



Università La Sapienza di Roma

FACOLTÀ DI INGEGNERIA

Corso di laurea in Ingegneria delle Telecomunicazioni
Tesina per il corso di "Schiere di Sensori"

Wideband & NarrowBand BeamForming per Reti di Sensori

Studente:

Marco Giovinazzi

Matricola: 788957

Professore:

Prof. Elio Di Claudio

Anno accademico 2004 - 2005

INDICE

1. <u>Introduzione</u>	<u>pag. 3</u>
2. <u>La Formulazione del Problema</u>	<u>pag. 6</u>
2.1. <u>Modello di segnale tempo-continuo</u>	<u>pag. 6</u>
2.2. <u>Modello di segnale tempo-discreto</u>	<u>pag. 7</u>
2.3. <u>Modello di segnale nel dominio della frequenza</u>	<u>pag. 8</u>
2.4. <u>BeamForming: concetti</u>	<u>pag. 9</u>
3. <u>Differenti metodologie di BeamForming</u>	<u>pag. 10</u>
3.1. <u>Beamforming Adattativo a Banda Stretta</u>	<u>pag. 10</u>
3.2. <u>Beamforming Adattativo a Banda Larga</u>	<u>pag. 13</u>
3.3. <u>Matched Field Beamforming</u>	<u>pag. 16</u>
3.4. <u>Steered Adaptive Beamforming</u>	<u>pag. 18</u>
3.4.1. <u>Minimum Variance STBF</u>	<u>pag. 20</u>
3.4.2. <u>Maximun Likelihood STBF</u>	<u>pag. 22</u>
4. <u>Applicazioni per le reti di sensori</u>	<u>pag. 26</u>
5. <u>Appendice: Lo strumento "Acoustic Field ToolBox"</u>	<u>pag. 30</u>
6. <u>Bibliografia</u>	

1. Introduzione al BeamForming

Il *Beamforming* è un tipo particolare di filtraggio spaziale che opera sull'output di un array di elementi (sensori) al fine di aumentarne la capacità di rilevazione di un fronte d'onda coerente, separandolo dal rumore di fondo e dalle interferenze. In pratica il beamforming "punta" l'array di sensori in modo diretto su un particolare campo spaziale, aumentandone la capacità di discriminazione dei segnali da esso provenienti. La direzione di puntamento generalmente è detta MRA (*Maximum Response Angle*). Nella figura 1.1 è rappresentato un possibile scenario in cui vengono utilizzati dei sensori in acqua (*hydrophone sensors*, o *staves*).

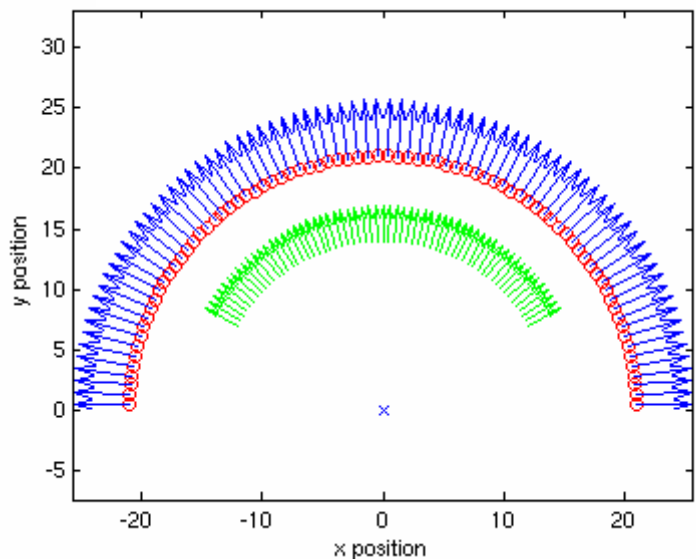


fig. 1.1 – Posizione degli staves rispetto all'MRA

L'obiettivo principale del beamformer, quindi, è quello di sommare gli elementi diretti e indiretti (*multipath*) di un segnale di interesse (SOI – *Signal of Interest*) proveniente da una sorgente esterna all'array in modo da focalizzare la schiera su questo (nella direzione MRA).

La risposta dell'array viene riportata generalmente in grafici polari dove gli angoli sono l'offset dalla direzione MRA e i radianti la magnitudine della risposta (in dB) nella specifica direzione (fig. 1.2 e 1.3).

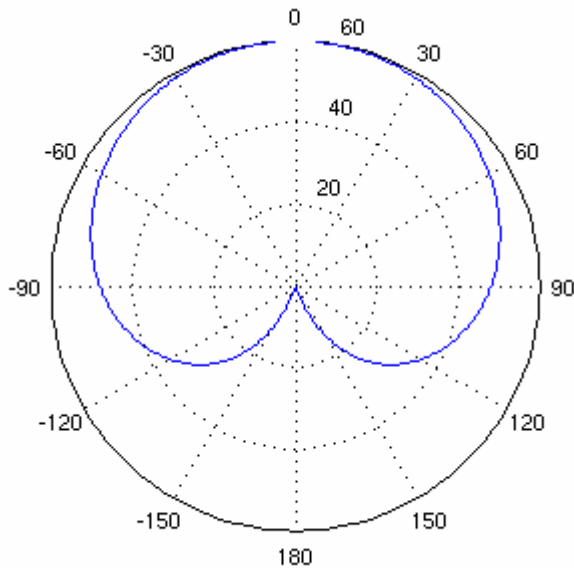


fig. 1.2 – Risposta della schiera

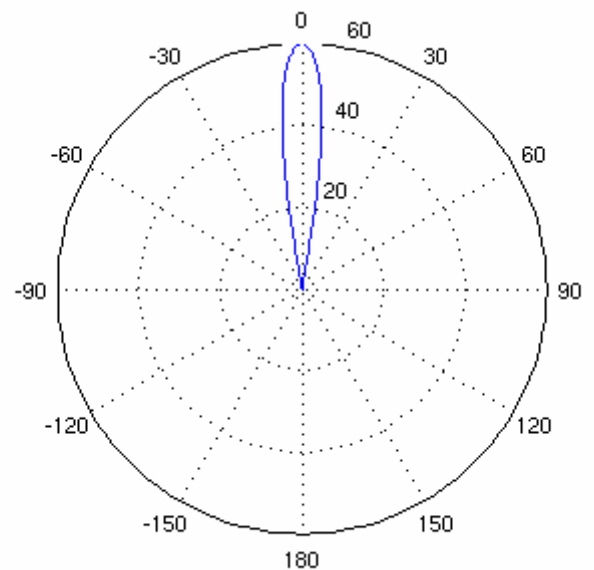


fig. 1.3 – Risposta della schiera orientata

Il *Matched Field Beamforming* (MFB) invece, è una generalizzazione del beamforming semplice nel quale gli *steering vectors* (risposte vettoriali del sistema relative al SOI, sia la sua parte diretta, sia le parti che si propagano per multipath) sono modellati come soluzione delle equazioni d'onda in un ambiente (*environment*) determinato.

L'evoluzione del filtraggio STAP (*Space-Time Array Processing*) in questo tipo di scenari ha portato non solo all'uso di nuove e più forti rappresentazioni multicanale dei segnali, ma anche alla nascita di circuiti adattativi per la loro analisi e sintesi.

Gli algoritmi che si sono sviluppati negli ultimi trenta anni per questi scopi hanno seguito principalmente due strade:

- Modellizzazione generica del campo (che trascura parametri importanti della propagazione in scenari attivi) unita all'uso di criteri di ricerca dell'ottimalità statistica seguendo la Teoria dell'Informazione.
- Utilizzo di modelli analitici o numerici della propagazione ondulatoria, molto accurati e precisi, ma insufficienti sul fronte dell'analisi statistica dei segnali irradiati dalle sorgenti.

Negli ultimi anni, però, le due strade hanno cominciato ad unirsi in nuovi algoritmi che non prescindono dal contesto particolare in cui il segnale si propaga (adattivi) riuscendo comunque a sfruttare i criteri matematici di ottimalità.

Nel seguito del documento questi concetti saranno chiariti congiuntamente alla caratterizzazione dei modelli utilizzati e delle principali tecniche di beamforming.

In ogni modo, le tecniche beamforming oggi hanno notevole importanza nell'array processing, e vengono utilizzate in un numero molto alto di applicazioni pratiche (cap. 4) in congiunzione con l'uso sempre più comune di potenti processori DSP.

2. La formulazione del problema

2.1. Modello di segnale tempo-continuo

Il segnale che vogliamo prendere in considerazione si assumerà emesso da una sorgente le cui coordinate sono descritte da un vettore \mathbf{p} . La sorgente sarà inoltre considerata puntiforme, ovvero di piccole dimensioni rispetto alla risoluzione spaziale della schiera di ricezione. Dato che vale l'ipotesi di linearità della schiera, ne consegue che può essere verificato anche il principio di sovrapposibilità degli effetti. Il segnale con queste caratteristiche sarà poi catturato dalla schiera di sensori, la cui risposta a tempo discreto è assunta di tipo multicanale, convolutivo, con risposta finita.

Sotto queste ipotesi il vettore degli $(M \times 1)$ *snapshot* (osservazioni del sistema temporalmente differenti) rispecchia il modello convolutivo:

$$\mathbf{x}(t) = \int_{t_0}^{+\infty} \mathbf{h}(\tau, \mathbf{p}) s(t - \tau) d\tau + \mathbf{v}(t)$$

dove:

- $\mathbf{h}(\tau, \mathbf{p})$ è la risposta impulsiva globale dell'array di ricezione
- $s(t)$ è il segnale utile ricevuto
- t_0 è il tempo di propagazione globale tra sorgente e ricevitori
- $\mathbf{v}(t)$ è il raggruppamento di tutti i termini interferenziali (multipath, riverbero, rumore di fondo,...) statisticamente indipendenti da $s(t)$

Questo tipo di modellizzazione del segnale è efficacemente conforme ai requisiti presupposti dalle applicazioni più comuni di *array processing*, da quelle sottomarine, a quelle acustiche o ancora a quelle multimediali.

2.2. Modello di segnale tempo-discreto

Convertendo il segnale in banda base e campionandolo a periodo T (e quindi sotto le stesse ipotesi del caso tempo-continuo), il vettore discreto degli snapshot per $n = 1, 2, \dots, N$ assume la forma di un modello M -canale di tipo *FIR*:

$$\mathbf{x}(n) = \sum_{k=N_0}^{N_f} \mathbf{h}(k, \mathbf{p}) s(n-k) + \mathbf{v}(n)$$

dove il significato dei singoli elementi può essere ricavato direttamente dal caso precedente e la risposta impulsiva è trascurabile per $t > N_f T$.

Come semplificazione del modello, è possibile esprimere la precedente nella forma scomposta:

$$\mathbf{x}(n) = \sum_{k=N_0}^{N_d \geq N_0} \mathbf{h}_d(k, \mathbf{p}) \cdot s(n-k) + \sum_{k=N_{r1} \geq N_0}^{N_{r2} \geq N_{r1}} \mathbf{h}_r(k, \mathbf{p}) \cdot s(n-k) + \mathbf{v}(n)$$

I valori $\mathbf{h}_d(n, \mathbf{p})$ e $\mathbf{h}_r(n, \mathbf{p})$ fanno sì che il segnale utile irradiato sia suddiviso nelle sue componenti diretta e riflessa (a causa di fenomeni di cammini multipli e riverberazione). La seconda componente, chiaramente, avrà durata temporale maggiore, dato che parte del segnale utile sarà sottoposto a ritardo di propagazione.

Il termine $\mathbf{v}(n)$, come ugualmente definito nel caso tempo-continuo, racchiude tutte le componenti di interferenza da sorgenti vicine ma statisticamente indipendenti dal segnale utile e il rumore.

2.3. Modello di segnale nel dominio della frequenza

In frequenza la sequenza $\mathbf{x}(n)$ può essere scomposta in blocchi non sovrapposti di lunghezza J , che poi possono essere processati attraverso una DFT a M canali e con finestra J . Sotto l'ipotesi $J \gg N_c$, e cioè che la finestra sia considerevolmente maggiore del tempo di correlazione, si ha che:

1. Ogni singolo snapshot di sottobanda $\mathbf{x}(\omega_j, l)$ all'uscita della DFT è riferito alla frequenza angolare ω_j , e soddisfa il modello a banda stretta:

$$\mathbf{x}(\omega_j, l) = \mathbf{h}(\omega_j, \mathbf{p})s(\omega_j, l) + \mathbf{v}(\omega_j, l)$$

Nel quale la risposta dell'array $\mathbf{h}(\omega_j, \mathbf{p})$ appartiene ad un sottospazio di dimensione unitaria ed è comunemente chiamato *steering vector*.

2. Il filtraggio attraverso DFT può isolare i picchi nello spettro e le frequenze di riverbero, attorno alle quali il modello perde consistenza.

3. Ogni variabile $\mathbf{x}(\omega_j, l)$ può essere considerata aleatoria a valor medio nullo, ma sotto l'ipotesi di ergodicità la sua distribuzione è approssimabile ad una gaussiana.

2.4. BeamForming: Concetti

Considerato il modello di segnale appena descritto nella sua rappresentazione tempo-discreto, il beamforming consiste nella inversione (*deconvoluzione*) dello stesso. Questa operazione può essere espressa come un filtraggio FIR multicanale e assumere la forma:

$$\bar{s}(n) = s(n - D) = \sum_0^{K-1} \mathbf{w}(k)^H \mathbf{x}(n - k)$$

nella quale $\mathbf{w}^H = [\mathbf{w}(0)^H, \dots, \mathbf{w}(K-1)^H]$ è detto *vettore peso* o *beamformer*.

Questa trasformazione è presa come lineare, considerando il fatto che le eventuali non linearità presenti in essa potrebbero inficiare il modello, generando inflazioni dei parametri e conseguente perdita di generalità.

Particolari considerazioni vanno fatte sul vettore peso. Esso infatti può essere calcolato a priori e applicato al modello in modo statico, oppure essere adattato in tempo reale al segnale ricevuto. Quest'ultima tecnica prende il nome di beamforming adattivo, e presuppone la minimizzazione di una funzione di costo $E(\mathbf{w})$ per il calcolo ottimale del vettore.

In linea di principio il beamforming adattivo consente di eliminare in modo più efficace interferenze presenti a ricezione, è molto più flessibile di quello statico e richiede un minor numero di sensori sul campo.

3. Differenti Metodologie di Beamforming

Il Beamforming consiste essenzialmente nel ricavare in condizioni reali e sotto le ipotesi fatte sui segnali una copia indistorta del segnale irradiato da una sorgente e miscelato a rumore, interferenze e ritardi.

Le tipologie di beamforming sono numerose, e la loro evoluzione ha portato la teoria matematica di base ad essere sempre più accurata. Nel seguito verranno elencate e caratterizzate le principali metodologie.

3.1. *Beamforming Adattativo a Banda Stretta*

Basato sul concetto di adattatività al segnale (vedi par. 2.4), il beamforming adattativo a banda stretta si applica se la larghezza di banda analogica di $x(n)$ è molto minore del reciproco della somma della lunghezza dell'intera risposta impulsiva del canale e del tempo di correlazione del segnale utile.

In questa condizione il modello del segnale tempo-continuo si può semplificare e esprimere come:

$$\mathbf{x}(n) = [\mathbf{h}_d(\omega_0, \mathbf{p}) + \mathbf{h}_r(\omega_0, \mathbf{p})] \cdot s(n) + \mathbf{v}(n)$$

Questa espressione contiene le componenti $\mathbf{h}_d(\omega_0, \mathbf{p})$ e $\mathbf{h}_r(\omega_0, \mathbf{p})$, dette *steering vectors* (come in 2.3) e che rappresentano le risposte vettoriali relative al cammino diretto e non diretto misurate alla pulsazione di centro banda.

Dato che il segnale utile è in uno spazio di rango 1, esso può essere considerato come un rumore bianco, sotto l'ipotesi che il periodo di campionamento sia sufficientemente lungo. Ipotizzando ancora l' $\mathbf{x}(n)$ gaussiano, il vettore peso ML (*Maximum Likelihood*) corrisponde al vettore peso MV (*Minimum Variance*) e si ottiene dalla soluzione del problema:

$$\mathbf{w}_{opt} = \arg \min_{\mathbf{w}} \sum_1^N |\mathbf{w}^H \mathbf{x}(n)|^2$$

che ha $1 \leq \nu \leq M$ vincoli lineari che soddisfano il sistema sottodeterminato $\mathbf{C}^H \mathbf{w} = \mathbf{d}$.

Inoltre il vincolo di non distorsione è espresso come:

$$[\mathbf{h}_d(\omega_0, \mathbf{p}) + \mathbf{h}_r(\omega_0, \mathbf{p})]^H \mathbf{w} = 1$$

Il problema è però vincolato. Per ovviare a questo problema e renderlo non vincolato si introduce la matrice \mathbf{C}_\perp , che è il complemento ortogonale dello spazio delle colonne di \mathbf{C} , e successivamente si decompone il vettore pesi nella somma di un vettore quiescente \mathbf{w}_0 (che ancora soddisfa $\mathbf{C}^H \mathbf{w}_0 = \mathbf{d}$) e un vettore adattativo $\mathbf{C}_\perp \mathbf{w}_1$.

L'architettura che rappresenta questo sistema è detta *generalized sidelobe canceller* (Griffiths, 1982) ed è rappresentata in figura 3.1.1.

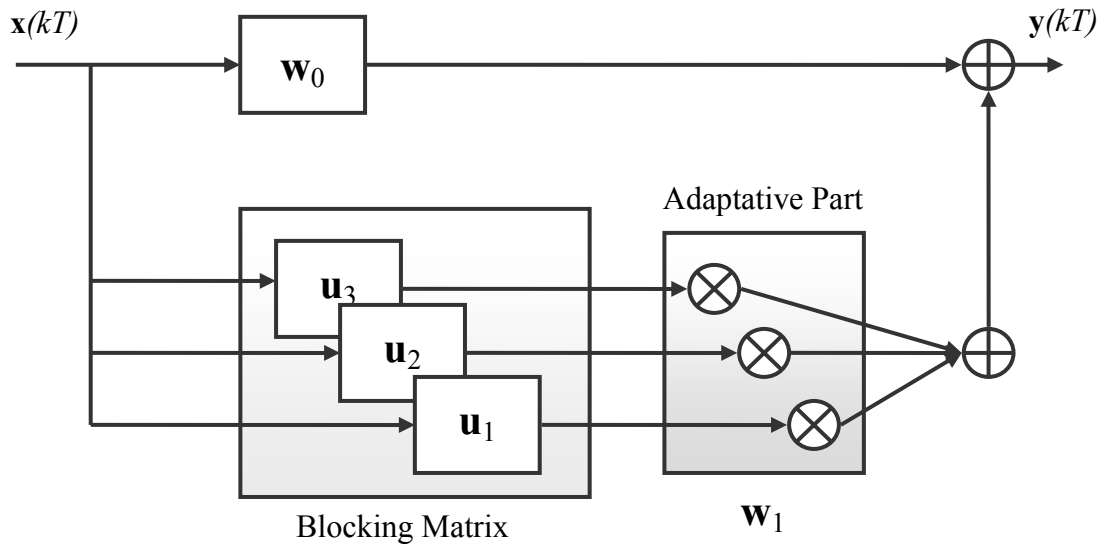


fig. 3.1.1 – Architettura del Generalized Sidelobe Canceller

Nel caso in cui il segnale ricevuto sia confuso a causa di una serie di cammini multipli (multipath), fenomeno imprevedibile, il vincolo di non distorsione perde validità. A causa di questo parte del segnale utile sarà presente all'uscita della *Blocking Matrix*, predicendo l'uscita del beamformer quiescente. Ne conseguono cancellazione di segnale utile e perdita di informazione.

Per ovviare a questo si potrebbe tentare una caratterizzazione più "realistica" del multipath (*matched field beamformer*), ma come vedremo tra poco questo presenta non pochi problemi.

Alternativa non ottima ma valida è quella di giocare sul vettore pesi inserendo vincoli lineari o quadratici, ma questo se da un lato allevia l'effetto dei cammini multipli, dall'altro riduce i gradi di libertà del sistema e la sua capacità di soppressione delle interferenze incorrelate.

3.2. Beamforming Adattativo a Banda Larga

Lo scopo del beamforming a banda larga è quello di recuperare una copia indistorta del SOI attraverso un'operazione lineare e convolutiva applicata all'output dell'array tanto in tempo quanto in frequenza. E' importante notare che la distorsione provocava sul beamformer a banda stretta solo uno sfasamento, in quanto era definita una coppia unica per tutto lo spettro di vettori di steering e pesi. Inoltre, a banda larga il rumore residuo all'uscita del beamformer è notevole, cosa che potrebbe essere distruttiva per molte applicazioni.

E' possibile definire il beamforming a banda larga sia in tempo che in frequenza. Di seguito entrambi sono velocemente definiti.

- W.B. nel dominio del tempo

Il W.B. nel dominio del tempo è classicamente definito come cascata lineare di combinazioni di copie temporalmente ritardate del SOI, soddisfacendo l'equazione:

$$y(n, \{\mathbf{w}(k)\}) = \sum_0^D \mathbf{w}^H(k) \cdot \mathbf{x}(n-k)$$

Per $n=1,2,3,\dots,N$ e dove $\mathbf{w}(k)$ rappresenta ($M \times 1$) vettori dei pesi. Introducendo poi le definizioni degli snapshot a banda larga e dei vettori dei pesi:

$$\mathbf{x}_{ST}(n) = [\mathbf{x}^T(n), \mathbf{x}^T(n-1), \dots, \mathbf{x}^T(n-D)]^T$$

$$\mathbf{w} = [\mathbf{w}^T(0), \mathbf{w}^T(1), \dots, \mathbf{w}^T(D)]^T$$

l'equazione può essere convenientemente scritta come:

$$y(n, \mathbf{w}) = \mathbf{w}^H \mathbf{x}_{ST}(n)$$

per $n = 1, 2, \dots, N$, che prende la forma schematica di un sistema di somma di copie ritardate (*delay & sum architecture*), rappresentato in figura 3.2.1.

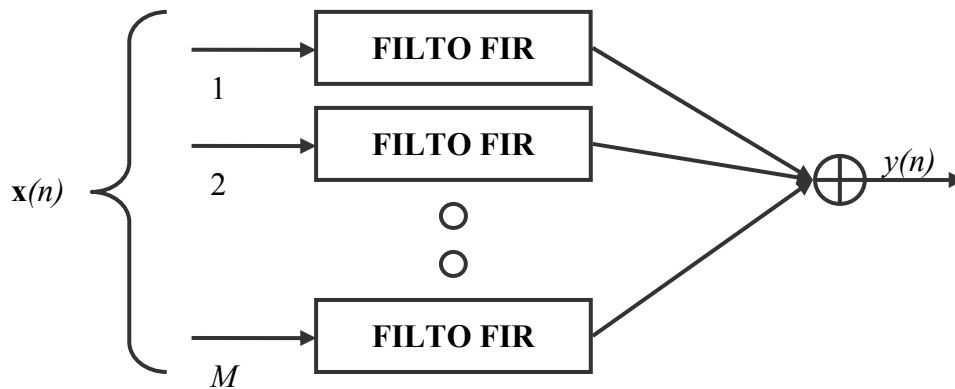


fig. 3.2.1 – Architettura del delay & sum beamformer

- WB nel dominio della frequenza

Nella classica rappresentazione del WB nel dominio della frequenza (FDBF) viene utilizzato un singolo $(M \times 1)$ vettore di pesi di sottobanda $\mathbf{w}_j (j=j_1, \dots, j_2)$ per ogni trasformata DFT al fine di ottenere la risposta $y(\omega_j, l, \mathbf{w}_j)$ ($l = 1, \dots, L$) relativa alla componente del SOI $s(\omega_j, l)$ come:

$$y(\omega_j, l, \mathbf{w}_j) = \mathbf{w}_j^H \mathbf{x}(\omega_j, l)$$

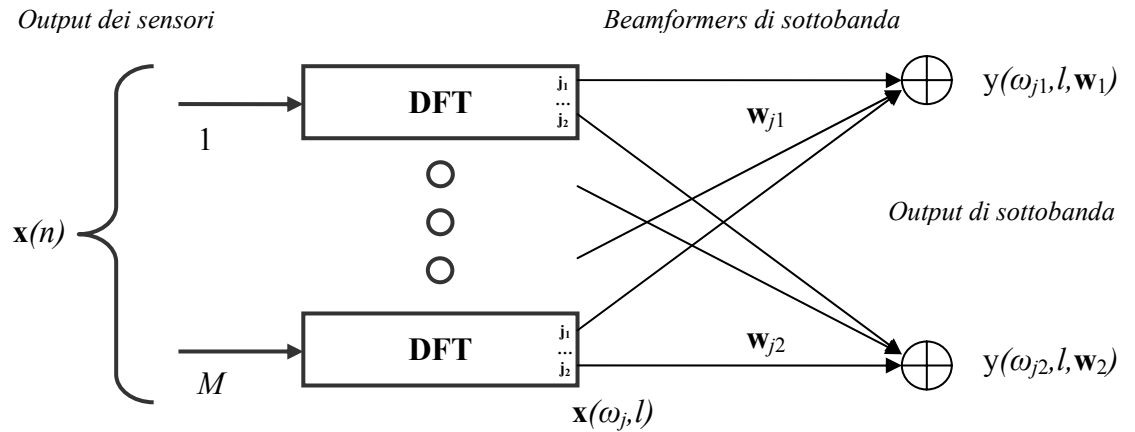


fig. 3.2.1 – Architettura classica del beamformer nel dominio della frequenza.

La schematizzazione che ne risulta è quella rappresentata in figura 3.2.1.

Dal momento che le uscite delle DFT $\mathbf{x}(\omega_j, l)$ sono assunte Gaussiane e statisticamente indipendenti attorno alle singole frequenze elaborate, il beamforming ML-FDBF coincide con il beamformer MV linearmente vincolato (LCMV) che risolve indipendentemente per ogni ω_j il problema LS:

$$\tilde{\mathbf{w}}_j = \arg \min_{\mathbf{w}_j} \frac{1}{L} \sum_1^L |y(\omega_j, l, \mathbf{w}_j)|^2$$

soggetto a $1 \leq M_c \leq M$ vincoli lineari, espressi in forma compatta come:

$$\mathbf{C}^H(\omega_j, \mathbf{p}) \mathbf{w}_j = \mathbf{f}(\omega_j, \mathbf{p})$$

essendo $\mathbf{f}(\omega_j, \mathbf{p})$ un vettore bersaglio ($M_c \times 1$) predeterminato.

3.3. Matched Field Beamforming

Il MF Beamforming può essere derivato come caso particolare del FDBF trasportato in un modello di propagazione casuale e multimodale. L'approccio utilizzato è quello di considerare la conoscenza dettagliata del $\mathbf{h}(\omega_j, \mathbf{p}, \mathbf{b})$ per localizzare il SOI in uno spazio tridimensionale.

Tuttavia in pratica le caratteristiche di propagazione possono cambiare in modo casuale (anche molto velocemente), cosa che fa deviare il $\mathbf{h}(\omega_j, \mathbf{p}, \mathbf{b})$ dal suo valore nominale. Come conseguenza la componente in uscita $y_1(\omega_j, l)$ porta con se una correlazione residua con la componente $s_1(\omega_j, l)$, e questo viene interpretato dal processo MV come una interferenza da sopprimere.

Per risolvere questo inconveniente è necessario che sia soddisfatta la condizione:

$$\mathbf{C}_{\perp}^H(\omega_j, \mathbf{p})\mathbf{h}(\omega_j, \mathbf{p}, \mathbf{b}) = 0 \quad \forall \mathbf{h}(\omega_j, \mathbf{p}, \mathbf{b})$$

Inoltre il vettore quiescente $\mathbf{w}_{0,j}$ può essere opportunamente scelto per essere proporzionale all'autovettore dominante $\mathbf{R}_{hh}(\omega_j, \mathbf{p})$.

Per sua naturale costruzione, però, questo tipo di beamforming presenta delle caratteristiche sfavorevoli che ne pregiudicano spesso la praticità. In particolare:

- La sua formulazione è essenzialmente orientata ad essere a banda stretta, e per questo non ottimale per discriminare gli sfasamenti temporali degli arrivi per identificare le riflessioni (multipath).

- Dal momento che le configurazioni modali possono cambiare sensibilmente al variare dei valori di \mathbf{p} e ω , $\mathbf{w}_{0,j}$ e $\mathbf{C}_{\perp}(\omega_j, \mathbf{p})$ devono essere ricalcolate per un numero considerevole di frequenze e posizioni (a formare una griglia di valori significativi), richiedendo una notevole capacità computazionale di base.
- Il beamformer deve essere inizializzato con un simulatore coerente di propagazione spaziale, ma questo lo rende inappropriato in applicazioni critiche. Inoltre gli attuali simulatori non forniscono una accuratezza tale da sopprimere gli errori residui sulla funzione di trasferimento del sistema, a causa del trattamento non ottimo (o a volte addirittura assente) degli errori di calibrazione del guadagno, dello sfasamento, degli accoppiamenti mutui e dello scattering.
- La capacità di soppressione di interferenze e disturbi è molto bassa (a causa dei pochi valori-autovettori liberi).
- Le applicazioni che in genere possono fare uso di tale tecnica sono quelle che hanno vincoli stringenti sulla durata delle osservazioni (come i sonar) a causa di non stazionarietà dell'ambiente. Questo vuol dire che in poco tempo devono essere elaborati molti parametri sulla base di pochi snapshot relativi alle sottobande, pregiudicando la ricostruzione ottimale del SOI e presentando notevoli problemi di calibrazione.

3.4. Steered Adaptive Beamforming

Introdotta nel 1989 da Krolik e Swingler, l'STBF è un beamformer a banda larga caratterizzato da un basso numero di parametri liberi e dall'utilizzo di un singolo vettore dei pesi per tutte le frequenze. Questo perché ogni singolo snapshot $\mathbf{x}(\omega_j, l)$ è trasformato linearmente per mezzo di una *matrice di focalizzazione* \mathbf{T}_j , unitaria per la conservazione dell'informazione.

Le caratteristiche del STBF sono notevolmente superiori a quelle dei beamformer analizzati nei paragrafi precedenti. L'STBF, infatti, è capace di minimizzare il problema del multipath grazie alle sue proprietà di decorrelazione in tempo delle componenti ritardate e l'eliminazione dallo spazio degli elementi delle stesse. In pratica il beamformer è capace di allineare lo steering vector relativo al solo cammino diretto su un vettore $\mathbf{a}_d(\mathbf{p})$ unico per tutte le frequenze, soddisfacendo l'equazione:

$$\mathbf{T}_j \mathbf{h}_d(\omega_j, \mathbf{p}) \cong \mathbf{a}_d(\mathbf{p}) \left| \mathbf{h}_d(\omega_j, \mathbf{p}) \right| e^{-j\omega_j \tau_0}$$

Inoltre la focalizzazione riesce a correggere la dispersione (se presente) del cammino diretto, rendendo il corrispondente snapshot focalizzato $\mathbf{x}_f(\omega_j, l)$ coerente con il modello:

$$\begin{aligned} \mathbf{x}_f(\omega_j, l) &= \mathbf{T}_j \mathbf{x}(\omega_j, l) \cong \\ &\cong \mathbf{a}_d(\mathbf{p}) \left| \mathbf{h}_d(\omega_j, \mathbf{p}) \right| e^{-j\omega_j \tau_0} s(\omega_j, l) + \mathbf{T}_j \mathbf{h}_r(\omega_j, \mathbf{p}) s(\omega_j, l) + \mathbf{T}_j \mathbf{v}(\omega_j, l) = \\ &= \mathbf{a}_d(\mathbf{p}) \left| \mathbf{h}_d(\omega_j, \mathbf{p}) \right| e^{-j\omega_j \tau_0} s(\omega_j, l) + \mathbf{h}_{rf}(\omega_j, \mathbf{p}) s(\omega_j, l) + \mathbf{v}_f(\omega_j, l) \end{aligned}$$

Come vantaggio di questa tecnica di beamforming la focalizzazione riesce a riallineare array di forma e dimensioni diverse, proprietà fondamentale in moltissime

situazioni di utilizzo pratico. Il contro sta invece nel fatto che le matrici di focalizzazione, per questioni geometriche, devono essere calcolate per ogni angolo o settore geometrico di interesse; in questo modo le interferenze presenti tendono ad apparire come sorgenti spazialmente estese. Come già detto poi è possibile (disponendo di un numero di sensori sufficientemente grande) utilizzare un solo vettore di pesi per tutte le frequenze.

La struttura in tempo di un possibile STBF è rappresentata in figura 3.4.1.

Lo schema risponde pienamente all'idea di base dell'STBF: visto che M sensori identici, caratterizzati da una risposta piatta e un comportamento lineare rispetto alla fase, generano M copie ritardate del SOI in uno scenario che può essere considerato non dispersivo, aggiungendo un particolare ritardo su ogni linea di uscita del sensore si possono allineare tutte le forme d'onda in tempo.

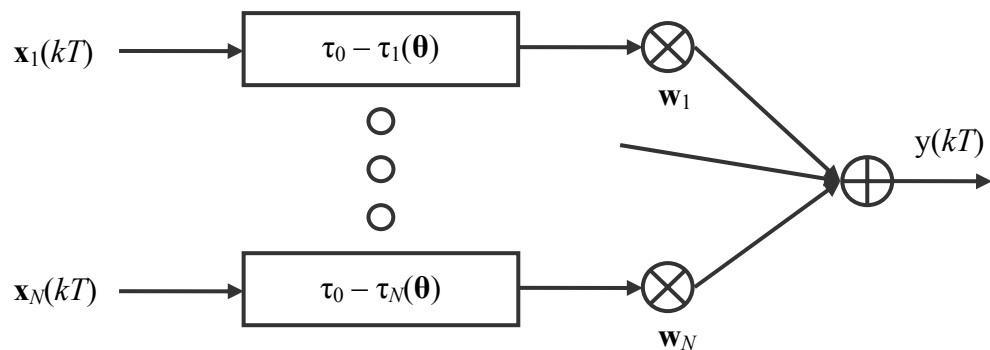


fig. 3.4.1 – Struttura in tempo del STBF

La stessa cosa si può vedere con le equazioni non vincolate del relativo generalized sidelobe canceller, che diventano:

$$\begin{aligned}
 y_0(\omega_j, l) &= \mathbf{w}_0^H \mathbf{x}_f(\omega_j, l) \\
 \mathbf{C}^H \mathbf{w}_0 &= \mathbf{d} \\
 \mathbf{y}_1(\omega_j, l) &= \mathbf{C}_\perp^H \mathbf{x}_f(\omega_j, l) \\
 y(\omega_j, l, \mathbf{w}_1) &= y_0(\omega_j, l) + \mathbf{w}_1^H \mathbf{y}_1(\omega_j, l)
 \end{aligned}$$

In queste l'uscita della blocking matrix $\mathbf{y}_1(\omega_j, l)$ contiene solo le componenti ritardate del multipath se si impone il vincolo di non distorsione $\mathbf{a}_d(\mathbf{p})^H \mathbf{w}_0 = 1$. In pratica se il ritardo rispetto al cammino di retto supera il tempo di correlazione del segnale utile, il multipath stesso viene cancellato come se fosse un'interferenza incorrelata.

3.4.1. Minimum Variance STBF

Prima evoluzione del STBF classico, applica a questo il criterio della minima varianza riformulato come:

$$\begin{aligned}
 \tilde{\mathbf{w}}_{1MV} &= \arg \min_{\mathbf{w}_1} \sum_{j_1}^{j_2} \tilde{\sigma}_j^2(\mathbf{w}_1) \\
 \tilde{\sigma}_j^2(\mathbf{w}_1) &= \frac{1}{L} \sum_1^N |y(\omega_j, l, \mathbf{w}_1)|^2
 \end{aligned}$$

Per cui la formulazione in tempo dell'MV-STBF è:

$$\tilde{\mathbf{w}}_{1MV} = \underset{\mathbf{w}_1}{\operatorname{argmin}} \frac{1}{N} \sum_1^N |y(n, \mathbf{w}_1)|^2$$

Il MV-STBF risolve un sistema LS singolo contenente $N \approx (j_2 - j_1 + 1)L$ equazioni in $M_b = M - M_c$ variabili complesse, mentre il generico MV-FDBF deve ottimizzare $(j_2 - j_1 + 1)M_b$ parametri liberi complessi utilizzando lo stesso numero di equazioni. Questa economizzazione dei parametri però porta a casi in cui c'è cancellazione del segnale utile, quando ci si avvicina troppo al limite temporale dato dal tempo di correlazione o per record di dati da osservazioni molto brevi.

Questo comportamento non ottimo del MV-STBF è particolarmente accentuato in presenza di sorgenti colorate. Infatti, se restringiamo l'attenzione sull'analisi spettrale del sistema si può vedere che nel caso in cui il sistema sia puntato sull'analisi di sorgenti colorate il termine $y(\omega_j, l, \mathbf{w}_{1MV})$ può essere ancora considerato gaussiano per una DFT di misura J sufficientemente larga, ma contiene in modo diffuso per ogni singola frequenza delle varianze differenti. In questo caso da una parte il sistema non calibrato presenta dei punti in cui l'SNR viene drasticamente ridotto, soprattutto nei punti concavi della risposta in frequenza del SOI, cosa che può pregiudicare la validità del segnale in uscita in questi punti; dall'altra il non corretto comportamento del modello può amplificare in modo sconveniente il segnale in presenza di picchi spettrali, cosa che non permette la ricostruzione del segnale utile.

Anche se un pre-trattamento del segnale di tipo "pre-whitening" può risolvere il problema sopprimendo l'interferenza nei punti di vallata del SOI, ottimizzando l'SNR per mascherare i punti di interferenza residua intorno ai picchi spettrali, e allo stesso tempo ridurre il tempo di correlazione N_s (e quindi del multipath), questo approccio non è di pratica applicazione in sistemi di tipo multicanale, in quanto lo spettro del SOI non può essere predetto dall'output del STBF.

3.4.2. Maximum Likelihood STBF

Per risolvere gli inconvenienti appena descritti del MV-STBF, è stato introdotto nel 2003 (Da Di Claudio) un nuovo STBF basato sulla stima ML del vettore adattativo, derivato da un modello stocastico dell'uscita $y(\omega_j, l, \mathbf{w}_1)$.

Il modello a cui si appoggia questo processo di beamforming è simile ai precedenti, ma sfrutta alcuni paradigmi relativi alle reti neurali che apportano una profonda innovazione. In particolare, ogni singola sequenza $y(\omega_j, l, \mathbf{w}_1)$ con $l=1, \dots, L$ può essere considerata avvalendosi del teorema del limite centrale come la realizzazione di un processo circolare gaussiano bianco, ergotico, complesso e a media nulla, caratterizzato da una varianza di sottobanda pari a $\sigma_j^2(\mathbf{w}_1)$. Inoltre il singolo processo relativo ad una sottobanda può ancora essere mutuamente visto come statisticamente indipendente per $J \gg N_c$.

Questo modello stocastico è definito nelle reti neurali come mistura di Gaussiane (MoG). La verosimiglianza associata da massimizzare allora diviene (a meno di costanti moltiplicative):

$$F(\mathbf{w}_1, \sigma_j^2) = \prod_{l=1}^L \prod_{j_1}^{j_2} \frac{1}{\sigma_j^2} e^{-\frac{|y(\omega_j, l, \mathbf{w}_1)|^2}{\sigma_j^2}}$$

Questo funzionale è interpretabile come la definizione di una architettura neurale di tipo RBF. Ma questo porterebbe ad un iviluppo non realizzabile da un punto di vista pratico, essendo questa particolare rete neurale notoriamente di difficile inizializzazione e ottimizzazione in tempi ragionevoli.

Per superare questo inconveniente si è passato ad una log-verosimiglianza risultante da un procedimento di concentrazione perfetta dei parametri, che minimizza il funzionale ML rispetto a ciascuna varianza σ_j^2 . Il risultato è una log-verosimiglianza negativa globale che (a meno di costanti moltiplicative) assume la forma:

$$F_1(\sigma_j^2, \mathbf{w}_1) = \sum_{j_1}^{j_2} \left[\log(\sigma_j^2) + \frac{1}{L} \sum_{l=1}^L \frac{|y(\omega_j, l, \mathbf{w}_1)|^2}{\sigma_j^2} \right]$$

Minimizzando ulteriormente rispetto a σ_j^2 si ottiene il funzionale ML concentrato:

$$\begin{aligned} \sigma_j^2 &= \tilde{\sigma}_j^2(\mathbf{w}_1) \\ E_{ML}(\mathbf{w}_1) &= \sum_{j_1}^{j_2} \log[\tilde{\sigma}_j^2(\mathbf{w}_1)] \end{aligned}$$

L'architettura risultante allora diviene quella di un *perceptron* a due strati in cui i pesi del primo strato sono precalcolati direttamente a partire dal modello di propagazione del cammino diretto e costituiscono l'insieme delle matrici di focalizzazione. I pesi dello strato di uscita (vettore \mathbf{w}_1) sono ottenuti minimizzando il funzionale ML concentrato, che però in certi casi può essere non lineare, non convesso, lontano dal minimo assoluto o addirittura non limitato inferiormente!

Per superare questo grave inconveniente di non-limitazione inferiore occorre regolarizzare lo sviluppo affinché il funzionale possa essere gestibile. In pratica è necessario che sia $\tilde{\sigma}_j^2(\mathbf{w}_1) > 0$ in ogni sottobanda. Questo può essere imposto regolarizzando opportunamente ciascuna matrice:

$$\mathbf{Y}_j = \frac{1}{\sqrt{L}} \begin{bmatrix} \mathbf{y}_1(\omega_j, l) & \dots & \mathbf{y}_1(\omega_j, L) \\ -y_0(\omega_j, l) & \dots & -y_0(\omega_j, L) \end{bmatrix}$$

in modo da far sì che questa abbia rango pieno. Questa soluzione può essere raggiunta con la tecnica di diagonal loading, che consiste nell'aggiunta di una particolare e orrimale quantità di rumore sintetico ed isotropico di deviazione standard ε_j a ciascuna sottobanda. In pratica quello che si fa è introdurre una matrice regolarizzata degli snapshot come:

$$\mathbf{Y}_j(\varepsilon_j) = \begin{bmatrix} \mathbf{Y}_j & \varepsilon_j \mathbf{I} \end{bmatrix}$$

Il valore di ε_j non è particolarmente critico nelle applicazioni, perciò la sua scelta può essere anche non rigorosa. Inoltre l'ulteriore imposizione sul funzionale del vincolo quadratico di *ridge regression*:

$$|\mathbf{w}_1|^2 < \gamma^2$$

permette di impedire la cancellazione involontaria del segnale utile realizzando un *diagonal loading adattivo*.

Lo step successivo è stato quello di ottimizzare il funzionale vincolato dalla ridge regression, problema non lineare, complesso che richiede algoritmi dedicati di

ottimizzazione vincolata del secondo ordine (*Newtoniano*), e quindi estremamente delicato. In più l'eccessivo malcondizionamento delle matrici dei dati non permetteva l'adozione di un metodo di tipo gradiente, che di certo non avrebbe portato a soluzioni di minimo (globale) ottimo. Per questo motivo si è ricorso alla tecnica di *backpropagation* nello spazio dei neuroni, fondamentale nell'apprendimento rapido delle reti neurali ed equivalente ad un algoritmo di newton modificato. Matematicamente, per il beamformer ML, si definisce il vettore gradiente nello spazio dei neuroni come:

$$\nabla_z E_{ML}(\mathbf{w}_1) = -\frac{2}{L} \begin{bmatrix} \frac{\mathbf{e}_{j_1}(\mathbf{w}_1)}{\tilde{\sigma}_{j_1}^2(\mathbf{w}_1)} \\ \dots \\ \frac{\mathbf{e}_{j_2}(\mathbf{w}_1)}{\tilde{\sigma}_{j_2}^2(\mathbf{w}_1)} \end{bmatrix}$$

dove $\mathbf{e}_j(\mathbf{w}_1)$ è il vettore degli errori delle equazioni relative alla pulsazione ω_j .

Anche se è necessario un sofisticato preconditionamento delle metriche di sistema al fine di evitare minimi locali e garantire la convergenza, e anche se il costo computazionale medio dell'intero algoritmo ML-STBF è pari al doppio di quello del suo predecessore MV-STBF, le prestazioni ottenibili sono di assoluta rilevanza rispetto ad ogni altra tecnica di beamforming esistente.

In particolare i risultati ottenuti da sperimentazioni del beamformer ML sono stati sorprendenti sia nella deverbrazione dei segnali a larga banda (prestazioni sorprendenti in esperimenti con segnali acustici da sorgenti incorrelate in array lineare a 10 sensori), dai quali si evince la fortissima pulizia degli spettri operata dal sistema e la resistenza agli errori di puntamento, sia nella cancellazione del multipath in esperimenti

di tipo SACLANT, sui quali il sistema è riuscito ad isolare ed evidenziare in modo chiaro i vari modi della guida d'onda acustica individuando correttamente la sorgente del segnale.

In conclusione, il beamformer ML ad oggi costituisce lo stato dell'arte nel suo campo, rappresentando un forte salto rispetto alle tecniche precedenti. Le sue prestazioni, come descritto nei paragrafi precedenti, sono molto superiori a qualsiasi altra tecnica di beamforming, e soprattutto quelle del MV-STBF non sono paragonabili a queste nella deverbazione, cancellazione del multipath e intelligibilità complessiva dei segnali elaborati.

4. Applicazioni per le reti di sensori

Come già detto in questo documento, il beamforming è un elemento teorico essenziale nelle reti di sensori. Per chiarezza, possiamo dire che il problema generale parte dalle informazioni disponibili, che sono il modello di propagazione del campo, la geometria dell'array e la caratterizzazione del segnale proveniente dai sensori, e si risolve negli obiettivi principali, che sono:

- La ricostruzione dei segnali emessi dalle sorgenti
- La stima della distribuzione dell'array
- La calibrazione dell'array

- Le misure relative al campo

I principali temi di ricerca su queste applicazioni si snodano invece su:

- La costruzione di modelli flessibili e realistici per la propagazione dei segnali e per l'array, per analisi teorica e simulazione.
- La calibrazione sul campo dell'array, dove i parametri diventano "reali" e dove sono messi a dura prova i modelli statistici di propagazione.
- La sintesi ottimale di filtri spaziali che consentano la ricezione (o la trasmissione) corretta dei segnali.
- La stima della posizione e dello spettro di sorgenti esterne per mezzo di array calibrati.

Il beamforming entra in gioco essendo per sua stessa natura un filtraggio spaziale che sfrutta una analogia formale tra array processing, analisi di fourier e circuiti digitali. In questo documento sono state analizzate le principali tecniche di beamforming a banda stretta o a banda larga; è importante notare che a seconda del contesto si utilizza l'una o l'altra con le rispettive varianti.

Un esempio di beamforming a banda stretta tipico è quello in trasmissione: la formazione del fasci in trasmissione permette di concentrare l'energia del segnale nelle direzioni utili e limitare l'emissione e l'interferenza. La possibilità, poi, di emettere fasci multipli in simultanea (space-time coding) è altamente utilizzata in reti

di trasmissione ad alta capacità, come sonar e radar polifunzionali. Le applicazioni tipiche che utilizzano il NB in trasmissione sono, per il campo civile, i sistemi di trasmissione punto-multipunto wireless (ancora in uno stato relativa gioventù), mentre per il campo militare sono la teleguida missilistica, i radar phased array e la maggior parte dei sonar attivi.

Esempio tipico è lo schermo EM attivo (Radar Jammer/Deceiver), dove il rumore interferente viene concentrato sui ricevitori ostili per mascherare o cancellare le proprie emissioni o gli echi radar di ritorno (invisibilità alla strumentazione), come rappresentato schematicamente in figura 4.1.

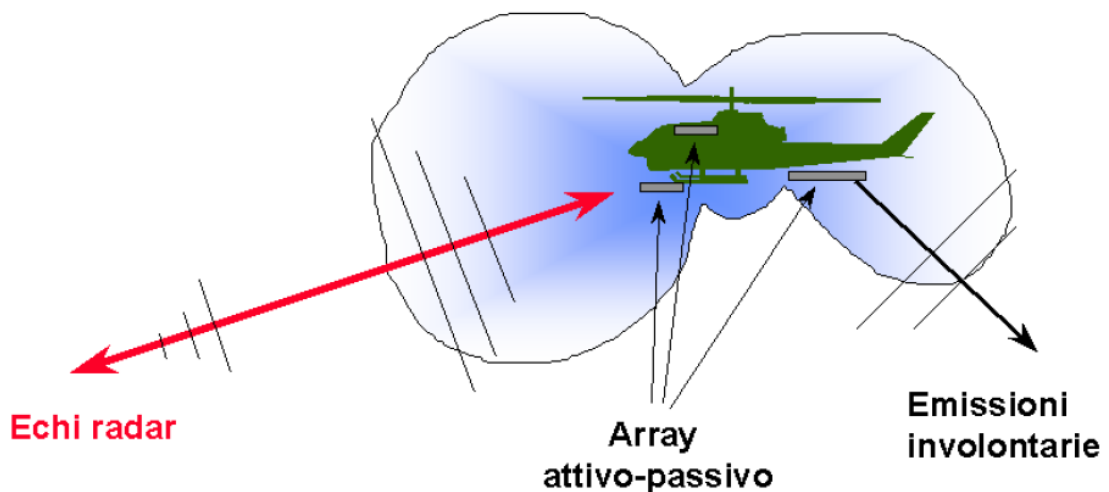


fig. 4.1 – Schermo EM attivo (Radar Jammer/Deceiver)

Per il beamforming a banda larga, invece, le applicazioni tipiche sono quelle di rilevazione sonora sottomarina (sonar), nel monitoraggio e analisi di fenomeni sismici, e nei sistemi acustici.

Le attuali architetture sono sia di tipo analogico e digitale e operano in tempo e/o in frequenza, anche se il modello di array di sensori subisce una pesante influenza dalla frequenza operativa.

Ulteriori applicazioni di interesse in cui il beamforming riveste un ruolo essenziale sono:

- Diagnostica biomedica avanzata.
- Prospezione sismica, vulcanologia, oceanografia, radio-astronomia, satelliti per telecomunicazioni, analisi di campi a scopo di ricerca.
- Registrazione audio, compatibilità elettromagnetica, teleconferenza, telesorveglianza e telecontrollo.
- Intrattenimento avanzato, apparecchiature commerciali audio/video alta fedeltà.

Tutte queste apportano un contributo essenziale alla ricerca, estendendo i campi applicativi e permettendo lo studio di contesti eterogenei.

In definitiva, anche se di certo i settori più sviluppati sono quelli della ricerca militare e delle applicazioni commerciali ad alto livello, il filtraggio spaziale applicato e congiunto a reti di sensori è sempre più presente in moltissimi campi operativi sia per la sua grande versatilità ed elasticità di utilizzo, sia per le attuali innovazioni che lo rendono più semplice ed economico da realizzare.

5. Appendice: lo strumento Acoustic Field Toolbox

Lo strumento AFT (Acoustic Field Toolbox) è un insieme di script compilati in fortran studiati per rappresentare una modellizzazione esatta dell'ambiente sottomarino e quindi per studiare la diffusione di segnali nello stesso. Gli usi che può avere questo framework possono essere molteplici, e vanno dal semplice studio di fondale al test di meccanismi di tipo sonar o simili.

Estremamente completo, AFT era inizialmente composto da un paio di routine di calcolo (KRAKEN e KRAKENC, il cui nome deriva dal leggendario mostro sottomarino presente nella letteratura scandinava) e qualche routine di visualizzazione dei risultati analizzati. Ad oggi, sono stati aggiunti molti moduli sia di calcolo che di rappresentazione/trattamento dei risultati. In particolare la struttura rappresentata in figura 5.1 si compone di:

- I moduli di processamento del modello sono KRAKEN (sia *normal mode* che *varianti*), BELLHOP come strumento di *tracing ray/beam*, SCOOTER come FFP (integratore) e SPARC come modello di FFP nel dominio del tempo.
- PLOTSSP produce una rappresentazione del profilo suono/velocità dai dati di *enviroment* (ENVFILE) prodotti dai modelli. PLOTFIELD e PLOTSLICE invece utilizzano gli *shade files* (sempre in output dei modelli) per rappresentare il primo l'attenuazione nelle direzioni di propagazioni sia orizzontali (*range*) sia verticali (*depth*), il secondo per plottare una “*slice*” del campo attorno alla profondità del ricevitore.

- Include altre routines di plotting e processamento per calcolare la probabilità di detection o un raggio di detection di sorgenti.
- TONRL e TOSAC permettono di convertire i dati dal formato NRL (kraken shade files) al formato SACLANTCEN, per permettere l'interoperabilità dei modelli.

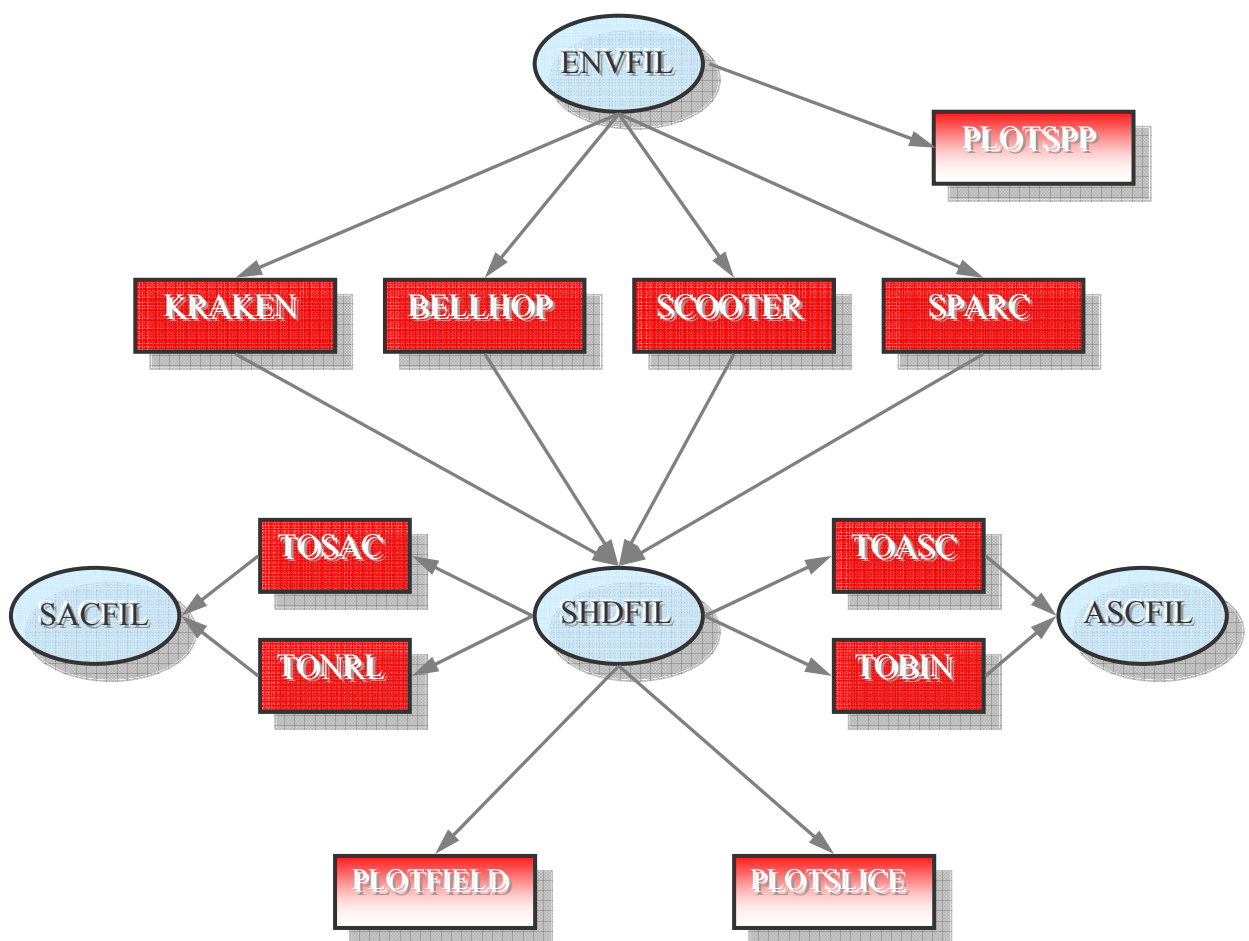


fig. 5.1 – Struttura dell'Acoustic Field Toolbox

Il cuore del sistema è comunque rappresentato da KRAKEN, lo strumento di sviluppo di calcolo basato sull'omonimo modello. Le sue attuali varianti sono rappresentate da KRAKENC e KRAKENL, più sofisticate del modello originario. Per

completezza la sua struttura è riportata in figura 5.2. Il funzionamento del programma può essere riassunto come:

1. KRAKEN o KRAKENC o KRAKENL per calcolare i modi.
2. PLOTTRL o PLOTTRD per plottare la transmission loss sul range o sulla profondità.
3. PLOTMODE calcola il singolo modo di propagazione, mentre PLOTGRN la funzione di Green associata.

In aggiunta il comando FIELD somma i modi per calcolare il campo di pressione (anche nella sua variante FIELD3D) e il comando PLOTTRAYXY ne descrive la rifrazione orizzontale.

L'uso dell'AFT è intuitivo e parzialmente assistito. La descrizione delle singole funzioni può essere trovata nel file KRAKEN.PDF incluso nel pacchetto compresso del framework e seppure in bozza riporta tutte le principali caratteristiche del programma.

Come particolare studio in questa tesina, l'import dei dati in matlab relativi sia ai file di environment, di modes e di shades può essere fatto tramite le funzioni "readenv.m", "read_modes_bin.m" e "read_shd_bin.m". Queste tuttavia non permettono una gestione ordinata dei dati. Per questo motivo è stata sviluppata una routine appropriata chiamata kraken_import.m che permette la corretta gestione e memorizzazione di questi dati nel workspace di matlab, descritta e riportata più avanti.

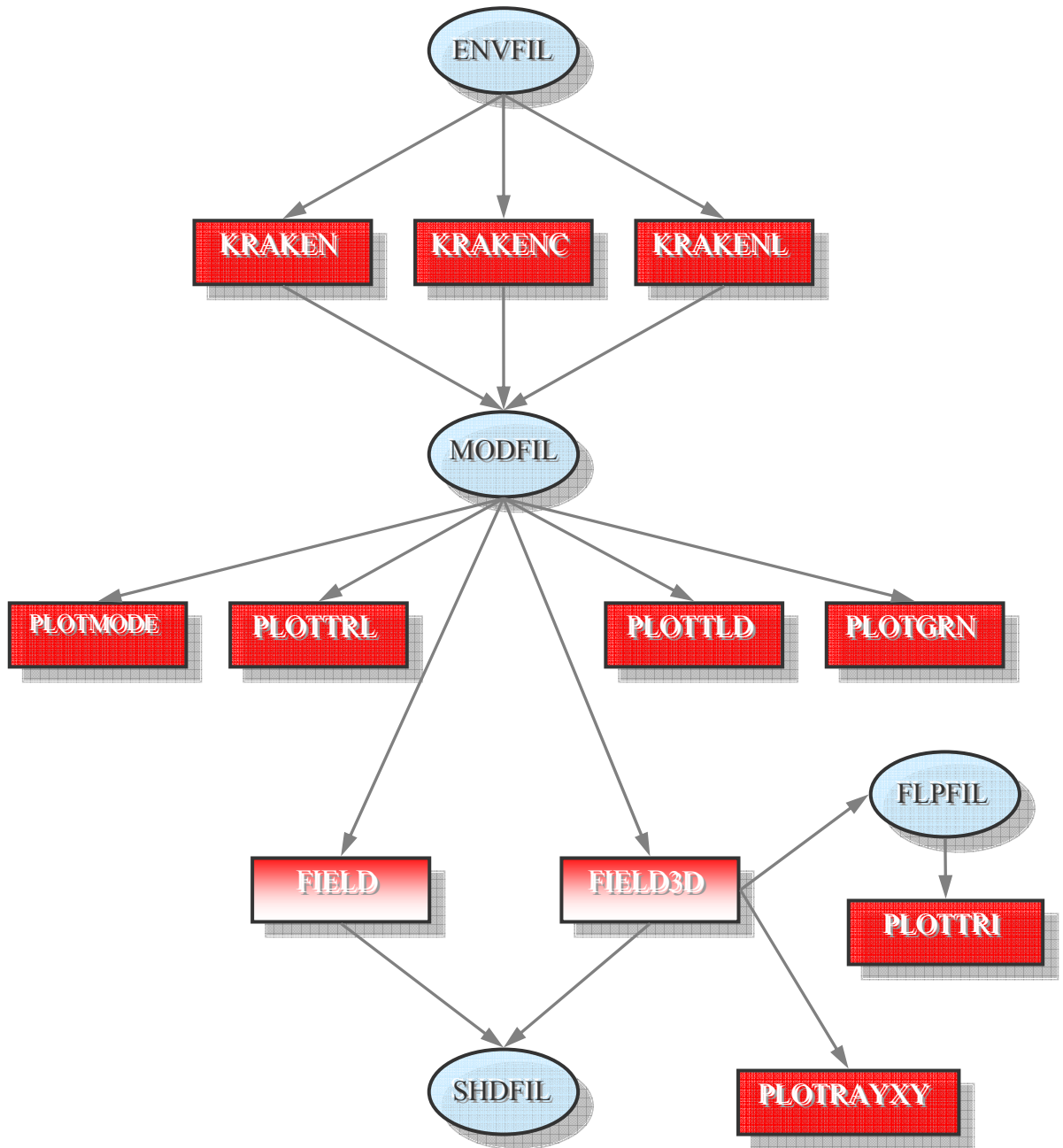


fig. 5.2 – Struttura della routine di calcolo kraken

Il formato dei files utilizzati dal framework è semplice. Le estensioni identificano in modo univoco i file in input o in output delle singole routines. In particolare:

- .env – files di environment, usati per specificare il contesto del singolo esperimento e mandati in input ai vari “propagation running codes”.

- .prn – files di log dell'output dei programmi.
- .shd – files di tipo “shade”, utilizzati per il calcolo dell'attenuazione di propagazione.
- .brc – files contenenti i coefficienti da riflessione da fondale. Utilizzati ad Bellhop.
- .ray – files contenenti i coefficienti che rappresentano il cammino dei raggi.
- .arr – rappresentano le ampiezze/ritardi per il singolo arrivo. Utilizzati da Bellhop.
- .grn – funzioni di Green specifiche dipendenti dalla profondità.
- .mod – files dei modi. Rappresentano l'output di kraken o krakenc

I .mod files sono in realtà delle copie del vero file di output di kraken, chiamato MODFIL, poi successivamente rinominato in .mod.

Come detto, per facilitare l'import in matlab tutte le variabili elaborate dal programma (con specifico riferimento a kraken) è stata sviluppata la routine “kraken_import.m”. Essendo uno script non ha bisogno di una chiamata diretta ma basta semplicemente lanciarlo. Inoltre il menu testuale permette di scegliere che tipo di file importare e la sua posizione. Le variabili che vengono memorizzate nel workspace sono:

- pltitf – Titolo dell'esperimento contenuto in formato testo del file .mod.
- freq – Frequenza di lavoro nell'esperimento [Hz].
- ck – Coefficienti dei modi.
- z – profondità relativa dei modi [m].
- phi – Fase.

- cpt – velocità delle p-wave a profondità relativa 0 (livello alto dell'esperimento) [m/s].
- cst – velocità delle s-wave a profondità relativa 0 (livello alto dell'esperimento) [m/s].
- rhot – densità a profondità relativa 0 [g/cm³].
- deptht – profondità del punto più alto dell'esperimento.
- cpb – velocità delle p-wave a profondità relativa massima (fondale) [m/s].
- csb – velocità delle s-wave a profondità relativa massima (fondale) [m/s].
- rhob – densità a profondità relativa massima [g/cm³].
- depthb – profondità al punto più basso dell'esperimento.
- NMedia – Numero dei modi.
- Depth – profondità dell'esperimento.
- Rho – densità.

In appendice sono riportati sia il file "*kraken.hlp*" (sono evidenziati i punti di maggior interesse nella scrittura dello script) presente nel framework, sia il file "*kraken_import.m*".

..**kraken.hlp**..

KRAKEN is the main program. It takes an environmental file, computes the modes, and writes them to disk for use by other modules. A print file is also produced, echoing the user input.

KRAKENC is a complex arithmetic version (hence the C in KRAKENC) of KRAKEN. By working in the complex domain, loss mechanisms such as ice scatter and material absorption may be included 'exactly' rather than perturbatively. In addition, leaky modes may be computed. The price of this non-perturbative treatment is a slowdown in speed by approximately a factor of 4. This factor principally represents the difference between complex and real arithmetic.

A further slow down by a factor of 2 or more may occur if the Twersky scatter option is used in KRAKENC. The calculation of the Twersky scatter function can require significant CPU time; enough to actually be a dominant part of the cost of computing the modes. KRAKEN incorporates the scatter perturbatively and is much less sensitive to the cost of Twersky scatter.

KRAKEN does not allow for losses in elastic media due to material attenuation. Thus, for attenuating elastic media, KRAKENC should be used.

Files:

	Name	Unit	Description
Input	*.ENV	1	ENVironmental data
	*.BRC	10	Bottom Refl. Coef. (opt1)
	*.TRC	11	Top Refl. Coef. (opt1)
	*.IRC	12	Internal Refl. Coef. (opt1)
Output	*.PRT	6	PRinT file
	*.MOD	20	MODE file

 EXAMPLE AND DESCRIPTION OF ENV FILE:

```
'FRAMIV Twersky S/S ice scatter'      ! TITLE
50.0                                   ! FREQ (Hz)
4                                       ! NMEDIA
'NSF'                                  ! OPTIONS
0.0092 8.2 5.1                         ! BUMDEN (1/m) ETA (m) XI (m)
750 0.0 3750.0                         ! NMESH SIGMA (m) Z(NSSP)
    0.0 1436.0 0.0 1.03/                ! Z(m) CP CS(m/s) RHO(gm/cm3)
    30.0 1437.4 /
    50.0 1437.7 /
    80.0 1439.5 /
    100.0 1441.9 /
    125.0 1444.6 /
    150.0 1450.0 /
    175.0 1456.1 /
    200.0 1458.4 /
    250.0 1460.0 /
    300.0 1460.5 /
    350.0 1460.6 /
```

```

400.0 1461.0 /
450.0 1461.5 /
500.0 1462.0 /
600.0 1462.9 /
700.0 1463.9 /
800.0 1464.8 /
900.0 1465.8 /
1000.0 1466.7 /
1100.0 1467.0 /
1200.0 1469.0 /
1300.0 1469.5 /
1400.0 1471.8 /
1600.0 1474.5 /
1800.0 1477.0 /
2000.0 1479.6 /
2500.0 1487.9 /
3750.0 1510.4 /
35 0.0 3808.33
3750.0 1504.6 0.0 1.50 .15 0.0
3808.33 1603.07 /
35 0.0 3866.66
3808.33 1603.07 0.0 1.533 .15 0.0
3866.66 1701.53 /
35 0.0 3925.0
3866.66 1701.53 0.0 1.566 .15 0.0
3925.0 1800.0 /
'A' 0.0 ! BOTOPT SIGMA (m)
3925.0 1800.0 0.0 1.60 .15 0.0
0.0 1504.0 ! CLOW CHIGH (m/s)
300.0 ! RMAX (km)
1 100.0 / ! NSD SD(1:NSD) (m)
1 200.0 / ! NRD RD(1:NRD) (m)

```

DESCRIPTION OF INPUTS:

(1) - TITLE

Syntax:

TITLE

Description:

TITLE: Title of run enclosed in single quotes.

(2) - FREQUENCY

Syntax:

FREQ

Description:

FREQ: Frequency in Hz.

(3) - NUMBER OF MEDIA

Syntax:

NMEDIA (<20)

Description:

NMEDIA: Number of media.

The problem is divided into media within which it is

assumed that the material properties vary smoothly. A new medium should be used at fluid/elastic interfaces or at interfaces where the density changes discontinuously. The number of media in the problem is defined excluding the upper and lower half-space.

(4) - OPTIONS

Syntax:

OPTION

Description:

OPT(1:1): Type of interpolation to be used for the SSP.
 'C' for C-linear,
 'N' for N2-linear (n the index of refraction),
 'S' for cubic Spline,
 'A' for Analytic. The user must modify the analytic formulas in PROFIL.FOR then compile and link.

If your not sure which option to take, I'd suggest you use 'C' or 'N'. Practically, you can pick either one: the choice has been implemented to facilitate precise intermodel comparisons.

Option 'S' is a little dangerous because splines yield a poor fit to certain kinds of curves, e.g. curves with sharp bends. If you insist on splines, you can fix a bad fit by dividing the water column into two 'media' at the bend.

Run PLOTSSP to check that the SSP looks the way you thought it should. Apart from potential typos, this will also show up fit-problems.

OPT(2:2): Type of top boundary condition.
 'V' VACUUM above top.
 'A' ACOUSTO-ELASTIC half-space.
 Requires another line as described in block (4a).
 'R' Perfectly RIGID.
 'F' Reflection coefficient from a FILE (available in KRAKENC only). Requires additional lines as described in block (4c).
 'S' for Soft-boss Twersky scatter.
 'H' for Hard-boss Twersky scatter.
 'T' for Soft-boss Twersky scatter, amplitude only.
 'I' for Hard-boss Twersky scatter, amplitude only. The Twersky scatter options require another line as described in block (4c). Mnemonically, T, I options are one letter after S, H in the alphabet. Current wisdom is that option T is most appropriate for ice scatter.

For open ocean problems option 'V' should be used for the top BC. The Twersky options are intended for under-ice modeling.

OPT(3:3): Attenuation units.
 'N' Nepers/m.

'F' dB/(kmHz) (F as in Freq. dependent)
 'M' dB/m (M as in per Meter)
 'W' dB/wavelength (W as in per Wavelength)
 'Q' quality factor.
 'T' Thorp attenuation formula. This overrides any other attenuations specified.

KRAKEN ignores material attenuation in elastic media. (KRAKENC treats it properly).

OPT(4:4): Slow/robust root-finder.
 '.' As in: I want all the modes and I don't care how long it takes. Period. (Available in KRAKENC only.)
 In certain problems with elastic layers the old root-finder has been known to skip modes.

(4a) - TOP HALFSPACE PROPERTIES

Syntax:

ZT CPT CST RHOT APT AST

Description:

ZT: Depth (m).
 CPT: Top P-wave speed (m/s).
 CST: Top S-wave speed (m/s).
 RHOT: Top density (g/cm3).
 APT: Top P-wave attenuation. (units as given in Block 2)
 AST: Top S-wave attenuation. (" " " " " ")

This line should only be included if OPT(2:2)='A', i.e. if the user has specified a homogeneous halfspace for the top BC.

(4b) - TOP REFLECTION COEFFICIENT

Syntax:

NTHETA
 THETA(1) RMAG(1) RPHASE(1)
 THETA(2) RMAG(2) RPHASE(2)
 .
 .
 .
 THETA(NTHETA) RMAG(NTHETA) RPHASE(NTHETA)

Description:

NTHETA: Number of angles.
 THETA(): Angle.
 RMAG(): Magnitude of reflection coefficient.
 RPHASE(): Phase of reflection coefficient (degrees).

Example:

```
3
0.0 1.00 180.0
45.0 0.95 175.0
90.0 0.90 170.0
```

These lines should be contained in a separate '.TRC' file. This file is only required if OPT(2:2)='F', i.e. if the user has specified that the top BC is read from a '.TRC'

(Top Reflection Coefficient) file.

This option for tabulated reflection coefficients is somewhat experimental at this time. I haven't worried about the multivalued character of the phase function: choose your reference and make sure the phase varies continuously. A complicated reflection coefficient may well cause problems for the mode-finder.

(4c) - TWERSKY SCATTER PARAMETERS

Syntax:

BUMDEN ETA XI

Description:

BUMDEN: Bump density (ridges/km).
 ETA: Principal radius 1 (m).
 XI: Principal radius 2 (m).

This line should only be included when one of the Twersky-scatter options is selected.

(5) - MEDIUM INFO

Syntax:

NMESH SIGMA Z(NSSP)

Description:

NMESH: Number of mesh points to use initially. The number of mesh points should be about 10 per vertical wavelength in acoustic media. In elastic media, the number needed can vary quite a bit; 20 per wavelength is a reasonable starting point.

The maximum allowable number of mesh points is given by 'MAXN' in the dimension statements. At present 'MAXN' is 50000. The number of mesh points used depends on the initial mesh and the number of times it is refined (doubled). The number of mesh doublings can vary from 1 to 5 depending on the parameter RMAX described below.

If you type 0 for the number of mesh points, the code will calculate NMESH automatically.

SIGMA: RMS roughness at the interface.

Z(NSSP): Depth at bottom of medium (m). This value is used to detect the last SSP point when reading in the profile that follows.

(5a) - SOUND SPEED PROFILE

Syntax:

Z(1) CP(1) CS(1) RHO(1) AP(1) AS(1)
 Z(2) CP(2) CS(2) RHO(2) AP(2) AS(2)
 .
 .
 .
 Z(NSSP) CP(NSSP) CS(NSSP) RHO(NSSP) AP(NSSP) AS(NSSP)

Description:

Z(): Depth (m).

The surface starts at the first depth point specified. Thus if you have say, XBT data which starts at 50 m below the surface, then you'll need to put in some SSP point at 0 m, otherwise the free-surface would be placed at 50 m giving erroneous results. The points Z(1) and Z(NSSP) MUST correspond to the depths of interfaces between media.

```

CP():      P-wave speed (m/s).
CS():      S-wave speed (m/s).
RHO():     Density (g/cm3).
           Density variations within an acoustic medium
           are at present ignored.
AP():      P-wave attenuation (units as given in Block 2)
AS():      S-wave attenuation ( " " " " " " ")
    
```

These lines should be omitted when the 'A' option is used (indicating that an analytic profile is supplied by a user written subroutine).

The '/' character signals that the remaining data on the line is the same as in the previous line of SSP data. For the very first line the default or 'previous' line is:

```
0.0 1500.0 0.0 1.0 0.0 0.0
```

This block should be repeated for each subsequent medium.

(6) - BOTTOM BOUNDARY CONDITION

Syntax:

```
BOTOPT SIGMA
```

Description:

```

BOTOPT: Type of bottom boundary condition.
        'V' VACUUM below bottom.
        'A' ACOUSTO-ELASTIC half-space.
           Requires another line with the half-space
           parameters. The format is the same as that
           used for specifying the top halfspace BC.
        'R' Perfectly RIGID.
        'F' reflection coefficient from a FILE (available
           in KRAKENC only). Requires a Bottom
           Reflection Coefficient file with
           extension '.BRC'. The format is the same as
           that used for a Top Reflection coefficient.
        'P' Precalculated internal reflection coefficient
           from a FILE (available in KRAKENC only).
           These files are generated using BOUNCE.
        Option 'A' is generally used for ocean bottom
        modeling.
SIGMA:  Interfacial roughness (m).
    
```

(7) - PHASE SPEED LIMITS

Syntax:

```
CLOW CHIGH
```

Description:

```

CLOW:   Lower phase speed limit (m/s).
        CLOW will be computed automatically if you set
    
```

it to zero. However, by using a nonzero CLOW you can skip the computation of slower modes. Mainly this is used to exclude interfacial modes (e.g. a Scholte wave). The root finder is especially slow in converging to these interfacial modes and when the source and receiver are sufficiently far from the interface the interfacial modes are negligible.

CHIGH: Upper phase speed limit (m/s). The larger CHIGH is, the more modes are calculated and the longer the execution time. Therefore CHIGH should be set as small as possible to minimize execution time.

On the other hand, CHIGH controls the maximum ray angle included in a subsequent field calculation-- ray paths are included which turn at the depth corresponding to CHIGH in the SSP. Thus a larger CHIGH means more deeply penetrating rays are included.

Choice of CHIGH then becomes a matter of experience. In the far-field and at high-frequencies, rays travelling in the ocean bottom are severely attenuated and one may set CHIGH to the sound speed at the ocean bottom. In the near-field, low-frequency case, rays refracted in the bottom may contribute significantly to the field and CHIGH should be chosen to include such ray paths.

KRAKEN will (if necessary) reduce CHIGH so that only trapped (non-leaky) modes are computed.

KRAKENC will attempt to compute leaky modes if CHIGH exceeds the phase velocity of either the S-wave or P-wave speed in the half-space. Leaky mode computations are somewhat experimental at this time.

(8) - MAXIMUM RANGE

Syntax:

RMAX

Description:

RMAX: Maximum range (km).

This parameter should be set to the largest range for which a field calculation will be desired.

During the mode calculation the mesh is doubled successively until the eigenvalues are sufficiently accurate at this range. If you set it to zero, then no mesh doublings will be performed. You don't need to worry too much about this parameter-- even if you set it to zero the results will usually be reasonable.

(9) - SOURCE/RECEIVER DEPTH INFO

Syntax:

```
NSD SD(1:NSD)
NRD RD(1:NRD)
```

Description:

```
NSD: The number of source depths.
SD(): The source depths (m).
NRD: The number of receiver depths.
RD(): The receiver depths (m).
```

This data is read in using list-directed I/O so you can type it just about any way you want, e.g. on one line or split onto several lines. Also if your depths are equally spaced then you can type just the first and last depths followed by a '/' and the intermediate depths will be generated automatically.

CPU time is essentially independent of the number of sources and receivers so that you can freely ask for up to 4095 depths. However, for high-frequencies the storage for the mode files can be excessive.

The source/rcvr depths are sorted and merged and then the modes are calculated at the union of the two sets of depths. Thus, it doesn't matter if you mix up source and receiver depths. Furthermore, you can leave out either the source or receiver specification (but not both simultaneously) simply by using a '/' for that line.

Sources and receivers cannot be placed in a half-space.

If you are going to be doing a coupled-mode calculation then you must specify a large number of receiver depths spanning the entire column (down to the half-space). Fine sampling (about 10 points/wavelength) is needed to calculate the coupling integrals accurately.

SAMPLE PRINT OUT

The print-out for this deck is shown below

```
KRAKEN- FRAMIV Twersky S/S ice scatter
Frequency = 20.00 NMEDIA = 4
```

```
N2-LINEAR approximation to SSP
Attenuation units: dB/mkHz
TWERSKY SOFT BOSS surface scatter model
```

```
Twersky ice model parameters:
Bumden = 0.920000E-02 Eta = 8.20 Xi = 5.10
```

Z	ALPHAR	BETAR	RHO	ALPHAI	BETAI
(Number of pts = 750 RMS roughness = 0.000E+00)					
0.00	1436.00	0.00	1.03	0.0000	0.0000
30.00	1437.40	0.00	1.03	0.0000	0.0000
50.00	1437.70	0.00	1.03	0.0000	0.0000

80.00	1439.50	0.00	1.03	0.0000	0.0000
100.00	1441.90	0.00	1.03	0.0000	0.0000
125.00	1444.60	0.00	1.03	0.0000	0.0000
150.00	1450.00	0.00	1.03	0.0000	0.0000
175.00	1456.10	0.00	1.03	0.0000	0.0000
200.00	1458.40	0.00	1.03	0.0000	0.0000
250.00	1460.00	0.00	1.03	0.0000	0.0000
300.00	1460.50	0.00	1.03	0.0000	0.0000
350.00	1460.60	0.00	1.03	0.0000	0.0000
400.00	1461.00	0.00	1.03	0.0000	0.0000
450.00	1461.50	0.00	1.03	0.0000	0.0000
500.00	1462.00	0.00	1.03	0.0000	0.0000
600.00	1462.90	0.00	1.03	0.0000	0.0000
700.00	1463.90	0.00	1.03	0.0000	0.0000
800.00	1464.80	0.00	1.03	0.0000	0.0000
900.00	1465.80	0.00	1.03	0.0000	0.0000
1000.00	1466.70	0.00	1.03	0.0000	0.0000
1100.00	1467.00	0.00	1.03	0.0000	0.0000
1200.00	1469.00	0.00	1.03	0.0000	0.0000
1300.00	1469.50	0.00	1.03	0.0000	0.0000
1400.00	1471.80	0.00	1.03	0.0000	0.0000
1600.00	1474.50	0.00	1.03	0.0000	0.0000
1800.00	1477.00	0.00	1.03	0.0000	0.0000
2000.00	1479.60	0.00	1.03	0.0000	0.0000
2500.00	1487.90	0.00	1.03	0.0000	0.0000
3750.00	1510.40	0.00	1.03	0.0000	0.0000

(Number of pts = 35 RMS roughness = 0.000E+00)
 3750.00 1504.60 0.00 1.50 0.1500 0.0000
 3808.33 1603.07 0.00 1.50 0.1500 0.0000

(Number of pts = 35 RMS roughness = 0.000E+00)
 3808.33 1603.07 0.00 1.53 0.1500 0.0000
 3866.66 1701.53 0.00 1.53 0.1500 0.0000

(Number of pts = 35 RMS roughness = 0.000E+00)
 3866.66 1701.53 0.00 1.57 0.1500 0.0000
 3925.00 1800.00 0.00 1.57 0.1500 0.0000

ACOUSTO-ELASTIC half-space, (RMS roughness = 0.000E+00)
 3925.00 1800.00 0.00 1.60 0.1500 0.0000

CLOW = 0.00000E+00 CHIGH = 1504.0
 RMAX = 300.00000000000000

Number of sources = 1
 100.0000

Number of receivers = 1
 200.0000

Mesh multiplier	CPU seconds
1	16.4
2	15.1

I	K	ALPHA	PHASE SPEED
1	0.8625082052E-01	-0.8519020992E-06	1456.956646
2	0.8582849772E-01	-0.1302695655E-06	1464.125663
3	0.8562855085E-01	-0.1059327457E-06	1467.544468
4	0.8545402623E-01	-0.1136748056E-06	1470.541667
5	0.8527187871E-01	-0.1192384459E-06	1473.682861
6	0.8510445198E-01	-0.1156165482E-06	1476.582050
7	0.8495255965E-01	-0.1130917467E-06	1479.222129

8	0.8479984039E-01	-0.1185453302E-06	1481.886116
9	0.8465149335E-01	-0.1314814525E-06	1484.483039
10	0.8450452348E-01	-0.1255743704E-06	1487.064845
11	0.8435857532E-01	-0.1276318031E-06	1489.637606
12	0.8421637950E-01	-0.1377681231E-06	1492.152796
13	0.8407780307E-01	-0.1377169389E-06	1494.612151
14	0.8393959060E-01	-0.1339925824E-06	1497.073136
15	0.8380370528E-01	-0.1378254389E-06	1499.500598
16	0.8367091002E-01	-0.1450063419E-06	1501.880476

If the program aborts in some way, examine the print file which is produced. Frequently an expected line has been omitted and the environmental file is therefore misinterpreted.

The message "FAILURE TO CONVERGE IN SECANT" occurs when KRAKEN requires more than 500 iterations to converge to a mode. Usually less than 20 iterations are needed but convergence to interfacial modes (Scholte or Stoneley waves) can be exceptionally slow, especially at higher frequencies. The simplest solution is to exclude interfacial modes by setting the lower phase-speed limit to the minimum p-wave speed in the problem. Alternately, you can increase the value of MAXNIT which controls the MAXimum Number of ITERations in the root finder.

[kraken_import.m]

BIBLIOGRAFIA

- [1] E. Di Claudio, R. Parisi. "*Robust Wideband Beamforming*", INFOCOM Dpt., University of Rome "La Sapienza", 2003 (?)
- [2] E. Di Claudio. "Paradigmi Neurali e Modelli di Campo Riverberatore per l'Elaborazione Adattiva del Segnale mediante Schiere di Sensori". INFOCOM Dpt., University of Rome "La Sapienza", 2003 (?)
- [3] Jian Li, Petre Stoica, Zhisong Wang. "*On Robust Capon Beamforming and Diagonal Loading*". IEEE Translation on Signal Processing, Vol. 51, No. 7, July 2003.
- [4] Micheal B. Porter. "*The KRAKEN Normal Mode Program (DRAFT)*", SACLAND Undersea Research Center, July 6, 2001.