

PROCESAMIENTO DE DATOS DE CTD ONDULANTE

A. Trasviña

Oceanografía Física, CICESE en B.C.S.

Miraflores 334 e/Muleg y La Paz, La Paz, B.C.S., 23050, México

E-mail: trasvi@cicese.mx

RESUMEN

Se presenta un método de post-procesamiento de datos de CTD ondulante y se propone mejorar el diseño de un vehículo ondulante en particular. Este método permite medir variables dinámicas de meso y submesoescala en el océano superior. El cruce de la línea Giro 1 (experimento Tehuano 2, febrero-marzo de 1996) ilustra las ventajas de usar CTD ondulante para estudiar la dinámica de mesoescala.

INTRODUCCIÓN

El uso de vehículos ondulantes (o perfiladores rápidos) para las sondas CTD (conductividad, temperatura, presión) se ha vuelto muy popular en años recientes. El sueño de realizar finas mallas sinópticas de perfiles verticales todavía está lejos de realizarse, sin embargo los CTD ondulantes (CTDO) permiten realizar recorridos a velocidades sustancialmente mayores que los métodos tradicionales. Esto, a su vez, permite visualizar escalas espacio-temporales más pequeñas que las que se observan con lances puntuales. El uso conjunto de CTDO y perfiladores acústicos de corrientes ya ha tenido un impacto importante en el estudio de la dinámica de mesoescala y submesoescala en el océano (ver por ejemplo Pollard, 1986; Pollard & Regier, 1990; Trasviña *et al.* 1995).

Se han desarrollado una variedad de modelos, siendo Seasoar (Chelsea Instruments Ltd.) uno de los más populares. Sin embargo, hay dos problemas con este tipo de observaciones. El primero es que los vehículos ondulantes son generalmente muy caros y difíciles de operar. En segundo lugar los CTDs que se usan para perfilar no están diseñados para medir a las velocidades relativamente altas (y variables) que caracterizan a los vehículos ondulantes. De aquí surgen dos razones para el presente manuscrito: proporcionar una metodología básica para el procesamiento de los datos y promover el uso de un vehículo ondulante de bajo costo y de fácil manejo.

Recientemente, Filonov, Monzón y Tereshchenko (1996, FMT96 de aquí en adelante) publicaron el diseño de un vehículo barato, basado en ideas muy simples y que no requiere de malacates especiales para realizar los arrastres. Este diseño es particularmente útil en el momento actual en que los CTDs son ya instrumentos confiables, en que su tamaño y peso se han reducido considerablemente y, más importante aún, en que cuentan con memorias internas para la grabación de gran cantidad de datos. El vehículo descrito en FMT96 aloja un CTD autónomo SBE19 (Seabird Electronics) y cuenta con aletas de sustentación para mantenerlo cerca de la superficie mientras es arrastrado por el buque. Para eliminar la fuerza de

sustentación, el buque realiza uno o más virajes de 360° y, con esto, el vehículo se hunde hasta la longitud del cable. Al realizar cada viraje el buque mantiene su posición, es decir, no deriva con las corrientes como sucede durante las hidrocalas tradicionales. A raíz de la publicación de este diseño, el autor ha experimentado con éxito adaptando un CTD SBE25 para convertirlo a ondulante. Este tiene la ventaja de contar con una razón de muestreo de 8 Hz, comparado con 2 Hz en el SBE19. En ambos casos se usan los sensores libres, es decir, sin el sistema de bombeo típico de estos instrumentos.

La simplicidad de este diseño tiene su precio. Por ejemplo, no se tiene comunicación con el instrumento mientras se realiza un arrastre. La otra desventaja es común a todos los tipos de perfiladores rápidos y consiste en que las mediciones de salinidad requieren de un laborioso post-procesamiento para obtener perfiles limpios de errores. Estos errores se producen por la diferencia entre el tiempo de respuesta de los sensores de temperatura y conductividad, y se visualizan como 'picos' en el registro 'crudo' (no post-procesado) de salinidad. En este manuscrito se describe un método interactivo para el post-procesamiento de datos de CTDO. Finalmente, este trabajo pretende proponer parámetros para mejorar el método de perfilado rápido descrito en FMT96.

INSTRUMENTACIÓN Y METODOLOGÍA

Los datos que se analizan aquí fueron obtenidos con el vehículo descrito en FMT96. Se utilizó un CTD SBE19 con sensores de flujo libre de temperatura, conductividad y presión. Este instrumento muestrea la columna de agua con una frecuencia de 2 Hz. La velocidad óptima de perfilado es de 1 ms⁻¹.

Los datos se obtuvieron de la forma siguiente:

- a) El CTD se inicializa en cubierta. La captura de datos empieza antes de que el instrumento entre al agua.
- b) Se larga el CTDO a buque parado, en este caso hasta la marca de 300 m de cable hidrográfico.

- c) Se inicia el arrastre y el vehículo sube a la superficie por primera vez. La velocidad promedio de arrastre fue de 8 nudos.
- d) Para realizar los perfiles el buque hace virajes de 360°. En este caso los perfiles se realizaron cada 5 ó 10 minutos. La profundidad que alcanza el vehículo depende del lastre del instrumento y de la velocidad y dirección de las corrientes.
- e) Para finalizar un arrastre y leer la información del CTD el buque se detiene y se sube el vehículo a cubierta. En este instrumento en particular el tamaño de la memoria nos permitió realizar arrastres continuos por 3 horas.

El vehículo que utilizamos para obtener este grupo de datos era todavía un prototipo y solamente perfiló entre los 30 ó 50 m y la longitud del cable. Es decir, nunca alcanzó la superficie durante un arrastre. Sin embargo, en base a esta experiencia se alteró el diseño del vehículo y se construyeron dos versiones mejoradas que alcanzan la superficie sin problemas. En la Figura 1a se muestra el vehículo ondulante antes de entrar al agua. La Figura 1b muestra un arrastre típico con este prototipo.

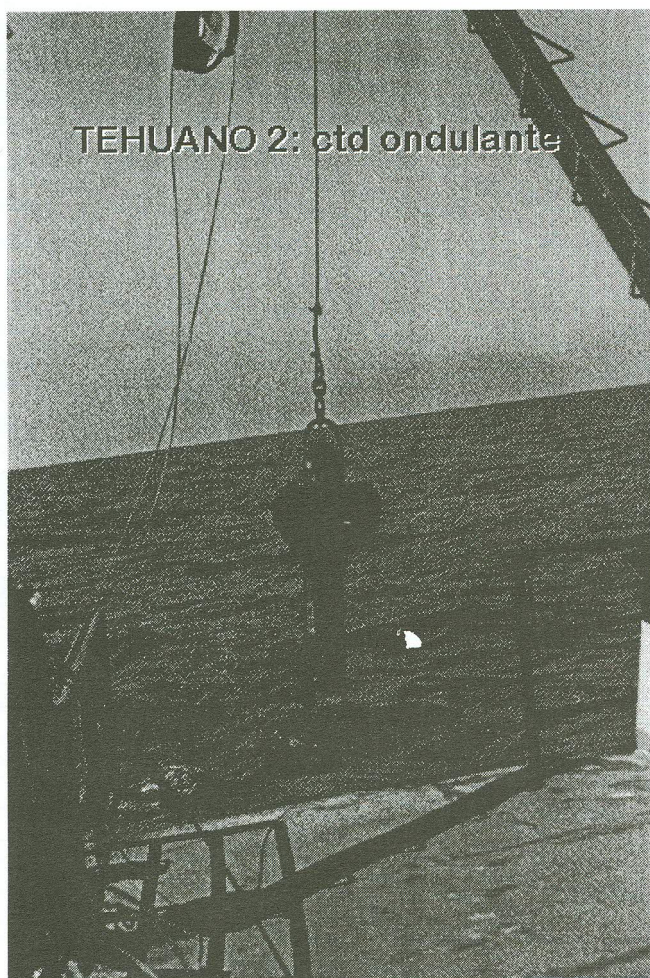


Figura 1a. Vehículo ondulante al momento de entrar al agua.

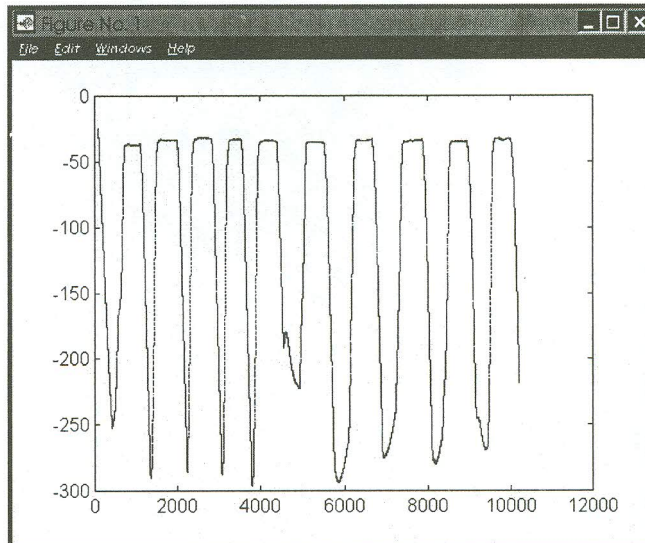


Figura 1b. Gráfica de tiempo (s) contra profundidad (db) de un arrastre típico de CTDO. Esta es la primera gráfica del programa *separa.m*.

UN MÉTODO DE POST-PROCESAMIENTO

El método que se describe a continuación se diseñó para ser utilizado con cualquier tipo de instrumento. Únicamente el primer paso requiere del uso de programas propietarios para la conversión de datos.

La compañía Seabird distribuye programas de procesamiento de datos para el SBE19. Estos producen perfiles de temperatura y salinidad de calidad a partir de hidrocalas puntuales, pero no pueden usarse para procesar arrastres de CTDO. Ello se debe a que el esquema tradicional de trabajo de un SBE19 incluye el perfilar a una velocidad constante cercana a 1 ms^{-1} . En un arrastre de CTDO las velocidades varían entre 0 y 3 ms^{-1} . La velocidad de perfilar es un factor crítico en el cálculo de los desfases de temperatura y conductividad. Los programas de Seabird no permiten variar este desfase dentro de cada lance y esto resulta en perfiles de salinidad demasiado ruidosos para ser utilizados, por ejemplo, en cálculos dinámicos. Además, la serie de presión de un lance normal de CTD incluye solamente una subida y una bajada, mientras que un arrastre de CTDO incluye varios ciclos de subidas y bajadas. Los programas de Seabird no están hechos para procesar ciclos de perfilar. El modelo de post-procesamiento que se describe a continuación resuelve estos problemas.

El modelo de post-procesamiento se divide en varias fases. La primera es la conversión de los datos crudos en binario o hexadecimal, a *ascii*. El resto del proceso se desarrolla en MATLAB y se divide en cuatro fases más. Cada fase corresponde a una función de MATLAB distinta que es llamada desde un programa principal (*prondul.m*). Las cinco fases del procesamiento son:

1. Conversión a ascii: DATCNV (Oceansoft de Seabird)
2. Separación de lances: *separa.m* (MATLAB)
3. Obtención de secciones de lance de velocidad constante: *proint1.m* (MATLAB).
4. Defasamiento de temperatura con respecto a presión y cálculo de salinidad: *proint2.m* (MATLAB).
5. Edición interactiva de salinidades, suavizado e integración de datos de navegación: *proint3.m* (MATLAB).

Se detallan a continuación.

PROCESAMIENTO EN DATCNV (OCEANSOFT, DE SEABIRD ELECTRONICS): OBTENCIÓN DE ARCHIVOS ASCCI

→ **Entrada:** archivo de datos crudos *.HEX (extensión de Seabird).

← **Salida:** archivo de datos crudos convertido a ascii *.CNV (extensión de Seabird).

Al recuperar los datos de un arrastre de CTDO la información se almacena en un archivo que contiene varios ciclos de subidas y bajadas. Cada archivo es traducido a ascii usando el programa DATCNV de la compañía Seabird. DATCNV se aplica a los archivos crudos con extensión HEX y **debe configurarse para que no elimine las inversiones de profundidad**. La salida del programa debe ser como:

- Tiempo en segundos: esto es importante, DATCNV presenta varias opciones (como días julianos), pero éstas dan problemas de precisión cuando se requiere obtener diferencias de tiempo en pasos subsiguientes.
- Conductividad en mS/cm
- Temperatura en T68
- Presión en db

PROCESAMIENTO EN MATLAB, PRIMERA FASE: SEPARACIÓN DE UN ARRASTRE EN LANCES INDIVIDUALES CON LA FUNCIÓN SEPARA.M

→ **Entrada:** salida de DATCNV en forma de un archivo ascii con extensión CNV. Para facilitar la lectura de los datos debe eliminarse el encabezado. Personalmente prefiero guardar el encabezado en otro archivo, no descartarlo. El programa puede modificarse fácilmente para incluir una función de lectura. Tal y como se encuentra ahora, el archivo de entrada debe contener tiempo, presión, temperatura y conductividad en las columnas 1 a 4, en ese orden.

← **Salida:** matrices de tiempo, presión, temperatura y conductividad, separadas en bajadas y subidas. Las columnas impares contienen bajadas y las pares subidas.

Llamada a la función separa.m, dentro de matlab:

[TIME,PRES,TEMP,CNDT]=separa(filename,path)

→ **Entrada:**

filename: nombre del archivo de datos, asume que la extensión del archivo de datos es cnv. Esto se puede modificar en la sección de lectura de la función.

path: trayectoria del archivo de datos

Ejemplo:

[TIME, PRES, TEMP, CNDT] = separa ('Te2o004', 'c:\tehuano2\ondulant\datos\giro1\');

← **Salida:** matrices de datos donde las columnas impares son las bajadas y las pares son subidas.

Requerimientos: función fpasabn, un filtro pasabajo tipo filtro de Godin.

CRITERIO Y VALOR UTILIZADO AQUÍ	SIGNIFICADO
dcrit=0.25;	Velocidad mínima en superficie o fondo al realizar un arrastre (unidades de diferencia de presión). Este valor se puede conocer cargando un archivo de arrastre (*.CNV) en MATLAB y graficando tiempo contra diff(p).
tsalto=100;	Salto en tiempo (segundos) entre bajadas y subidas sucesivas después de su separación (el valor es aproximado, no es importante que sea exacto). Un valor de 100 segundos entre bajadas (o subidas) sucesivas parece funcionar bien para lances a 250-300 m. Para lances más someros quizás haya que disminuir este valor.
nmax=800;	Numero máximo de valores en la vertical, por lance (subida o bajada), una vez separados. Esto depende del instrumento y de la profundidad máxima de los lances. En este caso el SBE19 muestrea a 2Hz. Si un ciclo dura 3 minutos habrá un total de 360 datos, de manera que nmax debe ser > 180. Si se usa un CTD con una razón de muestreo más alta, o si se hacen lances a más de 300 m, habrá que incrementar este valor. Se usa para especificar el número máximo de renglones en la matriz de salida. Aquí nmax=800 es sólo un número grande.
dmin=25;	Profundidad más somera a considerar. Este valor sirve para eliminar valores erróneos 'a priori'. En este caso sabemos que cualquier valor obtenido a profundidades menores que 25 m ocurre solo cuando se sube o se baja el vehículo. Esto porque el ondulante sólo subió a 30-50 m durante un arrastre. Si el ondulante normalmente sube hasta la superficie hay que especificar dmin=0 y eliminar los datos iniciales interactivamente (ver más adelante).
dmax=dmax(pres)/1.5;	Profundidad máxima de los lances (aproximada). Este valor se usa para eliminar marcas erróneas de fin de lance. Se calculó de los datos como máxima (presion) / 1.5.

Comentarios para el uso de la función *separa*:

Esta rutina se probó con datos obtenidos durante el experimento Tehuano II y los criterios de arriba están calibrados con estos datos. Seguramente habrá que cambiar algunos de estos valores al procesar un conjunto distinto de datos.

Esta función genera tres ventanas con gráficos:

1. La primera pantalla (Figura 1b) se usa para escoger los puntos iniciales y finales del arrastre usando el ratón de la computadora. El objetivo es 'escartar oscilaciones indeseables'. Debe utilizarse el zoom para escoger los puntos extremos (Figuras 2a y 2b):

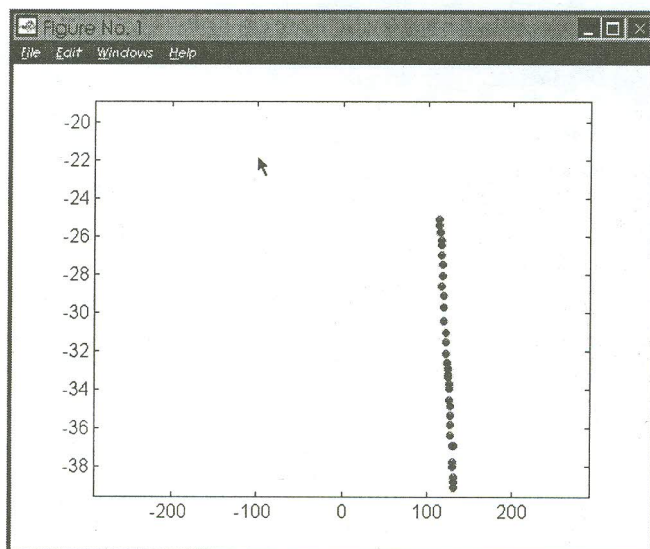


Figura 2a. Zoom de la Figura 1, esquina superior izquierda y posición del cursor para escoger el lance desde el inicio.

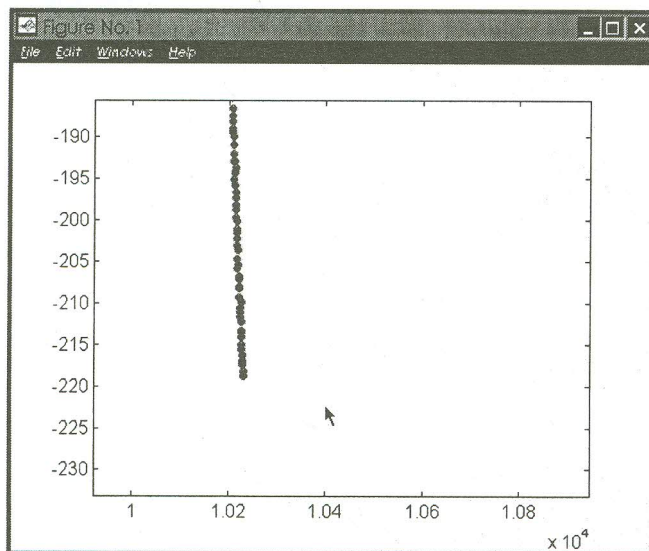


Figura 2b. Zoom de la Figura 1, esquina inferior derecha y posición del cursor para escoger el lance hasta el final.

2. La segunda pantalla (Figura 2c) muestra un panel doble: arriba se grafican los datos separados en subida+bajadas y abajo la serie original después de identificar los puntos finales de cada subida y cada bajada:

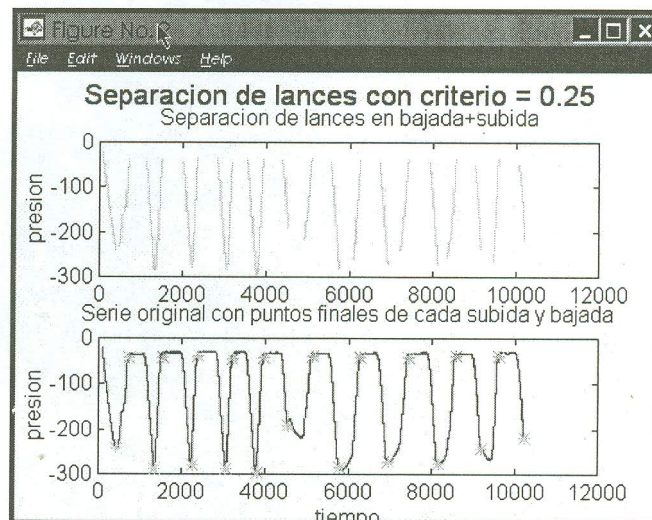


Figura 2c. Presentación de la separación en lances de subida+bajada y finales (*) de lances individuales.

3. La tercera pantalla (Figura 2d) muestra la separación de lances obtenida:

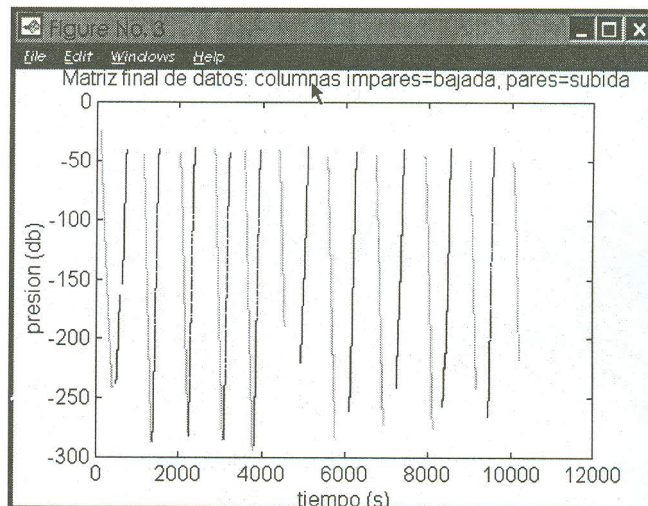


Figura 2d. Separación final de lances.

Si algo de lo que se menciona a continuación no es correcto, hay que repetir el proceso escogiendo una mejor opción de inicio y final de arrastre. Se deben eliminar todas las oscilaciones iniciales, por pequeñas que sean, dado que originan 'lances' espurios de pequeña longitud. Estos **no** podrán procesarse correctamente en fases posteriores. Para corroborar si el procedimiento se realizó correctamente es necesario revisar los puntos siguientes.

- Que los asteriscos que marcan el final de cada lance se encuentren, efectivamente, al final de cada lance. Si no es así, se aborta el proceso (ctrl-C) y se vuelve a escoger el inicio y final del lance.
- Que el número total de asteriscos que marcan el final de cada lance en la pantalla 2 sea igual al número de lances (subidas y bajadas) en la pantalla 3 (Figura 2d). Si no es así se aborta el proceso y se vuelve a empezar.
- En la ventana de matlab (command window, Figura 2e) aparece la variable k, la cual también debe ser igual al número de lances obtenidos. Si no es así se aborta el proceso y se vuelve a empezar:

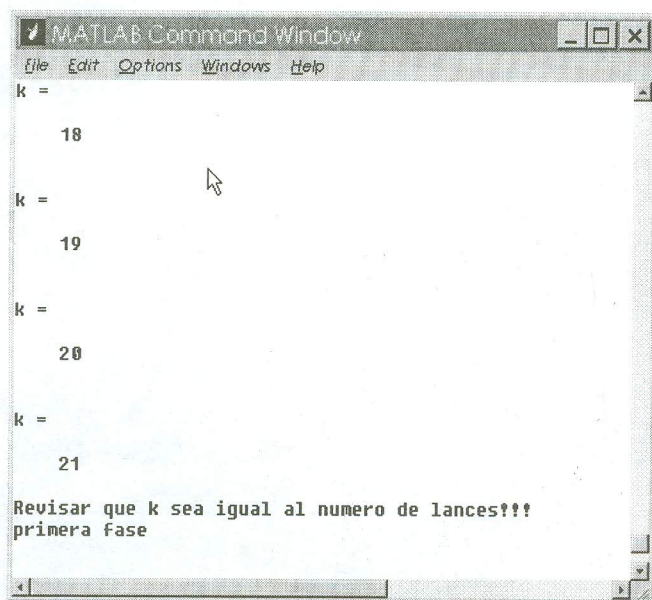


Figura 2e. Ventana de MATLAB presentando el número k de lances.

- Finalmente, es importante que **no** aparezcan puntos iniciales separados del primer lance. Esto usualmente ocurre cuando no se eliminó correctamente alguna oscilación inicial pequeña. En este caso también se debe abortar el proceso y se vuelven a escoger los puntos iniciales y finales. Esto se detecta fácilmente en la gráfica (pantalla 3, Figura 2d) porque el origen del eje de tiempo no coincide con el primer punto del primer lance de bajada.

En resumen, la fuente principal de problemas ocurre cuando no se eliminan correctamente los datos superficiales al inicio y final de lance. Allí hay oscilaciones que el programa confunde fácilmente con lances de bajada o subida. *Se aceptan sugerencias para mejorar esta parte del algoritmo.* Tal como está ahora hay que eliminar todos los datos muy superficiales o iniciales. Se debe ser especialmente cuidadoso con los extremos de la serie cuando el aparato llegó a la superficie muy rápido y/o se mantuvo oscilando en la superficie. Estas oscilaciones son extremadamente difíciles de descartar automáticamente. Por eso es necesario eliminarlas de los datos desde el inicio.

Además, los datos de salinidad que allí se obtuvieran no serían correctos. Este tipo de instrumentos (CTDs de flujo libre) no miden bien a velocidades bajas del flujo.

Una vez que se ha adquirido experiencia con esta parte del programa, el resto del proceso requiere de poca intervención del usuario. En general es un proceso rápido y puede hacerse rutinariamente.

PROCESAMIENTO EN MATLAB, SEGUNDA FASE: OBTENCIÓN DE SECCIONES DE LANCE DE VELOCIDAD CONSTANTE CON LA FUNCIÓN PROINT1.m

→ **Entrada:** TIEMPO, PRESIÓN: matrices de tamaño (n, m) donde m es el número de lances separados por la función *separa* y n es igual a n_{max} (proporcional a la profundidad).

← **Salida:** TLIM: tiempo en que se calcula el promedio de velocidad. VLIM: velocidades promedio cada 10s.

Para encontrar el defase temperatura-conductividad que elimina el mayor número de picos de salinidad, se divide el lance en secciones de 10 segundos y se calcula la velocidad promedio en cada sección. Durante este intervalo de tiempo el promedio de la velocidad caracteriza bien el comportamiento del vehículo ondulante. La velocidad promedio en cada intervalo determinará el defase a aplicar en el paso siguiente (*proint2*).

Algunos criterios incluidos en el programa son:

1. z_{max} No. máximo de secciones de velocidad constante. Debe ser algo como el tiempo mayor (s) para realizar un lance, dividido entre 10 (s).
2. DT Paso de interpolación de 1 segundo (estándar, no se debe modificar sin hacer pruebas extensas).

Llamada a la función *proint1.m*, dentro de matlab:

$[TLIM, VLIM] = \text{proint1}(\text{TIME}, \text{PRES})$

→ **Entrada:** matrices TIME, PRES (de *separa*)

← **Salida:** vector TLIM: tiempos del promedio de velocidad cada 10s. Vector VLIM: velocidades promedio cada 10s

Requerimientos: *lintrp* (interpolador lineal), *fpassbn* (filtro pasa bajo).

Comentarios sobre la función *proint1*:

Esta función determina la bondad de los defases que se realizarán en la fase siguiente (*proint2*). Si el vehículo se comporta bien, es decir, si los perfiles de velocidad contra tiempo (curva de aceleración) siguen una forma parabólica, los defases funcionarán razonablemente bien al calcular promedios de velocidad cada 10 segundos. Si el instrumento tiene una razón de muestreo mayor a 2 Hz y si las curva de aceleración fueran

más complicadas que una parábola, quizás sería conveniente disminuir DT y reprogramar esta función para calcular promedios en lapsos más pequeños (incrementando z_{max} de acuerdo a esto). En general el usuario no tendría que modificar esta función. Dada la razón de muestreo del SBE19 (2 datos por segundo) no es necesario calcular estos promedios en intervalos menores que 10 segundos.

Para este grupo de datos, como se explicará en la fase siguiente, el comportamiento de los lances de bajada no es bueno y se desechan del proceso en la fase siguiente. Esto no afecta la resolución horizontal del instrumento dado que subidas y bajadas se realizan casi en la misma posición. Hay que recordar que la bajada sucede durante un viraje de 360° y la subida al salir del viraje. Sin embargo, esto afecta la resolución vertical de las observaciones. Los sensores miden agua no perturbada solamente durante el lance de bajada. Es decir, al sólo poder usar los lances de subida, los sensores miden una versión 'suavizada' del perfil vertical. El suavizado se debe a la turbulencia inducida por el cuerpo del vehículo y no es una función conocida. Sin embargo, esto no representa un inconveniente importante porque el instrumento no está diseñado para medir estructura fina sino para revelar estructuras de meso y submesoescala. Estas tienen escalas de varios metros en la vertical y de decenas a cientos de kilómetros en la horizontal.

En el análisis de varios arrastres realizados con el CTDO/ SBE19, los promedios de velocidad en lances individuales sugieren que hay que mejorar dos aspectos del vehículo:

- La aceleración de bajada debe ser más regular. Hay que mejorar el diseño para obtener una curva parabólica de aceleración durante los lances de bajada.
- Los sensores deben colocarse físicamente en un lugar óptimo para aprovechar todos los datos. Esto implica encontrar la manera de tener sensores que midan flujo no perturbado durante la subida y la bajada del instrumento. Una opción es la colocación de sensores redundantes en posiciones opuestas.

Operación de la función *proint1*:

La función presenta en la pantalla todas las gráficas de los lances de subida y bajada. En principio hay que revisarlos todos para escoger aquellos cuyos promedios de velocidad (curvas de aceleración, marcadas con asteriscos) tengan un perfil parabólico (Figura 3). Esto servirá para descartar los lances inservibles en el paso siguiente (*proint2*). En el caso del grupo de datos usados aquí, los lances de bajada se deben descartar inmediatamente por no tener curvas regulares de aceleración. Sin embargo, se presentan las gráficas de los lances de bajada para mantener la capacidad de analizar el conjunto completo de datos. Con los lances de subida vale la pena experimentar. En la mayoría de los casos las curvas de aceleración serán parábolas regulares (Figura 3c). Por otro lado, las curvas muy irregulares se deben descartar inmediatamente. Sin embargo,

otras tendrán curvas aproximadamente parabólicas y otras aún con forma de parábolas dobles (Figura 3b, esto sucede cuando el vehículo se detiene en la subida). Todas estas pueden ser formas aceptables, aunque mientras más irregular sea la curva, más picos aparecerán en el perfil de final de salinidad.

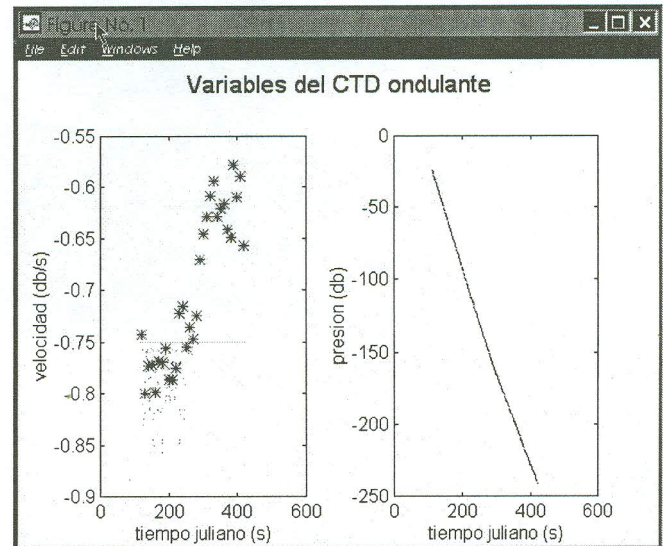


Figura 3a. Lance de bajada típico, con curva de aceleración (izquierda) irregular. Se descarta.

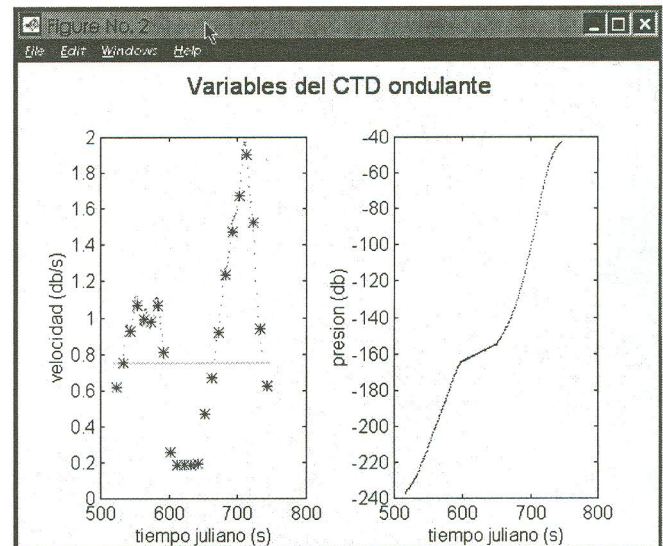


Figura 3b. Lance de subida atípico, con curva de aceleración (izquierda) irregular. Este caso producirá perfiles de salinidad con más picos que uno con un perfil regular (ver Figura 3c).

PROCESAMIENTO EN MATLAB, TERCERA FASE: FUNCIÓN *PROINT2.m*

- ➔ **Entrada:** TIME: TLIM, VLIM de *proint1*
TIME, PRES, TEMP, CNDT matrices separadas (de separa).
- ⬅ **Salida:** TI, CO, TE, PR, SA matrices de tiempo, conductividad, temperatura, presión y salinidad.

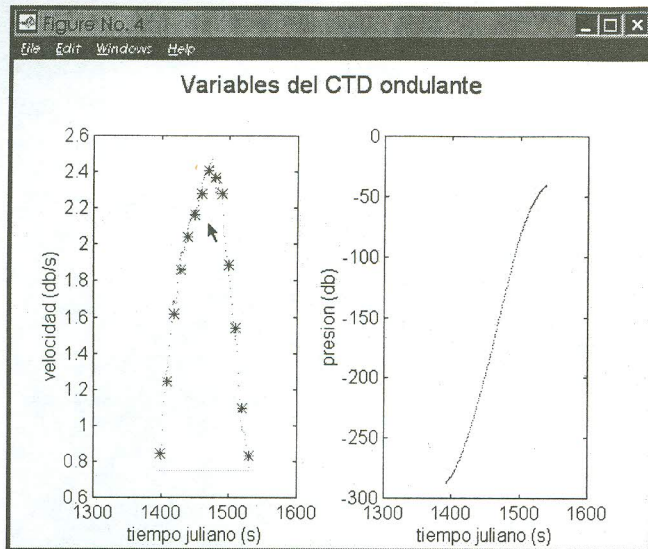


Figura 3c. Lance de subida típico, con curva de aceleración parabólica regular. Este lance producirá los mejores resultados.

Esta función aplica diferentes defases de temperatura relativos a la presión a cada una de las secciones de velocidad constante (obtenidas en *proint1*), dependiendo de la velocidad promedio del segmento. Después de obtener las series defasadas, el programa calcula la salinidad. **Actualmente funciona con lances de subida.** Sería fácil modificar esta función para procesar lances de bajada, si estos tuviesen la calidad requerida en sus curvas de aceleración. Para hacerlo sería necesario modificar el modelo de defasamiento. Éste solamente se ha probado para lances de subida por no contar con datos de bajada de buena calidad.

Esta función debería trabajar sin intervención del usuario en sus parámetros internos. Si algún usuario detectara problemas podría reportarlos a trasvi@cicese.mx, incluyendo una muestra del archivo problema.

Llamada a la función *proint2.m*, dentro de matlab:

[TI, CO, TE, PR, SA] = proint2 (TLIM, VLIM, TIME, PRES, TEMP, CNDT)

→ **Entrada:** TLIM, VLIM de *proint1*
TIME, PRES, TEMP, CNDT matrices separadas (de separa)

← **Salida:** TI, CO, TE, PR, SA matrices de tiempo, conductividad, temperatura, presión y salinidad.

Requerimientos: *fpasabn* (filtro pasa bajo), *lintrp* (interpolador lineal), *sw_salt* (para el cálculo de salinidad, librería para MATLAB seawater del CSIRO).

Operación de la función *proint2*:

Esta función hace varias preguntas, la primera:

quieres especificar los lances (n o s)?

Por omisión se escoge procesar todos los lances de subida. Si en el paso anterior se encuentra algún lance de subida que no tenga una curva parabólica de aceleración, éste puede omitirse aquí. Para esto se responde *si* (sólo se escribe la 's', sin comillas desde luego). Esto da lugar a la siguiente pregunta:

vector de lances? (ejemplo: 2:2:10)

Aquí se responde con un vector de lances en notación de MATLAB. El ejemplo 2:2:10 es equivalente a procesar los lances [2 4 6 8 10]; hay que recordar que los lances pares son los lances de subida. Si, por ejemplo, se desea omitir el lance 4, la respuesta sería como [2 6 8 10].

A continuación se inicia el procesamiento de cada lance. Esto abre 3 ventanas: la primera (Figura 4a), que marca el avance del programa en una gráfica de temperatura contra profundidad; la segunda (Figura 4b), que grafica la salinidad obtenida en cada segmento de 10 s y la tercera, que grafica el perfil de salinidad completo (Figura 4c). Al final de cada lance hay que dar 'return' para avanzar al siguiente.

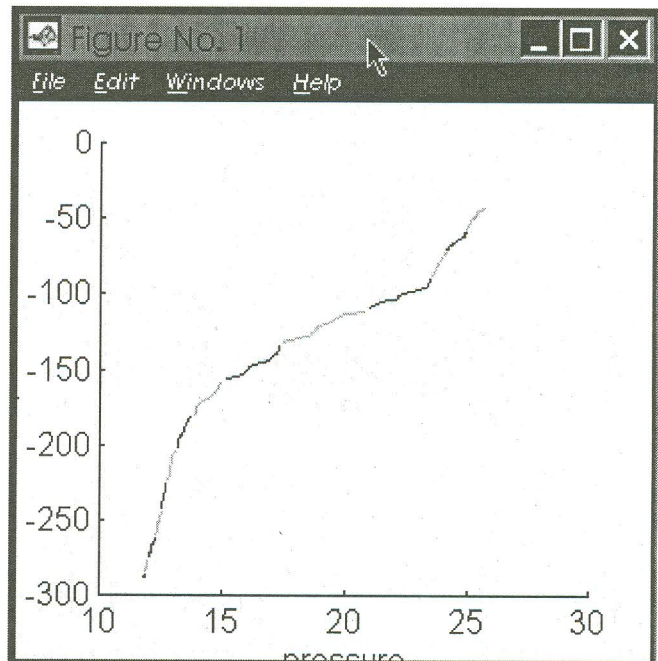


Figura 4a. Perfil de temperatura. Los diferentes colores marcan los segmentos de 10 s escogidos para efectuar los defases.

PROCESAMIENTO EN MATLAB, CUARTA FASE: FUNCIÓN *PROINT3.m*

→ **Entrada:** TI, CO, PR, SA (de *proint2*)

← **Salida:** Archivo *.mat que contiene los vectores TIM CON SAL PRE TEM LON LAT, es decir tiempo, conductividad, salinidad, presión, temperatura, longitud y latitud, para todo el arrastre de CTDO.

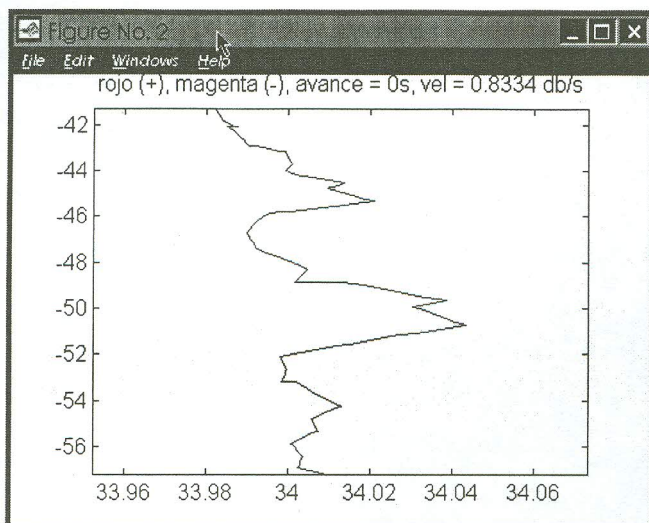


Figura 4b. Segmento de perfil de salinidad.

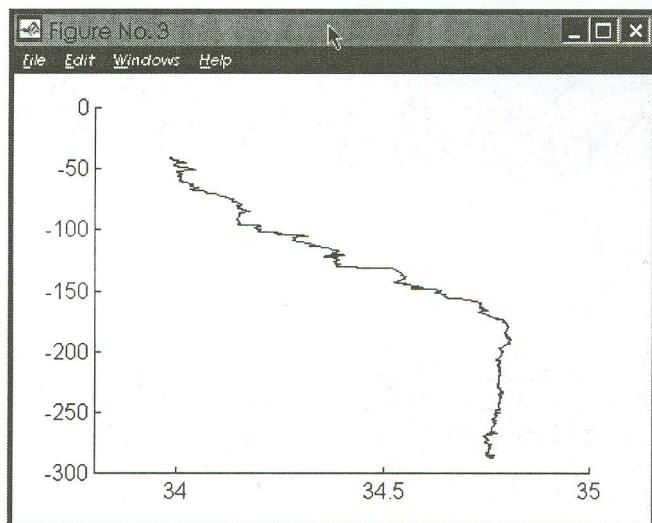


Figura 4c. Perfil completo de salinidad.

Esta función permite editar las salinidades, las suaviza y las integra con la navegación. Ésta, a su vez, llama a la función *navega*, que sincroniza los tiempos con las posiciones geográficas. Esta versión de *navega* requiere de un archivo de navegación, pero es una función sencilla que puede ser sustituida por el usuario, dependiendo de sus necesidades. El encabezado de *navega* explica sus características principales:

```
function [lon, lat]=navega(otime,archnav,ext);
% función que integra tiempo con navegación usando
% un archivo de datos de gps que contiene:
% día juliano decimal, longitud, latitud
%
% entrada:
% otime = tiempo promedio del lance (para ondulante)
% archnav = nombre del archivo de navegación
% ext = extensión del archivo ('.dat', por ejemplo)
%
% ATC, CICESE en BCS, 1/07/98
```

Llamada a la función *proint3.m*, dentro de matlab:

proint3 (TI, CO, TE, PR, SA, nombre)

➔ **Entrada:** TI, CO, PR, SA (tiempo, conductividad, presión, salinidad, de *proint2*)

nombre = nombre del archivo de salida

⬅ **Salida:** archivo 'nombre'.mat que contiene los vectores TIM CON SAL PRE TEM LON LAT.

Requerimientos: *fpasabn* (filtro pasabajo), *julday* (cálculo de día juliano), *navega* (integrador de tiempo y navegación).

Operación de la función *proint3*:

La función inicia con una pregunta al operador:

fecha inicial (ddmmaa hh:mm)?

La respuesta debe ser algo como: 230296 08:17 (no omitir ningún campo)

es decir: 23 de febrero de 1996 a las 8:17 de la mañana.

En seguida el programa presenta una ventana con la gráfica de salinidad y pregunta:

deseas eliminar secciones del lance? (s o n)

Si la gráfica de salinidad presenta picos que son evidentemente erróneos (Figura 5a), aquí se tiene la oportunidad de eliminarlos. Lo que hace el programa es sustituir un rango de valores con NaNs (not-a-number). Esto a veces es necesario cerca de la superficie o en secciones de la termoclina que no se pudieron defasar correctamente para el cálculo de la salinidad. En este caso se responde 's' y el programa pregunta entonces si se desea hacer un zoom de la ventana para escoger la región a eliminar.

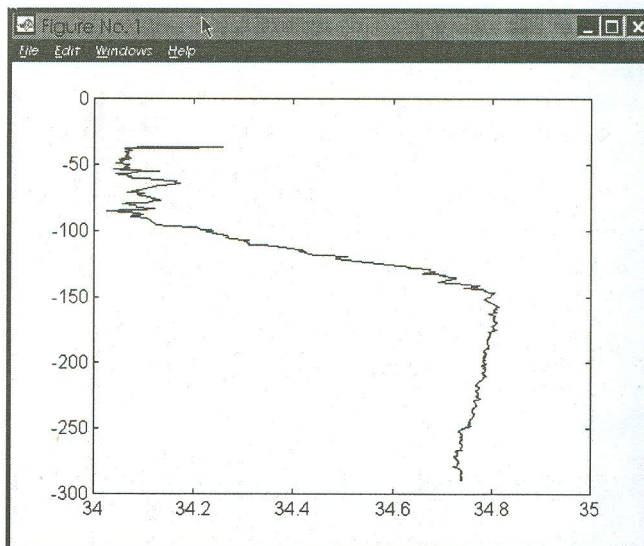


Figura 5a. Perfil de salinidad con datos erróneos en superficie.

deseas hacer zoom?

Si la respuesta es 's', el programa responde:

escoge con el mouse el rango de profundidades (primero la sup.) (ver Figura 5b).

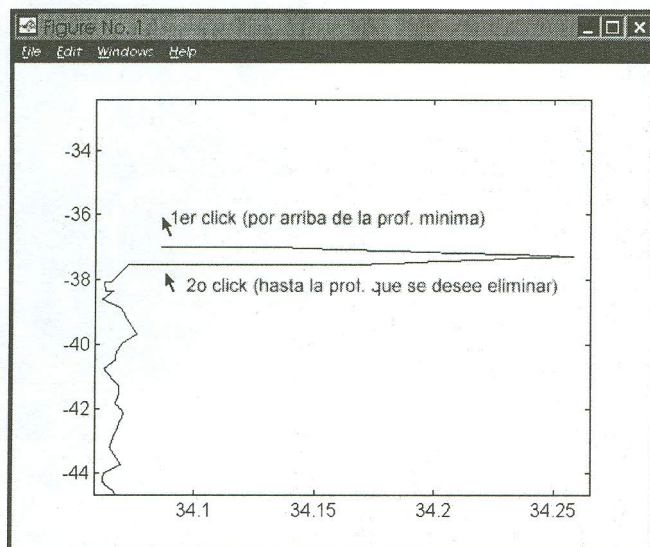


Figura 5b. Eliminación de datos con el cursor.

Luego aparece un cursor sobre la ventana del gráfico de salinidad vs. profundidad. Una vez que se escogen las dos profundidades, la ventana se actualiza con los datos eliminados (Figura 5c). El programa pregunta entonces:

deseas continuar? (s o n)

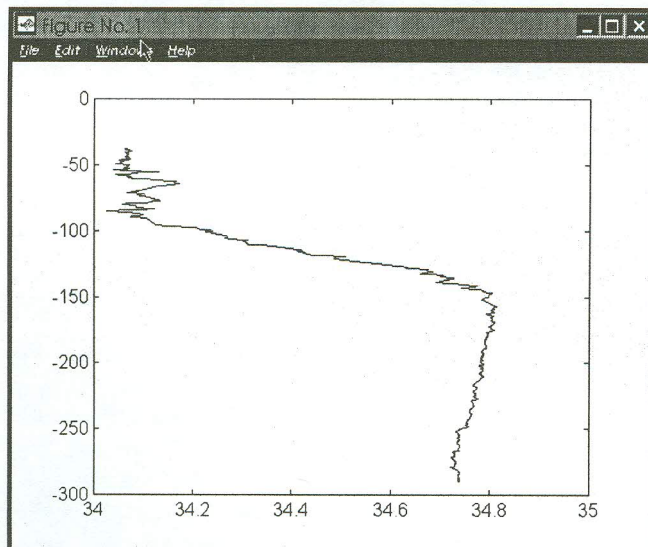


Figura 5c. Perfil actualizado.

En caso de responder afirmativamente, se tiene oportunidad de eliminar alguna otra sección de datos erróneos. Si se responde que no, el programa pasa a suavizar los datos y actualiza la gráfica con los datos suavizados en un color diferente,

superpuestos sobre los datos originales (Figura 5d). Después de hacer esto, el programa anuncia:

lance terminado (enter para continuar)

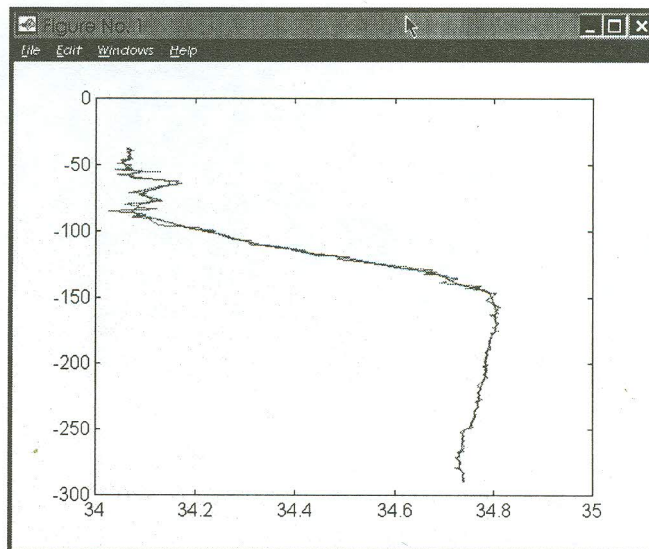


Figura 5d. Perfil final suavizado.

Y al presionar la tecla 'enter' se procede a repetir el proceso con el lance siguiente del arrastre.

Al finalizar el proceso, el programa escribe los vectores de datos a un archivo 'nombre'.mat.

Programa principal en MATLAB: prondul.m

Finalmente, se incluye un ejemplo de un programa principal que puede ser utilizado para correr las cuatro fases del procesamiento en MATLAB.

```
%programa que llama a todas las funciones para el
%procesamiento de CTD ondulante.
%
% OJO al correr datcnv (antes de este programa)
%
% Los datos de entrada para la función separa deben ser:
% El tiempo en segundos
% La conductividad en mS/cm
% La temperatura en T68
% La presión en db
%
% ATC, CICESE en BCS, 30/06/98

close all, clear
nombre='Te2o004';
[TIME, PRES, TEMP, CNDT] = separa (nombre,
'c:\tehuano2\ondulant\datos\giro1\');
disp('primera fase');
pause
close all
[TLIM, VLIM] = proint1 (TIME, PRES);
```

```

disp('segunda fase');
pause
close all
%La siguiente función corre de manera interactiva,
% el vector de lances se da como: 2.2:10
[TI, CO, TE, PR, SA] = prout2(TLIM, VLIM, TIME, PRES,
    TEMP, CNDT);
clear ans TLIM VLIM TIME PRES TEMP CNDT
disp('tercera fase');
close all, clear ans
%La última fase convierte el tiempo de segundos a
%días julianos y permite editar y suavizar los datos
%de salinidad. Finalmente, integra la navegacion y
%salva a un archivo 'nombre'.mat
prout3(TI, CO, TE, PR, SA, nombre);
clear

```

Cómo obtener estos programas.

Por solicitud mediante mensaje electrónico a trasvi@cicese.mx

UN EJEMPLO DE DATOS PROCESADOS POR ESTE MÉTODO

Durante el experimento Tehuano 2 (febrero-marzo de 1996) se hizo uso intensivo del CTDO. Se recorrieron varios giros con este procedimiento y se logró incrementar la resolución horizontal de manera significativa. A continuación se comparan secciones de densidad de la línea Giro 1 (Figura 6a). Ésta fue recorrida simultáneamente con lances tradicionales de CTD (SBE9/11) y con el CTDO descrito en este artículo.



Figura 6a. Imagen AVHRR del Giro 1 el 11 de febrero de 1996. Los tonos oscuros del interior del giro corresponden a temperaturas superiores a 28°C. Figura 6a. Imagen AVHRR del Giro 1 el 11 de febrero de 1996. Los tonos oscuros del interior del giro corresponden a temperaturas superiores a 28°C.

Las Figuras 6a y 6b muestran el giro mediante su firma superficial en el infrarrojo (Figura 6a) y mediante las trayectorias de 4 flotadores ARGOS liberados en su interior (Figura 6b, aquí se muestra también la trayectoria seguida para la línea Giro 1). Es evidente que se trata de un giro cálido, anticiclónico. Las Figuras 6c y 6d comparan la información obtenida mediante el método tradicional (Figura 6d) y el CTDO (Figura 6c). En ambas se destaca la forma cóncava de la picnoclina que es típica de los giros cálidos. Nótese también que el CTDO no llega a superficie y que no alcanza a registrar adecuadamente las isolíneas de menos de 23 kg m⁻³. Sin embargo, la sección del CTDO muestra claramente una 'lente' que ocupa la mitad oeste de la picnoclina. Esta característica se pierde en la sección de CTD debido a la menor resolución horizontal del campo de masa. Es claro que para entender la dinámica de este giro no debemos obviar la presencia de una lente tan conspicua y, sin embargo, prácticamente indetectable por los métodos tradicionales.

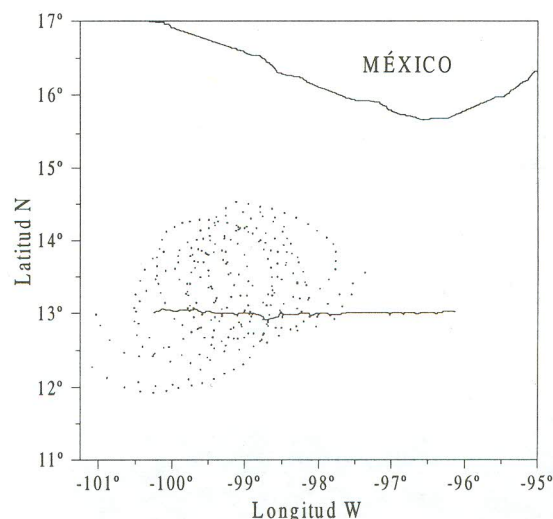


Figura 6b. Posiciones de flotadores dentro de Giro 1 (11 de febrero al 7 de marzo) y trayectoria de la línea Giro 1 (línea sólida).

CONCLUSIONES

Este método permite la observación de características que pueden modificar nuestro conocimiento del funcionamiento de estructuras dinámicas de mesoescala y submesoescala. Se muestra también que es posible observar la estructura horizontal 'fina' del océano con instrumentos y herramientas que ya están disponibles para investigadores en nuestro país.

Por otra parte, el análisis de los datos que se obtienen con el CTDO ondulante revela un problema importante del método actual. Este consiste en lograr que la velocidad de subida y bajada del instrumento sea uniforme. El requisito más importante para lograr buenas alineaciones de temperatura y conductividad es que el perfil de aceleraciones sea parabólico. Tanto el diseño del instrumento como la forma de trabajar en el

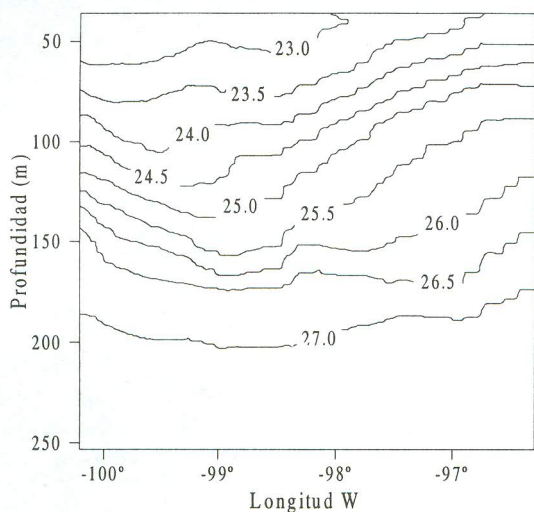


Figura 6c. Sección de densidad a lo largo de Giro 1, CTD ondulante. Nótese que no llega a la superficie.

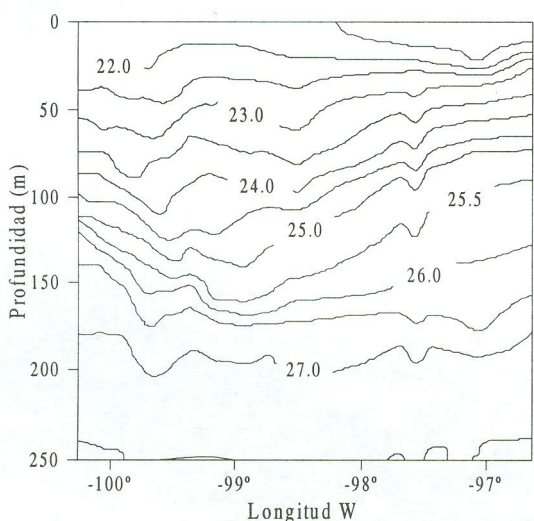


Figura 6d. Sección de densidad a lo largo de Giro 1, CTD convencional.

mar deben tomar esto en cuenta para obtener la máxima información posible del instrumento.

Finalmente, existen por lo menos dos versiones de CTDOs como el de FMT96. Anatoly Filonov trabaja actualmente con un CTDO de ala doble que contiene un SBE19 con razón de muestreo de 2 Hz. Es una versión mejorada del que se usó para coleccionar las observaciones que se presentan aquí. Este instrumento ha dado buenos resultados en el trabajo de campo y se le utiliza rutinariamente para perfilar entre la superficie y los 100 ó 150 m. El autor de este artículo construyó un CTDO de ala sencilla al que se le puede agregar un ala doble para mayor sustentación. Este CTDO incluye un CTD SBE25 con sensores externos y razón de muestreo de 8 Hz. Pruebas realizadas en el campo muestran que este CTDO puede perfilar entre 5 y 300 m de profundidad. Los interesados en utilizar

vehículos ondulantes pueden contactar a A. Filonov (afilonov@udgserv.cencar.udg.mx) para obtener detalles de su construcción y operación.

AGRADECIMIENTOS

Quiero agradecer en primer lugar el apoyo entusiasta de Anatoliy Filonov, quien proporcionó el CTDO usado para este trabajo. Los comentarios de María Luisa Argote contribuyeron a mejorar el texto sustancialmente.

Este trabajo se realizó con apoyo del CONACyT y de la División de Oceanología del CICESE.

REFERENCIAS

- Filonov, A.E., Monzón C.O., Tereshchenko I.E., 1996, A technique for fast conductivity-temperature-depth oceanographic surveys. *Geofísica Internacional*, vol. 35, N. 4, p. 415-420.
- Pollard, R.T., 1986, Frontal surveys with a towed profiling conductivity/temperature/depth measurement package (SeaSoar). *Nature*, Vol. 232, 2 October, pp 433-435
- Pollard, R.T. and Regier, L., 1990, Large variations in potential vorticity at small horizontal scales in the upper ocean. *Nature*, Vol. 348, 15 November, pp 227-229.
- Trasviña, A., Barton, E.D., Brown, J., Vélez, H.S., Kosro, M. and Smith, R.L., 1995, Offshore Wind Forcing in the Gulf of Tehuantepec, Mexico: the asymmetric circulation. *Journal of Geophysical Research*, OCEANS, Vol. 100, No. C10, pp. 20649-20663.