

LOGICIEL D'ECHANTILLONNAGE
POUR LE POLYCORDER

par

Richard Bastien

Rapport interne 86-08

Div. de la Gravité, Géothermie et Géodynamique,
Direction de la Physique du Globe,
Ministère de l'Énergie, Mines et Ressources,
1 Prom. de l'Observatoire,
Ottawa, Canada,
K1A 0Y3

Résumé

La Section de la Dynamique de la Croûte de la Direction de la Physique du Globe utilise le micro-ordinateur Polycorder de la compagnie Omnidata pour recueillir, filtrer et stocker des données numériques. Cet appareil a l'avantage d'être compact et léger malgré sa capacité de mémoire de 32 K. Un logiciel sophistiqué utilisant le filtre Kalman permet de filtrer en temps réel les séries chronologiques provenant simultanément de plusieurs canaux. L'utilisation du filtre numérique Kalman est justifiée par le fait qu'il permet de rallonger significativement la période d'autonomie du système en plus de fournir un résultat de haute fidélité.

Abstract

The Crustal Dynamics Section of the Earth Physics Branch is using the micro-computer Polycorder made by the company Omnidata to sample, filter and store numerical data. This computer has the advantage to be compact and light-weight despite having a memory capacity of 32 K. Sophisticated software using the Kalman filter allows real-time filtering of time series data that are recorded simultaneously from several channels. The use of the Kalman filter is justified by the fact that it allows a significant extension of the amount of time that the system can operate unattended while producing a high-fidelity output.

Table des matières

1.0	Introduction	3
1.1	Caractéristiques du filtre Kalman	5
1.2	Guide d'utilisation du logiciel	7
1.3	Contenu des registres et description des variables	9
1.4	Format des fichiers	12
2.0	Description du programme d'initialisation INITLOG	13
2.1	Programme INITLOG	14
3.0	Description du programme principal AUTOLOG,	16
3.1	Programme AUTOLOG	17
4.0	Description et listage du sous-programme KAF	20
4.1	Description et listage du sous-programme RINGB	23
4.2	Description et listage du sous-programme PREDIC	27
4.3	Description et listage du sous-programme RESTEP	31
4.4	Description et listage du sous-programme UPDATE	33
5.0	Description et listage du programme de transfert XMT	37
6.0	Appareillage utilisé pour le transfert des données	43
7.0	Instructions pour transférer un fichier de données du Polycorder au Portapac	44
7.1	Instructions pour transférer un fichier de données du Portapac au Polycorder	45
7.2	Instructions pour transférer les fichiers PGM, FMT et DAT du LSI au Polycorder	46
8.0	Distribution de la mémoire du Polycorder pour 8 canaux ...	47
9.0	Références	48

1.0 Introduction

Le présent logiciel a été écrit pour répondre à la nécessité de filtrer différents signaux géophysiques et météorologiques. Le logiciel est basé sur l'utilisation du filtre Kalman en mode continue. Voir Gelb, A. (1974). Cette méthode de traitement des données chronologiques a été publiée par John Halpenny (1984) et un programme en langage FORTRAN a été écrit en vue d'être utilisé pour supporter le système automatique d'observation géophysique de Charlevoix, Québec. En d'autres endroits, il était plus commode de recourir à un système compact et portatif. Le Polycorder devenait donc l'outil indiqué pour l'échantillonnage des données sur le niveau et la pression de l'eau dans les puits et divers données météorologiques pour des périodes plus ou moins longues. Ce système pourrait être utilisé avec profit pour toute observation de phénomènes variant dans le temps tels que l'inclinaison de la croûte, la résistivité électrique dans la croûte, le champ magnétique ou la marée terrestre.

Avant de traduire le code en langage Polycorder, l'algorithme fut simplifié au maximum en vue d'utiliser le moins d'espace mémoire possible. Malgré que l'appareil ne contienne qu'une mémoire vive de 32 K, il peut offrir une autonomie d'opération d'environ 250 jours sur un seul canal avec écriture une fois l'heure et de 25 jours avec un maximum de 10 canaux pour le même intervalle d'écriture. La période d'autonomie correspond à la capacité de l'espace mémoire. Cependant, puisque les données sont conservées dans un fichier dit circulaire, les dernières valeurs recueillies ne sont jamais perdues; ils prennent plutôt la place des plus anciennes. Les données peuvent être transférées directement à l'ordinateur principal (en l'occurrence un LSI 11/23) ou par l'intermédiaire d'un Portapac d'une capacité de 64 K. Les données brutes peuvent facilement être transférées vers un médium quelconque par l'intermédiaire de la sortie RS-232. Cette option permettra à l'opérateur d'examiner à loisir les données telles que reçues à la source.

Caractéristiques du Polycorder

Le Polycorder modèle 516B, version 5 est manufacturé par Omnidata International, Inc. Il possède 10 entrées parallèles analogues, 9 entrées et 5 sorties parallèles numériques via deux entrée/sortie RS-232; ASCII 8 bit standard. Une horloge interne pouvant accuser une erreur appréciable de 3 minutes par mois, permet un contrôle départ/arrêt automatique. L'option départ/arrêt peut aussi être contrôlé par ligne téléphonique. L'alimentation est fournie par deux piles de 9 volts internes pour un affichage continue de 40 heures ou par une alimentation extérieure. Le clavier est composé de 21 touches scellées comprenant les chiffres 0-9, les caractères A-Z et 4 touches de contrôle par flèches. L'affichage à cristaux liquides présente deux rangées de 16 colonnes.

Le Polycorder peut transmettre ou recevoir à des taux de 300 à 9600 bauds. Le langage codé comprend 100 énoncés symboliques de 2 caractères avec possibilité d'arguments. L'appareil mesure 20.3 cm X 11.4 cm X 7.6 cm et pèse 1.46 kg. Ses limites de fonctionnement selon la température ambiante sont de -20° à 55° C à 100% d'humidité relative.

Brève description du filtre Kalman

Ce filtre permet de traiter les données brutes de manière à éliminer le bruit de fond, les décharges et les sauts d'échelle. Puisque le filtrage se fait sur le champs, il devient nécessaire de ne conserver qu'une fraction des données ce qui rallonge d'autant la période d'autonomie du système. Une application typique enregistre une donnée par minute et ne conserve qu'une valeur par heure.

Le filtre Kalman applique la notion de régression linéaire sur une ou plusieurs valeurs mathématiques décrivant l'évolution d'une variable afin d'en estimer la meilleure courbe en terme des moindres carrés. A cet effet on utilise une matrice de covariance et un vecteur d'état.

Le vecteur d'état contient 4 éléments décrivant l'état de l'évolution de la série en un point donné. Le premier élément représente la valeur filtrée au temps t , les deuxième et troisième éléments contiennent les dérivées première et seconde en ce point. Le quatrième élément sera nul si la série filtrée est continue. Lorsqu'un saut est détecté, l'élément quatre conserve l'amplitude du saut présumé jusqu'à la fin du saut ou d'un délai choisi (le plus court des deux). Si le saut est accepté, cette valeur sera additionnée au premier élément du vecteur d'état. L'élément quatre sera remis à zéro dans tous les cas jusqu'au prochain saut éventuel.

La matrice de covariance permet d'estimer une valeur x basée sur l'observation, telle que la covariance entre x et sa dérivée première et seconde soit minimum. Elle permet ainsi de minimiser l'erreur entre la courbe estimée et la série brute. Cette erreur contribue à l'établissement du niveau de bruit qui est admis comme étant aléatoire. Tel qu'expliqué auparavant, l'algorithme du filtre Kalman a été modifié afin d'utiliser le moins possible d'espace mémoire. Pour cette raison, la matrice de covariance qui avait originalement la dimension 4X4 a été réduite à un vecteur de 10 éléments. En effet, pour les besoins des calculs, on peut utiliser seulement les éléments de la matrice triangulaire supérieure de la matrice de covariance. Les diverses manipulations matricielles ont ainsi été réduites à leur expression finale. Dans les lignes qui suivent nous utiliserons plutôt le terme vecteur de covariance.

Pour plus de détails concernant le développement mathématique du filtre Kalman, voir Gelb, A. (1974).

1.1 Caractéristiques du filtre Kalman

Le délai et le niveau de bruit sont les deux paramètres du filtre Kalman. Bien que le niveau de bruit puisse être évalué directement à partir de la série de données à filtrer, il n'est pas facile d'en faire autant pour établir correctement la durée du délai.

L'effet du délai sur le filtre Kalman pourrait être comparé à celui de la longueur N de l'échantillonnage d'une moyenne mobile. En effet, si le délai est assez long, il réussira à lisser suffisamment les variations d'un signal bruyant mais il risque aussi d'éliminer des oscillations de courtes périodes qui seraient représentatives au phénomène mesuré. Par ailleurs, si le délai est assez court, les oscillations de courtes périodes propres au phénomènes seront conservés mais le bruit associé au signal ne sera peut être pas suffisamment éliminé.

Variation de l'erreur en fonction du rapport période/délai.

Considérons la fonction $y = \cos(x)$ d'amplitude 1, pour l'intervalle 0° à 360° . Nous avons appliqué le filtre Kalman sur cet intervalle qui était constitué d'un nombre de points déterminés définissant la période. Le niveau de bruit a été fixé à 0.001 pour éviter d'utiliser zéro et le filtre a été initialisé à 1.0 pour qu'il adopte dès le départ la valeur à $\cos(0)$. Pour différents délais, on a mesuré l'erreur à 180° . Comme le montre les tableaux suivants, plus le délai augmente, plus l'erreur est grande. L'erreur est nulle lorsque le délai est égal à 1. On a aussi calculé le rapport entre la période et le délai. Le test a été fait sur cinq périodes différentes, mais selon les mêmes rapports période/délai pour vérifier si l'erreur à 180° variait en fonction de la période.

Période : 100 points		
délai	erreur % à 180°	$\frac{\text{période}}{\text{délai}}$
1	0.0000	100.
2	0.0001	50.
3	0.0071	33.3
4	0.0283	25.
5	0.0786	20.
6	0.177	16.7
7	1.63	10.

Période : 200 points		
délai	erreur % à 180°	$\frac{\text{période}}{\text{délai}}$
2	0.0001	100.
4	0.0018	50.
6	0.0110	33.3
8	0.0384	25.
10	0.0995	20.
12	0.214	16.7
20	1.82	10.

Période : 500 points		
délai	erreur % à 180°	période délai
5	0.0001	100.
10	0.0025	50.
15	0.0138	33.3
20	0.0450	25.
25	0.113	20.
30	0.238	16.7
50	1.94	10.

Période : 1000 points		
délai	erreur % à 180°	période délai
10	0.0002	100.
20	0.0028	50.
30	0.0148	33.3
40	0.0477	25.
50	0.118	20.
60	0.246	16.7
100	1.98	10.

Période : 2000 points		
délai	erreur % à 180°	période délai
20	0.0002	100.
40	0.0030	50.
60	0.0153	33.3
80	0.0489	25.
100	0.120	20.
120	0.250	16.7
200	2.00	10.

Les résultats indiquent que pour un rapport période/délai donné, l'erreur à 180° ne varie pas appréciablement en fonction de la période. Pour le choix d'un seuil d'erreur acceptable correspondra donc une valeur période/délai donnée. Par exemple, si on choisi un seuil maximum de 0.1% comme étant acceptable, nous pourrions conclure que pour un signal sans bruit, le rapport période/délai minimum devrait être environ de 20.

La constante de temps du filtre Kalman.

La constante de temps du filtre est une fonction du rapport période/délai applicable à un signal dont le bruit de fond est un processus aléatoire. Tout signal dont la période est égale ou plus courte que T tel que $T/\text{délai} = 1$ sera filtré. Par ailleurs, selon le critère de Nyquist (Hamming 1982), la période minimum que l'on peut enregistrer d'un signal est égale à 2 fois l'intervalle de l'échantillonnage. Ceci implique que toutes les fréquences supérieures à cette limite peuvent affecter le résultat du filtre en induisant un phénomène de battement. On devra donc s'assurer que le signal considéré ne contienne pas de fréquences supérieures au critère de Nyquist soit en le vérifiant à l'aide d'un analyseur spectral ou en installant un filtre électronique approprié. La constante de temps du filtre Kalman correspondant à l'application actuelle n'a pas été établie. La courbe de la fonction de transfert pourrait être obtenue en traçant le rapport entre l'amplitude du spectre des fréquences des données de sortie et d'entrée du filtre pour un délai donné.

1.2 Guide d'utilisation du logiciel

Lors d'un changement d'intervalle de lecture et d'écriture et/ou du nombre de canaux, il peut s'avérer nécessaire de modifier le logiciel.

La liste de corrections suivantes est présentée dans le but de faciliter la tâche de l'utilisateur lorsqu'il désire implanter le système ou s'il désire y amener des améliorations. Il s'agit donc ici d'une liste de points de vérifications qui lui permettra de s'assurer que rien n'a été laissé au hasard. Seuls les points correspondant à une amélioration spécifique devront être corrigés.

1- Le nombre de lignes dans le fichier KVAR doit correspondre au nombre de canaux actifs. Corriger les constantes du fichier KVAR pour chaque canal. Les deux premières colonnes de KVAR doivent être considérées s'il y a modification aux valeurs des constantes ou au nombre de lignes: le délai du filtre (DELAY) en colonne 1 et le bruit prévu (NOISE) en colonne 2. Ces ajustements ne sont pas obligatoires si le nombre de canaux actifs diminue.

2- Correction de AUTOLOG ou lignes à vérifier:

- Ligne 4 - Le balayage des canaux actifs (SCN).
- Ligne 56 - Canal à émettre au RS-232. Si la valeur ne représente pas un canal actif, il n'y aura pas d'émission
- Ligne 101 - Entrer <CTL 013> et <CTL 010> manuellement au clavier du Polycorder.
- Ligne 108 - L'argument constant de la fonction MOD doit être égal à l'intervalle des écritures en minutes.
- Ligne 123 - Nom du fichier index existant.
- Ligne 124 - Nom du fichier de données existant.
- Ligne 125 - Nom du fichier de données existant.

3- Correction de INITLOG ou lignes à vérifier:

- Ligne 3 - Nombre limite de pages dans le fichier de données (PMAX).
- Ligne 6 - La valeur de la constante doit être égale au nombre de canaux actifs (NC).

4- Correction de RINGB ou lignes à vérifier:

- Ligne 11 - Valeur de l'intervalle des écritures en minutes.
- Ligne 44 - Mettre 31.3999 s'il s'agit d'une année bissextile.

- 5- Correction de XMT ou lignes à vérifier:
- Ligne 2 - Valeur de l'intervalle des écritures en minutes.
 - Ligne 76 - Entrer <CTL 013> et <CTL 010> manuellement au clavier du Poly.
 - Ligne 81 - Nom du fichier index existant.
 - Ligne 100 - Nom du fichier de données existant.
 - Ligne 126 - L'argument constant de la fonction MOD doit être égal au nombre de lignes par page (LPP).
 - Ligne 154 - L'argument constant de la fonction MOD doit être égal au nombre de lignes par page (LPP).
 - Ligne 174 - Entrer <CTL 013> manuellement au clavier du Poly.
 - Ligne 176 - Entrer <CTL 026> manuellement au clavier du Poly.
- 6- S'il y a un changement dans le nombre de canaux actifs ou dans le nombre de lignes par page, le fichier FORMAT associé au fichier de données doit être effacé et remplacé en spécifiant le nouveau nombre de colonnes et de lignes par page. Avant d'effacer le fichier FORMAT, le fichier associé doit être d'abord effacé. Le nombre de lignes par page correspond au nombre d'écritures par 24 heures.
- 7- Lorsque les fichiers index et de données sont recréés, il doivent être remplis de zéros. Cette opération est nécessaire pour permettre l'utilisation de ces fichiers en mode d'accès direct. (Voir l'énoncé SLC). Un simple programme permet de remplir le fichier de zéros.
- 8- Le Polycorder peut aussi nécessiter un ajustement lorsque l'intervalle des lectures est modifié. Le MODE 5-1 permet de spécifier l'intervalle des lectures EN SECONDES et le temps des lectures EN SECONDES. Il est suggéré de fixer le temps des lectures à une seconde avant le temps exact désiré. Par exemple, si les lectures se font à toutes les minutes, on fixe START TIME = 59. Le Polycorder effectuera la lecture environ une seconde plus tard, soit au temps zéro. Le logiciel ne permet pas un intervalle de lecture inférieur à 10 secondes. Cette limite est due à la vitesse d'exécution du programme sur un canal. L'intervalle de lecture maximum est de 60 minutes. Ceci est dû au fait que l'intervalle maximum pour les écritures est aussi de 60 minutes.
- 9- Remettre le registre 15 à zéro s'il s'agit de la toute première fois qu'on utilise AUTOLOG. Ce registre contient le nombre de lignes (NREC) entrées dans le fichier de données depuis le dernier transfert. Le registre 15 sera remis à zéro lors du transfert des données

1.3 Contenu des registres et description des variables

0	Lecture du canal 1	31	COV1
1	Lecture du canal 2	32	COV2
2	Lecture du canal 3	33	COV3
3	Lecture du canal 4	34	COV4
4	Lecture du canal 5	35	COV5
5	Lecture du canal 6	36	COV6
6	Lecture du canal 7	37	COV7
7	Lecture du canal 8	38	COV8
8	Lecture du canal 9	39	COV9
9	Lecture du canal 10	40	COV10
10	Lecture du canal 11	41	PH1
11	Lecture du canal 12	42	PH2
12	START (Reservé)	43	PH3
13	MMM (Reservé)	44	PH4
14	JJJ (Reservé)	45	STATE1
15	NREC (Reservé)	46	STATE2
16	PPLLL (Reservé)	47	STATE3
17	DRAP (Reservé)	48	STATE4
18	PMAX (Reservé)	49	OUTPUT
19	NC (Reservé)	50	Compteur
20	DELAY	51	Compteur
21	NOISE	52	(Reservé)
22	OBSVAR	53	Compteur
23	RESLIM	54	Compteur
24	OBVARP	55	Compteur
25	DECAY	56	Compteur
26	DELSQ (DQ)	57	
27	RESID	58	
28	RHD	59	
29	RESD	60	
30	STEPTI	61	(Reservé)
		62	(Reservé)
		63	(Reservé)

START = Si le registre 12 contient la valeur zéro, c'est-à-dire si XMT vient juste d'être exécuté, la page et la ligne de départ du fichier de données y seront inscrites sous la forme: (1000*page + ligne).

MMM = Le nombre de minutes écoulées depuis 0h00 est calculé à partir de TIM et en utilisant la formule $MMM = hr_min - hr*40$. Cette valeur correspond à l'instant où la lecture a eu lieu.

JJJ = Le jour de l'année. Cette valeur est remise à jour par le sous-programme RINGB à chaque fois qu'une nouvelle page est rajoutée dans le fichier de données. Elle sera alors inscrite dans le fichier INDEX. Le programme devra être modifié s'il faut tenir compte de l'année bissextile.

- NREC = Le nombre de lignes contenues dans le fichier de données est maintenu à jour par le sous-programme RINGB dans le registre 15.
- PPLLL = La position du pointeur dans le fichier de données est maintenue à jour par l'expression $PPLLL = 1000 * PAGE + LIGNE$. Cette valeur est placée dans le registre 16 et sera conservée dans le fichier INDEX.
- DRAP = Lorsque, soit le programme initialisateur ou de transfert des données est exécuté, un drapeau est levé dans le registre 17. Il sera abaissé à la fin du programme AUTOLOG pour indiquer que ce dernier vient d'être exécuté.
- PMAX = Le nombre limite de pages dans la mémoire circulaire sera testé à mesure qu'une nouvelle page devra être commencée. Lorsque cette limite est atteinte, le pointeur sera replacé à la première page. Le registre 18 contient la valeur de PMAX.
- NC = Le nombre de données par ligne correspond au nombre de canaux lus. Les données brutes se retrouvent dans les registres 0,1,2,...,NC-1. NC est fixé lors de l'exécution de INITLOG et sa valeur est conservée dans le registre 19. Ce nombre doit correspondre au nombre de lignes dans le fichier KVAR.
- DELAY = Le délai du filtre est proportionnel au temps. Il représente le nombre de données qui seront lues avant qu'une valeur soit estimée. Ainsi, si $DELAY = 60$ et que les lectures sont faites à toutes les 10 secondes, le délai en temps sera de 10 minutes. Le délai numérique doit être non-nul et le délai en temps ne doit pas dépasser 60 minutes. Le délai est fixé d'après le niveau de bruit et la période minimum du signal.
- NOISE = Le bruit est défini ici comme étant l'écart type de la variation du signal lu par rapport au résultat filtré. Cette valeur peut être tirée directement de l'expérience en appliquant le filtre Kalman avec un bruit estimé et un délai correspondant à la période minimum du signal.
- OBSVAR = Le carré du bruit. ($OBSVAR = NOISE * NOISE$) Cette constante sera introduite dans le vecteur de covariance pour établir la valeur de RHO.
- RESLIM = Vingt-cinq fois le carré du bruit ($RESLIM = 25 * OBSVAR$). Cette constante sera utilisée pour vérifier si un saut doit être considéré ou non.
- OBVARP = Valeur additionnée à l'élément 6 du vecteur de covariance. Sa valeur est donnée par $NOISE * NOISE * (3.1415927 ** 6)$.

- DECAY = Coefficient d'amortissement. Il est fixé à $\text{EXP}(-1/\text{DELAY})$.
- DELSQ = Inverse du carré de la constante de temps numérique.
($1/\text{DELAY}^2$)
- RESID = Le résidu est la différence entre la donnée brute et la somme des trois premiers éléments du vecteur d'état. Cette valeur sera utilisée pour calculer le facteur RESD. Elle sera aussi comparée à RESLIM et selon le résultat du test, servira de mémoire à la valeur du saut décelé.
- RHO = Cette valeur sera utilisée pour calculer le facteur RESD et pour remettre à date le vecteur de covariance.
- RESD = Facteur appliqué sur le vecteur des coefficients. Il sera utilisé pour remettre à date le vecteur d'état.
- STEPTI = Compteur servant à déterminer si la durée d'un saut a atteint la limite prévue. Cette durée est égale au délai du filtre.
- C1 à C10 = Les dix éléments du vecteur de covariance. La constante OBSVAR est entrée dans la matrice via l'élément C10.
- P1 à P4 = Les quatre éléments du vecteur des coefficients. Ce vecteur sera utilisé pour remettre à date le vecteur d'état et le vecteur de covariance.
- S1 = Premier élément du vecteur d'état. Il représente la valeur filtrée.
- S2 = Deuxième élément du vecteur d'état. Il représente la dérivée première de la série filtrée au point considéré.
- S3 = Troisième élément du vecteur d'état. Il représente la dérivée seconde de la série filtrée au point considéré.
- S4 = Quatrième et dernier élément du vecteur d'état. Cet élément est généralement nul, mais si un saut est en instance d'être accepté, il servira de mémoire pour mettre à jour la valeur du saut qui sera éventuellement accepté ou rejeté.
- OUTPUT = Donnée filtrée à conserver. Cette valeur sera modifiée lorsque le temps réel moins le délai du filtre sera un multiple de 60 minutes.

1.4 Format des fichiers

Format de DATA1: Le nombre de ligne par page est déterminé en divisant le nombre de minutes dans 24 heures (1440) par l'intervalle des écritures. On obtient ainsi une page par jour.

Col.	Variable	Format
1	CANAL1	N9.5
2	CANAL2	N9.5
"	"	"

Format de KVAR:

Col.	Variable	Format	Col.	Variable	Format	Col.	Variable	Format
1	DELAY	N6	11	STEPTI	N6	21	COV10	F9.7
2	NOISE	F9.7	12	COV1	F9.5	22	PH1	F9.5
3	OBSVAR	F9.7	13	COV2	F9.5	23	PH2	F9.6
4	RESLIM	F9.6	14	COV3	F9.5	24	PH3	F9.6
6	OBVARP	F9.5	15	COV4	F9.6	25	PH4	F9.7
5	DELSQ	F9.7	16	COV5	F9.6	26	STATE1	F9.6
7	DECAY	F9.7	17	COV6	F9.6	27	STATE2	F9.7
8	RESID	F9.6	18	COV7	F9.7	28	STATE3	F9.7
9	RHO	F9.4	19	COV8	F9.7	29	STATE4	F9.6
10	RESD	F9.7	20	COV9	F9.7	30	OUTPUT	F9.6

Format de INDEX:

Col.	Variable	Format
1	PPLLL	N5
2	JJJ	N4

2.0 Description du programme d'initialisation INITLOG

Ce programme initialise le fichier KVAR. Il sera exécuté avant que le programme KAF soit utilisé pour la première fois ou dans le cas où KAF ait été interrompu pendant une période de temps suffisamment longue. L'exécution de INITLOG coupe définitivement le lien entre deux séries de données chronologiques traitées par KAF en initialisant les éléments 3 à 7 et 11 à 30 du fichier de travail KVAR. Le programme INITLOG calcule les valeurs de 5 constantes qui seront régulièrement utilisées dans le logiciel. Ces constantes sont conservées par les éléments 3 à 7 de KVAR. Les éléments 11 à 30 contiennent les variables correspondantes à chaque canal. Chaque ligne de KVAR représente un canal.

Appels de sous-routines

Aucun

Contenu des registres

Lecture seulement

Aucun

Lecture et écriture

R(51) = Compteur de canaux
 R(53) = Variable temporaire
 R(54) = Variable temporaire
 R(50) = Compteur de registres

Écriture seulement

R(17) = DRAP
 R(18) = PMAX
 R(19) = NC

Fichiers

La tâche du programme est d'écrire directement dans le fichier KVAR. A la fin du programme le fichier KVAR reste ouvert. Ce fichier est constitué d'une seule page. Il y a 30 colonnes et NC lignes, NC étant le nombre de canaux actifs.

2.1 Programme INITLOG

```

1 CON 1 ;A1
2 STO 17 ;A1=>R(17)DRAP DRAP = Drapeau pour RING8.POL
3 CON 34 ;A34
4 STO 18 ;A34=>R(18)PMAX PMAX = Nombre limite de pages
5 OPN KVAR ;Fichier de transfert utilisé par KAF
6 CON 2 ;A2 ===>> nombre de canaux actifs
7 STO 19 ;A2=>R(19)NC NC = nombre de canaux
8 STO 51 ;A2=>R(51)c COMPTEUR de canaux
9 CON 1 ;<=CANAL ;A1
10 RCL 51 ;Ac=>B1
11 CON 1 ;A(1)=>Bc=>C1
12 SLC ;Place le pointeur de KVAR à (C1,Lc,P1)
13 RCF ;A(DELAY)
14 INV ;A(1/DELAY)
15 STO 53 ;A(DEL)=>R(53)
16 ICP ;(C2,Lc,P1)
17 RCF ;A(NOISE)=>B(DEL)
18 PSH ;A(NOISE)=>B(NOISE)=>C(DEL)
19 MLT ;A(OBSVAR),B(DEL)
20 STO 54 ;A(OBSVAR)=>R(54)
21 ICP ;(C3,Lc,P1)
22 STF ;A(OBSVAR)=>KVAR(C3,Lc,P1) OBSVAR
23 CON 25 ;A(25)=>A(OBSVAR)=>C(DEL)
24 MLT ;A(RESLIM),B(DEL)
25 ICP ;(C4,Lc,P1)
26 STF ;A(RESLIM)=>KVAR(C4,Lc,P1) RESLIM
27 POP ;A(DEL)
28 RCL 54 ;A(OBSVAR)=>B(DEL)
29 CON 961.38919 ;A(PI**6)=>B(OBSVAR)=>C(DEL)
30 MLT ;A(OBVARP),B(DEL)
31 ICP ;(C5,Lc,P1)
32 STF ;A(OBVARP)=>KVAR(C5,Lc,P1) OBVARP
33 POP ;A(DEL)
34 CON -1 ;A(-1)=>B(DEL)
35 MLT ;A(-1*DEL)
36 EXP ;A(DECAY)
37 ICP ;(C6,Lc,P1)
38 STF ;A(DECAY)=>KVAR(C6,Lc,P1) DECAY
39 RCL 53 ;A(DEL)
40 PSH ;A(DEL)=>B(DEL)
41 MLT ;A(DELSQ)
42 ICP ;(C7,Lc,P1)
43 STF ;A(DELSQ)=>KVAR(C7,Lc,P1) DELSQ
44 ICP *;(C8,Lc,P1)
45 ICP ;(C9,Lc,P1)
46 ICP ;(C10,Lc,P1)
47 ICP ;(C11,Lc,P1)
48 CON 999 ;A999
49 STF ;A999=>KVAR(C11,Lc,P1) STEPTI = 999
50 ICP ;(C11+1,Lc,P1)

```

```
51 STF                ;A999=>KVAR(C12,Lc,P1)          COV(1) = 999
52 ICP                ;(C12+1,Lc,P1)
53 CON 17             ;A(17)
54 STO 50             ;A(17)=>R(50)K          COMPTEUR de registres
55 CON 0              ;A0
56 STF                ;<=INITCO;A0=>KVAR(C13+K,Lc,P1)
57 ICP                ;(C13+1,Lc,P1);(C13+2,Lc,P1);...;(C13+17,Lc,P1)
58 DJN 50,56         ;=>INITCO;R(50)K-1:IF(R(50).NE.0) GOTO INITCO
59 DJN 51,9          ;=>CANAL ;R(51)c-1:IF(R(51).NE.0) GOTO CANAL
60 END
```


3.0 Description du programme principal AUTOLOG

AUTOLOG est le programme principal qui contrôle tout le logiciel nécessaire pour obtenir des valeurs filtrées à partir des mesures prises par une sonde. INITLOG devra toujours être exécuté en premier lorsque AUTOLOG sera utilisé pour la première fois ou après une période suffisamment longue. Il faudra alors suivre la procédure décrite dans les pages précédentes.

Au moment de chaque lecture, l'affichage présentera l'heure de l'observation en minutes (t), la valeur de la lecture, la valeur filtrée correspondant à la lecture au temps (t-m, m = délai du filtre) et le numéro du canal correspondant. Le temps d'exécution de AUTOLOG peut être évalué en utilisant la formule: $TEMPS = 3.54 + 5.66 * NC$ où NC est le nombre de canaux filtrés.

Appel de sous-routines

KAF = Filtre Kalman produisant des valeurs ajustées à partir des données brutes qui lui sont soumises.

RINGB = Conserve les données filtrées par KAF dans une mémoire dite circulaire. Ce programme n'est appelé qu'à des multiples de l'intervalle des écritures en minutes.

Contenu des registres

Lecture seulement	Lecture et écriture
R(14) = JJJ	R(13) = MMMM
R(19) = NC	R(49) = OUTPUT
R(20) = DELAY	R(50) = Compteur de lignes pour canaux
R(45) = STATE(1)	R(51) = Compteur de canaux
R(27) à R(49)	R(53) = Compteurs divers
	R(54) = Utilisations diverses
Ecriture seulement	R(56) = Secondes pour RINGB
Aucun	

Fichiers

Le programme ouvre le fichier de données avant d'appeler le sous-programme RINGB.

Il fait continuellement des échanges avec le fichier KVAR en transférant pour chaque canal une ligne complète du fichier dans les registres 20 à 49. Lorsque le programme KAF a effectué une itération sur ces valeurs, les éléments 7 à 30 sont replacés dans KVAR et une autre ligne est transférée dans les registres, etc. Le temps d'un transfert des registres au fichier KVAR prend 2.6 secondes par canal. Lorsque le programme fonctionne sur un seul canal, il peut y avoir au maximum une lecture par 10 secondes. Pour une application sur 8 canaux actifs, AUTOLOG doit s'exécuter pendant 50 secondes.

3.1 Programme AUTOLOG

```

1 AON ;Section analog ON
2 XON ;Excitation de 5 volts ON
3 DLY 5 ;Délai de 1/2 seconde.
4 SCN 100,7 ;Balayage du canal 7 pendant .1 seconde
5 TIM ;As,B(hr_min),C(mois_jour)
6 STO 56 ;A(s)=>R(56) Temporairement pour RINGB
7 XFF ;Excitation de 5 volts OFF POUR TESTER LE PROG.,
8 AFF ;Section analog OFF METTRE NOP EN LIGNE 6.
9 POP ;A(hr_min)
10 PSH ;A(hr_min)=>B(hr_min)
11 CON 100 ;A100=>B(hr_min)=>
12 DIV ;A(hr_min),B(hr_min)
13 INT ;A(hr),B(hr_min)
14 CON 40 ;A(40)=>B(hr)=>C(hr_min)
15 MLT ;A(hr*40),B(hr_min)
16 SUB ;A(hr_min-hr*40) *** POUR TESTER, METTRE KYA ***
17 STO 13 ;A(MMM)=>R(13) Nb de min. depuis 0h00
18 WID 4 ;Réserve 4 cases pour l'affichage.
19 DCM 0 ;Aucun chiffre après le point.
20 VUA ;Affichage de l'heure en minutes.
21 OPN KVAR ;Ouvre le fichier KVAR
22 RCL 19 ;A(NC) NC = nombre de canaux
23 STO 51 ;Ac=>R(51)c COMPTEUR DE CANAUX
24 CON 0 ;A0
25 STO 50 ;A0=>R(50)0 COMPTEUR DE LIGNE POUR CANAUX
26 ICR 50 ;<=BOUCLE;R(50)1;R(50)2;...;R(50)c
27 CON 1 ;A1 Lecture de la ligne c du fichier+
28 RCL 50 ;Ak=>B1 KVAR et transfert dans les I
29 CON 1 ;A1=>Bk=>C1 registres 20 à 49 I
30 SLC ;Place le pointeur de KVAR à (C1,Lk,P1) I
31 CON 30 ;A30 30 éléments à lire. I
32 STO 53 ;A30=>R(53)30 I
33 CON 20 ;A20 à partir du registre 20 >
34 STO 54 ;A20=>R(54)20 I
35 RCF ;<=LIRE ;KVAR(Cn,Lk,P1)=>A I
36 STI 54 ;A=>R(20);R(21);R(22);...;R(20+29) I
37 ICR 54 ;R(54)21;R(54)22;R(54)23;...;R(54)50 I
38 ICP ;(Cn+1,Lk,P1) I
39 DJN 53,35 ;=>LIRE ;R(53)e-1:IF(R(53).NE.0) GOTO LIRE +
40 WID 2 ;Réserve 2 cases pour l'affichage
41 CDS 16,0 ;Place le curseur en case 16
42 VUM C- ;Affichage des symboles C-
43 RCL 50 ;Ak
44 DCM 0 ;Aucun chiffre après le point
45 CDS 18,0 ;Place le curseur en case 18
46 VUA ;Affichage du canal traité
47 CON 1 ;A1=>Bk
48 SUB ;A(k-1)
49 STO 53 ;A(k-1)=>R(53) canal (k-1)
50 RCI 53 ;AR(k-1)=Ax x = donnée brute du canal k-1

```

```

51 STO 54           ;A(x)=>R(54)x
52 WID 10          ;Réserve 10 cases pour l'affichage.
53 DCM 6           ;6 chiffres après le point
54 CDS 5,0         ;Place le curseur en case 5
55 VUA            ;Affichage de la donnée brute
56 CON 6           ;A(6)           Canal disponible
57 RCL 50          ;A(k)=>B(6)       Canal en cours
58 SUB            ;A(k-6)
59 JNZ 76          ;=>KAF      ;IF (k.NE.6) GOTO KAF
60 WID 5           ;Réserve 5 cases
61 DCM 0           ;Aucun chiffre après le point
62 RCL 14          ;A(JJJ)
63 MCP 208 74 208 221 74 210      ;Présenté au RS-232
64 MMC 222 154 243 136 66 108     ;pour lecture extérieure
65 MMC 51 104 2 66 4              ;de la donnée brute
66 RCL 13           ;A(MMMM)
67 MCP 208 74 208 221 74 210      ;Présenté au RS-232
68 MMC 222 154 243 136 66 108     ;pour lecture extérieure
69 MMC 51 104 2 66 4              ;de la donnée brute
70 RCL 54           ;A(x)
71 WID 10          ;Réserve 10 cases
72 DCM 6           ;Six chiffres après le point
73 MCP 208 74 208 221 74 210      ;Présenté au RS-232
74 MMC 222 154 243 136 66 108     ;pour lecture extérieure
75 MMC 51 104 2 66 4              ;de la donnée brute
76 GSB KAF          ;<=KAF      ;          **** KAF ****
77 RCL 45           ;A(S1)           S1 = STATE(1)
78 CDS 21,0        ;Place le curseur en case 21
79 VUA            ;Affichage de la valeur filtrée
80 RCL 13           ;A(MMMM)=>B(X)           dans le canal k-1
81 RCL 20           ;A(DELAY)=>B(MMMM)=>C(X)
82 SUB            ;A(MMMM-DELAY),B(X)
83 MOD 60          ;A(MD),B(X)       Correspondance en temps de X
84 JNZ 87          ;=>BAD      ;IF(DM.NE.0) GOTO BAD
85 POP            ;A(X)       Sera conservé dans le fichier de données
86 STO 49          ;A(X)=>R(49)X     à la prochaine écriture
87 CON 1           ;<=BAD      ;A(1)           Lecture des registres 27 à 49  +
88 RCL 50          ;Ak=>B1           et transfert dans la ligne k  I
89 CON 8           ;A(8)=>Bk=>C1     du fichier KVAR.          I
90 SLC            ;Place le pointeur de KVAR à (C8,Lk,P1)      I
91 CON 23          ;A23           22 éléments à écrire      I
92 STO 53          ;A23=>R(53)23
93 CON 27          ;A27           à partir du registre 27  >
94 STO 54          ;A27=>R(54)27      I
95 RCI 54          ;<=ECRIRE;R(R(54))=>A      I
96 STF            ;A=>KVAR(Cn,Lk,P1)      I
97 ICR 54          ;R(54)28;R(54)29;R(54)30;...;R(54)50      I
98 ICP            ;(Cn+1,Lk,P1)          I
99 DJN 53,95       ;=>ECRIRE;R(53)k-1;IF(R(53).NE.0) GOTO ECRIRE      +
100 DJN 51,26      ;=>BOUCLE;R(51)c-1;IF(R(51).NE.0) GOTO BOUCLE

```

```

101 CNS <CTL 013> <CTL 010>; <CR> <LF> => R(S)
102 TRS                               Transfert R(S) au RS-232
103 RCL 56                             ;A(s)
104 CON 500                             ;A(500)=>B(s)
105 SUB                                 ;A(s-500)
106 JGZ 127 ;=>ZZZ ;IF(s>0) GOTO ZZZ      Premier 10 sec. de la min?
107 RCL 13                               ;A(MMM)
108 MOD 60                               ;Ad                               Intervalle des écritures
109 JNZ 127 ;=>ZZZ ;IF(d.NE.0) GOTO ZZZ
110 RCL 19                               ;A(NC)
111 STO 53                               ;A(NC)=>R(53)                       Compteur de ligne
112 CON 0                                ;A(0)
113 STO 54                               ;A(0)=>R(54)                       Compteur de registre
114 CON 1                                ;A(1)
115 PSH                                  ;A(1)=>B(1)
116 CON 30                               ;A(30)=>B(1)=>C(1)
117 SLC                                  ;Place le pointeur de KVAR à (C30,L1,P1)
118 RCF ;<=PASSE ;KVAR=>A(X0);A(X1);...;A(XNC)
119 STI 54                               ;A(X)=>R(0);R(1);...;R(NC)
120 ICR 54                               ;R(54)<=R(54)+1
121 ILP                                  ;(C30,L(1+1),P1)
122 DJN 53,118;=>PASSE ;R(53)<=R(53)-1;IF(R(53).NE.0)GOTO PASSE
123 CNS INDEX                            ;S(INDEX)
124 CNS DATA1                           ;S(DATA1)=>T(INDEX)
125 OPN DATA1                           ;Ouvre le fichier de données DATA1 pour RINGB
126 GSB RINGB                             ;                               **** RINGB ****
127 ZZZ ;<=ZZZ ;Polycorder OFF *** POUR TESTER, METTRE END ***
128 END

```

4.0 Description du sous-programme KAF

Le filtre Kalman produit des valeurs ajustées à partir de données brutes contenant du bruit, des sauts d'échelle ou des interférences. La méthode utilisée est basée sur la théorie du filtre Kalman continue. Il y a un certain délai entre le temps où une donnée est entrée et le temps où une valeur est estimée. Si ce délai est m , alors $x(t)$ représente une estimation de la donnée brute recueillie au temps $(t-m)$. Le programme fait appel à trois sous-programmes nommés PREDIC, RESTEP et UPDATE. Ces trois sous-programmes font partie intégrante du programme KAF et ils ont été isolés pour faciliter la manipulation du logiciel. Incluant l'exécution de ces trois sous-programmes, KAF utilise les registres 20 à 49 qui lui sont transmis sous forme d'arguments par le programme appelant. Ces 30 registres contiennent l'information nécessaire au filtre Kalman pour lui permettre de poursuivre son processus de filtrage sur le canal concerné.

Appels de sous-routines

PREDIC = Prédit un vecteur d'état et un vecteur de covariance à partir des valeurs qui y sont mises à date par UPDATE.

RESTEP = Ramène le système dans un état sans saut.

UPDATE = Construit le vecteur des coefficients en utilisant les valeurs du vecteur de covariance prédites par PREDIC et remet à jour le vecteur d'état et le vecteur de covariance à partir de la nouvelle donnée brute.

Contenu des registres

Lecture seulement

R(20) = DELAY
 R(22) = OBSVAR
 R(23) = RESLIM
 R(45) = S1
 R(46) = S2
 R(47) = S3
 R(54) = Donnée brute

Lecture et écriture

R(27) = RESID
 R(30) = STEPTI
 R(48) = S4
 R(55) = RESID (Temporaire)

Écriture seulement

R(37) = COV7
 R(38) = COV8
 R(39) = COV9
 R(40) = COV10

Fichiers

Aucun

Sous-programme KAF

```

1 GSB PREDIC ; **** PREDIC ****
2 ICR 30 ;R(30)I+1 I = STEPTI => R(30)
3 RCL 30 ;AI
4 RCL 20 ;A(DELAY)=>BI DELAI DU FILTRE KALMAN
5 SUB ;AI-DELAY
6 JLZ 12 ;=>RESID ;IF(AI.LT.DELAY) GOTO RESID
7 RCL 30 ;AI
8 CON 999 ;A999=>BI
9 SUB ;AI-999
10 JPS 12 ;=>RESID ;IF(AI.GE.999) GOTO RESID
11 GSB RESTEP ; **** RESTEP ****
12 RCL 54 ;<=RESID ;Ax x = donnée brute du canal k-1
13 RCL 45 ;AS1=>Bx
14 SUB ;A(x-S1)
15 RCL 46 ;AS2=>B(x-S1)
16 SUB ;A(x-S1-S2)
17 RCL 47 ;AS3=>B(x-S1-S2)
18 SUB ;A(x-S1-S2-S3)
19 STO 27 ;A(x-S1-S2-S3)=>R(27)R R = RESID => R(27)
20 PSH ;AR=>BR
21 MLT ;AR**2
22 RCL 23 ;Ar=>BR**2 r = RESLIM
23 SUB ;A(R**2-r)
24 JLZ 44 ;=>TSTEP ;IF(R**2.LT.r) GOTO TSTEP
25 RCL 27 ;AR
26 RCL 48 ;AS4=>BR
27 SUB ;A(R-S4)
28 STO 55 ;A(R-S4)=>R(55) R = RESID => R(55) Temporaire
29 PSH ;A(R-S4)=>B(R-S4)
30 MLT ;A(R-S4)**2
31 RCL 23 ;Ar=>B(R2) R2 = (R-S4)**2
32 SUB ;A(R2-r)
33 JPS 37 ;=>TINCST ;IF(R2.GE.r) GOTO TINCST
34 RCL 55 ;AR
35 STO 27 ;AR=>R(27)R R = RESID => R(27)
36 JMP 54 ;=>UPDATE ;GOTO UPDATE
37 RCL 27 ;<=TINCST ;AR
38 STO 48 ;AR=>R(48)S4 S4 = STATE(4) => R(48)
39 RCL 22 ;A(OBSVAR)
40 STO 40 ;A(OBSVAR)=>R(40)C10 C10 = COV(10) => R(40)
41 CON 0 ;A0
42 STO 30 ;A0=R(30)I I = STEPTI => R(30)
43 JMP 55 ;=>END ;GOTO END
44 RCL 48 ;<=TSTEP ;AS4
45 JPZ 54 ;=>UPDATE ;IF(S4.EQ.0) GOTO UPDATE
46 CON 0 ;A0
47 STO 48 ;A0=>R(48)S4 S4 = STATE(4) => R(48)
48 STO 37 ;A(0)=>R(37)COV7
49 STO 38 ;A(0)=>R(38)COV8
50 STO 39 ;A(0)=>R(39)COV9

```

```
51 STO 40           ;A(0)=>R(40)COV10
52 CON 999          ;A999
53 STO 30           ;A999=>R(30)I      I = STEPTI => R(30)
54 GSB UPDATE;<=UPDATE;      **** UPDATE ****
55 END             ;<=END
```

4.1 Description du sous-programme RINGB

Ce programme permet de conserver des données FILTEREES dans une mémoire dite circulaire (ring-buffer). Les données d'une même ligne correspondent à un seul temps de lecture. Le programme écrit dans le fichier de données à chaque fois qu'il est exécuté. L'intervalle des écritures est donc déterminé dans le programme appelant. A chaque début de page, le programme écrit dans un fichier index la page et la ligne en cours et le jour de l'année. La ligne 1 de chaque page du fichier de données correspond à 0h00. Il s'agit du temps de la lecture et non de celui correspondant au moment où la valeur est filtrée. Ainsi, si l'intervalle des écritures est de 20 minutes, la valeur filtrée émise à 0h20 sera inscrite à la ligne 1. Notons que l'intervalle des écritures peut ne pas être égal au délai en temps. Par exemple, il peut y avoir une écriture par heure avec un délai réel de 20 minutes. Dans ce cas, la valeur filtrée émise à 0h20 sera inscrite à la ligne 1 seulement à 1h00.

Appels de sous-routines

Aucun

Contenu des registres

=====

Lecture seulement	Lecture et écriture	Ecriture seulement
R(13) = MMMM	R(12) = START	R(15) = NREC
R(18) = PMAX	R(14) = JJJ	
R(19) = NC	R(16) = PLLLL	
	R(17) = DRAP	
	R(51) = mois	
	R(53) = PAGE courante	
	R(54) = Compteur de canaux	
	R(55) = Valeur d'un canal	
	R(56) = LIGNE courante	

Fichiers

Le programme a accès à deux fichiers dont les noms se trouvent dans les registres S et T. Le registre S contient le nom du fichier de données et le registre T contient le nom du fichier index. Le fichier de données doit déjà être ouvert au moment de l'appel. Si le fichier index est ouvert durant l'exécution du programme RINGB, le fichier de données sera ré-ouvert avant de sortir. Les registres S et T ne sont pas modifiés.

Sous-programme RINGB

```

1 CON 0 ;A0
2 STO 54 ;A0=>R(54) Compteur de canaux
3 RCL 16 ;A(PPLLL) Contient la valeur 1000*PAGE+LIGNE
4 CON 1000 ;A(1000)=>B(PPLLL)
5 DIV ;A(PP.LLL)
6 INT ;A(PAGE)
7 STO 53 ;A(PAGE)=>R(53) Page selon PPLLL
8 RCL 13 ;A(MMMM) Nombre de min. depuis 0h00
9 JNZ 11 ;=>OK ;IF(MMMM.NE.0) GOTO OK
10 CON 1440 ;A(1440)=>B(0) 24 heures = 1440 minutes
11 CON 60 ;<=OK ;A(60)=>B(MMMM)=>C(0) Intervale des écrit. en min.
12 DIV ;A(MMMM/60)
13 STO 56 ;A(LIGNE)=>R(56) Ligne courante exacte
14 RCL 17 ;<=DRAP ;A(DRAP)
15 JNZ 20 ;=>TIME ;IF(DRAP.NE.0) GOTO TIME
16 POP ;A(LIGNE)
17 CON 1 ;A(1)=>B(LIGNE)
18 SUB ;A(LIGNE-1)
19 JNZ 51 ;=>FIN ;IF(LIGNE.NE.1) GOTO FIN Première écriture du jour
20 TIM ;<=TIME ;As,B(hr_min),C(mois_jour)
21 POP ;A(hr_min),B(mois_jour)
22 POP ;A(mois_jour)
23 PSH ;A(mois_jour)=>B(mois_jour)
24 CON 100 ;A(100)=>B(mois_jour)=>C(mois_jour)
25 DIV ;A(mois_jour/100),B(mois_jour)
26 INT ;A(mois),B(mois_jour)
27 STO 51 ;A(mois)=>R(51)mois
28 CON 2 ;A2=>B(mois)=>C(mois_jour)
29 SUB ;A(mois-2),B(mois_jour)
30 JGZ 40 ;=>MARS ;IF (mois.GT.2) GOTO MARS
31 POP ;A(mois_jour)
32 CON 69 ;A69=>B(mois_jour)
33 RCL 51 ;A(mois)=>B69=>C(mois_jour)
34 MLT ;A(69*mois),B(mois_jour)
35 SUB ;A(mois_jour-69*mois)
36 CON 31 ;A31=>B(mois_jour-69*mois)
37 SUB ;A(mois_jour-69*mois-31)
38 STO 14 ;A(JJJ)=>R(14)JJJ
39 JMP 51 ;=>FIN ;GOTO FIN
40 POP ;<=MARS ;A(mois_jour)
41 RCL 51 ;A(mois)=>B(mois_jour)
42 CON 30.6 ;A(30.6)=>B(mois)=>C(mois_jour)
43 MLT ;A(mois*30.6),B(mois_jour)
44 CON 32.3999 ;A(32.3999)=>B(mois*30.6)=>C(mois_jour)
45 SUB ;A(mois*30.6-32.3999),B(mois_jour)
46 XAB ;A(mois_jour)<=>B(mois*30.6-32.3999)
47 MOD 100 ;A(jour),B(mois*30.6-32.3999)
48 ADD ;A(mois*30.6-32.3999+jour)
49 INT ;METTRE 31.3999 POUR UNE ***** ANNEE BISSEXTILE *****
50 STO 14 ;A(JJJ)=>R(14)JJJ

```

```

51 RCL 17      ;<=FIN      ;A(DRAP)                Drapeau
52 JPZ 67      ;=>SHOW    ;IF(DRAP.NE.0) GOTO SHOW Est-ce une première fois
53 RCL 56      ;<=PPLLL   ;A(LIGNE)
54 RCL 16      ;A(PPLLL)=>B(LIGNE)
55 MOD 1000    ;A(LLL) =>B(LIGNE)
56 SUB        ;A(LIGNE-LLL)
57 JPS 72      ;=>DRAP    ;IF (LIGNE.GE.LLL) GOTO DRAP
58 ICR 53      ;R(53) = R(53) + 1
59 RCL 53      ;A(PAGE)
60 RCL 18      ;A(PMAX)=>B(PAGE)
61 XAB        ;A(PAGE)<=>B(PMAX)
62 SUB        ;A(PMAX-PAGE)
63 JPS 72      ;=>DRAP    ;IF (PMAX.GE.PAGE) GOTO DRAP
64 CON 1      ;A(1)
65 STO 53      ;A(1)=>R(53)
66 JMP 72
67 CON 1      ;<=SHOW    ;A1
68 RCL 56      ;A(LIGNE)=>B1
69 SUB        ;A(1-LIGNE)
70 JNZ 95      ;=>WRITE   ;IF(LIGNE.NE.1) GOTO WRITE Est-ce un début de page?
71 ICR 53      ;R(53)<=R(53)+1      PAGE = PAGE + 1
72 CON 0      ;<=DRAP    ;A0
73 STO 17      ;A0=>R(17)0        Baisse le drapeau
74 XST        ;S(INDEX)<=>T(DATA)
75 SWF        ;Ouvre le fichier de l'INDEX
76 CON 1      ;A1
77 RCL 53      ;A(PAGE)=>B1      N.B. PAGE devient LIGNE de INDEX
78 CON 1      ;A1=>B(PAGE)=>C1
79 SLC        ;Place le pointeur en C(1),L(PAGE),P(1)
80 POP        ;A(PAGE)
81 CON 1000   ;A1000=>B(PAGE)
82 MLT        ;A(1000*PAGE)
83 RCL 56      ;A(LIGNE)=>B(1000*PAGE)
84 ADD        ;A(1000*PAGE + LIGNE)
85 STF        ;A(PPLLL)=>Le fichier de l'INDEX
86 RCL 12      ;A(START)=>B(PPLLL)
87 JNZ 90      ;=>ICP    ;IF(START.NE.0) GOTO ICP
88 POP        ;A(PPLLL)
89 STO 12      ;A(PPLLL)=>R(12)
90 ICP        ;<=ICP    ;C(2),L(PAGE),P(1)
91 RCL 14      ;A(JJJ)                Jour de l'année
92 STF        ;A(JJJ)=>Le fichier de l'INDEX
93 XST        ;S(DATA)<=>T(INDEX)
94 SWF        ;Ouvre le fichier de données.
95 RCI 54      ;<=WRITE   ;Ax0;Ax1;...;AxNC
96 STO 55      ;AxN=>R(55)                Valeur d'un canal
97 ICR 54      ;R(54)I+1                Compteur de colonne
98 RCL 53      ;A(PAGE)
99 RCL 56      ;A(LIGNE)=>B(PAGE)
100 RCL 54     ;A(I),B(LIGNE),C(PAGE)

```

```

101 SLC ;Place le pointeur en C(I),L(LIGNE),P(PAGE)
102 RCL 55 ;AxN=>B(I) Valeur d'un canal
103 STF ;AxN=>Le fichier de données
104 ICP ;C(I+1),L(LIGNE),P(PAGE)
105 POP ;AI
106 RCL 19 ;A(NC)=>BI N= Nombre de canaux actifs
107 SUB ;A(I-NC)
108 JLZ 95 ;=>WRITE ;IF(I.LT.NC) GOTO WRITE
109 RLC ;A(COLONNE),B(LIGNE),C(PAGE) Du fichier de données
110 POP ;A(LIGNE),B(PAGE)
111 POP ;A(PAGE)
112 RCL 18 ;A(PMAX)=>B(PAGE) Nombre limite de pages
113 XAB ;A(PAGE)<=>B(PMAX)
114 SUB ;A(PAGE-PMAX)
115 JPS 120 ;=>OPLA ;IF(PMAX.GE.PAGE) GOTO OPLA Limite de pages = PMAX
116 CON 0 ;A(0) Page 0
117 STO 53 ;A(0)=>R(53) Le ring-buffer est bouclé.
118 CON 1 ;A(1)
119 STO 56 ;A(1)=>R(56) Ligne 1
120 RCL 53 ;<=OPLA ;A(PAGE)
121 CON 1000 ;A1000=>B(PAGE)
122 MLT ;A(1000*PAGE)
123 RCL 56 ;A(LIGNE)=>B(1000*PAGE)
124 ADD ;A(1000*PAGE+LIGNE)
125 STO 16 ;A(PPLLL)=>R(16) Servira pour une autre itération
126 ICR 15 ;R(15)=R(15)+1 Compteur du nombre des écritures
127 END

```

4.2 Description du sous-programme PREDIC

Ce programme prédit les prochaines valeurs du vecteur d'état et du vecteur de covariance. Ce programme assure la continuité des résultats sans être directement dépendant des données lues. Ainsi PREDIC sera exécuté dans tous les cas, c'est-à-dire même si une donnée brute ne permet pas l'appel de UPDATE.

Appels de sous-routines

Aucun

Contenu des registres

Lecture seulement	Lecture et écriture	Écriture seulement
R(20) = DELAY	R(31) = C1	Aucun
R(24) = OBVARP	R(32) = C2	
R(25) = DECAY	R(33) = C3	
R(26) = DELSQ	R(34) = C4	
	R(35) = C5	
	R(36) = C6	
	R(37) = C7	
	R(38) = C8	
	R(39) = C9	
	R(45) = S1	
	R(46) = S2	
	R(47) = S3	

Fichiers

Aucun

Sous-programme PREDIC

```

1 RCL 45 ;A(S1)
2 RCL 46 ;A(S2)=>B(S1)
3 RCL 20 ;A(DELAY)=>B(S2)=>C(S1)
4 DIV ;A(S2/DELAY),B(S1)
5 ADD ;A(S1+S2/DELAY)
6 RCL 26 ;A(DELSQ)=>B(S1+S2/DELAY)
7 RCL 47 ;A(S3)=>B(DELSQ)=>C(S1+S2/DELAY)
8 MLT ;A(DELSQ*S3),B(S1+S2/DELAY)
9 ADD ;A(S1+S2/DELAY+DELSQ*S3)
10 STO 45 ;A(S1)=>R(45)S1 STATE(1)
11 RCL 46 ;A(S2)
12 RCL 47 ;A(S3)=>B(S2)
13 RCL 20 ;A(DELAY)=>B(S3)=>C(S2)
14 DIV ;A(S3/DELAY),B(S2)
15 CON 2 ;A(2)=>B(S3/DELAY)=>C(S2)
16 MLT ;A(2*S3/DELAY),B(S2)
17 ADD ;A(S2+2*S3/DELAY)
18 STO 46 ;A(S2)=>R(46)S2 STATE(2)
19 RCL 47 ;A(S3)
20 RCL 25 ;A(DECAY)=>B(S3)
21 MLT ;A(S3*DECAY)
22 STO 47 ;A(S3)=>R(47)S3 STATE(3)
23 RCL 31 ;A(C1)
24 RCL 32 ;A(C2)=>B(C1)
25 RCL 20 ;A(DELAY)=>B(C2)=>C(C1)
26 DIV ;A(C2/DELAY),B(C1)
27 CON 2 ;A(2)=>B(C2/DELAY)=>C(C1)
28 MLT ;A(2*C2/DELAY),B(C1)
29 ADD ;A(C1+2*C2/DELAY)
30 RCL 33 ;A(C3)=>B(C1+2*C2/DELAY)
31 CON 2 ;A(2)=>B(C3)=>C(C1+2*C2/DELAY)
32 RCL 34 ;A(C4)=>B(2)=>C(C3)=>D(C1+2*C2/DELAY)
33 MLT ;A(2*C4),B(C3),C(C1+2*C2/DELAY)
34 ADD ;A(C3+2*C4),B(C1+2*C2/DELAY)
35 CON 2 ;A(2)=>B(C3+2*C4)=>C(C1+2*C2/DELAY)
36 RCL 35 ;A(C5)=>B(2)=C(C3+2*C4)=>D(C1+2*C2/DELAY)
37 MLT ;A(2*C5),B(C3+2*C4),C(C1+2*C2/DELAY)
38 RCL 20 ;A(DELAY)=>B(2*C5)=>C(C3+2*C4)=>D(C1+2*C2/DELAY)
39 DIV ;A(2*C5/DELAY),B(C3+2*C4),C(C1+2*C2/DELAY)
40 ADD ;A(C3+2*C4+2*C5/DELAY),B(C1+2*C2/DELAY)
41 RCL 26 ;A(DELSQ)=>B(C3+2*C4+2*C5/DELAY)=>C(C1+2*C2/DELAY)
42 RCL 36 ;A(C6)=>B(DQ)=>C(C3+2*C4+2*C5/DELAY)=>D(C1+2*C2/DELAY)
43 MLT ;A(DELSQ*C6),B(C3+2*C4+2*C5/DELAY),C(C1+2*C2/DELAY)
44 ADD ;A(C3+2*C4+2*C5/DELAY+DELSQ*C6),B(C1+2*C2/DELAY)
45 RCL 26 ;A(DQ)=>B(C3+2*C4+2*C5/DELAY+DQ*C6)=>C(C1+2*C2/DELAY)
46 MLT ;A(DQ*(C3+2*C4+2*C5/DELAY+DQ*C6)),B(C1+2*C2/DELAY)
47 ADD ;A(C1+2*C2/DELAY+DQ*(C3+2*C4+2*C5/DELAY+DQ*C6))
48 STO 31 ;A(C1)=R(31)C1 COV(1)
49 RCL 32 ;A(C2)
50 RCL 33 ;A(C3)=>B(C2)

```

```

51 CON 2 ;A(2)=>B(C3)=>C(C2)
52 RCL 34 ;A(C4)=>B(2)=>C(C3)=>D(C2)
53 MLT ;A(2*C4),B(C3),C(C2)
54 ADD ;A(C3+2*C4),B(C2)
55 RCL 20 ;A(DELAY)=>B(C3+2*C4)=>C(C2)
56 DIV ;A((C3+2*C4)/DELAY),B(C2)
57 ADD ;A(C2+(C3+2*C4)/DELAY)
58 CON 3 ;A(3)=>B(C2+(C3+2*C4)/DELAY)
59 RCL 35 ;A(C5)=>B(3)=>C(C2+(C3+2*C4)/DELAY)
60 MLT ;A(3*C5),B(C2+(C3+2*C4)/DELAY)
61 CON 2 ;A(2)=>B(3*C5)=>C(C2+(C3+2*C4)/DELAY)
62 RCL 36 ;A(C6)=>B(2)=>C(3*C5)=>D(C2+(C3+2*C4)/DELAY)
63 MLT ;A(2*C6),B(3*C5),C(C2+(C3+2*C4)/DELAY)
64 RCL 20 ;A(DELAY)=>B(2*C6)=>C(3*C5)=>C(C2+(C3+2*C4)/DELAY)
65 DIV ;A(2*C6/DELAY),B(3*C5),C(C2+(C3+2*C4)/DELAY)
66 ADD ;A(3*C5+2*C6/DELAY),B(C2+(C3+2*C4)/DELAY)
67 RCL 26 ;A(DQ)=>B(3*C5+2*C6/DELAY)=>C(C2+(C3+2*C4)/DELAY)
68 MLT ;A(DQ*(3*C5+2*C6/DELAY)),B(C2+(C3+2*C4)/DELAY)
69 ADD ;A(C2+(C3+2*C4)/DELAY+DELSQ*(3*C5+2*C6/DELAY))
70 STO 32 ;A(C2)=>R(32)C2 COV(2)
71 RCL 33 ;A(C3)
72 RCL 35 ;A(C5)=>B(C3)
73 RCL 20 ;A(DELAY)=>B(C5)=>C(C3)
74 DIV ;A(C5/DELAY),B(C3)
75 RCL 26 ;A(DELSQ)=>B(C5/DELAY)=>C(C3)
76 RCL 36 ;A(C6)=>B(DELSQ)=>C(C5/DELAY)=>D(C3)
77 MLT ;A(DELSQ*C6),B(C5/DELAY),C(C3)
78 ADD ;A(C5/DELAY+DELSQ*C6),B(C3)
79 CON 4 ;A(4)=>B(C5/DELAY+DELSQ*C6)=>C(C3)
80 MLT ;A(4*(C5/DELAY+DELSQ*C6)),B(C3)
81 ADD ;A(C3+4*(C5/DELAY+DELSQ*C6))
82 STO 33 ;A(C3)=>R(33)C3 COV(3)
83 RCL 25 ;A(DECAY)
84 RCL 34 ;A(C4)=>B(DECAY)
85 RCL 35 ;A(C5)=>B(C4)=>C(DECAY)
86 RCL 20 ;A(DELAY)=>B(C5)=>C(C4)=>D(DECAY)
87 DIV ;A(C5/DELAY),B(C4),C(DECAY)
88 ADD ;A(C4+C5/DELAY),B(DECAY)
89 RCL 26 ;A(DELSQ)=>B(C4+C5/DELAY)=>C(DECAY)
90 RCL 36 ;A(C6)=>B(DELSQ)=>C(C4+C5/DELAY)=>D(DECAY)
91 MLT ;A(DELSQ*C6),B(C4+C5/DELAY),C(DECAY)
92 ADD ;A(C4+C5/DELAY+DELSQ*C6),B(DECAY)
93 MLT ;A(DECAY*(C4+C5/DELAY+DELSQ*C6))
94 STO 34 ;A(C4)=>R(34)C4 COV(4)
95 RCL 35 ;A(C5)
96 CON 2 ;A(2)=>B(C5)
97 RCL 36 ;A(C6)=>B(2)=>C(C5)
98 RCL 20 ;A(DELAY)=>B(C6)=>C(2)=>D(C5)
99 DIV ;A(C6/DELAY),B(2),C(C5)
100 MLT ;A(2*C6/DELAY),B(C5)

```

```

101 ADD ;A(C5+2*C6/DELAY)
102 RCL 25 ;A(DECAY)=>B(C5+2*C6/DELAY)
103 MLT ;A(DECAY*(C5+2*C6/DELAY))
104 STO 35 ;A(C5)=>R(35)C5 COV(5)
105 RCL 36 ;A(C6)
106 RCL 25 ;A(DECAY)=>B(C6)
107 PSH ;A(DECAY)=>B(DECAY)=>C(C6)
108 MLT ;A(DECAY*DECAY),B(C6)
109 MLT ;A(DECAY*DECAY*C6)
110 RCL 24 ;A(OBVARP)=>B(DECAY*DECAY*C6)
111 ADD ;A(DECAY*DECAY*C6+OBVARP)
112 STO 36 ;A(C6)=>R(36) COV(6)
113 RCL 37 ;A(C7)
114 RCL 38 ;A(C8)=>B(C7)
115 RCL 20 ;A(DELAY)=>B(C8)=>C(C7)
116 DIV ;A(C8/DELAY),B(C7)
117 ADD ;A(C7+C8/DELAY)
118 RCL 26 ;A(DELSQ)=>B(C7+C8/DELAY)
119 RCL 39 ;A(C9)=>B(DELSQ)=>C(C7+C8/DELAY)
120 MLT ;A(DELSQ*C9),B(C7+C8/DELAY)
121 ADD ;A(C7+C8/DELAY+DELSQ*C9)
122 STO 37 ;A(C7)=>R(37)C7 COV(7)
123 RCL 38 ;A(C8)
124 CON 2 ;A(2)=>B(C8)
125 RCL 39 ;A(C9)=>B(2)=>C(C8)
126 RCL 20 ;A(DELAY)=>B(C9)=>C(2)=>D(C8)
127 DIV ;A(C9/DELAY),B(2),C(C8)
128 MLT ;A(2*C9/DELAY),B(C8)
129 ADD ;A(C8+2*C9/DELAY)
130 STO 38 ;A(C8)=>R(38)C8 COV(8)
131 RCL 25 ;A(DECAY)
132 RCL 39 ;A(C9)=>B(DECAY)
133 MLT ;A(DECAY*C9)
134 STO 39 ;A(C9)=>R(39) COV(9)
135 END

```

4.3 Description du sous-programme RESTEP

Ce programme sert à ramener le système dans un état sans saut. La valeur accumulée dans S4 est rajoutée à S1 et S4 est remis à zéro. S4 demeurera nul jusqu'à ce qu'un nouveau saut soit détecté.

Appels de sous-routines

Aucun

Contenu des registres

Lecture seulement	Lecture et écriture	Écriture seulement
Aucun	R(31) = C1 R(32) = C2 R(34) = C4 R(37) = C7 R(38) = C8 R(39) = C9 R(40) = C10 R(45) = S1 R(48) = S4	R(30) = STEPTI

Fichiers

Aucun

Sous-programme RESTEP

1 RCL 45	;A(S1)	
2 RCL 48	;A(S4)=>B(S1)	
3 ADD	;A(S1+S4)	
4 STO 45	;A(S1)=>R(45)S1	STATE(1)
5 CON 0	;A(0)	
6 STO 48	;A(0)=>R(48)S4	STATE(4)
7 RCL 31	;A(C1)	
9 CON 2	;A(2)=>B(C1)	
8 RCL 37	;A(C7)=>B(2)=>C(C1)	
10 MLT	;A(2*C7),B(C1)	
11 ADD	;A(C1+2*C7)	
12 RCL 40	;A(C10)=>B(C1+2*C7)	
13 ADD	;A(C1+2*C7+C10)	
14 STO 31	;A(C1)=>R(31)C1	COV(1)
15 RCL 32	;A(C2)	
16 RCL 38	;A(C8)=>B(C2)	
17 ADD	;A(C2+C8)	
18 STO 32	;A(C2)=>R(32)C2	COV(2)
19 RCL 34	;A(C4)	
20 RCL 39	;A(C9)=>B(C4)	
21 ADD	;A(C4+C9)	
22 STO 34	;A(C4)=>R(34)C4	COV(4)
23 CON 0	;A(0)	
24 STO 37	;A(0)=>R(37)0	COV(7)
25 STO 38	;A(0)=>R(38)0	COV(8)
26 STO 39	;A(0)=>R(39)0	COV(9)
27 STO 40	;A(0)=>R(40)0	COV(10)
28 CON 999	;A(999)	
29 STO 30	;A(999)=>R(30)I	STEPTI
30 END		

4.4 Description du sous-programme UPDATE

Ce programme construit le vecteur des coefficients en utilisant les valeurs du vecteur de covariance prédites par PREDIC et remet à jour le vecteur d'état et le vecteur de covariance à partir de la nouvelle donnée brute. Lorsque le début d'un saut a été décelé dans KAF, ce sous-programme ne sera pas exécuté lors de la première itération. Pour le reste de la durée du saut, UPDATE utilisera la valeur que RESID aurait s'il n'y avait pas de saut. C'est ainsi que les valeurs estimées par le filtre ne seront pas affectées par un saut d'échelle ou toute interférence dont la durée ne dépasse pas le délai fixé.

Appels de sous-routines

Aucun

Contenu des registres

Lecture seulement	Lecture et écriture	Ecriture seulement
R(22) = OBSVAR	R(28) = RHO	Aucun
R(27) = RESID	R(29) = RESD	
	R(31) = C1	
	R(32) = C2	
	R(33) = C3	
	R(34) = C4	
	R(35) = C5	
	R(36) = C6	
	R(37) = C7	
	R(38) = C8	
	R(39) = C9	
	R(40) = C10	
	R(41) = P1	
	R(42) = P2	
	R(43) = P3	
	R(44) = P4	
	R(45) = S1	
	R(46) = S2	
	R(47) = S3	
	R(48) = S4	

Fichiers

Aucun

Sous-programme UPDATE

```

1 RCL 31      ;A(C1)
2 RCL 32      ;A(C2)=>B(C1)
3 ADD        ;A(C1+C2)
4 RCL 34      ;A(C4)=>B(C1+C2)
5 ADD        ;A(C1+C2+C4)
6 RCL 37      ;A(C7)=>B(C1+C2+C4)
7 ADD        ;A(C1+C2+C4+C7)
8 STO 41      ;A(P1)=>R(41)P1      PH(1)
9 RCL 32      ;A(C2)
10 RCL 33     ;A(C3)=>B(C2)
11 ADD       ;A(C2+C3)
12 RCL 35     ;A(C5)=>B(C2+C3)
13 ADD       ;A(C2+C3+C5)
14 RCL 38     ;A(C8)=>B(C2+C3+C5)
15 ADD       ;A(C2+C3+C5+C8)
16 STO 42     ;A(P1)=>R(42)P1      PH(2)
17 RCL 34     ;A(C4)
18 RCL 35     ;A(C5)=>B(C4)
19 ADD       ;A(C4+C5)
20 RCL 36     ;A(C6)=>B(C4+C5)
21 ADD       ;A(C4+C5+C6)
22 RCL 39     ;A(C9)=>B(C4+C5+C6)
23 ADD       ;A(C4+C5+C6+C9)
24 STO 43     ;A(P3)=>R(43)P3      PH(3)
25 RCL 37     ;A(C7)
26 RCL 38     ;A(C8)=B(C7)
27 ADD       ;A(C7+C8)
28 RCL 39     ;A(C9)=>B(C7+C8)
29 ADD       ;A(C7+C8+C9)
30 RCL 40     ;A(C10)=>B(C7+C8+C9)
31 ADD       ;A(C7+C8+C9+C10)
32 STO 44     ;A(P4)=>R(44)P4      PH(4)
33 RCL 22     ;A(OBSVAR)
34 RCL 41     ;A(P1)=>B(OBSVAR)
35 ADD       ;A(OBSVAR+P1)
36 RCL 42     ;A(P2)=>B(OBSVAR+P1)
37 ADD       ;A(OBSVAR+P1+P2)
38 RCL 43     ;A(P3)=>B(OBSVAR+P1+P2)
39 ADD       ;A(OBSVAR+P1+P2+P3)
40 RCL 44     ;A(P4)=>B(OBSVAR+P1+P2+P3)
41 ADD       ;A(OBSVAR+P1+P2+P3+P4)
42 STO 28     ;A(RHO)=>R(28)RHO      RHO
43 RCL 27     ;A(RESID)=>B(RHO)
44 XAB       ;A(RHO)<=>B(RESID)
45 DIV       ;A(RESID/RHO)
46 STO 29     ;A(RESD)=>R(29)RESD      RESD
47 RCL 41     ;A(P1)=>B(RESD)
48 MLT       ;A(RESD*P1)
49 RCL 45     ;A(S1)=>B(RESD*P1)
50 ADD       ;A(RESD*P1+S1)

```

```

51 STO 45      ;A(S1)=>R(45)S1      STATE(1)
52 RCL 29      ;A(RESD)
53 RCL 42      ;A(P2)=>B(RESD)
54 MLT        ;A(RESD*P2)
55 RCL 46      ;A(S2)=>B(RESD*P2)
56 ADD        ;A(RESD*P2+S2)
57 STO 46      ;A(S2)=>R(46)S2      STATE(2)
58 RCL 29      ;A(RESD)
59 RCL 43      ;A(P3)=>B(RESD)
60 MLT        ;A(RESD*P3)
61 RCL 47      ;A(S3)=>B(RESD*P3)
62 ADD        ;A(RESD*P3+S3)
63 STO 47      ;A(S3)=>R(47)S3      STATE(3)
64 RCL 29      ;A(RESD)
65 RCL 44      ;A(P4)=>B(RESD)
66 MLT        ;A(RESD*P4)
67 RCL 48      ;A(S4)=>B(RESD*P4)
68 ADD        ;A(RESD*P4+S4)
69 STO 48      ;A(S4)=>R(48)S4      STATE(4)
70 RCL 31      ;A(C1)
71 RCL 41      ;A(P1)=>B(C1)
72 PSH        ;A(P1)=>B(P1)=>C(C1)
73 MLT        ;A(P1*P1),B(C1)
74 RCL 28      ;A(RHO)=>B(P1*P1)=>C(C1)
75 DIV        ;A(P1*P1/RHO),B(C1)
76 SUB        ;A(C1-P1*P1/RHO)
77 STO 31      ;A(C1)=>R(31)C1      COV(1)
78 RCL 32      ;A(C2)
79 RCL 41      ;A(P1)=>B(C2)
80 RCL 42      ;A(P2)=>B(P1)=>C(C2)
81 MLT        ;A(P1*P2),B(C2)
82 RCL 28      ;A(RHO)=>B(P1*P2)=>C(C2)
83 DIV        ;A(P1*P2/RHO),B(C2)
84 SUB        ;A(C2-P1*P2/RHO)
85 STO 32      ;A(C2)=>R(32)C2      COV(2)
86 RCL 33      ;A(C3)
87 RCL 42      ;A(P2)=>B(C3)
88 PSH        ;A(P2)=>B(P2)=>C(C3)
89 MLT        ;A(P2*P2),B(C3)
90 RCL 28      ;A(RHO)=>B(P2*P2)=>C(C3)
91 DIV        ;A(P2*P2/RHO),B(C3)
92 SUB        ;A(C3-P2*P2/RHO)
93 STO 33      ;A(C3)=>R(33)C3      COV(3)
94 RCL 34      ;A(C4)
95 RCL 41      ;A(P1)=>B(C4)
96 RCL 43      ;A(P3)=>B(P1)=>C(C4)
97 MLT        ;A(P1*P3),B(C4)
98 RCL 28      ;A(RHO)=>B(P1*P3)=>C(C4)
99 DIV        ;A(P1*P3/RHO),B(C4)
100 SUB       ;A(C4-P1*P3/RHO)

```

```

101 STO 34      ;A(C4)=>R(34)C4      COV(4)
102 RCL 35      ;A(C5)
103 RCL 42      ;A(P2)=>B(C5)
104 RCL 43      ;A(P3)=>B(P2)=>C(C5)
105 MLT         ;A(P2*P3), B(C5)
106 RCL 28      ;A(RHO)=>B(P2*P3)=>C(C5)
107 DIV         ;A(P2*P3/RHO), B(C5)
108 SUB         ;A(C5-P2*P3/RHO)
109 STO 35      ;A(C5)=>R(35)C5      COV(5)
110 RCL 36      ;A(C6)
111 RCL 43      ;A(P3)=>B(C6)
112 PSH         ;A(P3)=>B(P3)=>C(C6)
113 MLT         ;A(P3*P3), B(C6)
114 RCL 28      ;A(RHO)=>B(P3*P3)=>C(C6)
115 DIV         ;A(P3*P3/RHO), B(C6)
116 SUB         ;A(C6-P3*P3/RHO)
117 STO 36      ;A(C6)=>R(36)C6      COV(6)
118 RCL 37      ;A(C7)
119 RCL 41      ;A(P1)=>B(C7)
120 RCL 44      ;A(P4)=>B(P1)=>C(C7)
121 MLT         ;A(P1*P4), B(C7)
122 RCL 28      ;A(RHO)=>B(P1*P4)=>C(C7)
123 DIV         ;A(P1*P4/RHO), B(C7)
124 SUB         ;A(C7-P1*P4/RHO)
125 STO 37      ;A(C7)=>R(37)C7      COV(7)
126 RCL 38      ;A(C8)
127 RCL 42      ;A(P2)=>B(C8)
128 RCL 44      ;A(P4)=>B(P2)=>C(C8)
129 MLT         ;A(P2*P4), B(C8)
130 RCL 28      ;A(RHO)=>B(P2*P4)=>C(C8)
131 DIV         ;A(P2*P4/RHO), B(C8)
132 SUB         ;A(C8-P2*P4/RHO)
133 STO 38      ;A(C8)=>R(38)C8      COV(8)
134 RCL 39      ;A(C9)
135 RCL 43      ;A(P3)=>B(C9)
136 RCL 44      ;A(P4)=>B(P3)=>C(C9)
137 MLT         ;A(P3*P4), B(C9)
138 RCL 28      ;A(RHO)=>B(P3*P4)=>C(C9)
139 DIV         ;A(P3*P4/RHO), B(C9)
140 SUB         ;A(C9-P3*P4/RHO)
141 STO 39      ;A(C9)=>R(39)C9      COV(9)
142 RCL 40      ;A(C10)
143 RCL 44      ;A(P4)=>C(10)
144 PSH         ;A(P4)=>B(P4)=>C(C10)
145 MLT         ;A(P4*P4), B(C10)
146 RCL 28      ;A(RHO)=>B(P4*P4)=>C(C10)
147 DIV         ;A(P4*P4/RHO), B(C10)
148 SUB         ;A(C10-P4*P4/RHO)
149 STO 40      ;A(C10)=R(40)C10     COV(10)
150 END

```

5.0 Description du programme XMT

Ce programme permet de transférer un fichier de données du Polycorder qui a été créé par le programme RINGB vers tout médium capable de recevoir de l'information sous forme ASCII. Le protocole de communication peut être modifié selon les exigences du terminal receveur. Le temps de transfert d'un fichier dépend du nombre de colonnes "C" et du nombre de lignes "L" à transférer. Selon le protocole accepté dans la description suivante (9600 Bauds), le temps de transfert "t", en secondes, peut être calculé à l'aide de l'expression suivante:

$$t = 3. + 0.2 * C + (0.02 + 0.09 * C) * L$$

PROTOCOLE

Le Polycorder doit être à :

Baud	: 9600	INP LINE	: <- <CTL/013>
Parity	: E	INP MAX	: 80
Duplex	: F	XMT DY	: 1
XMT	: N	CHECKSUM	: N
Line req.	: A	END OF LINE	: <- <CTL/013>
Repeat req.	: B	END OF PAGE	: <- <CTL/013>
Cont. req.	: C	END OF FILE	: <-I/ <CTL/013 CTL/010>
Stop req.	: D	MATING CALL	: A

Description des variables

DRAP = Un drapeau est levé dans le registre 17 pour indiquer que le programme vient d'être exécuté.

PMAX = C'est le nombre limite de pages de la mémoire circulaire entretenue par le programme RINGB. Cette valeur servira à calculer LDF.

NC = Nombre de canaux actifs. Cette valeur sera affichée durant l'exécution du programme.

NREC = Le registre 15 contient le nombre de lignes qui ont été entrées dans le fichier de données depuis le dernier transfert. Le programme utilise cette valeur pour exécuter adéquatement le transfert. Le registre 15 est remis à zéro après l'exécution de XMT.

L0 = Ligne de départ du fichier index.

L1 = Ligne de la fin du fichier index.

LDEB = Nombre de lignes à transférer au début du fichier de données.

LDF = Nombre limite de lignes dans le fichier de données.

LFIN = Nombre de lignes à transférer à la fin du fichier de données.

LPP = Nombre de lignes par page du fichier de données.

PPLLL = Cette valeur représente la position de la dernière entrée selon l'expression $1000 * \text{PAGE} + \text{LIGNE}$.

START = Cette valeur, qui est contenue dans le registre 12, sera remise à zéro à chaque fois que XMT sera exécuté. Cela permettra au programme RINGB de déterminer si XMT vient d'être exécuté.

Appels de sous-routines

Aucun

Contenu des registres

Lecture seulement	Lecture et écriture	Écriture seulement
R(16) = PPLLL	R(12) = START	R(17) = DRAP
R(18) = PMAX	R(15) = NREC	
R(19) = NC	R(49) = LPPR	
	R(50) = LDF	
	R(51) = Compteur de lignes	
	R(53) = LDEB	
	R(54) = LFIN	
	R(55) = L1	
	R(56) = L0	

Fichiers

Le programme transfère deux fichiers. Il s'agit du fichier index qui contient l'information sur le fichier de données et le fichier de données lui-même. Seules les données qui ont été inscrites depuis le dernier transfert seront transférées. Les fichiers demeurent intacts après un transfert. Cela permet la récupération des données si le transfert ne s'est pas effectué correctement. Si une telle situation se présentait, il faudrait entrer dans le registre 12 la valeur PPLLL de la première ligne à transférer du fichier index et mettre 1 dans le registre 15 avant de faire exécuter XMT à nouveau. Si la première ligne à transférer du fichier index est inconnue, on peut transférer les deux fichiers au complet en mettant 2 dans le registre 15 et la valeur du registre 16 dans le registre 12. Pour éviter ce genre de contretemps, il est bon de suivre rigoureusement les instructions de transfert des données et de s'assurer que le protocole de communication du Polycorder est compatible avec celui du médium récepteur. Prendre note de la valeur contenue dans le registre 12 avant de faire exécuter XMT.

5.1 Programme XMT

```

1 CON 1440           ;A(1440)
2 CON 60             ;A(60)=>B(1440)
3 DIV               ;A(1440/60)
4 STO 49            ;A(LPP)=>R(49)           Lignes par page
5 RCL 18            ;A(PMAX)=>B(LPP)
6 MLT              ;A(LPP*PMAX)
7 STO 50            ;A(LDF)=>R(50)       Lignes totales dans le fichier
8 RCL 15            ;A(NREC)           Nombre d'écritures selon R(15)
9 JPZ 163          ;=>END ;IF(NREC.EQ.0) GOTO END
10 RCL 16           ;A(PPLLL)
11 CON 1000         ;A(1000)=>B(PPLLL)
12 DIV             ;A(PP.LLL)
13 INT             ;A(P1)
14 STO 55          ;A(P1)=>R(55)       L1 pour JOUR1
15 RCL 16           ;A(PPLLL)=>B(P1)
16 MOD 1000        ;A(L1)=>B(P1)
17 CON 1            ;A(1)=>B(L1)=>C(P1)
18 ADD             ;A(L1),B(P1)
19 XAB             ;A(P1)<=>B(L1)
20 RCL 12          ;A(START)=>B(P1)=>C(L1)
21 CON 1000        ;A(1000)=>B(START)=>C(P1)=>D(L1)
22 DIV             ;A(START/1000),B(P1),C(L1)
23 INT             ;A(P0),B(P1),C(L1)
24 STO 56          ;A(P0)=>R(56)       L0 pour JOUR1
25 SUB             ;A(P1-P0),B(L1)
26 STO 51          ;A(P)=>R(51)           Pour NPAGE
27 JPS 31          ;=>GOOD ;IF(P1.GE.P0) GOTO GOOD
28 RCL 18          ;A(PMAX)=>B(P1-P0)=>C(L1)
29 ADD             ;A(P1-P0+PMAX),B(L1)
30 STO 51          ;A(P)=>R(51)           Pour NPAGE
31 RCL 49          ;<=GOOD ;A(LPP)=>B(P)=>C(L1)
32 MLT             ;A(P*LPP),B(L1)
33 ADD             ;A(P*LPP+L1)
34 RCL 12          ;A(START)=>B(P*LPP+L1)
35 MOD 1000        ;A(L0),B(P*LPP+L1)
36 SUB             ;A(P*LPP+L1-L0)
37 PSH            ;A(NREC1)=>B(NREC1)
38 RCL 15          ;A(NREC)=>B(NREC1)=>C(NREC1)
39 XAB             ;A(NREC1)<=>B(NREC),C(NREC1)
40 SUB             ;A(NREC-NREC1),B(NREC1)
41 JGZ 45          ;=>PLEIN ;IF(NREC.GT.NREC1) GOTO PLEIN
42 POP            ;A(NREC1)
43 STO 15          ;A(NREC1)=>R(15)
44 JMP 51          ;=>OK ;GOTO OK
45 RCL 50          ;<=PLEIN ;A(LDF)
46 STO 15          ;A(LDF)=>R(15)
47 RCL 18          ;A(PMAX)
48 CON 1           ;A(1)=>B(PMAX)
49 SUB             ;A(PMAX-1)
50 STO 51          ;A(P)=>R(51)           Pour NPAGE

```



```

51 CON 1      ;<=OK      ;A(1)
52 RCL 51     ;A(P)=>B(1)
53 ADD        ;A(NPAGE)
54 WID 4      ;Réserve 3 espaces
55 DCM 0      ;Aucun chiffre après le point
56 VUA        ;Affiche NPAGE
57 TRA        ;A(NPAGE)=>Transmis
58 CNS ,      ;S(,)
59 TRS        ;Transmet une virgule
60 RCL 15     ;A(NREC)
61 CDS 4,0    ;Place le curseur en case 4
62 VUA        ;Affiche NREC
63 TRA        ;A(NREC)=>Transmis
64 CNS ,      ;S(,)
65 TRS        ;Transmet une virgule
66 RCL 19     ;A(NC)
67 CDS 8,0    ;Place le curseur en case 8
68 VUA        ;Affiche NC
69 TRA        ;A(NC)=>Transmis
70 CNS ,      ;S(,)
71 TRS        ;Transmet une virgule
72 RCL 49     ;A(LPP)
73 CDS 12,0   ;Place le curseur en case 12
74 VUA        ;Affiche LPP
75 TRA        ;A(LPP)=>Transmis
76 CNS <CTL 013> ;S(CR)
77 TRS        ;Transmet un <CR>
78 CON 1      ;A(1)
79 RCL 56     ;A(L0)=>B(1)
80 CON 1      ;A(1)=>B(10)=>C(1)
81 OPN JOUR1  ;Ouvre le fichier JOUR1
82 SLC        ;Place le pointeur en (1,L0,1)
83 TRL        ;<=TRANS ;Ligne courante transmise
84 RLC        ;A(C),B(L),C(P)
85 POP        ;A(L)<=B(P)
86 RCL 55     ;A(L1)=>B(L)
87 SUB        ;A(L-L1)
88 JPZ 100    ;=>LDEB ;IF(L.EQ.L1) GOTO LDEB
89 ILP        ;Avance le pointeur à la ligne suivante
90 RLC        ;A(C),B(L),C(P)
91 POP        ;A(L)<=B(P)
92 CON 1      ;A(1)=>B(L)
93 SUB        ;A(L-1)
94 JNZ 83     ;=>TRANS ;IF(L.NE.1) GOTO TRANS
95 CON 1      ;A(1)
96 PSH        ;A(1)=>B(1)
97 PSH        ;A(1)=>B(1)=>C(1)
98 SLC        ;Place le pointeur en (1,1,1)
99 JMP 83     ;=>TRANS ;GOTO TRANS
100 OPN DATA1 ;<=LDEB ;Ouvre le fichier DATA1

```

```

101 RCL 16           ;A(PLLLL)           Calcul de LDEB
102 MOD 1000        ;A(L1)
103 RCL 55          ;A(P1)=>B(L1)
104 CON 1           ;A(1)=>B(P1)=>C(L1)
105 SUB            ;A(P1-1),B(L1)
106 RCL 49          ;A(LPP)=>B(P1-1)=>C(L1)
107 MLT            ;A((P1-1)*LPP),B(L1)
108 ADD            ;A(L1+(P1-1)*LPP)
109 STO 53         ;A(LDEB)=>R(53)           Lignes du début
110 RCL 15         ;A(NREC)=>A(LDEB)           Calcul de LFIN
111 XAB           ;A(LDEB)<=>B(NREC)
112 SUB           ;A(NREC-LDEB)
113 STO 54         ;A(LFIN)=>R(54)           Lignes de la fin
114 JLZ 141        ;=>COUPE ;IF(LFIN.LT.0) GOTO COUPE
115 JPZ 134        ;=>FIN  ;IF(LFIN.EQ.0) GOTO FIN
116 RCL 50         ;A(LDF)=>B(LFIN)           Calcul de PGO et de LGO
117 XAB           ;A(LFIN)<=>B(LDF)
118 SUB           ;A(LDF-LFIN)
119 PSH           ;A(LDF-LFIN)=>B(LDF-LFIN)
120 RCL 49         ;A(LPP)=>B(LDF-LFIN)=>C(LDF-LFIN)
121 DIV           ;A((LDF-LFIN)/LPP),B(LDF-LFIN)
122 CON 1         ;A(1)=>B((LDF-LFIN)/LPP),C(LDF-LFIN)
123 ADD           ;A(((LDF-LFIN)/LPP+1),B(LDF-LFIN) Page de départ
124 INT           ;A(PGO),B(LDF-LFIN)
125 XAB           ;A(LDF-LFIN)<=>B(PGO)
126 MOD 24         ;A(LGO),B(PGO)           Ligne de départ
127 CON 1         ;A(1)=>B(LGO)=>C(PGO)
128 SLC           ;Place le pointeur en (1,LGO,PGO)
129 RCL 54         ;A(LFIN)           Lignes de la fin
130 STO 51         ;A(LFIN)=>R(51)           Compteur de ligne
131 ILP           ;<=TR2 ;Avance le pointeur sur la ligne suivante
132 TRL           ;Transmet la ligne courante sous forme ASCII
133 DJN 51,131 ;=>TR2 ;R(51)<=R(51)-1:IF(R(51).NE.0) GOTO TR2
134 CON 1         ;<=FIN ;A(1)
135 PSH           ;A(1)=>B(1)
136 PSH           ;A(1)=>B(1)=>C(1)
137 SLC           ;Place le pointeur en (1,1,1)
138 RCL 53         ;A(LDEB)           Lignes du début
139 STO 51         ;A(LDEB)=>R(51)           Compteur
140 JMP 160        ;=>LA ;GOTO LA
141 PSH           ;<=COUPE ;A(LFIN)=>B(LFIN)
142 RCL 53         ;A(LDEB)=>B(LFIN)=>C(LFIN)
143 ADD           ;A(LFIN+LDEB),B(LFIN)
144 STO 53         ;A(LDEB)=>R(53)           Lignes du début
145 POP           ;A(LFIN)           Calcul de PGO et de LGO
146 CHS           ;A(-LFIN)
147 PSH           ;A(LFIN)=>B(LFIN)
148 RCL 49         ;A(LPP)=>B(LFIN)=>C(LFIN) Lignes par page
149 DIV           ;A(LFIN/LPP),B(LFIN)
150 CON 1         ;A(1)=>B(LFIN/LPP)=>C(LFIN)

```

```

151 ADD ;A(LFIN/LPP+1),B(LFIN)
152 INT ;A(PGO)
153 XAB ;A(LFIN)<=>B(PGO)
154 MOD 24 ;A(LGO),B(PGO)
155 CON 1 ;A(1)=>B(LGO)=>C(PGO)
156 SLC ;Place le pointeur en (1,LGO,PGO)
157 RCL 53 ;A(LDEB) Lignes du début
158 STO 51 ;A(LDEB)=>R(51) Compteur de ligne
159 ILP ;<=TR1 ;Avance le pointeur sur la ligne suivante
160 TRL ;<=LA ;Transmet la ligne courante sous forme ASCII
161 DJN 51,159;=>TR1 ;R(51)<=R(51)-1:IF(R(51).NE.0) GOTO TR1
162 ILP ;Avance le pointeur d'une ligne (Position originale)
163 RCL 15 ;<=END ;A(NREC)
164 WID 4 ;Réserve 4 espaces
165 DCM 0 ;Aucun chiffre après le point
166 CDS 17,0 ;Place le curseur à la case 17
167 VUA ;Affiche le registre 15
168 CNS LIGNES ;S(LIGNES)
169 CDS 22,0 ;Place le curseur à la case 22
170 WID 7 ;Réserve 7 espaces
171 VUS ;Affiche le mot LIGNES
172 CNS <CTL/026> ;S(CTL/Z)
173 TRS ;Transmet le registre S sous forme ASCII
174 CON 0 ;A(0)
175 STO 15 ;A(0)=>R(15)NREC Initialise NREC
176 STO 12 ;A(0)=>R(12)START Initialise R(12)
177 CON 1 ;A(1)
178 STO 17 ;A(1)=>R(17)DRAP Initialise DRAP
179 END

```

6.0 Appareillage utilisé pour le transfert des données.

Le présent logiciel a été utilisé avec succès à deux endroits depuis mai 1985. Un Polycorder a servi à l'enregistrement du niveau d'eau dans le puits Dominion d'Ottawa à l'aide d'une sonde de type capacitance. Un autre a traité les 7 canaux fournies par une station météorologique placée sur le toit du Laboratoire de Géophysique d'Ottawa et comprenant la vitesse du vent, la direction du vent, la température extérieure, l'humidité relative, la pression atmosphérique, la radiation solaire et les précipitations. Un huitième canal a été rajouté pour mesurer la température de la pièce où se trouve le gravimètre ET-12. Chaque canal a été balayé une fois par minute. Après traitement en temps réel, on a conservé une donnée par heure dans la mémoire du Polycorder.

Les données ont été transférées une fois par mois par l'intermédiaire d'une unité de mémoire portable de 64 K. L'unité en question est un Portapac. C'est un appareil fabriqué par Cryptonics, Inc., Fountain Valley, Ca, USA. Il pèse 650 g et mesure 2.54 X 9.02 X 23.1 cm. Le Portapac peut être ajusté pour transmettre ou recevoir selon 14 taux de transfert allant de 50 à 19200 Bauds. Le taux compatible pour transférer à l'aide du logiciel XMT les données du Polycorder au Portapac a été établi à 9600 Bauds.

Les données sont ensuite transférées sur disque rigide par l'intermédiaire de l'ordinateur principal, un LSI 11/23 de Digital Equipment doté du système d'opération RSX/11M. Le logiciel TEM est utilisé pour ce deuxième transfert et permet une vitesse de transmission maximum de 300 Bauds.

7.0 Instructions pour transférer un fichier de données du Polycorder au Portapac en utilisant le programme Polycorder XMT.

Les instructions suivantes s'appliquent particulièrement dans le cas où le Polycorder est en opération. L'ordre des opérations permet d'interrompre l'enregistrement des données durant un minimum de temps.

- 1- Placer le Portapac à 9600 bauds.

1	2	3	4	5	6	7	8
0	0	0	1	0	0	0	0

1 = Interrupteur levé lorsque les chiffres sont à l'endroit.
(On doit s'assurer que le Polycorder soit aussi à 9600 bauds)
- 2- Brancher le RS-232 mâle sur le Portapac à TO TERMINAL.
- 3- Placer HOST-ON, XMIT-OFF et START-OFF.
- 4- Appuyer sur RESET pour placer le pointeur au début du buffer.
- 5- Appuyer sur START => on aura maintenant START-ON.
- 6- Débrancher l'alimentation du Polycorder.
- 7- Entre deux AUTOLOG ON appuyer sur le ON du Polycorder.
Si le Polycorder ne s'allume pas, débrancher tous les cables.
- 8- Prendre note du registre 12 du Polycorder à l'aide du MODE 6-1.
- 9- Brancher le RS-232 femelle sur le Polycorder à SERIAL I/O.
- 10- Démarrer l'exécution du programme XMT et attendre le mot MODE ?.
Au début du programme apparaissent les renseignements suivants:

PP	NREC	NC	LPP
----	------	----	-----

PP = Nombre de pages à transférer.
NREC = Nombre de lignes à transférer.
NC = Nombre de colonnes à transférer.
LPP = Nombre de lignes par page.

A la fin du programme apparaît l'information suivante: NREC LIGNES
- 11- Débrancher le RS-232 femelle du Polycorder.
- 12- Appuyer sur 9 pour éteindre le Polycorder.
- 13- Brancher l'alimentation du Polycorder.
- 14- Appuyer sur START du Portapac => on aura START-OFF.

Au cas où le transfert des données n'aurait pas été effectué correctement, il faudrait entrer dans le registre 12 la valeur PPLL de la première ligne à transférer du fichier index et mettre 1 dans le registre 15 avant de faire exécuter XMT à nouveau. Si la première ligne à transférer du fichier index est inconnue, on peut transférer les deux fichiers au complet en mettant 2 dans le registre 15 et la valeur du registre 16 dans le registre 12. Voir la description du programme XMT pour plus de renseignements.

7.1 Instructions pour transférer un fichier de données du Portapac au LSI en utilisant la commande TEM du LSI 11/23.

Ces instructions sont offerts à titre d'exemple.

- 1- Placer le Portapac à 300 bauds.

1	2	3	4	5	6	7	8
	1	0	1	1	0	0	0

1 = Interrupteur levé lorsque les chiffres sont à l'endroit.
- 2- Brancher le RS-232 mâle sur le Portapac à TO TERMINAL.
- 3- Brancher l'autre RS-232 mâle dans une porte du LSI 11/23.
- 4- Placer HOST-ON, XMIT-ON et START-OFF.
- 5- Appuyer sur RESET pour placer le pointeur au début du buffer.
- 6- A partir d'un terminal du LSI 11/23, faire >TEM et répondre aux questions dans l'ordre:
 Selection : 5, Baud rate : 300 et TEM>TTn: /LOG:fichier.DAT
- 7- Appuyer sur START => on aura maintenant START-ON.
- 8- Le fichier est fermé lorsque TEM reçoit un <CTL Z> du Portapac ou du terminal. Cependant, même si le Portapac transmet un <CTL Z> au LSI, le reste du contenu du Portapac continue de défiler à l'écran. Pour sortir de TEM, entrer au terminal : <CTL C> <CTL Z> .

7.2 Instructions pour transférer les fichiers PGM, FMT et DAT du LSI au Polycorder.

Ces instructions sont offerts à titre d'exemple.

1- Voir la première page des instructions pour XMT et vérifier si le protocole du Polycorder est conforme.

2- Un fichier PGM sur LSI doit être de la forme:

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	<= Numéro de la colonne
b P G M	b = espace vide
b P R O G R A M	Nom du programme
b b b 1 b C O N b 1	No de ligne, instruction
b b b 2 b S T O b 1 0	et argument
b b b 3 b E N D	
b &	

3- Un fichier FMT sur LSI doit être de la forme:

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	<= Numéro de la colonne
b F M T	b = espace vide
b I N D E X . F	Nom du fichier
b 1 8	Nombre de lignes par page
b b 1 . / N 6 / C - 1 / b b /	
b &	

4- Un fichier DAT sur LSI doit être de la forme:

1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	<= Numéro de la colonne
b D A T	
b D A T A 1	Nom du fichier
b D A T A 1 . F	Nom du fichier FMT correspondant
b &	

5- Appeler le programme TEM du LSI en entrant:

- TEM
- Selection : 4
- Baud rate : 1200

6- Lorsque le protocole a été établi correctement sur le Polycorder, placer ce dernier en mode 3-2 avant d'entrer l'instruction suivante de TEM:

```
TEM>TTn: /SEND:fichier.POL
```

7- Lorsque les fichiers sont tous transférés sur le Polycorder, faire <CTL C> et <CTL Z> pour sortir de TEM.

8.0 Exemple de la distribution de l'espace mémoire du Polycorder pour le traitement de 8 canaux :

Nom	Bytes	Application	
INITLOG	244	Initialisation des registres	
AUTOLOG KAF PREDIC RESTEP UPDATE RINGB	817 254 471 141 503 532	Programme principal et sous-programmes	Logiciel 11.6%
XMT	722	Transfert des données	
KVAR.F INDEX.F DATA.F	402 44 128	Fichiers format	Données 81.0%
KVAR INDEX DATA	1088 188 23868	Fichiers de données	
	2333	Espace inutilisé	7.4%
	31735	Espace total disponible	100.0%

Avec une telle distribution, le système possède une autonomie de 31 jours. En utilisant l'espace qui reste, on pourrait augmenter cette période à un maximum de 34 jours.

9.0 Références

Omidata International, Inc. Polycorder Operator's Manual. Sept. 1984

J. Halpenny. A method of editing time series observations.
Geophysics, 49, 521-524, 1984.

Hamming, R. W. Numerical methods for scientists and engineers,
McGraw-Hill, 1962.

Gelb, A., Applied optimal estimation, Cambridge, Mass.,
MIT Press, 1974.