

**ADCP de Coque
Campagne DRAKE
A bord du POLARSTERN
Janvier-Février 2006**



Rapport Interne LOCEAN
Septembre 2007



*Laboratoire D'Océanographie et du Climat :
Expérimentation et Approche Numérique*

**UNITE MIXTE DE RECHERCHE 7159
CNRS / IRD / UNIVERSITE PIERRE & MARIE CURIE
INSTITUT PIERRE-SIMON LAPLACE**



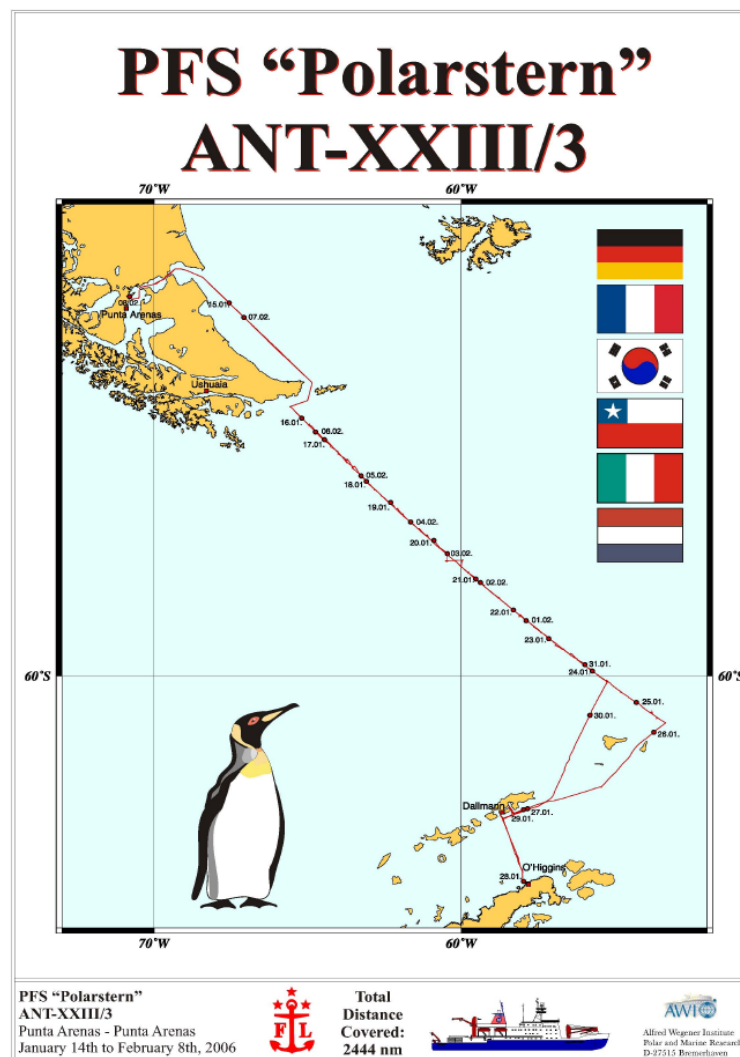
SOMMAIRE

Chapitre 1	La campagne	1
Chapitre 2	Acquisition des données	3
Chapitre 3	Chargement des fichiers bruts	5
Chapitre 4	Obtention et vérification de la navigation	7
Chapitre 5	Edition des données ADCP	9
Chapitre 6	Calibration des données	11
	Section 1 : Qualité des positions	11
	Section 2 : Erreur sur le cap	13
	Section 3 : Alignement du transducteur	15
	Section 4 : Résultats de la calibration	16
	Section 5 : Mise en forme définitive de la base de données	17
	Section 6 : Evaluation de l'erreur	18
	Section 7 : Obtention des profils moyennés en station	19
	Section 8 : Comparaison avec le LADCP	19
	Section 9 : Evaluation de l'influence de la marée	20
	Section 10 : References	21
	Section 11 : Remerciements	21
Chapitre 7	Présentation rapide des données	23
ANNEXE I	Système de navigation	27
ANNEXE II	Configuration de l'ADCP	28
ANNEXE III	Fichier de contrôle du logiciel VMDAS	32
ANNEXE IV	Procédure de calibration du gyro-compas (E. Firing)	34
ANNEXE V	Modèle de Marée OTPS	38

Chapitre 1

La campagne

La figure montre le trajet effectué.



Chapitre 2

Acquisition des données

La navigation est acquise par le biais d'une centrale inertielle MINS (Marine Inertial Navigation System) du constructeur Raytheon, dont les caractéristiques sont données en ANNEXE I.

Les données ADCP ont été acquises au moyen du logiciel RDI VMDAS, version 1.42. L'ADCP de coque du POLARSTERN, (Profileur acoustique à effet Doppler) est du type Ocean Surveyor Broadband/Narrowband RDI, 150 KHz, Firmware version 23.14.

Le détail de la configuration est donné en ANNEXE II. L'appareil est configuré en mode "Narrow Band" pour une meilleure pénétration (portée maximum de 500m). La configuration choisie est une longueur d'ensemble de 2 minutes, pour les fichiers "Short Term Averaging", et de 20 minutes pour les fichiers "Long term Averaging", respectivement LTA et STA pour le logiciel VMDAS. La longueur de bin est de 4m (la configuration nominale pour la fréquence de 150 KHz serait de 8m).

L'immersion réelle du transducteur est de 11 m. La figure suivante donne le détail de la "fenêtre" où est installé l'ADCP. Elle est remplie d'eau de mer.

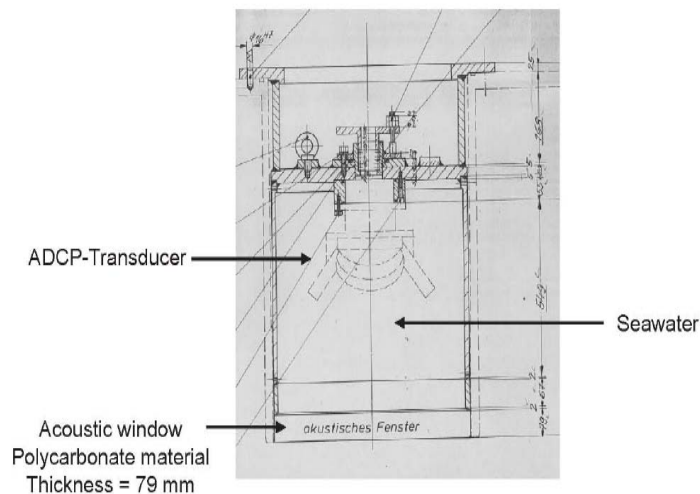


FIG. 2.1: Fenêtre ADCP

Le transducteur est situé 20 m à l'avant et 3 m à babord de la centrale inertielle. Un programme de la routine CODAS de traitement des données (corags) permet d'évaluer et de corriger l'influence de la distance du transducteur au système d'acquisition de la navigation. La figure ci-dessous illustre ces positions.

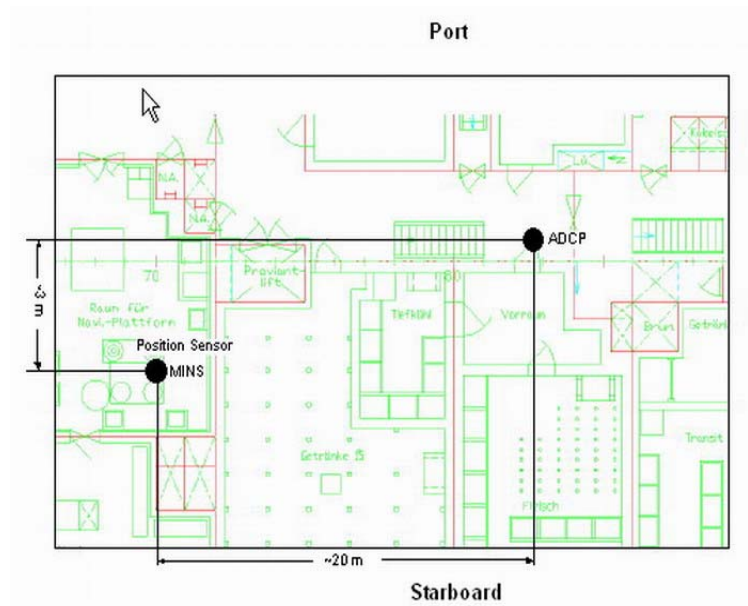


FIG. 2.2: Positions de l'ADCP et de la centrale inertielle

Le traitement proprement dit se déroule en 5 phases décrites ci-dessous, auxquelles on rajoute l'évaluation des erreurs, et dans le cas de DRAKE, la comparaison avec les données LADCP :

Chargement des fichiers bruts . Cette phase comporte aussi une première élimination des données non enregistrables (erreurs de chronologie, mauvaises durées d'ensembles, etc...).

Obtention et Vérification de la navigation . Il est préférable d'avoir une acquisition des données de navigation indépendante de l'acquisition ADCP .

Edition des données ADCP . Cette phase, très importante, permet d'éliminer toutes les mesures mauvaises ou douteuses et de repérer le fond si nécessaire.

Calibration des données . En deux temps, au moyen des données de navigation : correction des données de cap puis des défauts d'alignement du transducteur.

Mise en forme définitive . Introduction des données de navigation dans la base ADCP. Contrôle de la qualité des données. Construction de différentes grilles, spatiales et/ou temporelles, permettant de tracer les graphiques de présentation et de créer des fichiers ascii pour la communication.

Chapitre 3

Chargements des fichiers bruts

Le logiciel VMDAS permet l'acquisition de tout un ensemble de fichiers, dont seuls certains, ayant déjà subi un prétraitement, seront utilisés :

- .ENR, .ENS, .ENX** Fichiers totaux (monopings) de tous les ensembles, données brutes ("Beam Coordinates").
- .N1R** Fichiers des trames NMEA de navigation.
- .LTA, .STA** Fichiers moyennés par le logiciel VMDAS à partir des fichiers monopings, .LTA sur la longue durée choisie dans le fichier de configuration (20 mn), .STA sur la courte durée (2mn), données prétraitées ("Earth Coordinates"). Nous utiliserons les fichiers .STA dans la suite. Ces fichiers sont créés à chaque arrêt-remise en route du système d'acquisition.
- .LOG** Fichier Log enregistrant les commandes RDI exécutées à chaque remise en route du logiciel d'acquisition. Un exemple est donné en ANNEXE III.
- .VMO** Copie du fichier configuration, créée à chaque remise en route du logiciel d'acquisition.

Le logiciel CODAS permet le chargement en deux temps à partir des fichiers .STA ou .LTA (équivalents pour le traitement CODAS).

La première opération se fait sous Matlab (programmes load_sta.m ou load_sta_manual.m). Le programme décode les fichiers .STA, et crée plusieurs fichiers en sortie :

- .cmd** Fichiers exécutables de commande qui seront utilisés dans la procédure suivante pour la fin du chargement.
- .bin** Fichiers de données ADCP binaires.
- .gps1, .gps2** Fichiers ASCII de navigation, .GPS1 en début d'ensemble moyenné, .gps2 en fin.

La deuxième opération est le lancement de l'exécution des fichiers de commande .cmd, pour charger la base de données proprement dite.

A ce stade, on dispose d'un répertoire nommé généralement adcpdb (pour ADCP Data Base), dans lequel les données (binaire) ADCP sont maintenant organisées en blocs, dont la longueur et le contenu sont définis par l'utilisateur au moyen d'un fichier spécial de définition (vmadcp.def), placé dans le répertoire de chargement.

Pour la campagne DRAKE, on a choisi de séparer les deux sections en travers du Passage de Drake (DRAKE 1 pour l'aller, DRAKE 2 pour le retour), de l'ensemble des trajets dans le Sud (DRAKE 3). On a donc trois bases de données distinctes :

DRAKE 1 2006/01/16 06 :42 :44 à 2006/01/26 09 :26 :50
DRAKE 2 2006/01/31 00 :09 :55 à 2006/02/06 23 :51 :12
DRAKE 3 2006/01/26 09 :36 :48 à 2006/01/30 23 :59 :54

Plusieurs outils permettent de vérifier le contenu de la base de données, et d'obtenir des informations utiles :

La commande *lst_block* permet d'obtenir le nombre, la longueur(nombre de profils) et les dates des extrémités des blocs :

```
/* Database summary for : drake */
/* Number of blocks : 45 */
/* 0 n= 400 drake001.blk */ 2006/01/15 20 :34 :44 to 2006/01/16 09 :52 :42
/* 1 n= 111 drake002.blk */ 2006/01/16 09 :54 :42 to 2006/01/16 13 :34 :42
...
```

La commande *showdb* permet d'obtenir la structure précise de la base de données. L'exemple ci-dessous donne la liste des variables de la base DRAKE 1. La commande permet aussi de connaître le contenu d'une variable.

num	name	units	v-ty	ac	ac0	ac1
0	DEPTH	m	3	1	17192	160
7	AMP-SOUND-SCAT	none	1	2	0	0
8	U	m/s	3	2	1	1
9	V	m/s	3	2	2	2
32	PROFILE-COMMENTS	none	2	2	3	3
33	BLOCK-COMMENTS	none	2	1	0	0
34	PROFILE-FLAGS	none	1	2	4	4
35	CONFIGURATION-1	none	11	1	17352	76
36	CONFIGURATION-2	none	11	1	0	0
37	ANCILLARY-1	none	11	2	5	5
38	ANCILLARY-2	none	11	2	6	6
39	ACCESS-VARIABLES	none	11	2	7	7
54	W	m/s	3	2	8	8
55	ERROR-VEL	m/s	3	2	9	9
56	PERCENT-GOOD	none	1	2	10	10
57	PERCENT-3-BEAM	none	1	2	11	11
58	SPECTRAL-WIDTH	none	1	2	12	12
59	U-STD-DEV	m/s	3	2	13	13
60	V-STD-DEV	m/s	3	2	14	14
61	W-STD-DEV	m/s	3	2	15	15
62	EV-STD-DEV	m/s	3	2	16	16
63	AMP-STD-DEV	none	0	2	17	17
64	RAW-DOPPLER	none	3	2	18	18
65	RAW-AMP	none	1	2	19	19
66	RAW-SPECTRAL-WIDTH	none	11	2	20	20
67	BEAM-STATS	none	1	2	21	21
68	NAVIGATION	none	11	2	22	22
69	BOTTOM-TRACK	none	11	2	23	23
75	USER-BUFFER	none	10	2	24	24
76	ADCP-CTD	none	11	2	25	25

Chapitre 4

Obtention et vérification de la navigation

Une commande du logiciel CODAS permet d'obtenir à partir de la base ADCP chargée précédemment un fichier des caps moyens suivis pendant les ensembles ADCP (suffixe .hdg). On utilise les deux premières colonnes de ce fichier : la date "CODAS" en première colonne, et le cap moyen suivi pendant la durée de l'ensemble (2 minutes) en deuxième colonne. Ce fichier servira par la suite de référence pour obtenir les dates des ensembles, converties en unités spécifiques au logiciel CODAS (jours décimaux plus nombre de secondes dans le jour)...

La calibration effectuée ultérieurement par "water tracking" permettra de déterminer, non sans mal, quel type de mesure du cap est transmise à l'ADCP. A ce stade du traitement on ne dispose que des deux fichiers positions .gps1 (en principe heure de début des ensembles), et .gps2, (heure de fin des ensembles). Ces fichiers "positions", comprennent pour chaque ligne un temps, exprimé en unités spécifiques CODAS, une latitude et une longitude (degrés centièmes).

Le système d'acquisition centralisé du bord permet l'obtention d'un fichier "PODAS" (POLarstern Data Acquisition System). A priori, et selon les informations transmises par l'ingénieur électronicien, toutes les données de navigation transmises à l'ADCP proviennent de la centrale inertielle MINS. La vérification des données de navigation sera effectuée au moment du "water tracking".

Les différentes sources de positionnement répertoriées dans le fichier PODAS sont les suivantes :

MINS scientific navigation platform Centrale inertielle

NACOS ship navigation system Pas de détails

Ashtec Z-12 DGPS GPS différentiel

Trimble 1 et 2 GPS 2 GPS classiques

Leica MX400 GPS GPS classique

De même que pour le cap, les problèmes d'horloge ne seront visibles clairement qu'au moment de la calibration par water-tracking. Les trois fichiers dont on dispose, .hdg, .gps1 et .gps2, ne sont pas à la même date. Selon l'ingénieur électronicien, l'horloge ADCP (donc .hdg) serait l'horloge du GPS Trimble 1. Il y a vraisemblablement un décalage entre cette horloge et celle de la centrale MINS, mais il n'a pas été possible de corriger ce défaut.

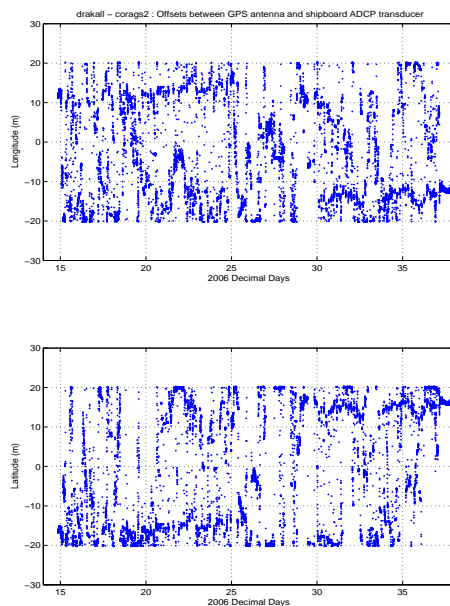
.hdg 14.857454 : 2006/01/15 20 :34 :44.02

.gps1 14.8560301 : 2006/01/15 20 :32 :41.00 (1er enregistrement)

.gps1 14.8574190 : 2006/01/15 20 :34 :41.00 (2ème enregistrement)

.gps2 14.8574074 : 2006/01/15 20 :34 :39.99 (1er enregistrement)

Après correction au moyen de la procédure corags, pour tenir compte de la distance entre le transducteur ADCP et le système d'acquisition de la navigation, le fichier de type .gps servira de base pour tous les calculs de calibration ultérieurs ("water tracking"). La figure suivante donne la correction effectuée en latitude et longitude par la procédure corags sur le fichier .gps2.



La carte suivante montre le tracé des données de navigation, d'après le fichier .gps2 corrigé au moyen de la procédure corags.

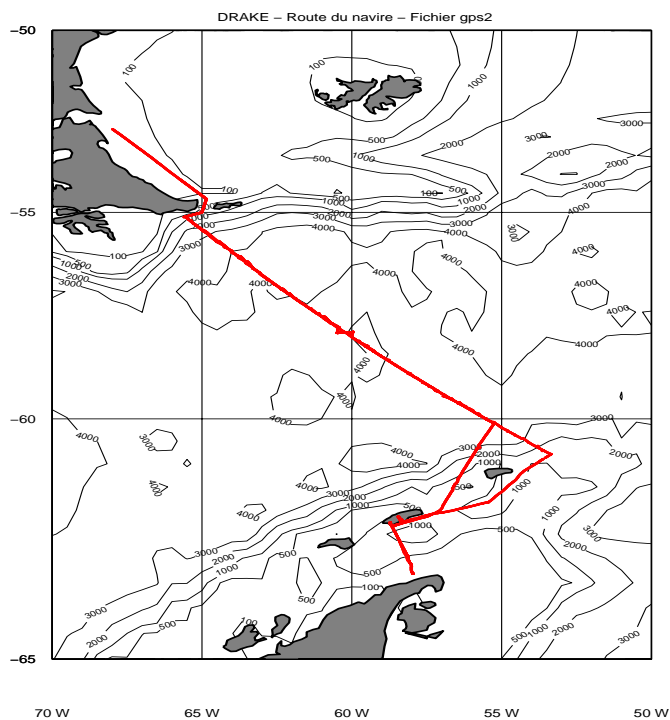


FIG. 4.1: Positions des ensembles ADCP : Fichier gps2 corrigé

Chapitre 5

Edition des données ADCP

Une première vérification est celle de la vitesse du son, qui peut être figée à une vitesse constante, ou varier suivant la température du transducteur (la salinité a une valeur constante dans le fichier de configuration). En effet une erreur sur la vitesse du son entraîne une erreur sur l'amplitude du courant mesuré par l'ADCP. Rappelons que la constante de proportionnalité entre le décalage Doppler et la vitesse de déplacement de la masse d'eau implique la vitesse du son au niveau du transducteur seulement.

On s'assure tout d'abord que la température du transducteur est correcte, en la comparant à la température du thermosalinographe, et que la vitesse du son utilisée par l'ADCP est également correcte. L'écart de 1 degré C. observé ici entraînerait une différence de 0.2% sur la vitesse du son. Un programme Matlab permet de tracer les deux températures ainsi que la vitesse du son.

La figure montre les tracés de la température du transducteur (rouge), du thermosalinographe (vert) et de la vitesse du son utilisée par l'ADCP (bleu), recalculée avec une salinité de 35 PSU, une immersion de 11m et la température du transducteur (cyan), et recalculée avec les mêmes salinité et immersion, mais une température diminuée de 1 (magenta). L'immersion de 11m est la position réelle du transducteur. On a également porté sur la figure 3 points (magenta) correspondant aux vitesses du son calculées d'après les mesures CTD (stations 1 : 1483.3 m/s, 50 : 1458.1 m/s et 107 : 1485.9 m/s).

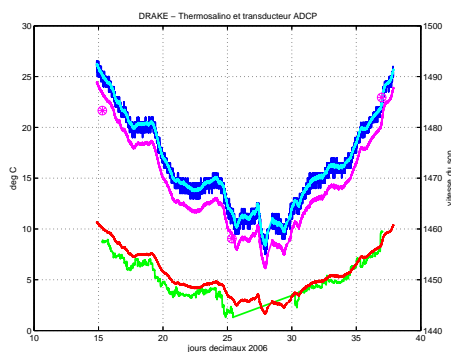


FIG. 5.1: Température (transducteur ADCP rouge - thermosalinographe vert) et vitesse du son

La calibration par "water tracking" effectuée dans la suite du traitement confirmera que la fenêtre était bien remplie d'eau salée, mais que la température donnée par le transducteur est sans doute trop élevée. En effet, si la vitesse du son est trop faible, la vitesse ADCP est aussi trop faible, et la méthode du "water tracking" donnera un facteur d'amplitude $A > 1$. La calibration permet de corriger les erreurs sur l'amplitude.

On procède ensuite à l'édition des profils ADCP. En ce qui concerne les données des profils, le premier stade est le passage de programmes de calculs de statistiques, qui permettent de déterminer les seuils d'erreurs maxima pour plusieurs paramètres : vitesse (composantes u et v), amplitude, erreur verticale. Ces calculs se font sous Matlab.

Les valeurs utilisées pour la campagne DRAKE sont les suivantes (une seule édition pour les trois bases) :

ev_threshld : 150

w_var_threshld : 12837

d2w_threshld : 206

d2uv_threshld : 390

La visualisation des profils se fait aussi au moyen de Matlab. Les différentes erreurs répertoriées sont marquées par des signes de couleurs et de formes différentes. L'opérateur peut accepter les suppressions de données automatiques, en supprimer plus ou bien décider d'annuler le marquage de valeurs erronées. Cette dernière possibilité est très utile dans le cas de couches de diffusion (plancton) qui font apparaitre des pics d'amplitude du signal, confondus dans le traitement automatique avec les pics d'amplitude produits par la réflexion sur le fond. Toutefois, la forme et la hauteur de ces pics ainsi que la position géographique permettent de lever l'ambiguïté. La figure ci-dessous montre l'aspect des profils d'amplitude pour DRAKE, à la sortie du détroit de Le Maire, afin de visualiser l'aspect du fond.

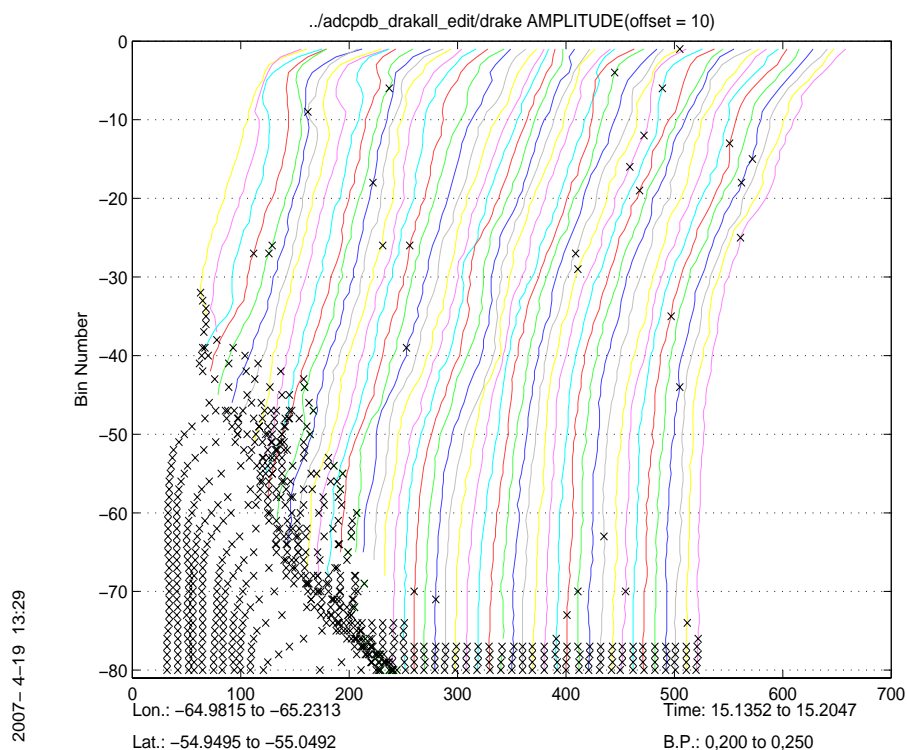


FIG. 5.2: Profils d'amplitude du signal ADCP

Les pics de la gauche de la figure (décroissance entre les bins 45 et 80) correspondent au fond. Les croix marquent tous les bins rejetés. D'une façon générale, les profils sont mauvais en route libre, et meilleurs pendant les stations CTD.

Les bins marqués comme mauvais, les profils supprimés, les bins correspondant au fond sont enregistrés respectivement dans 3 fichiers ASCII, badbin, badprf, bottom. Ces fichiers seront consultés ensuite pour la validation de la base de données.

Chapitre 6

Calibration des données

Le principe général de la calibration est de passer de vitesses de courant relatives au bateau à des vitesses absolues dans un référentiel terrestre. Pour obtenir ces vitesses de courant absolues, il faut soustraire aux mesures relatives la vitesse du navire par rapport au référentiel terrestre, cette vitesse étant donnée par la navigation.

Les erreurs proviennent de trois sources :

positions : Valeurs erronées de positions GPS.

cap : Variation du gyrocompas (ou du cap donné par toute autre source).

transducteur : Défaut éventuel d’alignement du transducteur ADCP par rapport à l’axe du bateau.

Ce dernier est déterminé par deux méthodes : "bottom tracking" et/ou "water tracking".

6.1 Qualité des positions

Après les premières vérifications effectuées lors du traitement des fichiers bruts de navigation, le contrôle final de la qualité des points GPS nécessite l’extraction à partir de la base ADCP validée (chapitre édition) d’un fichier des vitesses du navire (mesures ADCP) par rapport à une couche de référence (programme `adcpsect`, fichier créé nommé `.nav`), qui est pour chaque ensemble la moyenne des vitesses entre les bins 3 et 10 (soit 12m et 40m). Historiquement, la méthode de la couche de référence a été mise au point pour valider les positions satellite de type "Transit", peu fréquents et moins précis que le GPS, même dégradé. Cependant, dans de nombreux cas, dont celui de la campagne DRAKE, cette méthode permet de sortir des difficultés de traitement.

La couche de référence idéale serait une couche de vitesse constante, alors que les vitesses dans l’océan réel varient. Il faut donc choisir une couche de référence suffisamment fine pour que les variations soient le plus lentes possible, mais assez épaisse pour contenir le maximum de données (le tracé du "PGOOD", ou pourcentage de bonnes données donne une indication utile). On rappelle que la vitesse du navire par rapport à la couche de référence est l’inverse de la vitesse ADCP par rapport à la couche de référence. Les données de position permettent d’établir la vitesse du navire dans le référentiel terrestre. Si on admet que l’édition des profils ADCP est correcte (suppression des bins et des profils de mauvaise qualité, détection du fond, etc...), lorsqu’on soustrait de la vitesse du navire par rapport à la couche de référence la vitesse absolue du navire (programme `refabs`, fichier créé `.ref`), on obtient alors les vitesses absolues de la couche de référence, et les anomalies proviennent d’erreurs dans le fichier des positions (`.gps`).

Pour un contrôle graphique au moyen d'un programme Matlab (callrefp.m), les données sont lissées (programme smoothr utilisant une fenêtre de Blackman, sortie .sm). Un exemple est donné ci-dessous :

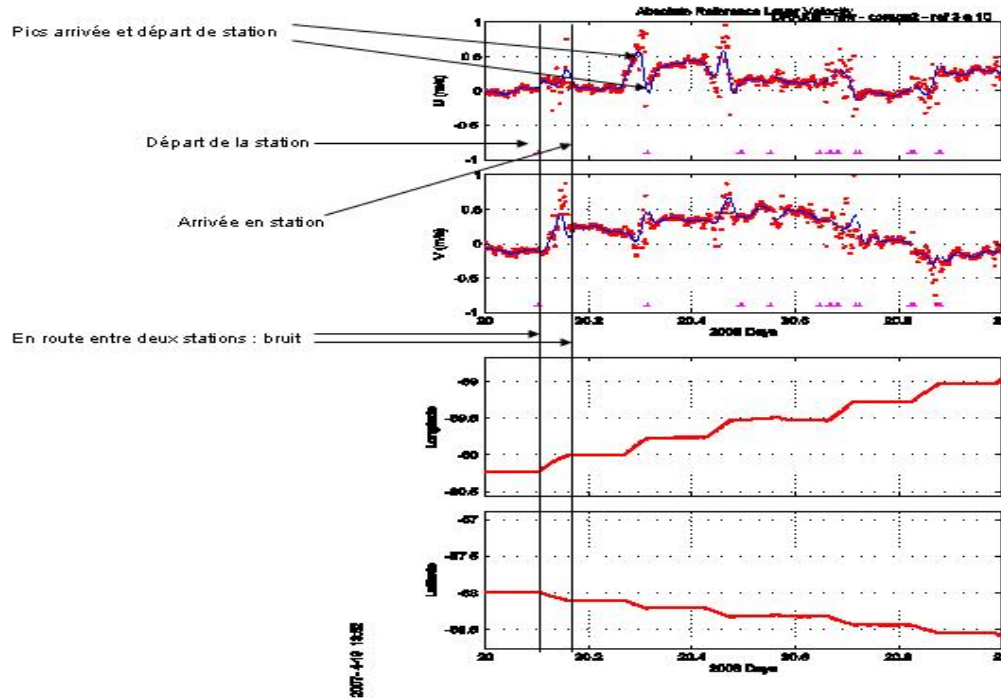


FIG. 6.1: Exemple de tracé de contrôle de la couche de référence

Dans le cas d'un tracé aussi catastrophique, nous nous sommes posé, ainsi que Gérard Eldin, plusieurs questions. La vitesse du son étant vérifiée et la correction de distance effectuée (corags, voir Fig. 3.1), le bruit en route et l'erreur d'angle seraient dûs à de mauvaises données de cap, et les pics aux arrivées-départ de stations à des erreurs d'horloge (plusieurs dizaines de secondes).

Avant d'éditer le fichier positions pour mettre en commentaire les points aberrants, et recommencer les opérations de contrôle avec callrefp, nous avons vérifié les données de cap.

La première vérification à faire était que le cap acquis dans la base ADCP était bien le même que le cap de la centrale inertielle. Pour cela, on a extrait de la base PODAS (Polarstern Data Acquisition System), tous les caps disponibles, et on les a comparés au cap enregistré dans les ensembles ADCP (fichier .hdg), au moyen des outils d'extraction d'angles similaires à ceux de la routine "corryge" de G. Eldin. Le seul cas comparable au hdg acquis par l'ADCP est le cap "gyro MINS", mais une différence moyenne de 0.1 reste inexpliquée. De plus les positions MINS "vraies" extraites de PODAS donnent le même résultat que les positions .gps2 avec le contrôle callrefp ci-dessus, ainsi d'ailleurs que les positions des fichiers N1R (trames NMEA non transformées).

Donc nous avons entrepris de supprimer toutes les valeurs aberrantes dans le fichier positions, avec contrôle par le moyen du programme callrefp, puis des tests de "water tracking" fréquents pour vérifier que les choses s'amélioreraient...

De plus, étant donné la mauvaise qualité évidente des données pendant les transits en route, nous avons pris la décision de privilégier les données acquises en station, en extrayant un profil moyen pour chaque station (moyenne de tous les profils validés effectués pendant la durée de la station).

6.2 Erreur sur le cap

On a ensuite tenté dévaluer les erreurs sur ce "cap gyro MINS", en créant un fichier de corrections angulaires, qui servirait à la fin de la phase de calibration à effectuer une rotation des valeurs de la base de données. Pendant la phase de tests de calibration, on effectue cette rotation (programme `rotnav`) sur le fichier des vitesses du navire par rapport à la couche de référence (fichier `.nav`), pour ne pas modifier la base elle-même.

Dans un premier temps, on a calculé les différences d'angles entre le cap enregistré dans les ensembles ADCP (fichier `.hdg`) et les routes sur le fond données par différentes sources de navigation du fichier PODAS : "course" MINS PODAS et "course" NACOS PODAS. Ces valeurs sont aberrantes.

Dans le cas d'un gyrocompas classique, nous avons déjà utilisé une formule de correction en fonction du cap, de la vitesse du navire et du cosinus de la latitude :

$$\text{Angle de correction} = V_{\text{navire}} * \text{Cos}(\text{Cap}) / V_{\text{angulaireterre}} * \text{Cos}(\text{Latitude}) \quad (6.1)$$

L'inconvénient de cette formule est qu'elle présuppose que le gyrocompas n'est pas du tout compensé, ce qui n'est sans doute pas le cas. Les résultats donneraient donc une estimation maximum de l'erreur sur le gyrocompas. D'autre part, il faut avoir constamment la vitesse du bateau, ce qui n'est pas le cas ici.

On a aussi souvent utilisé dans d'autres campagnes la procédure de correction Proc-gyro de E. Firing, qui crée un modèle des erreurs de gyrocompas, plus évolué que la formule brute ci-dessus, utilisant deux fichiers créés dans le logiciel CODAS, soit le fichier de navigation lissé (suffixe `.sm`) et le fichier des caps enregistrés dans la base ADCP (suffixe `.hdg`). Le mode d'emploi de cette procédure est donné en Annexe IV. Le fichier d'angles créé par cette procédure est lissé et inversé pour être utilisé dans les programmes de rotation de CODAS, au moyen d'une adaptation du logiciel Corryge de G. Eldin.

Mais ici, le cap provenant d'une centrale inertielle (gyro laser, accéléromètres, etc...), ces procédures ne semblent plus applicables. Un essai a été fait avec le modèle Proc_gyro de E. Firing, mais les valeurs d'angles estimées, inférieures à 0.1, sont tout à fait irréalistes compte tenu de l'ordre de grandeur apparent des corrections à effectuer.

Par contre, comme dans le cas des campagnes TIP2000 et INTERPOLE 99, le grand nombre de stations CTD effectuées permet une autre approche pour évaluer les corrections angulaires : dans la méthode du "water tracking" du logiciel CODAS, plus détaillée au paragraphe 5.3 suivant, le nombre de points extraits par le programme de détection des accélérations/décélérations (`timslip`) est théoriquement égal, au minimum, au double du nombre des stations et/ou des ralentissements et/ou des changements de direction. Ce programme `timslip` permet ainsi d'évaluer l'erreur sur le cap, dans le cas où l'on dispose de nombreux points de calibration (campagnes CTD avec nombreuses stations, et peu de transits à cap constant), en évaluant l'influence de la latitude sur l'angle entre l'axe du transducteur et l'axe longitudinal de la coque "misalignment", déterminé par le programme `timslip` ("phase"). Rappelons que cet angle devrait idéalement être une constante.

On trace la phase extraite au moyen du programme "timslip" décrit ci-dessus en fonction de la latitude, et on effectue un fit polynomial sur le nuage de points. On privilégie la variation en fonction de la latitude et non en fonction de la vitesse du navire (voir formule 5.1), car l'expérience a montré que c'est le facteur le plus important. Si l'erreur sur la mesure du cap était nulle, la phase resterait constante, quelle que soit la latitude. Dans le cas contraire, le polynôme obtenu permet d'évaluer la variation du gyrocompas en fonction de la latitude, en calculant un fichier de différences d'angles variant avec la latitude, à partir du fichier de navigation. Ce fichier d'angles est construit en prenant comme constante du polynôme la différence entre la constante calculée par le fit et la phase constante (angle du transducteur) obtenue au moyen de la calibration par water-tracking (adcpal). Le terme du premier degré est celui du polynôme.

Les figures suivantes montre le résultat de ce fit du premier degré pour les bases DRAKE 1 et DRAKE 2, pour lesquelles on disposait respectivement de 95 et 73 points de calibration. On constate que, après la rotation, la pente de la droite passant au mieux dans le nuage de points est quasi-nulle, que la phase est également quasiment annulée, et que l'écart-type est fortement diminué.

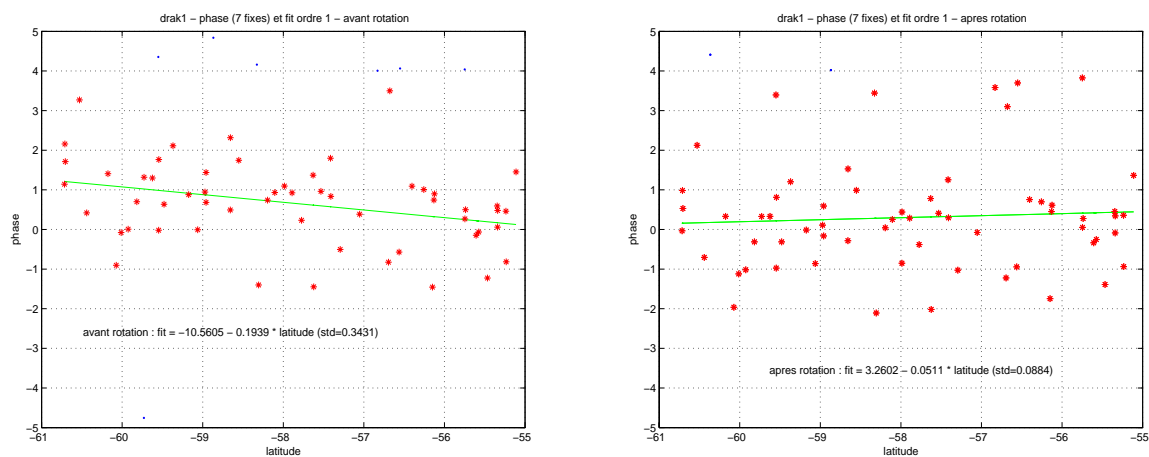


FIG. 6.2: DRAKE 1 : Tracé de la phase en fonction de la latitude, avant et après calibration

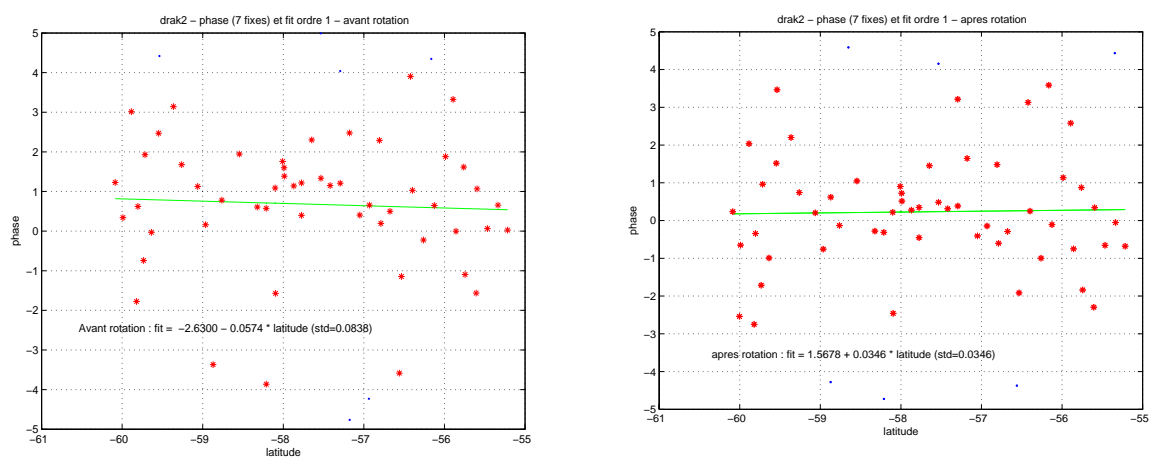


FIG. 6.3: DRAKE 2 : Tracé de la phase en fonction de la latitude, avant et après calibration

Ces fichiers d'angles de correction ont donc été utilisés pour effectuer une rotation des données de navigation. C'est sur ces données corrigées qu'ont été faites la vérification et la correction de l'alignement du transducteur (méthode du water-tracking).

Pour la partie Sud de la campagne (DRAKE 3), on ne disposait pas d'assez de points issus du programme timslip pour faire la même opération (29 points). On a utilisé un outil du logiciel CODAS, qui permet d'évaluer la variation en fonction du temps de la phase et de l'amplitude calculées par timslip.

6.3 Alignement du transducteur

On peut utiliser deux méthodes : comparer le déplacement du bateau par rapport au fond donné par la navigation (positions GPS) au déplacement par rapport au fond déduit de la détection du fond par l'ADCP ("bottom tracking"), ou bien comparer l'accélération par rapport à l'eau mesurée au moyen de l'ADCP, avec l'accélération calculée au moyen des points de positions ("water-tracking"). Ces deux méthodes donnent généralement des résultats identiques (à 0.5 % près) pour la détermination de l'orientation du transducteur selon l'axe longitudinal de la coque. Dans les deux cas, il faut également examiner la qualité des positions satellites (validation du fichier navigation) et des données de cap.

Les enregistrements de "bottom-tracking" n'ont pas été acquis.

Pour la calibration par "water-tracking" que nous avons utilisée pour corriger les défauts d'alignement du transducteur, on utilise le fichier .nav extrait de la base ADCP éditée et sur lequel on a effectué la rotation des angles de correction du cap.

On détecte ensuite les accélérations du navire à partir des vitesses ADCP, puis on les compare aux accélérations issues des données GPS, en faisant "glisser" les deux fichiers (vitesses de la couche de référence et vitesses GPS) de manière à faire correspondre les temps le mieux possible (programme timslip). Les accélérations se produisent lors des arrivées et départs de station CTD ou carottage, lors des changements de direction, et bien sûr lors des ralentissements et accélérations du navire sur sa route directe. Ce programme effectue également un filtrage des valeurs erronées. On a choisi une fenêtre de glissement de "7 fixes", soit 14 minutes (pour des ensembles de 2 mn). La longueur de la fenêtre doit être supérieure à la durée de l'accélération ou de la décélération. On détermine ainsi le décalage de temps entre l'horloge du PC d'acquisition des données ADCP, soit le temps de la base ADCP, et le temps GPS, ainsi que l'angle du transducteur par rapport à l'axe du navire (phase), et l'influence de cet angle sur le module de la vitesse (amplitude).

Le critère de choix de la fenêtre était de garder le plus grand nombre de points de calibration, avec des valeurs raisonnables d'écart-type. On admet 0.6 à 1.2 degrés pour la phase et 1 à 1.5 % pour l'amplitude. La différence de temps "nav-pc" doit rester de l'ordre de quelques secondes.

Un programme Matlab, adcpal, fait les moyennes et statistiques sur les valeurs données par le programme timslip. La valeur moyenne obtenue en angle et en amplitude permet de corriger le défaut d'alignement du transducteur (qui est normalement une constante).

Le tableau ci-dessous donne les valeurs de tolérance maximum pour les paramètres de la calibration prises en compte par le programme adcpal (on a utilisé les mêmes paramètres pour les 3 bases DRAKE1, 2 et 3) :

```
delta-u min = -100.00, max = 100.00 ;
delta-v min = -100.00, max = 100.00
clip-amp = 0.1, clip-ph = 3.0
clip-dt = 60, clip-var = 0.100
```

La figure ci-dessous illustre un exemple du résultat du programme adcpcal.

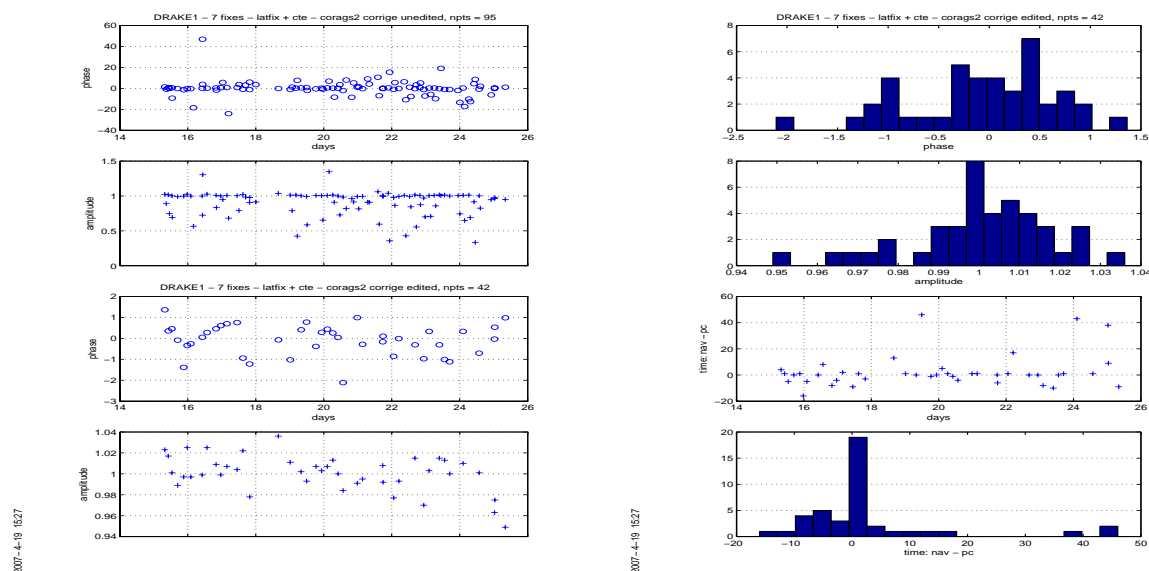


FIG. 6.4: Détermination de la phase, de l'amplitude et d'un décalage d'horloge

6.4 Résultats de la calibration

Le résultat de l'édition et de la calibration est le suivant :

Le décalage d'horloge est ramené à 2.5 secondes en moyenne (sur les deux radiales CTD DRAKE1 et DRAKE 2) avec toutefois un écart-type élevé (12.6), ce qui confirme l'existence d'un problème d'horloge lors de l'acquisition (décalage entre l'horloge de la centrale MINS et celle acquise par l'ADCP, qui proviendrait du GPS Trimble 1). Pour la phase, l'écart-type admissible est de l'ordre de 1.5 degré, et pour l'amplitude, il est de 2%. Les valeurs de la calibration sont donc raisonnables (0.75 degré et 2.1% en moyenne pour les deux radiales). Les valeurs de l'amplitude brute, très peu supérieures à 1, confirment que la fenêtre de logement du transducteur est bien remplie d'eau salée, mais que la température réelle est sans doute inférieure à celle mesurée par le transducteur (voir Fig 4.1).

Le tableau récapitule les dates de début et de fin, le nombre de points de calibration par water-tracking (qui permettent d'obtenir la correction de l'alignement du transducteur), ainsi que les valeurs de la phase, de l'amplitude et de l'erreur d'horloge obtenues par cette calibration. La dernière colonne indique le nombre de profils ADCP obtenus. Dans la colonne Nb de points, le premier chiffre est le nombre de points retenu, le deuxième chiffre le nombre de points de départ. Dans les colonnes Amplitude, Phase et Delta T, le premier nombre est la valeur avant la calibration, le deuxième nombre est la valeur après la calibration, les nombres de la deuxième ligne sont les écarts-type associés.

Campagne	Données gyrocompas		Water-tracking				Profils validés
	Début	Fin	Nb points	Amplitude	Phase	Delta T	
DRAKE 1	16 01 06	26 01 06	42/95	1.0145/1.0004	0.7809/-0.074	4.27/2.52	5679
	06 42 44	09 26 50		0.0223/0.0174	0.9114/0.7355	13.99/12.66	
DRAKE 2	31 01 06	06 02 06	37/73	1.0128/1.0000	0.9764/0.0004	3.61/2.41	4787
	00 09 55	23 51 12		0.0260/0.0249	0.9493/0.8150	14.24/12.54	
DRAKE 3	26 01 06	30 01 06	26/42	1.0368/0.9960	0.7530/-0.0486	1.63/2.30	1103
	09 36 48	23 59 54		0.0226/0.0220	1.4458/1.3086	-2/5.02	

Au final, après obtention de ces données de calibration (soit un fichier d'angles pour la correction du cap, puis un angle(phase) et une amplitude constants pour corriger l'alignement du transducteur), on effectue la rotation des données de la base (programme rotate). Il faut ensuite extraire le fichier .nav (adcpsect) à partir des données ayant subi la rotation, puis calculer la vitesse absolue de la couche de référence (refabs, fichier .ref), et ensuite lisser ce dernier fichier (smoothr, fichier .sm).

Le logiciel CODAS permet des tests de qualité globale, sur l'ensemble des profils. Ils permettent surtout de visualiser la différence entre les données en route et en station.

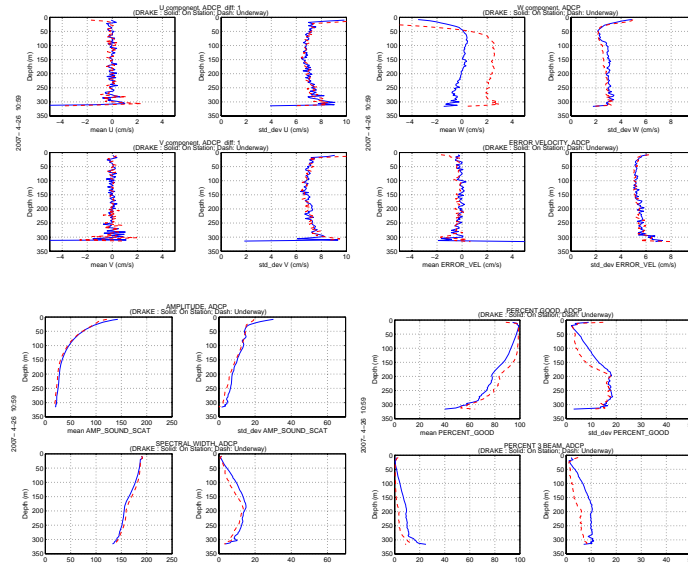


FIG. 6.5: Tests de qualité globale

6.5 Mise en forme définitive de la base de données

Tout d'abord a été créée une base des données éditées, donc avec des profils "propres", mais avec des données de navigation brutes. Puis nous avons effectué la calibration.

- Données juste éditées (suppression de tous les points ADCP aberrants).
- Rotation par un fichier d'angles calculés par le fit en latitude sur les points issus du "water tracking", puis par une constante résiduelle (correction de l'alignement du transducteur) pour DRAKE 1 et DRAKE 2, et par un angle et une amplitude variant en fonction du temps pour DRAKE 3.

Le dernier stade de l'élaboration de la base est l'intégration des données de navigation calibrées (fichier .sm) dans la base (programme putnav). Il est alors possible de construire des grilles de différents formats, suivant un maillage spatial ou temporel, qui serviront de base pour la conversion des données en ASCII sous forme de fichiers formattables pour être lus sous Matlab, et pour les tracés.

La construction des grilles spatiales se fait en deux temps, d'abord avec le programme lgrid, qui demande la définition de la grille en latitude et longitude, puis le résultat (un tableau de temps) sert de contrôle pour le programme adcpsect, soit pour créer les fichiers de contours .con, associés à des fichiers de statistiques .sta, soit des fichiers de vecteurs à différents niveaux : .vec. Ces derniers peuvent être tracés au moyen du programme vector de CODAS. Les figures de la page 21 montrent les tracés des vecteurs de courant pour les 2 radiales, sur les données après calibration complète. On utilise un programme de tracés de contours MATLAB pour les sections, qui utilise les fichiers .con.

La construction de grilles temporelles se fait également en deux temps, d'abord avec le programme timegrid, dans lequel on entre un pas de temps : régulier si on veut des moyennes temporelles, ou variable, par exemple le début et la fin des stations CTD, ce qui permettra d'obtenir des profils moyennés sur la durée de chaque station.

Si l'on utilise une grille avec la durée des ensembles comme pas de temps, et un programme `adcpssect` ne faisant aucune moyenne sur les données, on obtient tous les ensembles (profils) directement en ASCII (la base CODAS est en binaire). Un formatage postérieur permet de présenter les profils sous forme de matrices facilement exploitables sous Matlab.

6.6 Evaluation de l'erreur

Erreur sur la vitesse relative (erreurs sur l'amplitude)

La première source d'erreur est, comme nous l'avons vu plus haut, l'erreur sur la vitesse du son. Nous avons vu que l'erreur sur la salinité dans les fichiers de configuration, donc sur la vitesse du son, entraînait une erreur de 2 % au maximum (eau salée 35 PSU, eau douce 0 PSU) sur la vitesse relative. Par exemple, pour une vitesse de bateau de 10 noeuds, soit environ 5 m/s, l'erreur est alors de 10 cm/s pour la composante longitudinale (dans l'axe du bateau). L'incertitude sur la température du transducteur, entraîne ici une erreur de 0.2 %, qu'on considérera comme négligeable.

Les bulles peuvent aussi réduire la vitesse du son. Nous n'avons pas trop d'informations sur l'influence de la forme de la coque du POLARSTERN (brise glace, coque ronde), mais vu les fichiers de configuration utilisés "Narrow Band", bins de 4m), nous pensons que les bulles doivent être bien présentes. D'autre part, les eaux polaires sont souvent pauvres en particules, ce qui explique aussi cette configuration particulière (meilleure pénétration).

Erreur sur la vitesse absolue (erreurs sur la phase)

En supposant que la vitesse du bateau est bien connue (précision des points GPS), les erreurs sur la composante transversale proviennent des erreurs sur la mesure du cap et sur le positionnement (horizontal et vertical) du transducteur.

La méthode de la couche de référence permet de corriger les erreurs éventuelles sur les points GPS. Ces erreurs étaient fréquentes avec l'ancien système Transit, très rares avec le système GPS, même dégradé. La méthode est toutefois conservée pour ses outils de visualisation et de contrôle. Elle a été fort utile dans le cas de DRAKE. La distance entre l'antenne et le transducteur ADCP peut également influencer. Cette dernière erreur a été corrigée au moyen de la procédure `corags`. Mais ces erreurs sont très faibles en regard des suivantes.

L'erreur sur le positionnement horizontal (angle par rapport à l'axe longitudinal du navire) peut être évaluée et corrigée au moyen de la méthode du water-tracking (Pollard et Read), à condition d'avoir un nombre suffisant de points de calibration (idéalement, au moins 64, pour avoir une précision de 0.2° sur la calibration).

L'erreur sur le positionnement vertical (roulis, tangage, assiette) est normalement corrigée en configuration Janus (les axes du transducteur sont en X par rapport à l'axe du navire).

L'erreur sur le cap a été évaluée approximativement au moyen du fit en latitude. Une erreur de 0.6° entraîne une erreur de 1 % de la vitesse du bateau sur la composante transverse (soit 5 cm/s pour une vitesse de 10 noeuds).

En résumé, compte tenu du mauvais aspect au départ des données de navigation et des difficultés d'appréciation des erreurs sur le cap issu de la centrale inertielle, une incertitude de +/- 10cm/s sur le module de la vitesse paraît une estimation raisonnable. Toutefois les directions, à fortiori dans ces régions de forts courants, sont réalistes.

Comme il l'est rappelé dans la plupart des publications techniques sur les mesures ADCP de coque, ces mesures ne semblent pas utilisables pour le calcul de transports.

6.7 Obtention des profils moyennés en station

Un programme du logiciel CODAS, `adcpsect`, permet d'obtenir les statistiques sur les profils moyennés dans une gamme de temps (voir la construction des grilles temporelles). On a donc construit un fichier de gammes de temps, avec les dates de début et de fin de chaque station (CTD ou autres prélèvements), pour les radiales aller et retour (DRAKE1 et 2), et pour les stations de la partie Sud et du détroit de Bransfield (DRAKE 3). Un exemple de ce fichier est donné ci-dessous, pour la radiale DRAKE 1 :

```
2006/01/16 07 :13 :00 to 2006/01/16 07 :40 :00
2006/01/16 09 :12 :00 to 2006/01/16 10 :04 :00
...
2006/01/26 06 :59 :00 to 2006/01/26 07 :58 :00
```

En sortie, après reformatage, les profils moyennés en station se présentent sous la forme d'un en-tête pour chaque profil, avec les dates de début et de fin de chaque station, la position (latitude et longitude décimales), le nombre de profils moyennés pendant la durée de la station, et dans le corps du profil, la profondeur, les composantes zonale et méridienne de la vitesse du courant, avec pour chacune l'écart-type sur le calcul de la moyenne, le minimum et le maximum. Ce fichier peut être lu sous Matlab.

6.8 Comparaison avec le LADCP

Les profils LADCP en station, traités et validés au moyen du logiciel de Fischer et Visbeck, se présentent sous la forme d'une matrice Matlab par profil. Il a donc été aisé de faire une comparaison profil par profil. Les données sont en bon accord, sauf près de la surface dans certains cas (les données LADCP sont plus dispersées, et la meilleure résolution verticale du SADCP est plus visible en surface).

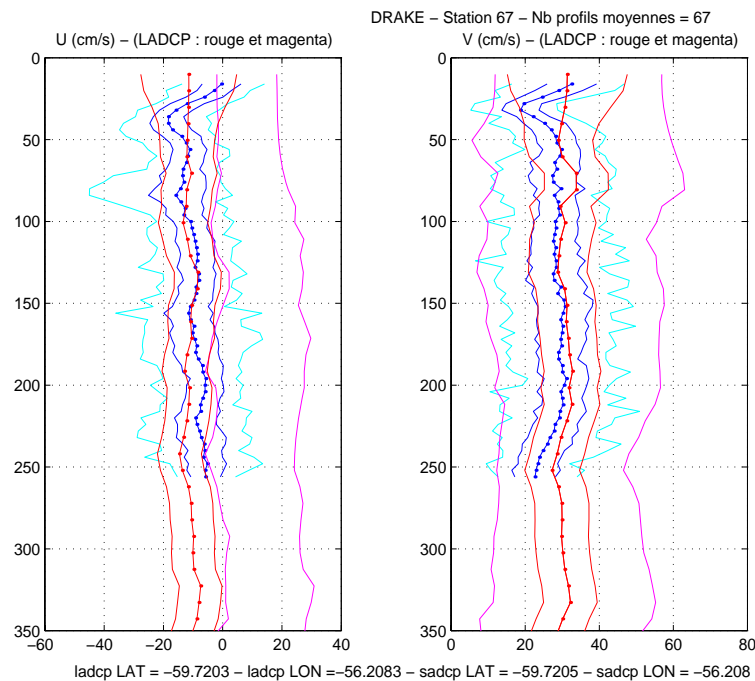
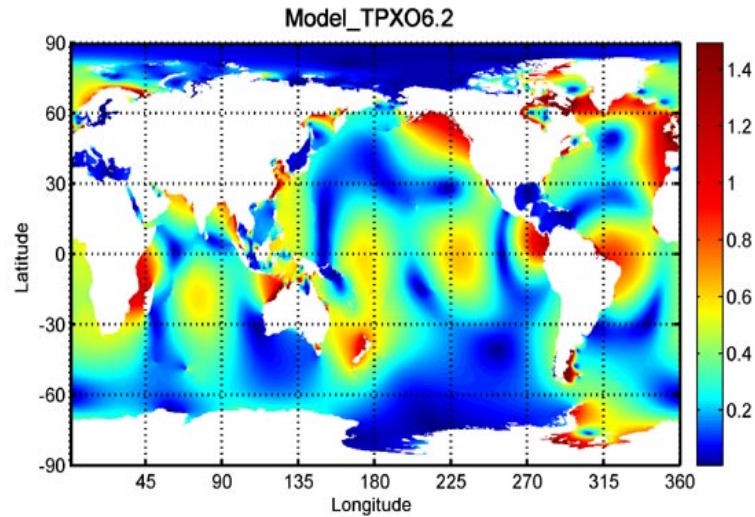


FIG. 6.6: SADCP moyen : points bleus, LADCP moyen : points rouges, std SADCP : traits bleus, std LADCP : traits rouges, extrema SADCP : cyan, extrema LADCP : magenta

6.9 Evaluation de l'influence de la marée

Nous avons utilisé le modèle TPXO de l'Oregon State University (TOPEX/POseidon Global Inverse Solution), dans sa version TPXO6.2, modèle global de marées adapté aux traces de TOPEX/Poseidon et Jason. Le modèle nécessite le chargement du logiciel de Prédiction de Marées de l'OSU (OTPS, pour OSU Tidal Prediction Software). La résolution du modèle global est de 1/4. Les données assimilées proviennent surtout de l'altimétrie radar de T/P, entre 66° N et 66° S. Le modèle inclut aussi des "patches" pour plusieurs zones côtières, en particulier npour l'Antarctique CADA00.10 (Circum-Antarctic inverse model) et Ross_Inv_2002 (Ross Sea height-based inverse model). Le mode d'emploi est détaillé en Annexe V. La figure donne une idée des sorties du modèle.



Pour évaluer l'effet des marées par rapport aux mesures ADCP dans le passage de Drake, nous avons fourni au modèle deux fichiers de positions horaires (en route et en station) pour l'aller et le retour. La figure ci-dessous illustre cet effet, très faible en mer ouverte, mais qui se renforce à l'approche des terres et particulièrement dans le détroit de Bransfield.

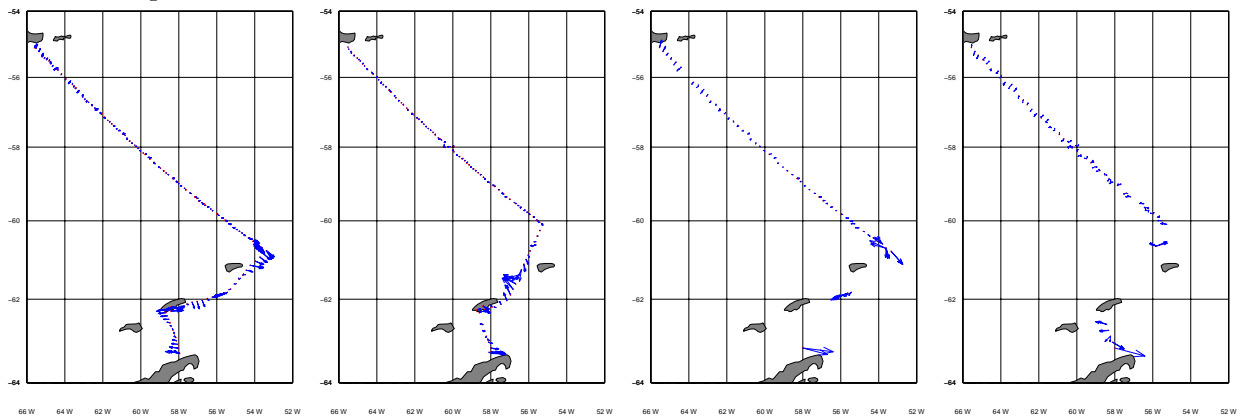


FIG. 6.7: Tests de l'influence de la marée en route (à gauche) et en station(à droite)

En station, la vitesse moyenne de courant de marée est de 3.3 cm/s, avec un maximum de 27.8 cm/s et un minimum de 0.04 cm/s, pour la descente. Pour la remontée, la moyenne est de 2.8 cm/s, avec des extrema de 23.2 et 0.13 cm/s.

En route, la vitesse moyenne est de 3.6 cm/s, avec un maximum de 24.2 cm/s et un minimum de 0.04 cm/s pour la descente. Pour la remontée, le moyenne est de 5.2 cm/s, avec des extrema de 52.1 et 0.13 cm/s.

6.10 Références

- Firing E.** Report from the WOCE/NOAA Workshop on ADCP measurements, held in Austin, Texas, March 1-2 1988, U.S. WOCE Planning Report No. 13, U.S. Planning Office for WOCE, College Station, TX.
- Joyce T. M.** On in situ "calibration" of shipboard ADCPs. *J. Atmos. Oceanic. Technol.*, 6, 169-172, 1989.
- Pollard R. and Read J.** , A method for calibrating shipmounted acoustic Doppler profilers, and the limitations of gyro compasses, *J. Atmos. Oceanic. Technol.*, 6, 859-865, 1989.
- RD Instruments** Acoustic Doppler Current Profilers Principles of Operation : A Practical Primer, 1989, RD Instruments, Businesspark Avenue, San Diego, CA 92131
- Fischer J., and Visbeck M.** , Deep velocity profiling with self-contained ADCPs, *J. Atmos. Oceanic Technol.*, 10(5), 764-773, 1993.
- Firing E. and Bahr F., Caldwell P., Ranada J., Zhu W.** Processing ADCP Data with the CODAS software System, Version 3.1, "unpublished manuscript", 1995, JIMAR, University of Hawaii, 1000 Pope Road, Honolulu, Hawaii 96822
- Eldin G.** Correction des erreurs gyro-compas par GPS System, version 1.2, "unpublished manuscript", août 2000
- Egbert G.D. and Erofeeva S.Y.** , Efficient inverse modeling of barotropic ocean tides, *J. Atmos. Oceanic Technol.*, 19(2), 183-204, 2002.

6.11 Remerciements

Nous remercions tout particulièrement l'ingénieur électronicien à bord du POLARSTERN, sans qui l'acquisition de ces données n'aurait pas pu se faire, Felix Stoehr, membre bilingue combien précieux de l'équipe scientifique, et Gérard Eldin, qui a toujours répondu à toutes les interrogations concernant gyro-compas et centrales inertielles et leurs problèmes de calibration.

Chapitre 7

Présentation rapide des données

Nous présentons les tracés des vecteurs de courant moyennés sur une grille spatiale de 1/20 ème de degré, pour la première couche, soit de 20 à 50 m, pour la radiale du Passage de DRAKE, dans les deux sens, données en route et en station mélangées. Les tracés présentent les données avec une correction du cap par le fit en latitude expliqué page 12, suivie d'une correction par une constante.

Sur ces données, on présente également les sections en fonction de la latitude pour les composantes U et V de la vitesse des profils moyennés en station, pour les deux radiales DRAKE 1 et DRAKE 2

DRAKE - Radiales 1 et 2

15 au 25 Janvier , 31 Janvier au 6 Fevrier 2006

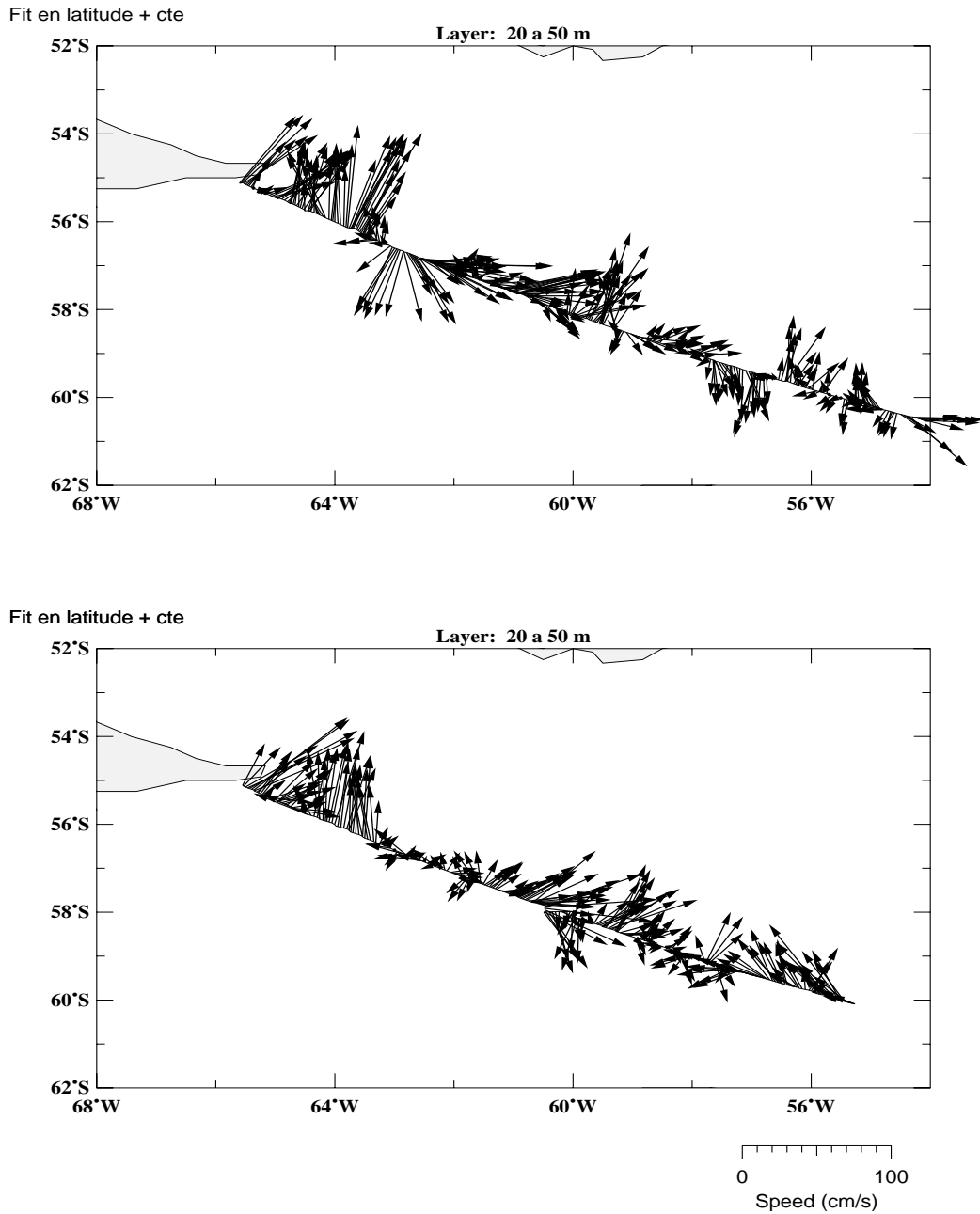


FIG. 7.1: Vecteurs de courant dans la couche 20 - 50 m pour les deux radiales DRAKE

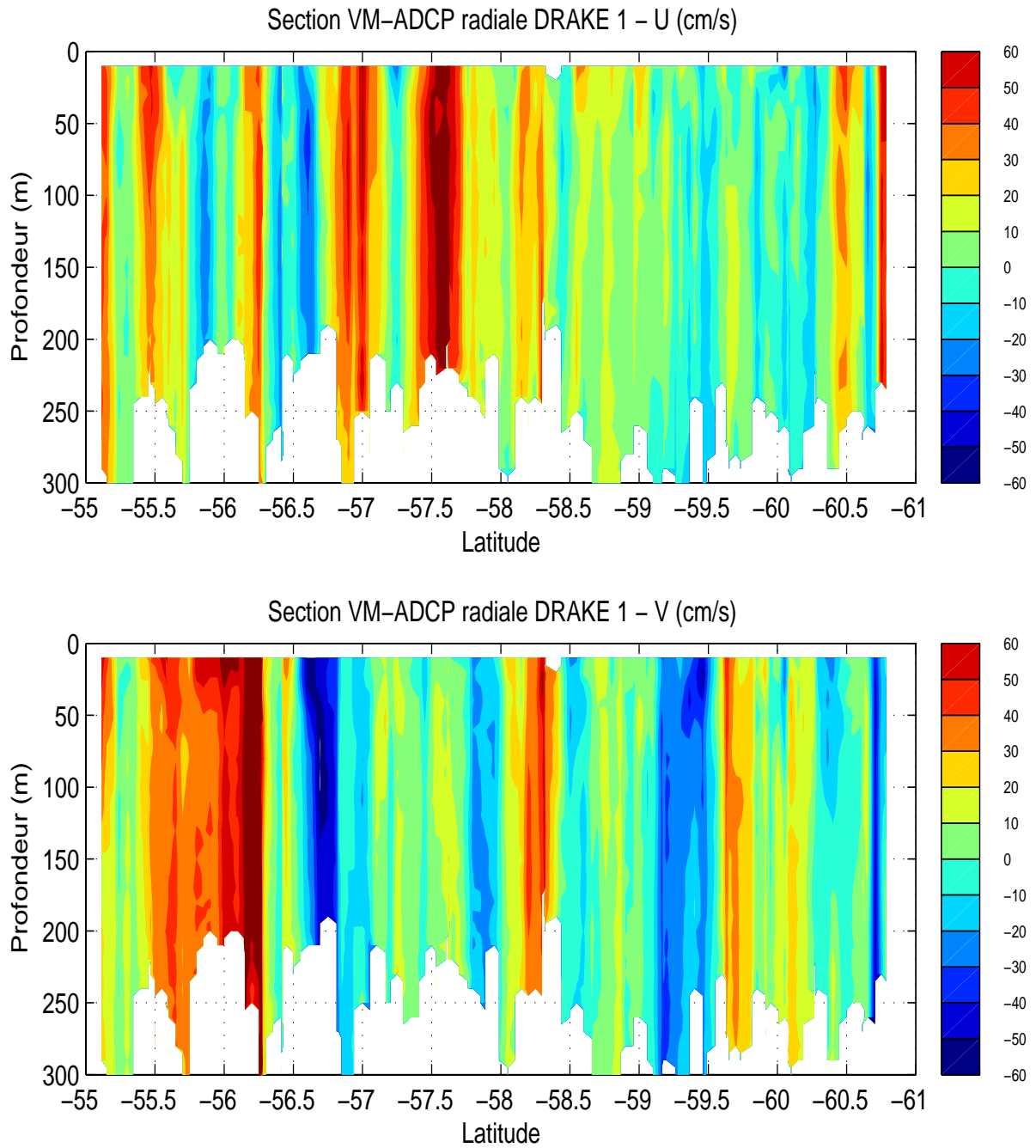


FIG. 7.2: Section en latitude pour la radiale aller DRAKE 1

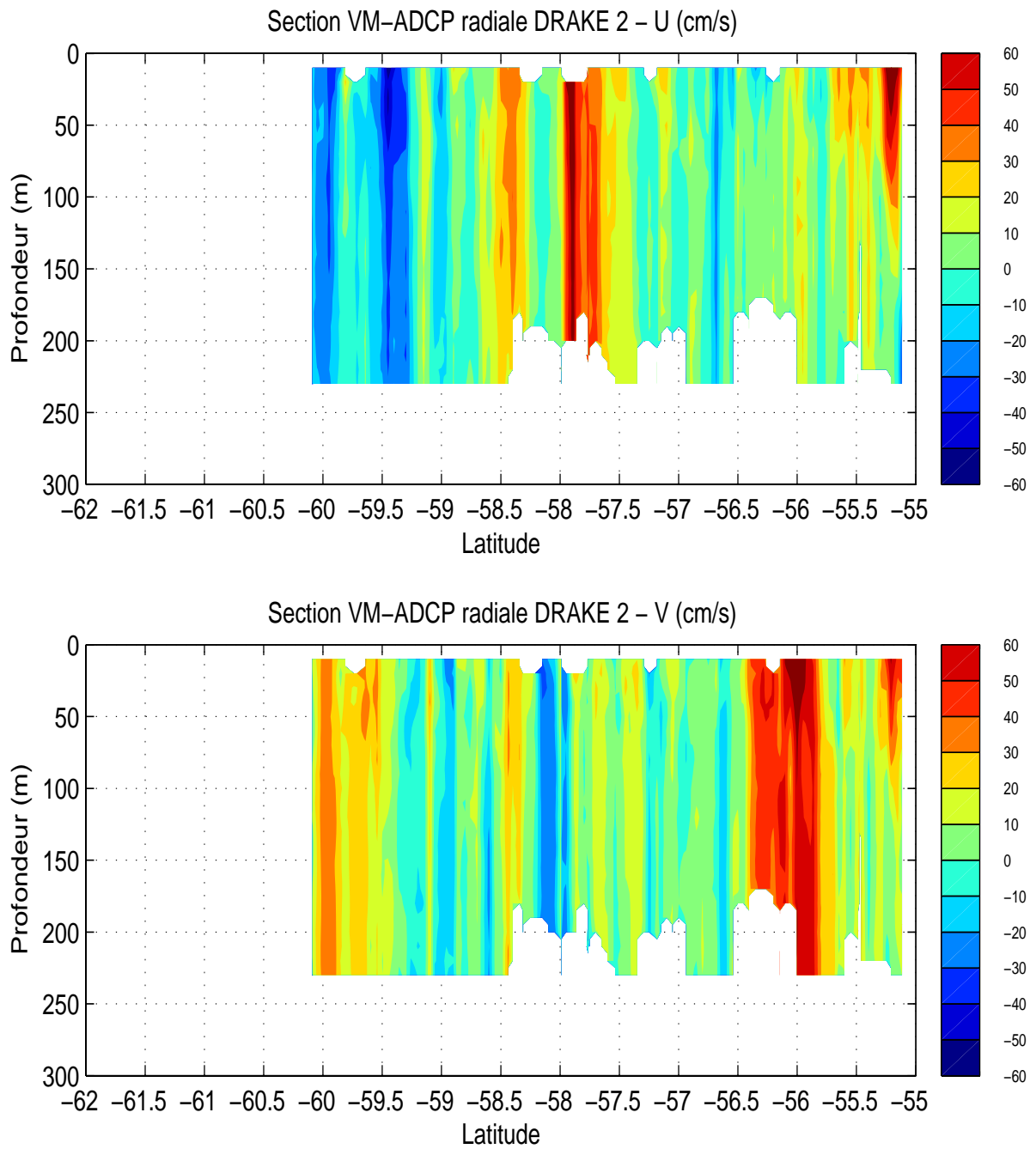


FIG. 7.3: Section en latitude pour la radiale retour DRAKE 2

ANNEXE I : SYSTEME DE NAVIGATION

MINS (Marine Inertial Navigation System)

Raytheon Marine GmbH (ehemals Anschütz)
 Zeyestraße16-24
 24106 Kiel
 P.O.Box 1166
 24106 Kiel

Genauigkeit nach Spezifikation (maximale Abweichung):

Heading 0,279°

Roll 0,057°

Stampf 0,057°

im Betrieb 1 mil Rad im Bogenmaß (bei 3 wird der Navigationsmode zugeschaltet)

Die Genauigkeit nach Prüfprotokoll liegt bei 0,025° und besser.

Die Ausgabe von Heading, Roll und Pitch erfolgt über Synchrosignal und digital.

Das MINS-Interface (Fa. WERUM) wandelt die Daten, die auf 3 Datenbussen mit unterschiedlicher update rate ankommen und speist sie mit 1 Hz in PODAS ein.

MINS liefert:

Broad Cast-Bus I

RS422A, 200 Hz update rate, Baudrate 307,2k

HDLC Protocol

- Heading, Roll, Pitch
- Heading-, Roll-, Pitch-Rates
- Status

Broad Cast-Bus II

RS422A, 5 Hz update rate, Baudrate 307,2k

HDLC Protocol

- Velocity Vx, Vy, Vz
- Speedlog velocity
- Position
- Status

Course Bus

RS422A

Es gibt 2 verschiedene Möglichkeiten der GPS-Empfänger, die die MINS stützen. Als zugelassene Version wird die MINS mit einem Raytheon Antennenempfänger Raystar 112 Product Code M93537 (12 Kanäle) betrieben. In wissenschaftlichen Fahrtgebieten wird die MINS durch einen Geodätischen Zweifrequenzempfänger Trimble MS750 gestützt.

Es kann immer nur eine MINS Daten an PODAS weiterleiten. Es handelt sich um getrennte Anlagen mit jeweils eigenem System (auch GPS jeweils einen Raystar- und einen Trimble-Empfänger).

ANNEXE II : CONFIGURATION DE L'ADCP

```

[Expert only options]
TurnedOffBeam=0
PashrImuFlagUseNormalInterpretation=TRUE
[ADCP Port Setup]
AdcpComPortName=COM1
AdcpComBaudRate=9600
AdcpComParity=NOPARITY
AdcpComStopBits=1
AdcpComDataBits=8
AdcpConfigFilename=C:\Program Files\RD Instruments\VmDas\OS38NBDEF.txt
[NMEA Port Setup]
NMEA NavComEnable=TRUE
NmeaNavComPortName=COM3
NmeaNavComBaudRate=9600
NmeaNavComParity=NOPARITY
NmeaNavComStopBits=1
NmeaNavComDataBits=8
NMEARPHComEnable=FALSE
NmeaRPHComPortName=None
NmeaRPHComBaudRate=4800
NmeaRPHComParity=NOPARITY
NmeaRPHComStopBits=1
NmeaRPHComDataBits=8
[Serial Port for Binary Ensemble Data Output]
BinaryEnsembleOutputComEnable=FALSE
BinaryEnsembleOutputComPortName=None
BinaryEnsembleOutputComBaudRate=9600
BinaryEnsembleOutputComParity=NOPARITY
BinaryEnsembleOutputComStopBits=1
BinaryEnsembleOutputComDataBits=8
BinaryEnsembleOutputDataType(0:none;1:enr;2:enx;3:sta;4:lta)=0
BinaryEnsembleOutputRefVelType(0:none;1:Bottom;2:Mean)=0
BinaryEnsembleOutputStartBin=1
BinaryEnsembleOutputEndBin=4
BinaryEnsembleOutputMeanStartBin=1
BinaryEnsembleOutputMeanEndBin=4
BinaryEnsembleOutputLeader(0:no;1:yes)=TRUE
BinaryEnsembleOutputBottomTrack(0:no;1:yes)=TRUE
BinaryEnsembleOutputNavigation(0:no;1:yes)=TRUE
BinaryEnsembleOutputVelocity(0:no;1:yes)=TRUE
BinaryEnsembleOutputIntensity(0:no;1:yes)=TRUE
BinaryEnsembleOutputCorrelation(0:no;1:yes)=TRUE
BinaryEnsembleOutputPercentGood(0:no;1:yes)=TRUE
BinaryEnsembleOutputStatus(0:no;1:yes)=TRUE
[Serial Port for ASCII Ensemble Data Output]
AsciiEnsembleOutputComEnable=FALSE
AsciiEnsembleOutputComPortName=None
AsciiEnsembleOutputComBaudRate=9600
AsciiEnsembleOutputComParity=NOPARITY
AsciiEnsembleOutputComStopBits=1
AsciiEnsembleOutputComDataBits=8
AsciiEnsembleOutputDataType(0:none;1:enr;2:enx;3:sta;4:lta)=0
AsciiEnsembleOutputRefVelType(0:none;1:Bottom;2:Mean)=0
AsciiEnsembleOutputStartBin=1
AsciiEnsembleOutputEndBin=4
AsciiEnsembleOutputStoreToDisk(0:no;1:yes)=FALSE
AsciiEnsembleOutMeanStartBin=1
AsciiEnsembleOutputMeanEndBin=4
AsciiEnsembleOutputLeader(0:no;1:yes)=TRUE

```

```

AsciiEnsembleOutputBottomTrack(0:no;1:yes)=TRUE
AsciiEnsembleOutputNavigation(0:no;1:yes)=TRUE
AsciiEnsembleOutputVelocity(0:no;1:yes)=TRUE
AsciiEnsembleOutputIntensity(0:no;1:yes)=TRUE
AsciiEnsembleOutputCorrelation(0:no,1:yes)=TRUE
AsciiEnsembleOutputPercentGood(0:no;1:yes)=TRUE
AsciiEnsembleOutputStatus(0:no;1:yes)=TRUE
[Serial Port for Speed Log Output]
SpeedLogComEnable=FALSE
Speed Log ComPortName=None
Speed Log ComBaudRate=9600
Speed Log ComParity=NOPARITY
Speed Log ComStopBits=1
Speed Log ComDataBits=8
SpeedLogDataSource=STA
SpeedLogWLSource=WP
SpeedLogWLStartBin=3
SpeedLogWLEndBin=5
[IP Port for Binary Ensemble Data Output]
BinaryEnsembleOutputNetEnable=FALSE
BinaryEnsembleOutputIPPortNumber=5433
[IP Port for ASCII Ensemble Data Output]
AsciiEnsembleOutputNetEnable=FALSE
AsciiEnsembleOutputIPPortNumber=5433
[IP Port for Speed Log Output]
SpeedLogNetEnable=FALSE
SpeedLogHostName/IPAddress=5434
[Fake Data Options]
AdcpFakeDataEnable=FALSE
AdcpFakeDataFilename=SimAdcp.enr
FakeDataTimeBetweenEnsembles=2
NMEAFakeDataEnable=FALSE
NMEAFakeDataFilename=SimNav.nmr
[File Name Components]
EnableDualRecordDir=FALSE
FileRecordPath=C:\ADCPData\ARK21_1b\
FileRecordBackupPath=D:\
DeploymentName=ARK21_1b
DeploymentNumber=1
MaximumFileSize=10
[Bottom Track Data Screening Options]
BTampScreenEnable=FALSE
BTCorScreenEnable=FALSE
BTerrScreenEnable=FALSE
BTvertScreenEnable=FALSE
BTFishScreenEnable=FALSE
BTPctGoodScreenEnable=TRUE
BTAmplitudeThreshold=30
BTCorrelationThreshold=220
BTErrVelThreshold=1000
BTVerticalVelThreshold=1000
BTFishThreshold=50
BTPctGoodThreshold=25
[Water Track Data Screening Options]
WTampScreenEnable=FALSE
WTCorScreenEnable=FALSE
WTerrScreenEnable=FALSE
WTvertScreenEnable=FALSE
WTFishScreenEnable=FALSE
WTPctGoodScreenEnable=TRUE
WTAmplitudeThreshold=30

```

```

WTCorrelationThreshold=180
WTErrorVelThreshold=1000
WTVerticalVelThreshold=1000
WTFishThreshold=50
WTPctGoodThreshold=25
[Profile Data Screening Options]
PRampScreenEnable=FALSE
PRCorScreenEnable=FALSE
PRErrScreenEnable=FALSE
PRVertScreenEnable=FALSE
PRFishScreenEnable=FALSE
PRPctGoodScreenEnable=TRUE
PRMarkBadBelowBottom=FALSE
PRAmplitudeThreshold=30
PRCorrelationThreshold=180
PRErrorVelThreshold=1000
PRVerticalVelThreshold=1000
PRFishThreshold=50
PRPctGoodThreshold=25
[Transformation Options]
XformToEarth=TRUE
Allow3Beam=TRUE
BinMap=TRUE
BeamAngleSrc(0:auto,1:man)=0
ManualBeamAngle=30
HeadingSource(0:adcp,1:navHDT,2:navHDG,3:navPRDID,4>manual)=3
ManualHeading=0
TiltSource(0:adcp,1:nav,2:man)=1
ManualPitch=0
ManualRoll=0
SensorConfigSrc(0:PRfixed,1:Pfixed,2:auto)=2
ConcavitySource(0:convex,1:concave,2:auto)=2
UpDownSource(0:dn,1:up,2:auto)=2
EnableHeadingCorrections=TRUE
SinCorrectionAmplitudeCoefficient=0
SinCorrectionPhaseCoefficient=0
MagneticOffsetEV=0
AlignmentOffsetEA=45
EnableVelocityScaling=FALSE
VelocityScaleFactorForBTVelocities(unitless)=1
VelocityScaleFactorForProfileAndWTVelocities(unitless)=1
EnableTiltAlignmentErrorCorrection=TRUE
TiltAlignmentHeadingCorr(deg)=0
TiltAlignmentPitchCorr(deg)=0
TiltAlignmentRollCorr(deg)=0
[Averaging Options]
AvgMethod(0:time,1:dist)=0
FirstAvgTime=60
SecondAvgTime=1200
FirstAvgDistance=10
SecondAvgDistance=1000
EnableRefLayerAvg=TRUE
RefLayerStartBin=3
RefLayerEndBin=10
[Reference Velocity Options]
RefVelSelect(0:none,1:BT,2:WT,3:LYR,4:NDP,5:NAP,6:NSPD)=1
VelRefLayerStartBin=3
VelRefLayerEndBin=5
RefVelUnitVel(0:mm/s,1:m/s,2:knots,3:ft/s)=1
RefVelUnitDepth(0:m,1:cm,2:ft)=0
[User Exit Options]

```



```
UserWinAdcpEnable=FALSE
UserWinAdcpPath=C:\Program Files\RD Instruments\WinAdcp\WinAdcp.exe
UserWinAdcpUpdateInterval(sec)=10
UserWinAdcpFileType(0:enr,1:enx,2:sta,3:lta)=3
UserAdcpScreening=FALSE
UserNavScreening=FALSE
UserTransform=FALSE
[Shiptrack Options]
ShipTrack1Source(0:Nav;1:BT;2:WT;3:Layer)=0
ShipTrack2Source(0:Nav;1:BT;2:WT;3:Layer)=1
ShipTrack1RedStickEnable=TRUE
ShipTrack1GreenStickEnable=TRUE
ShipTrack1BlueStickEnable=TRUE
ShipTrack2RedStickEnable=TRUE
ShipTrack2GreenStickEnable=TRUE
ShipTrack2BlueStickEnable=TRUE
ShipTrack1RedBin=1
ShipTrack1GreenBin=2
ShipTrack1BlueBin=3
ShipTrack2RedBin=1
ShipTrack2GreenBin=2
ShipTrack2BlueBin=3
ShipTrack1DisplaySelect(0:Lat/Lon;1:Distance)=0
ShipTrack2DisplaySelect(0:Lat/Lon;1:Distance)=0
ShipTrack1WaterLayerStartBin=3
ShipTrack1WaterLayerEndBin=5
ShipTrack2WaterLayerStartBin=3
ShipTrack2WaterLayerEndBin=5
ShipTrackDistanceUnit=0
[ADCP Setup Options]
SetProfileParameters=TRUE
NumberOfBins=60
BinSize(meters)=8
BlankDistance(meters)=8
TransducerDepth(meters)=11
SetBTEnable(0:SendBPCmd,1:Don'tSendBPCmd)=TRUE
ADCPSetupMethod(0:Options,1:CommandFile)=0
BtmTrkEnable(0:SendBP0,1:SendBP1)=1
MaxRange(meters)=500
SetHdgSensorType=TRUE
HdgSensorType(0:internal,1:external)=1
SetTiltSensorType=TRUE
TiltSensorType(0:internal,1:external)=1
SetProcessingMode=TRUE
BandwidthType(0:Wide,1:Narrow)=1
ADCPTimeBetweenEnsemblesSel=1
ADCPTimeBetweenEnsembles=1
```

ANNEXE III

I/O log for RD Instruments VmDas Version 1.42

[2006/01/29, 10:58:59.157]: NMEA [Nav] communication established

[2006/01/29, 10:59:00.095]:

Ocean Surveyor Broadband/Narrowband ADCP

RD Instruments (c) 1997-2005

All rights reserved.

Firmware Version: 23.14

>

[2006/01/29, 10:59:00.110]: TS 060129 105900

[2006/01/29, 10:59:00.157]: TS 060129 105900

>

[2006/01/29, 10:59:00.173]:

>

[2006/01/29, 10:59:00.173]: CR1

[2006/01/29, 10:59:00.251]: CR1

[Parameters set to FACTORY defaults]

>

[2006/01/29, 10:59:00.266]: TE00000000

[2006/01/29, 10:59:00.313]: TE00000000

>

[2006/01/29, 10:59:00.313]: -->Setting Bottom Track parameters...

[2006/01/29, 10:59:00.329]: BP0

[2006/01/29, 10:59:00.360]: BP0

>

[2006/01/29, 10:59:00.360]: TP000000

[2006/01/29, 10:59:00.391]: TP000000

>

[2006/01/29, 10:59:00.407]: -->Setting Surveyor low resolution mode...

[2006/01/29, 10:59:00.423]: WP0

[2006/01/29, 10:59:00.438]: WP0

>

[2006/01/29, 10:59:00.454]: NP1

[2006/01/29, 10:59:00.485]: NP1

>

[2006/01/29, 10:59:00.501]: -->Determining Surveyor mode...

[2006/01/29, 10:59:00.501]: WP?

[2006/01/29, 10:59:00.579]: WP?

WP 000 ----- Number of Pings

>

[2006/01/29, 10:59:00.595]: NP?

[2006/01/29, 10:59:00.673]: NP?

NP 001 ----- Number of Pings

>

[2006/01/29, 10:59:00.688]: -->Setting profiling parameters...

[2006/01/29, 10:59:00.704]: NN80

[2006/01/29, 10:59:00.735]: NN80

>

```
[2006/01/29, 10:59:00.735]: NS400
[2006/01/29, 10:59:00.766]: NS400

>
[2006/01/29, 10:59:00.782]: NF400
[2006/01/29, 10:59:00.813]: NF400

>
[2006/01/29, 10:59:00.829]: ED110
[2006/01/29, 10:59:00.845]: ED110

>
[2006/01/29, 10:59:00.860]:
-->Setting manual ping mode...
[2006/01/29, 10:59:00.876]: CF?
[2006/01/29, 10:59:00.970]: CF?
CF 11110 ----- Set Ctrl Flags {eipibisi*}

>
[2006/01/29, 10:59:00.985]: CF 01110
[2006/01/29, 10:59:01.016]: CF 01110

>
[2006/01/29, 10:59:01.016]:
-->Setting the sensor source...
[2006/01/29, 10:59:01.032]: EZ?
[2006/01/29, 10:59:01.126]: EZ?
EZ 10111010 ----- Sensor Source {cid;hipirisitiu}

>
[2006/01/29, 10:59:01.141]: EZ10222010
[2006/01/29, 10:59:01.188]: EZ10222010

>
[2006/01/29, 10:59:01.204]:
-->Disabling ADCP internal heading or tilt corrections...
[2006/01/29, 10:59:01.220]: EA0
[2006/01/29, 10:59:01.235]: EA0

>
[2006/01/29, 10:59:01.251]: EVO
[2006/01/29, 10:59:01.282]: EVO

>
[2006/01/29, 10:59:01.298]: EI0
[2006/01/29, 10:59:01.329]: EI0

>
[2006/01/29, 10:59:01.329]: EJO
[2006/01/29, 10:59:01.360]: EJO

>
[2006/01/29, 10:59:01.376]:
-->Setting beam radial output mode...
[2006/01/29, 10:59:01.391]: EX00000
[2006/01/29, 10:59:01.423]: EX00000

>
[2006/01/29, 10:59:02.001]: CS

[2006/01/29, 12:17:18.232]: NMEA [NAV_PROCESS] Invalid GGA data.
[2006/01/29, 12:17:18.264]: CScreenNav::ProcessNmea() - Invalid GGA data.
[2006/01/29, 12:17:18.279]: CScreenNav::ProcessNmea() - Invalid VTG data.
[2006/01/29, 19:55:25.413]:
```

ANNEXE IV

```

*****
**** Proc_gyro.doc ****
**** Notes on running "Proc_gyro.m" to create ****
**** a model of gyrocompass behavior. ****
**** ****
**** For additional notes on gyrocompass ****
**** modelling, see Gyro.doc ****
*****

```

This routine is set up to create a model of gyrocompass errors based upon input data from an ADCP CODAS database. The model is based upon theory from:

"Gyroscopic Theory, Design and Instrumentation"
by Wrigley, Hollister, and Denhard

Specifically, this model is for a Sperry gyrocompass as described in Chapter 10, Equations 10.36 to 10.39. This script employs a 3rd order Runge-Kutta solution of the second order differential equation modelling gyrocompass error (eq. 10.38b). (Note, this can be changed to a fifth order Runge-Kutta solution by switching the call in Gyro_dyn.m from RK3.m to RK5.m. However, given noise levels in input data, the higher order solution does not appear significantly better). In addition to the solution to the differential model, the script creates a steady-state solution which makes no attempt to track compass response time or oscillations. A filtered version of the steady state response is also generated, which typically shows some improvement over the steady state solution.

Input Data

Two input files are needed for the time period to be modelled. The first is output by the "smoothr" routine in CODAS navigation processing (run in the "/nav" directory) as an ascii flat file with the extension "*.sm". This file contains ensemble times and the best estimates of ship's u, ship's v, reference layer u, reference layer v, latitude and longitude data. The second output file is generated by the "lst_hdg" routine in CODAS (typically run in the "/cal/rotate" directory). This creates an ascii file with ensemble time, mean gyrocompass heading and last gyrocompass heading given the extension ".hdg". Note that this file must be manually edited to remove the comments at the beginning before Matlab can read it. These input files are first processed by the "splitset.m" script to break the data at large gaps.

As a general rule, the model output will not be sensitive to small changes in the input data. Thus, a complete ADCP calibration does not need to be done before generating the files for creating the model. In most cases where gyrocompass errors are relatively small (+/- 2-3 degrees at most), the "smoothr" and "lst_hdg" files can be generated before ADCP editing and calibration have been completed.

Model Parameters

At the top of the "Proc_gyro.m" script, there is a section of model parameters which must be edited.

1) Compensation terms

Two arrays must be defined to specify the gyrocompass compensation used with a particular data set. Gyrocompass compensation is set based upon latitude and north-south velocity. Latitude compensation is specified with an array called "clat" with the same size as the "lat" array (loaded from the "*.sm" input file). As with the input "lat" values, "clat" should be in degrees. Several possible examples are shown below.

- a) No latitude compensation


```
clat = zeros(size(lat));
```
- b) Constant latitude compensation


```
clat = ones(size(lat)) * mean(lat);
```
- c) Latitude compensation updated once per day


```
tt = fix(time);
clat = zeros(size(lat));
for ii=min(tt):max(tt)
    k=find(tt == ii);
    clat(k) = ones(size(k))*mean(lat(k));
end;
clear tt ii k
```

North-south velocity compensation is specified with an array called "cV" with the same size as the north-south "V" array (loaded from the input "*.sm" file). As with input "U" and "V" values, "cV" should be in m/s. Several examples are shown below.

- a) No velocity compensation


```
cV = zeros(size(V));
```
- b) Constant velocity compensation - set at cruising speed


```
cV = 5.14 * cos(hd*pi/180);    % 10 knot/hr = 5.14 m/s
```
- c) Velocity compensation set high when underway, zero otherwise


```
cV = zeros(size(V));
k = find(sqrt(U.^2 + V.^2) >= 2); % underway - sp > 2
cV(k) = 5.14 * cos(hd(k)*pi/180);
k = find(sqrt(U.^2 + V.^2) < 2); % on station - sp < 2
cV(k) = zeros(size(k));
```

2) System constants

Several constants must be set which affect model behavior. Of the four constants shown, "gamma" is the only one which would normally be modified. The constants, their meanings and typical values are shown below.

gamma - eccentric offset, a physical parameter of the compass itself, expressed in radians. Typical values: 0.015-0.025.

Wie - earth's inertial angular velocity. 7.29e-5 1/sec, or 7.29e-5*86400 1/day (since a decimal day time base is used).

R - earth's radius in meters, 6.37e6 meters.

ws - natural frequency of system, equal to $2\pi/84.4$ 1/min, if the compass is assumed to be Schuler tuned. Schuler tuning is used to minimize errors, and can be assumed to be present on all gyrocompasses. Set to $(2\pi/84.4)*1440$ 1/day to match decimal day time base.

Running the model

Once the beginning section of the "Proc_gyro.m" script has been edited, the model is run from within Matlab by simply typing "Proc_gyro". The routine prompts for the input file name containing the lat, lon, U, V, and hd (heading) data. Thus the script can be run on multiple input files using the same compensation without editing the script each time. Note that the model takes some time to run. To process one days worth of data at 5 minute ensembles (about 280 samples) requires about 20-25 minutes on a Sun workstation.

Output File

The model is saved in a Matlab format file, using the name base from the input file and the addition "_gyro.mat". The output file contains the following variables.

Name	Size	Elements	Bytes	Density	Complex
R	1 by 1	1	8	Full	No
U	10267 by 1	10267	82136	Full	No
V	10267 by 1	10267	82136	Full	No
Wie	1 by 1	1	8	Full	No
cV	10267 by 1	10267	82136	Full	No
clat	10267 by 1	10267	82136	Full	No
ddrV	10267 by 1	10267	82136	Full	No
dphi	10267 by 1	10267	82136	Full	No
drV	10267 by 1	10267	82136	Full	No
gamma	1 by 1	1	8	Full	No
have_ash	1 by 1	1	8	Full	No
hd	10267 by 1	10267	82136	Full	No
lat	10267 by 1	10267	82136	Full	No
lon	10267 by 1	10267	82136	Full	No
phi	10267 by 1	10267	82136	Full	No
phi_ss	10267 by 1	10267	82136	Full	No
phi_ssf	10267 by 1	10267	82136	Full	No
rV	10267 by 1	10267	82136	Full	No
rcV	10267 by 1	10267	82136	Full	No
rclat	10267 by 1	10267	82136	Full	No
rlat	10267 by 1	10267	82136	Full	No
time	10267 by 1	10267	82136	Full	No
ws	1 by 1	1	8	Full	No

The results are stored as shown below.

phi - gyrocompass error based on differential model
 phi_ss - gyrocompass error based on steady state model
 phi_ssf - filtered steady state model output

Example

Input: kn32p01.mat (output of splitset.m)

Execute: Matlab

Proc_gyro

Which data file to load (no extension) ? kn32p01

Loading data

Defining constants and system compensation

Running dynamic modelling routine

 Setting initial conditions

 Solving array of differential equations

Running steady state modelling routine

Saving Results

Output: kn32p_gyro.mat

ANNEXE V

OSU TIDAL PREDICTION Software (OTPS)

1. INTRODUCTION

OTPS accomplish 2 tasks:

- extracting harmonic constants from barotropic tidal solutions in OTIS format (see below) at given locations;
- predicting tides at given times and locations.

Predictions are based on global and/or regional barotropic inverse tidal solutions obtained with OTIS and available for download at:

<http://www.coas.oregonstate.edu/research/po/research/tide/>

2. MODEL FILE FORMAT

The files are big_endian binary, recorded in bytes on a UNIX system.

There is one header record, which gives the grid size (n grid cells in longitude and m grid cells in latitude), number of constituents (nc), limits of the area (theta_lim(2), phi_lim(2)), and constituent names (c_id(nc)). The header is followed by nc records, each giving the either elevation fields (m, ocean tide) OR transports fields (m²/s) for one constituent.

Everything is given as complex amplitudes, so that the partial tide for a single constituent of frequency w is given by

$$h(t,x) = \text{Re} [h(x) \exp \{ i [w (t - t_0) + V_0(t_0)] \}]$$

where $V_0(t_0)$ is the astronomical argument for the constituent at t_0 . Note that with the usual conventions, amplitude and phase are given by $\text{amp} = |h|$ $\text{phase} = \text{atan}(-\text{Im}(h)/\text{Re}(h))$.

To read with a simple FORTRAN program:

```

      .
      .
      .
      integer n,m,nc
      real theta_lim(2),phi_lim(2)
      character*4 c_id(21)
      complex, allocatable: h(:,:,:),uv(:,:,:)
      complex, allocatable: u(:,:,:),v(:,:,:)
      .
      .
      .
      open(unit=1,file='h_tpxo6.2',form='unformatted',
*        status='old',convert='big_endian')
ccc (header)
      read(1) n,m,nc,theta_lim,phi_lim,c_id(1:nc)
      allocate(h(nc,n,m))
      do ic=1,nc
        read(1)h(ic,:,:)
      enddo
      .
      .
      .
      allocate(uv(2,n,m),u(nc,n,m),v(nc,n,m))

```



```

        open(unit=1,file='u_tpxo6.2',form='unformatted',)
        *      status='old',convert='big_endian')
ccc  (skip header)
      read(1)
      do ic=1,nc
        read(1) uv
        u(ic, :, :)=uv(1, :, :)
        v(ic, :, :)=uv(2, :, :)
      enddo
      .
      .
      .

```

you now have elevations (m/s) in array h and transports in arrays u and v for nc constituents on a n x m grid, more precisely:

```

==> u(1,...) gives the complex amplitude of zonal transport
      (m**2/s) to the East ... call this U
==> u(2,...) gives meridional transport
      (m**2/s) to the North ... call this V

```

```

*****
3. GRID

```

There are "n" divisions in longitude, "m" in latitude. The elevations and transports are given on a C-grid ... so the locations at which U and V and h are given are not exactly the same. The layout of the grid looks like this:

```

      U(2,1) - h(2,1) - U(2,2) - h(2,2) - ....
                |                   |
                V(2,1)              V(2,2)
                |                   |
      U(1,1) - h(1,1) - U(2,1) - h(2,1) - ....<== theta_lim(1)
                |                   |
                V(1,1)              V(2,1)      <== theta_lim(1)-dy/2
                ^
                |
      ^          phi_lim(1)
      |
phi_lim(1)-dx/2

```

The lats and lons (in degrees) for individual grid nodes are:

```

For h-nodes   lat(i,j) = theta_lim(1) + (j-.5)*dy
              lon(i,j) = phi_lim(1) + (i-.5)*dx

```

```

For U-nodes:  lat(i,j) = theta_lim(1) + (j-.5)*dy
              lon(i,j) = phi_lim(1) + (i-1)*dx

```

```

For V-nodes:  lat(i,j) = theta_lim(1) + (j-1)*dy
              lon(i,j) = phi_lim(1) + (i-.5)*dx

```

Here dx,dy is resolution in longitude and latitude correspondingly. Most of OTIS solutions are given on C grids uniform in lats,lons, but some of them (i.e. Arctic solution) are given on C grids uniform in kilometers. The OTPS is applicable for such solutions also.

 4. SETUP file "setup.inp"

This is EXAMPLE of the file containing your input information.

The original version of file is included in OTPS and given below:

```

lat_lon_time      ! 1. tidal model control file
z                 ! 2. latitude/longitude/time file
m2,s2,n2,k2,k1,o1,p1,q1 ! 3. z/U/V/u/v
AP                ! 4. tidal constituents to include
oce              ! 5. AP/RI
1                ! 6. oce/geo
tmp              ! 7. 1/0 correct for minor constituents
                 ! 8. output file (ASCII)

```

Comments on lines 1-7

1. Tidal model control file (ASCII, supplied with the model, starting from "Model_") contains AT LEAST 3 lines:

```

elevation model file name
transport model file name
bathymetry grid file name
{name of function to convert x,y to lon,lat}

```

Leave line 1 blank, if are not sure which model to use. In this case all models from "model.list" are checked. File "./model.list" should contain all your DATA/Model_* files and is done as:

```
ls -l DATA/Model_*>model.list
```

Please have all Model_* files ready in subdirectory DATA

Please show FULL path to the model files in the "Model_*" file unless they are located in /DATA (download default).

4th line in Model_* file is used ONLY for models calculated on uniform grid in km. Converting functions are provided with current version of OTPS. Please upgrade OTPS if a function is missing.

2. latitude, longitude and time file is an ASCII file, consisting either of 2 OR 8 columns:

```
latitude (degrees) longitude (degrees) [yyyy mm dd hh mm ss]
```

Common sign convention:

```
lat>0 - degrees North, lon>0 - degrees East
```

```
lat<0 - degrees South, lon<0 - degrees West
```

```
[yyyy mm dd hh mm ss] - year month day hour seconds GMT
```

These 6 columns are needed for tide predictions only.

You may leave them empty if only extracting HC

3. Extract HC/predict tide for:

```
z/U/V/u/v - elevation(m) /WE transport(m^2/s)/ SN transport(m^2/s)/
              WE velocity (cm/s) / SN velocity (cm/s)
```

NOTE: Changed, Nov 2004: now for any of u/v/U/V all components (that is U V u v) are calculated by predict_tide.

4. Constituent names should be in LOWER case and separated by comma. Leave the line blank, if ALL model constituents are included
5. AP/RI - output amplitude and phase(GMT) OR real/imaginary parts.

Only used when extracting HC.
May leave blank, if predicting tide.

6. Only needed for tide prediction. 1/0 - Do/Not correct for the minor constituents defined in weights.h. Only works if the 8 tidal constituents fields are in the model: m2,s2,n2,k2,k1,o1,p1,q1

7. oce/geo - extract ocean/geocentric HC for elevations only.
Geocentric tide is appropriate for comparison to or correction of altimetry data.
May leave blank if not z on line 3.

5. COMPILING and RUNNING OTPS

To compile:

```
make extract_HC
make predict_tide
```

If you are NOT on a Unix system, you have to provide options to convert from big_endian binary when compiling (-convert big_endian -assume byterecl). Other possibility is to insert the option convert="big_endian" in the open statement in the source codes.

Your specific OS/Fortran compiler might not have these options. In this case please consult your local computer support on how to read big_endian binary files on your system or contact us on your particular purpose.

Run AFTER editing setup.inp:

```
extract_HC<setup.inp          # have lat/lon file ready
predict_tide<setup.inp      # have lat/lon/time file ready
```

6. ADDING new model

Download compressed and tarred model from
<http://www.coas.oregonstate.edu/research/po/research/tide/>

- a) into the OTPS directory
 - uncompress and tar xvf;
 - DONE!
- b) into any destination:
 - uncompress and tar xvf;
 - cd DATA;
 - edit Model_* control file for exact path to the model files;
 - cp file Model_* into subdirectory DATA of the OTPS directory

Any questions, contact Gary Egbert (541)-737-2947
egbert@coas.oregonstate.edu
or Lana Erofeeva (541)-737-3964
serofeev@coas.oregonstate.edu