**DSAP PROJECT**

**Positions of the sensors (Maximum delay information)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **7** | **-6129** | **9784** | **-4750** |  |
| **6** | **-6183** | **-4874** | **-4650** |  |
| **5** | **-6183** | **-12402** | **-4600** |  |
| **4** | **6865** | **12844** | **-4750** |  |
| **3** | **6520** | **5240** | **-4650** | **9.542 s** |
| **2** | **6635** | **-2132** | **-4500** | **4.7893 s** |
| **1** | **6566** | **-9617** | **-4500** | **0** |

**True delay looking at Audacity**

**Sensor Time event 1 Delay Time event 2 Delay 2**

1 2:35:131 0 8:52:275 0

2 2:31:544 -3,587 8:47:393 -4,882

3 2:29:867 -5,287 8:46:402 -5,873

4 2:31:150 -3,981 8:47:689 -4,586

5 2:41:920 6,789 8:58:482 6,207

6 2:39:040 3,909 8:55:710 3,435

7 2:37:829 2,698 8:54:295 2,020

**Primeros problemas**

- En el espectrograma de la señal de las ballenas minke se ven bandas espectrales de ruido muy grande y narrowband. Ademas, sabiendo que la señal es bastante débil, da una SNR muy baja. Tambien hay una señal superpotente y narrowband interferentea 20khz

- La baja frecuencia de la señal ruido enmascara la señal de la ballena, que es de frecuencias muy bajas teoricamente (100-400 Hz) en el audacity pone que llega a 2300 Hz***!!! ????***

- Se observan señales muy breves pero a alta frecuencia. Posiblemente proviniente de delfines

- Primeras simulaciones de dos sinus retrasado 8 muestras de 4 Khz a 250 Khz de freq sample y dos chirp iguales. Despues calculamos el time delay, donde da buenos resultados en XCORR y en GCC con la funcion constante. EN el caso de PHAT, da 0 o 1 samples de delay. ***Porque????*** Luego vemos que si que empezara a funcionar con real data. **FOTOS**

- Simulaciones de real data entre los 3 primeros sensores: Las primeras simulaciones, cogiendo rango de tiempos entre 0-3 minutos, 2:15-2:45 min y 7-9 min vemos que la xcorr y la gcc (cte) no funcionan correctamente, ya que nos da un delay de 2000 muestras aproximadamente. (Siendo la Fs=96000 y 3,5 s el delay real). Tambien vemos que dependiendo del rango de tiempo escogido, tenemos diferentes resultados. Incluso en simulaciones entre 5-7 minutos vemos como los resultados varian incluso usando los dos algoritmos teoricamente identicos (gcc (cte) y xcorr). En el caso de gcc phat vemos que da 0 samples de delay en todas las simulaciones.

Conclucion: El ruido fuertemente correlado, la señal no esta blanqueada y la presencia de inteferencias hacen que los algoritmos fracasen.

fallan estrepitosamente. En cambio, la Gcc phat da lo mismo que antes de procesar la P3.

-Scot hace algo. La correlacion es fea pero aun no hemos renovado la señal!!

- El proximo paso será reducir el ruido y decorrelarlo. Ademas, implementar otros algoritmos mas potentes que el gcc y el cc.

**Resultados simulaciones Real data**

**Sin procesar**

Entre sensores 1-2 (delay esperado: 1r evento 3,5s 2nd evento: 4,882s)

Rango de tiempo Xcorr GCCcte GCC phat Scot

0-3 -2409 -2408 0

0-4 -2409 -2409 0

7-9 -806 -4012 0

8:40-9 797 5587 0 -4012 WHALE

Entre sensores 1-3

Rango de tiempo Xcorr GCCcte GCC phat Scot

0-3 2378 2379 0

2:25-2:40 775 2378 0

5:30-5:35(sin evento) 775 2378 0

7-9 775 2378 0

8:50-9 -828 3981 0 -8819 WHALE

**Procesando con TImeGainNormalization**

Entre sensores 1-2

Rango de tiempo Xcorr GCCcte GCC phat Scot

2:20-2:45 372866 372866 368600 362029

**-3,587 3,884 3,8395 3,7711**

8:40-9 854685 8544685 856019 778456

**-4,882 8,902 8,916 8,108**

Entre sensores 1-3

Rango de tiempo Xcorr GCCcte GCC phat Scot

2:20-2:45 539331 539331 539331 503412

**-5,287 5,618 5,241**

8:40-9 1033350 1033350 1017704 1034272

**-5,873 10.76 10.601 10.77**

Entre sensores 1-4

Rango de tiempo Xcorr GCCcte GCC phat Scot

2:20-2:45 384215 384215 384215 384220

**- 3.981 4.003 4.002**

8:40-9 888563 888563 886879 820723

**-4.586 9.255 9.238 8.549**

**Procesando con FILTRO y TImeGainNormalization**

Entre sensores 1-2

Rango de tiempo Xcorr GCCcte GCC phat Scot

2:20-2:45 371957 371957 6 371957

**-3,587 3.8745**

8:40-9 383698 383698 7 383002

**-4,882 4 3.989**

**Procesando con FILTRO y TImeGainNormalization**

Entre sensores 1-3

Rango de tiempo Xcorr GCCcte GCC phat Scot

2:20-2:45 0 535141 -1 532634

**-5.287 5.574 5.548**

8:40-9 479999 479999 479999 1043223

**-5.873 4.999 10.866**

**EFECTO DE PRE PROCESING CON FILTER**

En el primer evento, donde ya obteniamos buenos resultados con el TIME GAIN NORMALIZATION, han sucedido varias cosas. Primero aplicamos el filtro de Xavi y despues el Time Gain.

- Con Xcorr, Gcc cte, el filtro consigue mejorar en milesimas o centesimas de segundo el time delay. Por lo que no vale mucho la pena!

- En el caso del phat, aplicando el filtro deja de funcionar el algoritmo.

- En el caso del Scot, el filtro lo que hace es empeorar la estimación del delay!

En el segundo evento, donde obteniamos malos resultados con el Time Gain Normalization, han sucedido varias cosas. Primero aplicamos el filtro de Xavi y luego el Time Gain.

- Con Xcorr, Gcc cte, el filtro consigue mejorar en segundos el time delay. Lo que pasa que aún así, el time delay de queda entre 5-8 decimas de segundo del time delay real

- En el caso del phat, aplicando el filtro deja de funcionar el algoritmo entre los sensores 1-2 y mejora el caso de solo TimeGain entre los sensores 1-3

- En el caso del Scot, el filtro entre los sensores 1-2 mejora y entre el sensor 1-3 empeora la estimacion del delay respecto el caso de solo Time Gain.

**Pruebas de reducción de ruido:**

- TK

-Spectral substraction

- Algorithms of power point, by Ludwig

**Otros algoritmos a implementar** sabiendo que PHAT está hecho mas para habitaciones

- Scot (gcc)

- Eigenvalue Descomposition

- LMS