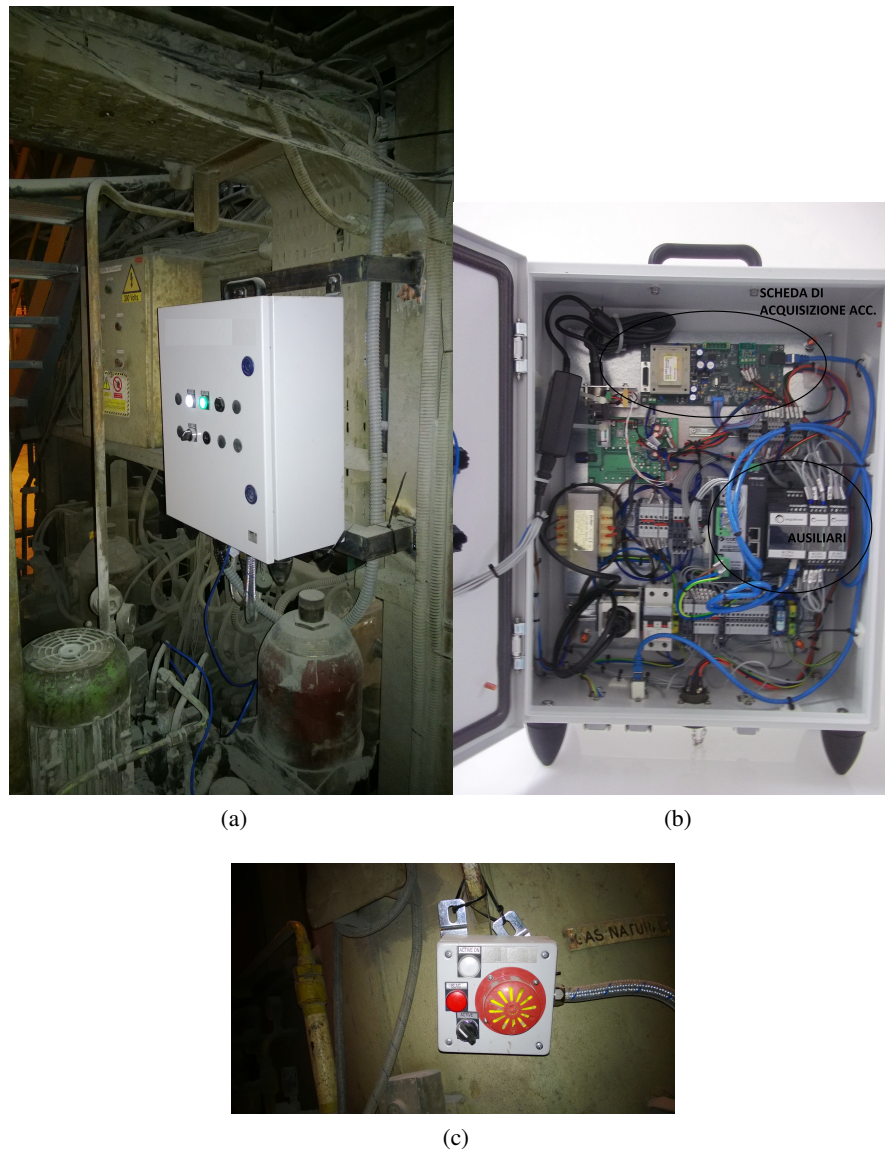


FIGURA 8.2: Quadro elettrico con unità elettronica e sistema opto-acustico di segnalazione.

Nell'esposizione che seguirà saranno riportati i casi più significativi, poiché lo scopo è quello di valutare l'efficacia delle tecniche nelle condizioni meno favorevoli. La strategia di analisi quindi si svolgerà in questo modo:

- analisi globale del segnale;
- analisi della densità spettrale di potenza;
- analisi specifiche con tecniche Wavelet ed Hilbert.

Il segnale grezzo proveniente dagli accelerometri è riportato in figura 8.3a, esso è caratterizzato da un andamento apparentemente stazionario con sovrapposti gli impulsi dovuti alla movimentazione del cassetto. Come punto di partenza è opportuno valutare il valore rms dei segnali.

Quest'analisi sebbene elementare permette di apprezzare una delle caratteristiche fondamentali ovvero il fatto che tendenzialmente la scoria essendo meno densa tende a ridurre il livello di potenza registrato dall'accelerometro. L'immagine 8.3b riporta proprio un caso molto favorevole, in cui l'evento di passaggio scoria sembra essere determinato da un calo del valore di potenza del segnale che avviene in modo marcato a 289 secondi, con successivo intervento dell'operatore. E' interessante anche notare qualora il cassetto siviera venga aperto il livello rms del segnale sale. In effetti all'incirca a 272 secondi l'operatore provvede ad aumentare l'area dello scarico della siviera per mantenere il livello costante in paniera, immediatamente si nota che l'energia aumenta. Tutti e tre gli assi dell'accelerometro sembrano avere un comportamento simile. Questa caratteristica è effettivamente abbastanza costante. Nel test numero due (figura 8.3c) invece si registra un segnale con una varianza maggiore, ma l'aspetto più importante è che la zona di transizione tra acciaio e scoria (figura 8.3d) appare molto meno definita. In particolare nei canali 2 (CH2) e 3 (CH3) essa sembra inesistente, mentre a 297 secondi l'operatore chiude la siviera avendo individuato otticamente il passaggio della scoria. Applicando il calcolo del segnale rms su buffer di lunghezza diversa non appaiono significative variazioni: infatti variando la durata del singolo buffer da 100 ms sino a 500 ms non ci sono differenze salvo una riduzione della varianza. Aumentando ulteriormente invece si osserva un eccessivo appiattimento della zona di transizione. Applicando questa semplice analisi a tutti i dati raccolti si possono identificare le seguenti caratteristiche in termini di potenza globale del segnale:

- la varianza del segnale varia secondo variabili al momento non identificate.
- I valori assoluti di rms dipendono dalle condizioni di colata secondo regole non identificate. Essi sono contenuti nell'intervallo 5-50 mV.
- Gli eventi impulsivi dovuti alla movimentazione del cassetto durano meno di 1 secondo e cambiano il livello di potenza del segnale.
- Il passaggio della scoria avviene in tempi brevi e precisamente in un intervallo di circa 2-3 secondi.
- Il passaggio della scoria generalmente si manifesta con una diminuzione della potenza del segnale.
- Non tutti gli assi rispondono in modo analogo anche se tendenzialmente il comportamento è molto simile.
- La zona immediatamente precedente al passaggio della scoria non presenta caratteristiche peculiari.

Le osservazioni sopra esposte non dipendono inoltre dalla posizione dell'accelerometro lungo l'asta porta tuffante. L'altra analisi elementare che può essere fatta si basa sulla trasformata

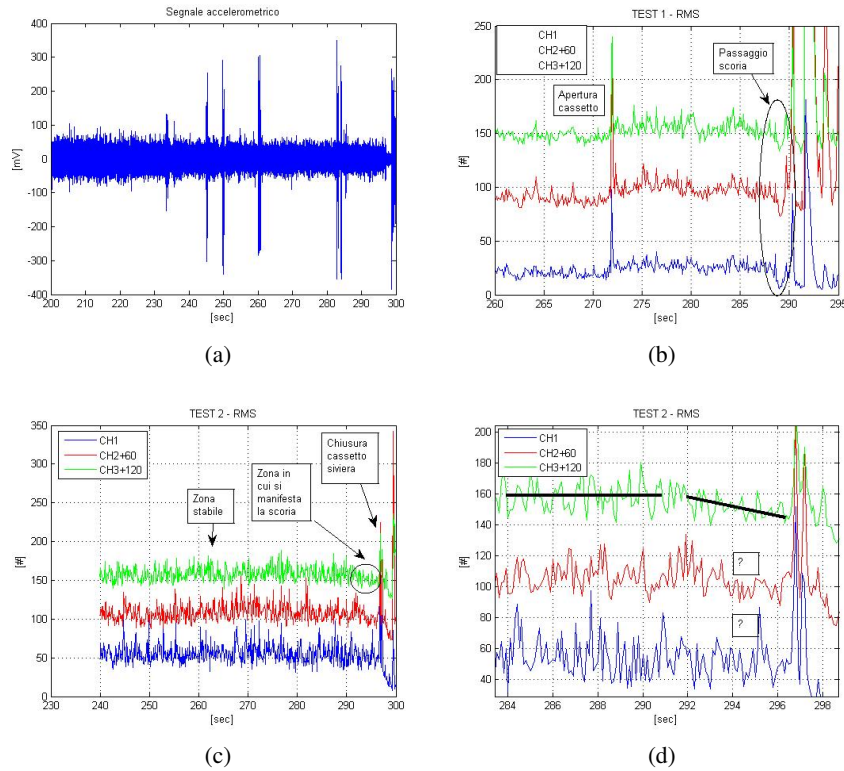


FIGURA 8.3: (a) Segnale tipico registrato dall'accelerometro, si notano i picchi impulsivi dovuti alla movimentazione del cassetto. (b) Valori rms relativi al test identificato come numero 1. Il canale CH1 ha un valore indicativo di circa 25 mV. (c) Valori rms relativi al test identificato come numero 2. (d) Valori rms della zona di passaggio scoria del test numero 2. Si nota che gli assi CH2 e CH3 non mostrano un chiaro andamento.

di Fourier. Sebbene il segnale non sia stazionario tuttavia considerando ugualmente dei valori medi si possono identificare alcune caratteristiche globali. Nella figura 8.4a sono riportate le trasformate di Fourier relative ai tre canali del test numero 2. L'andamento degli spettri non presenta un comportamento identico, anche se per tutti gli assi le ampiezze si attenuano oltre i 600 Hz. Questa in effetti è una caratteristica generale e sicuramente è dovuta al fatto che le aste porta-tuffante sono molto simili. La figura 8.4d permette di proprio di verificare questa ipotesi infatti in tutti e tre gli impianti il range delle frequenze sostanzialmente coincide. Nella figura 8.4b invece si può osservare che le ampiezze delle trasformate non coincidono esattamente tra colate successive. Per ricavare i precedenti grafici il segnale registrato è stato diviso in buffer da $[n]$ campioni ciascuno, per ogni buffer è stata calcolata la FFT:

$$F_i(\omega)^{[n]} = \text{FFT}(\text{buffer}_i^{[n]}) \quad i = 1, \dots, N_{\text{buffers}^{[n]}}. \quad (8.1)$$

Successivamente è stata calcolato il valore medio dell'ampiezza per ogni frequenza:

$$\langle F_i(\omega)^{[n]} \rangle = \frac{1}{N_{\text{buffers}^{[n]}}} \sum_{i=1}^{N_{\text{buffers}^{[n]}}} |F_i(\omega)^{[n]}|. \quad (8.2)$$

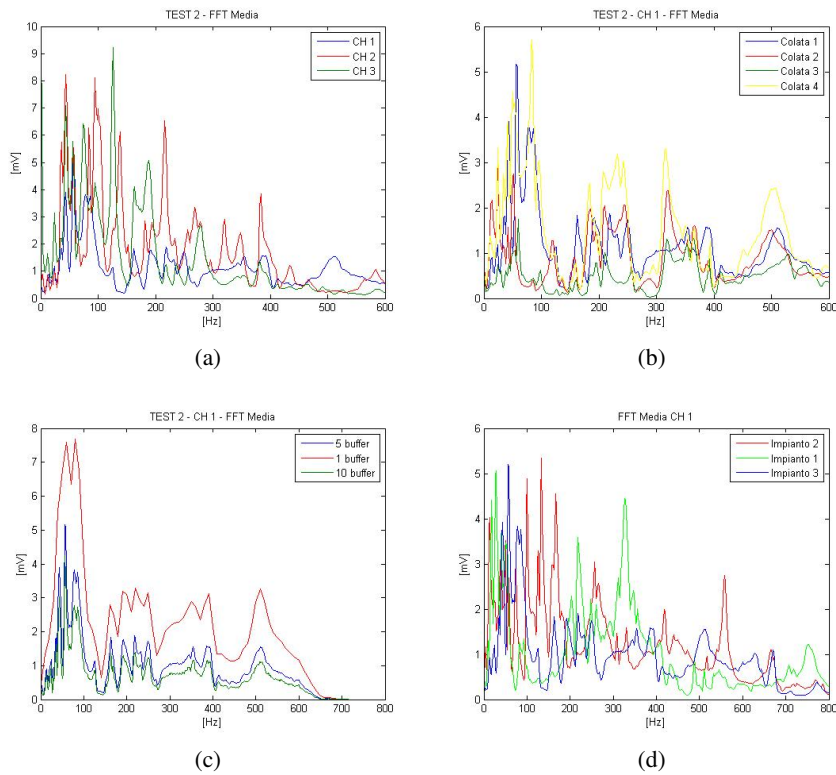


FIGURA 8.4: Analisi di Fourier. (a) Confronto tra i valori medi del test numero 2 relativi ai tre canali. (b) Confronto tra i valori medi relativi al canale 1 di diverse colate in uno stesso impianto. (c) Confronto tra i valori medi variando il numero di buffer. (d) Confronto tra i valori medi per i tre impianti.

I pacchetti dati vengono spediti ogni 100 ms, pertanto l'analisi in frequenza avrà una granularità pari a 10 Hz. Per aumentare la risoluzione è necessario costruire dei buffer di durata maggiore congiungendo pacchetti dati contigui. Nella figura 8.4c possiamo osservare che all'aumentare di n la risoluzione aumenta e che già con 5 pacchetti dati si raggiunge una granularità sufficiente ad apprezzare l'andamento qualitativo dello spettro. Analogamente al caso delle ampiezze è stato

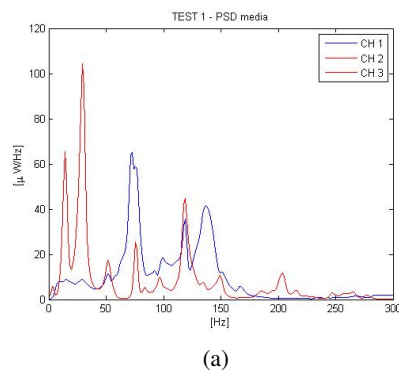


FIGURA 8.5: Analisi della densità di potenza media.

fatto un calcolo della densità di potenza media,

$$P_i(\omega)^{[n]} = \text{PSD}_{\text{WELCH}}(\text{buffer}_i^{[n]}) \quad i = 1, \dots, N_{\text{buffers}^{[n]}}. \quad (8.3)$$

Tra i vari metodi elencati nel capitolo 2 ho utilizzato quello di *Welch* con una finestra di *Hanning*. Il valore medio della PSD per ogni frequenza sarà:

$$\langle P_i(\omega)^{[n]} \rangle = \frac{1}{N_{\text{buffers}^{[n]}}} \sum_{i=1}^{N_{\text{buffers}^{[n]}}} P_i(\omega)^{[n]}. \quad (8.4)$$

Come si può osservare dalla figura 8.5a, che rappresenta un andamento tipico, in termini di potenza il contenuto in frequenza è compreso entro i primi 200 Hz. Inoltre si osservano delle bande piuttosto strette. Riassumendo quindi in termini armonici possiamo concludere quanto segue

- gli spettri di ampiezza non sono costanti durante colate successive;
- per una data tipologia di barre il contenuto significativo in frequenza risulta compreso in un intervallo inferiore a 600 Hz. In termini di potenza l'intervallo si restringe.
- Gli spettri di potenza mostrano una forma con picchi ben definiti.
- I picchi di potenza non sono invarianti per asse spaziale, sequenza di colata o impianto.

L'analisi generale sin qui riportata, permette di affermare che la potenza del segnale rappresenta un indicatore che deve essere tenuta in debita considerazione. Inoltre appare evidente che non emergono caratteristiche dipendenti in modo significativo dall'orientazione degli assi dell'accelerometro. Dalle registrazioni effettuate infatti le eventuali differenze tra i tre assi assumono un comportamento stocastico. L'asse che in alcuni casi sembra contenere un'evidenza maggiore del fenomeno relativo al passaggio scoria, non rappresenta un invariante. In effetti in termini del tutto casuali gli assi possono presentare un comportamento diverso l'uno dall'altro. Per esplorare in modo più approfondito il segnale il passo successivo consiste nell'eseguire un'analisi tempo-frequenza. In particolare mi soffermerò sulla serie di dati in cui risulta più arduo identificare il passaggio della scoria osservando solo il valore *rms* globale. In queste condizioni infatti si può apprezzare la potenza delle tecniche di analisi avanzata utilizzate in questo contesto. Nella discussione seguente la serie di dati analizzata presenta in apparenza maggiore informazione sul canale numero 3, però ribadisco che questa caratteristica non è peculiarità di tale canale. Per ogni buffer temporale di lunghezza $[n]$ è necessario calcolare la densità di potenza

$$P_i(\omega, t_i) = \text{PSD}_{\text{WELCH}}(\text{buffer}_i^{[n]}) \quad i = 1, \dots, N_{\text{buffers}^{[n]}}. \quad (8.5)$$

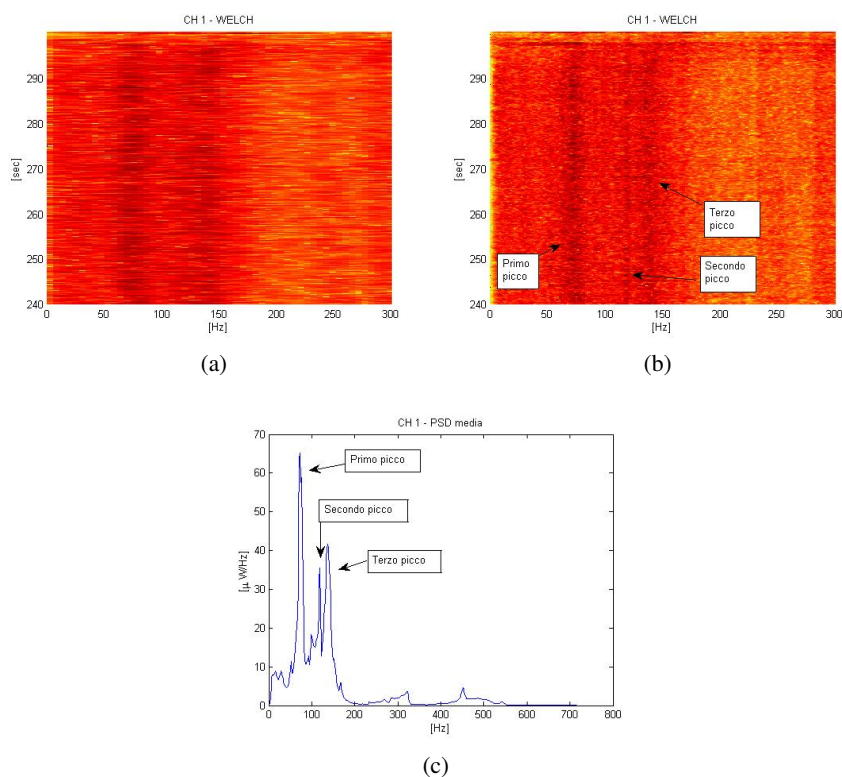


FIGURA 8.6: Analisi di PSD tempo-frequenza. (a) Spettrogramma con metodo *Welch* su singolo buffer. (b) Spettrogramma su 5 buffer contigui. (c) PSD media.

I valori $P_i(\omega, t_i)$ possono essere rappresentati in un grafico bidimensionale. La scelta della lunghezza del buffer risulta estremamente critica. Come si può osservare dalla figure 8.6a, 8.6b e 8.6c, la scelta di un intervallo di tempo non sufficiente non permette di mettere in risalto tutti i principali picchi. Nella figura 8.6a per il calcolo della PSD non è stato considerato un numero sufficiente di punti e la risoluzione in frequenza risulta non adeguata per cogliere la struttura delle bande. Nel grafico 8.6b l'aumento di $[n]$ permette di apprezzare la granularità: questo fatto tuttavia riduce la risoluzione temporale, come evidenziato nel capitolo dedicato alla teoria dell'analisi spettrale. Dai medesimi grafici si evince una caratteristica molto importante: le frequenze dominanti si mantengono inalterate fino alla zona in cui si manifesta il passaggio della scoria. Le figure 8.7a, 8.7b e 8.7c mettono in confronto il segnale rms con quello dello spettrogramma applicando il metodo di *Welch*. In effetti risulta particolarmente arduo identificare l'evento del passaggio scoria. Si nota comunque che le bande in frequenza interessate dal fenomeno, apparentemente permangono anche durante la fase di svuotamento. Osservando gli spettrogrammi con una vista tridimensionale si riesce ad apprezzare leggermente meglio la dinamica dell'evento. Le figure 8.8a, 8.8b e 8.8c permettono di riconoscere con che durante il passaggio scoria alcune bande tendono a spegnersi mentre altre accentuano il proprio contenuto energetico. Questo schema, in termini generali, si ripete indipendentemente dal tipo di colata o di impianto. Naturalmente non si ripresenta esattamente allo stesso, infatti l'andamento della potenza delle bande nel corso dei vari test non ha rivelato uno schema fisso. La particolare

struttura risonante dei grafici relativi alla densità di potenza suggerisce di utilizzare i modelli autoregressivi introdotti nel capitolo 2. Nella figura 8.9a è riportato lo spettrogramma ottenuto con un modello AR di ordine 30. Rispetto al metodo *Welch* l'andamento delle bande principali risulta molto ben definito e l'evento scoria appare più chiaramente identificabile. Nelle figure 8.9b e 8.9c ho riportato la stessa sequenza dati analizzata sia con il metodo della massima entropia (*Burg*) che con quello agli autovalori. I risultati sono simili, con tutti gli approcci si possono osservare distintamente le bande interessate.

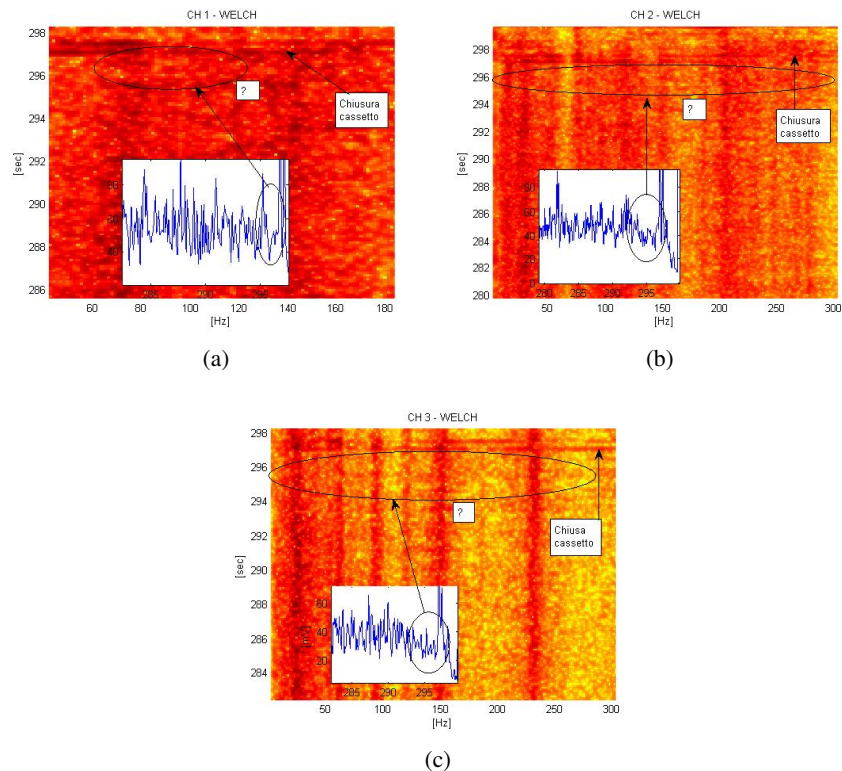


FIGURA 8.7: Spettrogramma con metodo di *Welch* relativo all'evento del passaggio scoria. Nei riquadri più piccoli è riportato il valore rms. L'identificazione risulta più ardua sui diagrammi bidimensionali.

La scelta ottimale del modello autoregressivo può essere fatta studiando l'andamento della varianza dell'errore di predizione in funzione dell'ordine N .

$$\mathbf{e}^{(N)}(n) = \mathbf{x}(n) - \hat{\mathbf{x}}^{(N)}(n) = \sum_{k=1}^N a_k^{(N)} x(n-k), \quad (8.6)$$

$$\text{var}\{\mathbf{e}^{(N)}(n)\} = \sum_{i=1}^M \frac{(\mathbf{e}^{(N)}(n))^2}{M} \quad (8.7)$$

Nella figura 8.10a è graficato l'andamento della varianza $\text{var}\{\mathbf{e}^{(N)}(n)\}$ al variare di N , in effetti l'andamento si stabilizza all'incirca dopo l'ordine 30. Per un modello AR si può dimostrare che

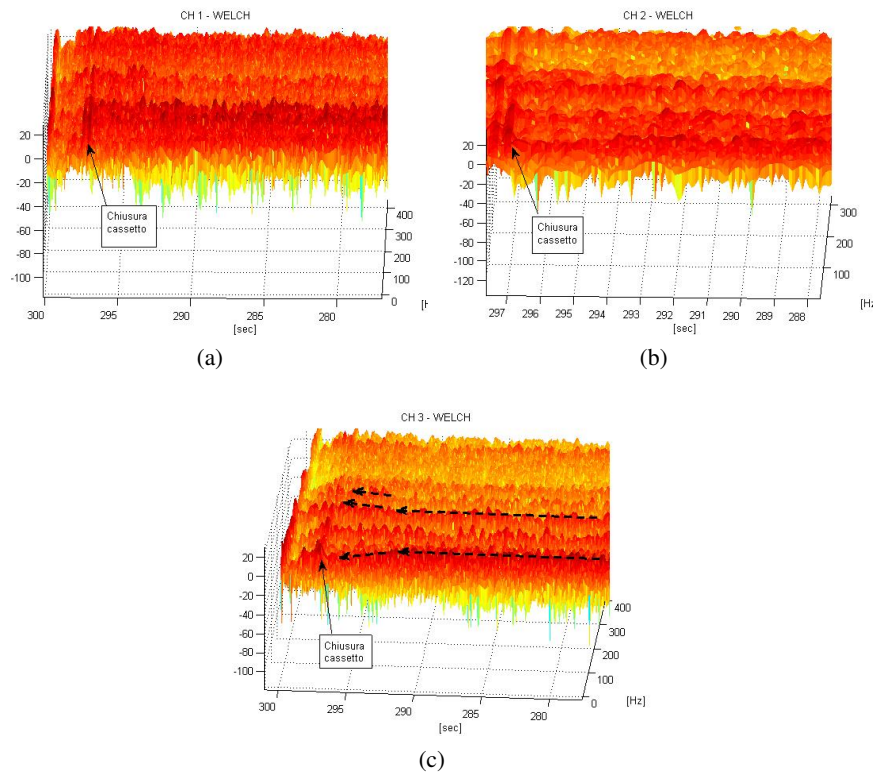


FIGURA 8.8: Spetrogramma con metodo di *Welch* relativo all'evento del passaggio scoria. Vista tridimensionale. I grafici (a), (b) e (c) si riferiscono ai tre assi dell'accelerometro. Il canale numero 3 è quello più rappresentativo della dinamica del fenomeno.

[57] l'errore di predizione è una funzione monotona decrescente, dell'ordine del modello:

$$e^{(N)}(n) \geq e^{(N+1)}(n) \geq e^{(N+2)}(n) \dots \quad (8.8)$$

In particolare all'ordine ottimo, l'errore non decresce più e si raggiunge la condizione di *stallo* $e^{(N)}(n) = e^{(N+1)}(n) = e^{(N+2)}(n) \dots$. Con dati reali esistono vari criteri [18] che possono essere utilizzati per individuare l'ordine corretto del modello, evitando l'*overfitting*:

- *Partial correlation analysis* (PAC).
- *Final prediction error* (FPE).

$$FEP[p] = \text{var}\{e^{(p)}(n)\} \left(\frac{N+p+1}{N-p+1} \right) \quad (8.9)$$

- *Akaike information criterion* (AIC).

$$AIC[p] = N \ln(\text{var}\{e^{(p)}(n)\}) + 2p \quad (8.10)$$

- *Minimum description length* (MDL).

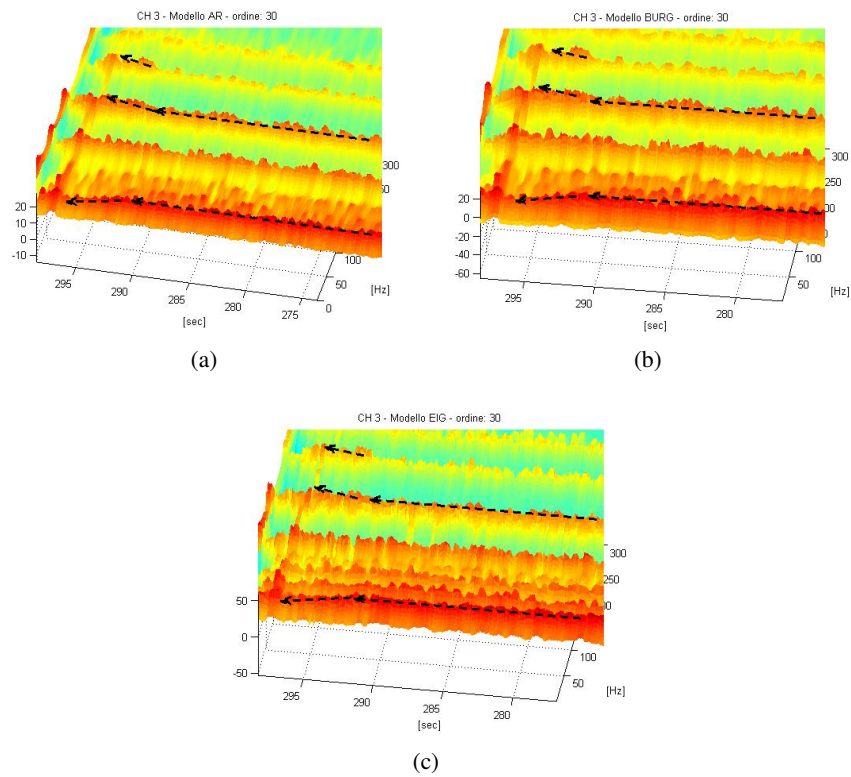


FIGURA 8.9: (a) Spettrogramma con modello autoregressivo con ordine 30. (b) Spettrogramma con metodo della massima entropia. (c) Spettrogramma con metodo agli autovalori. L'evento scoria è visibile con con tutti i metodi utilizzati ed in prossimità si nota come alcune bande diminuiscono mentre altre aumentino.

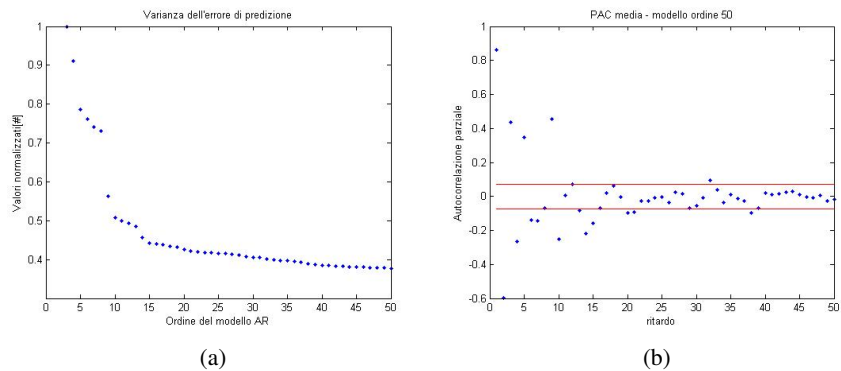


FIGURA 8.10: (a)-Varianza dell'errore di predizione per il modello AR. (b) Coefficiente di correlazione parziale per il modello AR.

$$MDL[p] = N \ln(\text{var}\{e^{(p)}(n)\}) + p \ln(N) \quad (8.11)$$

- *Criterion autoregressive transfer (CAT).*

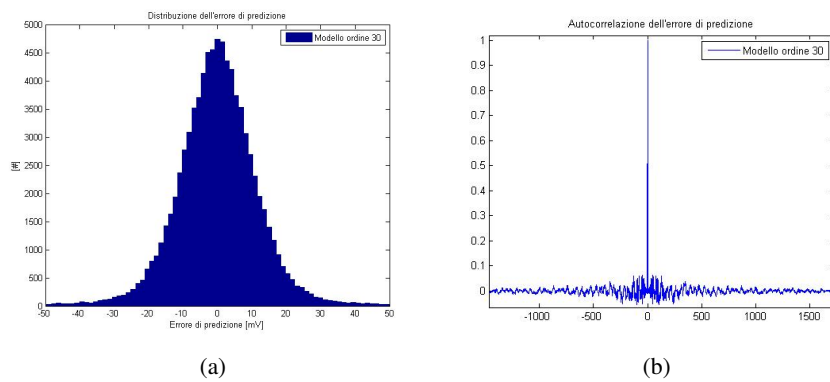


FIGURA 8.11: (a)- Distribuzione statistica dell'errore di predizione. (b) Autocorrelazione dell'errore di predizione.

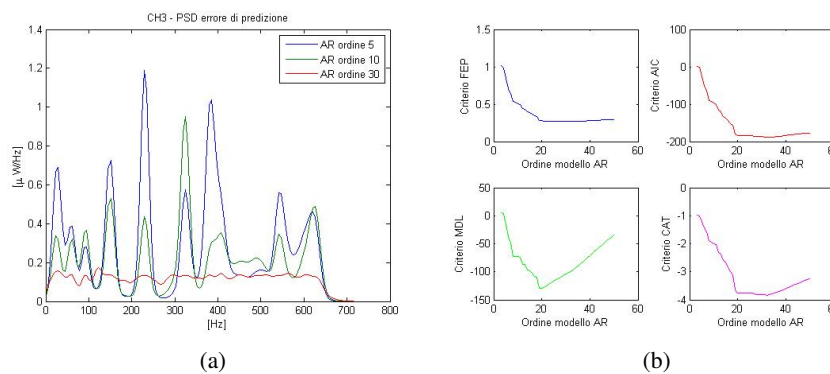


FIGURA 8.12: (a) Tendenza dell'errore di predizione verso una banda piatta al crescere dell'ordine del modello. (b) Andamento dei criteri di scelta dell'ordine del modello. Non tutti i criteri presentano il comportamento atteso.

$$\begin{aligned}
 CAT[p] &= \frac{1}{N} \left(\sum_{j=1}^p \frac{1}{\nu_j} \right) - \frac{1}{\nu_p} \\
 \nu_j &= \frac{N}{N-j} \text{var}\{e^{(p)}(j)\}^2
 \end{aligned}
 \tag{8.12}$$

Nella figura 8.10b è riportato il coefficiente di correlazione parziale. Come discusso nel capitolo 2 esso descrive la correlazione tra il campione $x(n)$ e quello $x(n - p - 1)$ dopo aver eliminato le dipendenze intermedie con la regressione lineare. Per un modello ideale AR di ordine p la correlazione parziale tra $x(n)$ e $x(n - p - 1)$ è nulla. Fissati quindi al 95% i limiti di confidenza dell'ipotesi di correlazione parziale nulla con la seguente formula,

$$\pm \frac{1.96}{\sqrt{N}}$$

deduciamo dal grafico 8.10b che l'ordine minimo del modello è all'incirca 20. Notiamo infine come l'errore di distribuzione (figura 8.11a) risulti normalmente distribuito e come la sua autocorrelazione sia confrontabile con quella del rumore bianco (figura 8.11b). In effetti mediante questa analisi si comprende come il modello AR catturi le caratteristiche principali del fenomeno: i residui infatti non presentano nessuna struttura correlata. E' estremamente interessante notare come all'aumentare dell'ordine del modello l'errore di predizione tenda ad un rumore bianco (figura 8.12a), più precisamente ad un rumore rosa essendo la banda in frequenza limitata. Nel grafico 8.12a sono riportati gli andamenti degli altri criteri, alcuni dei quali possono assumere un andamento poco utile per la determinazione del modello.

8.3 Analisi mediante i metodi *Wavelet* ed Hilbert

Le analisi eseguite con tecniche spettrali hanno evidenziato una struttura essenzialmente a bande, le quali appaiono invarianti solo rispetto alla singola siviera. Tuttavia l'analisi dei grafici bidimensionali non è certamente agevole in particolare se immaginiamo di calarla in un sistema in tempo reale. In questo tipo di rappresentazione sicuramente la visione di insieme aiuta a cogliere le informazioni contenute. Da punto di vista algoritmico sicuramente risulta più arduo riuscire ad estrarre l'informazione contenuta. I risultati precedenti suggeriscono che l'utilizzo di un banco di filtri possa rivelarsi una tecnica di analisi efficace. Poiché dalle immagini degli spettrogrammi le bande risultano comunque variabili, è chiaro che la progettazione di banchi di filtri dovrebbe essere ottimizzata per ogni siviera. Essendo ovviamente questa una procedura non applicabile, in alternativa si può utilizzare la tecnica *wavelet packets discreta*. Mediante questo approccio si crea l'equivalente di un banco di filtri, con il vantaggio che la potenza del segnale viene conservata come dimostrato nel capitolo 3. Definito quindi il livello di decomposizione L , si esegue la trasformata discreta *Wavelet packets* del segnale $x(t)$,

$$X_{WP}^L(t, i) = DWPT[x(t), i] \quad i = 1, \dots, 2^L. \quad (8.13)$$

Successivamente ogni trasformata $X_{WP}^L(t, i)$ viene antitrasformata in modo da ottenere solo il contributo della relativa banda in frequenza. La procedura è semplice poiché basta annullare tutti i coefficienti $X_{WP}^L(t, j)$ con $j \neq i$ e poi eseguire l'antitrasformata.

$$x(t, i) = DWPT^{-1}[X_{WP}^L(t, i)] \quad i = 1, \dots, 2^L. \quad (8.14)$$

Per poter infine stimare la potenza del segnale nella relativa banda si può considerare il valore *rms*. In effetti l'antitrasformata non sarebbe a rigor di logica indispensabile, tuttavia per facilitare il confronto con il dato globale ritengo sia utile utilizzarla. Nelle figure 8.13a, 8.13b e 8.13c è stata eseguita questa procedura, calcolando il valore rms su 5 buffer contigui per ogni asse dell'accelerometro. Come si può osservare il risultato è notevole poiché si ha una visione chiara del

comportamento in frequenza del segnale. Rispetto alle tecniche di densità spettrale esaminate nel paragrafo precedente l'andamento della potenza del segnale nella zona relativa al passaggio scoria presenta un comportamento facilmente identificabile. In effetti si nota che le frequenze basse mostrano un calo di potenza mentre quelle alte tendenzialmente aumentano come avevamo già notato in precedenza. E' bene sottolineare che in questo caso i risultati non cambiano variando la famiglia delle wavelet. I dati sono stati analizzati utilizzando le famiglie disponibili nel toolbox di matlab ed in effetti non ho osservato alcun elemento che permetta di preferire una famiglia rispetto alle altre. Come anticipato l'antitrasformata non è necessaria poiché il risultato ottenuto è già contenuto nei coefficienti $X_{WP}^L(t, i)$, tuttavia essa risulta molto utile per un confronto più immediato con le analisi classiche. Nelle figure esaminate la trasformata *wavelet* è stata eseguita su un numero di campioni molto elevato relativo ad un intervallo temporale di 300 secondi. Nelle analisi in tempo reale questo ovviamente non è possibile e la trasformata deve essere fatta ad ogni nuovo buffer acquisito. Nei grafici 8.14a, 8.14b e 8.14c si è considerato il caso peggiore in cui la procedura di decomposizione e ricostruzione viene applicata ad ogni singolo buffer acquisito. Il risultato è a tutti gli effetti analogo, infatti se osserviamo la figura 8.14d possiamo notare che il segnale ottenuto dalla trasformata di un numero molto elevato di buffer rappresenta una versione smussata di quello invece ottenuto nella modalità *realtime*. In pratica non si ha alcuna perdita di informazione e pertanto sarà sufficiente filtrare opportunamente il segnale verde per ottenere un risultato analogo a quello rappresentato dalla linea color rosso.

Da queste analisi emerge quindi in modo chiaro che il passaggio della scoria avviene secondo una modalità sostanzialmente definita e caratterizzata dal fatto che le bande in frequenza che rappresentano le risonanze del sistema porta-tuffante con siviera non cambiano nel corso di una colata e in prossimità dell'evento catastrofico modificano l'intensità relativa con uno schema certamente non invariante, ma nel complesso abbastanza ripetibile. In pratica si osserva che la banda più bassa si attenua sempre, mentre quelle superiori possono attenuarsi, aumentare di intensità oppure rimanere invariate. Le caratteristiche descritte suggeriscono che possa essere applicata la trasformata di Hilbert. In effetti mediante questo metodo si ottengono due grandezze caratteristiche come descritto nel capitolo 4, l'ampiezza istantanea $A_H(t)$ e la fase istantanea $\Phi_H(t)$. La variabile che può presentare l'interesse maggiore è però la derivata della fase ovvero la frequenza istantanea $\omega_H(t)$. Osservando infatti le figure 8.15a, 8.15b e 8.15c notiamo che la frequenza istantanea ha un comportamento molto ben definito in prossimità dell'evento catastrofico. In effetti nel grafico 8.15a si deduce che $\omega_H(t)$ si discosta dal trend precedente anche se $A_H(t)$ non sembra rilevare il passaggio della scoria. In effetti, in termini fisici possiamo interpretare l'attributo frequenza istantanea come l'equivalente del baricentro dello spettro del segnale. Anche se in termini energetici l'andamento delle bande può compensarsi in modo tale che la potenza del segnale risulti poco sensibile all'evento catastrofico, tuttavia il baricentro rivela il rimaneggiamento che avviene in quel breve istante. E' molto interessante notare come $A_H(t)$ presenti sostanzialmente la stessa informazione del valore *rms* a riprova del fatto che

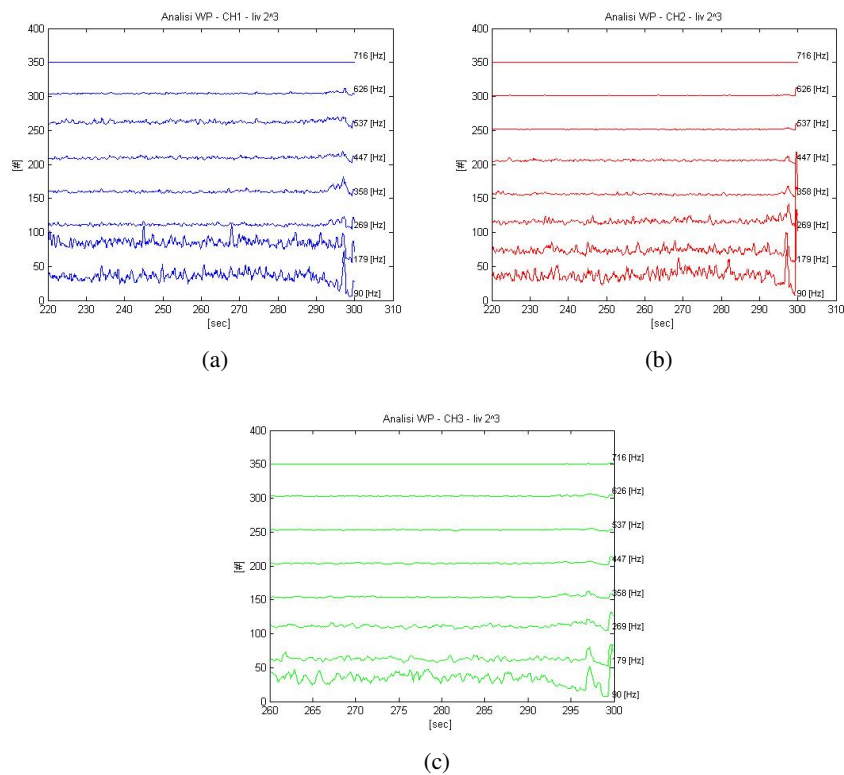


FIGURA 8.13: Analisi *Wavelet Packets* discreta. Ogni livello è stato ricostruito nella relativa banda ed è stato calcolato il valore *rms*. La trasformata è stata applicata all'intera serie temporale.

l'ampiezza istantanea sia da correlarsi alla potenza del segnale registrato. In effetti analogamente ai grafici che riportano il segnale *rms* si osserva che i canali CH1 e CH2 non registrano l'evento come invece avviene per il canale CH3. E' impressionante invece notare come l'attributo frequenza istantanea dimostri una sensibilità elevata su tutti e tre i canali. Sebbene sia la trasformata wavelet che quella di Hilbert siano in grado di cogliere l'evento scoria, tuttavia dobbiamo concludere che la seconda si presta meglio ad un controllo in tempo reale. Tuttavia per cogliere l'essenza del fenomeno ritengo sia importante utilizzare un approccio che faccia uso di più tecniche. In effetti tutti i metodi utilizzati hanno permesso di focalizzare l'attenzione su aspetti peculiari del fenomeno. Mediante la tecnica classica si ottiene una chiara visione della struttura periodica dovuta alle risonanze meccaniche del sistema, che tuttavia non si mantengono inalterate da siviera a siviera. Questo suggerisce che il sistema meccanico presenti dei gradi di libertà che evidentemente vengono modificati in modo casuale dalle operazioni manuali di sgancio ed aggancio del tuffante. Il fatto interessante è che una volta avviato il processo il sistema mantiene il particolare schema di risonanza. Durante il passaggio scoria da un punto di vista pratico si ha una variazione di densità, in pratica da circa 7000 kg/m³ si passa progressivamente a 2000 kg/m³. Poiché sperimentalmente le risonanze non variano si può concludere che il fluido agisce come un elemento eccitante che non altera in modo sensibile l'inerzia del sistema astutuffante. Questo risultato suggerisce che un attento studio della struttura meccanica potrebbe

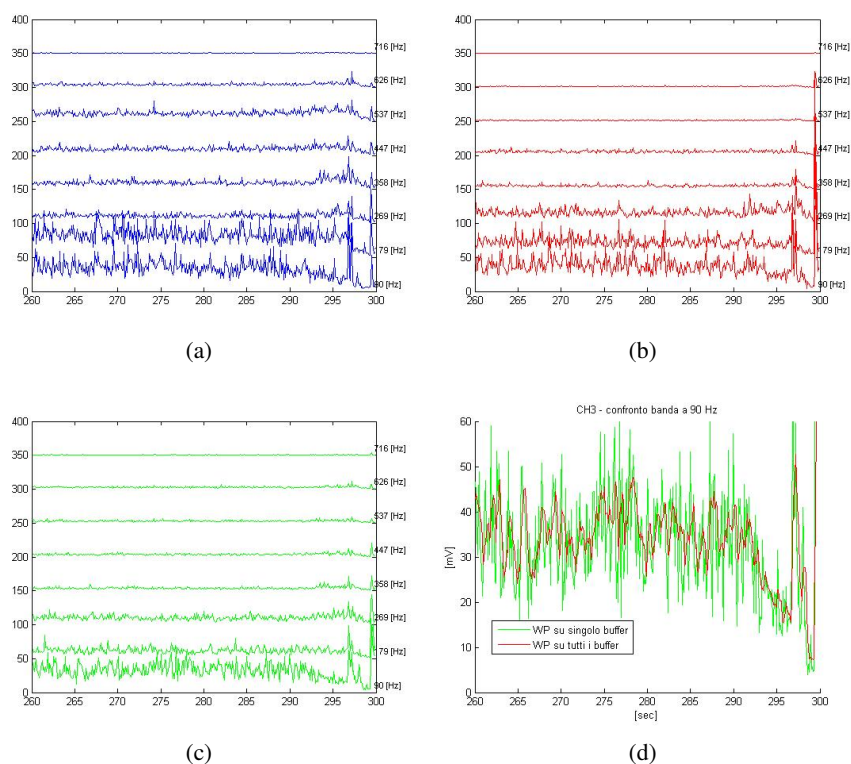


FIGURA 8.14: Analisi *Wavelet Packets* discreta eseguita su ogni singolo buffer di dati acquisito. Ogni livello è stato ricostruito nella relativa banda ed è stato calcolato il valore rms. Nella figura (d) è riportato il confronto dei valori rms ricostruiti relativi alla banda più bassa. Il segnale rosso rappresenta la DWPT eseguita su un numero di buffer molto maggiore rispetto al segnale verde.

aiutare sia a comprendere meglio la dinamica del processo che caratterizza l'evento scoria sia ad identificare il meccanismo che rende variabili le risonanze da siviera a siviera. La natura essenzialmente stazionaria dell'analisi classica entra in crisi proprio durante il passaggio della scoria, che invece viene registrato molto bene dall'analisi wavelet. Questa tecnica è in grado di mostrare la repentina variazione di energia anche quando viene applicata in *real-time*. Chiaramente la risoluzione in frequenza è inferiore rispetto a quella dello spettrogramma tuttavia essa è sufficiente per cogliere il fenomeno. Aumentare il livello di decomposizione è in questo caso impraticabile per due motivi. Il primo riguarda i tempi di calcolo che aumentano oltre il tempo di ciclo, il secondo si scontra con il numero finito di campioni del buffer. Il livello scelto rappresenta un buon compromesso tra la lunghezza del filtro, gli effetti al bordo e il numero di campioni disponibili.

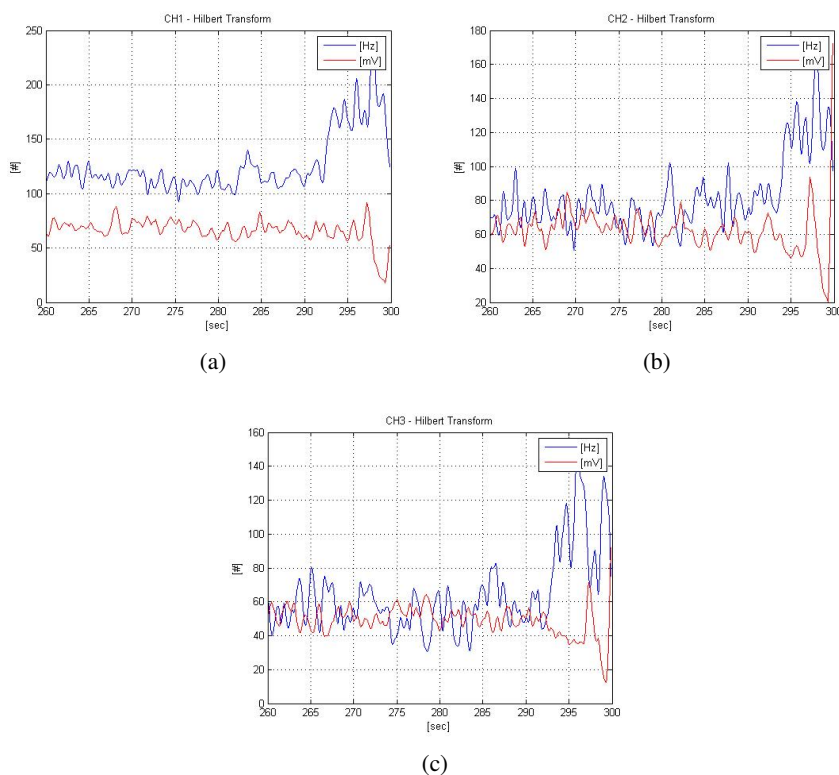


FIGURA 8.15: Ampiezza e frequenza istantanea ottenuti dalla trasformata di Hilbert.

8.4 Analisi degli effetti dovuti alla movimentazione del cassetto della siviera

In termini intuitivi appare corretto attendersi che aumentando la sezione del getto di uscita dell'acciaio, la potenza del segnale viene influenzata. In effetti osservando la figura 8.16a in seguito all'apertura del cassetto si nota un aumento dell'energia della serie registrata. Dall'analisi della figura 8.16c osserviamo come lo spettrogramma inequivocabilmente dimostri che la variazione della sezione del getto agisce come un fattore di guadagno ma senza modificare la struttura delle risonanze. Però si nota che il guadagno agisce in modo simile su tutte le bande. Questo risultato è in contraddizione con quanto visto nei paragrafi precedenti pertanto si può affermare che il passaggio della scoria **non** è equivalente ad una variazione di sezione del getto di acciaio. A conferma di ciò osserviamo che la trasformata wavelet riportata in figura 8.17b non ha uno schema simile a quello registrato in precedenza. L'attributo frequenza istantanea (figura 8.17a) rimane inalterato a conferma che il guadagno agisce in modo simile su tutte le bande del segnale.

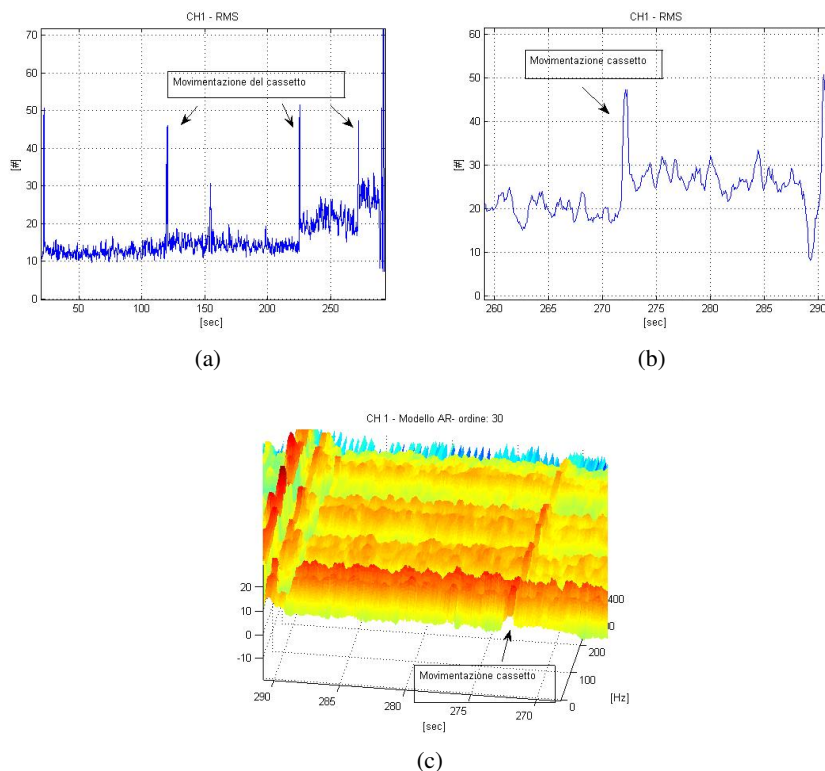


FIGURA 8.16: (a) Valore *rms* registrato dopo ogni variazione della sezione di uscita del getto. (b) Ingrandimento relativo all'ultimo picco della figura (a). (c) Spettrogramma relativo all'evento riportato nella figura (b). Si osserva come la variazione della sezione del cassetto agisce in modo equivalente su tutte le bande del sistema, contrariamente a quanto si osserva durante il passaggio di scoria.

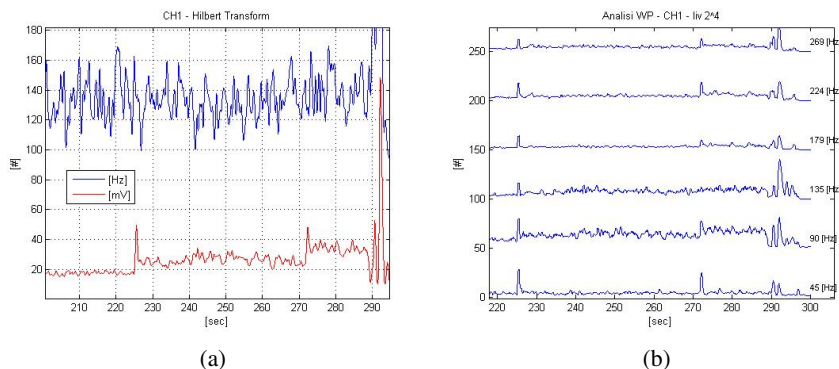


FIGURA 8.17: (a) Frequenza istantanea relativa all'evento riportato in figura 8.16a. (b) Trasformata wavelet discreta relativa all'evento 8.16a. La trasformata wavelet registra l'aumento di energia su tutte le bande senza modificare i contenuti relativi di frequenza come testimonia il valore della frequenza istantanea.

Capitolo 9

Conclusioni

L'attività di ricerca si è svolta in accordo con gli obiettivi descritti nel capitolo introduttivo. Inizialmente è stato condotto uno studio per identificare le tecniche di analisi più idonee. Per quanto riguarda l'analisi classica si è applicato lo schema di Welch per il calcolo delle densità di potenza nel piano tempo - frequenza. La ricerca si è poi estesa ai modelli autoregressivi con l'intento di valutare se la loro applicazione potesse migliorare ulteriormente la capacità di discriminare le caratteristiche del segnale. In effetti si è osservato che in questo caso le metodologie esplorate sono sostanzialmente equivalenti. Successivamente sono state analizzate la tecnica della trasformata Wavelet discreta e la tecnica della trasformata di Hilbert. Per quanto riguarda l'analisi *Wavelet* è stato applicato il metodo *Wavelets Packets*. Dopo aver adeguatamente identificato le tecniche più promettenti, l'attività è proseguita con lo sviluppo degli algoritmi per l'utilizzo in tempo reale. Si è cercato di non impiegare alcuna libreria esterna. L'unica eccezione è stata fatta per la trasformata di Fourier. In effetti si è sperimentato che la classe autocostruita della FFT aveva prestazioni inferiori a quelle della FFTW relativa al consorzio www.fftw.org, che costituisce uno standard di riferimento. Per questa ragione la libreria FFTW è stata utilizzata nel progetto al posto della FFT. Per l'implementazione software degli algoritmi è stato utilizzato il linguaggio C++. In genere l'intera attività di sviluppo degli algoritmi e del software ha richiesto un discreto tempo ed un impegno significativo. Ad ogni modo parallelamente è stato progettata e realizzata l'architettura hardware. Chiaramente la parte più impegnativa è stata quella relativa all'attrezzatura per i test in campo. L'ambiente infatti risulta essere particolarmente aggressivo e quindi l'accelerometro è stato inglobato in una robusta struttura in acciaio inossidabile che si è rivelata fondamentale. In effetti, sebbene la presente ricerca si basi su un'attività chiaramente sperimentale, per poter eseguire i test presso gli impianti selezionati, l'attrezzatura impiegata deve essere conforme agli standard industriali. In quest'ottica la scelta di utilizzare un sistema basato su ethernet si è rivelata sicuramente vantaggiosa: i vari moduli hardware sono stati posti nelle zone più idonee. Nel complesso il dispositivo in tempo reale ha funzionato con cicli alla frequenza di 10 Hz e con tempi di calcolo nettamente inferiori ai 100

ms su un computer desktop dalle medie prestazioni (intel i5). Per confronto gli stessi test sono stati realizzati su un prototipo in scala funzionante ad acqua. Esso è stato realizzato in modo da riprodurre lo stesso meccanismo di svuotamento della siviera. Il dispositivo inoltre è stato attrezzato con una cella di carico e con una telecamera per poter acquisire ulteriori informazioni utili all'identificazione degli eventi tipo scoria. Infine è stata avviata un'attività di simulazione numerica tridimensionale con lo scopo di comprendere meglio i fenomeni di *drain-sink* e di formazione del vortice all'interno della siviera.

Nei test con l'acqua si è cercato di fare una prima esplorazione nello spazio delle variabili del sistema. E' chiaramente emerso che quando lo scarico avviene in aria esiste una chiara differenza tra le due possibili modalità di svuotamento. Durante il *drain-sink* l'energia è inferiore e si nota molto bene una breve impennata del suo valore proprio nell'istante del collasso della superficie libera. Il vortice invece induce un livello energetico superiore. Tutte le tecniche di analisi impiegate danno un contributo alla visione del fenomeno. Con l'approccio classico si può apprezzare con facilità la struttura a bande ovvero un comportamento tipico di elementi risonanti. Emerge chiaramente che la modalità *drain-sink* contiene la potenza del segnale esclusivamente alla basse energie, mentre quella con il vortice mostra una struttura a bande che si estende anche a frequenze superiori. Questa tecnica permette anche di apprezzare il fatto che le bande non si modificano durante il processo, ma rimangono inalterate sino al termine dello svuotamento. Come invece era prevedibile l'attimo in cui la superficie libera collassa non è identificabile con chiarezza, infatti la tecnica classica basata sulla trasformata di Fourier non si presta ad identificare eventi transitori. La trasformata *Wavelet* discreta invece permette di avere una visione più adatta dei fenomeni transitori ed in effetti essa mostra chiaramente un'immagine del fenomeno cogliendo molto bene la dinamica dello svuotamento. Essa inoltre offre una scomposizione del segnale in frequenza sufficiente ad apprezzare il contributo delle alte frequenze rispetto alle basse. Chiaramente se paragonata allo spettrogramma questa tecnica offre una risoluzione inferiore e certamente non permette di cogliere chiaramente la struttura a bande. In effetti il livello di decomposizione è comunque limitato dal numero di punti del singolo buffer di acquisizione, che deve essere elaborato in tempo reale. Si può certamente concludere che l'insieme delle due tecniche offre una visione integrata ed in questo caso non è corretto affermare che una tecnica sia migliore dell'altra. Tuttavia ragionando in termini di risposta in tempo reale, ovvero osservando la capacità di identificare l'evento catastrofico, è indubbio che la trasformata *Wavelet* è nettamente superiore. L'analisi degli attributi con il metodo della trasformata di Hilbert rivela che l'ampiezza istantanea, come atteso, si comporta esattamente come il valore *rms* globale del segnale. La frequenza istantanea si rivela capace di distinguere nettamente le due modalità di svuotamento, inoltre essa identifica molto bene l'evento catastrofico finale. In questo caso vista la struttura risonante del segnale, essa può essere interpretata come il baricentro delle frequenze. In effetti si osserva chiaramente che la frequenza istantanea si modifica al variare dell'intensità relativa delle bande: in caso di presenza di energia nelle bande superiori allora esse tenda ad

umentare, contrariamente il suo valore diminuisce quando tutta la potenza è concentrata nelle bande inferiori. Si può certamente affermare che tra le tecniche usate essa è capace di cogliere la differente dinamica del sistema offrendo indubbiamente una visione alternativa rispetto allo spettrogramma e alla trasformata *Wavelet*. La frequenza istantanea inoltre si rivela adatta ad un utilizzo in tempo reale, permettendo di identificare chiaramente l'evento catastrofico. Si può quindi concludere che l'insieme delle tecniche utilizzate permettono un'analisi completa del segnale.

La situazione si complica notevolmente quando il tuffante viene inserito in un contenitore che simula la paniera. Gli schemi del segnale diventano più simili e non è possibile identificare con la stessa facilità le due modalità di svuotamento. In questo caso l'attributo della frequenza istantanea mantiene una maggiore sensibilità nell'edificare l'evento catastrofico. I test in impianto hanno mostrato come la struttura del segnale sia sostanzialmente equivalente al modello in scala. La caratteristica divisione in bande è mantenuta, mostrando la sua invarianza durante il singolo processo. La posizione dell'accelerometro non influenza la dinamica del segnale come nel prototipo con l'acqua. Ma uno degli aspetti sicuramente più sorprendenti è che anche in questo caso l'attributo della frequenza istantanea si dimostra capace di cogliere l'evento catastrofico, svelando una notevole analogia con il modello ad acqua.

L'analisi fluidodinamica si è limitata al caso del *drain-sink*. I risultati sono stati soddisfacenti: l'attimo e l'altezza in cui avviene il collasso della superficie concordano con i dati sperimentali. E' chiaramente emerso che i modelli analitici possono essere utilizzati solo in modo semi-quantitativo, per risultati più accurati la tecnica computazionale si dimostra una strada obbligata. Il problema tuttavia può essere affrontato solo con l'ausilio di risorse HPC. Il software openFoam si è dimostrato affidabile e scalabile con il numero di processori.

L'attività di ricerca svolta ha indubbiamente evidenziato che il metodo seguito ha permesso di affrontare proficuamente il fenomeno da diversi punti di vista. Sono emersi molti aspetti interessanti che potranno essere esplorati con attività dedicate. Nel caso dei vortici, il modello in scala dovrebbe essere dotato di un sistema motorizzato ed automatizzato per la generazione della componente tangenziale iniziale nel fluido. Il controllo di questo parametro permetterebbe di eseguire un accurato studio della velocità critica che rappresenta il confine tra le due modalità di svuotamento, ovvero il passaggio dal modo *drain-sink* a quello con vortice. In effetti questo aspetto risulta importante anche per la corretta applicazione delle condizioni al contorno al modello matematico della simulazione. Sicuramente è necessario approfondire anche gli aspetti geometrici relativi al fondo del contenitore in plexiglass. Sarebbe infatti interessante introdurre geometrie simili a quelle della siviera e questo richiederà la realizzazione di modelli più sofisticati. In particolare si ritiene che sia importante costruire un dispositivo meccanico che riproduca

il comportamento del cassetto: in effetti le esperienze in impianto dimostrano che la parzializzazione del getto modifica il contenuto energetico del segnale. Un altro argomento su cui bisognerà indagare in modo adeguato riguarda l'interazione tra il tuffante e il liquido nella vasca di scarico poiché profondità del tuffante e geometria della vasca di scarico possono influenzare la dinamica del segnale. Per quanto riguarda la simulazione fluidodinamica essa ha dimostrato di essere un valido strumento per comprendere le dinamiche di svuotamento. Certamente sarà necessario approfondire il modello matematico con turbolenza in quanto quello laminare non è applicabile al caso dei vortici. Tuttavia anche nel caso laminare, supponendo che il *drain-sink* sia la modalità di svuotamento dominante nella siviera, occorre provare a valutare come le diverse geometrie influiscano nel fenomeno del collasso della superficie libera. Chiaramente amplificare la dinamica reale del collasso potrebbe potenzialmente portare al potenziamento del segnale registrato e rendere più evidente l'evento. Lo studio di questo parametro risulta molto più agevole con tecniche virtuali, le quali hanno dimostrato la capacità di cogliere l'attimo del collasso della superficie.

Si può quindi concludere che l'approccio applicato si è dimostrato valido e le tecniche utilizzate sono adatte ad un impiego in tempo reale. Infine i risultati ottenuti permettono di proseguire nell'approfondimento della tematiche emerse nel corso dello svolgimento della ricerca.

Appendice A

Costruzione di filtri IIR

La classe *TFiltro* (capitolo 5) calcola i coefficienti dei filtri digitali di Butterworth, Chebyshev, Chebyshev inverso ed ellittico. L'implementazione di questa classe è basata sul lavoro di L. Thede[33] da cui ho tratto le formule generali, che ho specializzato mediante la trasformazione bilineare.

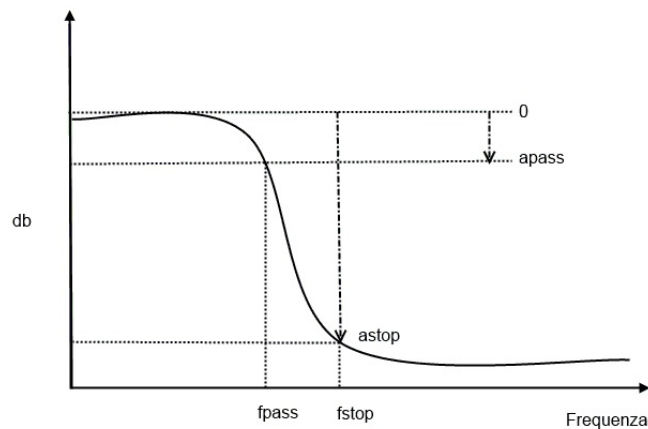


FIGURA A.1: Specifiche di un filtro

La procedura prevede che si parte dall'espressione generale approssimata del filtro passa basso nel dominio s , (A.1), e si procede alle sostituzioni che sono state spiegate nel paragrafo 5.1.4. Consideriamo prima il caso di ordine pari, quindi nella (A.1a) eseguiamo la sostituzione $S =$

s/ω_0 ,

$$H(S) = G_e \frac{\prod_m (A_{0m}S^2 + A_{1m}S + A_{2m})}{\prod_m (B_{0m}S^2 + B_{1m}S + B_{2m})} \quad (\text{A.1a})$$

$m = 0, \dots, n/2$ n pari

$$H(S) = \frac{A_{L1}S + A_{L2} \prod_m (A_{0m}S^2 + A_{1m}S + A_{2m})}{B_{L1}S + B_{L2} \prod_m (B_{0m}S^2 + B_{1m}S + B_{2m})} \quad (\text{A.1b})$$

$m = 0, \dots, n/2$ n dispari,

A.1 Filtro - passa basso

Consideriamo prima il caso di ordine pari.

$$\begin{aligned} H(S) &= \left(\frac{\prod_m (A_{0m}S^2 + A_{1m}S + A_{2m})}{\prod_m (B_{0m}S^2 + B_{1m}S + B_{2m})} \right)_{S=\frac{s}{\omega_0}} \\ H(s) &= \left(\frac{\prod_m (\frac{A_{0m}}{\omega_0^2} s^2 + \frac{A_{1m}}{\omega_0} s + A_{2m})}{\prod_m (\frac{B_{0m}}{\omega_0^2} s^2 + \frac{B_{1m}}{\omega_0} s + B_{2m})} \right)_{\frac{2}{T} \frac{z-1}{z+1}} \\ H(z) &= \left(\prod_m \frac{N_{0m}}{D_{0m}} \right) \frac{\prod_m (1 + \frac{N_{1m}}{N_{0m}} z^{-1} + \frac{N_{2m}}{N_{0m}} z^{-2})}{\prod_m (1 + \frac{D_{1m}}{D_{0m}} z^{-1} + \frac{D_{2m}}{D_{0m}} z^{-2})}, \end{aligned} \quad (\text{A.2})$$

con

$$\begin{aligned} N_{0m} &= a_{0m} + a_{1m} + a_{2m}, \\ N_{1m} &= -2a_{0m} + 2a_{2m}, \\ N_{2m} &= a_{0m} - a_{1m} + a_{2m}, \\ D_{0m} &= b_{0m} + b_{1m} + b_{2m}, \\ D_{1m} &= -2b_{0m} + 2b_{2m}, \\ D_{2m} &= b_{0m} - b_{1m} + b_{2m}, \end{aligned} \quad (\text{A.3})$$

e

$$\begin{aligned} a_{0m} &= \frac{A_{0m}}{\omega_N \pi^2} & a_{1m} &= \frac{A_{1m}}{\omega_N \pi} & a_{2m} &= A_{2m} \\ b_{0m} &= \frac{B_{0m}}{\omega_N \pi^2} & b_{1m} &= \frac{B_{1m}}{\omega_N \pi} & b_{2m} &= B_{2m} \\ \omega_N &= \frac{\omega_0}{\omega_s} & \omega_s &= 2\pi f_s & f_s &= \frac{1}{T} \end{aligned} \quad (\text{A.4})$$

Applicando la stessa logica al termine lineare otteniamo,

$$\begin{aligned}
 H(S) &= \left(\frac{(A_{L1}S + A_{L2})}{(B_{L1}S + B_{L2})} \right)_{S=\frac{s}{\omega_0}} \\
 H(s) &= \left(\frac{\frac{A_{L1}}{\omega_0}s + A_2}{\frac{B_{L1}}{\omega_0}s + B_{L2}} \right)_{\frac{2}{T} \frac{z-1}{z+1}} \\
 H(z) &= \left(\frac{N_{L0}}{D_{L0}} \right) \frac{1 + \frac{N_1}{N_0}z^{-1}}{1 + \frac{D_1}{D_0}z^{-1}}, \tag{A.5}
 \end{aligned}$$

con

$$\begin{aligned}
 N_{L0} &= a_{L1} + a_{L2}, \\
 N_{L1} &= -a_{L1} + a_{L2}, \\
 D_{L0} &= b_{L1} + b_{L2}, \\
 D_{L1} &= -b_{L1} + b_{L2}, \tag{A.6}
 \end{aligned}$$

e

$$\begin{aligned}
 a_{L1m} &= \frac{A_{L1m}}{\omega_N \pi} & a_{L2m} &= A_{L2m} \\
 b_{L1m} &= \frac{B_{L1m}}{\omega_N \pi} & b_{L2m} &= B_{L2m} \tag{A.7}
 \end{aligned}$$

A.2 Filtro passa alto

Consideriamo prima il caso di ordine pari.

$$\begin{aligned}
 H(S) &= \left(\frac{\prod_m (A_{0m}S^2 + A_{1m}S + A_{2m})}{\prod_m (B_{0m}S^2 + B_{1m}S + B_{2m})} \right)_{S=\frac{\omega_0}{s}} \\
 H(s) &= \left(\frac{\prod_m (A_{2m}s^2 + A_{1m}\omega_0s + A_{0m}\omega_0^2)}{\prod_m (B_{2m}s^2 + B_{1m}\omega_0s + B_{0m}\omega_0^2)} \right)_{\frac{2}{T} \frac{z-1}{z+1}} \\
 H(z) &= \left(\prod_m \frac{N_{0m}}{D_{0m}} \right) \frac{\prod_m (1 + \frac{N_{1m}}{N_{0m}}z^{-1} + \frac{N_{2m}}{N_{0m}}z^{-2})}{\prod_m (1 + \frac{D_{1m}}{D_{0m}}z^{-1} + \frac{D_{2m}}{D_{0m}}z^{-2})}, \tag{A.8}
 \end{aligned}$$

con

$$\begin{aligned}
 N_{0m} &= a_{0m} + a_{1m} + a_{2m}, \\
 N_{1m} &= -2a_{0m} + 2a_{2m}, \\
 N_{2m} &= a_{0m} - a_{1m} + a_{2m}, \\
 D_{0m} &= b_{0m} + b_{1m} + b_{2m}, \\
 D_{1m} &= -2b_{0m} + 2b_{2m}, \\
 D_{2m} &= b_{0m} - b_{1m} + b_{2m},
 \end{aligned} \tag{A.9}$$

e

$$\begin{aligned}
 a_{0m} &= \frac{A_{2m}}{\pi^2} & a_{1m} &= \frac{A_{1m}\omega_N}{\pi} & a_{2m} &= A_{0m}\omega_N^2 \\
 b_{0m} &= \frac{B_{2m}}{\pi^2} & b_{1m} &= \frac{B_{1m}\omega_N}{\pi} & b_{2m} &= B_{0m}\omega_N^2
 \end{aligned}$$

Applicando la stessa logica al termine lineare otteniamo,

$$\begin{aligned}
 H(S) &= \left(\frac{(A_{L1}S + A_{L2})}{(B_{L1}S + B_{L2})} \right)_{S=\frac{\omega_0}{s}} \\
 H(s) &= \left(\frac{(A_{L2}s + A_{L1}\omega_0)}{(B_{L2}s + B_{L1}\omega_0)} \right)_{\frac{2}{T} \frac{z-1}{z+1}} \\
 H(z) &= \left(\frac{N_{L0}}{D_{L0}} \right) \frac{1 + \frac{N_1}{N_0}z^{-1}}{1 + \frac{D_1}{D_0}z^{-1}},
 \end{aligned} \tag{A.10}$$

con

$$\begin{aligned}
 N_{L0} &= a_{L1} + a_{L2}, \\
 N_{L1} &= -a_{L1} + a_{L2}, \\
 D_{L0} &= b_{L1} + b_{L2}, \\
 D_{L1} &= -b_{L1} + b_{L2},
 \end{aligned} \tag{A.11}$$

e

$$\begin{aligned}
 a_{L1} &= \frac{A_{L2}}{\pi} & a_{L2} &= A_{L2}\omega_N \\
 b_{L1} &= \frac{B_{L1}}{\pi} & b_{L2} &= B_{L2}\omega_N
 \end{aligned} \tag{A.12}$$

A.3 Filtro passa banda

Consideriamo prima il caso di ordine pari.

$$\begin{aligned}
 H(S) &= \left(\frac{\prod_m (A_{0m} S^2 + A_{1m} S + A_{2m})}{\prod_m (B_{0m} S^2 + B_{1m} S + B_{2m})} \right)_{S = \frac{s^2 + \omega_0^2}{BW s}} \\
 H(s) &= \left(\frac{\prod_m (A_{0m} s^4 + A_{1m} BW s^3 + (2\omega_0^2 A_{0m} + A_{2m} BW^2) s^2 + A_{1m} \omega_0^2 BW s + A_{0m} \omega_0^4)}{\prod_m (B_{0m} s^4 + B_{1m} BW s^3 + (2\omega_0^2 B_{0m} + B_{2m} BW^2) s^2 + B_{1m} \omega_0^2 BW s + B_{0m} \omega_0^4)} \right)^{\frac{2}{T} \frac{z-1}{z+1}} \\
 H(z) &= \left(\prod_m \frac{N_{0m}}{D_{0m}} \right) \frac{\prod_m (1 + \frac{N_{1m}}{N_{0m}} z^{-1} + \frac{N_{2m}}{N_{0m}} z^{-2} + \frac{N_{3m}}{N_{0m}} z^{-3} + \frac{N_{4m}}{N_{0m}} z^{-4})}{\prod_m (1 + \frac{D_{1m}}{D_{0m}} z^{-1} + \frac{D_{2m}}{D_{0m}} z^{-2} + \frac{D_{3m}}{D_{0m}} z^{-3} + \frac{D_{4m}}{D_{0m}} z^{-4})}, \tag{A.13}
 \end{aligned}$$

con

$$\begin{aligned}
 N_{0m} &= a_{0m} + a_{1m} + a_{2m} + a_{3m} + a_{4m}, \\
 N_{1m} &= -4a_{0m} - 2a_{1m} + 2a_{3m} + 4a_{4m}, \\
 N_{2m} &= 6a_{0m} - 2a_{2m} + 6a_{4m}, \\
 N_{3m} &= -4a_{0m} + 2a_{1m} - 6a_{3m} + 4a_{4m}, \\
 N_{4m} &= a_{0m} - 2a_{1m} + a_{2m} - a_{3m} + a_{4m}, \\
 D_{0m} &= b_{0m} + b_{1m} + b_{2m} + b_{3m} + b_{4m}, \\
 D_{1m} &= -4b_{0m} - 2b_{1m} + 2b_{3m} + 4b_{4m}, \\
 D_{2m} &= 6b_{0m} - 2b_{2m} + 6b_{4m}, \\
 D_{3m} &= -4b_{0m} + 2b_{1m} - 6b_{3m} + 4b_{4m}, \\
 D_{4m} &= b_{0m} - 2b_{1m} + b_{2m} - b_{3m} + b_{4m},
 \end{aligned} \tag{A.14}$$

e

$$\begin{aligned}
 a_{0m} &= \frac{A_{0m}}{\pi^4} & a_{1m} &= \frac{A_{1m} BW_N}{\pi^3} \\
 a_{2m} &= 2 \frac{\omega_N^2}{\pi^2} A_{0m} + \frac{BW_N^2}{\pi^2} A_{2m} & a_{3m} &= \frac{\omega_N^2 BW_N}{\pi} A_{1m} \\
 a_{4m} &= \omega_N^4 A_{0m} & BW_N &= \frac{BW}{\omega_s} \\
 b_{0m} &= \frac{B_{0m}}{\pi^4} & b_{1m} &= \frac{B_{1m} BW_N}{\pi^3} \\
 b_{2m} &= 2 \frac{\omega_N^2}{\pi^2} B_{0m} + \frac{BW_N^2}{\pi^2} B_{2m} & b_{3m} &= \frac{\omega_N^2 BW_N}{\pi} B_{1m} \\
 b_{4m} &= \omega_N^4 B_{0m}.
 \end{aligned} \tag{A.15}$$

Applicando la stessa logica al termine lineare otteniamo,

$$\begin{aligned}
 H(S) &= \left(\frac{A_{L1}S + A_{L2}}{B_{L1}S + B_{L2}} \right)_{S = \frac{s^2 + \omega_0^2}{BW s}} \\
 H(s) &= \left(\frac{A_{L1}s^2 + A_{L2}BW s + A_{L1}\omega_0^2}{B_{L1}s^2 + B_{L2}BW s + B_{L1}\omega_0^2} \right)_{\frac{z}{z+1}} \\
 H(z) &= \left(\frac{N_{L0}}{D_{L0}} \right) \frac{1 + \frac{N_{L1}}{N_{L0}}z^{-1} + \frac{N_{L2}}{N_{L0}}z^{-2}}{1 + \frac{D_{L1}}{D_{L0}}z^{-1} + \frac{D_{L2}}{D_{L0}}z^{-2}}, \tag{A.16}
 \end{aligned}$$

con

$$\begin{aligned}
 N_{L0} &= a_{L0} + a_{L1} + a_{L2}, \\
 N_{L1} &= -2a_{L0} + 2a_{L2}, \\
 N_{L2} &= a_{L0} - a_{L1} + a_{L2}, \\
 D_{L0} &= b_{L0} + b_{L1} + b_{L2}, \\
 D_{L1} &= -2b_{L0} + 2b_{L2}, \\
 D_{L2} &= b_{L0} - b_{L1} + b_{L2}, \tag{A.17}
 \end{aligned}$$

e

$$\begin{aligned}
 a_{L0} &= \frac{A_{L1}}{\pi^2} & a_{L1} &= \frac{A_{L2}BW_N}{\pi} & a_{L2} &= A_{L1}\omega_N^2 \\
 b_{L0} &= \frac{B_{L1}}{\pi^2} & b_{L1} &= \frac{B_{L2}BW_N}{\pi} & b_{L2} &= B_{L1}\omega_N^2 \tag{A.18}
 \end{aligned}$$

A.4 Filtro di Butterworth

$$\begin{aligned}
 A_{0m} &= 0 & A_{1m} &= 0 & A_{2m} &= 1 \\
 B_{0m} &= 1 & A_{L1} &= 0 & A_{L2} &= R \\
 B_{L1} &= 1 & B_{L2} &= R & B_{1m} &= -2\sigma_m \\
 B_{2m} &= \sigma_m^2 + \omega_m^2 \tag{A.19}
 \end{aligned}$$

con

$$\begin{aligned}
 \epsilon &= \sqrt{10^{-0.1a_p} - 1} & R &= \epsilon^{-\frac{1}{n}} \\
 \sigma_R &= -R & \sigma_m &= R \cos(\theta_m) & \omega_m &= R \sin(\theta_m) \\
 \theta_m &= \frac{\pi(2m + n + 1)}{2n} & m &= 0, 1, \dots, \text{int}\left(\frac{n}{2}\right) - 1 \tag{A.20}
 \end{aligned}$$

con a_p guadagno della banda passante.

A.5 Filtro di Chebyshev

$$\begin{aligned}
 A_{0m} &= 0 & A_{1m} &= 0 & A_{2m} &= 1 \\
 B_{0m} &= 1 & A_{L1} &= 0 & A_{L2} &= \sinh(D) \\
 B_{L1} &= 1 & B_{L2} &= \sinh(D) & B_{1m} &= -2\sigma_m \\
 & & B_{2m} &= \sigma_m^2 + \omega_m^2 & G_e &= 10^{0.05a_p}
 \end{aligned} \tag{A.21}$$

con

$$\begin{aligned}
 \epsilon &= \sqrt{10^{-0.1a_p} - 1} & D &= \frac{\sinh^{-1}(\epsilon^{-1})}{n} \\
 \sigma_R &= -\sinh(D) & \sigma_m &= -\sinh(D) \sin(\theta_m) & \omega_m &= \cosh(D) \cos(\theta_m) \\
 \theta_m &= \frac{\pi(2m+1)}{2n} & m &= 0, 1, \dots, \text{int}\left(\frac{n}{2}\right) - 1
 \end{aligned} \tag{A.22}$$

con a_p guadagno della banda passante.

A.6 Filtro di Chebyshev inverso

$$\begin{aligned}
 A_{0m} &= 1 & B_{0m} &= 1 & A_{1m} &= 0 \\
 A_{L1} &= 0 & A_{L2} &= \sinh(D)^{-1} & B_{L1} &= 1 \\
 & & B_{L2} &= \sinh(D)^{-1} & B_{1m} &= -2\sigma_m \\
 & & B_{2m} &= \sigma_m^2 + \omega_m^2 & A_{2m} &= \omega_{zm}^2
 \end{aligned} \tag{A.23}$$

con

$$\begin{aligned}
 \epsilon &= \frac{1}{\sqrt{10^{-0.1a_p} - 1}} & D &= \frac{\sinh^{-1}(\epsilon^{-1})}{n} \\
 \sigma_R &= -\sinh(D)^{-1} & \sigma'_m &= -\sinh(D) \sin(\theta_m) & \omega'_m &= \cosh(D) \cos(\theta_m) \\
 \theta_m &= \frac{\pi(2m+1)}{2n} & m &= 0, 1, \dots, \text{int}\left(\frac{n}{2}\right) - 1 \\
 \sigma_m &= \frac{\sigma'_m}{\sigma_m'^2 + \omega'^2} & \omega_m &= \frac{-\omega'_m}{\sigma_m'^2 + \omega'^2} & \omega_{zm} &= \sec(\theta_m)
 \end{aligned} \tag{A.24}$$

con a_p guadagno della banda passante.

A.7 Filtro ellittico

$$\begin{aligned}
 A_{0m} &= 1 & B_{0m} &= 1 & A_{1m} &= 0 \\
 A_{L1} &= 0 & A_{L2} &= \sigma_R & B_{L1} &= 1 \\
 B_{L2} &= \sigma_R & B_{1m} &= -2\sigma_m & B_{2m} &= \sigma_m^2 + \omega_m^2 \\
 A_{2m} &= \omega_{zm}^2 & G_e &= 10^{0.05a_p}
 \end{aligned} \tag{A.25}$$

con

$$\begin{aligned}
 \epsilon &= \sqrt{10^{-0.1a_p} - 1} & rt &= \frac{\omega_p}{\omega_{\text{stop}}} \\
 kn &= \sqrt{\frac{10^{-0.1a_p} - 1}{10^{-0.1a_{\text{stop}}} - 1}} & \nu_0 &= \frac{K(rt)sc^{-1}(\epsilon^{-1}, kn)}{nK(kn)}
 \end{aligned} \tag{A.26}$$

$$\begin{aligned}
 \sigma_m &= \frac{cn(f(m), rt)dn(f(m), rt)sn(\nu_0, \sqrt{1 - rt^2})cn(\nu_0, \sqrt{1 - rt^2})}{1 - dn(f(m), rt)^2sn(\nu_0, \sqrt{1 - rt^2})^2} \\
 \omega_m &= \frac{sn(f(m), rt)dn(\nu_0, \sqrt{1 - rt^2})}{1 - dn(f(m), rt)^2sn(\nu_0, \sqrt{1 - rt^2})^2} \\
 f(m) &= \frac{K(rt)(2m + 1)}{n}, \quad m = 0, 1, \dots, \frac{n}{2} - 1 \quad n \text{ pari} \\
 f(m) &= \frac{K(rt)(2m + 2)}{n}, \quad m = 0, 1, \dots, \frac{n}{2} - 1 \quad n \text{ dispari} \\
 \sigma_R &= \frac{sn(\nu_0, \sqrt{1 - rt^2})cn(\nu_0, \sqrt{1 - rt^2})}{1 - sn(\nu_0, \sqrt{1 - rt^2})^2} \\
 \omega_{zm} &= \frac{1}{rtsn(f(m), rt)}
 \end{aligned} \tag{A.27}$$

con a_p guadagno della banda passante, $K(\dots)$ integrale ellittico completo di prima specie e sn , cn , dn , sc funzioni ellittiche di Jacobi. Per il calcolo di queste funzioni ho utilizzato il codice riportato in seguito ed il risultato è stato confrontato con le analoghe routine di MATLAB.

```

double TMatematica::IntegraleEllitticoCompletoPrimaSpecie(double k, double ERR_SMALL)
2 { int i; /* Loop counter. */
   int MAX_TERMS=500;
4   double A[500], B[500],
   C[500];
6   k = k * k;
   A[0] = 1;
8   B[0] = sqrt(1-k);
   C[0] = sqrt(k);
10  for(i = 1; i < MAX_TERMS ;i++)
   { A[i] = (A[i-1] + B[i-1])/2;
12   B[i] = sqrt(A[i-1]*B[i-1]);
     C[i] = (A[i-1] - B[i-1])/2;
14   if(C[i] < ERR_SMALL){ break;}

```

```

    }
16 return PI / (2 * A[i]);
    }
18 //-----
void TMatematica::sncndn(double x, double k2, double eps, double &sn, double &cn, double &
    dn) {
20 double x2, sn2, sn4, denom;
    if (fabs(x) < eps) {
22     x2 = x*x/2.0;
        sn = x*(1.0-(1.0+k2)*x2/3.0);
24     cn = 1.0-x2;
        dn = 1.0-k2*x2;
26 } else {
        sncndn(x/2.0, k2, eps, sn, cn, dn); //chiamata ricorsiva
28     sn2=sn*sn;
        sn4=k2*sn2*sn2;
30     denom = 1.0-sn4;
        sn = 2.0*sn*cn*dn/denom;
32     cn = (1.0-2.0*sn2+sn4)/denom;
        dn = (1.0-2.0*k2*sn2+sn4)/denom;
34 }
    }
}

```

A.8 Confronto con MATLAB

Per verificare i risultati della classe *TFiltro* è stato utilizzato MATLAB. A titolo di esempio riporto il confronto con due casi. Il primo riguarda un filtro passa banda di tipo Butterworth con ordine 6, frequenza centrale 500 Hz e banda 180 Hz con attenuazione -3.01 db. Come si può osservare dalla figura A.2 il risultato è identico a MATLAB. Il secondo caso (A.3) invece tratta un filtro passa alto di tipo Chebyshev di ordine 6, frequenza di taglio 30 Hz con attenuazione -3.01 dB. Anche con queste specifiche il risultato è conforme a MATLAB. Nella figura A.4 possiamo osservare un confronto tra le varie tipologie di filtri a parità di specifiche; come è noto infatti il filtro ellittico permette di ottenere la massima rapidità di transizione tra la banda passante e quella soppressa. Questo risultato è ottenuto a discapito della stabilità del guadagno, il filtro di Butterworth al contrario presenta la una transizione lenta ma il guadagno nella banda passante è particolarmente uniforme.

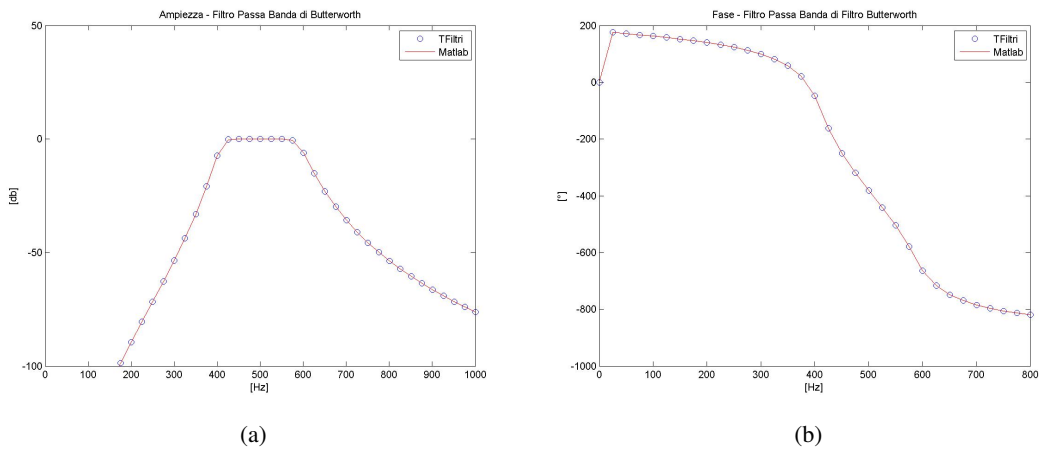


FIGURA A.2: Filtro di Butterwoth

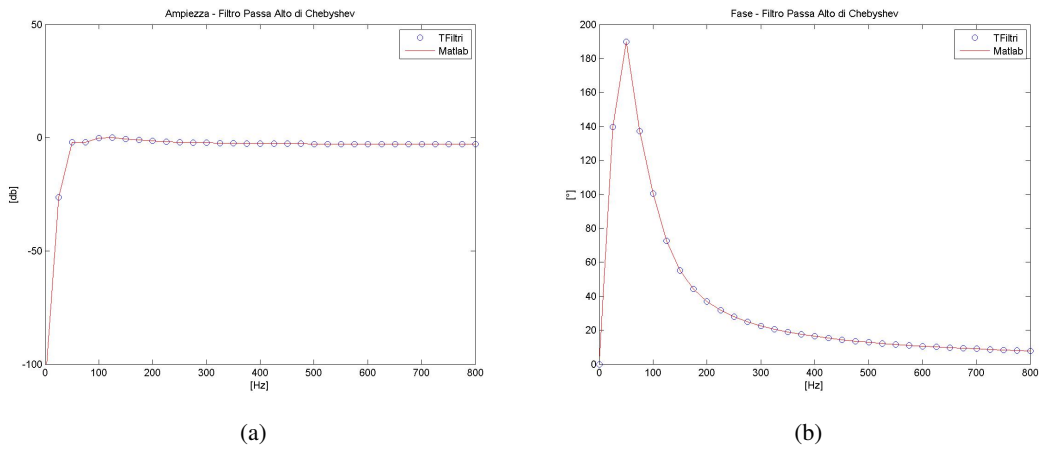


FIGURA A.3: Filtro di Butterwoth

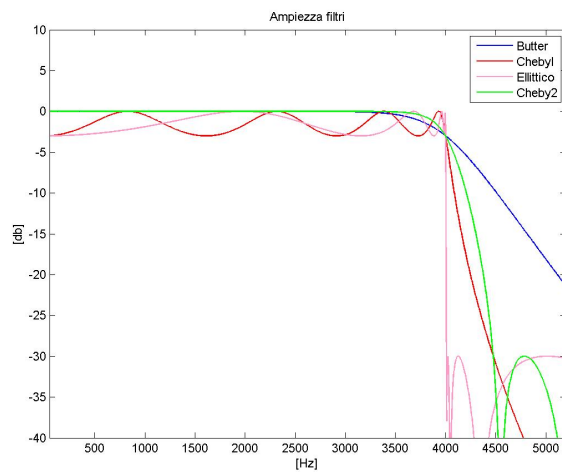


FIGURA A.4: Confronto tra tipologie di filtri a parità di specifiche.

Bibliografia

- [1] A. Cohen and R.D. Ryan. *Wavelets and multiscale signal processing*. Chapman & Hall, London, 1995.
- [2] E. N. DA C. Andrade. Bath-tub vortex. *Nature*, 197, 2 1963.
- [3] P. Andrzejewski, A. Diener, and W. Pluschkell. Model investigations of slag flow during last stages of ladle teeming. *Steel Research*, 58:547–552, 1987.
- [4] A.V. Oppenheim and R.W. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall, New Jersey, 1998.
- [5] M.S. Bartlett. A historical perspective of spectrum estimation. *Nature*, (161):686?687, 1948.
- [6] M.S. Bartlett. Periodogram analysis and continuous spectra. *Biometrika*, 37:1–16, 1950.
- [7] E. Bedrosian. A product theorem for Hilbert transforms. *Proceedings of the IEEE*, 51(5):868–869, may 1963.
- [8] S. Bittanti. *Identificazione dei modelli e controllo adattativo*. Pitagora, 1990.
- [9] R.B. Blackman and J.W. Tukey. *The Measurement of Power Spectra, from the Point of View of Communications Engineering*. Dover Publications, 1958.
- [10] J.L. Brown. A hilbert transform product theorem. *Proceedings of the IEEE*, 74(3):520–521, 1986.
- [11] J. P. Burg. Maximum entropy spectral analysis. In *37th Annual International Meeting*. Soc. of Explor. Geophysics, 1967. Oklahoma.
- [12] J.P. Burg. *Maximum entropy spectral analysis*. Stanford Exploration project. Stanford University, 1975.
- [13] Kinnor Chattopadhyay, Mihaiela Isac, and Roderick Ian Lawrence Guthrie. Modelling of non-isothermal melt flows in a four strand delta shaped billet caster tundish validated by water model experiments. *ISIJ International*, 52(11):2026–2035, 2012.

- [14] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301, 1967.
- [15] Chandrashekhar Damle and Yogeshwar Sahai. A criterion for water modeling of non-isothermal melt flows in continuous casting tundishes. *ISIJ International*, 36(6):681–689, 1996.
- [16] O. Davila, L. Garcia-Demedices, and R.D. Morales. Mathematical simulation of fluid dynamics during steel draining operations from a ladle. *Metallurgical and Materials Transactions B*, 37, 2006.
- [17] J. Durbin. The fitting of time series models. *Rev. Inst. Int. Stat.*, 28:233–243, 1960.
- [18] Robert Fildes. Methods for determining the order of an autoregressive-moving average process: a survey : Jan G. De Gooijer, Bovas Abraham, Ann Gould and Lecily Robinson, International Statistical Review 53 (1985) 30. *International Journal of Forecasting*, 2(3):384–385, 1986.
- [19] S. Gannot, D. Burshtein, and E. Weinstein. Iterative and sequential kalman filter-based speech enhancement algorithms. *IEEE Transactions on Speech and Audio Processing*, 6, 1998.
- [20] G.M. Jenkins and D.G. Watts. *Spectral Analysis and Its Applications*. Holden-Day Inc., San Francisco, 1968.
- [21] G.M. Jenkins and G.D. Watts. *Spectral Analysis of Time Series*. John Wiley & Sons, Inc., New York, 1967.
- [22] Carl Erik Grip, Hano Olof Lampinen, Magnus Lundqvist, and Sebastian Videhult. Determination of thermal stratification and emptying flow in ladles by continuous temperature measurement and tracer addition. *ISIJ International*, 36(Suppl):S211–S214, 1996.
- [23] P. Hammersmid, K. H. Tacke and H. Popper, and K. Schwerdtfeger. Vortex formation during drainage of metallurgical vessels. *Iron and Steelmaking*, 11, 1984.
- [24] I. Daubechies. *Ten Lectures on wavelets*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1992.
- [25] K. Shin and J. K. Hammond. *Fundamentals of Signal Processing for Sound and Vibration Engineers*. Wiley, 2008.
- [26] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [27] H. Kim. *Physical Modelling of Two Phase Flows in Ladle-shroud Systems*. McGill University, 1998.

- [28] Frederick King. *Hilbert Transforms Volume 1*. Cambridge University Press, 2009.
- [29] Frederick King. *Hilbert Transforms Volume 2*. Cambridge University Press, 2009.
- [30] N. Kojola, S. Takag, S. Yokoya, and Pär Jönsson. Prediction and disarming of drain sink formation during unsteady-state bottom teeming. *ISIJ International*, 49(1):1–9, 2009.
- [31] N. Kojola, S. Yokoya, and Pär Jönsson. Effect of interfacial energy on the drain sink formation height. *ISIJ International*, 49(4):463–469, 2009.
- [32] F. Kreith. *Principi di trasmissione del calore*. Liguori editore, 1988.
- [33] L. Thede. *Partical Analog and Digital Filter Design*. Artech House, Microwave Library, 2004.
- [34] B. T. Lubin and G. S. Springer. The formation of a dip on the surface of a liquid draining from a tank. *Journal of Fluid Mechanics*, 29:385–389, 1967.
- [35] M. Feldman. *Hilbert Transform Applications in Mechanical Vibration*. Wiley, 2011.
- [36] Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2008.
- [37] L. Marple. Computing the discrete-time analytic signal via FFT. *IEEE Transactions on Signal Processing*, 47(9):2600–2603, 1999.
- [38] S. L. Marple. *Digital Spectral Analysis With Applications*. Prentice Hall, 1987.
- [39] G. Mazzaferro, M. Piva, and S. Ferro. Experimental and numerical analysis of ladle teeming process. *Ironmaking & Steelmaking*, 31(6):503–508, December 2004.
- [40] Rodolfo D. Morales, Omar Dávila-Maldonado, Ismael Calderón, and Ken Morales-Higa. Physical and mathematical models of vortex flows during the last stages of steel draining operations from a ladle. *ISIJ International*, 53(5):782–791, 2013.
- [41] Kusuhiro Mukai, Liangcai Zhong, and Masafumi Zeze. Interfacial properties of molten low carbon steel containing ti, nb or b in relation to the behavior of fine particles in continuous casting process. *ISIJ International*, 46(12):1810–1816, 2006.
- [42] R.P. Numes, J.A.M. Pereira, A.C.F. Vilela, and F.T.V. Der Laan. Visualisation and analysis of the fluid flow structure inside an elliptical steelmaking kadle through image processing techniques. *Journal of Engineering Science and Technology*, 2, 2007.
- [43] Öz. Yilmaz. *Seismic data analysis - Volume I*. Society of Exploration Geophysicists, 2001.
- [44] Yuhua Pan and Bo Bjoumlrkman. Numerical analysis on the similarity between steel ladles and hot-water models regarding natural convection phenomena. *ISIJ International*, 42(1):53–62, 2002.

- [45] A.A. Pavelyev and A.A. Shtarev. Effect of sinkhole position symmetry on the formation of nonstationary vortex funnels. *Fluid Dynamics*, 40(5):829–834, 2005.
- [46] M. Piva, M. Iglesias, P. Bissio, and A. Calvo. Experiments on vortex funnel formation during drainage. *Physica A: Statistical Mechanics and its Applications*, 329(1-2):1–6, November 2003.
- [47] J. Richaud, W. Chung, and J. Rogler. Enhanced ladle draining by ladle bottom design optimization. Metec InSteel Conference, Düsseldorf, July 2011.
- [48] E.A. Robinson. A historical perspective of spectrum estimation. *Proceedings of the IEEE*, 70(9):885–907, Sept 1982.
- [49] R. Sankaranarayanan and R.I.L. Guthrie. Slag entraining vortexing funnel formation during ladle teeming: similarity criteria and scale-up relationships. *Ironmaking & Steelmaking*, 29(2):147–153, April 2002.
- [50] Ascher H. Shapiro. Bath-tub vortex. *Nature*, 196, 12 1962.
- [51] Pavan Kumar Singh. Scale model experiments and numerical study on a steel teeming process. Master’s thesis, University of Kentucky, 2004.
- [52] Petre Stoica and Randolph L. Moses. *Spectral Analysis of Signals*. Prentice Hall, 2005.
- [53] DaPeng Tan, ShiMing Ji, PeiYu Li, and XiaoHong Pan. Development of vibration style ladle slag detection methods and the key technologies. *Science China Technological Sciences*, 53(9):2378–2387, 2010.
- [54] K. Ramamurthi; T. John Tharakan. Intensification of a vortex during free draining. *The Canadian Journal of Chemical Engineering*, 73, 1995.
- [55] M. Trefethen LLOYD, R. W. Bilger, P. T. Fink, R. E. Luxton, and R. I. Tanner. The bath-tub vortex in the southern hemisphere. *Nature*, 207, 9 1965.
- [56] John W. Tukey. The sampling theory of power spectrum estimates. In *Symposium on Applications of Autocorrelation Analysis to Physical Problems (NAVEXOSP-735)*, pages 47–67, Washington, DC, USA, 1950. Office of Naval Research.
- [57] P.P. Vaidyanathan. *The Theory of Linear Prediction*. Synthesis Lectures on Engineering Series. Morgan & Claypool, 2008.
- [58] Saeed V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons, 2006.
- [59] V.N. Katsikis. *MATLAB: A Fundamental Tool for Scientific Computing and Engineering Applications*. InTech, 2012.

-
- [60] W.B. Davenport. *Probability and Random Processes*. Academic Press, New York, 1970.
- [61] P. D. Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15:70–73, 1967.
- [62] N. Wiener. *Generalized Harmonic Analysis*. Contributions (Massachusetts Institute of Technology. Department of Mathematics). Massachusetts Institute of Technology, 1930.
- [63] N. Wiener. *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*. John Wiley & Sons, Inc., New York, 1949.
- [64] W.K. Pratt. *Digital Image Processing*. John Wiley & Sons, Inc., New York, 2004.
- [65] Qiao-Nian Zhou and W. P. Graebel. Axisymmetric draining of a cylindrical tank with a free surface. *Journal of Fluid Mechanics Digital Archive*, 221, 1990.