

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 MONITORING DE LA COUCHE PHYSIQUE D'UN RÉSEAU OPTIQUE	5
1.1 Introduction.....	5
1.2 Liaison optique longue distance.....	5
1.3 Effets linéaires de dégradation du signal optique	6
1.3.1 L'atténuation	7
1.3.2 La dispersion chromatique.....	8
1.4 La polarisation du signal optique.....	8
1.4.1 Définition d'un état de polarisation	9
1.4.2 Représentation de Stokes	10
1.4.3 Les pertes dépendantes de la polarisation.....	13
1.4.4 Délai de groupe différentiel et dispersion modale de polarisation	14
1.4.5 La fluctuation des états de polarisation.....	15
1.5 Monitoring de couche optique	16
1.5.1 Mesure de qualité du signal optique	17
1.5.1.1 La puissance du signal	17
1.5.1.2 Le taux d'erreur sur les bits	17
1.5.1.3 Le rapport signal sur bruit optique.....	18
1.5.2 Paramètres de performance importants.....	18
1.5.3 Outils et instruments de monitoring.....	19
1.6 Transpondeur cohérent.....	21
1.6.1 Description de la technologie cohérente	21
1.6.2 Présentation du transpondeur cohérent	22
1.6.3 Paramètres mesurés aux fins de compensation	25
1.7 Revue des expériences de monitoring.....	27
1.7.1 Récapitulatif des expériences de monitoring	27
1.7.2 Monitoring de liaison commerciale avec un transpondeur cohérent	29
1.7.3 Monitoring de liaison commerciale avec un polarimètre	30
1.8 Présentation du lien optique étudié et des données mesurées	33
1.9 Conclusion du chapitre	35
CHAPITRE 2 MÉTHODE DE TRAITEMENT DES DONNÉES DE MONITORING	37
2.1 Introduction.....	37
2.2 Les données issues du transpondeur cohérent	37
2.2.1 Description des paramètres mesurés et méthode d'acquisition des données	37
2.2.2 Traitement initial des données du transpondeur cohérent.....	39
2.2.2.1 Conversion des fichiers en structures Matlab	40

	2.2.2.2 Traitement des valeurs transitoires et stockage temporaire des résultats.....	42
	2.2.2.3 Concaténation des données traitées	42
2.3	2.2.3 Moyennage des paramètres mesurés par le transpondeur cohérent.....	43
	2.2.4 Utilisation du calcul parallèle	47
2.3	Les données issues du polarimètre.....	50
	2.3.1 Description des paramètres mesurés et méthode d'acquisition des données	50
	2.3.2 Traitement initial des données du polarimètre.....	51
	2.3.2.1 Normalisation de la matrice des états de polarisation et calcul de la matrice des vitesses de rotation du vecteur de Stokes	52
	2.3.2.2 Traitement des valeurs transitoires	54
	2.3.2.3 Calcul des paramètres statistiques par fichier, construction de l'histogramme des vitesses de rotation du vecteur de Stokes et détermination des événements de polarisation	55
	2.3.2.4 Détection des événements de polarisation	56
	2.3.2.5 Fusion des données	60
	2.3.3 Moyennage des paramètres mesurés par le polarimètre	60
	2.3.4 Utilisation du calcul parallèle	62
2.4	Évaluation du gain de performance obtenu par le calcul parallèle	63
	2.4.1 Résultats pour le traitement des données du polarimètre.....	64
	2.4.2 Résultat pour le traitement des données du transpondeur cohérent.....	65
2.5	Conclusion du chapitre	66
 CHAPITRE 3 ANALYSE STATISTIQUE DES RÉSULTATS DE MESURE D'UNE LIAISON COMMERCIALE AVEC UN TRANSPONDEUR COHÉRENT ET UN POLARIMÈTRE.....67		
3.1	Introduction.....	67
3.2	Outils d'analyse des données mesurées	67
	3.2.1 Calcul de la moyenne des échantillons à la minute	68
	3.2.2 Autocorrélation des paramètres de la liaison	68
	3.2.2.1 Interpolation linéaire des valeurs des paramètres	69
	3.2.2.2 Méthode de filtrage du bruit du signal.....	70
	3.2.2.3 Calcul de la fonction d'autocorrélation normalisée et des temps de corrélation caractéristiques.....	73
3.3	Analyse des mesures du transpondeur cohérent	74
	3.3.1 Périodes couvertes par les mesures du transpondeur cohérent	75
	3.3.2 Analyse des mesures brutes du transpondeur cohérent.....	75
	3.3.3 Analyse de la moyenne à la minute des paramètres mesurés	77
	3.3.4 Fonction d'autocorrélation de $\langle DGD \rangle$ et de $\langle \varphi \rangle$	81
3.4	Analyse des mesures du polarimètre.....	84
	3.4.1 Périodes couvertes par les mesures du polarimètre	84
	3.4.2 Analyse des mesures du polarimètre.....	84
3.5	Comparaison des mesures du polarimètre à celles du transpondeur cohérent.....	87

3.6	Analyse de l'influence de la température sur la liaison.....	92
3.6.1	Acquisition et traitement des données de température	93
3.6.2	Activité de polarisation et température	94
3.7	Conclusion du chapitre	95
CHAPITRE 4 CARACTÉRISATION DES EFFETS DE POLARISATION AVEC UN POLARIMÈTRE		97
4.1	Introduction.....	97
4.2	Effets de polarisation dans une liaison optique.....	98
4.2.1	État de polarisation de la lumière et représentation de Stokes.....	98
4.2.2	Seuils pour la détection d'événement de polarisation	99
4.2.3	Événements de polarisation élastiques et inélastiques.....	100
4.3	Distribution des vitesses de rotation du vecteur de Stokes	102
4.3.1	Détermination de la largeur optimale des colonnes de l'histogramme... 103	103
4.3.2	Histogramme des vitesses de rotation du vecteur de Stokes.....	103
4.3.3	Ajustement de l'histogramme des vitesses de rotation du vecteur de Stokes par une distribution de Rayleigh.....	105
4.3.4	Ajustement de l'histogramme pour les vitesses de rotation du vecteur de Stokes les plus élevées	109
4.4	Analyses complémentaires des états de polarisation	111
4.4.1	Analyse du bruit du polarimètre	111
4.4.1.1	Distribution des vitesses de rotation du vecteur de Stokes induites par le bruit	112
4.4.1.2	Caractérisation du bruit généré par le polarimètre.....	113
4.4.1.3	Simulation du bruit du polarimètre	115
4.4.2	Analyse harmonique des mesures du polarimètre	120
4.5	Analyse des effets rapides de polarisation.....	125
4.5.1	Étude de la distribution de la durée, de la déflexion angulaire et de l'amplitude des événements de polarisation	126
4.5.2	Analyse des événements extrêmes (en vitesse et en durée).....	130
4.5.2.1	Analyse de l'événement le plus rapide	130
4.5.2.2	Analyse de l'événement le plus long	132
4.5.3	Corrélation entre les déflexions angulaires maximales et les vitesses de rotation du vecteur de Stokes maximales observées durant les effets de polarisation	134
4.6	Conclusion du chapitre	136
CONCLUSION.....		139
RECOMMANDATIONS		143
ANNEXE I	TRAITEMENT INITIAL DES DONNÉES DU TRANSPONDEUR COHÉRENT	145
ANNEXE II	CONVERSION DES FICHIERS DE MESURE DU TRANSPONDEUR COHÉRENT EN FORMAT MATLAB	151

ANNEXE III	TRAITEMENT DES VALEURS TRANSITOIRES DU TRANSPONDEUR COHÉRENT	153
ANNEXE IV	MOYENNAGE DES MESURES DU TRANSPONDEUR COHÉRENT	155
ANNEXE V	PARITIONNEMENT DES FICHIERS DE MESURE PRÉALABLEMENT AU CALCUL PARALLÈLE	161
ANNEXE VI	TRAITEMENT INITIAL DES DONNEES DU POLARIMÈTRE	163
ANNEXE VII	CALCUL DE LA MATRICE DES VITESSES DE ROTATION DU VECTEUR DE STOKES.....	173
ANNEXE VIII	DÉTECTION DES ÉVÉNEMENTS DE POLARISATION	175
ANNEXE IX	ALGORITHME DE DÉTECTION DES ÉVÉNEMENTS DE POLARISATION	179
ANNEXE X	MOYENNAGE DES MESURES DU POLARIMÈTRE	181
ANNEXE XI	SEGMENTATION TEMPORELLE DES MESURES DU POLARIMÈTRE.....	185
ANNEXE XII	CALCUL ET TRACÉ DE LA FONCTION D'AUTOCORRÉLATION DES MESURES.....	191
ANNEXE XIII	COMPARAISON ENTRE LES MESURES DU TRANSPONDEUR COHÉRENT ET CELLES DU POLARIMÈTRE	197
ANNEXE XIV	TRACÉ SUR UN MÊME GRAPHIQUE DE L'ÉVOLUTION DE LA TEMPÉRATURE ET D'UN PARAMÈTRE MESURÉ	203
ANNEXE XV	CONVERSION DES FICHIERS CSV DU NOAA AU FORMAT MATLAB.....	207
ANNEXE XVI	TRACÉ DE L'HISTOGRAMME DES VITESSES DE ROTATION DU VECTEUR DE STOKES	211
ANNEXE XVII	AJUSTEMENT DE L'HISTOGRAMME DES VITESSES DE ROTATION DU VECTEUR DE STOKES À UNE DISTRIBUTION DE RAYLEIGH ET À UNE FONCTION EXPONENTIELLE	215
ANNEXE XVIII	CARACTÉRISATION DE LA DISTRIBUTION DU BRUIT DU POLARIMÈTRE.....	223
ANNEXE XIX	GÉNÉRATION D'UN POINT ALEATOIRE SUR UNE SPHÈRE	231

ANNEXE XX	SIMULATION DU BRUIT DU POLARIMÈTRE	233
ANNEXE XXI	TRACÉ DE L'EVOLUTION TEMPORELLE DES PARAMETRES DE STOKES MESURÉS PAR LE POLARIMÈTRE	239
ANNEXE XXII	ANALYSE DE LA DENSITÉ SPECTRALE DES MESURES DU POLARIMÈTRE.....	243
ANNEXE XXIII	DISTRIBUTIONS DES EFFETS RAPIDES DE POLARISATION.....	245
ANNEXE XXIV	HISTOGRAMME 3D DES DÉFLEXIONS ANGULAIRES MAXIMALES ET DES VITESSES DE ROTATION DU VECTEUR DE STOKES MAXIMALES	249
BIBLIOGRAPHIE.....		255

LISTE DES TABLEAUX

	Page
Tableau 1.1 Récapitulatif des expériences de monitoring	28
Tableau 2.1 Paramètres monitorés par le transpondeur cohérent	41
Tableau 2.2 Valeurs de sortie du script de détection des événements de polarisation	57
Tableau 3.1 Caractérisation des valeurs de PMD, PDL et BER brutes mesurées par le transpondeur cohérent sur trois saisons	76
Tableau 3.2 Caractérisation des valeurs de PMD, PDL et BER moyennées à la minute et mesurées sur trois saisons	77
Tableau 3.3 Mesures du polarimètre	84
Tableau 3.4 Mesures de $\langle\phi\rangle$ par le polarimètre utilisées pour calculer la fonction d'autocorrélation	85
Tableau 4.1 Caractérisation des valeurs de vitesse de rotation du vecteur de Stokes mesurées par le polarimètre entre octobre 2012 et septembre 2014	102
Tableau 4.2 Informations contenues dans la structure de donnée des événements rapides de polarisation	126

LISTE DES FIGURES

	Page	
Figure 1.1	Schéma d'une liaison optique longue distance	5
Figure 1.2	Effets de l'atténuation et de la dispersion sur le signal optique.....	7
Figure 1.3	Montage permettant de mesurer les composantes de Stokes	11
Figure 1.4	Représentation de la sphère de Poincaré et de l'évolution temporelle de l'état de polarisation de deux signaux optiques	12
Figure 1.5	Représentation visuelle de la DGD.....	14
Figure 1.6	Représentation d'un effet rapide de polarisation élastique et inélastique sur la sphère de Poincaré	16
Figure 1.7	Architecture du polarimètre	21
Figure 1.8	Architecture du transmetteur cohérent DP-QPSK	23
Figure 1.9	Diagramme bloc d'un transmetteur QPSK	24
Figure 1.10	Architecture du récepteur cohérent.....	25
Figure 1.11	Diagramme bloc du traitement du signal optique au niveau du récepteur à 40 Gbit/s de Ciena	26
Figure 1.12	Schéma du montage expérimental utilisé	34
Figure 1.13	Représentation visuelle des mesures réalisées dans le projet par le polarimètre et le transpondeur cohérent.....	35
Figure 2.1	Chronogramme du cycle d'acquisition des mesures du transpondeur cohérent.....	38
Figure 2.2	Étapes du traitement initial des données du transpondeur cohérent	39
Figure 2.3	Première étape du moyennage à la seconde : répartition des valeurs d'horodatage	44
Figure 2.4	Seconde étape du moyennage à la seconde : calcul des valeurs moyennes et des écarts types	44

Figure 2.5	Représentation visuelle des moteurs de calcul parallèle.....	48
Figure 2.6	Fonctionnement du script de partitionnement.....	49
Figure 2.7	Calcul parallèle et fusion des données du transpondeur cohérent	49
Figure 2.8	Chronogramme du cycle d'acquisition des mesures du polarimètre	51
Figure 2.9	Étapes du traitement initial des données du polarimètre	52
Figure 2.10	Étapes de normalisation de l'histogramme des vitesses de rotation du vecteur Stokes	56
Figure 2.11	Représentation de la longueur d'arc entre deux vecteurs de Stokes.....	58
Figure 2.12	Application de l'algorithme à un effet rapide de polarisation	59
Figure 2.13	Étapes de traitement pour l'analyse temporelle des mesures du polarimètre	61
Figure 2.14	Calcul parallèle et fusion des données du polarimètre.....	63
Figure 2.15	Tracé de la comparaison du temps d'exécution pour le polarimètre	65
Figure 2.16	Tracé de la comparaison du temps d'exécution pour les mesures du transpondeur cohérent.....	66
Figure 3.1	Étapes du calcul de la fonction d'autocorrélation d'un paramètre mesuré (DGD, PDL, S ₁).....	69
Figure 3.2	Effet du filtrage super gaussien d'ordre 4 de <DGD> sur une période de 2 jours.....	73
Figure 3.3	Évolution temporelle de <DGD>, <PDL>, la valeur absolue de leurs dérivées temporelles et <BER> pour chaque minute sur le lien de Verizon	79
Figure 3.4	Évolution temporelle de <DGD>, <PDL>, leurs dérivées temporelles et <BER> pour chaque minute sur le lien de Verizon	80
Figure 3.5	Évolution temporelle de <DGD>, <PDL>, leurs dérivées temporelles et <BER> pour chaque minute sur le lien de Verizon	80
Figure 3.6	Définition de l'angle φ sur la sphère de Poincaré	81
Figure 3.7	Fonction d'autocorrélation (partie positive) de <DGD>	82
Figure 3.8	Fonction d'autocorrélation (partie positive) de <φ>.....	83
Figure 3.9	Fonction d'autocorrélation de <φ> sur la période d'hiver 2013	86

Figure 3.10	Fonction d'autocorrélation de $\langle\varphi\rangle$ sur la période de printemps 2014.....	86
Figure 3.11	Fonction d'autocorrélation de $\langle\varphi\rangle$ sur la période d'été 2014	87
Figure 3.12	Première étape du calcul de φ à partir des mesures du polarimètre et des mesures du transpondeur cohérent.....	88
Figure 3.13	Seconde étape de calcul de φ à partir des mesures du polarimètre et des mesures du transpondeur cohérent.....	89
Figure 3.14	Représentation temporelle de φ mesuré par le polarimètre et le transpondeur cohérent.....	90
Figure 3.15	Représentation de la corrélation croisée des mesures du polarimètre et des mesures du transpondeur cohérent.....	92
Figure 3.16	Traitements des données météorologiques.....	93
Figure 3.17	Évolution temporelle de $\langle\varphi\rangle$ mesurée par le transpondeur cohérent comparée à l'évolution de la température sur une même période	95
Figure 4.1	Évolution du vecteur de Stokes sur une période de 25 millisecondes	99
Figure 4.2	Représentation d'un changement d'état de polarisation élastique et inélastique sur la sphère de Poincaré	101
Figure 4.3	Évolution temporelle des trois composantes de Stokes lors d'un changement d'état de polarisation élastique et inélastique	101
Figure 4.4	Évolution temporelle de la vitesse de rotation du vecteur de Stokes lors d'un changement d'état de polarisation élastique et inélastique.....	101
Figure 4.5	Histogramme des vitesses de rotation du vecteur de Stokes sur la liaison aérienne. L'histogramme est présenté sur une échelle logarithmique	104
Figure 4.6	Ajustement d'une loi de Rayleigh sur les vitesses de rotation du vecteur de Stokes sur la liaison aérienne	107
Figure 4.7	Ajustement de la partie droite de l'histogramme des vitesses de rotation du vecteur de Stokes pour l'ensemble de la période de mesure ..	110
Figure 4.8	Histogramme des vitesses de rotation du vecteur de Stokes résultant du bruit généré par le polarimètre	112
Figure 4.9	Histogramme des valeurs de S_1 observées lors des mesures dos à dos du polarimètre	113

Figure 4.10	Méthode de filtrage des mesures dos-à-dos du polarimètre.....	114
Figure 4.11	Distribution des valeurs de S_1 résultant du bruit généré par le polarimètre	115
Figure 4.12	Étapes de la génération d'un fichier de polarimètre	116
Figure 4.13	Distribution de 5000 états de polarisation aléatoires en utilisant la méthode proposée par Weisstein (2015).....	118
Figure 4.14	Histogramme des vitesses de rotation du vecteur de Stokes obtenues par simulation du bruit du polarimètre (DOP = 1)	119
Figure 4.15	Histogramme des vitesses de rotation du vecteur de Stokes obtenues par simulation du bruit du polarimètre ($0.8 < \text{DOP} < 1$)	120
Figure 4.16	Représentation de l'évolution temporelle de l'état de polarisation du signal sur la sphère de Poincaré sur une durée d'une seconde, de 10 secondes, de 100 secondes	121
Figure 4.17	Évolution temporelle des trois composantes de Stokes mesurées à partir des données brutes du polarimètre	122
Figure 4.18	Spectre de puissance de la première composante de Stokes pour une durée de mesure de 10 secondes	124
Figure 4.19	Distribution de la durée de l'ensemble des événements de polarisation observés sur la liaison aérienne.....	127
Figure 4.20	Distribution de la déflexion angulaire finale de l'ensemble des événements de polarisation observés sur la liaison aérienne	128
Figure 4.21	Distribution de la vitesse de rotation du vecteur de Stokes maximale de l'ensemble des événements de polarisation observés sur la liaison aérienne	129
Figure 4.22	Représentation de l'évolution temporelle de la vitesse de rotation du vecteur de Stokes, des paramètres de Stokes et du vecteur de Stokes sur la sphère de Poincaré durant l'effet de polarisation le plus rapide enregistré sur la liaison aérienne	130
Figure 4.23	Représentation de l'évolution temporelle de la vitesse de rotation du vecteur de Stokes, des paramètres de Stokes et du vecteur de Stoke sur la sphère de Poincaré durant l'effet de polarisation le plus long enregistré sur la liaison aérienne	132
Figure 4.24	Histogramme des vitesses maximales de rotation du vecteur de Stokes et des déflexions maximales des effets transitoires de polarisation	135

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

APD	Avalanche Photodiode
ACF	Autocorrelation Function
ASE	Amplified Spontaneous Emission
BER	Bit Error Rate
CD	Chromatic Dispersion
CSV	Comma-Separated Values
DCF	Dispersion Compensating Fiber
DGD	Differential Group Delay
DOP	Degree Of Polarization
DP-QPSK	Dual Polarization Quadrature Phase Shift Keying
DSP	Digital Signal Processing
EDFA	Erbium-doped Fiber Amplifier
FEC	Forward Error Correction
FIR	Finite Impulse Response
NDSF	Non Dispersion-shifted Fiber
NGM	New Generation Modem
OCLD	Optical Channel Laser Detector
OPGW	Optical Ground Wire
OSA	Optical Spectrum Analyser
OSNR	Optical Signal to Noise Ratio
PDL	Polarization Dependant Loss

PMD	Polarization Mode Dispersion
PSK	Phase-Shift Keying
ROADM	Reconfigurable Add Drop Multiplexer
SNR	Signal to Noise Ratio
SOP	State Of Polarization
WDM	Wavelength Division Multiplexing

LISTE DES SYMBOLES ET UNITÉS DE MESURE

UNITÉS LIÉES AU TEMPS

s	seconde
ms	milliseconde (10^{-3} s)
μ s	microseconde (10^{-6} s)
ns	nanoseconde (10^{-9} s)
ps	picoseconde (10^{-12} s)
Hz	Hertz
kHz	kilohertz (10^3 Hz)
MHz	mégahertz (10^6 Hz)
GHz	gigahertz (10^9 Hz)
THz	térahertz (10^{12} Hz)

UNITÉS DE PUISSANCE

W	Watt
mW	milliwatt (10^{-3} W)
dB	décibel
dBm	décibel (référence à 1 mW)

UNITÉS DE DISTANCE

m	mètre
km	kilomètre (10^3 m)

UNITÉS DE DÉBIT BINAIRE

bit	bit
Mbit	mégabit (10^6 bits)
Gbit	gigabit (10^9 bits)

SYMBOLES

λ	longueur d'onde
γ	déflexion angulaire
E	amplitude du champ électrique
Φ	phase du champ électrique
φ	activité de polarisation
rad	radian
$^\circ$	degré

INTRODUCTION

Les systèmes de transmission basés sur la technologie cohérente constituent aujourd’hui les solutions privilégiées pour augmenter la capacité binaire des liaisons optiques. Ces systèmes sont souvent associés à un format de modulation à double polarisation et à modulation de phase différentielle décalée en quadrature (DP-QPSK). La technologie cohérente est accompagnée de l'utilisation de modules de traitement numérique du signal (DSP) capables de compenser à la fois les effets linéaires tels que la dispersion chromatique (CD), les effets de polarisation tels la dispersion modale de polarisation (PMD) ou encore les fluctuations rapides d'état de polarisation. Les modules DSP sont conçus pour pouvoir compenser ces phénomènes dans une certaine limite définie lors de leur conception, de sorte que ces limites pourraient éventuellement être dépassées et entraîner des pannes de ces systèmes de transmission. Des travaux de monitoring ont été effectués sur des liaisons commerciales afin de caractériser certains effets de polarisation. Cependant, la quasi-totalité de ces travaux a porté sur des liaisons enterrées. Il nous a semblé pertinent de caractériser les fluctuations d'états de polarisation sur une liaison aérienne, sur la base de l'hypothèse que c'est dans ce type de liaison que se produisent vraisemblablement les variations de performance et les fluctuations d'états de polarisation les plus intenses et les plus rapides.

Ce projet, qui a été réalisé en collaboration avec l'équipementier Ciena et l'opérateur Verizon, a eu pour objectif de **caractériser, sur une longue période, le comportement d'une liaison optique aérienne cohérente** à l'aide de deux instruments : d'une part, un transpondeur cohérent DP-QPSK à 40 Gbit/s de Ciena et, d'autre part, un polarimètre Agilent N7786B. L'utilisation d'un transpondeur cohérent comme moniteur de performance s'inscrit dans la continuité des travaux de (Woodward et al., 2014) et (Mortelette, 2014). Notre objectif a été dans un premier temps **d'analyser l'évolution temporelle des différents paramètres d'une liaison optique aérienne cohérente**. Le second objectif porte sur la **caractérisation des effets de polarisation sur la liaison aérienne avec un polarimètre**. Cette étude s'inscrit quant à elle dans la continuité des travaux de (Peterson Jr, 2004), (Nelson, 2012) et (Mortelette, 2014). La valeur ajoutée de ce projet porte sur l'acquisition d'une quantité très

importante de mesures réalisées avec un polarimètre sur une liaison aérienne commerciale, qui représente plus de 180 jours de mesure sur une période de deux ans.

Ma contribution a porté dans un premier temps sur **l'amélioration des scripts de traitement Matlab** développés par (Mortelette, 2014). Cette amélioration repose principalement sur la **mise en place du calcul parallèle** afin de diminuer le temps nécessaire pour traiter les données mesurées par les deux instruments. **Diverses optimisations** ont aussi été réalisées dans le code afin d'accélérer le processus de traitement des données. Ma seconde contribution a porté sur le **traitement des nouvelles données mesurées**, ce qui représentait un défi compte tenu de la quantité importante de données recueillies sur le terrain et traitées dans le cadre de ce projet. Par ailleurs, ma contribution a aussi porté sur le développement d'outils Matlab permettant de procéder à de nouvelles analyses : le premier outil développé est relatif à la **comparaison des données mesurées par transpondeur cohérent avec celles mesurées par le polarimètre**. Le second outil a permis de **comparer l'évolution de certains paramètres de la liaison avec l'évolution de la température moyenne** sur celle-ci. Le troisième a permis **d'analyser l'évolution temporelle des paramètres mesurés par le polarimètre**. Le quatrième a permis de **simuler le bruit du polarimètre** afin de quantifier l'influence du bruit sur les mesures. Enfin, le dernier outil a permis d'effectuer une **analyse harmonique des mesures du polarimètre**.

Ce mémoire est organisé en quatre chapitres. Le premier chapitre présente les notions de base relatives au monitoring de la couche physique d'un réseau optique, en mettant l'emphase sur la notion de polarisation. Ce chapitre s'achève par la présentation de la liaison optique étudiée et des données mesurées. Le second chapitre présente de manière détaillée la méthode de traitement des données adoptée, en mettant l'emphase sur la mise en place du calcul parallèle. Le chapitre se conclut sur la comparaison des performances des scripts développés dans le cadre de ce projet par rapport à ceux développés par (Mortelette, 2014). Le troisième chapitre présente l'analyse statistique des résultats de mesure en mettant l'emphase sur les effets de polarisation et présente une comparaison des mesures du

transpondeur cohérent avec celles du polarimètre. Enfin, le chapitre quatre porte sur la caractérisation des effets de polarisation avec un polarimètre.

CHAPITRE 1

MONITORING DE LA COUCHE PHYSIQUE D'UN RÉSEAU OPTIQUE

1.1 Introduction

Ce chapitre présente dans un premier temps les éléments constituant une liaison optique longue distance. Les paramètres qui caractérisent le comportement d'une liaison optique sont ensuite détaillés en mettant l'emphase sur les effets de polarisation qui jouent un rôle critique dans la performance des liaisons optiques à très haute vitesse. Un aperçu haut niveau des expériences de monitoring de réseaux optiques est enfin présenté.

1.2 Liaison optique longue distance

Une liaison optique à longue distance a pour objectif de transporter un trafic de données (séquence de bits sous forme d'un signal électrique) d'un point source à un point destination, séparés d'une distance de l'ordre de la centaine voire des milliers de kilomètres. Une telle liaison est toujours constituée des mêmes composants de base et est présentée à la Figure 1.1.

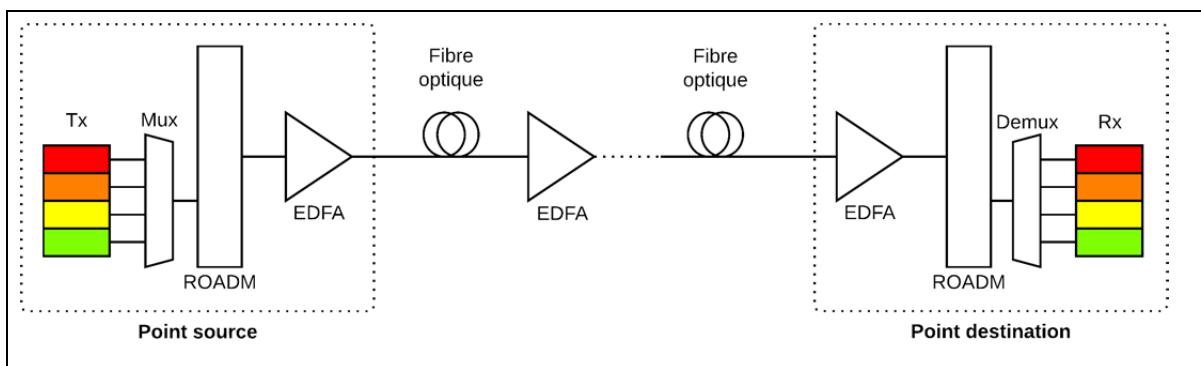


Figure 1.1 Schéma d'une liaison optique longue distance

Les transmetteurs, notés Tx, sont basés sur des diodes laser et assurent la conversion de l'information électrique en signal optique. Les transmetteurs optiques utilisés pour les liaisons à haute vitesse (c'est-à-dire supérieures à 2.5 Gbit/s) utilisent une modulation

externe. Les récepteurs optiques, notés Rx, sont situés au niveau du point de réception du signal optique et assurent la conversion du signal optique en signal électrique, contenant l'information transmise. Ces récepteurs sont basés sur des photodiodes qui peuvent être de deux types : les photodiodes PIN et les photodiodes à avalanche (*Avalanche Photodiode, APD*). Ces dernières offrent une meilleure sensibilité et sont plus coûteuses.

Les réseaux optiques modernes utilisent majoritairement un multiplexage en longueur d'onde (*Wavelength Division Multiplexing, WDM*), qui permet d'injecter dans une seule fibre optique plusieurs canaux optiques indépendants correspondant chacun à une longueur d'onde (couleur) donnée. Le dispositif permettant d'assurer le multiplexage de plusieurs couleurs sur une fibre est un multiplexeur, noté Mux. L'opération inverse est assurée par un démultiplexeur, noté Démux. L'indépendance des canaux rend les réseaux WDM très flexibles puisqu'il est possible en n'importe quel point de la liaison d'extraire et/ou d'ajouter une longueur d'onde donnée. Cette tâche est assurée par des éléments de commutation optiques parmi lesquels on peut citer les modules d'ajout/extraction de canal optique reconfigurables (*Reconfigurable Optical Add-Drop Multiplexer, ROADM*). Les amplificateurs optiques dopés à l'erbium permettent d'amplifier le signal optique.

1.3 **Effets linéaires de dégradation du signal optique**

Le signal optique qui se propage dans une fibre optique subit diverses perturbations qui vont en altérer la qualité. Les perturbations élémentaires ou effets linéaires principaux sont l'atténuation et la dispersion et sont présentés à la Figure 1.2.

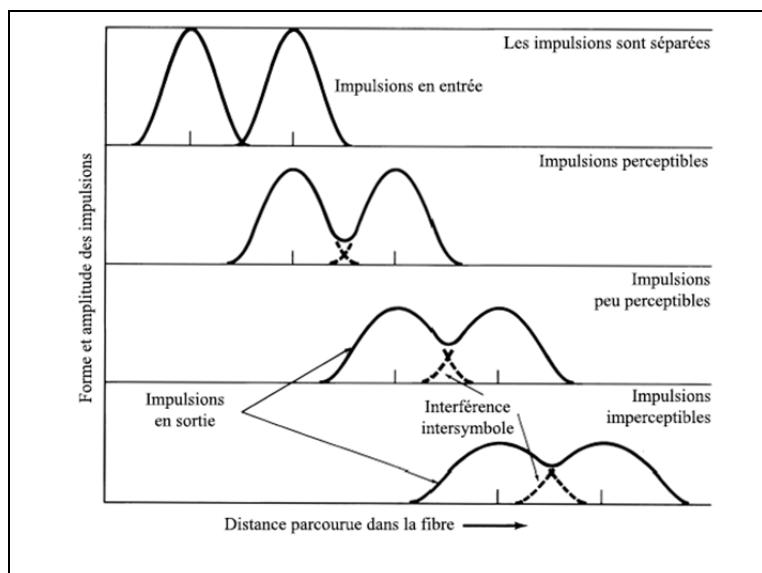


Figure 1.2 Effets de l'atténuation et de la dispersion sur le signal optique
Adaptée de Keiser (2010)

1.3.1 L'atténuation

L'atténuation correspond à l'affaiblissement de signal optique au fur et à mesure qu'il se propage dans la fibre. En effet, la puissance du signal optique décroît de manière exponentielle en fonction de la distance. Il en résulte une baisse de l'amplitude des signaux optiques transmis. Afin de faciliter les calculs de puissance, l'échelle logarithmique (décibel, dB) est couramment utilisée lors des calculs des pertes de puissance. Dans cette échelle, l'évolution de la puissance dans une fibre devient linéaire. L'atténuation du signal se manifeste par la diminution de l'amplitude des impulsions, qui est visible à la Figure 1.2. L'utilisation d'amplificateurs optiques, tels que les amplificateurs dopés à l'erbium (*Erbium Doped Fibre Amplifier*, EDFA) permet d'amplifier le signal optique par pompage optique des atomes d'erbium contenus dans le segment de fibre amplificatrice. Pour une fibre optique de type NDSF (*Non Dispersion-shifted Fiber*), c'est-à-dire une fibre optique dont la composition n'a pas été modifiée afin de modifier la longueur d'onde de dispersion nulle, le coefficient d'atténuation est de l'ordre de 0,20 dB/km.

1.3.2 La dispersion chromatique

La dispersion chromatique est un effet linéaire qui peut dégrader la qualité des impulsions transmises dans une liaison optique. Ce phénomène résulte du fait que l'indice de réfraction d'une fibre optique diffère en fonction de la longueur d'onde. Ainsi, chaque longueur d'onde se propage dans la fibre à une vitesse différente, entraînant un élargissement des impulsions. Les effets de la dispersion chromatique augmentent proportionnellement au carré du débit binaire dans le cas d'un système à modulation en intensité et détection directe. Ce phénomène entraîne des pénalités 16 fois supérieures sur des liaisons à 40 Gbit/s comparativement aux liaisons à 10 Gbit/s (Chan, 2010). Chaque fibre optique est caractérisée par son coefficient de dispersion chromatique qui est généralement exprimée en ps/(nm.km). Ce coefficient est fourni par le fabricant de la fibre et est de l'ordre de 17 ps/(nm.km) pour une fibre optique de type NDSF. La Figure 1.2 met en évidence l'élargissement des impulsions induit par la dispersion chromatique.

1.4 La polarisation du signal optique

L'augmentation du débit par canal des liaisons optiques conventionnelles (de type modulation en intensité avec détection directe de l'amplitude) au-delà de 2.5 Gbit/s est accompagnée d'une plus grande sensibilité aux effets de polarisation comme la dispersion modale de polarisation (*Polarization Mode Dispersion*, PMD). De plus, les formats de modulation en amplitude et en phase les plus récents reposent sur le multiplexage en polarisation de la lumière. Les sous-sections qui suivent présentent les notions théoriques associées à la polarisation de la lumière. Il y a plusieurs façons de représenter l'état de polarisation d'une onde lumineuse, en particulier la représentation de Jones, qui permet de bien comprendre la polarisation de la lumière et le fonctionnement des dispositifs optiques agissant sur la polarisation d'un signal lumineux. Une autre représentation utile à ce projet est la représentation de Stokes, qui sera décrite ensuite.

1.4.1 Définition d'un état de polarisation

La lumière est une onde électromagnétique. Ce type d'onde est constitué d'un champ magnétique et d'un champ électrique. Les variations des champs électrique et magnétique sont liées par les équations de Maxwell. Par convention, nous nous intéresserons uniquement au champ électrique, qui peut être représenté dans le plan complexe par l'équation (1.1).

$$\vec{E}(t) = \overrightarrow{E}_0 e^{i(\omega t - \vec{k} \cdot \vec{r} + \varphi)} \quad (1.1)$$

où \overrightarrow{E}_0 désigne le vecteur d'amplitude de l'onde, ω la pulsation de l'onde, \vec{k} le vecteur d'onde, \vec{r} le vecteur position du point d'observation et φ la phase à la source de l'onde.

L'état de polarisation (*State Of Polarization*, SOP) d'une onde plane peut se représenter comme la superposition de deux champs orthogonaux polarisés linéairement. Chaque champ est dépendant du temps et dispose d'une amplitude réelle (E) et d'une phase (ϕ). Ces deux paramètres sont fixes ou variables. L'équation (1.2), tirée de (Xu, 2010), présente une expression du champ électrique,

$$\vec{E}(t) = \overrightarrow{a}_x E_{x0} \cos(\omega t + \phi_x) + \overrightarrow{a}_y E_{y0} \cos(\omega t + \phi_y) \quad (1.2)$$

où \overrightarrow{a}_x et \overrightarrow{a}_y désignent les vecteurs unitaires du plan orthogonal à la direction de propagation de l'onde et ϕ_x et ϕ_y désignent la phase du champ électrique selon la direction x et y respectivement.

On peut distinguer 3 états d'ondes principaux, dépendants de la relation qui existe entre ϕ_x et ϕ_y . Ainsi,

- si $\phi_x = \phi_y$, les deux composantes du champ électrique sont en phase, et le champ résultant est décrit par une droite : on parle de polarisation linéaire;

- si $\phi_x \neq \phi_y$, les deux composantes du champ électrique sont déphasées, et le champ résultant est décrit par une ellipse : on parle de polarisation elliptique.

Il existe par ailleurs une notion d'orientation de polarisation, correspondant au sens de rotation du champ électrique dans le cas d'une polarisation elliptique ainsi :

- si $\phi_x < \phi_y$, alors la composante y est en avance par rapport à x et le vecteur tourne dans le sens des aiguilles d'une montre : on parle de polarisation droite;
- si $\phi_x > \phi_y$, alors la composante en y est en retard par rapport à x et la rotation du vecteur se fait dans le sens contraire des aiguilles d'une montre : on parle de polarisation gauche.

1.4.2 Représentation de Stokes

Il existe plusieurs représentations de l'état de polarisation d'une onde lumineuse parmi lesquelles on peut citer la représentation de Jones et la représentation de Stokes. C'est cette dernière représentation qui est la plus appropriée pour ce projet, car elle permet non seulement de caractériser la partie polarisée du signal, comme la représentation de Jones, mais aussi des informations relatives au degré de polarisation du signal (*Degree Of Polarization*, DOP), qui correspond à un rapport entre la puissance du signal polarisé sur la puissance totale du signal optique (Chan, 2010).

La représentation de Stokes repose sur la définition des quatre paramètres de Stokes, notés S_0 , S_1 , S_2 et S_3 . Ces quatre paramètres peuvent être obtenus en mesurant l'intensité lumineuse observée dans le montage présenté à la Figure 1.3.

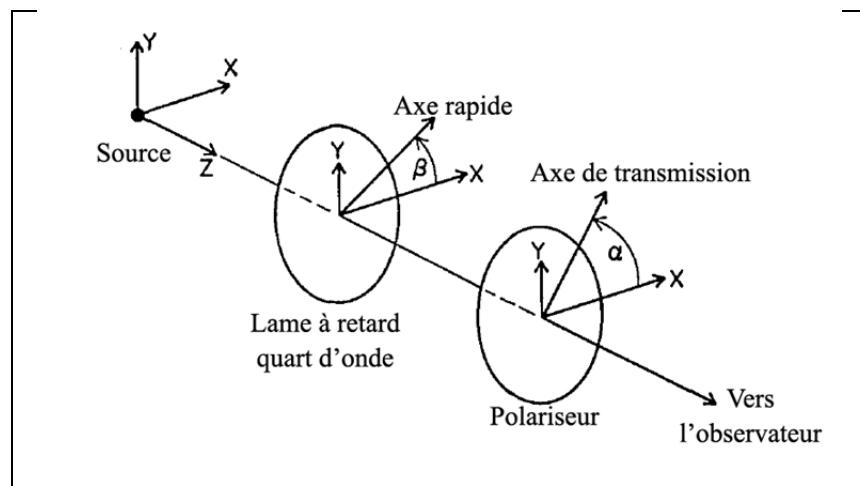


Figure 1.3 Montage permettant de mesurer les composantes de Stokes
Adaptée de Berry, Gabrielse et Livingston (1977)

Dans cette figure, α désigne l'angle entre la direction X et l'axe de transmission du polariseur et β désigne l'angle entre la direction X et l'axe rapide de la lame à retard quart d'onde. Une lame à retard quart d'onde est un dispositif dont l'état de polarisation de sortie dépend de l'état de polarisation en entrée et entraînant un retard de phase de 90° entre l'état de polarisation selon la direction X et l'état de polarisation selon la direction Y. En utilisant ce montage, il est possible de déterminer les quatre composantes de Stokes en effectuant plusieurs mesures d'intensité lumineuse (Kohns et Macheckin, 2002) :

- S_0 est l'intensité totale de l'onde lumineuse;
- S_1 est la différence entre l'intensité de l'onde mesurée avec $\alpha = 0^\circ$ et l'intensité de l'onde mesurée $\alpha = 90^\circ$ sans lame à retard;
- S_2 est la différence entre l'intensité de l'onde mesurée avec $\alpha = 45^\circ$ et l'intensité de l'onde mesurée $\alpha = 135^\circ$ sans lame à retard;
- S_3 est la différence entre l'intensité de l'onde mesurée avec $\beta = 45^\circ$ et l'intensité de l'onde mesurée avec $\beta = 135^\circ$, tandis que le polariseur est placé à 0° .

La sphère de Poincaré correspond à une représentation dans l'espace cartésien de l'espace de Stokes. Les trois principaux axes de la sphère correspondent aux trois paramètres de Stokes : S_1 , S_2 et S_3 . Ainsi, tout état de polarisation peut être représenté par un point sur la sphère de

Poincaré. Il est ainsi possible d'obtenir le vecteur de Stokes correspondant à cet état de polarisation entre l'origine de la sphère et le point sur la sphère correspondant à l'état de polarisation. La Figure 1.4 illustre deux exemples de l'évolution temporelle de l'état de polarisation de signaux optiques sur la sphère de Poincaré.

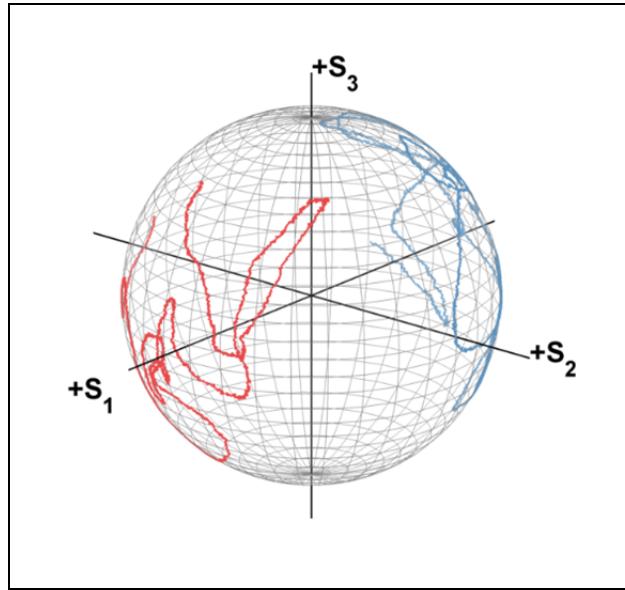


Figure 1.4 Représentation de la sphère de Poincaré et de l'évolution temporelle de l'état de polarisation de deux signaux optiques, en rouge et en bleu respectivement

Le paramètre S_0 est quant à lui relié au degré de polarisation du signal : il est possible de calculer le degré de polarisation du signal en utilisant la formule donnée à l'équation (1.3), tirée de (Kohns et Macheckhin, 2002).

$$DOP = \frac{\sqrt{{S_1}^2 + {S_2}^2 + {S_3}^2}}{S_0} \quad (1.3)$$

Dans cette équation, DOP désigne le degré de polarisation du signal optique. Un signal totalement polarisé est caractérisé par un $DOP = 1$ tandis qu'un signal dépolarisé ou non polarisé a un $DOP < 1$. Par ailleurs, la représentation de Stokes d'un signal totalement

polarisé se situe sur la surface de la sphère alors que la représentation d'un signal dépolarisé se situe à l'intérieur de la sphère.

L'hémisphère nord de la sphère correspond aux états de polarisation droite tandis que l'hémisphère sud correspond aux états de polarisation gauche. Par ailleurs, l'équateur correspond aux états de polarisation linéaire. Ces deux résultats se déduisent immédiatement de l'expression de S_3 . Les autres états de polarisation correspondent ainsi à de la polarisation elliptique.

Le signal optique sortant d'un transmetteur optique est polarisé linéairement avec un degré de polarisation est égal à 1 : sa représentation de Stokes se trouve donc dans l'équateur de la sphère ($S_3 = 0$). Cependant, l'état de polarisation du signal optique évolue au fur et à mesure qu'il se propage dans la fibre optique, entraînant ainsi un état de polarisation variant au cours du temps au niveau du récepteur. On peut ainsi tracer l'évolution de l'état de polarisation du signal sur la sphère de Poincaré. La Figure 1.4 illustre deux exemples d'évolution temporelle de l'état de polarisation d'un signal se propageant dans une fibre optique durant deux périodes distinctes (représentées respectivement en bleu et en rouge sur la figure).

1.4.3 Les pertes dépendantes de la polarisation

Dans une liaison optique, une des pertes liées aux effets de polarisation correspond aux pertes dépendantes de la polarisation (*Polarization Dependent Loss*, PDL). Ce phénomène correspond à la variation maximale de la perte d'insertion du dispositif en fonction de l'état de polarisation. En effet, les pertes d'un dispositif optique (tel que multiplexeur, démultiplexeur, fibre, etc.) dépendent de l'état de polarisation du signal à l'entrée. La nature aléatoire du changement des états de polarisation dans une fibre optique entraîne des variations aléatoires de la PDL. La PDL en échelle logarithmique est exprimée par l'équation (1.4), tirée de (Abuzariba, 2010).

$$PDL_{dB} = 10 \log \left(\frac{P_{max}}{P_{min}} \right) \quad (1.4)$$

1.4.4 Délai de groupe différentiel et dispersion modale de polarisation

Une fibre optique monomode comprend deux états indépendants de polarisation correspondant chacun à un mode de propagation (Keiser, 2010). Chacun de ces modes se propage à une vitesse différente : il s'agit du délai de groupe différentiel (*Differential Group Delay*, DGD), qui est une forme de dispersion intramodale. Bien que ce phénomène ait un impact limité sur les liaisons optiques de première génération, il entraîne des difficultés pour les systèmes de transmission optique à plus de 10 Gbit/s (Peterson Jr, 2004).

La différence de vitesse de propagation des deux modes résulte de la biréfringence de la fibre optique. On parle de la biréfringence lorsque le cœur de la fibre optique ne présente pas une symétrie circulaire parfaite. Autrement dit, le cœur de la fibre est à section elliptique où la fibre ne subit pas de contraintes parfaitement symétriques. On observe alors un indice de réfraction différent selon les deux directions de propagation principales. La vitesse de propagation de la lumière étant dépendante de l'indice de réfraction du milieu dans lequel elle se propage, les deux modes se propagent à une vitesse différente. La DGD est illustrée à la Figure 1.5.

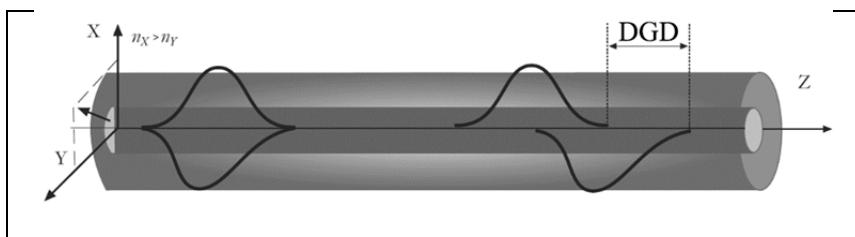


Figure 1.5 Représentation visuelle de la DGD
Adaptée de Ten et Edwards (2010)

La biréfringence dans les fibres optiques peut apparaître aussi bien durant le procédé de fabrication des fibres que pendant leur installation sur le terrain. Le contrôle rigoureux de l'ellipticité du cœur lors de la fabrication des fibres ne s'est réalisé qu'à partir de 1994 puisque c'est à cette période que le phénomène de PMD est devenu perceptible avec l'apparition des premiers systèmes de transmission à 10 Gbit/s et des systèmes sous-marins à longue portée. Les contraintes mécaniques subies par la fibre tendent aussi augmenter la

biréfringence de la fibre et donc la PMD d'après (Keiser, 2010). Ces contraintes peuvent être distinguées en deux catégories : les contraintes internes, liées au processus de fabrication et les contraintes externes, parmi lesquelles on peut citer les paramètres environnementaux des installations (torsion excessive des fibres par exemple).

La DGD est un phénomène qui est dépendant de la longueur d'onde, du temps et est propre à chaque segment de fibre optique constituant la liaison. En effet, chaque segment a sa biréfringence propre, ce qui va entraîner des transferts d'énergie aléatoires entre les différents modes de propagation. Il n'est donc possible d'appréhender la DGD que de manière statistique. On utilise pour cela la dispersion modale de polarisation (*Polarization Mode Dispersion*, PMD), qui correspond à la valeur moyenne de la DGD, et dont l'intensité évolue proportionnellement à la racine carrée de la longueur de fibre parcourue. Chaque fibre optique est caractérisée par un coefficient de PMD qui est exprimé en ps/\sqrt{km} . Une fibre optique standard à un coefficient de PMD de l'ordre de $0.5 ps/\sqrt{km}$. Nous allons nous intéresser dans la section suivante aux causes de la DGD.

1.4.5 La fluctuation des états de polarisation

Les événements transitoires de polarisation, qui correspondent à un changement notable de l'état de polarisation du signal optique durant un court intervalle de temps, trouvent leurs origines dans les vibrations, les travaux à proximité des fibres, le vent ou encore d'autres facteurs environnementaux. Leur étude est primordiale, car elle permet de dimensionner le suiveur de polarisation du transceiver en définissant la vitesse maximale du module de traitement numérique du signal (Roberts et al., 2010). La Figure 1.6 présente deux exemples d'effets de polarisation que l'on peut rencontrer sur une liaison optique. Les tracés vert et bleu correspondent respectivement aux états précédent, et suivant l'événement, dont le tracé est représenté en rouge.

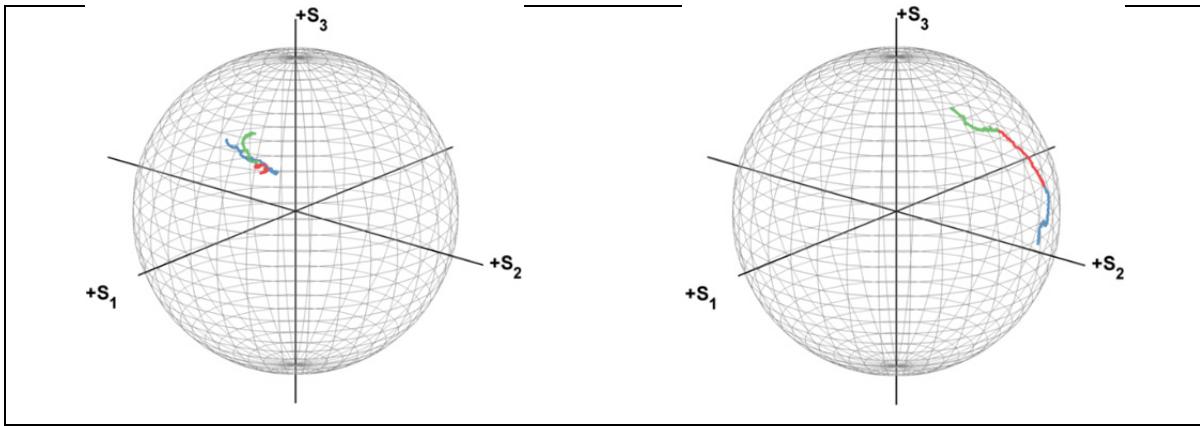


Figure 1.6 Représentation d'un effet rapide de polarisation élastique (gauche) et inélastique (droite) sur la sphère de Poincaré (novembre 2012)

Un événement transitoire de polarisation est défini par convention comme un changement d'état de polarisation supérieur à 10 degrés pendant 100 millisecondes (Nelson et al., 2011). Les événements transitoires de polarisation peuvent être soit élastiques soit inélastiques. Pour chaque événement transitoire, on définit son élasticité par la différence angulaire, sur la sphère de Poincaré, entre l'état initial de polarisation et l'état final de l'événement. La Figure 1.6 présente un événement élastique (à gauche) et un événement inélastique (à droite). À cette élasticité s'ajoute une autre mesure permettant de quantifier l'évolution de l'état de polarisation : il s'agit de la longueur d'arc sur la sphère de Poincaré, ou déflexion angulaire, définie entre deux instants t_1 et t_2 , qui s'exprime de la manière suivante, selon (Bao, Waddy et Chen, 2004) :

$$\gamma(t_1, t_2) = \cos^{-1}[\vec{S}(t_1) \cdot \vec{S}(t_2)] \quad (1.5)$$

où γ désigne la longueur d'arc, \vec{S} le vecteur de Stokes et t_1 et t_2 les deux instants de mesure.

1.5 Monitoring de couche optique

L'un des principaux enjeux de ce projet porte sur le monitoring de la couche optique des réseaux à fibre optique. Face à l'augmentation continue de la capacité de ces réseaux et du niveau de fiabilité requis, il est nécessaire de s'assurer du bon fonctionnement des systèmes de transmission. Le monitoring de la couche optique des réseaux permet d'une part de mieux

comprendre les phénomènes se produisant sur des liaisons longue distance déployées, par opposition aux montages de laboratoire, et d'autre part d'obtenir des informations pertinentes pour la conception des transpondeurs. Cette section présente dans un premier temps les différents paramètres permettant de mesurer la qualité du signal optique ainsi que d'autres paramètres jugés importants. Une brève présentation des outils de monitoring est ensuite proposée.

1.5.1 Mesure de qualité du signal optique

Cette section présente les trois paramètres fondamentaux qui permettent de caractériser la mesure de qualité d'un signal optique : la puissance du signal, le taux d'erreur sur les bits (*Bit Error Rate*, BER) et le rapport signal sur bruit optique. Ces trois paramètres sont corrélés dans une liaison optique amplifiée.

1.5.1.1 La puissance du signal

Le paramètre le plus basique permettant de mesurer la qualité d'un signal optique est la puissance du signal, qui est généralement exprimée en échelle logarithmique (rapport de puissance en décibels entre la puissance mesurée et un milliwatt, dBm). La puissance du signal peut être mesurée dans diverses parties d'une liaison optique, notamment à travers des diviseurs de puissance avec un faible rapport (*taps*) qui se trouvent à divers endroits de la liaison, par exemple au niveau du transmetteur, du récepteur ou des amplificateurs.

1.5.1.2 Le taux d'erreur sur les bits

Le taux d'erreurs sur les bits constitue un indicateur de performance de base pour les liaisons optiques numériques. Il se définit comme étant le rapport, sur une période temporelle donnée, du nombre d'erreurs sur le nombre total de bits transmis. Les erreurs que l'on peut observer interviennent au niveau du récepteur optique, lors du processus de prise de décision sur la valeur d'un bit donné (1 ou 0) après réception du signal. En fonction de la distorsion de signal

observée sur la liaison, causée notamment par le bruit, on peut déterminer une probabilité que la mauvaise décision soit prise quant à la valeur d'un bit transmis.

L'utilisation de codes correcteurs d'erreurs (*Forward Error Correction*, FEC) permet de diminuer la valeur du BER en ajoutant des informations supplémentaires dans les données transmises. (Keiser, 2010) rapporte que l'utilisation de la FEC peut diviser la valeur du BER d'un facteur 10.

1.5.1.3 Le rapport signal sur bruit optique

Le rapport signal sur bruit optique (*Optical Signal to Noise Ratio*, OSNR) correspond au rapport de la puissance maximale du signal et la puissance du bruit. Le rapport signal sur bruit est généralement exprimé en décibel (dB). Dans les liaisons optiques, et plus particulièrement les liaisons WDM amplifiées par des EDFA, la principale source de bruit provient de l'émission de photons issus du phénomène d'amplification spontanée (*Amplified Spontaneous Emission*, ASE).

1.5.2 Paramètres de performance importants

Bien que les deux paramètres précédents offrent une représentation assez fiable de la qualité du signal optique transmis, d'autres paramètres importants sont à prendre en compte.

Un de ces paramètres importants est la dispersion chromatique, qui a été présentée à la section 1.3.2. Le monitoring de la dispersion chromatique constitue un indicateur important dans la mesure où ce paramètre va directement influencer la qualité des impulsions lumineuses transmises. Ce paramètre est susceptible d'évoluer en raison d'opérations de maintenance au niveau de la liaison ou encore de changement de température qui sont notamment significatifs dans les liaisons aériennes (Chan, 2010).

Une autre catégorie importante de paramètres mesurés correspond aux paramètres liés à la polarisation du signal. Le suivi de ces paramètres est de plus en plus critique compte tenu de

l'utilisation croissante de formats de modulation faisant intervenir la polarisation du signal d'une part et de l'augmentation du débit par canal au-delà de 2.5 Gbit/s d'autre part.

Les paramètres critiques liés à la polarisation sont la DGD, la PDL et les composantes de Stokes qui ont été présentées à la section 1.4. Le monitoring de la DGD est important, car ses variations sont aléatoires, dépendantes du temps, des vibrations et de la température (Chan, 2010). Ces derniers paramètres sont susceptibles d'être assez variables dans une liaison aérienne, donnant au monitoring de la DGD un intérêt certain puisqu'une valeur trop importante de la DGD peut engendrer une panne. Le monitoring de la PDL permet quant à lui de diagnostiquer le fonctionnement des différents éléments de la liaison optique.

Le monitoring des paramètres de Stokes permet quant à lui de caractériser la stabilité de l'état de polarisation observé dans la liaison. Il permet notamment de mettre en évidence des événements de polarisation qui, lorsqu'ils sont trop rapides, peuvent entraîner une panne du système de transmission. L'utilisation de ce paramètre permet en outre de mieux comprendre le comportement en polarisation des liaisons optiques.

1.5.3 Outils et instruments de monitoring

Cette section présente brièvement les outils qui peuvent être utilisés pour assurer le monitoring des liaisons optiques.

Le premier dispositif permettant d'assurer le monitoring de la couche optique se trouve au niveau des systèmes de gestion des systèmes de transmission, et permet d'obtenir certains paramètres basiques donnant un aperçu de la qualité du signal optique transmis ou reçu. Les paramètres de base mesurés par ces systèmes sont la puissance du signal ou encore le taux d'erreur sur les bits mesuré avant l'application du code correcteur d'erreur, présenté à la section 1.5.1.2. D'autres instruments génériques peuvent être utilisés comme le puissancemètre, dont le rôle est de mesurer la puissance du signal à un point donné de la liaison optique.

D'autres outils de mesures plus complexes permettent de moniturer le signal optique à un point donné de la liaison optique. Parmi ces outils figure l'analyseur d'un spectre optique (*Optical Spectrum Analyzer*, OSA) dont le rôle est d'afficher les différentes fréquences contenues dans le signal optique ainsi que leurs amplitudes respectives. Il est aussi possible d'afficher le diagramme de l'œil du signal optique afin de tirer des conclusions sur la qualité du signal.

L'autre outil de mesure présentant un intérêt capital pour ce projet est un polarimètre. Un polarimètre est un instrument de mesure permettant de mesurer l'état de polarisation d'un signal optique, c'est-à-dire les quatre composantes du vecteur de Stokes : S_0 , S_1 , S_2 et S_3 . Les mesures ainsi obtenues rendent possible la représentation de l'état de polarisation du signal à chaque instant sur la sphère de Poincaré et de caractériser son évolution.

L'instrument utilisé dans le cadre du projet pour mesurer la polarisation du signal est un synthétiseur de polarisation Agilent N7786B, utilisé en mode polarimètre. Le principe de mesure de l'instrument est basé sur la division du signal à l'entrée du polarimètre en quatre signaux équivalents, qui subissent chacun un filtrage en polarisation, afin d'obtenir quatre niveaux de puissance indépendants (Agilent Technologies, 2013).

L'analyse théorique de la représentation de Stokes permet d'établir des relations permettant de déterminer les quatre composantes de Stokes à partir de quatre mesures de puissance optique indépendantes (Hui et O'Sullivan, 2009). Ces mesures indépendantes peuvent être obtenues en soumettant le signal à différents filtres polarisant et à des lames quart d'onde. Une lame quart d'onde produit un déphasage de 90° entre les deux directions principales de polarisation. La Figure 1.7 présente les quatre mesures indépendantes utilisées pour calculer les composantes de Stokes pour le polarimètre Agilent N7786B.

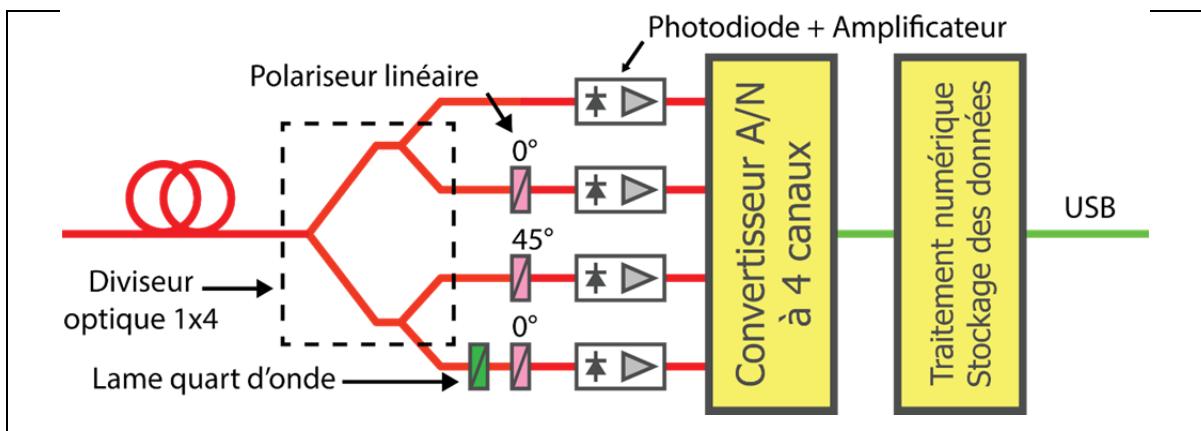


Figure 1.7 Architecture du polarimètre
Adaptée de Agilent Technologies (2013)

Le polarimètre permet deux types d’acquisition de mesure : d’une part l’acquisition en mode rafale des paramètres de Stokes et d’autre part l’acquisition à un seuil déterminé de changement d’état de polarisation.

1.6 Transpondeur cohérent

Cette partie propose dans un premier temps une introduction à la technologie cohérente qui est au cœur des mesures réalisées dans ce projet. Le principe de fonctionnement d’un transpondeur cohérent est ensuite présenté. Ensuite, une revue de littérature sur les travaux antérieurs de monitoring est réalisée afin d’évaluer la performance des transpondeurs cohérents comme moniteurs de performance.

1.6.1 Description de la technologie cohérente

Les récepteurs optiques ont été traditionnellement basés sur la méthode de la détection directe. Cette méthode utilise un photo-courant qui est proportionnel au carré de l’amplitude du champ électrique du signal.

Les phénomènes perturbateurs de communication optiques tels que l’atténuation ou la dispersion chromatique sont principalement des fonctions linéaires du champ électrique.

Ainsi, le passage au carré de la valeur du champ tend à rendre un problème linaire a priori simple en un problème non linéaire. Il est par ailleurs nécessaire, dans le cas de la dispersion, de recourir à des fibres compensatrices, impliquant elle-mêmes le recours à des amplificateurs pour compenser leur atténuation d'après (Roberts et al., 2009).

Ces différents inconvénients peuvent être palliés en ayant recours à la détection cohérente, qui permet non seulement d'accéder à l'amplitude du signal, mais aussi à sa phase, rendant possible le traitement électronique de multiples perturbations à l'aide de modules de traitement numérique du signal. La détection possible de la phase permet aussi d'utiliser un plus vaste ensemble de formats de modulation.

1.6.2 Présentation du transpondeur cohérent

Le transpondeur cohérent utilisé dans le cadre du projet fait partie de la plate-forme optique OM6500 de Ciena. Il s'agit d'un transceiver cohérent, dit OCLD (*Optical Channel Laser and Detector*), fonctionnant à 40 Gbit/s. Le format de modulation utilisé est le DP-QPSK : deux signaux modulés en quadrature par changement de phase (*Quadrature Phase-Shift Keying*, QPSK) sont émis sur chacune des directions principales de polarisation, ce qui permet d'injecter sur une seule fibre optique un total de 4 signaux à 11.5 Gbit/s chacun. Cela représente un débit binaire brut de 46 Gbit/s. Certaines informations, dont les informations de FEC, viennent s'ajouter à l'information transmise, diminuant le débit binaire utile à 40 Gbit/s. Ce format de modulation permet d'atteindre une efficacité spectrale de 2 bits/s/Hz d'après (Roberts et al., 2009). L'architecture du transmetteur DP-QPSK est présentée à la Figure 1.8 ci-dessous. Dans cette figure, on distingue les 4 signaux logiques (D1, D2, D3 et D4) destinés à être transmis. Pour chaque direction de polarisation, les deux signaux logiques à 11.5 Gbit/s chacun sont modulés en QPSK.

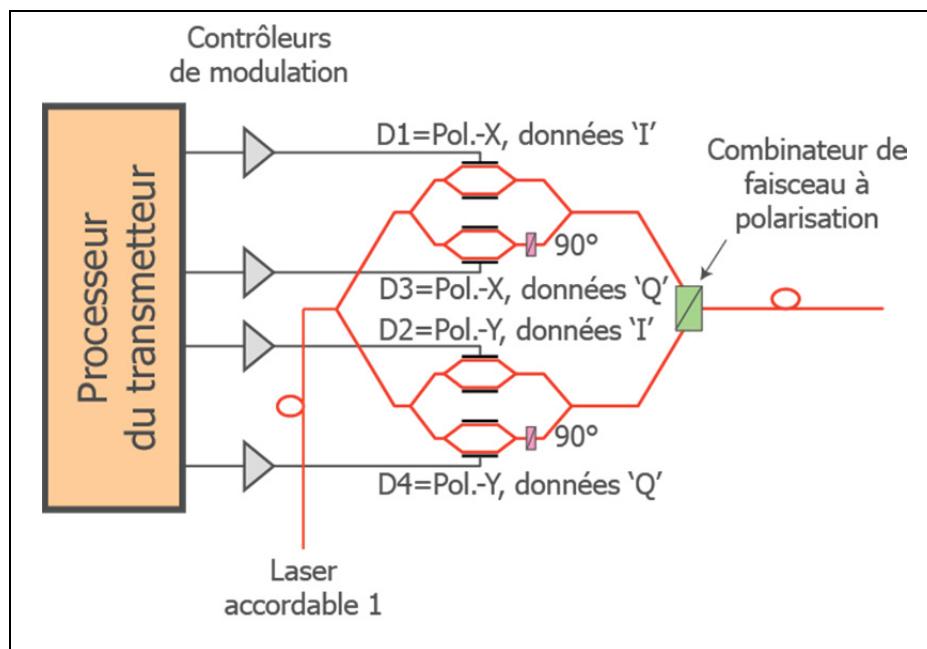


Figure 1.8 Architecture du transmetteur cohérent DP-QPSK
Adaptée de Nelson et al. (2009)

Afin de faciliter la compréhension, un transmetteur QPSK sur une direction de polarisation est présenté à la Figure 1.9. Un modulateur Mack-Zehnder (représenté par un bloc MZ sur la figure), relié à une diode laser accordable en longueur d'onde, permet de convertir chacun de signaux électriques en un signal optique modulé par changement de phase (*Phase-Shift Keying*, PSK). Un coupleur optique assure ensuite le mélange du signal optique I-opt avec le signal Q-opt déphasé de 90° via un déphaseur (représenté par un bloc 90° sur la figure). On obtient ainsi un signal optique QPSK, dont la constellation peut être représentée dans le « plan IQ ». Les informations sur les deux directions principales de polarisation X et Y sont ensuite combinées par l'intermédiaire d'un combinateur de polarisation. On distingue ainsi 4 signaux logiques indépendants : XI, XQ, YI et YQ.

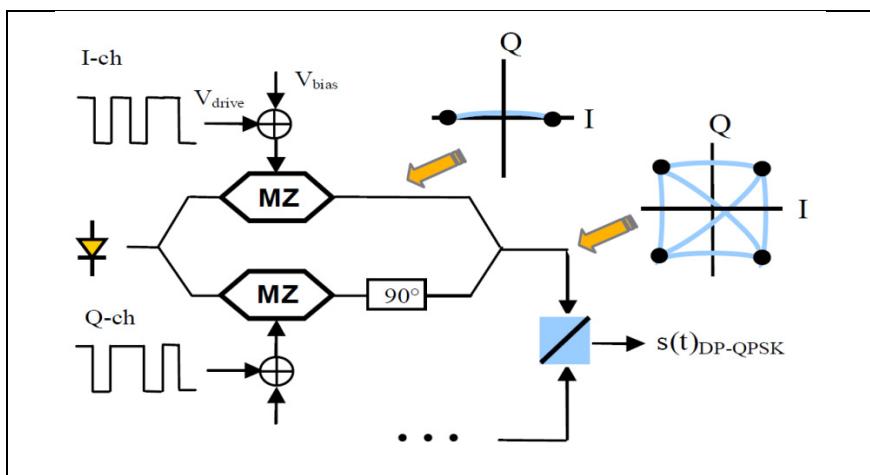


Figure 1.9 Diagramme bloc d'un transmetteur QPSK
Adaptée de Sun, Wu et Roberts (2008)

Le traitement du signal au niveau récepteur se fait en deux étapes correspondant aux domaines physiques de l'optique puis de l'électrique. Le traitement optique consiste à restaurer les signaux optiques Q-opt et I-opt pour chacune des directions principales de polarisation. Le principe de la détection cohérente est de mélanger le signal reçu avec une onde lumineuse émise par un oscillateur local avant de le soumettre au photo-détecteur. Il est ainsi possible de récupérer à la fois l'amplitude et la phase du signal optique.

L'architecture d'un récepteur cohérent est présentée à la Figure 1.10. Deux composants optiques vont rendre possible cette étape : un oscillateur local, accordable en longueur d'onde (laser accordable 2 sur la figure) et un hybride optique à 90° (présenté en vert sur la figure). La fréquence optique de l'oscillateur local est réglée de façon à être très proche de la fréquence de la porteuse optique réglée au niveau du transmetteur (Sun, Wu et Roberts, 2008).

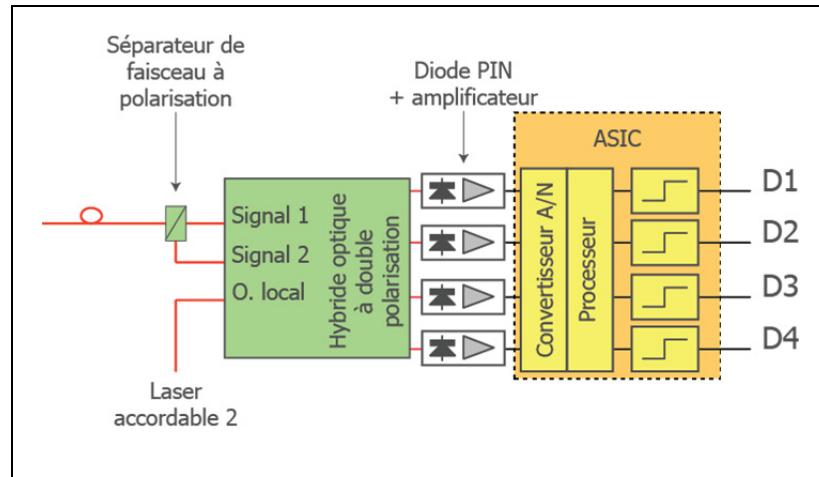


Figure 1.10 Architecture du récepteur cohérent
Adaptée de Nelson et al. (2009)

Le signal optique, une fois syntonisé, est ensuite traité par un ensemble de quatre photodiodes qui vont effectuer la conversion en signal électrique.

1.6.3 Paramètres mesurés aux fins de compensation

Le signal électrique obtenu au niveau du récepteur est traité à travers un module de traitement numérique du signal (*Digital Signal Processing*, DSP) qui implémente un ensemble de trois principales fonctions (Roberts et al., 2009). La Figure 1.11 présente sous forme de diagramme bloc l'architecture du récepteur cohérent. Dans cette figure, ADC désigne un convertisseur analogique-numérique et FIR un filtre à réponse impulsionnelle finie.

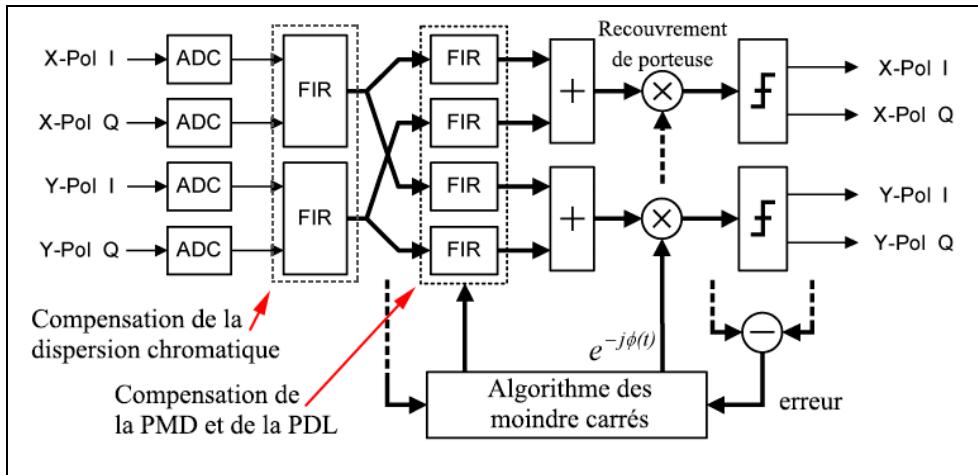


Figure 1.11 Diagramme bloc du traitement du signal optique au niveau du récepteur à 40 Gbit/s de Ciena
Adaptée de Roberts et al. (2009)

La première consiste à compenser la dispersion chromatique à travers un filtre à réponse impulsionnelle finie (*Finite Impulse Response*, FIR). La seconde fonction consiste à compenser la PMD et la PDL à travers une autre série de filtres. La dernière fonction consiste à démoduler les quatre porteuses du signal : les composantes en phase (I) et en quadrature de phase (Q) selon les deux directions principales de polarisation.

L'utilisation de filtres à réponse impulsionnelle finie va aussi permettre l'estimation des différents paramètres compensés pour des fins de monitoring. Il est ainsi possible de récupérer :

- la puissance du canal optique;
- le taux d'erreur sur les bits;
- la dispersion chromatique;
- la PMD;
- la PDL;
- le rapport signal sur bruit électrique selon les deux directions de polarisation;
- l'ensemble des coefficients utilisés pour les filtres compensateurs;
- les trois composantes de Stokes S_1 , S_2 et S_3 .

1.7 Revue des expériences de monitoring

Cette partie propose une revue des différentes expériences de monitoring que l'on peut trouver dans la littérature. Ces expériences permettent d'une part d'orienter l'étude menée dans le cadre de ce projet et d'autre part d'apporter des éléments de comparaison avec les résultats obtenus.

1.7.1 Récapitulatif des expériences de monitoring

Face au nombre important d'études portant sur les effets de polarisation, il semble pertinent de dresser un tableau récapitulatif présentant l'ensemble des expériences de monitoring menées par le passé sur des liaisons optiques, en soulignant les caractéristiques de celles-ci (Tableau 1.1).

Tableau 1.1 Récapitulatif des expériences de monitoring
Adapté de Crawford (2014)

Référence	Type de lien	Instrument Signal	Durée	Objectif
(Bulow et al., 1999)	Boucle enterrée de 52 km	Fréquence de battement avec modulation 30G	36 heures	Caractérisation de la vitesse maximale des filtres égaliseurs de PMD
(Bao, Waddy et Chen, 2004)	Aérien : 34 km Enterré/aérien : 52 km	Polarimètre (supposé)	5 jours	Fluctuations de la polarisation due au vent, variations entre le jour et la nuit
(Krummrich et al., 2005)	Souterrain, 15 segments de 55 km chacun	Polarimètre analogique et oscilloscope	2.5 mois	Caractérisation de la vitesse requise par les compensateurs de PMD
(Boroditsky et al., 2005)	Ville 40 km Banlieue 7 km	Polarimètre (10G)	4 mois sur 2 liens	Montrer les différents types d'événements transitoires
(Poirrier et al., 2008)	210 km, segments enterrés non profondément	Testeur SDH Polarimètre 10G	11 semaines	Démontrer la robustesse de la méthode du maximum de vraisemblance (MLSE)
(Dumas, 2009)	N/A	Polarimètre	N/A	Effet des impacts mécaniques sur les DCM
(Roberts et al., 2009)	N/A	Transpondeur cohérent. 40G, double porteuse	N/A	Caractériser le potentiel et la performance des modules de traitement numérique du signal

Tableau 1.1 Récapitulatif des expériences de monitoring (suite)
Adapté de Crawford (2014)

Référence	Type de lien	Instrument Signal	Durée	Objectif
(Salaun et al., 2009)	1000 km en 13 segments; 450 km en 5 segments	Polarimètre 10G	6 mois	Rapidité des compensateurs optiques de PMD, fluctuations rapides de polarisation
(Nelson et al., 2011)	2 fibres de 900 km en boucle type non précisé	Polarimètre 10G	79 jours sur 4 mois	Valeurs typiques de la vitesse maximale des effets de polarisation
(Woodward et al., 2012)	3 liaisons à 2 fibres enterrées, pre-1994	Transpondeur cohérent 40G	Jusqu'à 500 heures	Observation de : SOP, DGD, PDL, BER (technologie cohérente)
(Woodward et al., 2014)	3 liaisons (2 fibres par liaison) enterrées, pre-1994	Transpondeur cohérent 40G	Plus de 10000 heures sur 18 mois	Observation de : SOP, DGD, PDL, BER (technologie cohérente)

1.7.2 Monitoring de liaison commerciale avec un transpondeur cohérent

Une première expérience de monitoring dont l'objectif était de quantifier le comportement des fibres commerciales déployées en termes de polarisation, notamment à travers l'influence de la température et des contraintes mécaniques, a été réalisée par (Woodward et al., 2012). Il s'agit de l'une des premières expériences utilisant les transpondeurs cohérents comme outils de mesure. Les paramètres mesurés étaient la DGD, la PDL, le BER précédant la correction d'erreur et la première composante de Stokes (S_1) afin d'en déduire la composante ϕ , définie par $\phi = \cos^{-1}(S_1)$, permettant d'obtenir l'activité cette polarisation.

L'étude apporte des innovations par rapport à la ligne étudiée : il s'agit de liaisons enterrées sans module de compensation de dispersion, réparties en plusieurs segments et étudiées dans les deux directions. Chacun de ces segments a une architecture variable : passage en zone urbaine, présence de multiples amplificateurs et une longueur variant de 30 à 170 km. Les mesures ont été effectuées à la fois sur des échantillons à la minute et à la seconde sur des périodes dépassant les 400 heures. L'analyse de l'évolution temporelle des différents paramètres permet de conclure qu'il y a une corrélation entre l'activité des différents paramètres d'une part et que le comportement d'un même type de fibre peut radicalement varier radicalement selon la situation de déploiement.

Une seconde étude, menée par (Woodward et al., 2014), fait état des résultats de l'étude de la même liaison, mais sur plus long terme : les mesures s'étendent sur 18 mois. L'étude porte l'emphase sur l'étude de la composante φ et de la DGD. Le protocole de mesure utilisé est la capture des mesures toutes les secondes sur une période de 2.6 heures puis le transfert des données mesurées. Une moyenne à la minute est ensuite obtenue par interpolation linéaire. L'analyse sur des longues périodes a permis de réaliser des autocorrélations afin de détecter des tendances dans les mesures. Dans les résultats de l'analyse, il apparaît que l'activité de polarisation et les valeurs de PMD ont un temps de polarisation de l'ordre de quelques jours et que la distribution des valeurs de DGD peut être approchée par une distribution de Maxwell.

1.7.3 Monitoring de liaison commerciale avec un polarimètre

Face à l'augmentation des débits et de la complexification des formats de modulation, les liaisons optiques sont devenues de plus en plus sensibles aux variations de la polarisation du signal. En particulier, les récepteurs cohérents doivent assurer le suivi de l'état de polarisation du signal afin d'assurer la continuité de la transmission (Szafraniec, Marshall et Nebendahl, 2013). Un des paramètres critiques de l'élaboration d'un transpondeur est sa tolérance aux variations brusques d'état de polarisation. L'objectif des expériences de monitoring est de caractériser les évolutions possibles et de les quantifier afin d'être en

mesure d'anticiper les pannes que peuvent engendrer les variations de polarisation. Il est donc nécessaire de caractériser l'évolution de la polarisation du signal optique et de chercher l'origine de phénomènes pouvant perturber le suivi du signal au niveau du récepteur.

Une autre étude a porté sur l'évolution temporelle de l'état de polarisation de plusieurs liaisons optiques, enterrées et aériennes, munies de modules de compensation de dispersion. Ces mesures s'accompagnent aussi de la caractérisation de la PMD et de la PDL observées dans ces routes fibrées. L'évolution des états de polarisation est mesurée par l'intermédiaire d'un polarimètre mesurant les paramètres de Stokes à une fréquence de 4 kHz (Peterson et al., 2002). L'outil utilisé afin de caractériser les évolutions les plus importantes de l'état de polarisation est un histogramme des vitesses de rotation angulaires maximales observées toutes les millisecondes de mesure. Par ailleurs, ces expériences ont permis de constater que certaines valeurs maximales des paramètres de Stokes sont anormalement élevées, car séparées de la distribution obtenue. Ces mesures sont en réalité des artefacts de mesure résultant de la méthode de mesure utilisée par les instruments, en particulier de convertisseur analogique numérique utilisé. Une étude proposée par (Leo et al., 2003) met l'emphase sur ce phénomène et propose une méthode de correction, basée sur la recherche de période temporelle durant laquelle au moins deux paramètres de Stokes subissent une variation inférieure à un certain seuil.

L'analyse de l'histogramme des évolutions des vitesses angulaires permet de constater que les observations réalisées à la fois sur des liaisons commerciales enterrées et des liaisons en laboratoire à l'aide d'un brouilleur de polarisation suivant une loi de probabilité de Rayleigh, qui résulte de la combinaison de deux distributions de la loi normale indépendantes. En revanche, les observations réalisées sur des fibres aériennes de type câble de garde à fibres optiques (*Optical Ground Wire*, OPGW) sur pylônes électriques à haute tension ne suivent pas la seule loi de Rayleigh et sont donc influencées par d'autres facteurs d'après (Peterson, Leo et Rochford, 2004).

L'influence des paramètres environnementaux tels que la température et le vent sur l'activité de polarisation des liaisons optiques ont fait l'objet de plusieurs études. (Bao, Waddy et Chen, 2004) proposent l'étude de l'activité de polarisation d'une fibre aérienne de 34 km à travers l'évolution temporelle de la longueur d'arc, définie à l'équation (1.5), et ces données sont ensuite comparées à l'évolution de la température et du vent dans la région traversée par la fibre. Une corrélation de ces derniers paramètres avec l'activité de polarisation est alors établie. Les travaux de (Leeson, Bao et Côté, 2009) ont porté sur une ligne OPGW d'une longueur de 33 km et située au Canada. Des mesures ont été effectuées en été et en automne sur cette liaison, munie d'un bouclage. Les mesures ont été échantillonnées à 1kHz à l'aide d'un analyseur de polarisation Agilent 8509 et numérisé par une carte d'acquisition National Instruments PCI-6034E. L'étude de l'évolution de la longueur d'arc des états de polarisation a permis d'établir aussi une corrélation entre des facteurs extérieurs tels que la température ou le vent et l'activité de polarisation du signal optique.

D'autres études ont mis l'emphase sur les effets de polarisation et leur classification. Ainsi, (Boroditsky et al., 2005) proposent une analyse de la fluctuation des états de polarisation sur deux systèmes de transmission commerciaux situés dans une zone urbaine et une zone de banlieue dans l'agglomération New-Yorkaise sur une période de trois jours. L'appareil de mesure utilisé est un polarimètre fonctionnant en mode déclenché. Le taux d'échantillonnage utilisé est fixé à 50 Hz. L'analyse permet de dresser un catalogue des différentes variations rapides de polarisation qui peuvent être réparties en 3 grandes catégories : les effets rapides et élastiques, les effets lents et élastiques et les effets lents et inélastiques. Un effet de polarisation est qualifié de rapide dès lors que sa durée est de l'ordre de la milliseconde, similairement, un effet lent est de l'ordre de la centaine de millisecondes. L'analyse présentée propose aussi certaines explications sur l'origine des effets de polarisation : les événements rapides résultent probablement de facteurs tels que les vibrations mécaniques dues à la circulation automobile à proximité des fibres tandis que les événements les plus lents résultent probablement des interventions humaines dans les locaux des opérateurs.

Une autre expérience a été réalisée plus récemment sur une période couvrant 79 jours, et sur une liaison commerciale de 900 km utilisée en boucle, par (Nelson et al., 2011). La liaison est constituée de 12 segments et comporte des fibres à compensation de dispersion (*Dispersion Compensating Fiber*, DCF). L'appareil utilisé est un polarimètre avec un taux d'échantillonnage de 5 kHz fonctionnant en mode déclenché. Cette expérience a permis d'observer plus de 500 événements rapides de polarisation sur la période de 79 jours. La vitesse de rotation du vecteur de Stokes de l'événement de polarisation le plus rapide observé est de 300 rad/ms. Un nuage de points faisant apparaître la longueur d'arc entre les états de polarisation et début et en fin d'événement en fonction de la vitesse de rotation maximale du vecteur de Stokes est présenté et permet de constater qu'il n'y a pas de corrélation évidente entre ces deux paramètres.

1.8 Présentation du lien optique étudié et des données mesurées

La liaison optique étudiée dans le cadre du projet, illustrée à la Figure 1.12, est une liaison commerciale exploitée par l'opérateur Verizon. Il s'agit d'une liaison de 1020 km constituée de 750 km de fibre installée dans un câble de garde à fibres optiques OPGW, au sommet de pylônes électriques à haute tension, et de 270 km de câble enterré dans un conduit à proximité d'un chemin de fer. La liaison optique étudiée est composée de 14 sections de fibres amplifiées par des EDFA pour une longueur totale de 1020 km. La liaison inclut 2 ROADM mais aucun compensateur optique de dispersion, la compensation électronique de dispersion est utilisée. La fibre optique constituant la liaison est une fibre monomodes G.652, de type NDSF, dont le coefficient d'atténuation est de l'ordre de 0,20 dB/km, le coefficient de dispersion de l'ordre de 17 ps/(nm.km) et un coefficient de PMD de l'ordre de $0,2 \text{ ps}/\sqrt{\text{km}}$. Les terminaux situés aux deux extrémités de la liaison sont équipés de transpondeurs DP-QPSK à 40 Gbit/s de l'équipementier Ciena.

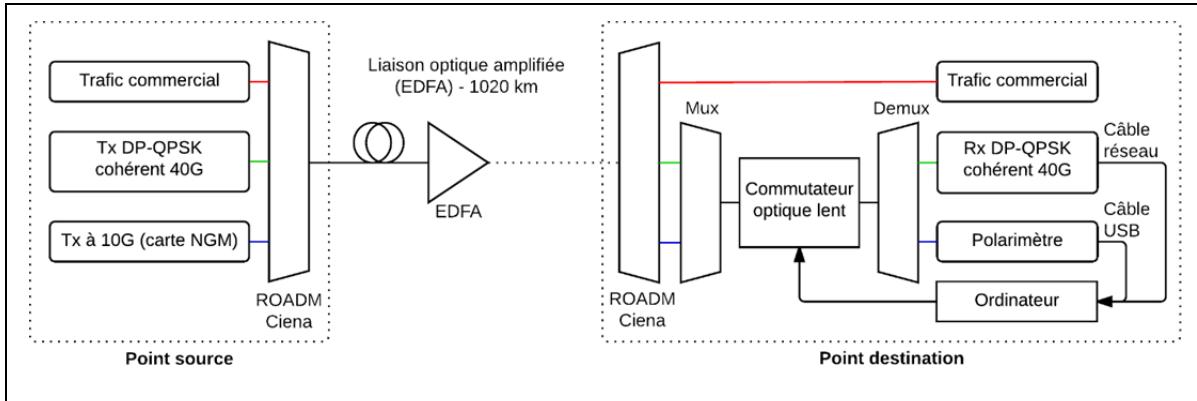


Figure 1.12 Schéma du montage expérimental utilisé

Deux instruments de mesure sont utilisés pour le monitoring dans le cadre du projet :

- le premier est un transpondeur cohérent DP-QPSK à 40 Gbit/s fourni par Ciena. Le canal optique utilisé est centré sur une longueur d'onde de 1530.33 nm;
- le second est un polarimètre Agilent N7786B utilisant un signal optique fourni par un transmetteur Ciena NGM à 10 Gbit/s (*New Generation Modem*) fourni par Ciena. Le signal transmis est polarisé linéairement. Le canal optique utilisé est centré sur la longueur d'onde de 1531.15 nm.

Ces deux signaux, qui sont combinés avec le trafic commercial (constitué d'autres canaux à 100 Gbit/s) dans un multiplexeur WDM avant d'être injectés dans la liaison OPGW commerciale. Les canaux WDM utilisés pour le monitoring correspondent à deux canaux adjacents sur la grille WDM 100 GHz définie par l'union internationale des télécommunications (*International Telecommunication Union*, ITU). L'expérience de monitoring réalisée dans le cadre du projet présente une grande opportunité puisqu'une liaison optique commerciale a été mise à notre disposition par Verizon sur une durée dépassant les deux ans. Il a ainsi été possible d'étudier l'évolution des différents paramètres caractérisant la liaison sur plusieurs saisons.

La Figure 1.13 offre une représentation des données mesurées par le transpondeur cohérent et le polarimètre dans le cadre du projet.

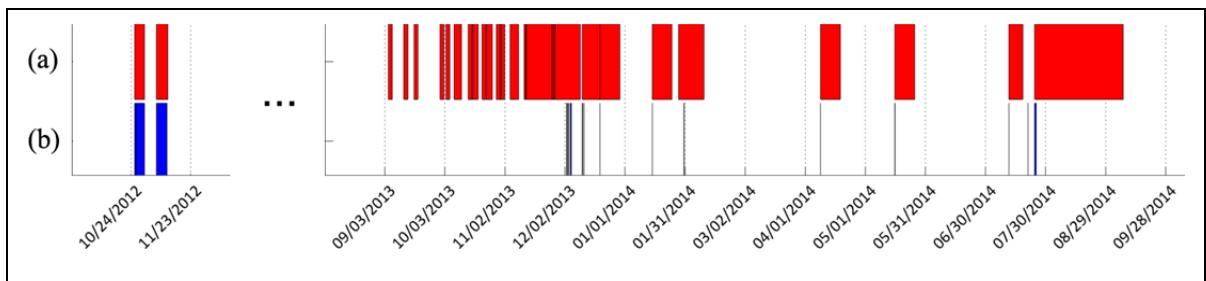


Figure 1.13 Représentation visuelle des mesures réalisées dans le projet par (a) le polarimètre et (b) le transpondeur cohérent

1.9 Conclusion du chapitre

La capacité de monitoring d'un transmetteur cohérent a été mise à profit dans une expérience de monitoring d'une liaison aérienne sur le réseau de Verizon pendant près de deux ans. Un polarimètre a été utilisé en parallèle pour mesurer l'activité de polarisation du lien. Notre projet permet de répondre au besoin de comprendre les phénomènes associés aux fluctuations d'état de polarisation. Nous avons vu que ces fluctuations ont fait l'objet de multiples études sur des liaisons enterrées. On peut logiquement s'attendre à ce que ces fluctuations soient plus importantes sur une liaison aérienne. Le monitoring de la liaison de Verizon nous offre donc une grande opportunité pour mieux comprendre les fluctuations des états de polarisation ainsi que l'évolution temporelle des différents paramètres caractérisant la liaison. Le chapitre suivant porte sur la méthode de traitement des données de monitoring.

CHAPITRE 2

MÉTHODE DE TRAITEMENT DES DONNÉES DE MONITORING

2.1 Introduction

Ce chapitre décrit la méthode qui a été suivie pour traiter les données issues des deux instruments de mesure utilisés dans le cadre du projet de recherche : le transpondeur cohérent et le polarimètre. Un des principaux défis posés par le projet est la quantité importante de données recueillies et le temps de calcul nécessaire pour traiter ces données qui sont issues des différentes mesures. L'ensemble des fichiers recueillis dans le cadre du projet représente un total de plus de 320 000 fichiers pour le polarimètre et de plus de 25 000 fichiers pour le transpondeur cohérent représentant une taille totale de 1,6 téraoctet. Face à un tel volume de données, il a été nécessaire d'organiser les données de manière à pouvoir faciliter à la fois l'accès aux données et d'accélérer leur temps de traitement.

2.2 Les données issues du transpondeur cohérent

Cette partie présente d'une manière générale le traitement des données liées au transpondeur cohérent, qui a été présenté à la section 1.6. Les différents paramètres mesurés par celui-ci sont introduits dans un premier temps, puis l'algorithme de traitement des données du transpondeur cohérent est présenté.

2.2.1 Description des paramètres mesurés et méthode d'acquisition des données

Le transpondeur cohérent, à travers ses filtres d'égalisation, permet d'estimer de manière fiable les différents effets linéaires se produisant sur une liaison optique donnée (Woodward et al., 2008). Le transpondeur cohérent utilisé dans le cadre du projet nous permet de mesurer différents paramètres avec un taux d'échantillonnage de 1 kHz.

Compte tenu de la vitesse élevée d'acquisition requise, la capture des paramètres de liaison ne peut pas se faire en continu sur une longue durée. En effet, l'écriture des données mesurées occupe un temps non négligeable et l'acquisition des paramètres doit être suspendue lors de l'opération de sauvegarde. L'évolution temporelle d'un cycle d'acquisition est représentée dans la Figure 2.1.

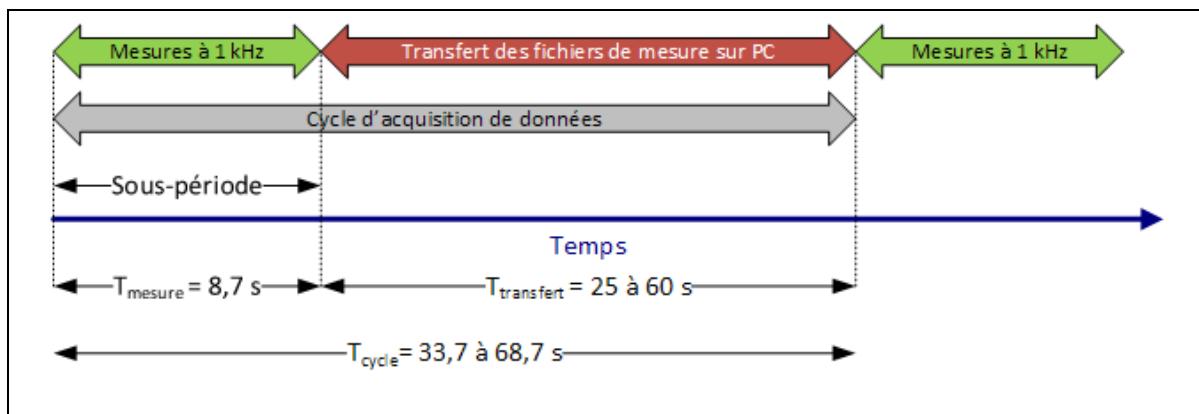


Figure 2.1 Chronogramme du cycle d'acquisition des mesures du transpondeur cohérent

Le montage permettant de récupérer les données du transpondeur cohérent est présenté à la Figure 1.12. Ce montage repose sur une carte d'acquisition, développée par Ciena pour des besoins spécifiques, directement reliée au circuit électronique du transpondeur cohérent. Cette carte est reliée à l'ordinateur stockant les mesures par un câble réseau et est configurée pour récupérer en continu les paramètres monitorés par le transpondeur à une vitesse d'échantillonnage réglable. Les paramètres récupérés sont transmis sous format binaire à l'ordinateur à travers l'interface réseau. Un programme installé sur le serveur convertit alors ces fichiers binaires en fichiers de format CSV (*Comma-Separated Values*, format de stockage de données en format texte). Le protocole de mesure ainsi défini apporte une contrainte : le transpondeur cohérent doit être réinitialisé après chaque acquisition de 8,7 s. L'initialisation de la carte va entraîner l'apparition de perturbations sur les valeurs mesurées, résultant du processus de récupération du signal perdu lors de la réinitialisation.

Par la suite, on désignera par *période de mesure* du transpondeur cohérent une période temporelle de mesure continue. Chacune des périodes de mesure est divisée en *sous-périodes*, que l'on peut voir à la Figure 2.1, d'une durée maximale de $T_{\text{mesure}} = 8,7 \text{ s}$. Une sous-période correspond à un fichier de mesure, au format CSV, sur le serveur de stockage.

2.2.2 Traitement initial des données du transpondeur cohérent

Ma contribution a porté sur l'optimisation des scripts développés par (Mortelette, 2014) et le développement de nouveaux scripts afin d'améliorer la vitesse de traitement des données. Le traitement initial des données se divise en quatre étapes présentées à la Figure 2.2. Une étape préliminaire au traitement, qui ne sera pas détaillée ici, consiste à calculer les valeurs correctes de PMD à partir des coefficients des coefficients utilisés par le transpondeur cohérent pour les filtres compensateurs. Le script permettant d'effectuer ce calcul est fourni par Ciena et doit être utilisé avant d'entreprendre tout traitement sur les données du transpondeur cohérent. Un script de traitement initial a ainsi été développé sous le nom de *ocld_processing_columns* et est présenté à l'*ANNEXE I*, p.145. Ce script couvre les trois premières étapes indiquées à la Figure 2.2.

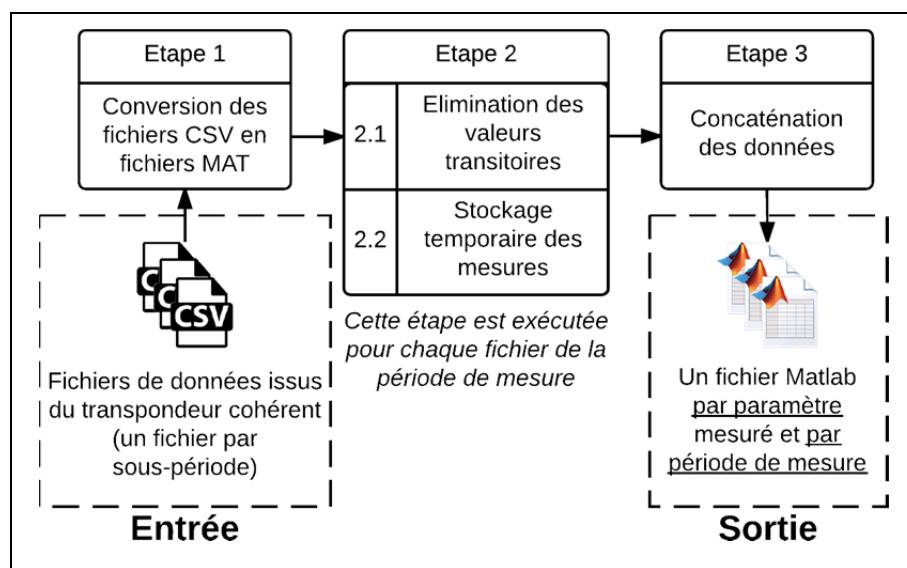


Figure 2.2 Étapes du traitement initial des données du transpondeur cohérent

Afin de faciliter l'accès et l'organisation des différents fichiers de mesure issus du transpondeur cohérent, ceux-ci sont organisés, avant de procéder au traitement initial des données, en dossiers correspondant à chaque période de mesure. Le nom des dossiers est choisi de telle sorte que le nom de l'instrument de mesure et la période, exprimée au mois près, soient accessibles rapidement. Le nom des dossiers est de la forme : ‘*2013 12-19 to 2014 04-08 Recordings_OCLD*’, où la date du 19 décembre 2013 correspond au début de la période de mesure et la date du 8 avril 2014 correspond à la fin de la période de mesure.

Le nom de chacun des fichiers CSV issus du transpondeur cohérent est divisé en deux parties séparées par un tiret : la première partie correspond à un compteur incrémental du nombre de fichiers pour une sous-période donnée et le second correspond à un horodatage propre à Matlab, exprimé par un nombre réel. Par convention, on considère ce nombre comme la date de la première mesure figurant dans le fichier. Par exemple, pour un fichier en date du 19 décembre 2013 à 09h40 et 10 secondes, Matlab générera le nom de fichier suivant : ‘*I_735587.4029_21.csv*’, où ‘*735587.4029*’ correspond à l'horodatage Matlab de la date.

Chaque étape du traitement initial des données, présentée à la Figure 2.2, est décrite dans les sous-sections suivantes.

2.2.2.1 Conversion des fichiers en structures Matlab

Telle que présentée à la Figure 2.2, la première étape du traitement est la conversion des fichiers CSV, « Entrée » sur la figure, en structures au format Matlab (MAT, format binaire), offrant un meilleur temps d'accès ainsi qu'une réduction de taille de fichier. Cette conversion s'effectue par l'intermédiaire d'une première fonction Matlab, *convert_csv_mat* présenté à l'ANNEXE II, p.151, dont le rôle est de lire de manière séquentielle chaque fichier CSV et de le convertir en une structure Matlab, sous forme d'une matrice dont les colonnes correspondent aux données mesurées (voir Tableau 2.1). La matrice obtenue est ensuite enregistrée sous le format MAT, en conservant un nom identique à celui du fichier CSV afin de conserver les informations sur l'horodatage. Cette étape permet aussi de supprimer

certaines mesures qui ne sont pas utiles dans le cadre du projet tels que la dispersion chromatique ou encore les coefficients des filtres compensateur, qui ne sont utilisés que pour calculer la PMD avant de commencer le traitement.

Tableau 2.1 Paramètres monitorés par le transpondeur cohérent

Nom de l'index (fichier CSV)	Description
Timestamp	Valeur d'horodatage
inst_PMD	PMD instantanée
inst_PDL	PDL instantanée
PreFECBER	Taux d'erreur sur les bits avant FEC
SNR_X	Rapport signal sur bruit électrique (polarisation X)
SNR_Y	Rapport signal sur bruit électrique (polarisation Y)

Tableau 2.1 Paramètres monitorés par le transpondeur cohérent (suite)

Nom de l'index (fichier CSV)	Description
S ₀	Intensité totale du signal
S ₁	Composante de Stokes S ₁
S ₂	Composante de Stokes S ₂
S ₃	Composante de Stokes S ₃
Ch_Pwr	Puissance optique observée au niveau du récepteur
Hxx, Hyy, Hxy, Hyx	Coefficients des filtres numériques à réponse impulsionnelle finie

La valeur d'horodatage nécessite un traitement spécifique : l'horodatage de la première mesure correspond à la valeur d'horodatage contenu dans le titre du fichier et les points suivants sont obtenus en incrémentant de 1 ms ligne par ligne, correspondant à la période d'échantillonnage des mesures du transpondeur cohérent. Le changement le plus significatif apporté par rapport à l'approche de (Mortelette, 2014) consiste à utiliser une fonction interne à Matlab, *csvread*, qui permet d'accélérer la conversion des fichiers CSV en fichiers MAT.

2.2.2.2 Traitement des valeurs transitoires et stockage temporaire des résultats

La seconde étape de traitement consiste à supprimer les valeurs transitoires situées au début de chaque sous-période. Ces valeurs transitoires, qui sont inhérentes au protocole d'acquisition des données, entraînent l'apparition de valeurs négatives ou anormalement élevées de certains paramètres. La méthode utilisée pour détecter la fin des données transitoires est de chercher, pour chaque sous-période, le dernier point de mesure du fichier correspondant présentant soit des valeurs négatives pour la PMD et la PDL, soit une valeur de BER égale à 1 due à la perte de signal. Le changement effectué par rapport à l'approche de (Mortelette, 2014) porte sur le changement de stratégie pour détecter la fin de la période transitoire de mesure : la nouvelle approche évite la lecture complète des mesures de chaque sous-période et permet donc d'augmenter la vitesse de traitement. Une fonction Matlab a été créée pour rechercher le premier point de mesure valide : *find_glitch_end*, présentée à l'ANNEXE III, p.153. Cette seconde étape s'achève par le stockage temporaire des paramètres mesurés exploitables, c'est-à-dire sans valeur transitoire, en vue de les fusionner avec les mesures des autres sous-périodes.

2.2.2.3 Concaténation des données traitées

La troisième étape de traitement consiste à regrouper, paramètre par paramètre, l'ensemble des mesures sur une période donnée. Afin de rendre le traitement flexible et d'optimiser la consommation en mémoire, chacun des paramètres est stocké dans un fichier Matlab qui lui est propre, contrairement à l'approche de (Mortelette, 2014). À titre d'exemple, la période de mesure du 5 au 11 novembre 2012 correspond à 116 829 245 points de mesures après suppression des valeurs transitoires et l'ensemble des seules valeurs de BER sur cette période occupe un espace de 195 Mo en format Matlab, qui se traite plus facilement qu'un fichier de taille 10 fois supérieure. Le traitement d'une période de mesure fournit ainsi 10 fichiers Matlab, correspondant aux mesures suivantes : Timestamp, inst_PMD, inst_PDL, PreFECBER, SNR_X, SNR_Y, S1, S2, S3, Ch_Pwr. L'ensemble des fichiers obtenus constitue la « Sortie » présentée à la Figure 2.2.

2.2.3 Moyennage des paramètres mesurés par le transpondeur cohérent

La quatrième étape de traitement initial des données du transpondeur cohérent consiste à moyenner les valeurs mesurées à la seconde ou à la minute. En effet, le transpondeur à 40 Gbit/s échantillonne ses mesures à une fréquence de 1 kHz soit un point de mesure toutes les millisecondes et la majeure partie des tracés se font sur une période temporelle dépassant la dizaine de jours. Il est donc nécessaire d'obtenir des points de mesure moyennés à la seconde, la minute, voire l'heure pour procéder à des analyses d'évolution temporelle. A titre de comparaison, les travaux de monitoring d'une liaison optique commerciale par (Woodward et al., 2014) s'effectuent avec des valeurs moyennées à la minute.

Ma contribution a porté sur l'élaboration d'un script Matlab, *compute_mean_ocl*, présenté à l'ANNEXE IV, p.155. Ce script permet de calculer la moyenne et l'écart type des paramètres mesurés pour chaque seconde de mesure. Le script prend en paramètre le résultat du script *traitement_global_ocl*. Le calcul se divise en deux étapes : la première consiste à répartir les valeurs d'horodatage des mesures, pour une période donnée, en sous-groupes correspondant chacun à une seconde de mesure. Cette étape de répartition, qui est présentée Figure 2.3, se base sur les valeurs d'horodatage calculées précédemment. Ces dernières étant communes à l'ensemble des mesures dans le cas du transpondeur cohérent, la segmentation peut être réutilisée pour chacun des paramètres.

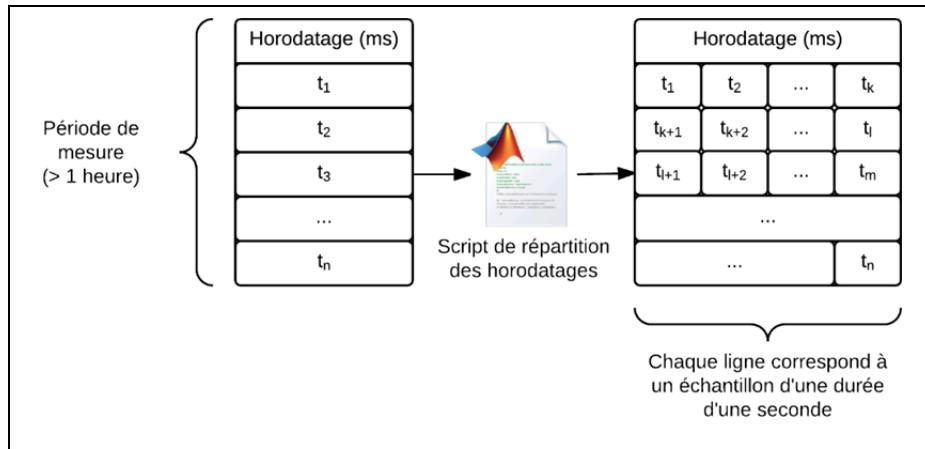


Figure 2.3 Première étape du moyennage à la seconde : répartition des valeurs d'horodatage

La seconde étape, présentée Figure 2.4, consiste à calculer, pour chaque seconde de mesure définie précédemment, la moyenne et l'écart type des valeurs mesurées pendant cette seconde.

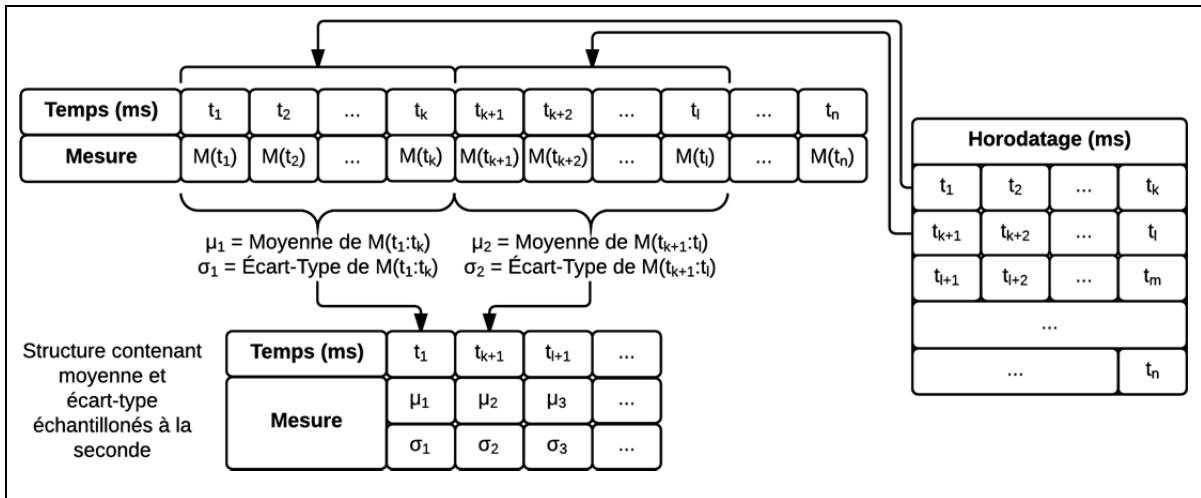


Figure 2.4 Seconde étape du moyennage à la seconde : calcul des valeurs moyennes et des écarts types

La formule menant au calcul de la moyenne \bar{x} de l'échantillon est donnée par l'équation (2.1),

$$\bar{x} = \frac{1}{n} \sum_{k=1}^n x_k \quad (2.1)$$

où n désigne le nombre de points de mesure dans le sous-échantillon et x_k la k-ième mesure de l'échantillon. La formule menant au calcul de l'écart type s de l'échantillon est donnée par les équations (2.2),

$$s = \sqrt{\sum_{k=1}^n \frac{(x_k - \bar{x})^2}{n-1}} = \sqrt{\frac{1}{n-1} \left[\sum_{k=1}^n x_k^2 - n \cdot \bar{x}^2 \right]} \quad (2.2)$$

où \bar{x} désigne la moyenne de l'échantillon, n désigne le nombre de points de mesure dans le sous-échantillon et x_k la k-ième mesure de l'échantillon.

Il est ensuite possible de calculer la moyenne et l'écart type d'un paramètre donné sur une période globale. Considérons deux échantillons notés 1 et 2 de taille respective n_1 et n_2 , de moyenne respective \bar{x}_1 et \bar{x}_2 , et d'écart types respectifs σ_1 et σ_2 . Alors l'ensemble contenant l'union des éléments de 1 et des éléments de 2 à une taille $n = n_1 + n_2$, une moyenne donnée par $\bar{x} = (n_1 \cdot \bar{x}_1 + n_2 \cdot \bar{x}_2) / n$ et un écart type σ donné par $\sigma = ((n_1 - 1) \cdot \sigma_1^2 + (n_2 - 1) \cdot \sigma_2^2 - n \cdot \bar{x}^2) / (n - 1)$. Ce résultat peut se généraliser pour calculer la moyenne \bar{x} d'un ensemble de sous-échantillons, à travers l'équation (2.3).

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n n_i \cdot \bar{x}_i \quad (2.3)$$

où n désigne le nombre de sous échantillons, n_i désigne le nombre de points de mesure dans le i-ième sous-échantillon et \bar{x}_i la moyenne du i-ième sous échantillon. De la même manière, on peut calculer l'écart type σ d'un ensemble de sous échantillons à travers l'équation (2.4)

$$\sigma = \frac{1}{n-1} \left(\sum_{i=1}^n (n_i - 1) \cdot \sigma_i^2 + n_i \cdot \bar{x}_i^2 - n \cdot \bar{x}^2 \right) \quad (2.4)$$

où n désigne le nombre de sous échantillons, n_i désigne le nombre de points de mesure dans le i -ième sous-échantillon, \bar{x}_i la moyenne du i -ième sous échantillon, σ_i l'écart type du i -ième sous échantillon et \bar{x} la moyenne de l'ensemble des sous échantillons, donnée par l'équation (2.3).

Cette formule permet ainsi d'évaluer la moyenne et l'écart type de grands échantillons sans nécessiter de les stocker intégralement en mémoire, ce qui est un atout pour les calculs réalisés dans ce projet, car les données brutes, c'est-à-dire non moyennées, occupent une place importante en mémoire.

La troisième et dernière étape de traitement du script consiste à estimer la valeur absolue de la dérivée temporelle des différents paramètres. Pour cela, l'équation (2.5) est appliquée en utilisant la fonction *diff*, interne à Matlab et calculant la différence entre les éléments adjacents d'un vecteur, où Δt correspond à 1 milliseconde. La valeur approchée de la dérivée est ensuite moyennée à la minute en utilisant le même algorithme que pour la valeur moyenne de la mesure.

$$\left| \frac{dx}{dt} \right| = \left| \frac{x(t + \Delta t) - x(t)}{\Delta t} \right| \quad (2.5)$$

Suite à ce traitement, l'ensemble des paramètres est stocké dans un fichier unique puisque le nombre global de points de mesures est au moins divisé par 1000 s'il y a échantillonnage à la seconde.

2.2.4 Utilisation du calcul parallèle

La mise en place du calcul parallèle est au cœur de ma contribution dans le cadre de l'amélioration du script de traitement initial des données et calculs subséquents (présentés aux chapitres 3 et 4) du transpondeur cohérent. La méthode de traitement initiale proposée par (Mortelette, 2014) est peu adaptée au traitement d'un grand volume de données, en termes de temps de traitement. L'approche adoptée dans notre projet met l'emphase sur l'optimisation des scripts afin de réduire le temps de traitement.

Le calcul parallèle, par définition, consiste à utiliser simultanément plusieurs ressources de calcul pour résoudre un problème informatique (Blaise Barney, 2014). L'approche conventionnelle en programmation consiste à exécuter un ensemble de tâches en série, qui ne sont exécutées que sur une seule unité de calcul. Cette dernière approche est celle adoptée par défaut dans Matlab et dans les travaux menés précédemment. L'implémentation du calcul parallèle dans Matlab se fait par l'intermédiaire de la boîte à outils *Parallel Computing Toolbox*TM qui permet de paralléliser les tâches, sur une seule machine équipée d'un processeur multi-cœurs et donc de plusieurs unités de calculs. La boîte à outils permet de lancer à partir de l'interface principale de Matlab un ensemble de moteurs de calcul en arrière-plan (voir Figure 2.5). Le nombre maximal de moteurs de calculs appartenant à l'ensemble ainsi créé correspond au nombre de cœurs présents sur le processeur de l'ordinateur effectuant le calcul. Par la suite, on désignera par parc de calcul l'ensemble des moteurs de calcul utilisés pour un calcul donné.

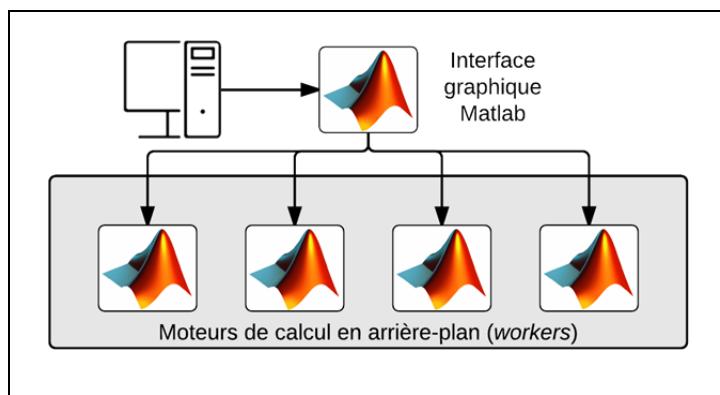


Figure 2.5 Représentation visuelle des moteurs de calcul parallèle

La technique utilisée dans le cadre du projet est l'approche *single program, multiple data*, notée SPMD, qui consiste à donner à l'ensemble des unités de calcul un même programme, mais avec des paramètres d'entrée différents (Blaise Barney, 2014). Les paramètres d'entrée correspondent, dans le cadre du projet, à une liste de fichiers à traiter : il est donc nécessaire de partitionner l'ensemble des fichiers pour une période donnée en n sous-ensembles, où n correspond au nombre d'unités de calcul. L'opération de partitionnement s'effectue à travers un script Matlab portant le nom de *partitionnement_parallel*, et présenté à l'ANNEXE V, p.161, prenant en entrée une liste de fichiers et donnant en sortie k sous-ensembles de fichiers répartis de manière égale, où k correspond au nombre de moteurs de calcul. Le fonctionnement du script de partitionnement est illustré à travers la Figure 2.6.

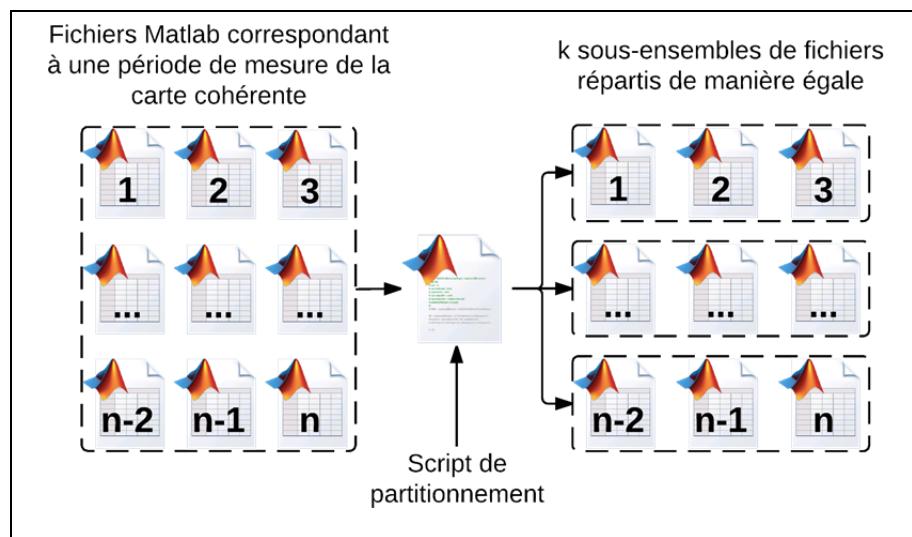


Figure 2.6 Fonctionnement du script de partitionnement

Le calcul parallèle des données du transpondeur cohérent est présenté à la Figure 2.7.

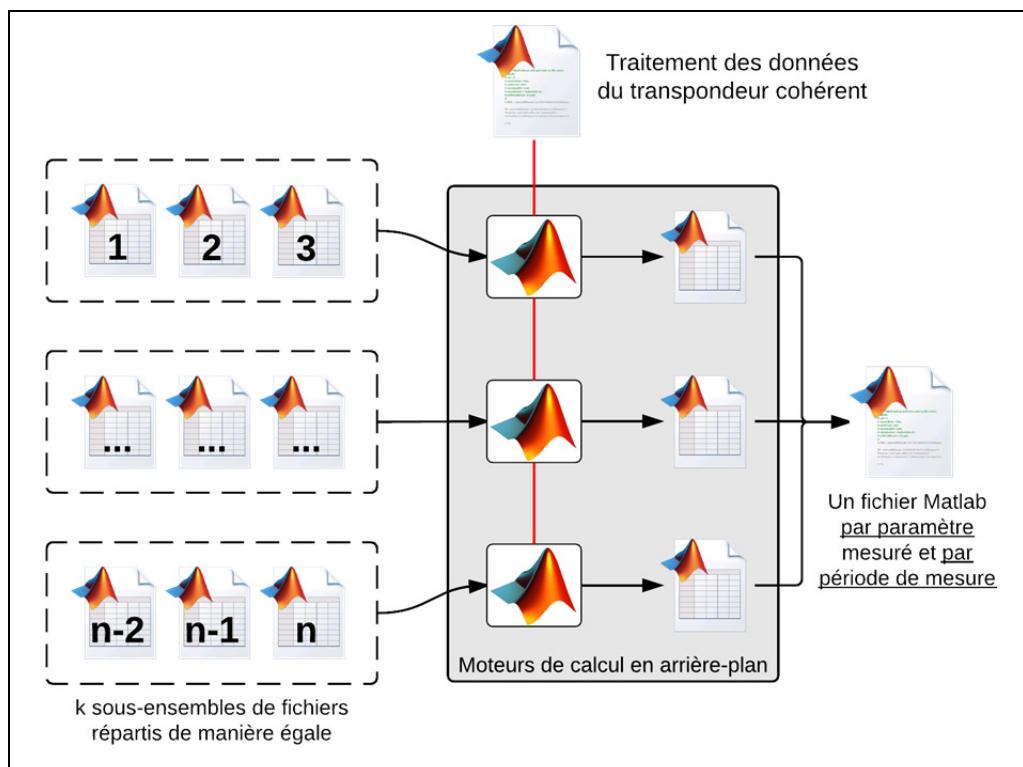


Figure 2.7 Calcul parallèle et fusion des données du transpondeur cohérent

Chaque moteur de calcul assure le traitement initial des données du transpondeur cohérent, correspondant aux étapes 2 et 3 de la Figure 2.2, en travaillant sur son propre sous-ensemble de fichiers, défini par le script de partitionnement. L'étape finale du calcul parallèle consiste à concaténer les matrices obtenues par chacun des moteurs de calcul, matérialisées par la sortie présentée à la Figure 2.2, en vue d'obtenir un fichier Matlab par paramètre et par période de mesure.

2.3 Les données issues du polarimètre

Cette partie présente d'une manière générale le traitement des données mesurées par le polarimètre. Le polarimètre a été présenté à la section 1.5.3. Dans un premier temps, les paramètres mesurés par le polarimètre sont décrits. Ensuite, la méthode de traitement des données du polarimètre est présentée.

2.3.1 Description des paramètres mesurés et méthode d'acquisition des données

Le polarimètre est relié par un câble USB à un ordinateur assurant le paramétrage de l'acquisition des données de polarisation. Dans le cadre du projet, une acquisition continue est choisie, avec un taux d'échantillonnage de 20 kHz. Cependant, comme dans le cas du transpondeur cohérent, le processus d'écriture de données sur un disque dur rend impossible une capture continue. L'instrument est configuré pour effectuer des mesures d'une durée de 10 secondes, ce qui correspond à 200 000 points de mesure par fichier. Le chronogramme des mesures effectuées par le polarimètre est présenté à la Figure 2.8. Les données capturées sont directement consignées dans un fichier Matlab lors du processus d'enregistrement, permettant ainsi de faciliter le traitement des données. Chacun de ces fichiers Matlab contient, pour l'ensemble mesures instantanées de la période correspondante au fichier, les données suivantes :

- le degré de polarisation du signal;
- la puissance optique du signal, exprimée en dBm;
- les trois paramètres de Stokes.

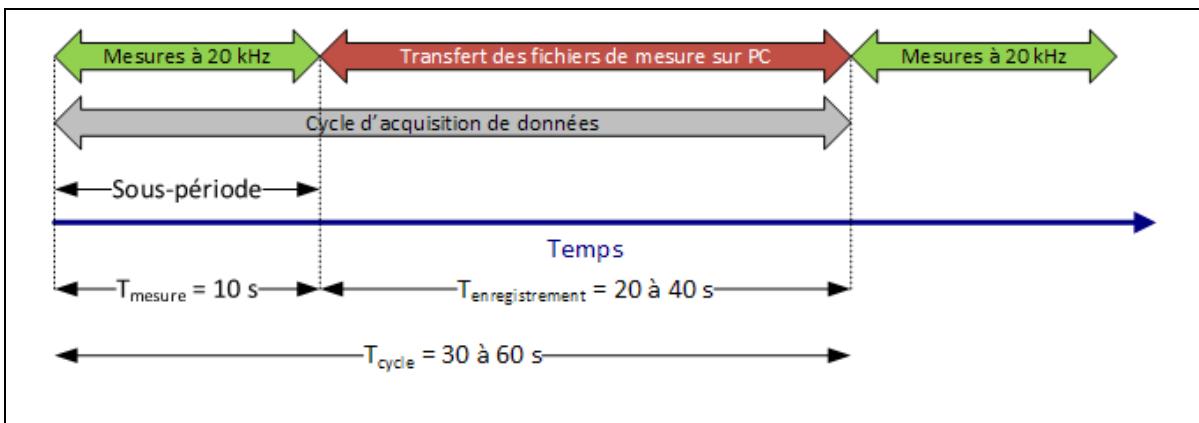


Figure 2.8 Chronogramme du cycle d'acquisition des mesures du polarimètre

Contrairement aux mesures du transpondeur cohérent, une *sous-période* du polarimètre a une durée maximale $T_{\text{mesure}} = 10 \text{ s}$. Une sous-période correspond à un fichier de mesure, au format MAT, sur le serveur de stockage.

2.3.2 Traitement initial des données du polarimètre

Ma contribution a porté dans un premier temps sur l'optimisation des scripts développés précédemment par (Mortelette, 2014) et de mettre en place le calcul parallèle pour gérer efficacement le volume important de données à traiter : l'ensemble des mesures de polarimètre exploitées dans le cadre du projet correspond à environ 180 jours de mesure, répartis en 483 811 fichiers occupant un espace total dépassant le téraoctet.

Le nom des fichiers de mesures enregistrés par le polarimètre est décomposé en 7 parties séparées chacune par un tiret du bas. La première correspond à un index incrémental pour numérotter les fichiers. Cet index est suivi par l'année, le mois, le jour, l'heure, la minute et enfin la seconde du début d'acquisition des paramètres de Stokes. Un exemple de fichier de mesure au format MAT est '1_2012_11_5_14_36_44.mat', qui correspond à un fichier enregistré le 5 novembre 2012 à 14 heures, 36 minutes et 44 secondes.

De la même manière que pour le transpondeur cohérent, chaque période de mesure est consignée dans un dossier qui lui est propre et le dossier est nommé de telle sorte qu'on puisse accéder rapidement à la période de mesure au jour près et à l'instrument utilisé. Le

nom d'un dossier de mesures du polarimètre est de la forme : '2012 11-05 to 11-11 Recordings_Polarimeter', où la date du 5 novembre 2012 correspond au début de la période de mesure et la date du 11 novembre 2012 correspond à la fin de la période de mesure.

Les étapes de traitement des données du polarimètre sont présentées à la Figure 2.9. Chacune des étapes est décrite dans les sous-sections suivantes. Le script de traitement initial développé porte le nom de *pola_processing_stats* et est présenté à l'ANNEXE VI, p.163.

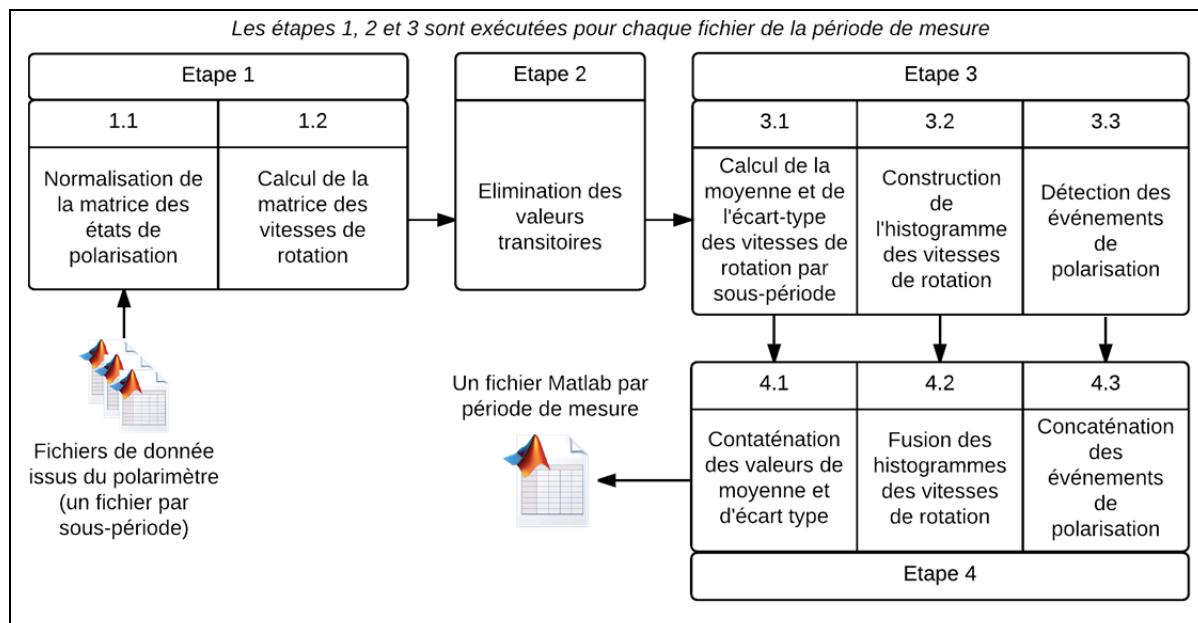


Figure 2.9 Étapes du traitement initial des données du polarimètre

2.3.2.1 Normalisation de la matrice des états de polarisation et calcul de la matrice des vitesses de rotation du vecteur de Stokes

La structure de données contenant les mesures des composantes de Stokes se présente comme une matrice à trois colonnes (une colonne par composante) et dont le nombre de lignes correspond au nombre de mesures de la sous période (200 000 points de mesure par sous-période). Chaque matrice, correspondant à une sous-période, comprend donc 200 000 lignes et 3 colonnes. Une représentation de cette matrice est présentée à l'équation (2.6).

$$Mat(S) = \begin{bmatrix} S_{1,t} & S_{2,t} & S_{3,t} \\ S_{1,t+\Delta t} & S_{2,t+\Delta t} & S_{3,t+\Delta t} \\ \vdots & \vdots & \vdots \\ S_{1,t+k\Delta t} & S_{2,t+k\Delta t} & S_{3,t+k\Delta t} \end{bmatrix} \quad (2.6)$$

où $S_{i,t}$ représente la valeur de la i -ième composante de Stokes à l'instant t , Δt représente le temps d'échantillonnage du polarimètre, en l'occurrence $50 \mu s$ et $k = 199\,999$. La valeur de chacune des composantes S_1 , S_2 et S_3 de Stokes est ensuite divisée par la puissance totale du signal. La formule utilisée pour la normalisation est donnée par l'équation (2.7).

$$\overrightarrow{S_{norm}} = \frac{\vec{S}}{\sqrt{S_1^2 + S_2^2 + S_3^2}} \quad (2.7)$$

où \vec{S} désigne le vecteur de Stokes non normalisé, $\overrightarrow{S_{norm}}$ le vecteur de Stokes normalisé et $\sqrt{S_1^2 + S_2^2 + S_3^2}$ la puissance optique totale du signal lumineux. La formule est appliquée pour chaque ligne de la matrice S_{norm} . Le calcul suivant porte sur la matrice des vitesses de rotation du vecteur de Stokes, correspondant à la vitesse de rotation du vecteur dans la sphère de Poincaré et définie par la formule (2.8), adaptée de (Bao, Waddy et Chen, 2004),

$$\frac{\gamma(t + \Delta t)}{\Delta t} = \frac{\cos^{-1}[\vec{S}(t) \cdot \vec{S}(t + \Delta t)]}{\Delta t} \quad (2.8)$$

où \vec{S} désigne le vecteur de Stokes, Δt à la période d'échantillonnage des mesures et $\frac{\gamma(t + \Delta t)}{\Delta t}$ la vitesse de rotation du vecteur de Stokes. Dans le cadre du projet, Δt est exprimée en millisecondes pour obtenir une matrice des vitesses de rotation du vecteur de Stokes en rad/ms.

Le calcul de la matrice des vitesses de rotation du vecteur de Stokes se fait à partir du vecteur de Stokes normalisé et est obtenu en effectuant le produit scalaire du vecteur de Stokes à

l'instant t avec le vecteur de Stokes à l'instant $t+\Delta t$. L'opération matricielle permettant d'obtenir la matrice des vitesses de rotation angulaires est représentée dans l'équation (2.9),

$$\text{Matrice des vitesses de rotation} = \begin{bmatrix} S_{1,t} & S_{2,t} & S_{3,t} \\ S_{1,t+\Delta t} & S_{2,t+\Delta t} & S_{3,t+\Delta t} \\ \vdots & \vdots & \vdots \\ S_{1,t+k\Delta t} & S_{2,t+k\Delta t} & S_{3,t+k\Delta t} \end{bmatrix}. \quad (2.9)$$

$$\begin{bmatrix} S_{1,t+\Delta t} & S_{2,t+\Delta t} & S_{3,t+\Delta t} \\ S_{1,t+2\Delta t} & S_{2,t+2\Delta t} & S_{3,t+2\Delta t} \\ \vdots & \vdots & \vdots \\ S_{1,t+(k-1)\Delta t} & S_{2,t+(k-1)\Delta t} & S_{3,t+(k-1)\Delta t} \end{bmatrix} / \Delta t$$

où l'opérateur « . » correspond au produit élément par élément, Δt à la période d'échantillonnage, en l'occurrence $20 \mu s$ et k à la taille de la matrice contenant les paramètres de Stokes, autrement dit le nombre de mesures contenues dans la matrice. La fonction Matlab assurant le calcul de la matrice des vitesses de rotation du vecteur de Stokes est intitulée *pola_angular_velocity_matrix* et est présentée à l'ANNEXE VII, page 173.

2.3.2.2 Traitement des valeurs transitoires

La deuxième étape de traitement, telle que présentée dans la Figure 2.2, consiste à détecter et supprimer les effets transitoires associés au commutateur optique lent. La méthode utilisée pour éliminer les données non valides associées à la période transitoire consiste à calculer la matrice des vitesses de rotation angulaires et à déterminer la valeur maximale observée durant la sous-période. En effet, les effets transitoires entraînent l'apparition de variations rapides des composantes de stokes résultant par une vitesse de rotation du vecteur de Stokes très élevée. L'instant correspondant à cette valeur maximale est considéré comme étant la fin de la période transitoire. Une marge de 50 points de mesure, correspondant à 2,5 millisecondes, est ajoutée. Cette étape entraîne, en moyenne, la suppression de 265 points de mesure par fichier ce qui représente environ 0,13 % de mesures par fichier. Pour valider le bon fonctionnement de ce processus, le script enregistre, pour chaque sous-période, l'indice correspondant au premier point de mesure des mesures valides, la valeur de vitesse angulaire

correspondante et la valeur maximale de la vitesse sur l'ensemble des données valides de cette sous-période.

2.3.2.3 Calcul des paramètres statistiques par fichier, construction de l'histogramme des vitesses de rotation du vecteur de Stokes et détermination des événements de polarisation

La troisième étape du traitement consiste à calculer, pour chaque sous-période de mesure, la valeur moyenne et l'écart type des valeurs des vitesses de rotation du vecteur de Stokes. Ces paramètres sont calculés sur les parties valides de mesures, c'est-à-dire excluant les valeurs transitoires. Chacun de ces paramètres est enregistré dans une matrice colonne. Il est ainsi possible de savoir à quel fichier correspond une mesure donnée à partir de la position de cette mesure dans la liste des fichiers traités, représentée par une matrice dans le script.

L'étape qui suit consiste à calculer l'histogramme des vitesses de rotation du vecteur de Stokes. Pour cela, la fonction *histc* de Matlab est utilisée. Cette fonction prend en paramètre une matrice colonne contenant les valeurs dont on souhaite tracer l'histogramme et une matrice colonne contenant les valeurs des intervalles correspondant aux colonnes de l'histogramme. La valeur rentrée est une matrice colonne contenant les occurrences correspondantes à chacun des intervalles, tel que défini à l'équation (2.10), tirée de la documentation de Matlab, où x_k^* désigne le k-ième élément de la matrice contenant la valeur des intervalles (un intervalle correspond à une barre de l'histogramme), x désigne une valeur quelconque de vitesse de rotation du vecteur de Stokes et n_k le k-ième élément de la matrice de sortie.

$$x_k^* \leq x \leq x_{k+1}^* \Rightarrow n_k = n_k + 1 \quad (2.10)$$

L'étape finale du traitement consiste à calculer histogramme normalisé des valeurs de rotation du vecteur de Stokes. Les étapes suivies pour la normalisation de l'histogramme des vitesses de rotation sont présentées à la Figure 2.10.

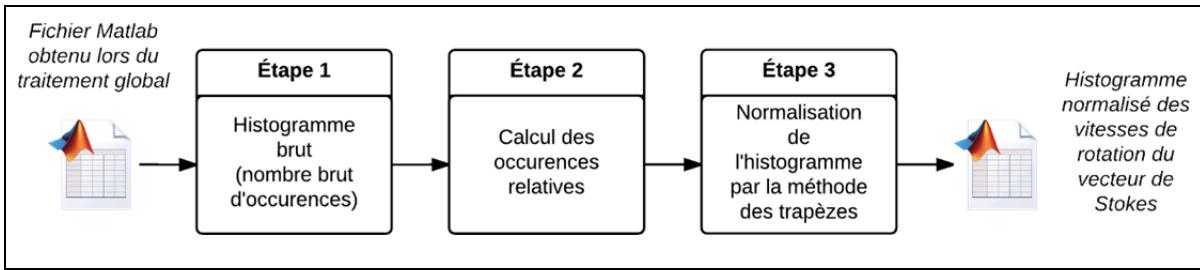


Figure 2.10 Étapes de normalisation de l'histogramme des vitesses de rotation du vecteur Stokes

La normalisation consiste à faire en sorte que l'aire située sous l'histogramme soit égale à 1 afin de préparer les ajustements statistiques qui seront effectués ensuite. Cette propriété provient de la définition d'une densité de probabilité, qui est présentée à l'équation (2.11), tirée de (Walpole, 2011), où f désigne une densité de probabilité.

$$\int_{-\infty}^{+\infty} f(x) dx = 1 \quad (2.11)$$

Les mesures de l'histogramme étant discrètes, il est nécessaire d'avoir recours à une méthode numérique afin d'estimer l'aire délimitée par l'histogramme en vue de le normaliser. La méthode utilisée est la méthode des trapèzes, définie par l'équation (2.12), tirée de (Kreyszig, 2011), où J désigne l'intégrale que l'on souhaite approcher, f la fonction dont on souhaite évaluer l'intégrale entre les points a et b , h la largeur de barre de l'histogramme et x_k^* l'abscisse de la k -ième barre de l'histogramme.

$$J = \int_a^b f(x) dx = h \cdot [f(x_1^*) + f(x_2^*) + \dots + f(x_n^*)] \quad (2.12)$$

2.3.2.4 Détection des événements de polarisation

La dernière étape de cette phase est la détection et l'analyse des variations rapides d'état de polarisation, qui ont été définies dans le premier chapitre. L'objectif de cette analyse est

d'obtenir une matrice contenant les informations présentées dans le Tableau 2.2. Dans le tableau, les indices sont indexés sur la matrice des vitesses de rotation du vecteur de Stokes.

Tableau 2.2 Valeurs de sortie du script de détection des événements de polarisation

Horodatage du fichier contenant l'événement
Indice correspondant au début de l'événement
Indice correspondant à la fin de l'événement
Valeur maximale de la rotation angulaire durant l'événement
Valeur maximale de la déflexion angulaire
Valeur de la déflexion angulaire en fin d'événement

La déflexion angulaire, ou longueur d'arc, est notée γ . Il s'agit, en tout instant d'un événement, de la variation angulaire, en termes de vecteur de Stokes, entre l'état de polarisation au début de l'événement et l'état de polarisation en cet instant. La formule permettant de calculer la déflexion angulaire est donnée à l'équation (1.5). Une définition visuelle de la longueur d'arc entre deux vecteurs de Stokes est proposée à la Figure 2.11, où le tracé en bleu correspond à la longueur d'arc, notée γ , entre les vecteurs de Stokes rouge et vert.

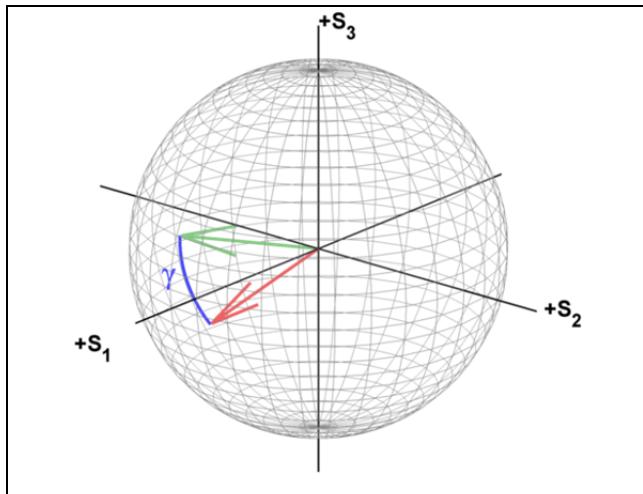


Figure 2.11 Représentation de la longueur d'arc (en bleu) entre deux vecteurs de Stokes (représentés en rouge et en vert)

Une fonction Matlab, intitulée *pola_compute_events*, a été créé afin d'assurer la détection des événements de polarisation et est présenté à l'ANNEXE VIII, page 175. Cette fonction prend en paramètre la matrice des vitesses de rotation du vecteur de Stokes, la matrice contenant les valeurs des paramètres de Stokes et l'index du début de mesures valides. La fonction retourne une structure Matlab dont les colonnes contiennent les informations présentées au Tableau 2.2.

L'algorithme de détection des événements de polarisation est présenté à l'ANNEXE IX, page 179. Cet algorithme utilise trois paramètres définis pour l'ensemble du projet : les deux premiers paramètres correspondent aux seuils de détection d'événements rapides et sont fixés à 0,9 rad/ms et 0,45 rad/ms et le troisième paramètre est la fenêtre temporelle dans laquelle deux valeurs de vitesse de rotation du vecteur de Stokes supérieurs à 0,9 rad/ms sont considérées comme formant un seul événement rapide de polarisation. Ce paramètre est fixé à 10 millisecondes (Crawford et al., 2014).

On peut ainsi définir un événement rapide de polarisation de la manière suivante : le début d'un effet rapide de polarisation est, dans une fenêtre temporelle de 10 ms, le premier instant dans lequel la valeur de vitesse de rotation du vecteur de Stokes dépasse 0,9 rad/ms. La fin de

l'effet rapide de polarisation est le dernier pic dans cette même fenêtre temporelle, où la valeur de vitesse de rotation du vecteur de Stokes dépasse 0,9 rad/ms. Afin de ne pas tronquer artificiellement un événement, on considère qu'il commence quand la vitesse de rotation du vecteur de Stokes dépasse 0,45 rad/ms, qui correspond à la moitié du premier seuil de 0,9 rad/ms, avant le premier pic à 0,9 rad/ms et qu'il se termine quand la vitesse de rotation du vecteur de Stokes revient en dessous de 0,45 rad/ms.

Afin de faciliter la compréhension de cette définition, la Figure 2.12 présente l'application de cette définition à un exemple d'événement. Dans la figure, les deux lignes verticales vertes représentent le début et la fin de l'événement. Les deux lignes horizontales rouges représentent les deux seuils de vitesse de rotation du vecteur de Stokes fixés à 0,45 rad/ms et 0,9 rad/ms. On observe le premier pic de la vitesse de rotation à $t = 0,5$ ms et le dernier pic à $t = 10,3$ ms.

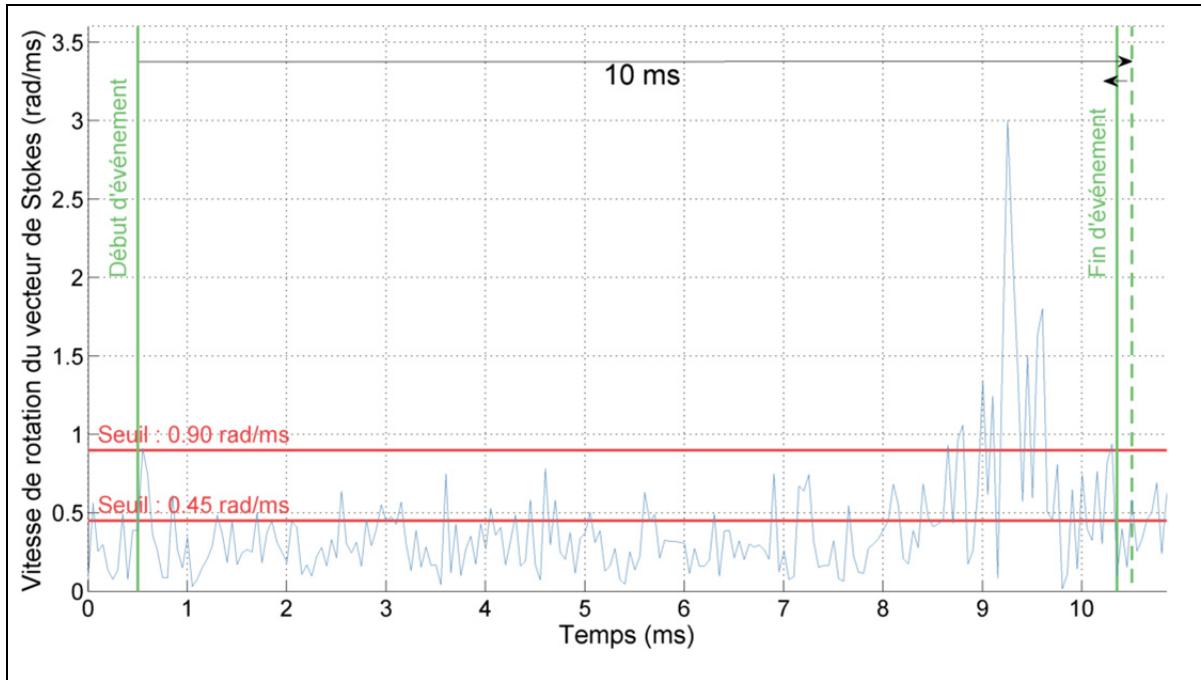


Figure 2.12 Application de l'algorithme à un effet rapide de polarisation

2.3.2.5 Fusion des données

La quatrième et dernière étape du traitement (Figure 2.9) spécifique à chaque fichier consiste à concaténer les matrices contenant les paramètres statistiques, à fusionner les histogrammes des vitesses de rotation du vecteur de Stokes et enfin à concaténer les matrices correspondant aux événements de polarisation.

2.3.3 Moyennage des paramètres mesurés par le polarimètre

Compte tenu du nombre important de fichiers de mesures du polarimètre et du taux d'échantillonnage élevé de l'instrument, à savoir 20 kHz soit un point de mesures toutes les 50 µs, un traitement spécifique a été nécessaire pour calculer la moyenne des trois paramètres de Stokes à la minute. En effet, sur une période équivalente et pour un paramètre, le stockage des données du polarimètre occupe en moyenne 20 fois plus de place que les données du transpondeur.

Ma contribution a porté sur la création d'un script *compute_mean_pola*, présenté à l'ANNEXE X, page 181, permettant un traitement de données du polarimètre. Le processus de traitement des données est décrit à la Figure 2.13. Ce processus doit être exécuté pour chaque période de mesure, telle que définie dans la section 2.3.1.

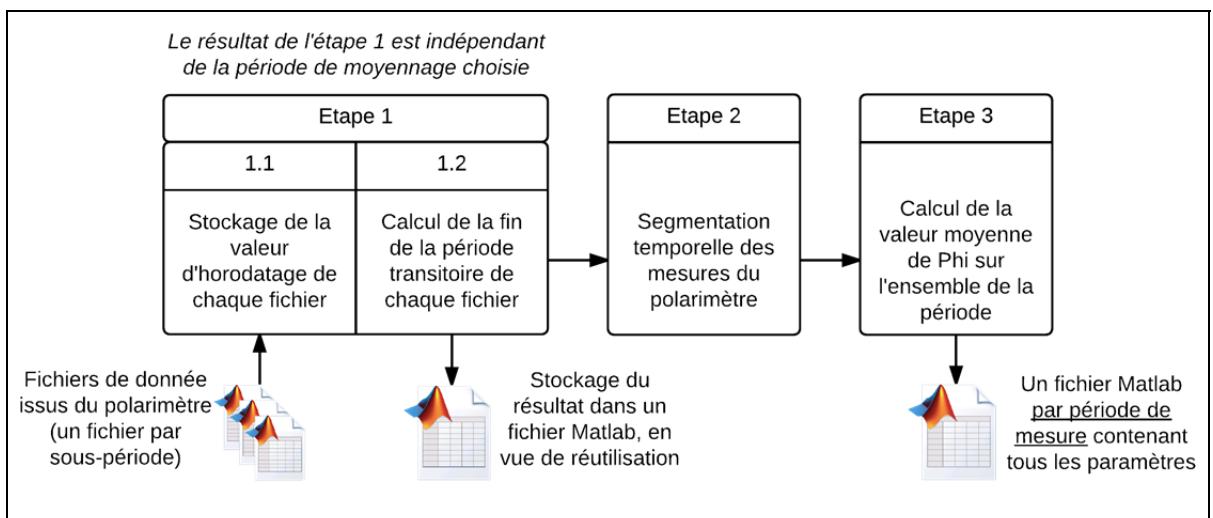


Figure 2.13 Étapes de traitement pour l’analyse temporelle des mesures du polarimètre

La première étape consiste à récupérer l’ensemble des valeurs d’horodatage des fichiers du polarimètre dans une matrice Matlab, c’est-à-dire de convertir les noms des fichiers Matlab en valeurs d’horodatage, puis à identifier, pour chaque fichier, la durée de la période transitoire présente au début de chaque sous-période. La méthode utilisée pour déterminer la durée de la période transitoire, qui est présentée à la section 2.2.2.2, produit une structure Matlab contenant, pour chaque fichier de la période, l’indice correspondant à l’instant de fin de la période transitoire. Cet indice correspond à la ligne de la matrice des paramètres de Stokes à partir de laquelle les mesures sont valides. La première étape est déjà réalisée lors du traitement initial des données du polarimètre : les données alors obtenues sont réutilisées.

La seconde étape consiste à diviser en bloc d’une minute une période de mesure du polarimètre. Notons t_1 et t_2 les valeurs d’horodatage d’une période de mesure du polarimètre. Dans un premier temps, l’intervalle de temps $[t_1, t_2]$ est divisé en sous-intervalles d’une durée d’une minute. Il est ensuite nécessaire de déterminer l’ensemble des points de mesure appartenant à chacun de ces intervalles. La segmentation temporelle est assurée par la fonction *segmentation_temporelle_pola* développée dans le cadre de ce projet et présentée à l’ANNEXE XI, page 185. Cette fonction trie d’abord les fichiers par ordre chronologique à l’aide de la matrice contenant les valeurs d’horodatage, calculée lors de la première étape.

Les points de mesure valides sont ensuite distribués dans chacun des blocs d'une minute définis précédemment.

La troisième et dernière étape consiste à lire de manière incrémentale la structure précédemment créée et d'ouvrir séquentiellement les fichiers de mesure du polarimètre concernés afin de calculer, pour chaque bloc d'une minute, la valeur moyenne de S_1 , S_2 et S_3 . L'avantage de cette méthode par rapport à celle utilisée pour les mesures du transpondeur cohérent, présentée à la Figure 2.2, est que les résultats intermédiaires du traitement ne sont pas stockés sous forme de fichiers Matlab, comme la « Sortie » présentée à la Figure 2.2.

2.3.4 Utilisation du calcul parallèle

L'utilisation du calcul parallèle des données du polarimètre a permis d'observer une diminution de temps de traitement significative par rapport à l'approche de (Mortelette, 2014). En effet, l'utilisation de plusieurs moteurs de calcul a permis d'exploiter au maximum les capacités de traitement fournies par les processeurs multi-cœurs.

Le principe de partitionnement est le même que celui utilisé décrit à la section 2.2.4. En revanche, la fusion des données diffère dans la mesure où les opérations à exécuter consistent à la fois à fusionner des matrices (dans le cas de la matrice contenant les valeurs moyennes et les écarts types par sous période) et à les sommer (dans le cas de l'histogramme des vitesses de rotation angulaire). Le processus de fusion des données dans le cadre du calcul parallèle est présenté dans la Figure 2.14. On y retrouve les k sous-ensembles de fichues répartis de manière égale obtenus en utilisant le script de partitionnement présenté à la section 2.2.4.

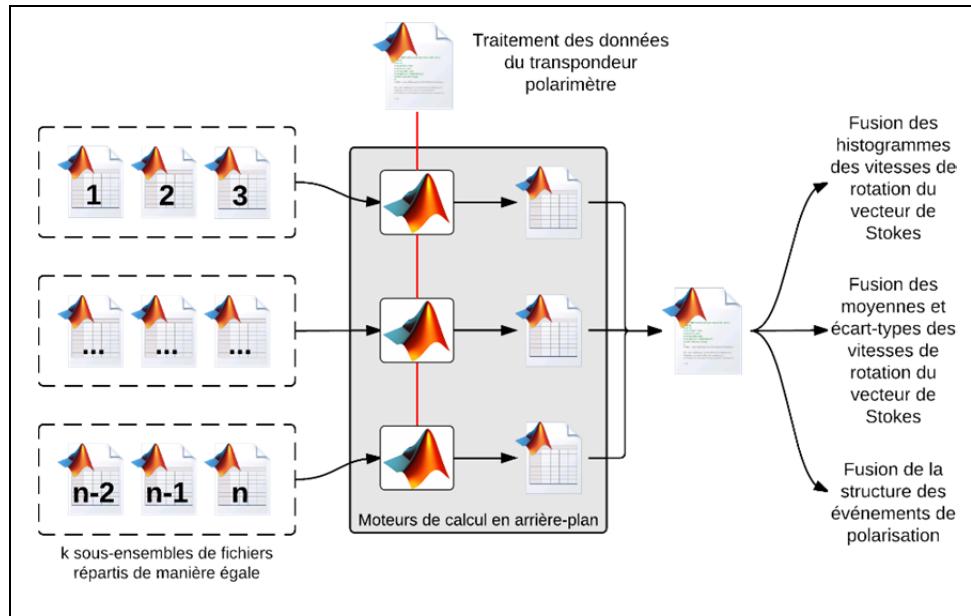


Figure 2.14 Calcul parallèle et fusion des données du polarimètre

Le résultat final après cette étape est un fichier Matlab contenant l'histogramme des vitesses de rotation du vecteur de Stokes pour l'ensemble des fichiers, une structure contenant la moyenne et l'écart type des vitesses de rotation du vecteur de Stokes pour chaque sous-période et une structure Matlab dont les colonnes contiennent les informations présentées au Tableau 2.2.

2.4 Évaluation du gain de performance obtenu par le calcul parallèle

Face au nombre important de mesures à analyser dans le cadre du projet, ma contribution a porté d'une part sur l'optimisation, d'un point de vue algorithmique, des méthodes de calcul proposées par (Mortelette, 2014) et d'autre part de mettre en place une méthode de calcul parallèle. L'objectif de cette section est de quantifier le gain de performance ainsi réalisé.

L'environnement de test utilisé dans le cadre du projet est une machine tournant sous le système Ubuntu 14.04 (Linux), équipé d'un processeur à 4 cœurs cadencé à 3.5 GHz. Pour comparer les temps d'exécution, le traitement de données a été lancé sur des échantillons de

fichiers de différentes tailles (typiquement séparées d'un ordre de grandeur de 10) avec 3 scripts :

- le script de traitement initial des données du transpondeur cohérent en série proposé par (Mortelette, 2014);
- le script de traitement initial des données du polarimètre en série proposé par (Mortelette, 2014);
- le script de traitement initial des données du transpondeur cohérent développé dans le cadre de ce projet en utilisant plusieurs nombres de moteurs de calcul (1, 2 et 4);
- le script de traitement initial des données du polarimètre développé dans le cadre de ce projet en utilisant plusieurs nombres de moteurs de calcul (1, 2 et 4).

La phase de comparaison est automatisée à travers le développement d'un script exécutant de manière séquentielle les quatre scripts énumérés ci-dessus, en consignant à chaque étape les temps d'exécution mesurés à l'aide des outils de chronométrage fournis par Matlab.

2.4.1 Résultats pour le traitement des données du polarimètre

Le traitement statistique des données du polarimètre a été lancé sur 5 échantillons de fichiers contenant respectivement 100, 500, 1000, 5000 et 10000 fichiers (200 000 points de mesure par fichier du polarimètre). Les résultats sont présentés à la Figure 2.15. On peut constater que l'utilisation de quatre unités de calcul permet de gagner un temps de traitement de l'ordre de 36% pour traiter 10 000 fichiers par rapport au script développé par (Mortelette, 2014). Par ailleurs, certaines optimisations effectuées au sein des scripts de traitement initial des données du polarimètre ont permis de réduire le temps de calcul de 640 à 400 s pour traiter 10 000 fichiers.

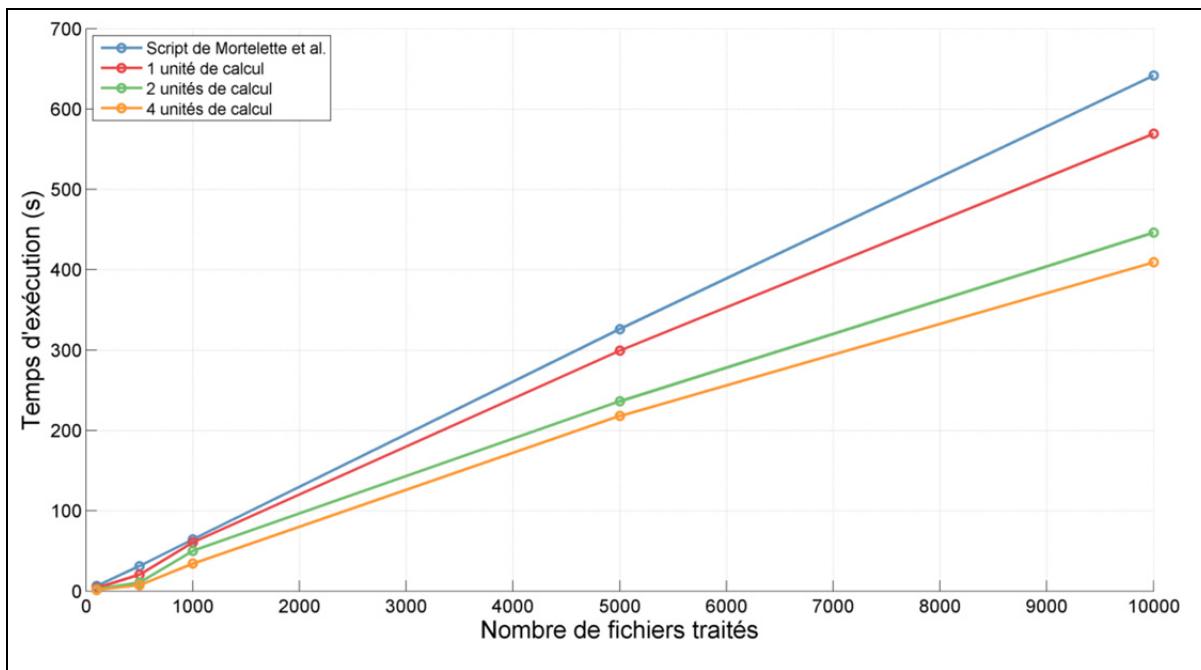


Figure 2.15 Tracé de la comparaison du temps d'exécution pour le polarimètre

2.4.2 Résultat pour le traitement des données du transpondeur cohérent

Le traitement des données du transpondeur cohérent a été ensuite lancé sur trois échantillons de fichiers contenant respectivement 100, 500 et 1000 fichiers (8738 points de mesure par fichier du transpondeur cohérent). Le traitement consiste à rassembler les valeurs des paramètres d'horodatage, de PMD et de PDL de chacun des fichiers de l'échantillon et de calculer les valeurs moyennes à la minute de ces paramètres. Les résultats sont présentés à la Figure 2.16. On peut constater que l'utilisation de quatre unités de calcul permet sur 10 000 fichiers de diminuer le temps de calcul par un facteur supérieur à 10 par rapport au script de calcul en série développé par (Mortelette, 2014).

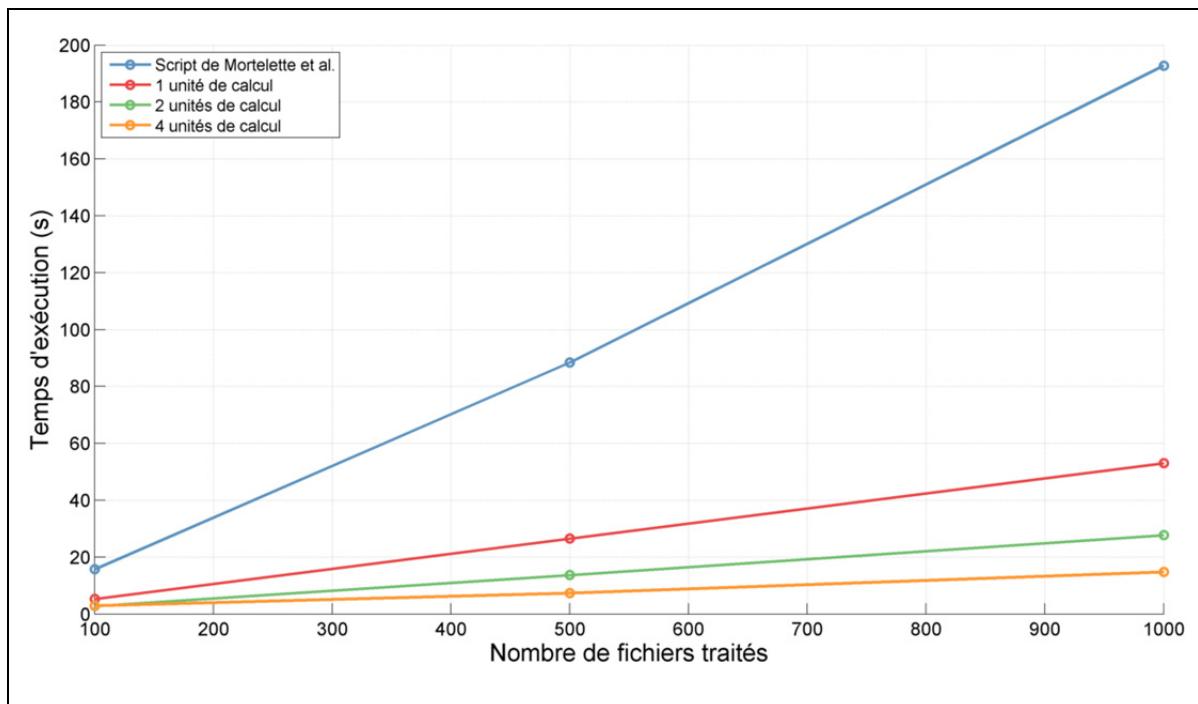


Figure 2.16 Tracé de la comparaison du temps d'exécution pour les mesures du transpondeur cohérent

2.5 Conclusion du chapitre

Ce chapitre a présenté les méthodes de traitement appliquées aux données du polarimètre et du transpondeur cohérent. Un des principaux défis posés par ce projet porte sur la flexibilité nécessaire au niveau des outils pour s'adapter à de grands volumes de données. Pour cela, une méthode de calcul parallèle a été mise en application et a permis de diminuer le temps de calcul des données du polarimètre par un facteur proche de 2 et celles du transpondeur cohérent par un facteur supérieur à 10.

CHAPITRE 3

ANALYSE STATISTIQUE DES RÉSULTATS DE MESURE D'UNE LIAISON COMMERCIALE AVEC UN TRANSPONDEUR COHÉRENT ET UN POLARIMÈTRE

3.1 Introduction

Ce chapitre présente les expériences de monitoring menées dans le cadre du projet. L'objectif de ces expériences est d'améliorer la compréhension des phénomènes se produisant sur les liaisons optiques cohérentes commerciales et notamment les effets de polarisation. L'approche abordée dans le cadre de ce projet tire son originalité du fait qu'une liaison aérienne commerciale longue distance a été monitorée à la fois par un transpondeur cohérent et par un polarimètre rapide et ce, sur plusieurs saisons. Les présents travaux font suite aux travaux rapportés par (Mortelette, 2014). Le traitement des données présentées dans ce chapitre a été effectué en utilisant la méthode de calcul parallèle décrite dans le chapitre précédent.

3.2 Outils d'analyse des données mesurées

Cette section présente les outils utilisés pour analyser et exploiter les mesures effectuées par le transpondeur cohérent et le polarimètre. L'ensemble des paramètres qui sont mesurés par le transpondeur cohérent a été présenté à la section 1.6.3. Les paramètres qui nous intéressent ici sont :

- le BER;
- la PMD, qui représente une valeur moyenne de la DGD;
- la PDL;
- les trois paramètres de Stokes (S_1 , S_2 et S_3), plus particulièrement S_1 .

Les calculs communs aux données du transpondeur cohérent et aux données du polarimètre sont présentés dans les sous-sections suivantes. Sauf mention contraire, les structures de

données utilisées pour traiter les données du polarimètre et du transpondeur cohérent sont celles qui ont été présentées au chapitre 2. Les étapes qui suivent s'appliquent aux données issues des scripts de traitement initial.

3.2.1 Calcul de la moyenne des échantillons à la minute

La première méthode d'analyse utilisée porte sur l'étude de l'évolution de la valeur moyenne des différents paramètres mesurés par le transpondeur cohérent. Cela permettra une comparaison avec les travaux de (Woodward et al., 2014) et (Woodward et al., 2012) qui ont effectué une caractérisation du comportement d'une liaison commerciale en mesurant l'évolution de la DGD, la PDL et de l'activité de polarisation telle qu'évaluée à l'aide de la première composante de Stokes. Woodward a effectué des moyennes des mesures par périodes d'une minute. Cette méthode peut être vue comme une manière de filtrer le bruit présent dans les mesures. A tire d'exemple, le transpondeur cohérent effectue une mesure toutes les millisecondes, nous allons donc calculer la moyenne et l'écart type de chaque paramètre mesuré sur des échantillons de 6000 valeurs. Le calcul de la valeur moyenne des échantillons à la minute permet une comparaison directe avec les résultats antérieurs (Woodward et al., 2014).

La méthode de moyennage des mesures du transpondeur cohérent a été présentée à la section 2.2.3. Par la suite, on désignera par $\langle x \rangle$ la valeur moyenne sur une minute du paramètre x , obtenue en suivant la méthode présentée précédemment.

3.2.2 Autocorrélation des paramètres de la liaison

Pour un phénomène (ou un paramètre) qui varie dans le temps, un paramètre statistique important est l'autocorrélation (*Auto-Correlation Function*, ACF) où la corrélation entre les mesures d'un paramètre à un temps donné et à un temps décalé variable. Cet outil, très utilisé en traitement de signal, consiste à effectuer une corrélation croisée d'un signal avec lui-même, après lui avoir fait subir un décalage temporel. Cette partie présente dans un premier

temps la méthode suivie pour calculer l'autocorrélation de la DGD. L'autocorrélation de la DGD est ensuite présentée pour deux saisons.

Ma contribution a porté sur l'adaptation du script de traitement Matlab développé par (Mortelette, 2014) afin de comparer nos résultats avec ceux de (Woodward et al., 2014). Le script de traitement, qui porte le nom de *compute_autocorrelation* et est présenté à l'ANNEXE XII, page 191, a été développé de sorte à s'adapter à plusieurs paramètres tels que la PMD, la PDL pour chaque période de mesure. Pour alléger la rédaction, la méthode est présentée ici pour le calcul de <DGD>. Le calcul peut être généralisé à n'importe quel paramètre mesuré par le transpondeur cohérent (DGD, PDL, S_1) ou le polarimètre (S_1 , S_2 , S_3). Les différentes étapes du calcul de la fonction d'autocorrélation sont synthétisées à la Figure 3.1.

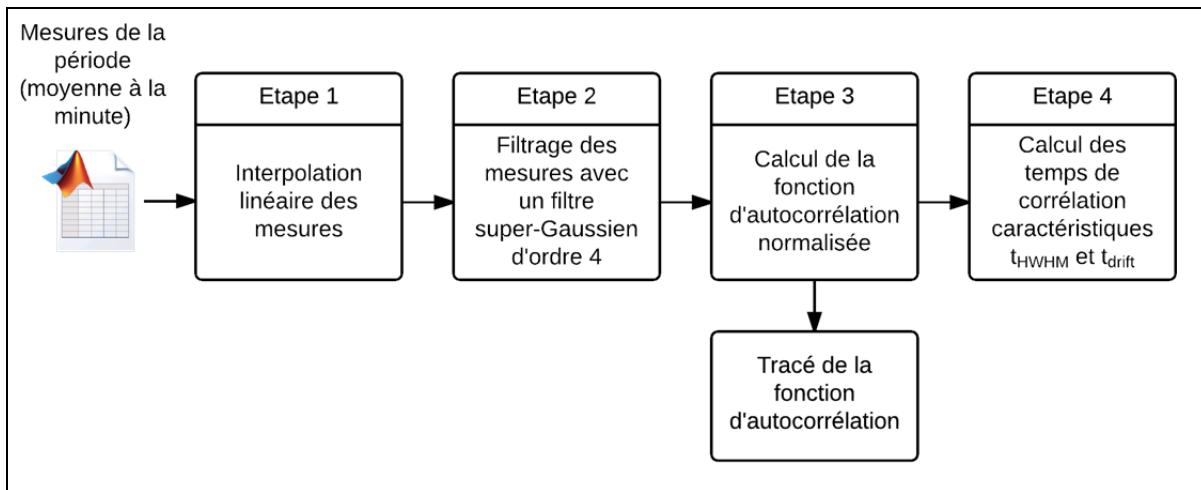


Figure 3.1 Étapes du calcul de la fonction d'autocorrélation d'un paramètre mesuré (DGD, PDL, S_1)

Chacune des étapes illustrées à la Figure 3.1 est décrite dans les sous-sections qui suivent.

3.2.2.1 Interpolation linéaire des valeurs des paramètres

La première étape, qui consiste à interpoler linéairement les différentes valeurs du paramètre mesuré sur l'ensemble de l'intervalle temporel traité est primordiale, car les mesures

effectuées par les instruments ne sont pas rigoureusement continues et il peut donc y avoir des temps morts entre deux mesures successives qui vont entraîner des discontinuités sur l'ensemble des données. L'interpolation des mesures est effectuée par l'intermédiaire de la fonction Matlab *interp1*. Cette fonction prend en paramètres trois matrices : la première contenant les valeurs d'horodatage (obtenue à l'issue du moyennage à la minute des valeurs du paramètre mesuré, décrit à la section 2.2.3), la seconde les paramètres mesurés (obtenue à l'issue du moyennage à la minute des valeurs du paramètre mesuré, décrit à la section 2.2.3) et la troisième contenant l'ensemble des points sur lesquels effectuer l'interpolation (tous les points espacés d'une minute, situés entre les bornes de la matrice contenant les valeurs d'horodatage). La structure de données obtenue en sortie contient la valeur moyenne du paramètre étudié sur toutes les minutes appartenant à l'intervalle de temps spécifié comme troisième paramètre à la fonction *interp1*.

3.2.2.2 Méthode de filtrage du bruit du signal

Même moyennées à la minute, les mesures effectuées avec le transpondeur cohérent sont entachées de bruit. Certains travaux ont porté sur l'étude comparative de plusieurs méthodes pour filtrer le bruit des mesures de DGD (Mortelette, 2014). Ce dernier a comparé trois types de filtre et les résultats les plus concluants ont été obtenus en utilisant un filtre super gaussien d'ordre 4.

La seconde étape de traitement va consister à filtrer les effets du bruit à l'aide d'un filtrage super gaussien d'ordre 4. Cette méthode de filtrage propose d'appliquer à la transformée de Fourier du signal contenant les mesures de <DGD> un fenêtrage temporel à l'aide d'une fenêtre super gaussienne, définie par (Hui, Guiliang et Wang, 1991). Ces derniers définissent l'équation d'un filtre super gaussien d'ordre 4, à travers l'équation (3.1)

$$H(n) = e^{-4\left(\frac{n}{\sigma}\right)^4} \quad (3.1)$$

où σ correspond à la bande passante du filtre. Pour atténuer le bruit dans les mesures de DGD, une valeur de largeur de bande de 500 Hz permet d'obtenir des résultats satisfaisants (Mortelette, 2014).

Le filtrage du bruit est effectué dans le domaine fréquentiel, c'est-à-dire sur la représentation de Fourier. Les données échantillonnées par le transpondeur cohérent étant discret, il est nécessaire de calculer la transformée de Fourier discrète. Cette étape est réalisée à travers la fonction *fft* de Matlab, qui prend en paramètre un signal discret et retourne sa transformée discrète, calculée à l'aide de la formule (3.2), tirée de la documentation de Matlab :

$$X(j) = \sum_{j=1}^N x(j)(e^{(-2\pi i)/N})^{(j-1)(k-1)} \quad (3.2)$$

où x désigne le vecteur contenant les données de <DGD>, N le nombre d'éléments contenus dans x et X la transformée de Fourier discrète de <DGD>.

L'étape suivante consiste à calculer le produit de convolution entre la fenêtre super gaussienne d'ordre 4 définie précédemment et le signal <DGD>. Dans le domaine fréquentiel, cela revient à calculer le produit de la transformée de Fourier discrète et des coefficients du filtre super gaussien d'ordre 4, tel qu'illustré à l'équation (3.3).

$$X_f(n) = X(n) * H(n) \quad (3.3)$$

où X_f désigne la représentation de Fourier du signal filtré, X la représentation de Fourier du signal avant filtrage et H les coefficients de Fourier du filtre super gaussien d'ordre 4 calculé précédemment.

La dernière étape du filtrage consiste à calculer la transformée de Fourier inverse afin d'obtenir la représentation de <DGD> filtrée dans le domaine temporel. Cette dernière étape s'effectue à travers la fonction *ifft* de Matlab, prenant en paramètre la représentation de

Fourier d'un signal et retournant une représentation temporelle. L'opération appliquée à la représentation de Fourier est décrite à l'équation (3.4),

$$x(j) = \frac{1}{N} \sum_{k=1}^N X(k) (e^{(-2\pi i)/N})^{-(j-1)(k-1)} \quad (3.4)$$

où X désigne la transformée de Fourier discrète de $\langle DGD \rangle$, N le nombre d'éléments contenus dans x et x la représentation de X dans le domaine temporel.

L'effet du filtrage super gaussien sur les valeurs de $\langle DGD \rangle$ mesurées par le transpondeur cohérent pendant le mois de novembre 2012 est présenté à la Figure 3.2 ci-dessous. Le filtrage donne des résultats satisfaisants dans la mesure où le signal obtenu est moins bruyant. On peut remarquer que le filtrage entraîne par contre une diminution de l'amplitude des variations de $\langle DGD \rangle$. Cependant, ce phénomène n'entraîne pas d'erreur dans les calculs d'autocorrélation d'après (Orfanidis, 2007).

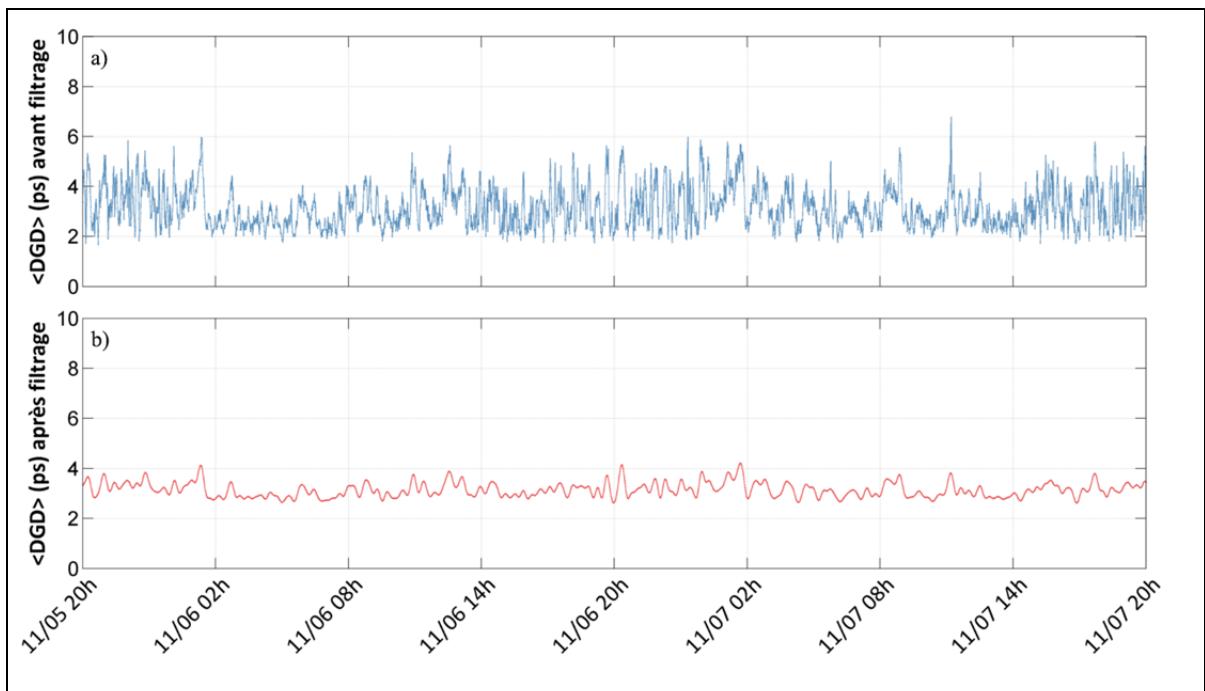


Figure 3.2 Effet du filtrage super gaussien d'ordre 4 de $\langle \text{DGD} \rangle$ sur une période de 2 jours
 (mesures du transpondeur cohérent de novembre 2012)
 a) mesures de $\langle \text{DGD} \rangle$ non filtrées, b) mesures de $\langle \text{DGD} \rangle$ filtrées

3.2.2.3 Calcul de la fonction d'autocorrélation normalisée et des temps de corrélation caractéristiques

L'étape suivante du traitement consiste à calculer la matrice d'autocorrélation normalisée des valeurs de $\langle \text{DGD} \rangle$. Par définition, l'autocorrélation d'une variable correspond au rapport de l'autocovariance de cette variable. L'équation (3.5), tirée de (Orfanidis, 2007), présente la formule utilisée pour calculer l'autocorrélation de $\langle \text{DGD} \rangle$,

$$ACF(m) = \frac{\sum_{n=0}^{N-m-1} (x_{n+m} - \bar{x})(x_n - \bar{x})}{\sum_{k=1}^N (x_k - \bar{x})^2} \quad (3.5)$$

où ACF désigne la matrice d'autocorrélation de $\langle \text{DGD} \rangle$ et x le vecteur contenant les valeurs de $\langle \text{DGD} \rangle$ à la minute. Il est ainsi possible de tracer la fonction d'autocorrélation de $\langle \text{DGD} \rangle$ pour les différentes périodes de mesure du transpondeur cohérent.

La dernière étape consiste à calculer les indicateurs qui vont rendre possible la comparaison des résultats obtenus à partir des fonctions d'autocorrélation. Les indicateurs proposés par (Woodward et al., 2014) sont la demi-largeur à mi-hauteur (*Half-Width at Half-Maximum, HWHM*), et le temps de dérive (*drift time*). Le temps de dérive est une notion qui est définie par (Karlsson, Brentel et Andrekson, 2000) et qui correspond à un temps caractéristique de la variation de la DGD dans une fibre. Ce temps de dérive peut être calculé en ajustant à la fonction d'autocorrélation de $\langle \text{DGD} \rangle$ l'équation (3.6),

$$C(t) = \frac{1 - e^{-\frac{|t|}{t_{drift}}}}{\frac{|t|}{t_{drift}}} \quad (3.6)$$

où t_{drift} désigne le temps de dérive de $\langle \text{DGD} \rangle$.

Le résultat de l'exécution de la fonction Matlab est une matrice contenant, pour chaque valeur de décalage temporel (pas d'une minute) la valeur d'autocorrélation associée. Le calcul de la demi-largeur à mi-hauteur de la fonction d'autocorrélation peut se déterminer directement de la matrice contenant la fonction d'autocorrélation de $\langle \text{DGD} \rangle$: il suffit de rechercher de manière itérative le décalage temporel à partir duquel la valeur de l'ACF de $\langle \text{DGD} \rangle$ passe en dessous de 0.5.

3.3 Analyse des mesures du transpondeur cohérent

L'objectif de cette section est d'analyser les mesures obtenues à l'aide du transpondeur cohérent. Dans un premier temps est présentée une revue des périodes de mesure couvertes dans le cadre du projet. Cette revue est suivie par la présentation des outils utilisés pour analyser les différentes mesures puis de l'analyse des mesures.

3.3.1 Périodes couvertes par les mesures du transpondeur cohérent

Ma contribution dans le cadre du projet a été d'analyser l'évolution des mesures du transpondeur cohérent sur un nombre significatif de périodes, s'étendant sur plusieurs saisons. Les mesures analysées dans le cadre du projet de (Mortelette, 2014) correspondent à la première période de mesures effectuées à l'automne 2012 (du 25 octobre au 11 novembre 2012).

Le module DSP intégré au transpondeur cohérent et présenté à la section 1.6.3 a permis de mesurer les différents paramètres de la liaison optique étudiée, présentée à la Figure 1.12. Les paramètres qui sont étudiés dans cette partie sont les suivants : la dispersion des modes de polarisation (PMD), les pertes dépendantes en polarisation (PDL), le taux d'erreur sur les bits précédant le FEC (PréFEC-BER) et le paramètre de Stokes S_1 .

3.3.2 Analyse des mesures brutes du transpondeur cohérent

Afin de comparer de manière quantitative les variations des paramètres entre les saisons, la valeur numérique des paramètres mesurés par le transpondeur cohérent sont présentés au Tableau 3.1. Les paramètres présentés ici sont les paramètres issus de l'étape de traitement initial présentée à la section 2.2.2, avant l'étape de moyennage.

Tableau 3.1 Caractérisation des valeurs de PMD, PDL et BER brutes mesurées par le transpondeur cohérent sur trois saisons

Paramètre (unité)	Saison (nombre de points de mesure)	Valeur			
		Min	Max	Moyenne	Écart type
BER	Aut. 2012 ($1,9 \times 10^8$)	$8,38 \times 10^{-7}$	$1,69 \times 10^{-3}$	$4,15 \times 10^{-6}$	$1,60 \times 10^{-6}$
	Hiv. 2013 ($3,4 \times 10^7$)	$8,99 \times 10^{-7}$	$3,34 \times 10^{-5}$	$4,58 \times 10^{-6}$	$2,04 \times 10^{-6}$
	Été 2014 ($1,8 \times 10^7$)	$1,02 \times 10^{-6}$	$1,05 \times 10^{-5}$	$3,50 \times 10^{-6}$	$8,20 \times 10^{-7}$
PDL (dB)	Aut. 2012 ($1,9 \times 10^8$)	$1,17 \times 10^{-3}$	4,46	1,89	0,680
	Hiv. 2013 ($3,4 \times 10^7$)	$5,43 \times 10^{-3}$	4,30	1,61	0,672
	Été 2014 ($1,8 \times 10^7$)	$6,21 \times 10^{-3}$	3,58	1,38	0,525
DGD (ps)	Aut. 2012 ($1,9 \times 10^8$)	$3,99 \times 10^{-3}$	13,2	3,08	1,35
	Hiv. 2013 ($3,4 \times 10^7$)	$3,87 \times 10^{-3}$	14,0	3,51	1,59
	Été 2014 ($1,8 \times 10^7$)	$1,81 \times 10^{-2}$	15,2	3,14	1,43

On peut constater que globalement, la valeur moyenne de la DGD est plus importante en hiver comparativement aux mesures réalisées en automne ou en été. Ainsi, la DGD passe d'une valeur de 3,08 ps en automne à une valeur de 3,51 ps en hiver soit une augmentation de l'ordre de 14%. De même, l'écart type de la valeur moyenne de la DGD, σ_{DGD} passe d'une valeur de 1,35 ps en automne à 1,59 ps en hiver soit une augmentation de 18%. Au contraire, la valeur moyenne de la PDL est plus importante en automne qu'en hiver ou en été. On observe ainsi une valeur passant de 1,89 dB en automne à une valeur de 1,61 dB en hiver et 1,38 en été ce qui traduit une diminution respective de 17 % et 37 % par rapport aux observations réalisées en automne. On peut conclure de cette analyse que la variation des effets de polarisation dans une fibre optique est non seulement influencée par les paramètres intrinsèques aux fibres considérées (abordés dans le premier chapitre de ce mémoire) mais aussi aux facteurs environnementaux puisqu'on observe une variation significative de l'ensemble des paramètres en fonction de la saison.

3.3.3 Analyse de la moyenne à la minute des paramètres mesurés

Une seconde analyse a porté sur la valeur moyenne à la minute des paramètres mesurés par le transpondeur cohérent, ainsi que la valeur moyenne de la valeur absolue de la dérivée temporelle de ces paramètres. Le Tableau 3.2 présente les résultats obtenus.

Tableau 3.2 Caractérisation des valeurs de PMD, PDL et BER moyennées à la minute et mesurées sur trois saisons

Paramètre (unité)	Saison	Valeur			
		Min	Max	Moyenne	Écart type
$\langle PDL \rangle$ (dB)	Automne 2012	$1,17 \times 10^{-3}$	4,46	1,89	0,680
	Hiver 2013	$5,43 \times 10^{-3}$	4,30	1,61	0,672
	Été 2014	$6,21 \times 10^{-3}$	3,58	1,38	0,525
$\langle \left \frac{dPDL}{dt} \right \rangle$ (dB/s)	Automne 2012	1,61	1,73	0,210	0,160
	Hiver 2013	1,63	1,69	0,211	0,161
	Été 2014	1,56	1,56	0,201	0,153
$\langle DGD \rangle$ (ps)	Automne 2012	$3,99 \times 10^{-3}$	$1,32 \times 10^1$	3,08	1,35
	Hiver 2013	$3,87 \times 10^{-3}$	$1,40 \times 10^1$	3,51	1,59
	Été 2014	$1,81 \times 10^{-2}$	$1,52 \times 10^1$	3,14	1,43
$\langle \left \frac{dDGD}{dt} \right \rangle$ (ps/s)	Automne 2012	8,97	9,00	1,00	0,775
	Hiver 2013	8,54	9,32	1,09	0,863
	Été 2014	8,83	8,83	0,969	0,763

Dans ce tableau, $\langle \left| \frac{dPDL}{dt} \right| \rangle$ et $\langle \left| \frac{dDGD}{dt} \right| \rangle$ désignent respectivement la moyenne à la minute de la valeur absolue de la dérivée temporelle de la PDL et de la DGD.

Les paramètres mesurés sur la liaison commerciale de Verizon peuvent être comparés à ceux observés par (Woodward et al., 2014) sur un ensemble de trois liaisons enterrées mesurant respectivement 30, 75 et 273 km, munies de fibre monomode, pré-1994. Les mesures

présentées dans cet article sont des valeurs moyennes des paramètres, calculées à la minute. Afin de comparer les résultats de ce projet à ceux obtenus par Woodward, la valeur moyenne de la DGD à la minute, notée $\langle \text{DGD} \rangle$, la valeur moyenne de la PMD à la minute, notée $\langle \text{PMD} \rangle$ et la valeur moyenne à la minute de leurs dérivées temporelles ont été calculées. L'étude comparative des résultats présentés par (Woodward et al., 2014) avec ceux obtenus sur la liaison aérienne présente un intérêt certain car elle va permettre de distinguer les caractéristiques d'une liaison optique aérienne et d'une liaison enfouie.

Les quatre paramètres cités précédemment ont été mesurés par (Woodward et al., 2014) durant 18 mois à l'aide d'un transpondeur cohérent : les résultats obtenus montrent que la valeur de $\langle \text{DGD} \rangle$ varient entre 8,3 et 22,4 ps, l'amplitude entre ces deux valeurs est due à la longueur différente des deux liaisons étudiées par Woodward. Ces dernières valeurs sont d'une manière générale supérieures à celles obtenues dans le cadre de ce projet, puisque les valeurs de $\langle \text{DGD} \rangle$ mesurées sur notre lien aérien prennent une valeur allant de 3,08 ps en automne 2012 à 3,51 ps en hiver 2013. Cette différence peut s'expliquer par le fait que (Woodward et al., 2014) rapportent que leur liaison est caractérisée par des valeurs de PMD exceptionnellement élevées.

Les valeurs de la valeur absolue de la dérivée temporelle de la DGD sont en revanche nettement plus importantes dans le cadre de ce projet. En effet, les résultats proposés par (Woodward et al., 2014) une valeur de $\langle \left| \frac{d\text{DGD}}{dt} \right| \rangle$ inférieure ou égale à 0,1 ps/s tandis que les mesures effectuées sur notre liaison varient entre 0,20 ps/s et 0,21 ps/s. Cette différence s'explique par le caractère essentiellement aérien de la liaison Verizon et donc d'une exposition plus importante aux facteurs extérieurs tels que le vent ou la température.

Ma contribution a porté sur le tracé de l'évolution temporelle des paramètres sur plusieurs saisons. Il a été possible de tracer l'évolution temporelle, durant les périodes de mesure les plus longues, de paramètres tels que la DGD, la PDL et le BER moyennés à la minute en automne (Figure 3.3), en hiver (Figure 3.4) et en été (Figure 3.5).

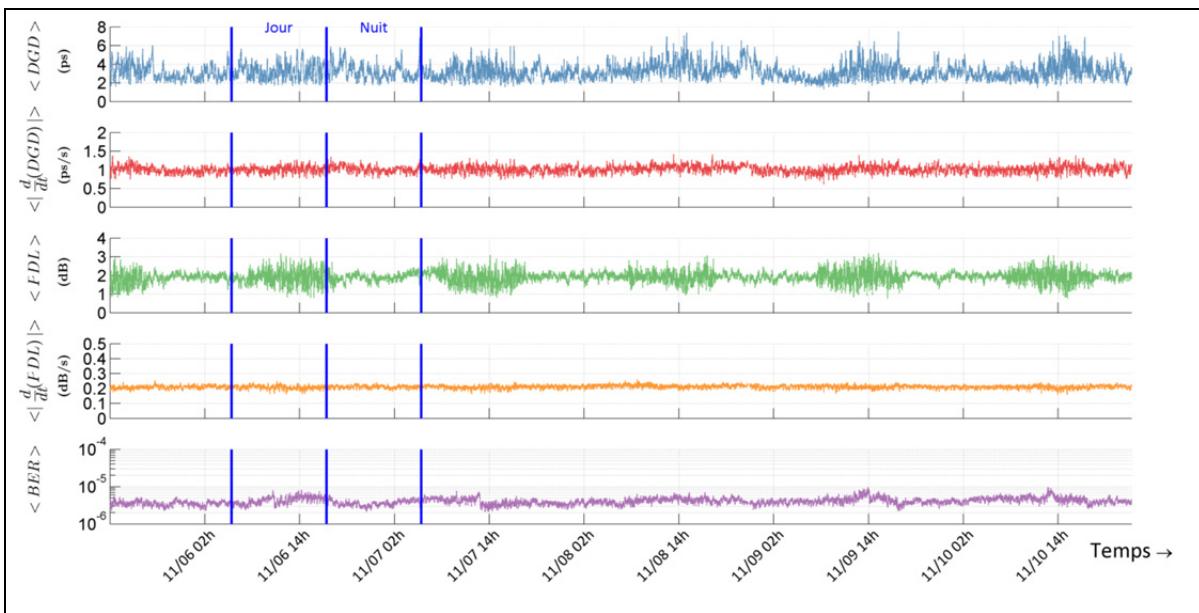


Figure 3.3 Évolution temporelle de $\langle DGD \rangle$, $\langle PDL \rangle$, la valeur absolue de leurs dérivées temporelles et $\langle BER \rangle$ pour chaque minute sur le lien de Verizon (période de 5 jours consécutifs en automne 2012)

La Figure 3.3 précédente permet de constater l'évolution des différents paramètres, durant une période temporelle supérieure à 5 jours. On constate que l'activité des paramètres, en termes d'amplitude de variation, connaît une variation cyclique par rapport au cycle jour/nuit représenté par des droites verticales sur la figure. Cette propriété est particulièrement visible pour les valeurs de $\langle DGD \rangle$ et de $\langle PDL \rangle$. On observe ainsi des variations plus importantes de 8 h à 18 h durant le jour

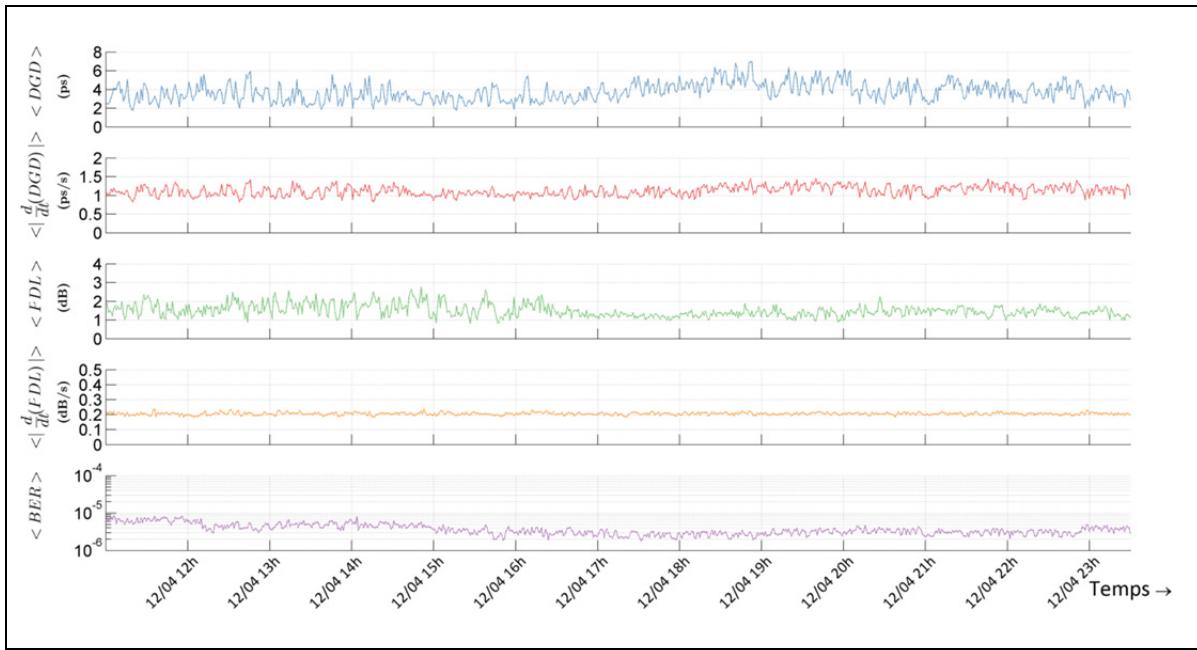


Figure 3.4 Évolution temporelle de $\langle DGD \rangle$, $\langle PDL \rangle$, leurs dérivées temporelles et $\langle BER \rangle$ pour chaque minute sur le lien de Verizon (période de 12 heures consécutives en hiver 2013)

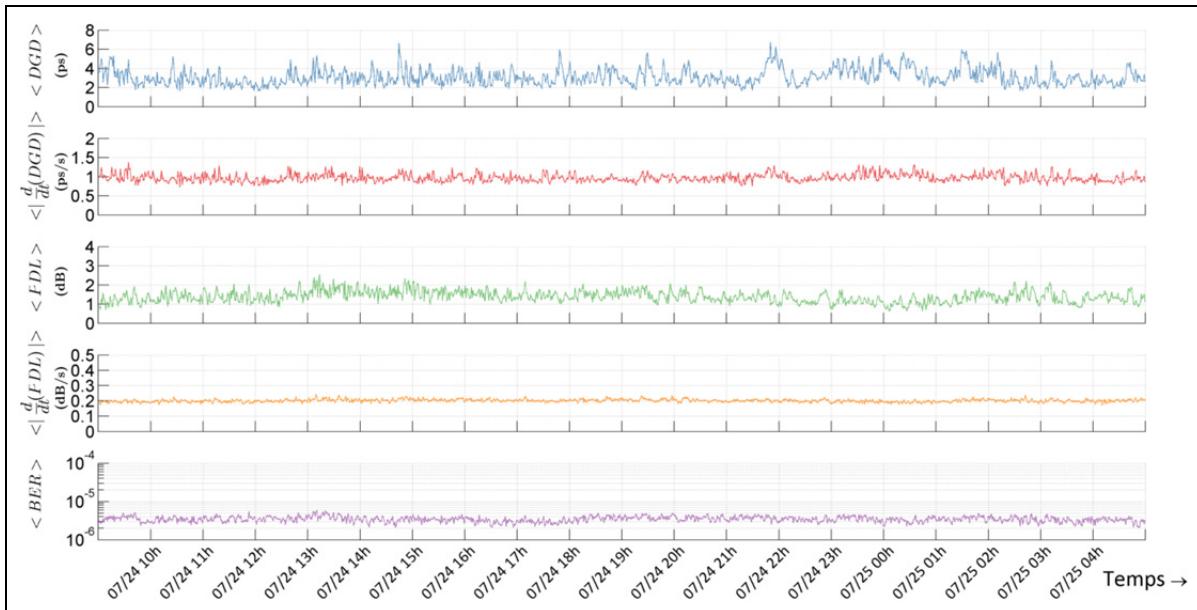


Figure 3.5 Évolution temporelle de $\langle DGD \rangle$, $\langle PDL \rangle$, leurs dérivées temporelles et $\langle BER \rangle$ pour chaque minute sur le lien de Verizon (période de 20 heures consécutives en été 2014)

La Figure 3.4 et la Figure 3.5 permettent d'avoir un aperçu de la fluctuation des paramètres sur une période de l'ordre de la dizaine d'heures. Compte tenu des courtes périodes temporelles en jeu, il n'est pas possible d'observer de variations significatives comme pour la période d'automne 2012. On peut cependant remarquer que les différents paramètres ont une tendance assez stable durant ces périodes.

3.3.4 Fonction d'autocorrélation de $\langle \text{DGD} \rangle$ et de $\langle \phi \rangle$

Afin de caractériser l'évolution de l'état de polarisation d'une liaison optique, (Woodward et al., 2014) ont étudié la partie positive de la fonction d'autocorrélation de $\langle \text{DGD} \rangle$ et de $\langle \phi \rangle$, où ϕ désigne la composante de polarisation définie par $\phi = \text{acos}(S_1)$. Cette composante correspondant à l'angle entre le vecteur de Stokes et l'axe S_1 de la sphère de Poincaré, présenté dans la Figure 3.6 ci-dessous, sur laquelle le vecteur de Stokes est représenté en bleu et l'angle ϕ par l'arc rouge.

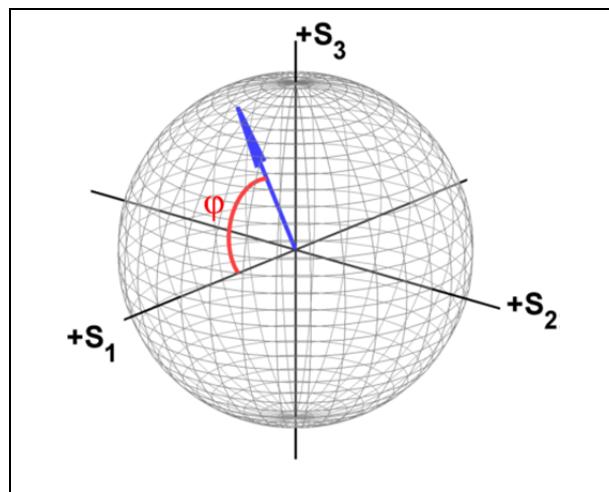


Figure 3.6 Définition de l'angle ϕ
sur la sphère de Poincaré

Seule la partie positive de l'autocorrélation est tracée car cette fonction est par nature symétrique. Afin de comparer les résultats observés, les fonctions d'autocorrélation de $\langle \text{DGD} \rangle$ (Figure 3.7) et de $\langle \phi \rangle$ (Figure 3.8) ont été tracées pour la liaison de Verizon sur les mois de novembre 2012. Pour ces différentes périodes, la fonction d'autocorrélation a été

calculée sur une période de 4 jours et les temps d'autocorrélation et de dérive de la PMD, définis à la section 3.2.2.3, ont été calculés. En raison du faible nombre de mesures continues réalisées sur les autres périodes, un nombre de points non suffisant n'a pas rendu possible le calcul de la fonction d'autocorrélation.

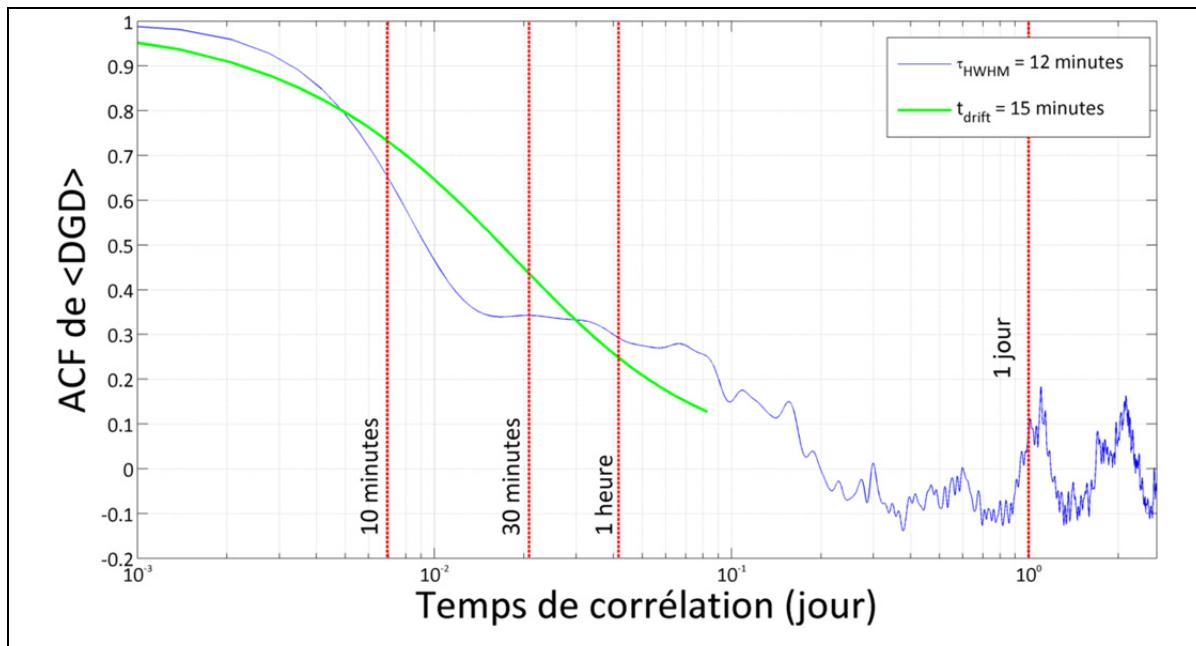


Figure 3.7 Fonction d'autocorrélation (partie positive) de $\langle DGD \rangle$ (novembre 2012)

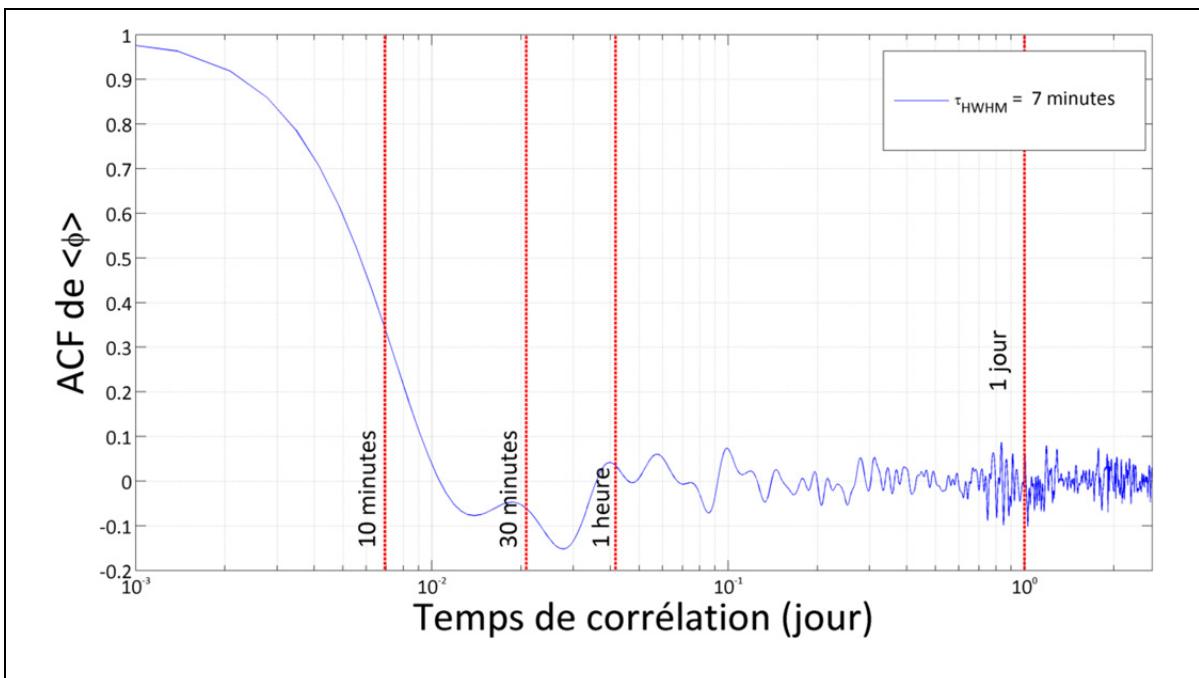


Figure 3.8 Fonction d'autocorrélation (partie positive) de $\langle\phi\rangle$ (novembre 2012)

On peut observer pour les deux périodes un temps de corrélation, qui correspond à la demi-largeur à mi-hauteur de la fonction d'autocorrélation, de l'ordre de 12 minutes, ainsi qu'un temps de dérive de l'ordre de 15 minutes pour le mois de novembre. Le temps de corrélation observé est sensiblement inférieur à celui constaté par (Woodward et al., 2014), qui varie entre 1,3 et 4,1 jours pour un ensemble de liaisons enterrées, et à celui du temps de dérive qui varie entre 3 et 5,7 jours. La différence de temps de corrélation peut s'expliquer par la nature différente des deux liaisons : la liaison étudiée par (Woodward et al., 2014) étant essentiellement enfouie comparativement à la liaison de Verizon qui est presque entièrement aérienne. De la même manière, on observe un temps de corrélation HWHM lié au paramètre $\langle\phi\rangle$ de l'ordre de 7 minutes et un temps de dérive de la polarisation de l'ordre de 2,6 minutes. Ces résultats permettent d'observer les mêmes tendances que pour l'expérience de (Woodward et al., 2014), à savoir que les temps de corrélation de ϕ sont inférieurs à ceux de la DGD, ces derniers observent une demi-largeur à mi-hauteur variant de 0,18 à 1,7 jour pour $\langle\phi\rangle$ et variant de 1,3 à 4,1 jours pour la DGD.

3.4 Analyse des mesures du polarimètre

Cette partie présente les mesures réalisées par le polarimètre ainsi que la méthode suivie pour exploiter les données expérimentales obtenues. Les données issues du polarimètre ont été mesurées à l'aide d'un synthétiseur de polarisation Agilent N7786B utilisé en mode polarimètre. Le montage expérimental est conçu de sorte à capturer simultanément les données mesurées par le polarimètre et les données mesurées par le transpondeur cohérent étudié dans la partie précédente.

3.4.1 Périodes couvertes par les mesures du polarimètre

Ma contribution dans le cadre de ce projet a porté sur l'étude de l'évolution temporelle des paramètres mesurés par le polarimètre. Cette étude présente un intérêt double : elle permet d'une part d'étudier l'évolution de l'état de polarisation du signal optique avec une fiabilité élevée, car cet instrument est exclusivement conçu pour mesurer les paramètres de Stokes et d'autre part, l'étude des mesures obtenues rend possible la comparaison des résultats obtenus avec les transpondeurs cohérents. Un récapitulatif du nombre de fichiers de mesure du polarimètre par saison est présenté au tableau Tableau 3.3 ci-dessous.

Tableau 3.3 Mesures du polarimètre

Saison	Nombre de points de mesure	Période couverte
Automne 2012	5.99×10^9	25/10/2012 au 11/11/2012
Automne 2013	1.78×10^{10}	04/09/2013 au 30/11/2013
Hiver 2013	2.99×10^{10}	01/12/2013 au 09/02/2014
Printemps 2014	1.20×10^{10}	08/04/2014 au 25/05/2014
Été 2014	2.74×10^{10}	11/07/2014 au 31/08/2014
Automne 2014	3.38×10^9	01/09/2014 au 06/09/2014

3.4.2 Analyse des mesures du polarimètre

L'analyse temporelle des mesures du polarimètre nécessite un traitement préalable qui consiste à calculer la valeur moyenne des paramètres mesurés. Cette étape a été détaillée à la

section 2.3.3. A la suite du calcul de la valeur moyenne des mesures de $\langle\varphi\rangle$ par le polarimètre, la fonction d'autocorrélation a été calculée pour l'ensemble des périodes de mesures. Le calcul du temps caractéristique de la fonction d'autocorrélation et de la demi-largeur à mi-hauteur a été réalisé et les résultats sont présentés par saison au Tableau 3.4, qui offre un aperçu de la valeur de ces temps pour chaque saison. Les sous-périodes de mesure n'étant pas uniformément réparties (il peut y avoir certaines mesures de $\langle\varphi\rangle$ espacées de plus d'une minute), le calcul de la fonction d'autocorrélation a été effectué, pour chaque période de mesure, sur le plus grand ensemble de mesures continues de $\langle\varphi\rangle$, c'est-à-dire espacées au maximum d'une minute. Le nombre de points utilisés pour le calcul (un point par minute de mesure), ainsi que les dates de début et de fin des périodes ainsi obtenues sont présentés au Tableau 3.4.

Tableau 3.4 Mesures de $\langle\varphi\rangle$ par le polarimètre utilisées pour calculer la fonction d'autocorrélation

Début de la période	Fin de la période	Nombre de points de mesure	t_{HWHM} (min)
19/12/2013 10:39:14	29/12/2013 08:13:13	12960	12
08/04/2014 11:55:03	18/04/2014 08:43:02	12960	11
25/07/2014 05:07:20	06/09/2014 14:28:20	12960	12

La fonction d'autocorrélation de $\langle\varphi\rangle$ a été présentée graphiquement pour trois saisons : hiver 2014 (Figure 3.9), printemps 2014 (Figure 3.10) et été 2014 (Figure 3.11).

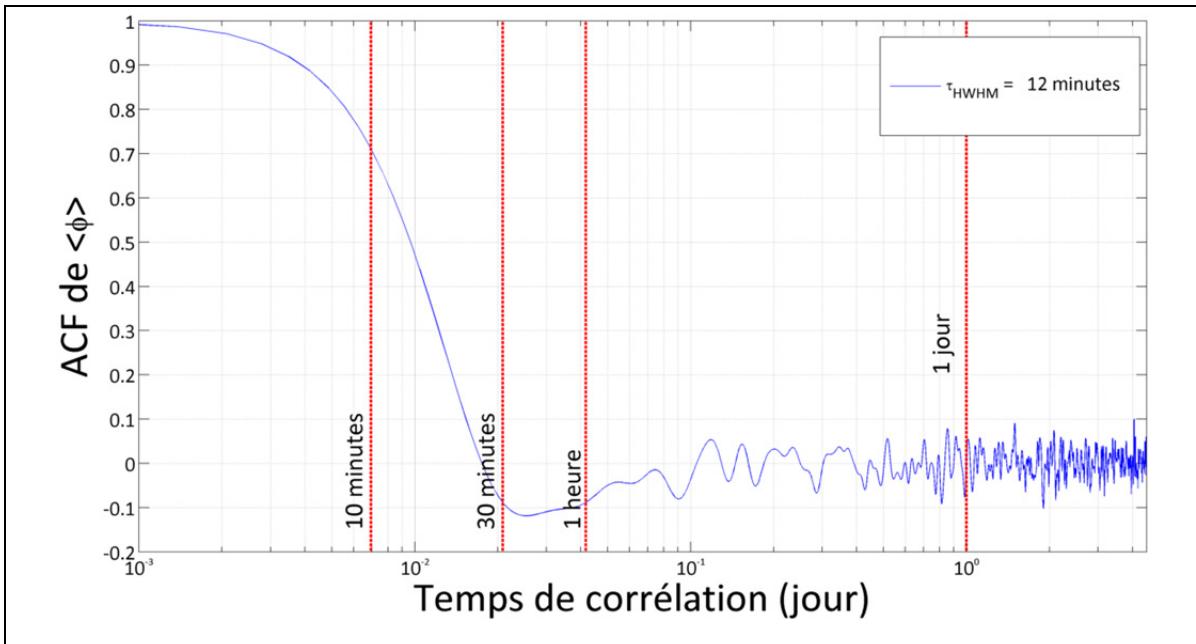


Figure 3.9 Fonction d'autocorrélation de $\langle \phi \rangle$ (d'après les mesures du polarimètre) sur la période d'hiver 2013

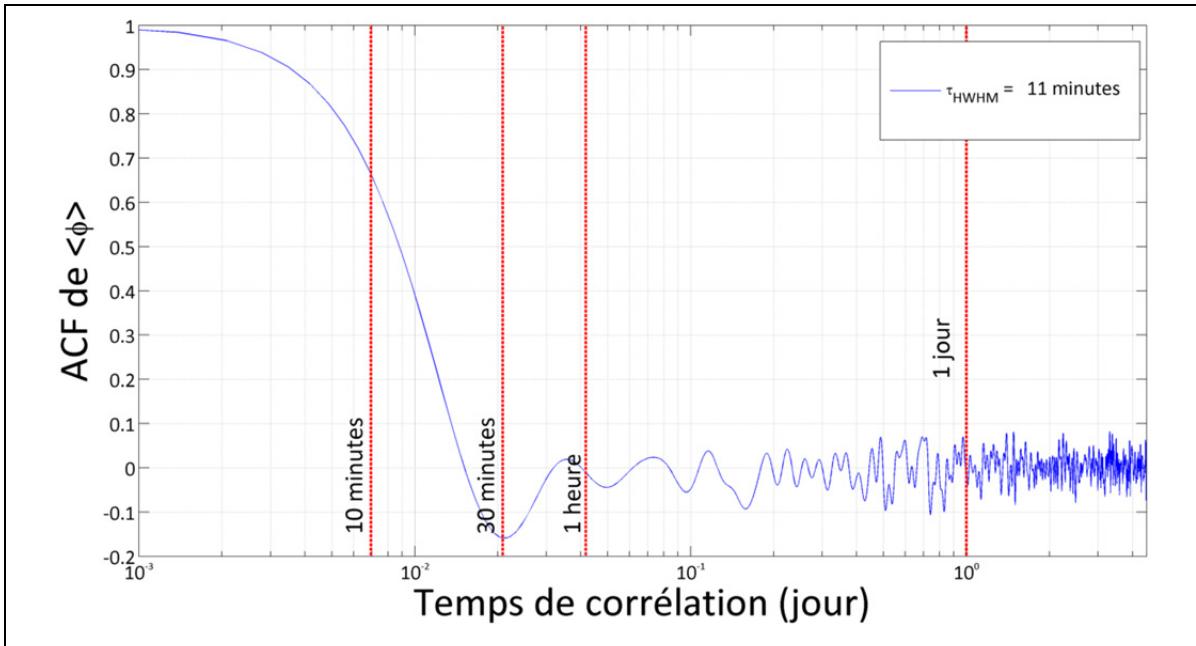


Figure 3.10 Fonction d'autocorrélation de $\langle \phi \rangle$ (d'après les mesures du polarimètre) sur la période de printemps 2014

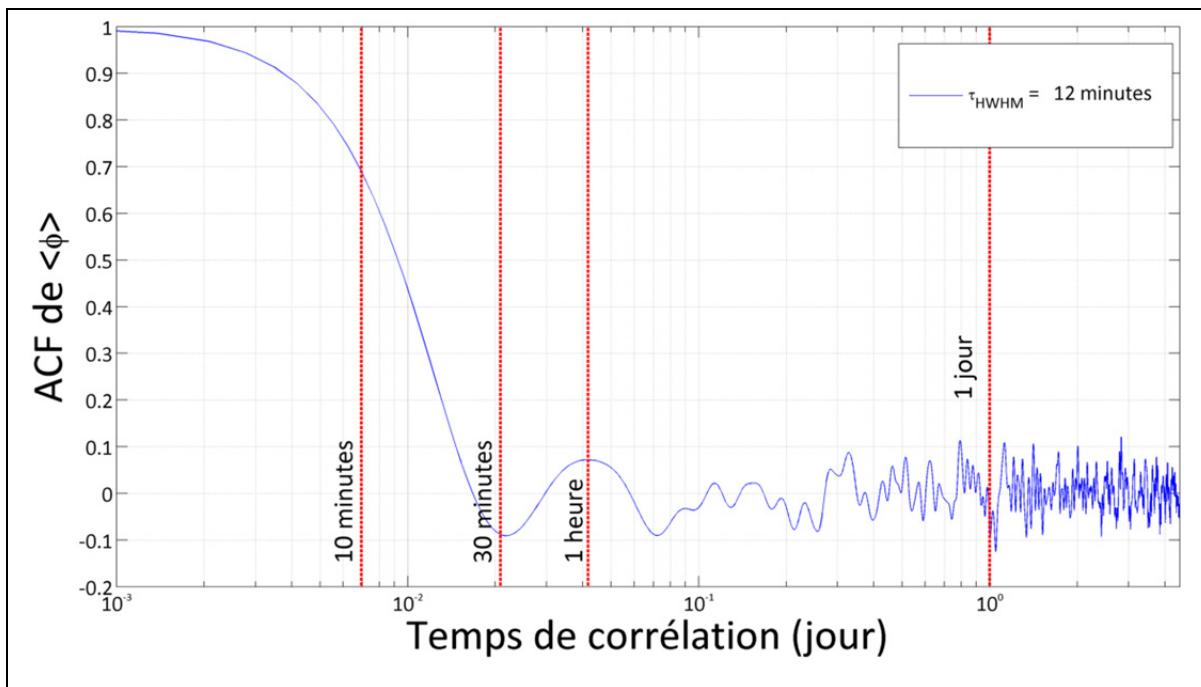


Figure 3.11 Fonction d'autocorrélation de $\langle \phi \rangle$ (d'après les mesures du polarimètre) sur la période d'été 2014

Les valeurs de la demi-largeur à mi-hauteur observées durant les saisons d'hiver printemps et été sont similaires. On peut en déduire que quelle que soit la saison, les variations périodiques de l'activité de polarisation sont globalement de même durée avec un temps de corrélation de l'ordre de la dizaine de minutes. Le temps caractéristique de corrélation de $\langle \phi \rangle$ semble donc être une valeur qui est propre à chaque liaison optique, compte tenu des résultats différents observés entre la liaison entrée de (Woodward et al., 2014), variant de 0,18 à 1,7 jour, et la liaison aérienne, dont le temps caractéristique est de l'ordre de la dizaine de minutes. Cela tend à confirmer l'hypothèse de départ selon laquelle l'activité de polarisation d'une liaison aérienne est plus importante que celle d'une liaison enfouie.

3.5 Comparaison des mesures du polarimètre à celles du transpondeur cohérent

Ce projet a permis d'étudier, sur une longue durée, une liaison optique avec deux types d'instruments : le transpondeur cohérent et le polarimètre. Il a ainsi été possible de comparer l'évolution temporelle d'un paramètre mesuré simultanément par les deux instruments : la

première composante de Stokes, notée S_1 et le paramètre φ , qui correspond à l'angle, sur la sphère de Poincaré, entre l'axe S_1 et le vecteur de Stokes. Cette étude comparative présente un intérêt certain pour les opérateurs pour deux raisons : les transpondeurs cohérents sont à l'heure actuelle déployés dans une grande partie des liens optiques et d'autre part, la capacité de monitorer des paramètres directement depuis les transpondeurs est très intéressante d'un point de vue financier comparativement aux coûts de déploiement d'instruments de test comme les polarimètres. Par ailleurs, les polarimètres ne sont jamais déployés comme outils de monitoring dans une liaison commerciale.

Les données utilisées dans le cadre de la comparaison des instruments correspondent à la période durant laquelle l'acquisition en continu des données du transpondeur cohérent a fonctionné sur une durée supérieure à un jour, correspondant aux mois d'octobre et novembre 2012. La représentation chronologique de ces données est donnée à la Figure 1.8. On optera pour la période de novembre qui correspond à la plus grande période de mesure continue et simultanée entre le polarimètre et le transpondeur cohérent.

Ma contribution a porté sur la comparaison entre les mesures du transpondeur cohérent et les mesures du polarimètre. La première étape de ce protocole de comparaison est présentée à la Figure 3.12.

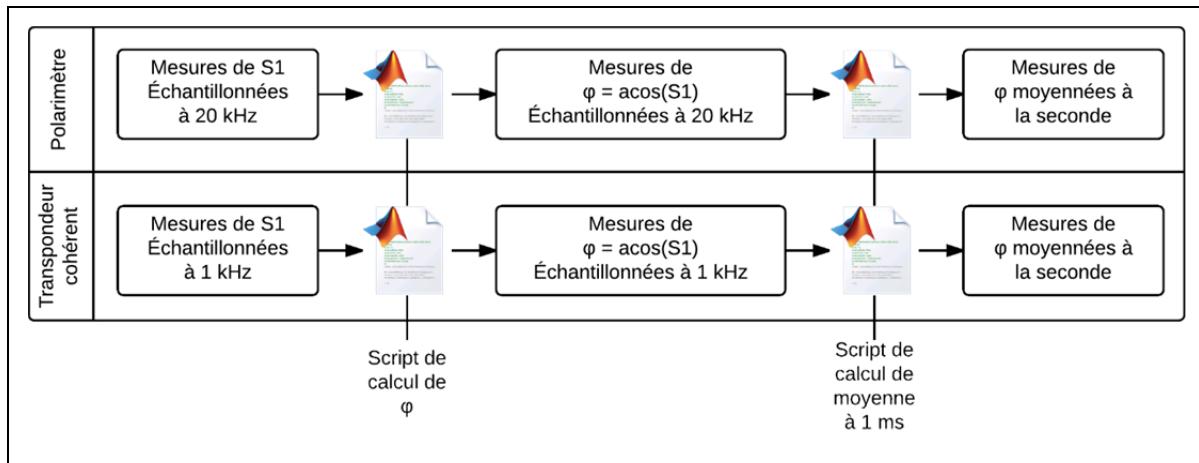


Figure 3.12 Première étape du calcul de φ à partir des mesures du polarimètre et des mesures du transpondeur cohérent

Cette première étape consiste à rendre comparables les mesures effectuées par chaque instrument. Pour cela, la valeur de φ est calculée à chaque instant à partir des mesures brutes de chaque instrument, échantillonnées à 20 kHz pour le polarimètre et à 1 kHz pour le transpondeur cohérent. Les valeurs obtenues sont ensuite moyennées à la seconde en utilisant les scripts Matlab présentés aux sections 2.2.3 et 2.3.3. Ces données peuvent être utilisées pour procéder à l'étape suivante qui consiste à comparer les deux types de mesure.

La seconde étape, présentée à la Figure 3.13 ci-dessous consiste, dans un premier temps, à synchroniser les mesures effectuées par les deux instruments. Pour cela, la synchronisation des mesures du polarimètre avec celles du transpondeur cohérent est assurée en interpolant les valeurs de φ à l'aide de la fonction Matlab *interp1*.

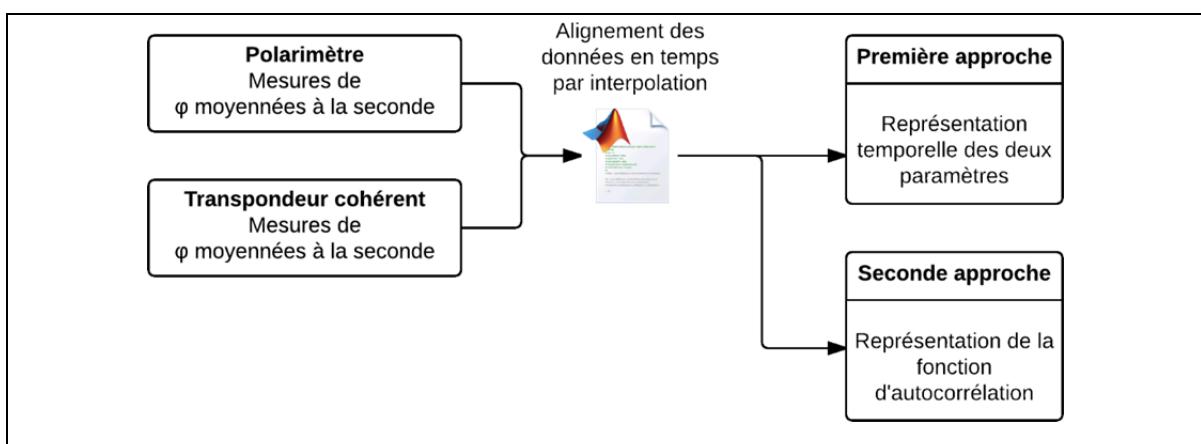


Figure 3.13 Seconde étape de calcul de φ à partir des mesures du polarimètre et des mesures du transpondeur cohérent

Deux analyses comparatives ont ensuite été menées pour comparer les deux instruments. Ces analyses sont réalisées à travers le script Matlab *comparison_ocld_pola*, qui est présenté à l'ANNEXE XIII, page 197.

La première analyse comparative consiste en une analyse dans le temps de l'évolution temporelle de S_1 mesuré par les deux instruments sur une même période. Ce graphique est présenté à la Figure 3.14. Compte tenu de la période temporelle relativement importante couverte par les mesures (5 jours), un moyennage temporel à la minute est effectué avant le

tracé graphique en utilisant la fonction *filter* de Matlab représentant un filtre à moyenne glissante dont l'équation est présentée à l'équation (3.7), tirée de la documentation de Matlab,

$$y(n) = \frac{1}{w_{size}} \left(\sum_{k=n}^{n-(w_{size}-1)} x(k) \right) \quad (3.7)$$

où w_{size} représente la largeur de fenêtre glissante, x désigne le vecteur contenant les valeurs des paramètres avant le filtrage et y le vecteur contenant les valeurs des paramètres après filtrage.

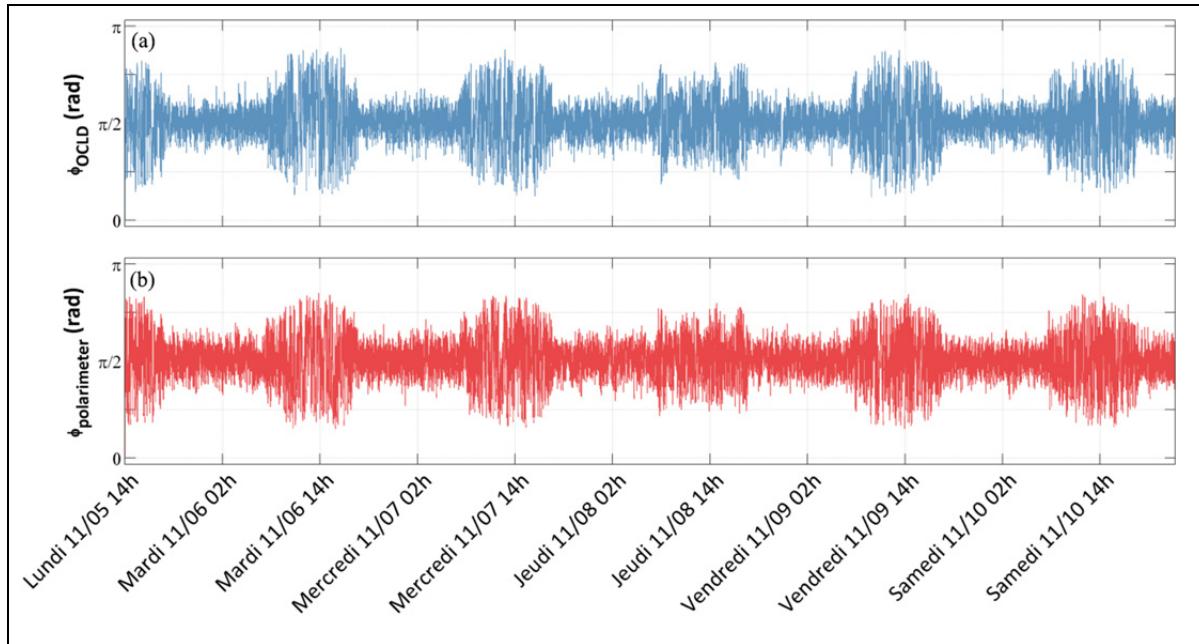


Figure 3.14 Représentation temporelle de ϕ mesuré par (a) le polarimètre et (b) le transpondeur cohérent (novembre 2011)

On peut observer sur les deux courbes une même tendance : la fréquence de variation de ϕ , que l'on peut voir comme une représentation de l'activité de polarisation, subit une évolution cyclique de l'ordre du jour. L'activité semble ainsi plus élevée chaque jour entre 7h00 et 20h00. Ce phénomène a déjà été observé à travers les travaux de (Leeson, Bao et Côté,

2009), qui ont étudié l'influence de la température et du vent sur l'activité de polarisation du signal dans une liaison OPGW de 33 km. Une comparaison des mesures réalisées sur la liaison de Verizon et l'évolution de la température est proposée dans la partie suivante. Cette représentation temporelle constitue un point de départ sur des études comparatives ultérieures qui pourront être réalisées.

La seconde analyse comparative porte sur le calcul de la corrélation croisée entre les mesures du polarimètre et celles du transpondeur cohérent. La méthode de calcul utilisée est similaire à celle utilisée pour la fonction d'autocorrélation à la partie 3.2.2. La fonction *xcov* de Matlab est alors non pas utilisée sur un seul vecteur de mesures mais sur les mesures respectives du transpondeur cohérent et du polarimètre. L'autocovariance ainsi mesurée est ensuite divisée par le produit de la variance des deux ensembles de données du polarimètre et du transpondeur cohérent pour obtenir l'autocorrélation, selon l'équation (3.8), tirée de la documentation de Matlab.

$$CCF(x, y) = \frac{\sum_{n=0}^{N-m-1} (x_{n+m} - \bar{x})(y_n - \bar{y})}{(\sum_{k=1}^N (x_k - \bar{x})^2)(\sum_{k=1}^N (y_k - \bar{y})^2)} \quad (3.8)$$

où CCF désigne la matrice de corrélation croisée, x et y les vecteurs contenant les valeurs de φ mesurées respectivement par le polarimètre et le transpondeur cohérent et \bar{x} et \bar{y} les moyennes respectives de x et y.

La représentation de la fonction de corrélation croisée appliquée aux données de novembre 2012 est proposée à la Figure 3.15.

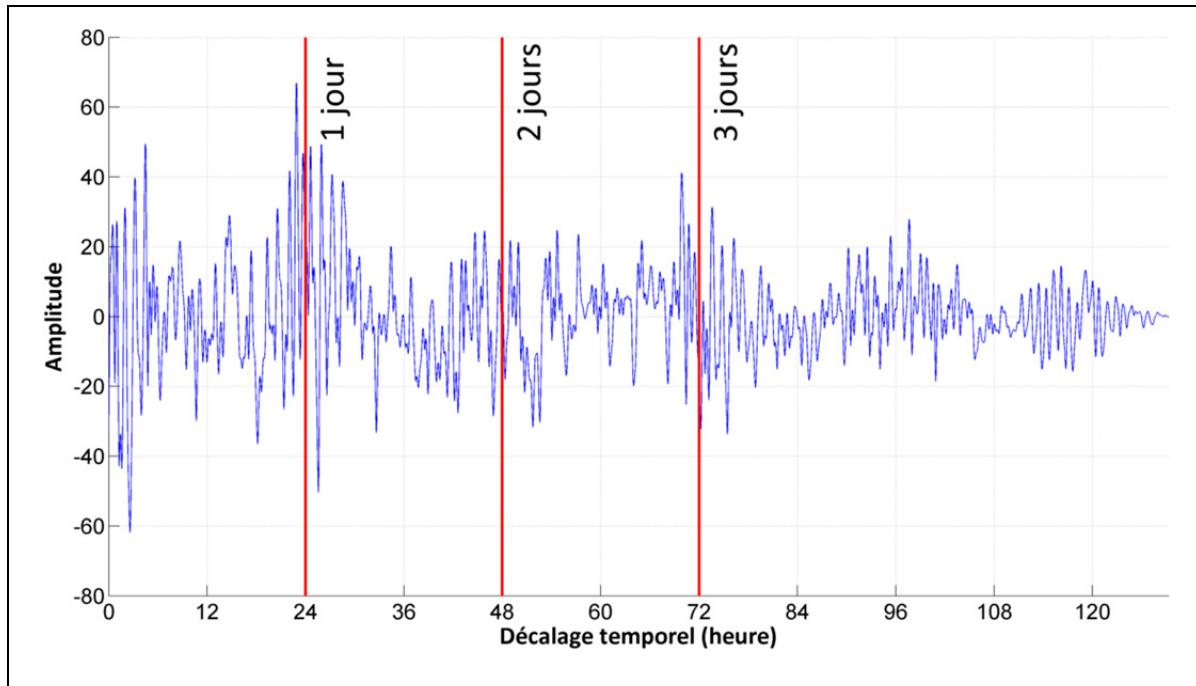


Figure 3.15 Représentation de la corrélation croisée des mesures du polarimètre et des mesures du transpondeur cohérent (novembre 2012)

La figure ci-dessus nous permet de constater qu'il y a une forte corrélation entre les mesures du transpondeur cohérent et les mesures du polarimètre, qui se traduit par des valeurs extrêmales observées dans corrélation croisée pour des décalages temporels qui sont approximativement des multiples d'une période de 24 heures. On peut constater globalement que les mesures sont entachées de bruit, ce qui se traduit par de multiples oscillations dans la corrélation croisée. Cette étude permet donc d'avoir un premier aperçu de la comparaison des deux instruments mais l'analyse pourrait être approfondie en travaillant sur l'alignement des états de polarisation entre les instruments d'après (Crawford et al., 2014).

3.6 Analyse de l'influence de la température sur la liaison

Cette partie présente une méthode permettant de comparer l'évolution temporelle de paramètres donnés, tels que la DGD, la PDL ou les composantes de Stokes à l'évolution de la température moyenne du lien étudié. Ma contribution a porté sur la création d'un script Matlab *plot_temperature_parameter*, présenté à l'ANNEXE XIV, page 155 permettant de

tracer simultanément l'évolution temporelle d'un paramètre et de la température, pourvu que les mesures des paramètres soient datées de manière précise.

3.6.1 Acquisition et traitement des données de température

La méthode utilisée pour traiter les données de température est présentée à la Figure 3.16 et se divise en trois étapes qui vont permettre de traiter les données de température en vue de les tracer pour une période quelconque.

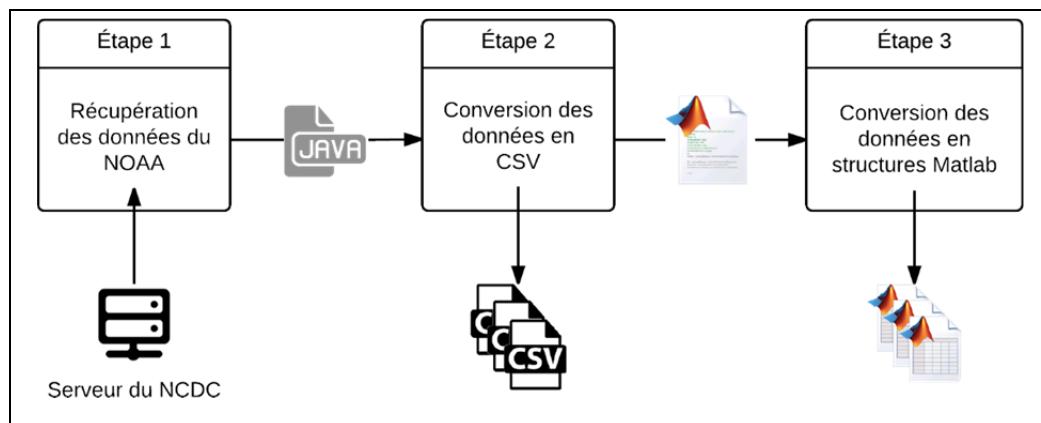


Figure 3.16 Traitement des données météorologiques

La première étape a consisté à récupérer des informations sur la localisation géographique des différents points de passage de la liaison de Verizon. Il a ainsi été possible d'obtenir 14 points de passage de la liaison, soit en moyenne un point de passage tous les 70 kilomètres. L'étape suivante consiste à récolter les données météorologiques de ces 14 lieux. Dans le cadre du projet, les données utilisées sont celles de l'organisme *National Oceanic and Atmospheric Administration (NOAA)*, archivées sur les serveurs du *National Climatic Data Center (NCDC)*. Le NOAA met à disposition des archives pour chaque station météorologique et pour chaque année. Les données sont stockées dans un format propre au NOAA et accompagnées d'une documentation expliquant comment les convertir en format CSV.

La seconde étape repose sur la création d'un script Matlab convertissant les structures météo du format CSV au format Matlab : *DS3505_processing* présenté à l'ANNEXE XV, page 207. Cette étape de conversion rend possible le tracé graphique de la température. Il est important, lors du stockage de la structure contenant les évolutions de température, de consigner aussi les dates correspondant aux relevés et de les convertir au bon fuseau horaire car les données du NOAA sont enregistrées au fuseau horaire GMT.

La troisième étape consiste à calculer une valeur de température représentative de l'ensemble de la liaison. L'approche adoptée a été de calculer à chaque instant la valeur moyenne sur l'ensemble des 14 stations. Les valeurs obtenues sont ensuite enregistrées dans un fichier Matlab afin de pouvoir être récupérées ultérieurement.

3.6.2 Activité de polarisation et température

(Leeson, Bao et Côté, 2009) suggèrent que la température peut avoir une influence sur les variations d'états de polarisation. La capacité à récupérer l'évolution de la température sur la liaison de Verizon a permis de tracer sur les mêmes graphiques l'évolution de la température et de la composante de polarisation ϕ . Les résultats obtenus durant le mois de novembre 2011 sont présentés à la Figure 3.17. On peut constater que les périodes où l'activité de polarisation est la plus élevée correspondent aux valeurs maximales de température. Ceci semble confirmer que la température peut avoir une influence sur l'activité de polarisation. Ce phénomène peut aussi s'expliquer par l'augmentation des contraintes mécaniques exercées sur la fibre lorsqu'elle est soumise à des températures élevées (Leeson, Bao et Côté, 2009). Une autre hypothèse porte sur le fait que l'activité de polarisation est plus généralement importante le jour. Cette observation pourrait résulter de l'activité humaine dans les locaux des opérateurs contenant les systèmes de transmission.

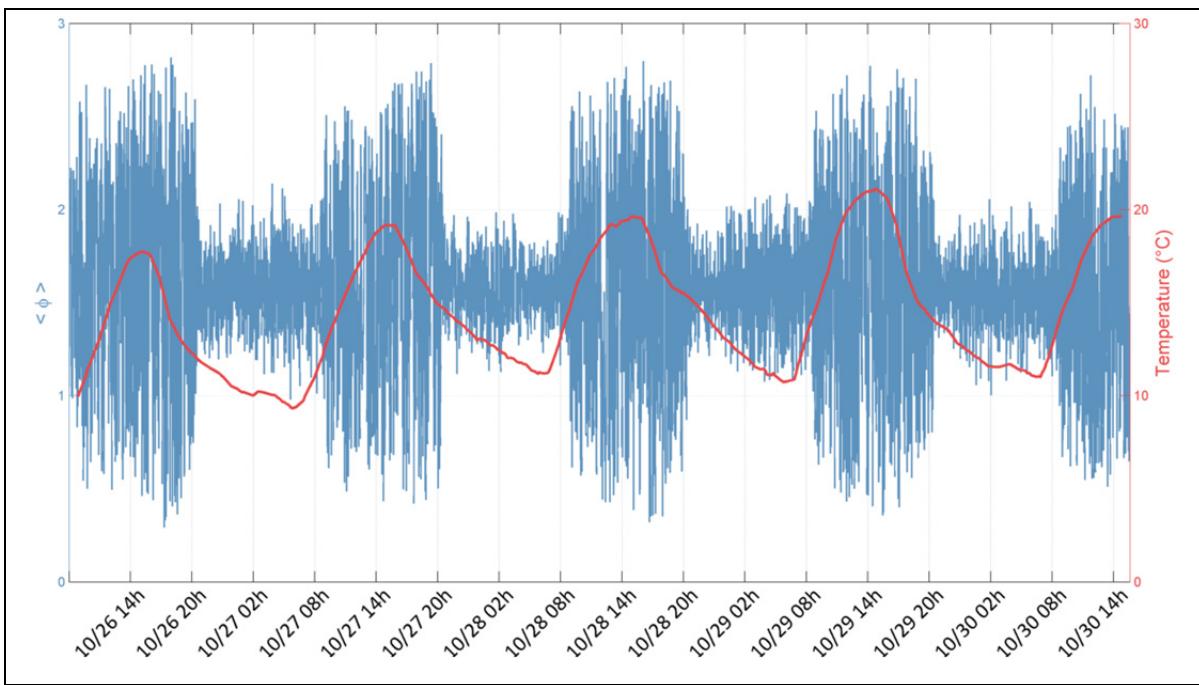


Figure 3.17 Évolution temporelle de $\langle\phi\rangle$ mesurée par le transpondeur cohérent (représentée en bleu) comparée à l'évolution de la température (représentée en rouge) sur une même période (novembre 2012)

3.7 Conclusion du chapitre

L'analyse présentée dans ce chapitre a permis d'assurer la continuité des travaux de (Mortelette, 2014) et de poursuivre le traitement et l'analyse des données sur plusieurs saisons. Par ailleurs, la méthode de traitement de données a été optimisée afin de pouvoir mieux s'adapter à l'exploitation des mesures, notamment par rapport à la fonction d'autocorrélation adaptée de (Woodward et al., 2014) et le calcul du temps de dérive de la DGD.

Les efforts ont aussi porté sur une nouvelle étude : l'analyse temporelle des mesures du polarimètre. En raison du volume de données largement plus important que celui provenant du transpondeur cohérent, le développement d'une nouvelle méthode de traitement a été nécessaire pour calculer la valeur moyenne des paramètres mesurés. Cette analyse a rendu possible l'étude de l'évolution des effets de polarisation sur plusieurs saisons.

Cette dernière étude a aussi permis de comparer les mesures du polarimètre avec celles du transpondeur cohérent. En effet, compte tenu du déploiement progressif des transpondeurs cohérents, l'utilisation de ceux-ci comme moniteurs de performances en matière de polarisation pourrait faciliter la gestion des réseaux optiques par les opérateurs.

Enfin, une comparaison des mesures effectuées avec la température moyenne observée dans la liaison apporte des explications possibles sur l'origine des variations cycliques des effets de polarisation avec une période d'approximativement une journée.

Les différents points abordés dans ce chapitre permettent donc de disposer de bases expérimentales pour les travaux futurs relatifs aux effets de polarisation.

CHAPITRE 4

CARACTÉRISATION DES EFFETS DE POLARISATION AVEC UN POLARIMÈTRE

4.1 Introduction

Ce chapitre est consacré à la caractérisation des effets de polarisation détectés dans la liaison aérienne de Verizon au moyen du polarimètre. L'emphase est portée, dans un premier temps, sur la définition des effets de polarisation ainsi que sur leurs caractéristiques. L'analyse des effets de polarisation est ensuite réalisée sur l'ensemble des mesures effectuées durant la période 2012 – 2014, ainsi que l'analyse en fréquence des paramètres de Stokes. Enfin, une analyse des caractéristiques du bruit du polarimètre est présentée.

Le suivi de l'évolution des effets de polarisation dans une liaison optique constitue un aspect capital d'un système de transmission à multiplexage en polarisation dont le récepteur doit assurer en permanence le suivi de l'état de polarisation du signal pour le décoder. Chaque système de transmission est conçu pour tolérer les variations de l'état de polarisation jusqu'à un certain seuil de rapidité et d'amplitude. L'expérience prouve que, bien que la connaissance de distributions statistiques permette de définir les seuils de performance des systèmes de transmission, certains événements isolés peuvent survenir et provoquer une brève interruption dans le suivi de l'état de polarisation, qui peut entraîner une panne et une rupture de service. Un des objectifs du projet est de récolter suffisamment de données dans une liaison aérienne pour détecter et caractériser en amplitude et en vitesse les variations les plus brusques d'états de polarisation du signal optique, et ainsi de fournir des lignes directives pour la conception des modules de traitement du signal des systèmes de transmission cohérents. Il est à noter que l'hypothèse de nos travaux était que les variations les plus rapides et les plus grandes de l'état de polarisation se produisent vraisemblablement dans les liaisons aériennes.

Les données du polarimètre collectées dans ce projet présentent plus de 450 000 blocs de mesures de 10 secondes chacun correspondant à 181 jours de collecte sur une période de 22 mois couvrant plusieurs saisons. Les données ont été acquises durant des intervalles continus allant de 2 à 44 jours.

4.2 Effets de polarisation dans une liaison optique

Cette partie présente les effets de polarisation observables dans une liaison optique et leurs spécificités. L'étude des effets de polarisation nécessite l'introduction de certains concepts qui sont indispensables à leur compréhension. Ces concepts sont présentés dans un premier temps. Une présentation des différents types de fluctuations d'états de polarisation observés dans la liaison aérienne est ensuite introduite.

4.2.1 État de polarisation de la lumière et représentation de Stokes

Nous avons vu au premier chapitre que le signal optique transporté dans les liaisons optiques est une onde électromagnétique polarisée. C'est-à-dire que le signal optique dispose en tout instant d'un état de polarisation clairement défini, qui peut être représentée par un point sur la sphère de Poincaré. Le vecteur partant du centre de cette sphère et joignant le point correspondant à son état de polarisation est appelé vecteur de Stokes. Un exemple d'évolution temporelle du vecteur de Stokes, mesurée par le polarimètre sur une période de 25 millisecondes, est présenté à la Figure 4.1, dans laquelle les vecteurs vert et rouge correspondent respectivement au vecteur de Stokes au début et à la fin de la période et le tracé en bleu correspond aux positions successives de ce vecteur durant ce laps de temps.

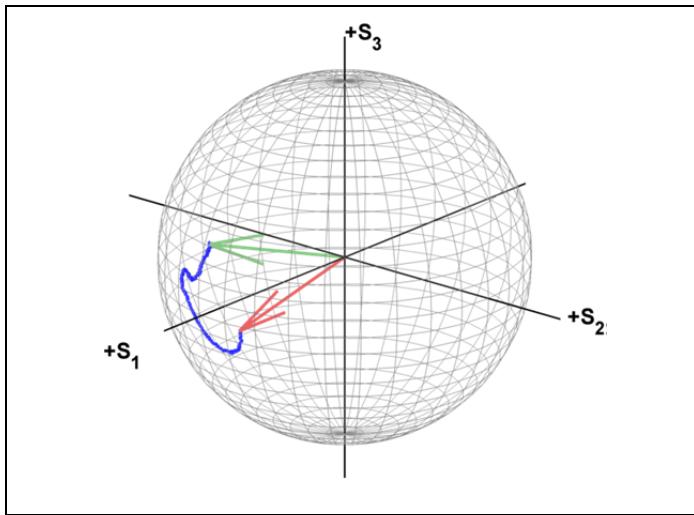


Figure 4.1 Évolution du vecteur de Stokes (en bleu) sur une période de 25 millisecondes, les vecteurs en début et en fin de période sont représentés en rouge et en vert

La figure illustre qu'il est possible, sur une période donnée, de tracer l'évolution de l'état de polarisation du signal optique à travers le tracé du vecteur de Stokes. C'est justement à travers l'étude de la vitesse de variation de l'état de polarisation que nous allons définir les effets de polarisation.

4.2.2 Seuils pour la détection d'événement de polarisation

La notion d'événement de polarisation a été présentée à la section 2.3.2.4. Nous avons vu que la définition d'un événement repose sur une valeur seuil de 0,9 rad/ms et une fenêtre temporelle de 10 millisecondes.

La valeur de seuil (0,9 rad/ms) a été choisie sur la base de l'observation que la majorité des valeurs de vitesse enregistrées sont inférieures à cette valeur et donc que les événements "intéressants" à caractériser sont situés au-dessus de cette valeur. L'hypothèse choisie est donc que les valeurs de rotations inférieures à 0,9 rad/ms résultent soit du bruit, soit des fluctuations de polarisation faibles par rapport à la vitesse du module de traitement numérique du signal associé au transpondeur cohérent. La fenêtre temporelle de 10 ms a été

choisie arbitrairement sur la base de l'hypothèse que deux fluctuations séparées de plus de 10 ms peuvent être considérées comme indépendantes.

On peut ainsi déterminer trois valeurs caractéristiques lors d'un événement : la vitesse de rotation du vecteur de Stokes maximal durant l'événement, la déflexion angulaire maximale et la déflexion angulaire finale de l'événement. Si cette dernière valeur est proche de 0, il s'agit d'un événement élastique. Dans le cas contraire, il s'agit d'un événement inélastique.

4.2.3 Événements de polarisation élastiques et inélastiques

On peut distinguer deux catégories d'événements de polarisation : les variations élastiques et inélastiques (Boroditsky et al., 2005). Ainsi, un événement est qualifié d'élastique si, à la fin de celui-ci, l'état de polarisation revient à un état proche de celui au début de l'événement. Similairement, une fluctuation rapide d'état de polarisation est qualifiée d'inélastique si l'état de polarisation à la fin de celui-ci est radicalement différent de l'état de polarisation initial.

La Figure 1.6 présente les deux types d'événements de polarisation cités précédemment mesurés sur la liaison de Verizon par le polarimètre en novembre 2012. Sur cette figure, le tracé en rouge correspond à l'effet de polarisation tandis que les 50 millisecondes précédant et suivant cet effet sont représentées respectivement en bleu et en vert. La Figure 4.3 et la Figure 4.4 présentent quant à elles respectivement l'évolution temporelle des composantes de Stokes et la vitesse de rotation du vecteur de Stokes pour ces mêmes événements. Sur les figures 4.3 et 4.4, les lignes rouges verticales représentent le début et la fin de l'événement. On peut remarquer que dans le cas d'un effet élastique, les trois composantes de Stokes tendent à rejoindre leur valeur en début d'effet contrairement à l'effet inélastique.

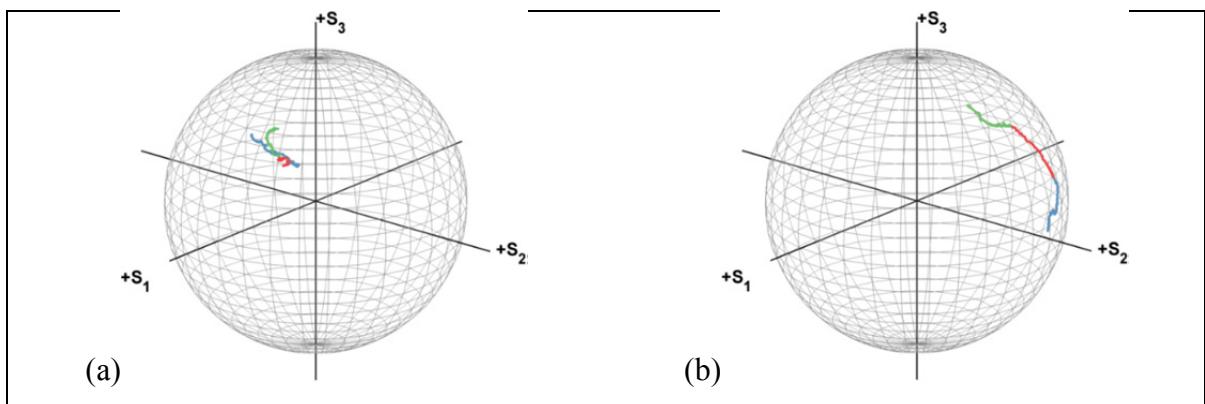


Figure 4.2 Représentation d'un changement d'état de polarisation élastique (a) et inélastique (b) sur la sphère de Poincaré (novembre 2012)

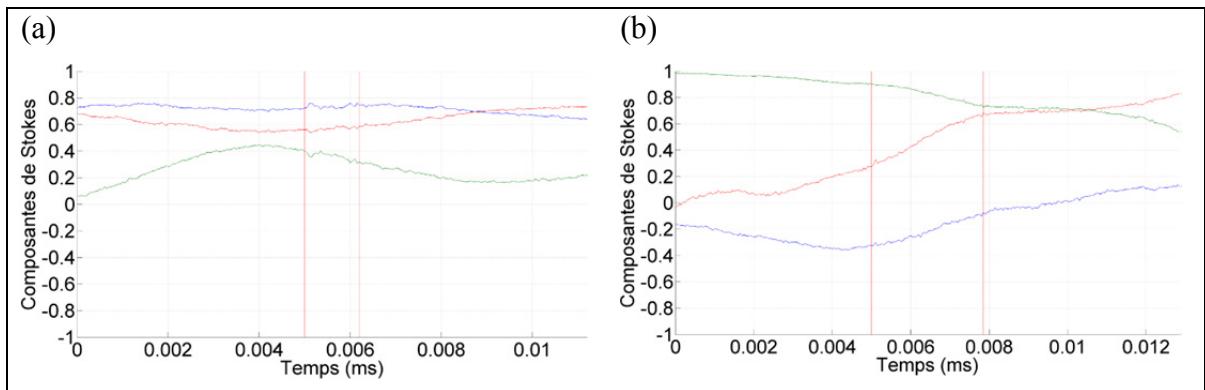


Figure 4.3 Évolution temporelle des trois composantes de Stokes lors d'un changement d'état de polarisation élastique (a) et inélastique (b) (novembre 2012)

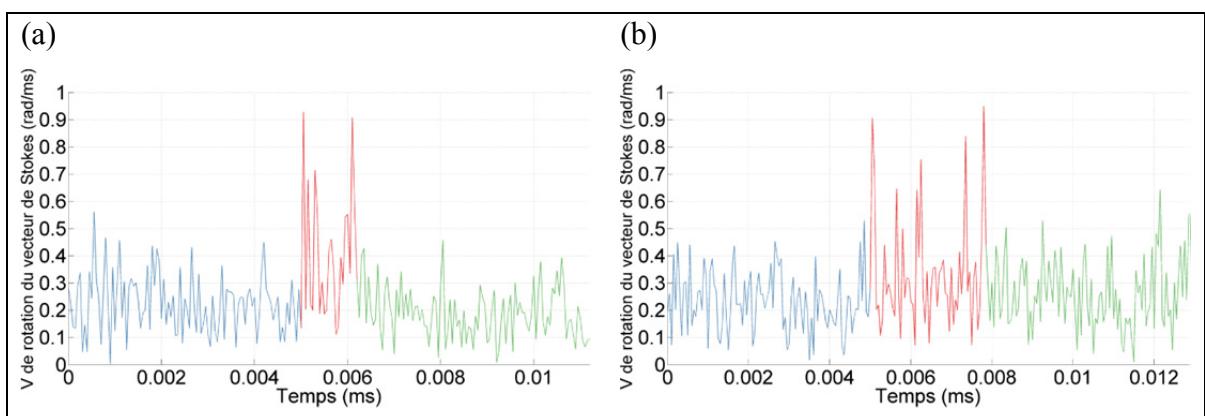


Figure 4.4 Évolution temporelle de la vitesse de rotation du vecteur de Stokes lors d'un changement d'état de polarisation élastique (a) et inélastique (b) (novembre 2012)

Une partie de l'analyse des données du polarimètre va consister à dresser une base de données de l'ensemble des fluctuations rapides de polarisation se produisant dans la liaison. Il est pour cela nécessaire de déterminer une règle permettant de distinguer les effets de polarisation rapides du bruit du polarimètre lors du traitement des données mesurées.

4.3 Distribution des vitesses de rotation du vecteur de Stokes

Le Tableau 4.1 présente un sommaire des valeurs de vitesse de rotation du vecteur de Stokes observées sur la liaison entre 2012 et 2014. Un histogramme de l'ensemble des valeurs de rotation du vecteur de Stokes observées dans la liaison aérienne a été réalisé. La méthode de traitement des données du polarimètre a été présentée à la section 2.3.2. Nous allons nous intéresser ici à la construction de l'histogramme et à l'analyse des résultats obtenus.

Tableau 4.1 Caractérisation des valeurs de vitesse de rotation du vecteur de Stokes mesurées par le polarimètre entre octobre 2012 et septembre 2014

Saison (nombre de points de mesure)	Min	Max	Moyenne	Écart type
	(rad/ms)			
Automne 2012 ($5,99 \times 10^9$)	0	4,92	0,203	0,11
Automne 2013 ($1,78 \times 10^{10}$)	0	9,09	0,177	0,0966
Hiver 2013 ($2,99 \times 10^{10}$)	0	15,2	0,187	0,102
Printemps 2014 ($1,20 \times 10^{10}$)	0	5,16	0,181	0,0979
Été 2014 ($2,74 \times 10^{10}$)	0	6,35	0,174	0,0941
Automne 2014 ($3,38 \times 10^9$)	0	2,75	0,173	0,0930

On peut observer que la valeur moyenne des vitesses de rotation du vecteur de Stokes reste sensiblement comparable sur l'ensemble des périodes de mesure, avec une valeur allant de 0,173 rad/ms à 0,203 rad/ms. Cependant, les valeurs de vitesse maximale observées ont une plus grande amplitude, avec des valeurs de vitesse allant de 2,75 rad/ms en automne 2014 à une valeur maximale de 15,2 rad/ms en hiver 2013. Il n'apparaît pas de corrélation évidente entre les vitesses maximales observées et les saisons correspondantes.

4.3.1 Détermination de la largeur optimale des colonnes de l'histogramme

Une des premières étapes précédant le tracé de la matrice des vitesses de rotations du vecteur de Stokes consiste à déterminer la valeur optimale de la largeur des colonnes de l'histogramme. (Scott, 1992) préconise l'utilisation de la méthode de Freedman–Diaconis. Cette règle définit la largeur optimale des colonnes d'histogramme par l'équation (4.1),

$$h = \frac{2 \cdot EI(x)}{\sqrt[3]{n}} \quad (4.1)$$

où h désigne la largeur de colonne de l'histogramme, EI l'écart interquartile des mesures de vitesse de rotation dans un fichier et n le nombre de points de mesure dans un fichier.

Le calcul de la valeur moyenne de h sur l'ensemble des fichiers du polarimètre donne une valeur moyenne de l'ordre de 0,01 rad/ms. C'est donc cette valeur qui sera utilisée pour le tracé de l'histogramme des valeurs de vitesse de rotation.

4.3.2 Histogramme des vitesses de rotation du vecteur de Stokes

En plus d'optimiser le traitement des données du polarimètre comme présenté à la section 2.2.2, ma contribution a porté sur l'intégration de l'ensemble des résultats mesurés par le polarimètre à l'histogramme des vitesses de rotation du vecteur de Stokes. L'ensemble des données analysées totalise $9,68 \times 10^{10}$ points de mesure, capturés entre octobre 2012 à octobre 2014, alors que les données analysées par (Mortelette, 2014) totalisaient $7,80 \times 10^9$ points de mesure. Le script Matlab ayant permis le tracé de l'histogramme porte le nom de *plot_hist_arr* et est présenté à l'ANNEXE XVI, page 211.

La valeur moyenne et l'écart type des vitesses de rotation ont été calculés pour chaque fichier, ce qui a permis d'obtenir des propriétés pour l'ensemble des données. L'histogramme des vitesses de rotation du vecteur de Stokes pour l'ensemble des données est présenté à la

Figure 4.5. La moyenne \bar{x} et l'écart type σ des valeurs de vitesses de rotation observées sont de 0,1812 rad/ms et 0,0987 rad/ms respectivement.

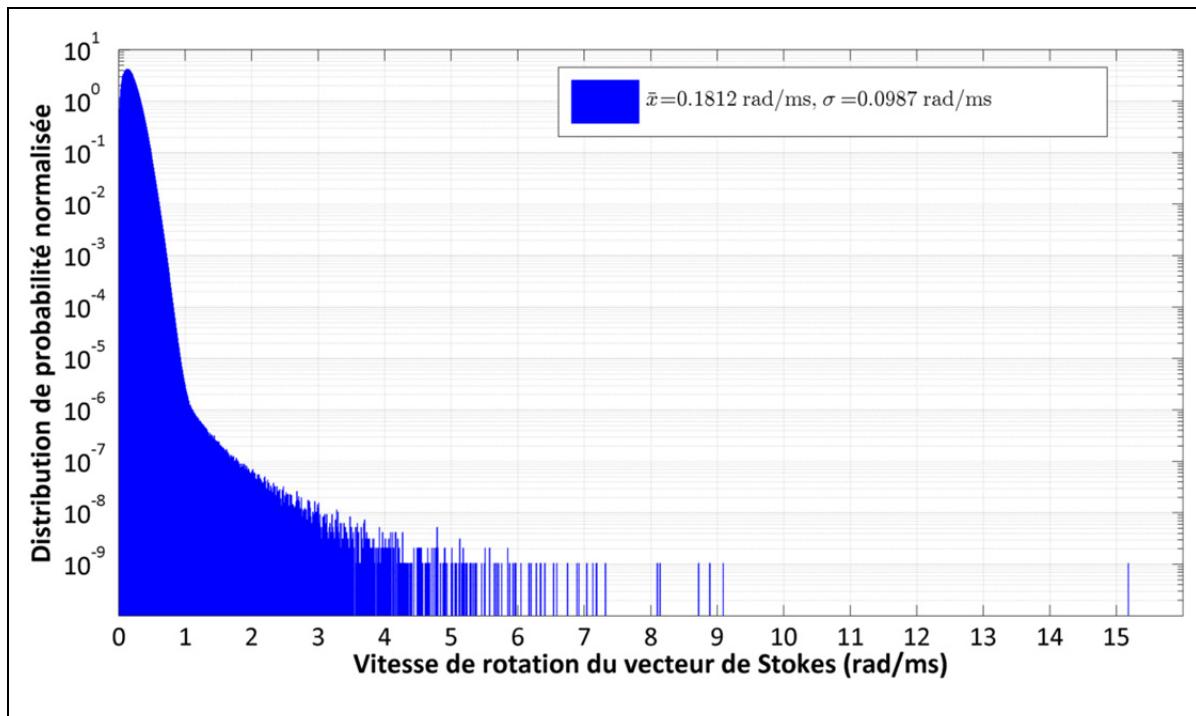


Figure 4.5 Histogramme des vitesses de rotation du vecteur de Stokes sur la liaison aérienne (octobre 2012 à octobre 2014). L'histogramme est présenté sur une échelle logarithmique

Tel qu'observé précédemment par (Mortelette, 2014), la distribution des vitesses de rotation du vecteur de Stokes comporte deux parties : la première correspond aux vitesses de rotation inférieures à 1,2 rad/ms et suit une loi de type Rayleigh tandis que les valeurs les plus élevées tendent à suivre une loi de type exponentielle. La présence de deux distributions résulte selon toute vraisemblance de la superposition de deux événements d'origine physique différente.

On peut observer que les vitesses de rotation ont une valeur moyenne de 0,1812 rad/ms et un écart type de 0,0987 rad/ms. Ces observations restent dans le même ordre de grandeur que les observations faites par (Mortelette, 2014), qui rapportaient pour une période de mesure plus courte une valeur moyenne de 0,1982 rad/ms et un écart type de 0,1054 rad/ms.

4.3.3 Ajustement de l'histogramme des vitesses de rotation du vecteur de Stokes par une distribution de Rayleigh

L'étape suivante consiste à analyser plus en détail la forme de la distribution de probabilité des vitesses de rotation de l'état de polarisation. (Matsuda et al., 2009) a proposé d'ajuster cette distribution par une distribution de Rayleigh, définie à l'équation (4.2),

$$v(r, \alpha) = \frac{r}{\alpha^2} e^{-\frac{1}{2}(\frac{r}{\alpha})^2} \quad (4.2)$$

où r désigne la variable aléatoire étudiée, en l'occurrence la vitesse de rotation α désigne le paramètre strictement positif de la loi de Rayleigh.

Ma contribution a porté sur la réalisation d'un script Matlab adapté des travaux de (Mortelette, 2014), permettant de vérifier dans quelle mesure la distribution correspond à une distribution de Rayleigh. Ce script, qui porte le nom de *fit_hist_arr* et est présenté à l'ANNEXE XVII, page 215, prend en entrée l'histogramme des valeurs normalisées de vitesses de rotation et procède à l'ajustement statistique avec une loi de Rayleigh.

Les ajustements de courbe ont été réalisés à l'aide de la librairie logicielle *Statistics Toolbox* interne à Matlab. La fonction utilisée est la fonction *NonLinearModel.fit* qui permet d'ajuster une fonction à une distribution en utilisant l'algorithme de Levenberg-Marquardt, qui est décrit dans les travaux de (Mortelette, 2014). Le fonctionnement de cet algorithme, qui est itératif, est basé sur la méthode pondérée des moindres carrés.

La fonction *NonLinearModel.fit* prend en paramètre plusieurs variables :

- la variable contenant les valeurs limites des colonnes de l'histogramme;
- la variable contenant les valeurs de l'histogramme normalisé;
- l'expression de la fonction que l'on souhaite ajuster, qui correspond à l'équation (4.2).

Il est à noter que l'algorithme de Levenberg-Marquardt est une méthode itérative et qu'il est nécessaire de fournir une valeur initiale du paramètre à estimer, en l'occurrence le paramètre de Rayleigh noté α . La valeur initiale choisie est l'estimateur biaisé du maximum de vraisemblance, défini à l'équation (4.3).

$$\hat{\alpha} = \sqrt{\frac{1}{2N} \sum_{i=1}^N x_i^2} \quad (4.3)$$

où N désigne le nombre d'éléments du vecteur x. L'utilisation d'un algorithme pour effectuer des ajustements de courbe s'accompagne de deux indicateurs permettant de juger de la qualité de l'ajustement. Le premier indicateur utilisé est le coefficient de corrélation de Pearson, noté R_p , et permet de mesurer une dépendance linéaire entre les paramètres estimés par une distribution de Rayleigh et les paramètres mesurés. Le second indicateur repose sur les intervalles de confiance relatifs aux paramètres estimés : cet indicateur est défini pour un intervalle de confiance donné et est donné sous la forme d'une paire de valeurs $(\alpha_{min}, \alpha_{max})$ correspondant respectivement à la valeur minimale et la valeur maximale du paramètre estimé.

Pour un échantillon donné, le coefficient de corrélation de Pearson est donné par l'équation (4.4),

$$R_p = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.4)$$

ou x désigne le paramètre mesuré et y le paramètre correspondant à l'ajustement de x. La valeur du coefficient de Pearson est située entre -1 et 1.

Dans la librairie logicielle *Statistics Toolbox*, Matlab met à disposition la fonction *coefCI* permettant de déterminer les intervalles de confiance relatifs à l'estimation des paramètres.

L'application de cette fonction à l'équation de Rayleigh permet d'obtenir l'intervalle de confiance sur l'estimation du paramètre σ à 95%.

Le script de traitement décrit précédemment a été appliqué à l'ensemble des données mesures par le polarimètre et le résultat est présenté à la Figure 4.6,

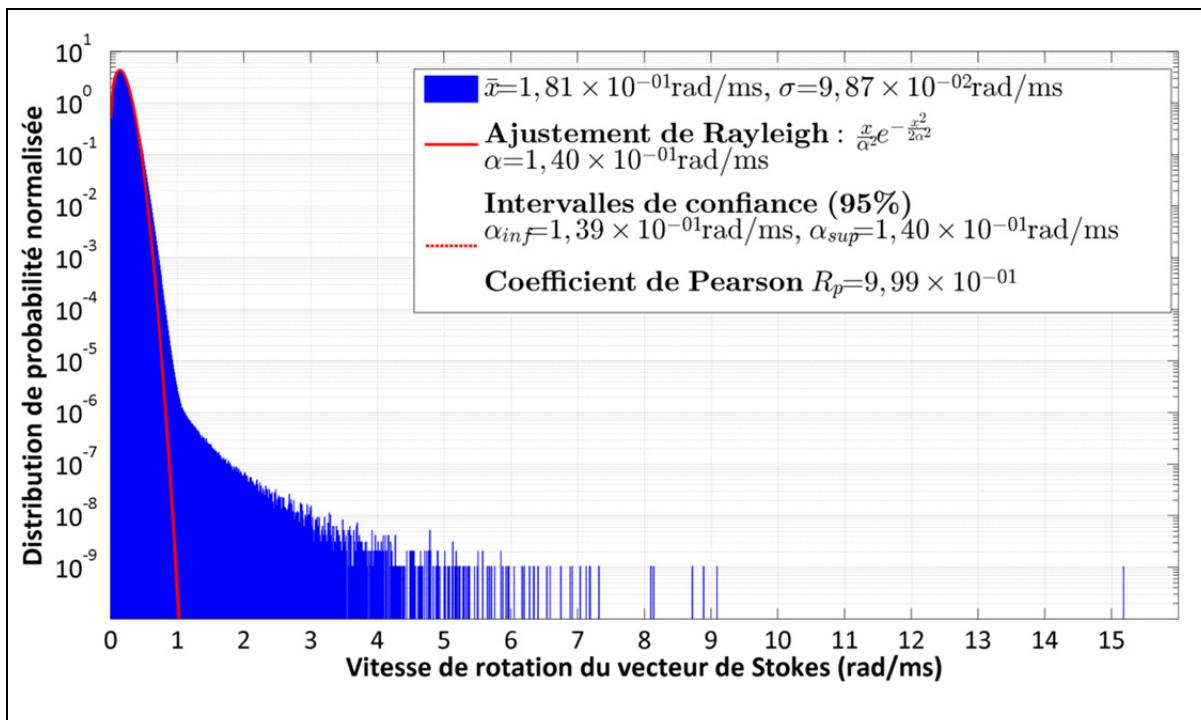


Figure 4.6 Ajustement d'une loi de Rayleigh sur les vitesses de rotation du vecteur de Stokes sur la liaison aérienne (octobre 2012 à octobre 2014)

où R_p désigne le coefficient de Pearson et $(\alpha_{min}, \alpha_{max})$ correspondant respectivement à l'estimation du paramètre de Rayleigh par défaut et par excès.

L'ajustement statistique obtenu permet d'obtenir une valeur du paramètre α de 0,1395 rad/ms avec un coefficient de Pearson $R_p = 0,9987$. Les valeurs des intervalles de confiance ont aussi été calculées et il a été permis de définir les bornes de confiance à 95 % entre 0,1390 rad/ms et 0,1400 rad/ms. A titre de comparaison (Mortelette, 2014) a obtenu une valeur de $\alpha = 0,1576$ rad/ms pour la seule période d'octobre/novembre 2012.

Deux études relatives à la distribution des vitesses de rotation du vecteur de Stokes avaient été menées précédemment : l'une par (Peterson, Leo et Rochford, 2004) dans une liaison enterrée de 110 km, avec un temps d'échantillonnage d'une milliseconde avait montré que la valeur maximale de vitesse de rotation observée est de l'ordre de 0,02 rad/ms et que la distribution obtenue est proche d'une loi de Rayleigh. L'autre étude a été mené par (Leo et al., 2003) et a porté sur l'étude des vitesses de rotation du vecteur de Stokes sur quelques mètres de fibre optique soumises à d'importantes secousses. L'ensemble des observations réalisées porte sur des valeurs de vitesse de rotation inférieures à 1,0 rad/ms. Les vitesses de rotation alors obtenues suivent bien une distribution de Rayleigh. Dans les deux études, les vitesses de rotation ne dépassaient pas 1 rad/ms et leur distribution suivait une loi de Rayleigh. Fait important, aucune trace de la partie exponentielle de la distribution n'avait alors été rapportée.

Par ailleurs, (Matsuda et al., 2009) ont étudié la distribution des vitesses de rotation du vecteur de Stokes supérieures à 0,001 rad/ms sur une liaison aérienne de 33,4 km pendant 75 jours et l'ont ajustée à une loi de Rayleigh en obtenant un paramètre $\alpha = 3,4 \times 10^{-3}$ rad/ms. L'événement de polarisation avec la plus grande amplitude observée dans cette période correspond à une vitesse de rotation de 0,59 rad/ms. La faible valeur du paramètre α comparativement aux résultats obtenus dans ce projet peut s'expliquer d'une part par la faible durée des mesures réalisées (75 jours, comparativement à 2 ans dans le cas présent) et d'autre part par la faible longueur de la liaison. D'autres mesures ont été effectuées par (Roberge, Audet et Bernier, 2009) sur une liaison aérienne de 90 km sur une durée de l'ordre de l'heure et obtient aussi un ajustement de Rayleigh. L'événement le plus rapide alors observé à une vitesse maximale de 1000 degrés par seconde soit en moyenne 0,0175 rad/ms. Cependant, mis à part les travaux de (Mortelette, 2014), il n'y a pas eu d'observations de valeurs de vitesse de rotation supérieure à 2,0 rad/ms.

On remarque que pour les valeurs de vitesses de rotation du vecteur de Stokes inférieures à 1,1 rad/ms montrent un bon ajustement avec une distribution de Rayleigh ce qui est prévisible compte tenu des variations aléatoires des états de polarisation en l'absence d'effets

rapides de polarisation selon (Crawford et al., 2014). Cependant, la distribution dérive significativement d'une distribution de Rayleigh pour des valeurs de vitesse de rotation supérieure à 2,0 rad/ms, ce qui laisse supposer que d'autres phénomènes physiques sont à l'origine de vitesses de rotation plus rapides.

4.3.4 Ajustement de l'histogramme pour les vitesses de rotation du vecteur de Stokes les plus élevées

Nous avons pu voir à la section précédente que la distribution des vitesses de rotation du vecteur de Stokes inférieures à 1,1 rad/ms suivait une courbe de Rayleigh. Cependant il est important d'ajuster la partie de l'histogramme correspondant aux valeurs de vitesse les plus élevées car c'est dans cette partie de la distribution que se situent les valeurs élevées de vitesse susceptibles de causer des pannes et ruptures de service. Suite à une analyse comparative de différents seuils pour l'ajustement exponentiel, le choix s'est porté sur une valeur de seuil de 2,0 rad/ms (Crawford et al., 2014) car celle-ci permettait d'obtenir l'ajustement le plus satisfaisant.

(Crawford et al., 2014) proposent d'ajuster la partie supérieure à 2,0 rad/ms de l'histogramme des vitesses de rotation du vecteur de Stokes à l'aide d'une fonction exponentielle donnée à l'équation (4.5). La partie droite de l'histogramme est définie comme étant l'ensemble des valeurs de vitesse rotation supérieures à 2,0 rad/ms.

$$f(x, \alpha, \beta) = \alpha \cdot e^{x \cdot \beta} \quad (4.5)$$

L'ajustement de la partie droite de la distribution des valeurs de vitesse de rotation avec une courbe exponentielle a été effectué en utilisant la fonction *fit* de Matlab avec un modèle exponentiel préexistant intitulé *exp1* et correspondant à l'équation ci-dessus. La fonction, qui a été configurée de manière à utiliser l'algorithme de Levenberg-Marquardt, demande deux paramètres d'entrée : un premier vecteur contenant les valeurs des abscisses à ajuster, qui correspondent à l'ensemble des intervalles de vitesse de rotation, et un second vecteur contenant les valeurs d'ordonnées à ajuster, à savoir les la distribution de probabilité normalisée correspondante. Le script permettant l'ajustement exponentiel, porte le nom de

fit_hist_arr et est présenté à l'ANNEXE XVII, page 215. Le traitement a été lancé pour l'ensemble des données de l'histogramme normalisé des vitesses de rotation du vecteur de Stokes supérieures à 2,0 rad/ms et le résultat est présenté à la Figure 4.7. Dans cette figure, les tracés en pointillés correspondent aux intervalles de confiance à 95 %.

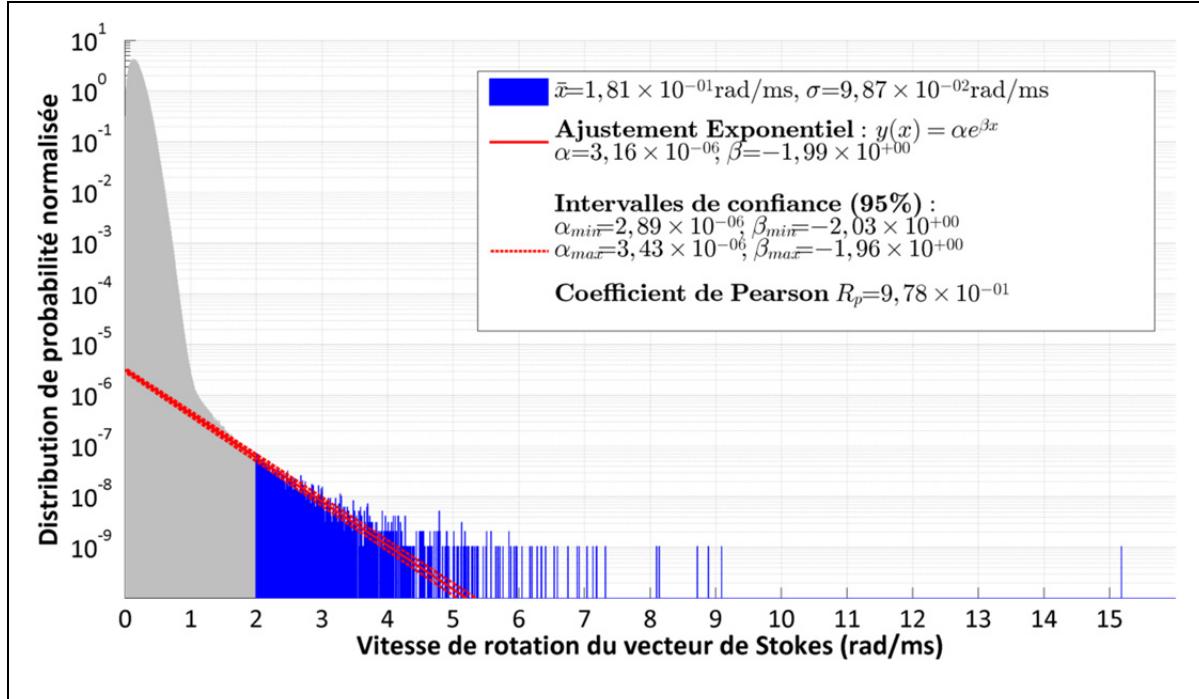


Figure 4.7 Ajustement de la partie droite de l'histogramme (valeurs supérieures à 2.0 rad/ms) des vitesses de rotation du vecteur de Stokes pour l'ensemble de la période de mesure (octobre 2012 à octobre 2014)

Ces résultats montrent que la partie droite de l'histogramme peut être ajustée à une courbe exponentielle de paramètres $\alpha = 3,16 \times 10^{-6}$ et $\beta = -1,99$. Les valeurs limites de paramètres pour l'intervalle confiance à 95% sont : $(\alpha_{min}, \beta_{min}) = (2,89 \times 10^{-6}, -2,03)$ et $(\alpha_{max}, \beta_{max}) = (3,43 \times 10^{-6}, -1,96)$, où $(\alpha_{min}, \alpha_{max})$ désignent respectivement l'estimation par défaut et par excès du paramètre α et $(\beta_{min}, \beta_{max})$ désignent respectivement l'estimation par défaut et par excès du paramètre β . Le coefficient de Pearson obtenu pour cet ajustement est $R_p = 0,9784$.

4.4 Analyses complémentaires des états de polarisation

L'analyse précédente a permis de mettre en évidence deux grandes catégories d'effets de polarisation différant par leur vitesse de rotation du vecteur de Stokes associée. A ce propos, on peut distinguer deux types de vitesse de rotation: les vitesses que l'on pourrait qualifier de lentes correspondant à des valeurs inférieures à 1,1 rad/ms, et les vitesses rapides, supérieures à 1,1 rad/ms.

Nous avons vu précédemment que les effets associés aux vitesses lente et modérée ont fait l'objet de plusieurs observations et études. Il y a cependant deux énigmes à éclaircir : d'une part la déviation de la distribution des vitesses inférieures à 1,0 rad/ms par rapport à une loi de Rayleigh et d'autre part la présence des effets rapides, caractérisés par une distribution de type exponentiel, qui n'a jusqu'ici fait l'objet d'aucune étude dans la littérature. Nous allons chercher ici à proposer des hypothèses quant à l'origine de ces phénomènes.

4.4.1 Analyse du bruit du polarimètre

Afin d'apporter une explication à l'écart de la distribution des vitesses de rotation du vecteur de Stokes inférieures à 1,1 rad/ms par rapport à une loi de Rayleigh, une investigation a été menée sur le bruit généré par le polarimètre et plus particulièrement sur une solution pour distinguer les fluctuations des états de polarisation et le bruit généré par l'instrument.

Afin de caractériser le bruit généré par le polarimètre Agilent N7786B, des mesures ont été effectuées dans le laboratoire de Ciena en reliant directement, via un court câble optique, le polarimètre à un transmetteur Ciena NGM à 10 Gbit/s, similaire à celui utilisé comme source polarisée du polarimètre dans la liaison aérienne. Des mesures ont ainsi été effectuées durant deux jours différents, à raison de 21 heures de mesures par jour. Ceci a permis de totaliser plus de 9×10^8 points de mesure du bruit généré par le polarimètre.

4.4.1.1 Distribution des vitesses de rotation du vecteur de Stokes induites par le bruit

Le cordon de raccordement optique reliant l'instrument à la carte étant très court, on peut considérer que les fluctuations de polarisation observées par le polarimètre correspondent au bruit généré par celui-ci. Ma contribution a porté sur la préparation d'un script Matlab permettant de traiter les données de bruit obtenues afin de caractériser le bruit généré par le polarimètre. Le traitement réalisé est le même que celui qui a été effectué sur les mesures de la liaison, avec pour résultat un histogramme des vitesses de rotation résultant uniquement du bruit du polarimètre, présenté à la Figure 4.8. \bar{x} et σ désignent respectivement la moyenne et l'écart type des valeurs de vitesses de rotation observées, et sont égaux respectivement à 0,1280 rad/ms et 0,0681 rad/ms.

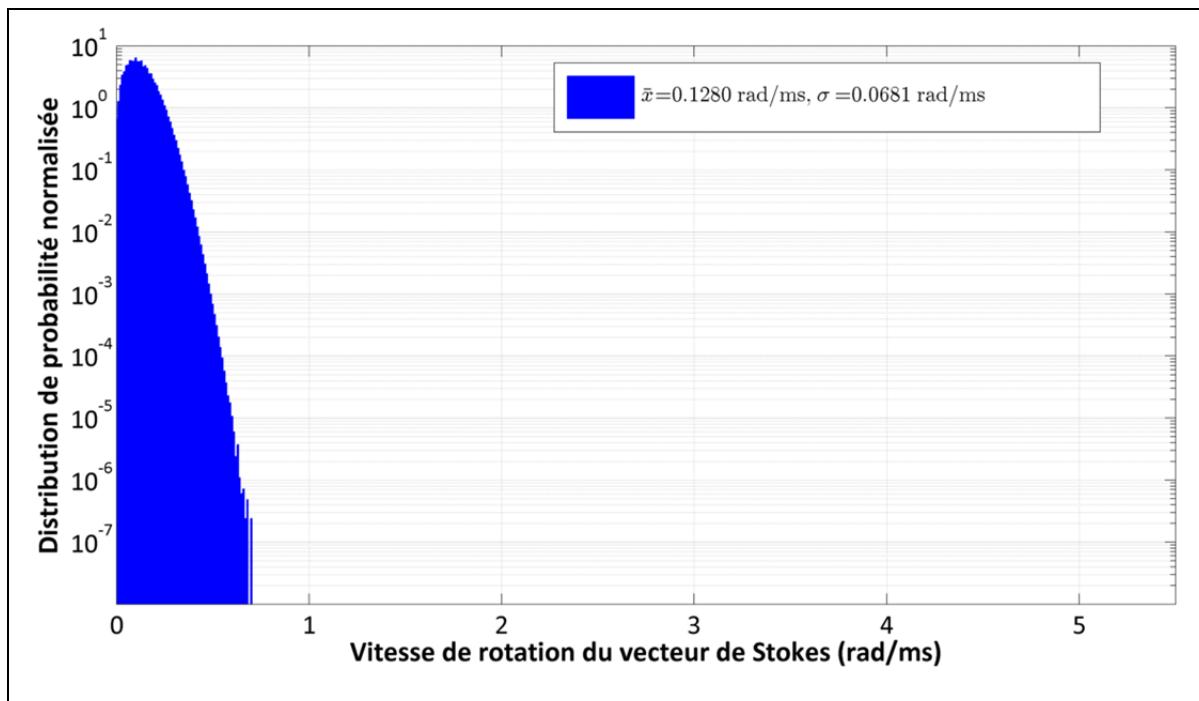


Figure 4.8 Histogramme des vitesses de rotation du vecteur de Stokes résultant du bruit généré par le polarimètre

On peut observer que le bruit généré par le polarimètre génère une distribution de vitesses de rotation avec une moyenne de 0,1280 rad/ms et un écart type de 0,0681 rad/ms. Ces résultats nous montrent que les erreurs systématiques induites par le polarimètre ne peuvent pas avoir

d'influence sur la partie de la distribution des vitesses de rotation contenant les valeurs supérieures à 2.0 rad/ms.

4.4.1.2 Caractérisation du bruit généré par le polarimètre

Le second volet de l'analyse du bruit du polarimètre a porté sur la caractérisation de la distribution du bruit généré dans les mesures. Pour cela, il a été nécessaire d'étudier la distribution des valeurs de la première composante de Stokes (S_1) dans les mesures dos à dos réalisées avec le polarimètre.

Un script Matlab, intitulé *characterization_noise_pola* et présenté à l'ANNEXE XVIII, page 223, a été développé afin d'obtenir l'histogramme des valeurs de S_1 . L'histogramme obtenu pour l'ensemble des mesures est présenté à la Figure 4.9.

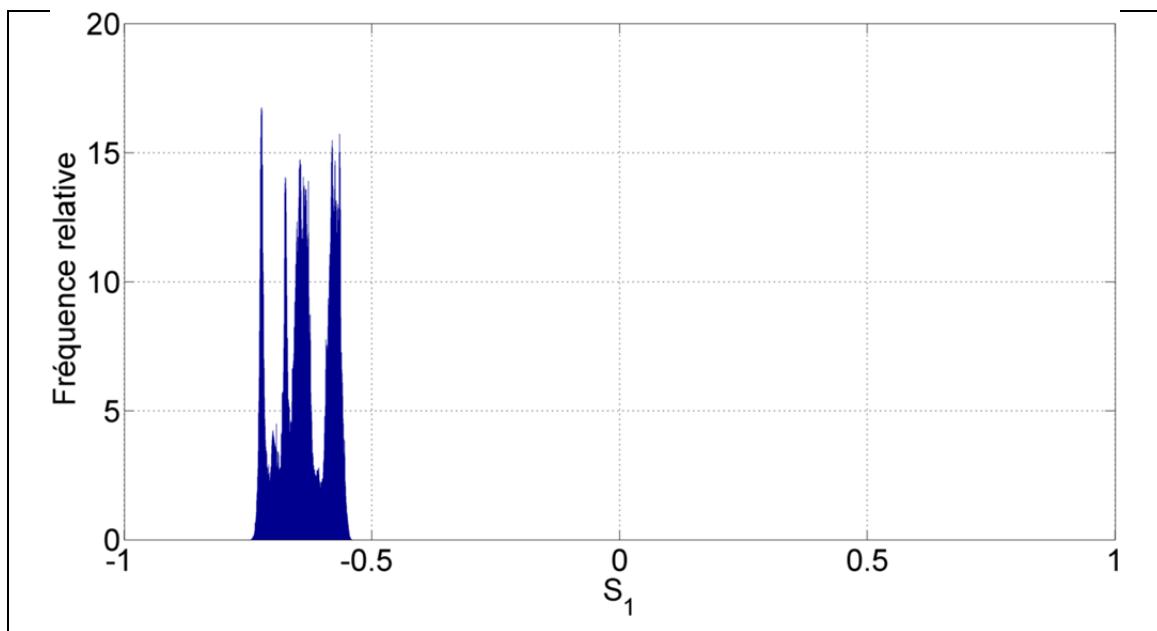


Figure 4.9 Histogramme des valeurs de S_1 observées lors des mesures dos à dos du polarimètre

On peut constater que la distribution des valeurs de S_1 se présente sous la forme de plusieurs Gaussiennes. Ce phénomène résulte d'une lente dérive temporelle des paramètres de Stokes

et peut potentiellement être responsable de la dérive observée dans l'histogramme des vitesses de rotation du vecteur de Stokes. Il est nécessaire de procéder à un filtrage passe-haut afin de s'affranchir de cet effet. La méthode suivie pour procéder au filtrage est présentée à la Figure 4.10, où FFT désigne l'algorithme de la transformée de Fourier rapide et IFFT l'algorithme de la transformée de Fourier inverse rapide. Le profil de filtre choisi est un filtre du type « mur de brique », c'est-à-dire que ce filtre passe à zéro l'ensemble des coefficients de Fourier correspondant à une fréquence inférieure à la fréquence de coupure et laisse intacts les coefficients correspondant aux fréquences supérieures.

Le script de traitement procède aux opérations présentées à la Figure 4.10 pour chaque fichier (un fichier de mesure représente 2×10^5 points de mesures, correspondant à 10 secondes) du polarimètre, et produit pour chacun de ces fichiers un histogramme des valeurs de S_1 . Le traitement consiste à calculer la transformée de Fourier discrète du signal donné en paramètre (matrice contenant les mesures du paramètre S_1), à appliquer un filtrage dans le domaine fréquentiel et enfin à récupérer la représentation temporelle du signal filtré en utilisant la transformée de Fourier inverse.

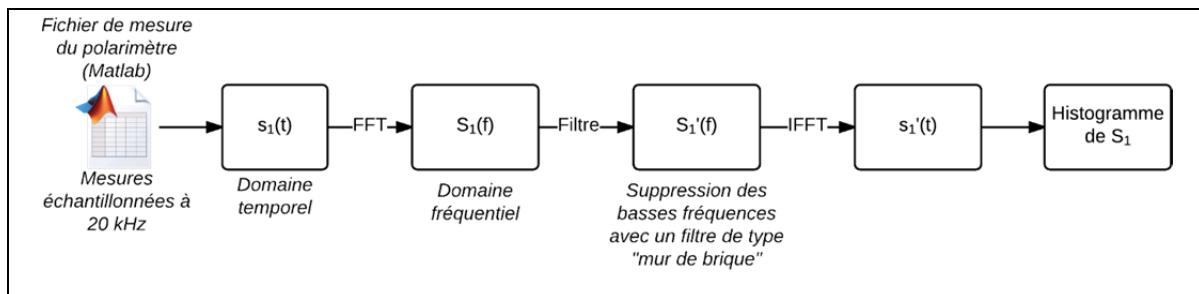


Figure 4.10 Méthode de filtrage des mesures dos-à-dos du polarimètre

L'ensemble des histogrammes obtenus est ensuite fusionné pour obtenir un histogramme global correspondant à l'ensemble de la période étudiée. L'histogramme obtenu pour le premier jour de mesure et une valeur de fréquence de coupure de 1 Hz est présenté à la Figure 4.11. Afin de déterminer la distribution à laquelle correspondent les valeurs de S_1 , un ajustement statistique à une loi normale de moyenne μ et d'écart type σ a été réalisé. La densité de probabilité de la loi normale est présentée à l'équation (4.6),

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (4.6)$$

où μ et σ désignent respectivement la moyenne et l'écart type de la loi normale.

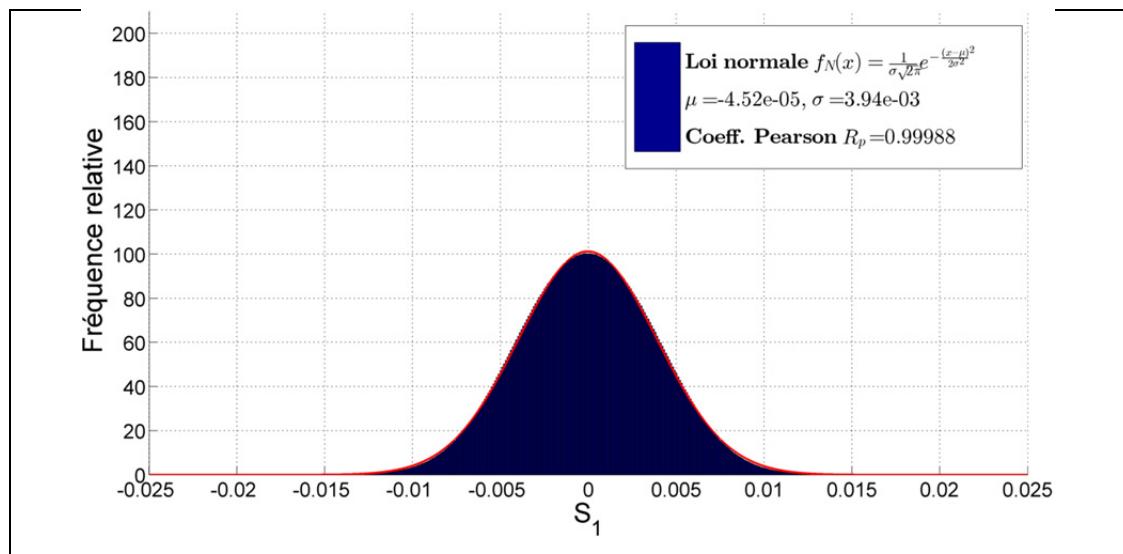


Figure 4.11 Distribution des valeurs de S_1 résultant du bruit généré par le polarimètre

On peut observer que l'ensemble des valeurs filtrées de S_1 correspond à une distribution gaussienne de moyenne approximativement nulle et d'écart type 0,00394. L'ajustement statistique a un coefficient de Pearson $R_p = 0,99988$. On en déduit que le polarimètre produit un bruit de type Gaussien.

4.4.1.3 Simulation du bruit du polarimètre

Une seconde étude liée au polarimètre a été menée dans une tentative d'apporter une piste d'explication à la déviation que l'on peut observer entre la distribution des vitesses de rotation du vecteur de Stokes et une distribution de Rayleigh. Cette étude consiste à étudier l'influence de certains phénomènes physiques tels que le bruit et le degré de polarisation du signal (*Degree of Polarization*, DOP).

Ma contribution a porté sur la modélisation du bruit du polarimètre au moyen d'un script Matlab. L'étape la plus importante a consisté à générer un fichier de polarimètre. Pour cela, (Crawford et al., 2014) ont proposé de modéliser le polarimètre en générant des mesures synthétiques (générées aléatoirement) sous forme de plusieurs fichiers. Les étapes de génération pour un fichier sont présentées à la Figure 4.12. Dans cette figure, μ_{bruit} et σ_{bruit} désignent respectivement la moyenne et l'écart type du bruit Gaussien généré par le polarimètre, $DOP_{fichier}$ désigne le degré de polarisation du fichier et $randn$ désigne la fonction Matlab générant une variable aléatoire normalement répartie.

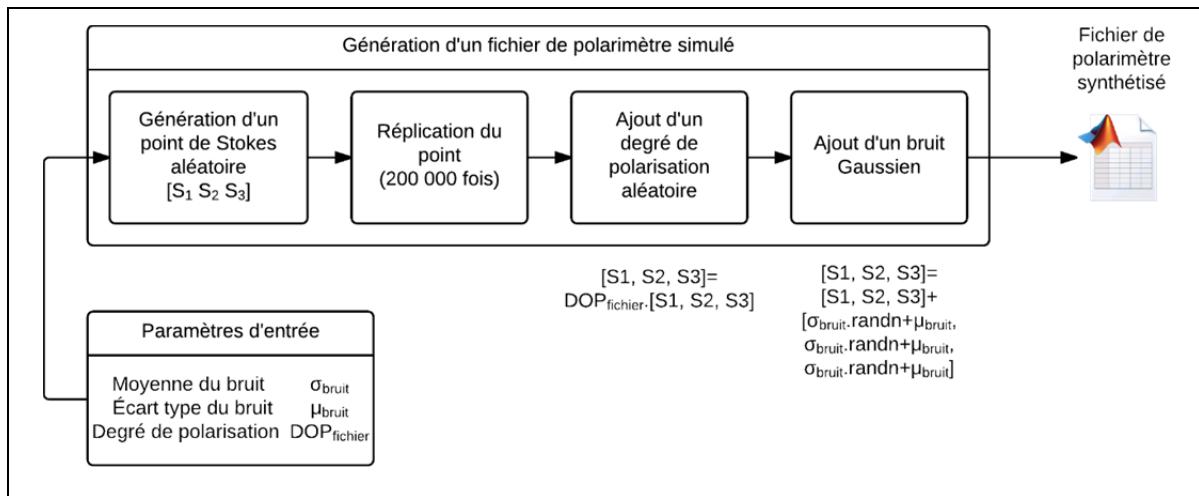


Figure 4.12 Étapes de la génération d'un fichier de polarimètre

Chaque fichier est généré en choisissant un point aléatoire sur la sphère de Poincaré, en répliquant ce point 200 000 fois (taille de fichier de polarimètre utilisée dans le projet), en appliquant un degré de polarisation aléatoire pour chaque fichier, situé entre une limite donnée (valeur minimale fixée à chaque simulation) et 1. Un bruit Gaussien, dont les caractéristiques ont été déterminées à la section 4.4.1.2, est ensuite ajouté aux 2×10^5 points.

Les deux étapes qui ont posé un problème d'ordre pratique sont la génération d'un point de Stokes aléatoire (correspondant par définition à un état de polarisation aléatoire) et la génération d'un bruit Gaussien.

La sélection d'un point aléatoire sur la sphère de Poincaré ne doit pas se faire en générant aléatoirement les deux coordonnées sphériques conventionnelles, car cette approche a tendance à regrouper les résultats vers les pôles de la sphère d'après (Weisstein, 2015). Il est alors possible de tirer les trois paramètres de Stokes S_1 , S_2 et S_3 par l'équation (4.7) ci-dessous.

$$\begin{aligned} S_1 &= \sqrt{1 - u^2} \cos \theta \\ S_2 &= \sqrt{1 - u^2} \sin \theta \\ S_3 &= u \end{aligned} \tag{4.7}$$

où u est un paramètre aléatoire situé entre 0 et 1 et θ un autre paramètre aléatoire situé entre 0 et 2π .

Afin de vérifier le caractère uniforme des points générés par la méthode proposée par (Weisstein, 2015), un script Matlab a été préparé sous le nom de *generate_random_sphere_point*, présenté à l'ANNEXE XIX, page 231. Le script génère aléatoirement 5000 points et les affiche sur la sphère de Poincaré. La sélection des deux variables aléatoires s'est faite à l'aide de la fonction *rand* de Matlab qui génère un nombre aléatoire uniformément réparti entre 0 et 1. Les résultats sont présentés à la Figure 4.13 : on peut observer que la distribution obtenue est uniforme.

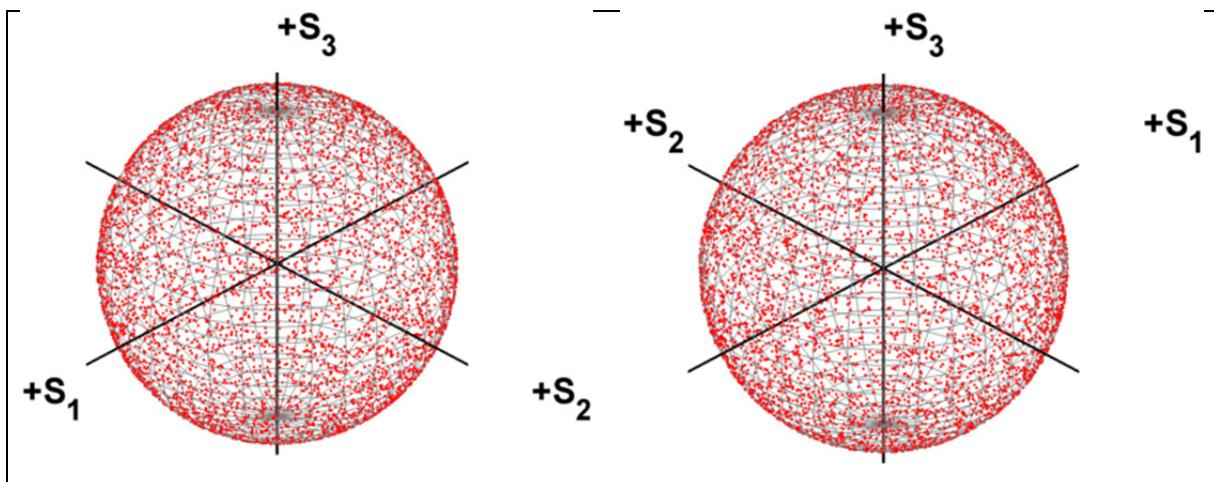


Figure 4.13 Distribution de 5000 états de polarisation aléatoires en utilisant la méthode proposée par Weisstein (2015)

La génération d'un bruit Gaussien s'est réalisée à travers la fonction Matlab `randn`. Cette fonction génère un nombre aléatoire tiré d'une distribution de la loi normale. Il est alors possible d'ajuster ce nombre à une moyenne et un écart type donné en utilisant l'équation (4.8) tirée de la documentation de Matlab :

$$N = \sigma \cdot \text{randn} + \mu \quad (4.8)$$

où N désigne une variable aléatoire normalement répartie de moyenne μ et d'écart type σ . Les valeurs de moyenne et d'écart type utilisées dans le cadre de la simulation sont celles qui ont été obtenues à la partie 4.4.1.2. Le script Matlab simulant le bruit du polarimètre porte le nom de `pola_noise_simulator` et est présenté à l'ANNEXE XX, page 233.

Ainsi définies, les simulations sont programmées pour générer un total de 15 000 fichiers contenant chacun 200 000 mesures. Afin de limiter l'espace mémoire requis pour le calcul, les fichiers sont générés en mémoire vive, traités et directement supprimés puisque la seule information qui nous intéresse est l'histogramme des valeurs de vitesse de rotation. Deux simulations ont été lancées avec deux valeurs de degré de polarisation, $DOP = 1$ et $0.8 < DOP < 1$, dans ce dernier cas une valeur est choisie pour chaque fichier généré. Les

histogrammes obtenus par ces deux simulations, accompagnés d'un ajustement de Rayleigh, sont présentés à la Figure 4.14 et à la Figure 4.15.

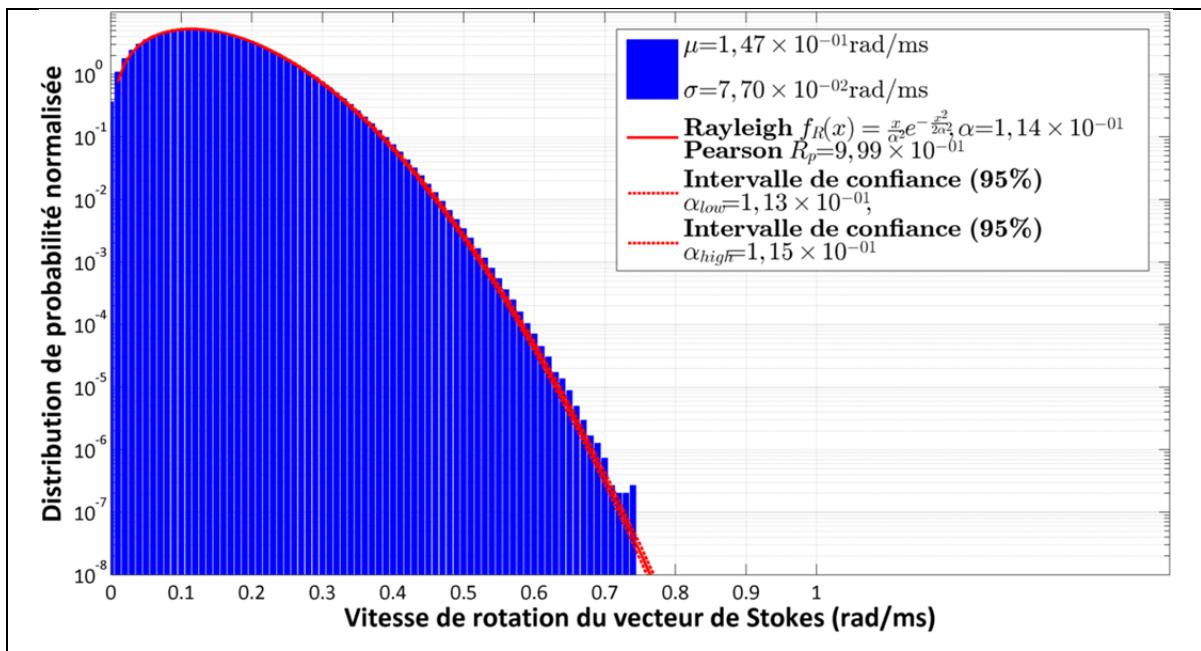


Figure 4.14 Histogramme des vitesses de rotation du vecteur de Stokes obtenues par simulation du bruit du polarimètre (DOP = 1)

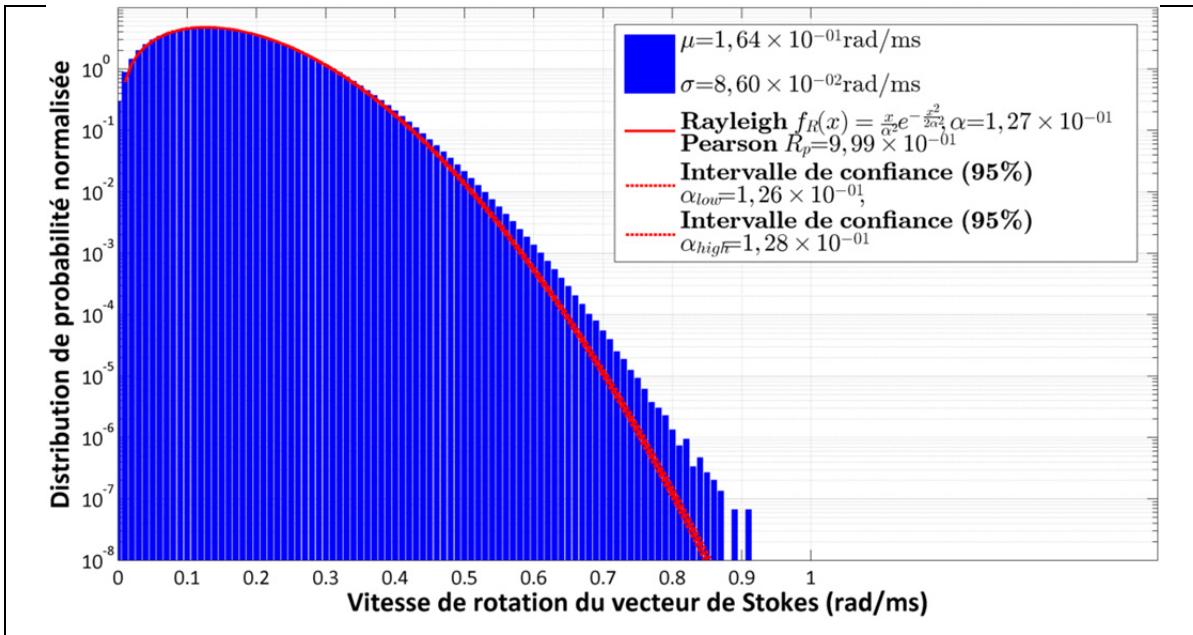


Figure 4.15 Histogramme des vitesses de rotation du vecteur de Stokes obtenues par simulation du bruit du polarimètre ($0.8 < \text{DOP} < 1$)

Les résultats obtenus pour $\text{DOP} = 1$, on observe que l'histogramme normalisé s'ajuste bien à une distribution de Rayleigh. En revanche, lorsque le degré de polarisation obtenu est défini de manière aléatoire entre 0,8 et 1 (une valeur générée par fichier), on observe une déviation par rapport à une distribution de Rayleigh. Bien que la valeur des coefficients de Pearson pour les deux ajustements soit comparable, on observe cependant visuellement une déviation plus importante dans le cas d'un degré de dépolarisation aléatoire. Le résultat obtenu dans cette partie propose une source possible de la déviation observée par rapport à une distribution de Rayleigh dans les données mesurées par le polarimètre, puisque les mesures du lien aérien n'ont pas un degré de polarisation constant.

4.4.2 Analyse harmonique des mesures du polarimètre

La seconde analyse complémentaire des mesures du polarimètre a porté sur l'analyse harmonique des mesures du polarimètre. Cette analyse s'est effectuée en deux parties : la première a porté sur l'étude de l'évolution temporelle des composantes de Stokes sur une

longue période, de l'ordre de l'heure. La seconde a porté sur l'analyse du spectre des mesures du polarimètre.

Afin d'illustrer la rapidité de l'activité de polarisation dans la liaison aérienne sur notre liaison, l'évolution temporelle du vecteur de Stokes est présentée sur la sphère de Poincaré à la Figure 4.16.

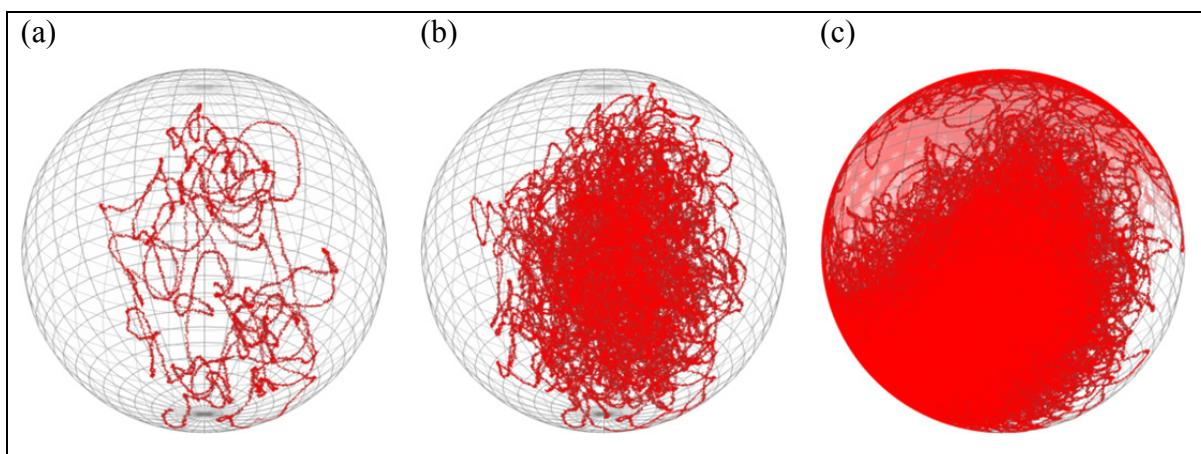


Figure 4.16 Représentation de l'évolution temporelle de l'état de polarisation du signal sur la sphère de Poincaré sur une durée
 (a) d'une seconde, (b) de 10 secondes, (c) de 100 secondes

Une autre étude complémentaire a porté sur l'oscillation des composantes du vecteur de Stokes sur des périodes allant de 10 à 15 minutes. Ma contribution a porté sur la recherche de caractéristiques particulières dans les mesures recueillies par le polarimètre. L'utilisation de la transformée de Fourier afin d'identifier ces oscillations de période supérieure à 10 secondes n'a pas été possible, car les mesures du polarimètre ne se sont faites que par période de 10 secondes au plus. Il n'était donc pas possible d'identifier des harmoniques inférieurs à 0.1 Hz.

L'analyse a donc porté sur le tracé de la représentation temporelle des trois composantes de Stokes à partir des mesures brutes du polarimètre, présentée à la Figure 4.17 et réalisée à travers le script Matlab *plot_SOP_realtime* et est présenté à l'ANNEXE XXI, page 239.

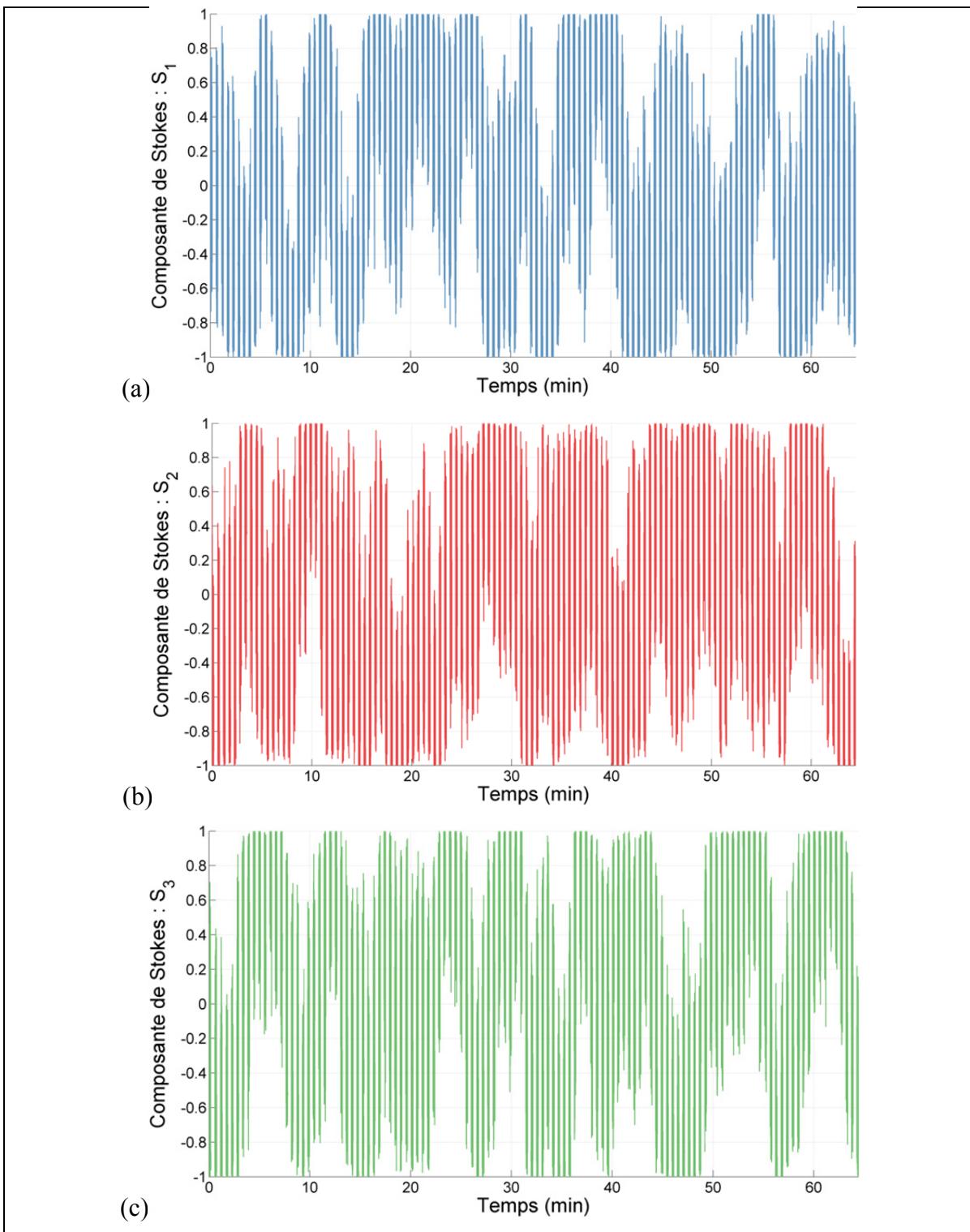


Figure 4.17 Évolution temporelle des trois composantes de Stokes mesurées à partir des données brutes du polarimètre (5 novembre 2012, de 14:36:44 à 15:41:10)

On peut ainsi observer de multiples oscillations irrégulières des composantes de Stokes dont la période semble être de l'ordre de quelques minutes. Ce phénomène cyclique peut résulter des conditions environnementales et notamment du vent, qui est susceptible d'affecter la quasi-totalité de la liaison aérienne étudiée dans notre projet. Ce phénomène d'oscillation avait déjà été observé par (Roberge, Audet et Bernier, 2009).

Des études similaires ont déjà été menées par (Roberge, Audet et Bernier, 2009), qui se sont intéressé aux effets du vent sur la polarisation du signal optique transmis dans une liaison aérienne de 70 km. Ils distinguent deux grands types de comportement en termes de variation d'état de polarisation : un état de vent modéré durant lequel l'état de polarisation subit une lente dérive et un état de vent élevé durant lequel l'état de polarisation varie de manière assez rapide.

Dans le cadre de notre projet, nous ne disposons pas d'information significative en ce qui concerne l'évolution de la vitesse du vent pendant les phases de mesure. Cependant, notre liaison étant aérienne, il est pertinent de comparer l'évolution de la polarisation du signal comparativement aux observations réalisées par (Roberge, Audet et Bernier, 2009). Ces derniers ont effectué des mesures de la vitesse de rotation du vecteur de Stokes avec une résolution temporelle de 10 millisecondes : la grande majorité les valeurs obtenues s'échelonne entre 0 rad/s et 8 rad/s avec quelques rares vitesses de rotation élevées allant de 17,5 rad/s à 23,5 rad/s. La plus grande valeur de vitesse de rotation du vecteur de Stokes se ramène ainsi en valeur moyenne à 0,024 rad/ms. On peut constater que ces valeurs sont largement inférieures à celles observées dans notre projet et présentés au Tableau 4.1. Cette différence peut s'expliquer par la différence d'ordre de grandeur de la fréquence d'acquisition qui est de 20 kHz dans notre projet et de 100 Hz dans les travaux de (Roberge, Audet et Bernier, 2009).

La liaison aérienne étudiée dans ce projet étant de type OPGW, il semble pertinent d'étudier la densité spectrale de puissance d'une composante de Stokes mesurée par le polarimètre afin d'identifier d'éventuels harmoniques à 60 Hz correspondant à la fréquence du courant

véhiculé par les liaisons à haute tension. Le calcul de la densité de puissance de la composante S_1 a été effectué sur un fichier de mesure du polarimètre datant du 14 juillet 2014 et qui représente 10 secondes de mesure de S_1 avec un taux d'échantillonnage de 20 kHz (Figure 4.18). Le calcul de la densité de puissance s'est réalisé à l'aide de l'algorithme de la transformée de Fourier rapide à travers la fonction *fft* de Matlab. Le script Matlab permettant de tracer la densité spectrale de puissance d'un fichier de mesure du polarimètre, *plot_harmonics_pola*, est présenté à l'ANNEXE XXII page 243.

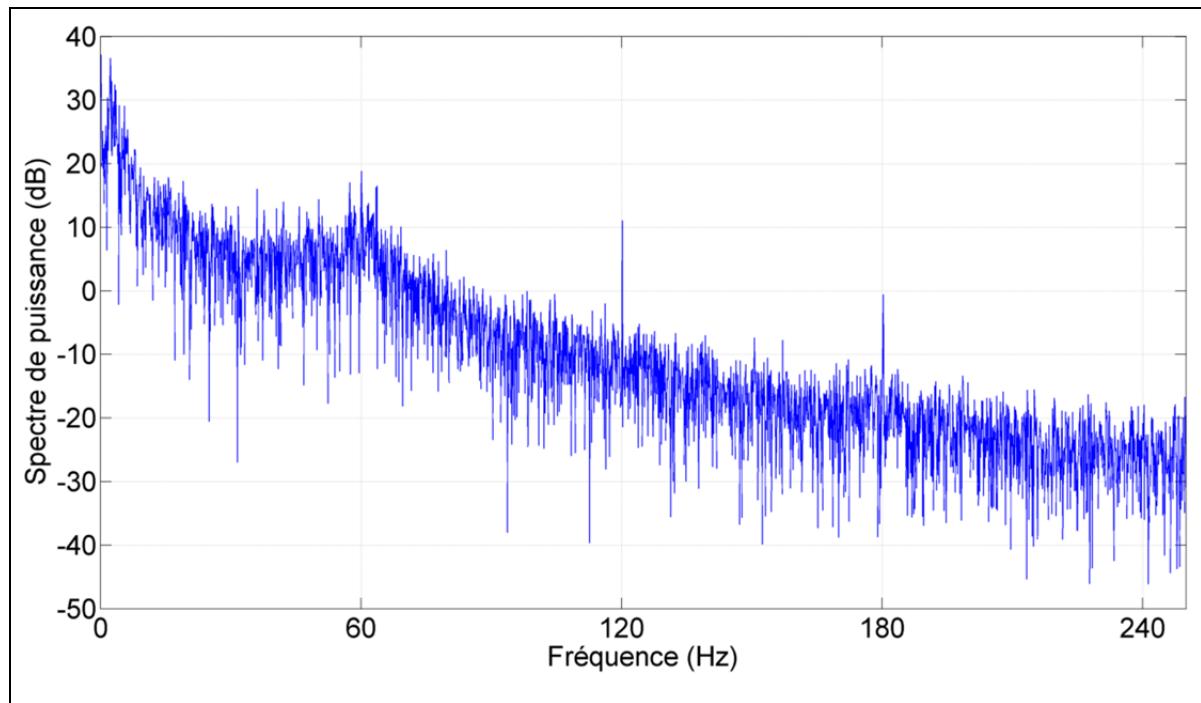


Figure 4.18 Spectre de puissance de la première composante de Stokes pour une durée de mesure de 10 secondes

L'analyse spectrale des mesures du polarimètre permet de mettre en évidence une composante spectrale importante à 60 Hz ainsi que la présence des harmoniques suivants à 120 Hz et 180 Hz.

Des observations similaires ont été réalisées par (Leeson, Bao et Côté, 2009), qui ont effectué des mesures sur une liaison aérienne de type OPGW comparable à celle étudiée dans ce projet en mettant l'emphase de l'influence des courants importants transportés sur les

pylônes électriques sur l'état de polarisation du signal optique. En effet, bien que semblant appartenir à deux domaines physiques distincts en première approche, le courant électrique véhiculé par les conducteurs électriques génère un champ magnétique qui peut moduler l'état de polarisation du signal optique circulant dans la fibre par effet Faraday. L'effet Faraday est un effet magnéto-optique par lequel un champ magnétique peut influencer l'état de polarisation de la lumière en lui imposant une rotation proportionnelle à ce champ magnétique (Bruhat, Kastler et Bouchareine, 2005).

D'autres travaux portant sur l'étude de l'oscillation des composantes de Stokes ont été menés sur des fibres optiques aériennes de type OPGW et d'une longueur de l'ordre de 20 km (Ednay, 2009). Les observations alors réalisées permettent de constater que les périodes d'oscillation les plus significatives sont respectivement de 2,5 s, 400 ms et 20 ms (fréquence équivalente : 50 Hz). Les oscillations observées par (Ednay, 2009) ont une période qui est d'une manière générale plus faible que celle des oscillations observées dans la liaison aérienne étudiée dans ce mémoire.

4.5 Analyse des effets rapides de polarisation

Nous allons détailler ici la méthode utilisée pour l'analyse des événements de polarisation rapides. Ces événements sont définis à la section 2.3.2.4. Ma contribution a porté sur l'amélioration de la méthode développée par (Mortelette, 2014) en mettant en place une méthode de calcul parallèle. Ceci a permis de produire une fonction Matlab intitulée *pola_compute_events* présenté à l'ANNEXE VII, page 173, assurant un temps de calcul raisonnable. Cette fonction, dont le mode de fonctionnement est présenté à la section 2.3.2.4, prend en paramètre d'entrée la matrice contenant les paramètres de Stokes pour un fichier, la matrice des vitesses de rotation du vecteur de Stokes et génère une structure de données contenant tous les événements de polarisation dont la vitesse de rotation mesurée est supérieure au seuil de 0.9 rad/ms.

La structure de données retournée par la fonction *pola_compute_events* est présentée au Tableau 4.2. Chaque ligne de la structure ainsi obtenue contient la valeur d'horodatage du

fichier contenant l'événement, les indices (relatifs aux matrices en entrée) correspondant au début et à la fin de l'événement, la vitesse de rotation angulaire du vecteur de Stokes maximale observée durant l'événement et les déflexions angulaires maximale et finale de l'événement. La base de données obtenue permet d'isoler, selon plusieurs critères tels que la déflexion angulaire ou la valeur de rotation du vecteur de Stokes maximale, les événements observés sur l'ensemble des mesures afin de procéder à des études ultérieures.

Tableau 4.2 Informations contenues dans la structure de donnée des événements rapides de polarisation

Horodatage du fichier contenant l'événement
Instant de début d'événement (indice de la ligne correspondante dans la matrice des vitesses de rotation du vecteur de Stokes)
Instant de fin d'événement (indice de la ligne correspondante dans la matrice des vitesses de rotation du vecteur de Stokes)
Vitesse de rotation du vecteur de Stokes maximale durant l'événement
Déflexion angulaire maximale observée pendant l'événement
Déflexion angulaire observée en fin d'événement

4.5.1 Étude de la distribution de la durée, de la déflexion angulaire et de l'amplitude des événements de polarisation

Une fois créée la structure de données, l'étape suivante a été de déterminer la durée et la nature élastique ou non élastique des fluctuations rapides des états de polarisation. La propriété d'élasticité d'un événement peut être établie en calculant la déflexion angulaire entre l'état de polarisation en début d'événement et l'état de polarisation en fin d'événement, tel que défini à la section 2.3.2.4.

Ma contribution a porté sur la création d'un script Matlab intitulé *events_distributions* et présenté à l'ANNEXE XXIII, page 245, affichant l'histogramme de la durée, la vitesse de rotation du vecteur de Stokes maximale, la déflexion angulaire maximale et la déflexion angulaire finale observés durant les 95 891 événements enregistrés d'octobre 2012 à octobre

2014. Le calcul de l'histogramme est réalisé à partir de la structure de données obtenue précédemment.

La Figure 4.19 présente la distribution de la durée de l'ensemble des événements de polarisation observés. \bar{x} et σ désignent respectivement la moyenne et l'écart type des valeurs de vitesses de rotation observées, et sont égaux respectivement à 0,5889 ms et 1,7048 ms.

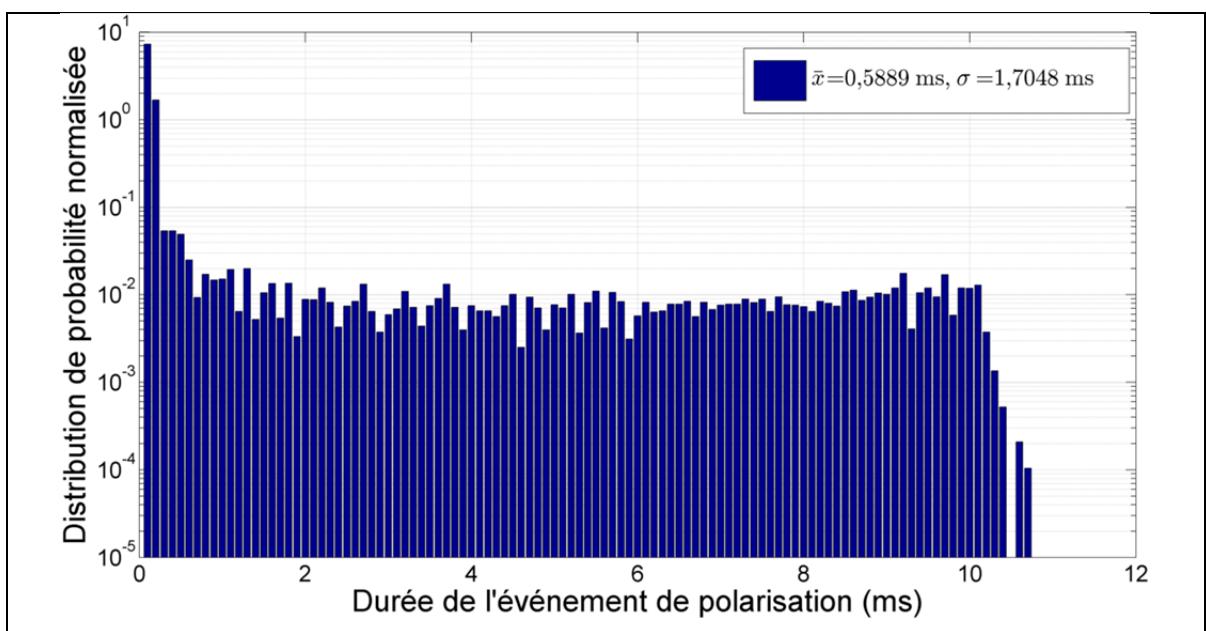


Figure 4.19 Distribution de la durée de l'ensemble des événements de polarisation observés sur la liaison aérienne

La plupart des effets rapides de polarisation observés sont courts, avec 86 % des effets transitoires qui durent moins de 0,25 ms et seulement 5% des événements qui dépassent 4,20 ms. La valeur maximale de durée observée est de 10,75 ms. Enfin, on peut remarquer que les événements d'une durée située entre 0,6 ms et 10,1 ms sont approximativement répartis uniformément.

La Figure 4.20 présente la distribution de la déflection angulaire calculée entre l'état de polarisation en début d'événement et l'état de polarisation en fin d'événement. On peut observer une valeur moyenne de 0,0543 rad ainsi qu'un écart type de 0,1010 rad.

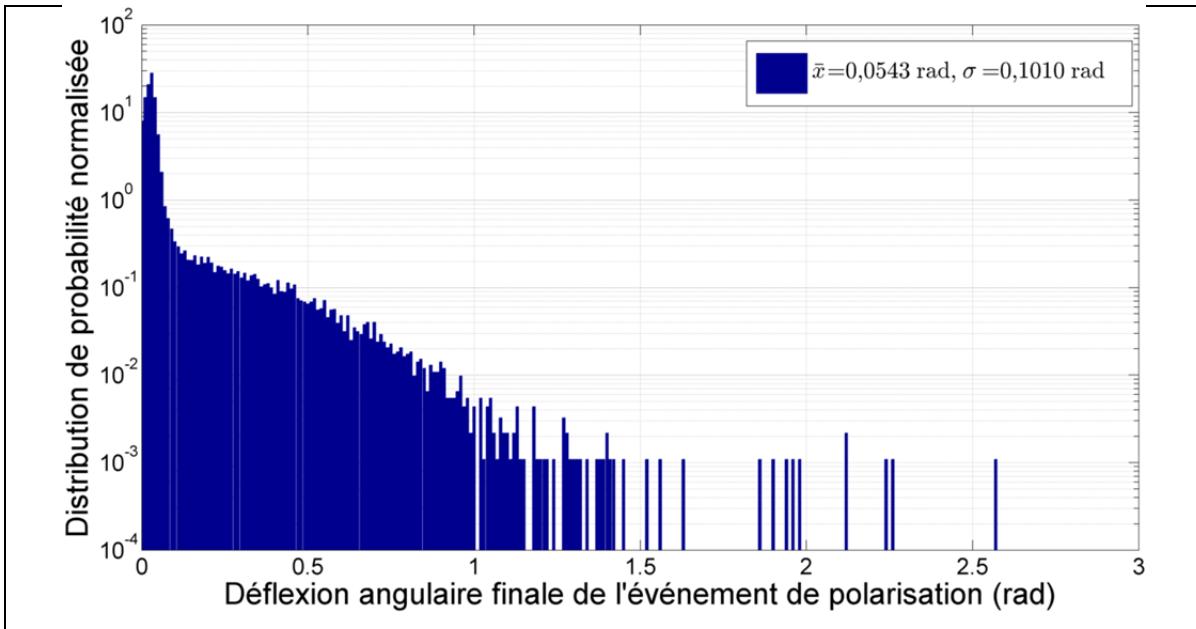


Figure 4.20 Distribution de la déflexion angulaire finale de l'ensemble des événements de polarisation observés sur la liaison aérienne

Sur l'ensemble des événements de polarisation, 90,8% ont une élasticité inférieure à 0,07 rad soit 8 degrés. Les événements dont la déflexion est inférieure ou égale à ce seuil peuvent être considérés comme des événements élastiques. Seuls 5,5% des effets rapides ont une valeur de déflexion dépassant 0,1745 rad soit 10 degrés, ce qui représente au total 5320 événements sur toute la période de mesure. On peut donc considérer que sur l'ensemble des 5320 événements mesurés, 90,8 % sont élastiques et 9,2 % sont inélastiques.

La Figure 4.21 présente la distribution de la vitesse de rotation du vecteur de Stokes maximale observée durant chaque événement de polarisation. On observe une valeur de vitesse de rotation moyenne de 1,0204 rad/ms et un écart type de 0,2897 rad/ms.

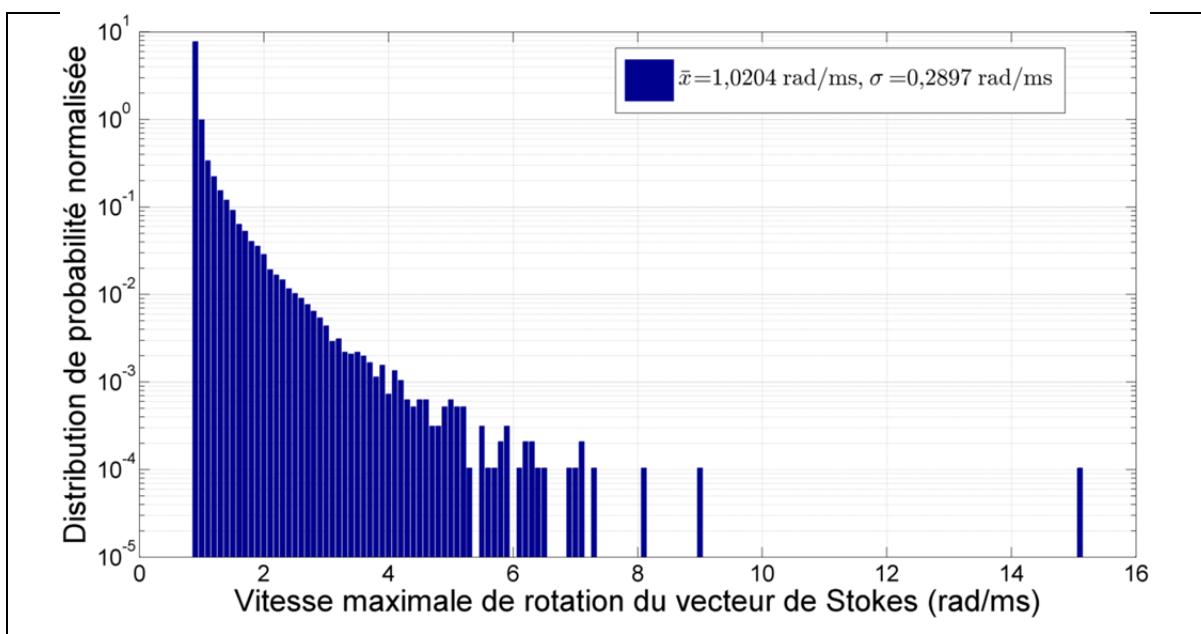


Figure 4.21 Distribution de la vitesse de rotation du vecteur de Stokes maximale de l'ensemble des événements de polarisation observés sur la liaison aérienne

La plupart des événements ont une vitesse de rotation maximale relativement faible : 90.5% des événements sont une vitesse maximale inférieure à 1,2 rad/ms. Seuls 1 % des événements sont une vitesse de rotation maximale supérieure à 2,3 rad/ms.

Une étude caractéristique des événements de polarisation a déjà été effectuée dans la littérature : (Nelson et al., 2011) ont étudié une liaison commerciale enfouie de 900 km à 40 Gbit/s sur une durée de 79 jours et ont enregistré ainsi plus de 500 événements. Le polarimètre est utilisé en mode déclenchement et le seuil de détection d'événement est fixé à une variation angulaire d'état de polarisation de 10 degrés sur une fenêtre temporelle de 100 ms, ce qui correspond en moyenne à une vitesse de rotation seuil de 0,0017 rad/ms. Un total de 515 événements a été enregistré avec seulement deux événements dépassant une vitesse de 0,1745 rad/ms. La vitesse de rotation du vecteur de Stokes de l'événement le plus rapide observé est de 78,5 rad/ms. Par ailleurs, les quatre événements les plus rapides sont approximativement élastiques.

En conclusion, l'activité de polarisation dans un lien enfoui apparaît comme étant beaucoup moins grande que dans une liaison aérienne.

4.5.2 Analyse des événements extrêmes (en vitesse et en durée)

L'ensemble des mesures du polarimètre a permis d'isoler deux événements de polarisation extrêmes, l'un par sa durée (la plus longue observée) et l'autre par sa vitesse la plus élevée.

4.5.2.1 Analyse de l'événement le plus rapide

L'effet de polarisation le plus rapide observé durant la période d'octobre 2012 à septembre 2014 à une vitesse de rotation du vecteur de Stokes de 15,19 rad/ms. Afin de caractériser cet événement particulier, la représentation temporelle de la vitesse de rotation ainsi que des composantes de Stokes durant l'événement sont présentés à la Figure 4.22.

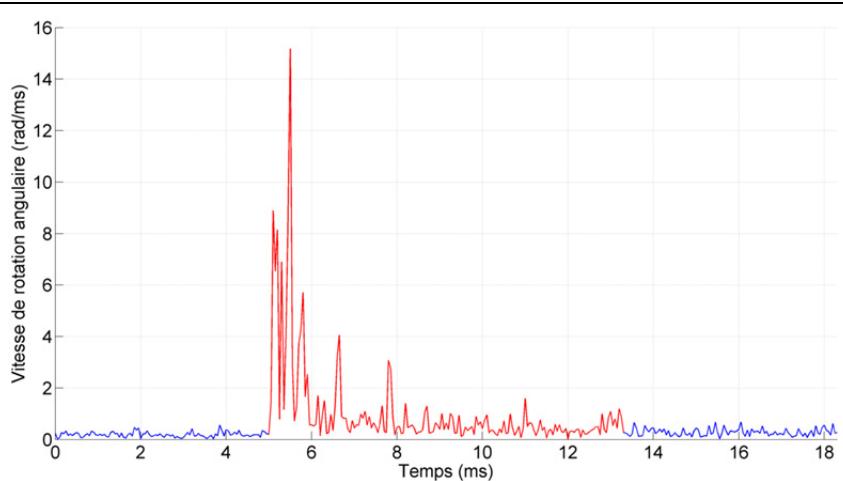


Figure 4.22 (a) Représentation de l'évolution temporelle de la vitesse de rotation du vecteur de Stokes durant l'effet de polarisation le plus rapide enregistré sur la liaison aérienne (5 décembre 2013)

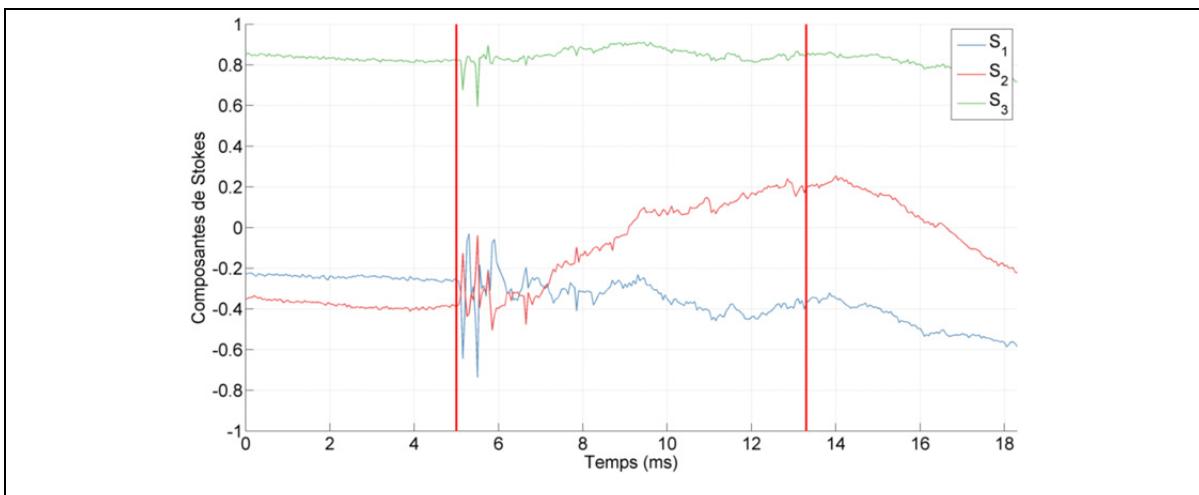


Figure 4.22 (b) Représentation de l'évolution temporelle des paramètres de Stokes durant l'effet de polarisation le plus rapide enregistré sur la liaison aérienne (5 décembre 2013)

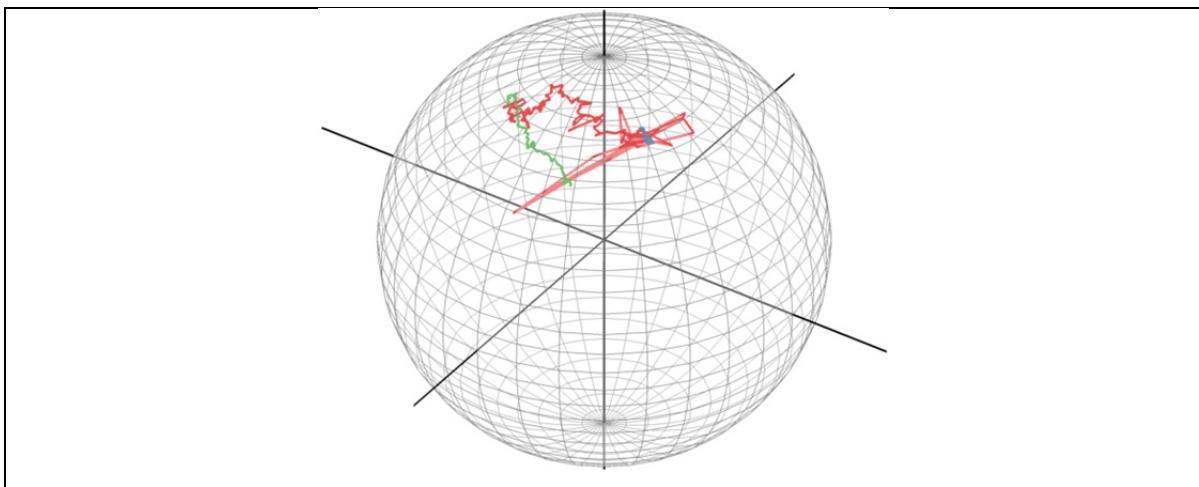


Figure 4.22 (c) Représentation de l'évolution temporelle du vecteur de Stokes sur la sphère de Poincaré durant l'effet de polarisation le plus rapide enregistré sur la liaison aérienne (5 décembre 2013)

On peut constater sur la Figure 4.22 (c) que l'événement le plus rapide est un événement inélastique. La durée de cet événement est de 8,3 ms. Ces résultats peuvent être comparés avec ceux obtenus par (Roberge, Audet et Bernier, 2009) sur une liaison aérienne de 70 km. La vitesse de rotation du vecteur de Stokes maximale alors obtenue est de 1346 degrés par seconde, soit 0,0235 rad/ms. Cette différence par rapport à nos observations peut s'expliquer

par le taux d'échantillonnage utilisé dans l'article, 100 Hz, qui est très faible comparativement à celle du polarimètre : 20 kHz.

4.5.2.2 Analyse de l'événement le plus long

L'effet de polarisation le plus long observé durant la période d'octobre 2012 à septembre 2014 à une durée de 10,7 ms et est présentés à la Figure 4.23. La durée proche de 10,7 ms résulte directement du fait que la fenêtre d'observation a été limitée à 10 ms dans la définition d'un événement. La valeur de durée légèrement supérieure à 10 ms peut s'expliquer par le fait qu'un second seuil de vitesse fixé à 0,45 rad/ms permet d'encadrer l'événement. Afin de caractériser cet événement, la représentation temporelle de la vitesse de rotation ainsi que des composantes de Stokes durant l'événement sont présentés à la Figure 4.23.

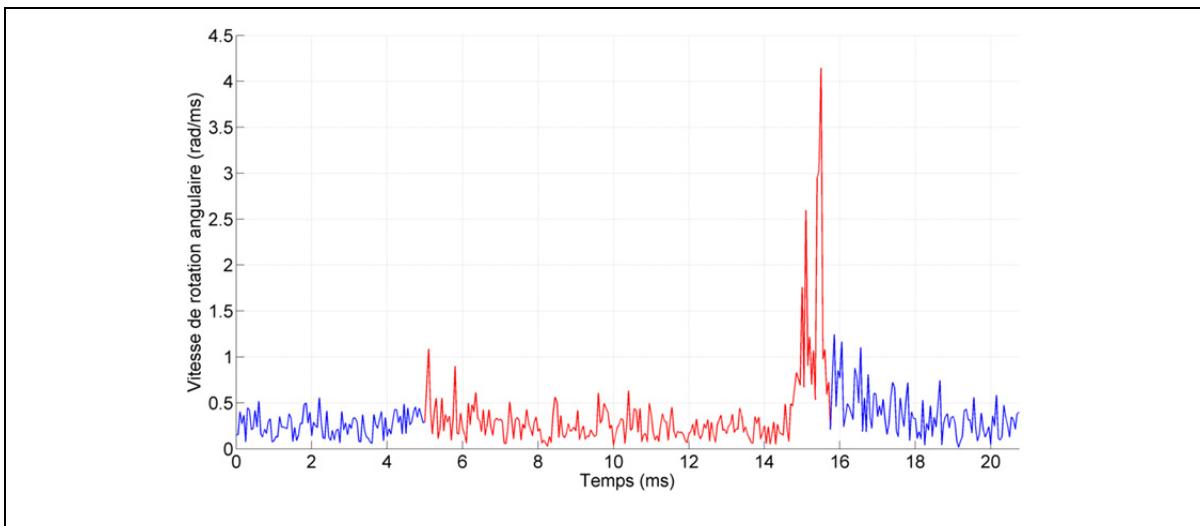


Figure 4.23 (a) Représentation de l'évolution temporelle de la vitesse de rotation du vecteur de Stokes durant l'effet de polarisation le plus long enregistré sur la liaison aérienne (14 juillet 2014)

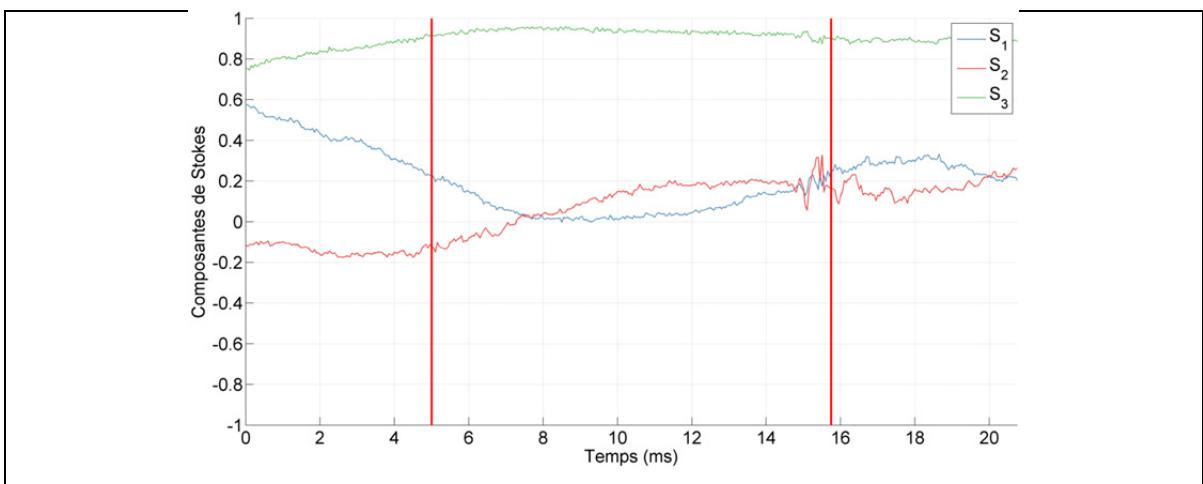


Figure 4.23 (b) Représentation de l'évolution temporelle des paramètres de Stokes durant l'effet de polarisation le plus long enregistré sur la liaison aérienne (14 juillet 2014)

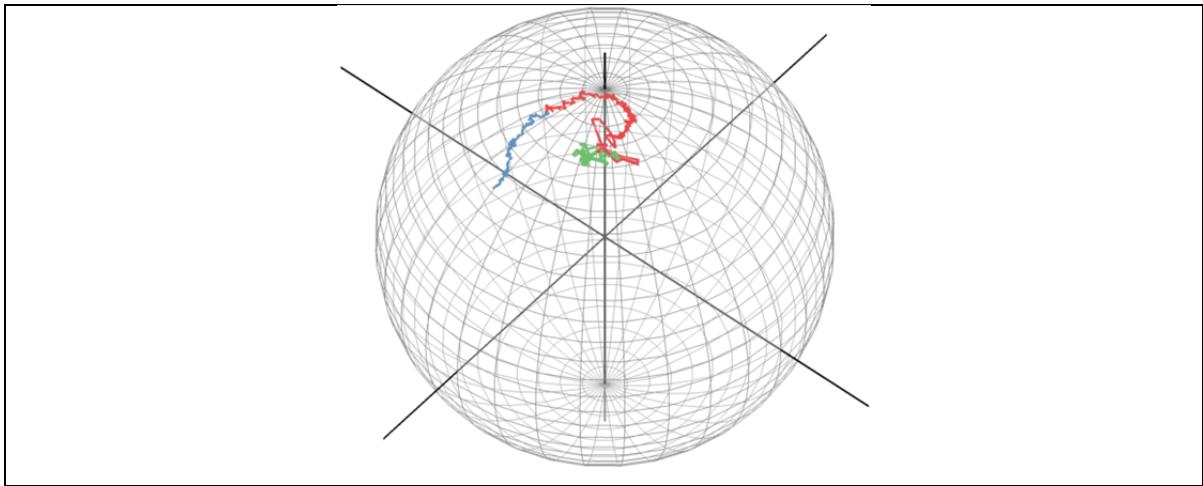


Figure 4.23 (c) Représentation de l'évolution temporelle du vecteur de Stokes sur la sphère de Poincaré durant l'effet de polarisation le plus long enregistré sur la liaison aérienne (14 juillet 2014)

La vitesse de rotation du vecteur de Stokes maximale associée à cet événement est de 4,15 rad/ms. On peut constater que cet événement est lui aussi inélastique. Par ailleurs, la Figure 4.23 (a) permet de constater que l'utilisation d'une fenêtre temporelle de 10 ms pour la définition des événements n'est pas toujours représentative de la réalité : la partie la plus significative de l'événement présenté semble se produire sur les deux dernières millisecondes de mesure dans les faits.

4.5.3 Corrélation entre les déflexions angulaires maximales et les vitesses de rotation du vecteur de Stokes maximales observées durant les effets de polarisation

Afin de mieux comprendre la corrélation entre les vitesses de rotation du vecteur de Stokes et la déflexion angulaire, il semble pertinent de tracer sur un même graphique les déflexions angulaires maximales et les vitesses de rotation maximales observées. À cette fin, (Mortelette, 2014) a représenté un graphique en 3 dimensions permettant voir les différents effets de polarisation mesurés en fonction d'une part de la vitesse de rotation du vecteur de Stokes maximale observée durant cet événement et d'autre part de la déflexion angulaire maximale observée durant l'événement.

Ma contribution a porté sur l'amélioration du script développé par (Mortelette, 2014), en utilisant notamment la fonction *hist3* faisant partie de la librairie logicielle *Statistics Toolbox* interne à Matlab. Cette fonction prend deux paramètres :

- le premier est une matrice à deux colonnes contenant les variables dont on souhaite calculer l'histogramme, en l'occurrence les vitesses de rotation du vecteur de Stokes maximales et les déflexions maximales événement par événement;
- le second est une structure de données contenant les frontières des barres de l'histogramme, variable par variable. La vitesse de seuil des événements rapide étant fixée à 0.9 rad/ms pour le projet, c'est cette valeur qui correspondra à la valeur minimale de l'axe des vitesses de rotation du vecteur de Stokes.

L'étape d'amélioration a permis de produire un nouveau script plus rapide : *hist_3d_pola*, présenté à l'ANNEXE XXIV, page 249, qui prend en paramètre la structure de données obtenue à l'issue du traitement initial des données du polarimètre, présentée à la section. Le script a été exécuté en utilisant la structure de données et le résultat est présenté à la Figure 4.24.

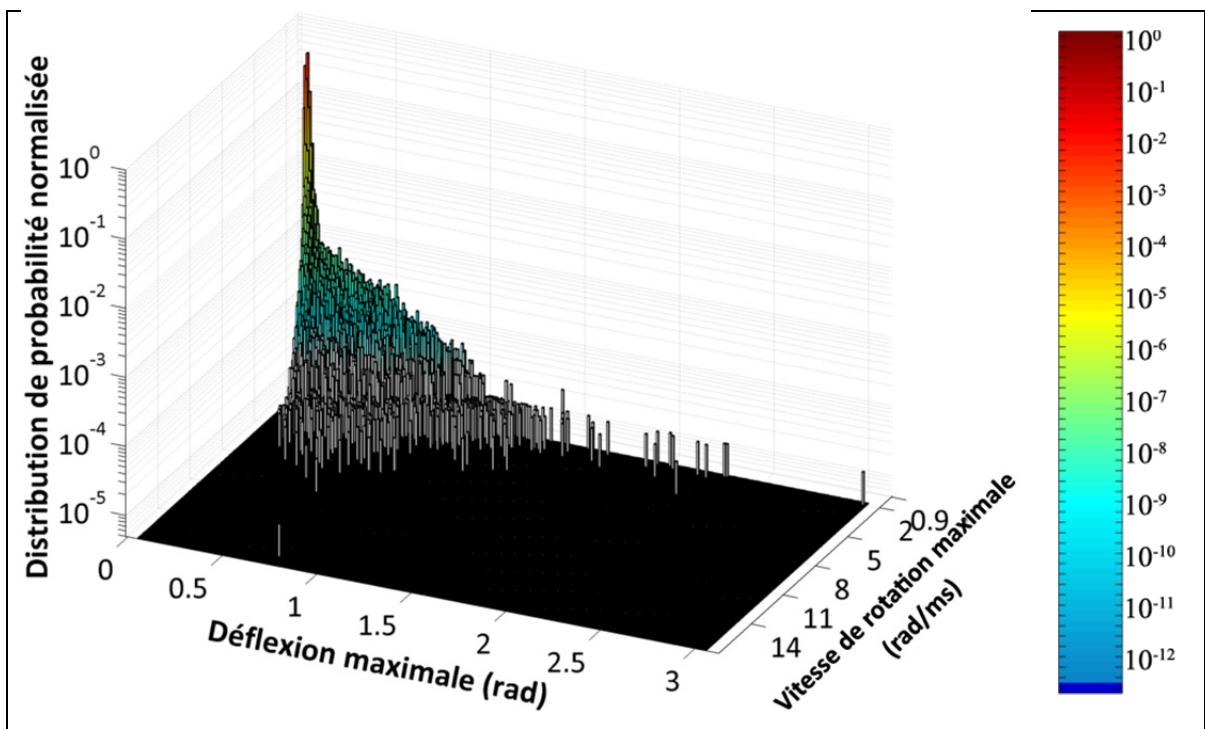


Figure 4.24 Histogramme des vitesses maximales de rotation du vecteur de Stokes et des déflexions maximales des effets transitoires de polarisation
(octobre 2012 à octobre 2014)

Les résultats obtenus permettent de constater que 97% des événements ont simultanément une vitesse de rotation du vecteur de Stokes inférieure à 2 rad/ms et à une valeur de déflexion maximale inférieure à 0,5 rad. En revanche, les effets rapides se situant au-delà de ces limites ne montrent pas une corrélation évidente entre la déflexion angulaire maximale et la vitesse de rotation du vecteur de Stokes maximale.

Comparativement aux résultats obtenus par (Mortelette, 2014) sur une période d'environ deux mois de mesures, on peut constater que les mesures sur de longues périodes ont permis d'aboutir à une plus grande diversité de cas présentés dans l'histogramme. En effet, les résultats observés par (Mortelette, 2014) correspondaient à une valeur maximale de vitesse de rotation du vecteur de Stokes de 9,09 rad/ms ainsi qu'à une valeur maximale de déflexion angulaire proche de 1,4 rad. Les valeurs obtenues dans notre projet sont respectivement de 15,19 rad/ms et de 2,99 rad.

(Nelson et al., 2011), sur la liaison commerciale de 900 km à 40 Gbit/s présentée à la section 4.5.1, ont suivi une méthode analogie en traçant sur un même graphique la vitesse de rotation du vecteur de Stokes maximale et la déflexion angulaire entre l'état de polarisation de début d'événement et l'état de polarisation en fin d'événement, avec un résultat similaire à celui observé dans la liaison aérienne : il n'y a pas de corrélation entre la déflexion angulaire maximale et la vitesse de rotation du vecteur de Stokes maximale.

Par ailleurs, l'étude simultanée de la vitesse de rotation du vecteur de Stokes et de la déflexion maximale, telle que présentée à la Figure 4.24, constitue une étape importante dans la conception d'un transceiver. En effet, la gestion par un module de traitement numérique du signal d'une déflexion angulaire élevée coïncidant avec une vitesse de rotation du vecteur de Stokes modérée pourrait mener à des caractéristiques différentes comparativement à la gestion d'une déflexion angulaire modérée se produisant dans un bref intervalle de temps (Crawford et al., 2014).

4.6 Conclusion du chapitre

L'analyse des données mesurées par le polarimètre sur la période 2012 - 2014 a permis d'améliorer la connaissance des effets de polarisation se produisant dans les liaisons optiques cohérentes aériennes. La présence d'un important volume de mesures a rendu nécessaire l'adaptation des outils existants afin de réduire les temps de calcul.

Les deux paramètres les plus pertinents qui ont permis de caractériser les événements rapides de polarisation sont la vitesse de rotation du vecteur de Stokes, illustrant la rapidité des changements d'état de polarisation, et la déflexion angulaire qui reflète l'amplitude de ces changements.

Concernant l'étude des vitesses de rotation du vecteur de Stokes, le nombre important de nouvelles données à a permis de compléter les travaux de (Mortelette, 2014). Par ailleurs, de nouveaux événements de polarisation, remarquables par leur durée ou leur vitesse de rotation

élevée, ont été enregistrés et ont permis, à travers plusieurs essais, de montrer que les valeurs de vitesse de rotation du vecteur de Stokes excédant 2 rad/ms semblent suivre une distribution exponentielle dont l'origine physique reste à expliquer.

Enfin, ce chapitre inclut une analyse du bruit généré par le polarimètre afin d'en dégager les caractéristiques. Il a été possible de simuler le fonctionnement du polarimètre via Matlab et de confirmer que le bruit généré par le polarimètre était de nature gaussienne et d'observer une lente dérive du niveau de signal qui pourrait être éliminé à l'avenir via l'application d'un filtre passe-haut sur les mesures. Des simulations ont montré par ailleurs qu'un certain degré de dépolarisation du signal pourrait expliquer du moins en partie, l'écart entre la distribution des valeurs de vecteur de rotation de Stokes inférieures à 1 rad/ms par rapport à une loi de Rayleigh. Cependant, faute de temps, cette méthode n'a pas pu être appliquée aux mesures réalisées sur le terrain.

CONCLUSION

Le monitoring d'une liaison optique cohérente aérienne, sur une période de deux ans, nous a permis d'améliorer la compréhension des phénomènes se produisant sur de telles liaisons, en portant l'emphasis sur les effets de polarisation qui jouent un rôle de plus en plus critique dans la conception des transpondeurs cohérents. L'expérience qui a été menée tire son origine du fait que nous avons eu l'opportunité de montrer une liaison commerciale aérienne sur une longue période à la fois avec un transpondeur cohérent et un polarimètre.

Ma contribution s'est articulée en premier lieu sur l'amélioration des outils de traitement développés précédemment par (Mortelette, 2014), ce qui a permis de diminuer de manière importante la vitesse de traitement des mesures des deux instruments. L'utilisation du calcul parallèle a permis de diviser les temps de traitement d'un facteur 2 pour les mesures du polarimètre à un facteur 10 pour les mesures du transpondeur cohérent. Le temps de traitement plus court a rendu possible de traiter l'ensemble des mesures du projet dans un temps raisonnable. Ma contribution a aussi porté sur le développement de nouveaux outils qui ont permis d'effectuer des analyses plus poussées des paramètres de performance qui sont liées au monitoring de liaison optique.

La caractérisation des différents paramètres de performance de la liaison mesurés par le transpondeur cohérent a ensuite été réalisée par le biais d'une analyse de l'évolution temporelle de ces paramètres sur plusieurs saisons. Cela a notamment permis de mettre en évidence un comportement cyclique, selon les périodes jour/nuit, des paramètres liés à la polarisation du signal tels que la DGD. Afin de mieux comprendre les variations cycliques des paramètres de performance liés à l'état de polarisation du signal, qui sont de l'ordre de 12 heures, une étude a été menée afin de montrer l'influence de la température sur l'activité de polarisation de la fibre : à cet effet, des données météorologiques ont été récupérées et il a été possible de tracer simultanément l'évolution temporelle de la température moyenne sur la liaison et l'évolution temporelle de l'activité de polarisation.

Une étude de l'évolution de la valeur moyenne à la minute de certains paramètres tels que la DGD ou l'activité de polarisation a été effectuée. Le recours à un moyennage à la minute a permis d'assurer un premier filtrage du bruit afin de pouvoir comparer les mesures du transpondeur cohérent avec celles du polarimètre. Une étude de la fonction d'autocorrélation des mesures moyennées a aussi permis d'établir les temps caractéristiques liés à la DGD et l'activité de polarisation de la liaison. Les résultats ainsi obtenus ont pu être comparés directement avec ceux de (Woodward et al., 2014) qui ont porté sur une liaison enfouie, il a ainsi été possible de constater que les temps caractéristiques de la liaison aérienne 50 à 100 fois moins importants que pour la liaison enfouie. Ce résultat est conforme à l'hypothèse selon laquelle les variations les plus rapides de l'état de polarisation se produisent vraisemblablement dans les liaisons aériennes. Le moyennage des paramètres de performance a aussi été utilisé afin de comparer les mesures réalisées par les deux instruments. Cette comparaison a été réalisée à travers une représentation temporelle des deux types de mesures et l'étude d'une fonction de corrélation croisée qui a permis de mettre en évidence une forte corrélation entre les mesures des deux instruments. Faute de temps, des analyses plus poussées n'ont pas été réalisées, mais ce résultat préliminaire démontre que l'activité de polarisation pourrait éventuellement être mesurée dans les liaisons commerciales au moyen de transpondeurs cohérents. Il pourrait être pertinent par la suite de vérifier si les observations réalisées sont généralisables à d'autres liens de type OPGW.

Une autre partie importante de ma contribution a porté sur la caractérisation des effets de polarisation se produisant sur la liaison à l'aide du polarimètre. Cette caractérisation a reposé dans un premier lieu sur l'étude de l'histogramme des vitesses de rotation du vecteur de Stokes, qui constituent un indicateur de référence pour étudier l'évolution de l'état de polarisation du signal lumineux dans une liaison optique. Le tracé de cet histogramme comprenant plus de 90 milliards de mesures a permis de constater que la distribution des vitesses de rotation suit une loi de Rayleigh pour les vitesses inférieures à 1,0 rad/ms, tandis que la distribution suit une loi exponentielle pour les vitesses de rotation les plus élevées. Cependant, dans les deux cas, on observe une certaine dérive de la distribution par rapport à une loi de Rayleigh pour les vitesses de rotation situées entre 1,0 et 2,0 rad/ms. Cette

nouvelle approche diffère de celle énoncée par (Mortelette, 2014) qui a proposé une distribution unique pour ajuster l'ensemble de l'histogramme.

L'étude des effets de polarisation s'est accompagnée d'autres analyses complémentaires portant dans un premier temps sur l'analyse du bruit du polarimètre, afin de chercher à expliquer la dérive de la distribution des vitesses de rotation par rapport à une loi de Rayleigh pour les vitesses de rotation du vecteur de Stokes inférieures à 1,0 rad/ms. L'analyse a permis de constater que le signal du polarimètre est caractérisé par un bruit gaussien dont la moyenne et l'écart type ont pu être déterminés, et est accompagné d'une lente dérive des paramètres de Stokes. Il a ensuite été possible de simuler le bruit généré par le polarimètre à l'aide de Matlab. Les résultats obtenus ont permis d'observer une dérive de la distribution des vitesses de rotation du vecteur de Stokes inférieures à 1,0 rad/ms par rapport à une loi de Rayleigh qui est similaire à celle observée dans les mesures.

La seconde analyse complémentaire a porté sur l'analyse harmonique des mesures du polarimètre : ces mesures ont été réalisées d'une part sur les mesures brutes du polarimètre afin de comparer les résultats de la liaison aérienne à ceux observés par (Ednay, 2009) sur des liaisons aériennes plus courtes et d'autre part sur les harmoniques à 60 Hz observés dans les mesures, afin de mettre en évidence l'effet Faraday lié aux lignes à haute tension mesuré par (Bao, Waddy et Chen, 2004). Les différents comportements harmoniques observés sont alors comparables à ceux présentés dans la littérature.

Enfin, la dernière analyse a porté sur la caractérisation des événements rapides de polarisation observés sur la liaison durant les deux années de mesure du polarimètre. Cette analyse a commencé par la caractérisation des événements de polarisation en termes de durée, de déflection angulaire et d'amplitude à l'aide d'histogrammes. Cela a permis de classifier les différents types d'événements observés afin de les comparer aux travaux précédents, notamment ceux de (Nelson et al., 2011). L'analyse s'est conclue par l'étude de la corrélation entre les déflections angulaires maximales et les vitesses de rotation du vecteur de

Stokes maximales observées durant les événements de polarisation. Nous avons pu constater qu'il n'y a pas de corrélation évidente entre ces deux paramètres.

RECOMMANDATIONS

Les outils d'analyse développés et améliorés dans le cadre de ce projet constituent une base solide pour toute étude ultérieure relative au monitoring de liaisons optiques à l'aide d'un transpondeur cohérent et d'un polarimètre. En plus d'être optimisés pour les architectures de processeurs multi-cœurs les plus récentes, les outils de traitement Matlab ont été conçus de sorte à être facilement modulables pour être utilisés avec d'autres types de transpondeurs cohérents tels que les transpondeurs à 100 Gbit/s de dernière génération.

Des analyses plus poussées restent à faire à plusieurs niveaux :

Tout d'abord, en ce qui concerne les paramètres mesurés par le transpondeur cohérent, certains problèmes rencontrés lors des phases de capture n'ont pas permis d'obtenir un échantillon de mesures significatif pour chaque saison. Une étude des paramètres de performance mesurés par des transpondeurs cohérents sur plusieurs saisons et plusieurs autres types de liaisons reste à faire, ce qui permettra de mieux comprendre l'influence de facteurs environnementaux sur la polarisation du signal.

Le second axe d'analyse qui mériterait d'être approfondi porte sur la comparaison entre les mesures du transpondeur cohérent et celles du polarimètre. En effet, une analyse plus poussée nécessiterait de prendre en compte le décalage de l'état de polarisation de référence entre les deux instruments (Crawford et al., 2014). Une meilleure connaissance de la fiabilité des mesures des paramètres de Stokes par un transpondeur cohérent pourrait permettre aux opérateurs d'utiliser éventuellement les modems cohérents pour effectuer le monitoring des effets de polarisation dans leurs réseaux. Une telle utilisation permettrait de développer des outils permettant d'analyser activement l'évolution de l'état de polarisation des liaisons aériennes afin d'anticiper d'éventuels problèmes.

Le troisième axe qui mériterait une étude complémentaire serait une étude des phénomènes physiques associés aux vitesses de rotation du vecteur de Stokes supérieures à 1,0 rad/ms qui sont caractérisés par une distribution de type exponentiel.

Le quatrième axe qui pourrait faire l'objet d'une étude complémentaire porte sur le filtrage des mesures d'état de polarisation réalisées avec le polarimètre. En effet, une étude du bruit du polarimètre a permis d'obtenir des éléments pouvant permettre de mieux comprendre la déviation de la distribution des vitesses de rotation du vecteur de Stokes inférieures à 1,0 rad/ms par rapport à une loi de Rayleigh.

ANNEXE I

TRAITEMENT INITIAL DES DONNÉES DU TRANSPONDEUR COHÉRENT

Le Script-A I-1 effectue le traitement initial des données du transpondeur cohérent.

Entrée

Ce script prend deux paramètres obligatoires :

- **folder**, chaîne de caractères contenant le chemin du dossier contenant les fichiers à traiter;
- **fileList**, contenant la liste des fichiers à traiter (sans chemin absolu). Cette liste doit être de type *Character Array*.

Ce script accepte plusieurs paramètres facultatifs :

- **monitorTime**, pour activer (1) ou désactiver (0) le chronométrage du calcul;
- **verbose**, pour activer (1) ou désactiver (0) l'affichage des étapes intermédiaires du calcul;
- **timestamp**, **inst_PMD**, **inst_PDL**, **PreFECBER**, **SNR_X**, **SNR_Y**, **S1**, **S2**, **S3**, **Ch_Pw**, pour activer (1) ou désactiver (0) à la demande le traitement d'un ou des paramètres donnés.

Sortie

Le script écrit un fichier par paramètre, contenant l'ensemble des mesures du dossier.

Dépendances

Le script dépend :

- de la fonction *find_glitch_end*, présentée au Script-A III-1;
- de la fonction *partitionnement_parallel*, présentée au Script-A V-1.

Script-A I-1 Traitement initial des données du transpondeur cohérent

```

001 function [ output_struct, opts ] = ocl_processing_columns(varargin)
002
003 nom_colonnes = {'Timestamp','inst_PMD','inst_PDL','PreFECBER',...
004      'SNR_X','SNR_Y','S1','S2','S3','Ch_Pwr'};
005
006 p = inputParser;
007 p.addRequired('Folder', @isstr);
008 p.addRequired('FileList', @ischar);
009 for workerId=1:length(nom_colonnes)
010     p.addValue(nom_colonnes{workerId}, 0, @(x) ((x==0) || (x==1)));
011 end
012 p.addValue('MonitorTime', 1, @(x) ((x == 0) || (x == 1)));
013 p.addValue('Verbose', 1, @(x) ((x == 0) || (x == 1)));
014 p.parse(varargin{:});
015 opts = p.Results;
016
017 folder = opts.Folder;
018 files_list = opts.FileList;
019
020 nFiles=size(files_list,1);
021
022 nbWorker = matlabpool('size');
023 if nbWorker==0, nbWorker=1; end
024
025 colonne_a_traiter=opts.Timestamp+2*opts.inst_PMD+3*opts.inst_PDL+...
026      4*opts.PreFECBER+5*opts.SNR_X+6*opts.SNR_Y+7*opts.S1+...
027      8*opts.S2+9*opts.S3+10*opts.Ch_Pwr;
028
029 if opts.Verbose
030     fprintf('%d files (%d worker(s))\nFolder : %s , column %d)\n',...
031           nFiles, nbWorker, folder, colonne_a_traiter);
032 end
033
034 % Initialisation des variables
035 output_struct = struct();

```

```

036 output_struct.glitch_tab= [] ;
037 output_struct.file_error_list = cell(0) ;
038
039 for workerId=1:length(nom_colonnes)
040     if opts.(nom_colonnes{workerId})
041         output_struct.(nom_colonnes{workerId}) = [] ;
042         output_struct.(strcat('min_max_',nom_colonnes{workerId}))= [] ;
043     end
044 end
045
046 D=partitionnement_parallel(1:nFiles,nbWorker) ;
047
048 spmd
049     [ output_struct_ocld_comp ] =...
050         ocld_processing_columns_sub_worker( ...
051             folder, ...
052             files_list(D{labindex},:), ...
053             nom_colonnes{1}, opts.Timestamp, ...
054             nom_colonnes{2}, opts.inst_PMD, ...
055             nom_colonnes{3}, opts.inst_PDL, ...
056             nom_colonnes{4}, opts.PreFECBER, ...
057             nom_colonnes{5}, opts.SNR_X, ...
058             nom_colonnes{6}, opts.SNR_Y, ...
059             nom_colonnes{7}, opts.S1, ...
060             nom_colonnes{8}, opts.S2, ...
061             nom_colonnes{9}, opts.S3, ...
062             nom_colonnes{10}, opts.Ch_Pwr, ...
063             'MonitorTime', opts.MonitorTime, ...
064             'Verbose',opts.Verbose );
065 end
066
067 % Rassemblement des valeurs
068 for workerId=1:nbWorker
069     output_struct_ocld_temp = output_struct_ocld_comp{workerId} ;
070
071     output_struct.glitch_tab((end+1):(end+...
072         size(output_struct_ocld_temp.glitch_tab,1)),1:2)=...

```

```

073     output_struct_ocld_temp.glitch_tab;
074
075     for columnId=1:length(nom_colonnes)
076         if opts.(nom_colonnes{columnId})
077             output_struct.(nom_colonnes{columnId})...
078                 ((end+1):(end+size(output_struct_ocld_temp....
079                     (nom_colonnes{columnId}),1)),1)=...
080                     output_struct_ocld_temp.(nom_colonnes{columnId});
081             output_struct.(strcat('min_max_',...
082                 nom_colonnes{columnId}))((end+1):(end+...
083                     size(output_struct_ocld_temp.(strcat('min_max_',...
084                         nom_colonnes{columnId})),1)),1:2)=...
085                     output_struct_ocld_temp.(strcat('min_max_',...
086                         nom_colonnes{columnId}));
087         end
088     end
089
090     for fileErrorId=1:length(output_struct_ocld_temp.file_error_list)
091         output_struct.file_error_list{end+1}=...
092             output_struct_ocld_temp.file_error_list{fileErrorId};
093     end
094 end
095 end
096
097 function [output_struct]=ocld_processing_columns_sub_worker(varargin)
098 nom_colonnes = {'Timestamp','inst_PMD','inst_PDL','PreFECBER',...
099     'SNR_X','SNR_Y','S1','S2','S3','Ch_Pwr'};
100
101 p = inputParser;
102 p.addRequired('Folder', @isstr);
103 p.addRequired('FileList', @ischar);
104 for i=1:length(nom_colonnes)
105     p.addValue(nom_colonnes{i},0,@(x)((x == 0) || (x == 1)));
106 end
107 p.addValue('MonitorTime', 1, @(x)((x == 0) || (x == 1)));
108 p.addValue('Verbose', 1, @(x)((x == 0) || (x == 1)));
109 p.parse(varargin{:});

```

```
110 opts = p.Results; % Structure of input options
111
112 folder = opts.Folder;
113 files_list = opts.FileList;
114 nFiles=size(files_list,1);
115 % Initialisation des variables
116 output_struct=struct();
117 output_struct.glitch_tab = zeros(size(files_list,1),1);
118 output_struct.file_error_list = cell(0);
119
120 for i=1:length(nom_colonnes)
121     if opts.(nom_colonnes{i})
122         output_struct.(nom_colonnes{i}) = [];
123         output_struct.(strcat('info_',nom_colonnes{i})) = [];
124     end
125 end
126
127 if opts.MonitorTime
128     tic
129 end
130
131 for fileId=1:size(files_list,1)
132
133     if opts.Verbose&&(mod(fileId,100)==0)&&(labindex==1)
134         fprintf('%d files (%.2f%%)\n',fileId,fileId/nFiles*100);
135         toc
136     end
137     file_name=strupr(files_list(fileId,:));
138     chemincomplet=strcat(folder,file_name);
139
140     try
141         FileData = load(chemincomplet);
142     catch
143         fprintf('Cannot read %s\n',file_name(:));
144         output_struct.file_error_list{end+1}=file_name;
145
146         continue;

```

```
147     end
148
149     SORTIE = FileData.SORTIE;
150
151     init_index = find_glitch_end(SORTIE);
152     output_struct.glitch_tab(fileId,1:2)=[init_index length(SORTIE)];
153
154     if (length(SORTIE)-init_index)<1000
155         continue
156     end
157
158     for i=1:length(nom_colonnes)
159         if opts.(nom_colonnes{i})
160             output_struct.(nom_colonnes{i})...
161                 ((end+1):(end+size(SORTIE(init_index:end,i),1)),1)...
162                 =SORTIE(init_index:end,i);
163             output_struct.(strcat('min_max_',nom_colonnes{i}))...
164                 (fileId,1:2) = [min(SORTIE(init_index:end,i))...
165                               max(SORTIE(init_index:end,i))];
166         end
167     end
168
169 end
170
171 if opts.MonitorTime
172     toc
173 end
174 end
```

ANNEXE II

CONVERSION DES FICHIERS DE MESURE DU TRANSPONDEUR COHÉRENT EN FORMAT MATLAB

Le Script-A II-1 permet de convertir au format Matlab les fichiers de mesures brutes du transpondeur cohérent.

Entrée

Ce script prend deux paramètres :

- **folder**, chaîne de caractères contenant le chemin du dossier contenant les fichiers à traiter;
- **file_list**, contenant la liste des fichiers à traiter (sans chemin absolu). Cette liste doit être de type *Character Array*.

Sortie

Le script lit séquentiellement l'ensemble des fichiers donnés en paramètre et écrit pour chaque fichier, un fichier MAT contenant les données du fichier CSV source.

Script-A II-1 Conversion des fichiers de mesure du transpondeur cohérent du format CSV au format MAT.

```
001 function [] = convert_csv_mat(folder, file_list)
002 clear
003 clc
004 for fileIx=1:nFiles
005     nomfichier=strtrim(files_list(fileIx,:));
006     chemincomplet=strcat(folder,nomfichier);
007
008     try
009         % Lecture du fichier csv
010         tab_mesure=csvread(chemincomplet,2,0);
011
012         % Determination du timestamp du fichier
```

```

013     parsing=textscan(nomfichier, '%f_%f_%f');
014     TS = parsing{1,2};
015
016     % Remplissage de la matrice
017     SORTIE=zeros(length(tab_mesure),9);
018     for k=1:length(tab_mesure)
019         SORTIE(k,1)=TS;
020         TS=addtodate(TS,1,'millisecond');
021     end
022     SORTIE(:,2)=tab_mesure(:,9);      %inst_PMD
023     SORTIE(:,3)=tab_mesure(:,10);    %inst_PDL
024     SORTIE(:,4)=tab_mesure(:,8);     %PreFECBER
025     SORTIE(:,5)=tab_mesure(:,6);     %SNR_X
026     SORTIE(:,6)=tab_mesure(:,7);     %SNR_Y
027     SORTIE(:,7)=tab_mesure(:,12);   %S1
028     SORTIE(:,8)=tab_mesure(:,13);   %S2
029     SORTIE(:,9)=tab_mesure(:,14);   %S3
030     SORTIE(:,10)=tab_mesure(:,35);  %Ch_Pwr
031
032     % Exportation des donnees
033     save(strcat(nomfichier',...
034         1:length(nomfichier)-4),'.mat'), 'SORTIE', 'nomfichier');
035
036     catch
037         fprintf('Cannot read %d \n',nomfichier);
038         continue;
039     end
040 end
041 end
042
043 end

```

ANNEXE III

TRAITEMENT DES VALEURS TRANSITOIRES DU TRANSPONDEUR COHÉRENT

Le Script-A III-1 permet de déterminer l'ensemble des données transitoires d'un fichier de mesure du transpondeur cohérent.

Entrée

Le script prend en paramètre la matrice **SORTIE**, résultat donné par le Script-A II-1.

Sortie

Le script retourne l'indice **init_index** correspondant à la première ligne valide des mesures contenues dans la matrice *SORTIE*, c'est-à-dire la fin des mesures transitoires.

Script-A III-1 Traitement des valeurs transitoires du transpondeur cohérent

```
001 function [ init_index ] = find_glitch_end( SORTIE )
002 % Critère pour la PMD (colonne 2) : 0<PMD
003 % Critère pour la PDL (colonne 3) : 0<PDL
004 % Critère pour la BER (colonne 4) : BER<1
005
006 init_index=length(SORTIE);
007 while ( SORTIE(init_index,2)>0 &&...
008     SORTIE(init_index,3)>0 &&...
009     SORTIE(init_index,4)<1 &&...
010     (init_index>1) )
011     init_index=init_index-1;
012 end
013 % On se situe ici sur la dernière valeur NON VALIDE
014 init_index=min(init_index+200,length(SORTIE));
015 end
```


ANNEXE IV

MOYENNAGE DES MESURES DU TRANSPONDEUR COHÉRENT

Le Script-A IV-1 permet de calculer la valeur moyenne des paramètres mesurés par le transpondeur cohérent à la minute et à la seconde.

Entrée

Ce script prend trois paramètres :

- **structure_timestamp**, qui est la structure retournée par le script de traitement initial des données du transpondeur cohérent, appliquée aux valeurs d'horodatage;
- **structure_variable**, qui est la structure retournée par le script de traitement initial des données du transpondeur cohérent, appliquée à un des paramètres (*inst_PMD* par exemple);
- **calcul_acos**, pour activer (1) ou désactiver (0) le calcul de la valeur moyenne de l'arc cosinus de *sctucture_variable*. Cela permet de calculer le paramètre ϕ à partir de S_1 .

Sortie

Le script retourne une structure, **structure_moyennes**, contenant la moyenne et l'écart type du paramètre étudié à la minute et à la seconde. Chaque ligne est constituée de cinq colonnes : horodatage, moyenne, écart-type, instants de début et de fin pour le calcul des paramètres statistiques (moyenne et écart-type) de la ligne.

Script-A IV-1 Moyennage des mesures du transpondeur cohérent

```
001 function [structure_moyennes] =...
002     compute_mean_ocld( structure_timestamp, ...
003     structure_variable, ...
004     calcul_acos )
005
006 nom_classes = {'sec','min'};
```

```

007
008 structure_moyennes = struct();
009
010 % On suppose que toutes les mesures sont avec le même TS (logique car
011 % traites par un même script initialement). On charge alors une
012 % colonne pour travailler sur les TS
013
014 % Chargement des données du fichier
015 [ts, signature]=sort(structure_timestamp);
016
017 for index_classe=1:2
018     fprintf('Partitionning timestamps ');
019     switch (index_classe)
020         case 1
021             fprintf('(seconds processing) : ');
022             pas=datenum(0,0,0,0,0,1);
023             segments = segmentation_temporelle( ts, pas );
024
025         case 2
026             fprintf('(minutes processing) : ');
027             pas=datenum(0,0,0,0,1,0);
028             segments = segmentation_temporelle( ts, pas );
029     end
030
031     % Identification et suppression des variations invalides
032     diff_ts=diff(ts);
033     blacklist=find(diff_ts>pas);
034     ts_var=ts(1:(end-1),1);
035     ts_var(blacklist)=[];
036     fprintf('Partitionning timestamps for derivate : ');
037     segments_var = segmentation_temporelle( ts_var, pas );
038
039     input=structure_variable;
040     input=input(signature);
041
042     fprintf('Computing %s : ', nom_classes{index_classe});
043     structure_moyennes.(nom_classes{index_classe})=...

```

```

044     moyenne_tableau(input,segments);
045
046     input_var=abs(diff(input));
047     input_var(blacklist)=[];
048     fprintf('Computing diff (%s) : ',nom_classes{index_classe});
049     structure_moyennes.(strcat('d',nom_classes{index_classe}))=...
050         moyenne_tableau(input_var,segments_var);
051
052     % Post-Traitements spécifiques
053     if calcul_acos
054         input=acos(input);
055         fprintf('Computing %s : ', nom_classes{index_classe});
056         structure_moyennes.(nom_classes{index_classe})=...
057             moyenne_tableau(input,segments);
058
059         input_var=abs(diff(input));
060         input_var(blacklist)=[];
061         fprintf('Computing diff (%s) : ',...
062             nom_classes{index_classe});
063         structure_moyennes.(strcat('d',...
064             nom_classes{index_classe}))=...
065             moyenne_tableau(input_var,segments_var);
066
067     end
068
069 end
070
071 end
072
073 function [ segments ] = segmentation_temporelle( input, PAS )
074 % Retourne les segments de mesures contigues espaces de PAS
075
076 % input=sortrows(input,1);
077 taille_input = length(input);
078 segments=[];
079 curseur=1;
080 while input(curseur,1)==0

```

```

081     curseur=curseur+1;
082 end
083 timestamp_S1 = input(curseur,1);
084 reverseStr='';
085 while curseur<taille_input
086     % Affichage de l'avancement
087     if mod(curseur,1000)==0
088         percentDone = curseur / taille_input * 100;
089         msg = sprintf('%.3.1f% %%', percentDone);
090         fprintf([reverseStr, msg]);
091         fprintf('%%');
092         reverseStr = repmat(sprintf('\b'), 1, length(msg)+1);
093     end
094
095     % Traitement des segments
096     curseur_1=curseur;
097     while (input(curseur,1)>=timestamp_S1) &&...
098         (input(curseur,1)<timestamp_S1 + PAS) &&...
099         (curseur<taille_input)
100         curseur=curseur+1;
101     end
102     curseur_2=curseur-1;
103     segments(end+1,1:3)=[timestamp_S1 curseur_1 curseur_2];
104     if curseur<taille_input
105         timestamp_S1=timestamp_S1+PAS;
106         TS_s1_temp=input(curseur,1);
107         while ~((timestamp_S1<=TS_s1_temp) &&...
108             (TS_s1_temp<timestamp_S1+PAS))
109             timestamp_S1=timestamp_S1+PAS;
110         end
111     end
112
113 end
114 fprintf([reverseStr, 'done!\n']);
115 end
116
117 function [ output ] = moyenne_tableau( input, segments )

```

```
118 % input=sortrows(input,1);
119 taille_segments = length(segments);
120 output=zeros(size(segments,1),3);
121 reverseStr='';
122 for intervalle = 1:size(segments,1)
123     if mod(intervalle,1000)==0
124         percentDone = intervalle / taille_segments * 100;
125         msg = sprintf('%3.1f% %%', percentDone);
126         fprintf([reverseStr, msg]);
127         fprintf('%%');
128         reverseStr = repmat(sprintf('\b'), 1, length(msg)+1);
129     end
130     timestamp_S1=segments(intervalle,1);
131     intervalle_1=segments(intervalle,2);
132     intervalle_2=segments(intervalle,3);
133     output(intervalle,1:5)=[timestamp_S1, ...
134         mean(input(intervalle_1:intervalle_2,1)), ...
135         std(input(intervalle_1:intervalle_2,1)), ...
136         intervalle_1,intervalle_2];
137 end
138 fprintf([reverseStr, 'done!\n']);
139 end
```


ANNEXE V

PARITIONNEMENT DES FICHIERS DE MESURE PRÉALABLEMENT AU CALCUL PARALLÈLE

Le Script-A V-1 est une fonction assurant le partitionnement d'un ensemble d'indices *liste* (représentant chacun un fichier) en *nb_part* sous-ensembles répartis de manière égale. Le script retourne une variable de type *cell* contenant *nb_part* intervalles dont la réunion est égale à *liste*.

Entrée

Ce script prend deux paramètres :

- **liste**, qui est un vecteur Matlab dont l'une des dimensions est unitaire;
- **nbpart**, qui est un entier désignant le nombre de sous ensembles désiré.

Sortie

Ce script retourne une variable **intervalles** de type *Cell* à *nbpart* éléments dont chaque élément correspond à un sous ensemble de *liste*.

Script-A V-1 Partitionnement des fichiers de mesure préalablement au calcul parallèle

```
001 function [ intervalles ] = partitionnement_parallelle(liste,nb_part)
002 nb_elem = length(liste);
003 k=floor(nb_elem/nb_part);
004 i=1;
005 r=nb_elem-k*nb_part;
006 part = k*ones(1,nb_part);
007 while r>0
008     part(1,i)=part(1,i)+1;
009     i=i+1;
010     r=r-1;
011 end
012 for i=nb_part:-1:1
```

```
013     temp=1;
014     for j=1:(i-1)
015         temp=temp+part(1,j);
016     end
017     part(1,i)=temp;
018 end
019 part(1,nb_part+1)=nb_elem+1;
020 intervalles = cell(0);
021 for i = 1:nb_part
022     intervalles{i} = liste(part(i):part(i+1)-1);
023 end
024 intervalles=intervalles';
025 end
```

ANNEXE VI

TRAITEMENT INITIAL DES DONNEES DU POLARIMÈTRE

Le Script-A VI-1 effectue le traitement initial des données du polarimètre.

Entrée

Ce script prend deux paramètres obligatoires :

- **folder**, chaîne de caractères contenant le chemin du dossier contenant les fichiers à traiter;
- **fileList**, contenant la liste des fichiers à traiter (sans chemin absolu). Cette liste doit être de type *Character Array*.

Ce script accepte d'autres paramètres facultatifs :

- **bins**, pour définir manuellement la largeur des colonnes de l'histogramme des vitesses de rotation du vecteur de Stokes;
- **manageHist**, pour activer (1) ou désactiver (0) le calcul de l'histogramme des vitesses de rotation du vecteur de Stokes;
- **manageEvents**, pour activer (1) ou désactiver (0) la détection des effets rapides de polarisation;
- **manualGlitch**, pour définir manuellement la fin de la période transitoire pour chaque fichier;
- **monitorTime**, pour activer (1) ou désactiver (0) le chronométrage du calcul;
- **verbose**, pour activer (1) ou désactiver (0) l'affichage des étapes intermédiaires du calcul.

Sortie

Le script retourne une variable, **output_struct_stats**, qui est une structure Matlab contenant neuf variables :

- **output_struct_stats.TS**, matrice à une dimension contenant la valeur d'horodatage de l'ensemble des fichiers traités. Ce vecteur a autant d'éléments que de fichiers traités;

- **output_struct_stats.glitch_tab**, matrice à trois colonnes et autant de lignes que de fichiers traités. La première colonne contient l'indice correspondant au premier point de mesure valide (c'est-à-dire la fin des mesures transitoires), la seconde colonne contient la valeur maximale de vitesse de rotation observée avant l'élimination des valeurs transitoires. La troisième colonne contient la valeur maximale de vitesse de rotation observée à partir de la fin de la période transitoire;
- **output_struct_stats.file_error_list**, variable de type Cell contenant la liste des fichiers corrompus (c'est-à-dire qui n'ont pas pu être ouverts par le script);
- **output_struct_stats.mean_ARR**, matrice à une dimension contenant la valeur moyenne de la vitesse de rotation du vecteur de Stokes observée par fichier (sur les mesures non transitoires);
- **output_struct_stats.std_ARR**, matrice à une dimension contenant l'écart type de la vitesse de rotation du vecteur de Stokes observée par fichier (sur les mesures non transitoires);
- **output_struct_stats.min_ARR**, vecteur Matlab à une dimension contenant la valeur minimale de la vitesse de rotation du vecteur de Stokes observée par fichier (sur les mesures non transitoires);
- **output_struct_stats.max_ARR**, matrice à une dimension contenant la valeur maximale de la vitesse de rotation du vecteur de Stokes observée par fichier (sur les mesures non transitoires);
- **output_struct_stats.hist_ARR**, matrice à une dimension contenant les valeurs de l'histogramme de la vitesse de rotation du vecteur de Stokes pour chaque mesure du polarimètre. Les colonnes correspondent à la valeur du vecteur *Bins* donné comme paramètre au script;
- **output_struct_stats.events_SOP**, structure Matlab décrite à la section 2.3.2.4.

Dépendances

Le script dépend :

- de la fonction *partitionnement_parallel*, présentée au Script-A V-1;

- de la fonction *pola_angular_velocity_matrix*, présentée au Script-A VII-1.

Script-A VI-1 Traitement initial des données du polarimètre

```

001 function [output_struct_stats] = pola_processing_stats(varargin)
002 %pola_global_processing Traitement des donnees du polarimetre.
003
004 p = inputParser;
005 p.addRequired('Folder', @isstr);
006 p.addRequired('FileList', @ischar);
007 p.addValue('Bins', [0:.01:20 inf], @(x) isreal(x) &&...
008     issorted(x) && x(1)<x(end));
009 p.addValue('ManageHist', 1, @(x)((x == 0) || (x == 1)));
010 p.addValue('ManageEvents', 1, @(x)((x == 0) || (x == 1)));
011 p.addValue('MonitorTime', 1, @(x)((x == 0) || (x == 1)));
012 p.addValue('ManualGlitch', 0, @(x)((x >= 0) && (x == fix(x))));
013 p.addValue('Verbose', 1, @(x)((x == 0) || (x == 1)));
014 p.parse(varargin{:});
015 opts = p.Results;
016
017 folder = opts.Folder;
018 files_list = opts.FileList;
019
020 %#ok<*AGROW>
021
022 % Recuperation du nombre de workers en vue du calcul parallele
023 nbWorker = matlabpool('size');
024 if nbWorker==0, nbWorker=1; end
025
026 % Initialisation de la structure de sortie
027 nFiles=size(files_list,1);
028 glitch_tab=zeros(nFiles,3);
029 file_error_list = cell(0);
030
031 mean_ARR = zeros(nFiles,1);
032 std_ARR = zeros(nFiles,1);
033

```

```

034 min_ARR = zeros(nFiles,1);
035 max_ARR = zeros(nFiles,1);
036
037 TS = zeros(nFiles,1);
038
039 hist_ARR = zeros(length(opts.Bins),1);
040 events_SOP=[];
041
042 if opts.Verbose
043     fprintf('%d files , %d worker(s)\nFolder : %s\n',...
044             nFiles,nbWorker,folder);
045 end
046
047 D=partitionnement_parallel(1:nFiles,nbWorker);
048
049 spmd
050     [output_struct_stats_comp] = pola_processing_stats_sub_worker(...
051         folder, ...
052         files_list(D{labindex},:), ...
053         'Bins', opts.Bins, ...
054         'ManageHist', opts.ManageHist, ...
055         'ManageEvents', opts.ManageEvents, ...
056         'MonitorTime', opts.MonitorTime, ...
057         'ManualGlitch', opts.ManualGlitch, ...
058         'Verbose', opts.Verbose );
059 end
060
061 % Rassemblement des valeurs
062 for i=1:nbWorker
063     output_struct_stats_temp = output_struct_stats_comp{i};
064
065     % Fusion des timestamps
066     TS(D{i})=output_struct_stats_temp.TS;
067
068     %Fusion des parametres statistiques
069     mean_ARR(D{i})=output_struct_stats_temp.mean_ARR;
070     std_ARR(D{i})=output_struct_stats_temp.std_ARR;

```

```
071
072     min_ARR(D{i})=output_struct_stats_temp.min_ARR;
073     max_ARR(D{i})=output_struct_stats_temp.max_ARR;
074
075     %Fusions des glitches
076     glitch_tab(D{i}, :)=output_struct_stats_temp.glitch_tab(:, :);
077
078     if opts.ManageHist
079         %Fusion des histogrammes d'ARR
080         hist_ARR=hist_ARR+output_struct_stats_temp.hist_ARR;
081     end
082
083     if opts.ManageEvents
084         % Fusion des evenements
085         events_SOP((end+1):(end+...
086             size(output_struct_stats_temp.events_SOP)), :)=...
087             output_struct_stats_temp.events_SOP;
088     end
089
090     for j=1:length(output_struct_stats_temp.file_error_list)
091         file_error_list{end+1}=...
092             output_struct_stats_temp.file_error_list{j};
093     end
094 end
095
096 output_struct_stats = struct();
097
098 output_struct_stats.TS=TS;
099
100 output_struct_stats.glitch_tab=glitch_tab;
101 output_struct_stats.file_error_list = char(file_error_list);
102
103 output_struct_stats.mean_ARR = mean_ARR;
104 output_struct_stats.std_ARR = std_ARR;
105 output_struct_stats.min_ARR = min_ARR;
106 output_struct_stats.max_ARR = max_ARR;
107
```

```

108 output_struct_stats.parametres = opts;
109
110 if opts.ManageHist
111     output_struct_stats.hist_ARR = hist_ARR;
112 end
113
114 if opts.ManageEvents
115     output_struct_stats.events_SOP=events_SOP;
116 end
117
118 end
119
120 function [ output_struct_stats ]=...
121     pola_processing_stats_sub_worker(varargin)
122 %pola_global_processing_sub_worker Traitement du travail reparti
123 %entre workers pour le traitement des donnees du polarimetre.
124
125 p = inputParser;
126 p.addRequired('Folder', @isstr);
127 p.addRequired('FileList', @ischar);
128 p.addValue('Bins', [0:.01:20 inf], @(x) isreal(x) &&...
129     issorted(x) && x(1)<x(end));
130 p.addValue('ManageHist', 1, @(x)((x == 0) || (x == 1)));
131 p.addValue('ManageEvents', 1, @(x)((x == 0) || (x == 1)));
132 p.addValue('MonitorTime', 1, @(x)((x == 0) || (x == 1)));
133 p.addValue('ManualGlitch', 0, @(x)((x >= 0) && (x == fix(x))));
134 p.addValue('Verbose', 1, @(x)((x == 0) || (x == 1)));
135 p.parse(varargin{:});
136 opts = p.Results; % Structure of input options
137
138 folder = opts.Folder;
139 files_list = opts.FileList;
140
141 %#ok<*AGROW>
142
143 %% Initialisation de la structure de sortie
144 nFiles=size(files_list,1);

```

```
145  glitch_tab=zeros(nFiles,3);
146  file_error_list = cell(0);
147
148  mean_ARR = zeros(nFiles,1);
149  std_ARR = zeros(nFiles,1);
150
151  min_ARR = zeros(nFiles,1);
152  max_ARR = zeros(nFiles,1);
153
154  TS = zeros(nFiles,1);
155
156  hist_ARR = zeros(length(opts.Bins),1);
157  events_SOP=[];
158
159 if opts.MonitorTime
160     tic
161 end
162
163 for fileId = 1:nFiles
164     if opts.Verbose&&(mod(fileId,100)==0)&&(labindex==1)
165         fprintf('%d files (%.2f%%)\n',fileId,fileId/nFiles*100);
166         if opts.MonitorTime
167             toc
168         end
169     end
170
171     file_name=strupr(files_list(fileId,:));
172     chemincomplet=strcat(folder,file_name);
173
174     % Sauter le traitement d'un fichier en cas d'erreur
175     try
176         FileData = load(chemincomplet);
177     catch %#ok<CTCH>
178         fprintf('Cannot read %s\n', file_name(:));
179         file_error_list{end+1}=file_name;
180         continue;
181     end
```

```

182
183     % Chargement de la matrice des etats de polarisation
184     S = FileData.sop;
185
186     % Normalisation du vecteur de Stokes
187     Snorm = bsxfun(@rdivide, S, sqrt(sum(S.^2,2)));
188
189     % Calcul de la matrice des vitesses angulaires
190     angular_velocity=pola_angular_velocity_matrix(Snorm);
191
192     if ~opts.ManualGlitch
193         % Localisation du 'glitch' en debut de mesure
194         [MaxVal, loc_glitch] = max(angular_velocity(:,1));
195         initialisation = min(loc_glitch+50,size(angular_velocity,1));
196     else
197         initialisation = opts.ManualGlitch;
198         MaxVal = angular_velocity(initialisation,1);
199     end
200     glitch_tab(fileId,1:3) = [initialisation, MaxVal, ...
201                               max(angular_velocity(initialisation:end,1))];
202
203     % Calcul de l'horodatage en fonction du nom du fichier
204     parsed_name=cell2mat(textscan(file_name, ...
205                           '%f_%f_%f_%f_%f_%f'));
206     TS_file = datenum(parsed_name(2),parsed_name(3),...
207                         parsed_name(4),parsed_name(5),parsed_name(6),parsed_name(7));
208     TS(fileId)=TS_file;
209
210     % Calcul de la valeur moyenne des rotations pour le fichier
211     mean_ARR(fileId)= mean(angular_velocity(initialisation:end),1);
212
213     % Calcul de l'écart type des rotations pour le fichier
214     std_ARR(fileId)= std(angular_velocity(initialisation:end),1);
215
216     % Calcul la valeur minimale des rotations pour le fichier
217     min_ARR(fileId)= min(angular_velocity(initialisation:end));
218

```

```

219      % Calcul la valeur maximale des rotations pour le fichier
220      max_ARR(fileId)= max(angular_velocity(initialisation:end));
221
222      if opts.ManageHist
223          % Calcul de l'histogramme des rotations angulaires
224          hist_ARR=hist_ARR+...
225              histc(angular_velocity(initialisation:end),opts.Bins);
226      end
227
228      if opts.ManageEvents
229          % Determination des evenements de polarisation
230          SOP_events_temp = pola_compute_events(...,
231              initialisation, angular_velocity, Snorm');
232          if size(SOP_events_temp,1)>0
233              size_SOP_events=size(events_SOP,1);
234              events_SOP((size_SOP_events+1):(size_SOP_events+...
235                  size(SOP_events_temp)),1)=TS_file;
236              events_SOP((size_SOP_events+1):(size_SOP_events+...
237                  size(SOP_events_temp)),2:6)=SOP_events_temp;
238          end
239      end
240
241  end
242
243 output_struct_stats = struct();
244 output_struct_stats.TS=TS;
245 output_struct_stats.glitch_tab=glitch_tab;
246 output_struct_stats.file_error_list = file_error_list;
247 output_struct_stats.mean_ARR = mean_ARR;
248 output_struct_stats.std_ARR = std_ARR;
249 output_struct_stats.min_ARR = min_ARR;
250 output_struct_stats.max_ARR = max_ARR;
251
252 if opts.ManageHist
253     output_struct_stats.hist_ARR = hist_ARR;
254 end
255

```

```
256 if opts.ManageEvents
257     output_struct_stats.events_SOP=events_SOP;
258 end
259
260 if opts.MonitorTime
261     toc
262 end
263 end
```

ANNEXE VII

CALCUL DE LA MATRICE DES VITESSES DE ROTATION DU VECTEUR DE STOKES

Le Script-A VII-1 permet de calculer la matrice des vitesses de rotation du vecteur de Stokes à partir de la matrice contenant les paramètres de Stokes.

Entrée

Le script prend un paramètre, S , qui correspond à une matrice à trois colonnes contenant les mesures des trois paramètres de Stokes S_1 , S_2 et S_3 .

Sortie

La fonction retourne une matrice *angular_velocity* à une dimension, contenant les valeurs de vitesse de rotation du vecteur de Stokes relatives à la matrice S donnée en entrée.

Script-A VII-1 Calcul de la matrice des vitesses de rotation du vecteur de Stokes

```
001 function [ angular_velocity ] = pola_angular_velocity_matrix( S )
002 % Duree d'echantillonnage, exprimee en millisecondes
003 dt = 0.05;
004
005 dot_product = sum(S(1:end-1,:) .* S(2:end,:),2);
006
007 % Elimination des artefacts de calcul
008 dot_product(dot_product < -1) = -1;
009 dot_product(dot_product > +1) = +1;
010
011 angular_velocity = acos(dot_product)./dt; % Unite : [rad/msec]
012
013 end
```


ANNEXE VIII

DÉTECTION DES ÉVÉNEMENTS DE POLARISATION

Le Script-A VIII-1 assure la détection des événements de polarisation. Ce script prend en paramètre l'instant correspondant à la fin de la période transitoire des mesures du polarimètre, la matrice des vitesses de rotation angulaires et la matrice normalisée des vecteurs de Stokes.

Entrée

Le script prend trois paramètres en entrée :

- **initialisation**, entier correspondant à l'indice de fin de période de mesure transitoire;
- **angular_velocity**, matrice à une dimension correspondant à la matrice des vitesses de rotation du vecteur de Stokes;
- **sop**, matrice à trois colonnes contenant les mesures des trois paramètres de Stokes S_1 , S_2 et S_3 .

Sortie

Le script retourne une structure Matlab **SOP_events** constituée de 5 colonnes. Chaque ligne de cette structure correspond à un événement de polarisation. Les informations stockées dans chaque colonne sont les suivantes :

- horodatage du fichier contenant l'événement;
- instant de début de l'événement;
- instant de fin de l'événement;
- vitesse de rotation maximale observée durant l'événement;
- déflexion angulaire maximale durant l'événement;
- déflexion angulaire finale de l'événement.

Script-A VIII-1 Détection des événements de polarisation

```
001 function [ SOP_events ] = ...
```

```

002     pola_compute_events( initialisation, angular_velocity, sop )
003 %compute_defl_d1 Returns a matrix : event_begin_index|event_end_index
004 %|max angularvelocity in event|max relative angular_velocity
005 %|last angular velocity using first definition of events
006
007 threshold1=0.9;
008 threshold2=0.45;
009
010 size_angular_velocity=size(angular_velocity,1);
011 SOP_events = [];
012 curseur=initialisation;
013
014 %% Parcours du fichier
015
016 while curseur<(size_angular_velocity-200)
017     while (angular_velocity(curseur)<threshold1) &&...
018         (curseur<(size_angular_velocity-200))
019         curseur=curseur+1;
020     end
021     if curseur~=(size_angular_velocity-200)
022         event_start_th1=curseur;
023         event_start_th2=event_start_th1;
024         while (angular_velocity(event_start_th2,1)>threshold2) &&...
025             (event_start_th2>1)
026             event_start_th2 = event_start_th2-1;
027         end
028         curseur=curseur+200;
029         event_end_th1=curseur;
030         while (angular_velocity(event_end_th1,1)<threshold1) &&...
031             (event_end_th1>1)
032             event_end_th1 = event_end_th1-1;
033         end
034         event_end_th2=event_end_th1;
035         while (angular_velocity(event_end_th2,1)>threshold2) &&...
036             (event_end_th2<size_angular_velocity)
037             event_end_th2 = event_end_th2+1;
038         end

```

```
039
040      %% Calcul des deflexions
041      matdf=zeros(3,event_end_th2-event_start_th2);
042      sub_matrix=sop(1:3,(event_start_th2+1):(event_end_th2));
043      for k=1:size(sub_matrix,2)
044          matdf(1:3,k)=sop(1:3,event_start_th2).*sub_matrix(1:3,k);
045      end
046
047      dot_product_relative = sum(matdf);
048
049      % Cette étape permet de supprimer les artefacts de calcul
050      dot_product_relative(dot_product_relative < -1) = -1;
051      dot_product_relative(dot_product_relative > +1) = +1;
052
053      angular_position_relative=...
054          acos(dot_product_relative);
055      angular_position_relative=...
056          transpose(angular_position_relative);
057
058      SOP_events(end+1,1:5)=...
059          [event_start_th2,event_end_th2,...]
060          max(angular_velocity(event_start_th2:event_end_th2,1))...
061          ,max(angular_position_relative(:,1)),...
062          angular_position_relative(end,1)];
063      end
064  end
```


ANNEXE IX

ALGORITHME DE DÉTECTION DES ÉVÉNEMENTS DE POLARISATION

La Figure-A IX-1 présente, l'algorithme suivi pour la détection d'un événement de polarisation.

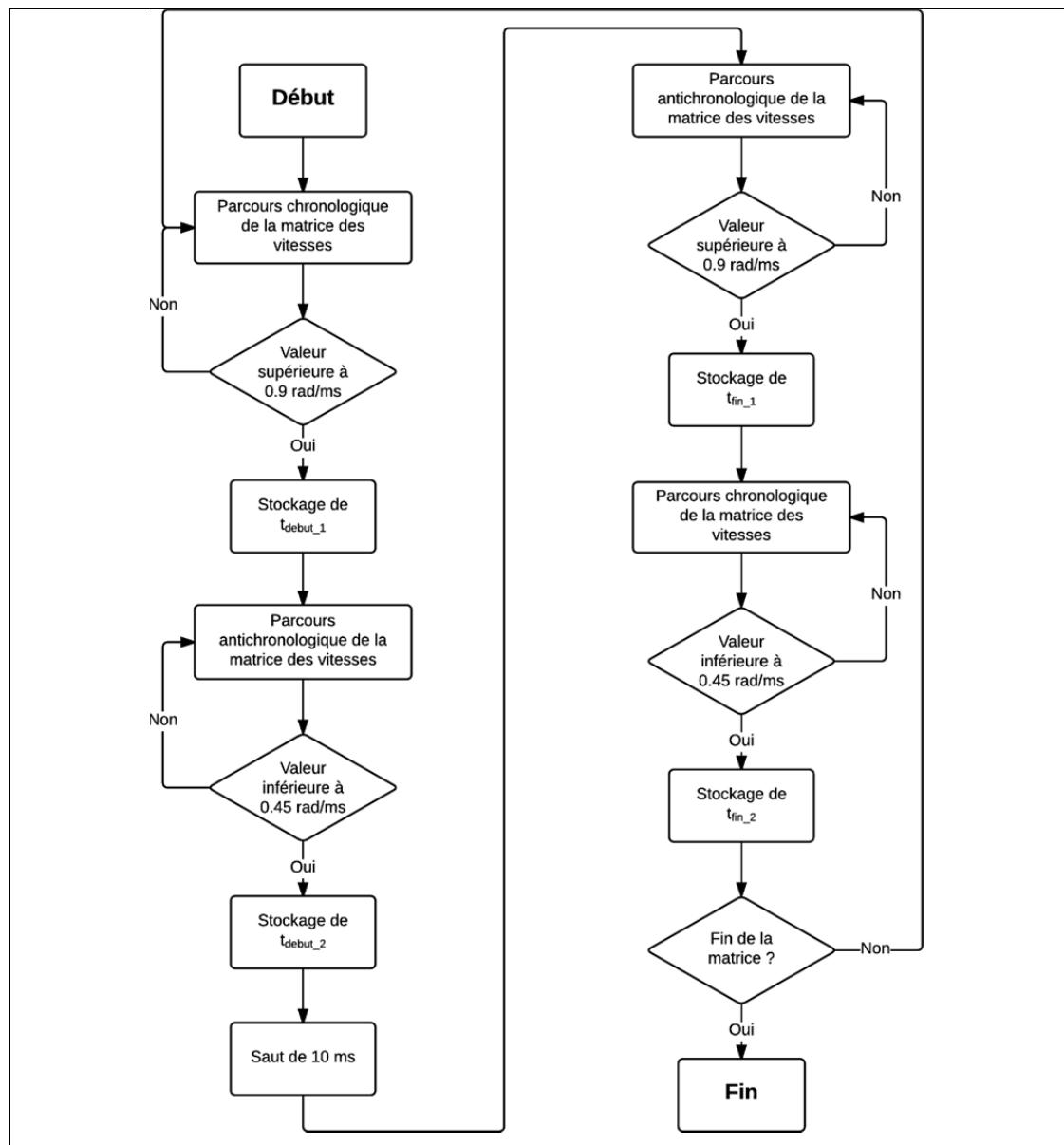


Figure-A IX-1 Algorithme utilisé pour la détection des événements de polarisation

ANNEXE X

MOYENNAGE DES MESURES DU POLARIMÈTRE

Le Script-A X-1 permet de moyenner les mesures du polarimètre à la seconde.

Entrée

Ce script prend en paramètre la structure de données, **stats_file**, obtenue à l'issue du traitement initial des données du polarimètre, réalisé à travers le Script-A VI-1.

Sortie

Le script retourne une structure Matlab, **output_struct**, contenant deux structures : S et Phi contenant chacune trois variables :

- **ts**, matrice à une dimension contenant les valeurs d'horodatage des mesures moyennées;
- **mean**, matrice à une dimension contenant la moyenne du paramètre pour chaque valeur d'horodatage indiquée dans la matrice TS;
- **std**, matrice à une dimension contenant l'écart type du paramètre pour chaque valeur d'horodatage indiquée dans la matrice TS.

Script-A X-1 Moyennage des mesures du polarimètre à la seconde

```
001 function [ output_struct ] = compute_mean_pola( stats_file )
002
003 output_struct = struct();
004
005 mean_struct = struct();
006
007 nFiles = size(stats_file.fichiers_traites,1);
008 glitch_tab = stats_file.glitch_tab(:,1);
009
010 TS = stats_file.TS;
011
```

```

012 fprintf('Segmentation...\n');
013
014 [ mean_struct.groups, mean_struct.periods ] =...
015 segmentation_temporelle_pola( TS, glitch_tab, 2e5, ...
016 datenum(0,0,0,0,0,1/(20e3)), datenum(0,0,0,0,0,1));
017
018 mean_struct.files = stats_file.fichiers_traites;
019 mean_struct.folder = stats_file.chemin_dossier;
020
021
022 fprintf('done!\n');
023
024 fichier_courant = mean_struct.groups(1,2);
025 CurrentFile=load(sprintf('%s%s',mean_struct.folder, ...
026      strtrim(mean_struct.files(fichier_courant,:)))) ;
027 fprintf('Loaded : %s\n',...
028      strtrim(mean_struct.files(fichier_courant,:)));
029 S_period = [];
030 involved_points = 0;
031
032 S=struct();
033 Phi=struct();
034
035 S.TS = mean_struct.periods(:,2);
036 S.mean = zeros(size(mean_struct.periods,1),1);
037 S.std = zeros(size(mean_struct.periods,1),1);
038 S.points = zeros(size(mean_struct.periods,1),1);
039
040 Phi.TS = mean_struct.periods(:,2);
041 Phi.mean = zeros(size(mean_struct.periods,1),1);
042 Phi.std = zeros(size(mean_struct.periods,1),1);
043
044
045 curseur_group = 1;
046 fichier_courant=0;
047 while (curseur_group<=size(mean_struct.groups,1))
048     %           curseur_group

```

```

049     periode_courante = mean_struct.groups(curseur_group,1);
050     while (curseur_group<=size(mean_struct.groups,1)) &&...
051             (mean_struct.groups(curseur_group,1)==periode_courante)
052
053     if mean_struct.groups(curseur_group,2) ~=fichier_courant
054         % Le fichier n'est pas encore chargé
055         fichier_courant = mean_struct.groups(curseur_group,2);
056         CurrentFile=load(...%
057                     sprintf('%s%s',mean_struct.folder,...%
058                     strtrim(mean_struct.files(fichier_courant,:)))) ;
059     end
060
061     S_temp = double(CurrentFile.sop(...%
062                     mean_struct.groups(curseur_group,3):...%
063                     mean_struct.groups(curseur_group,4),1:3));
064     involved_points = involved_points+size(S_temp,1);
065     S_period((end+1):(end+size(S_temp,1)),1:3)=S_temp;
066     curseur_group = curseur_group+1;
067
068 end
069 if ~isempty(S_period)
070     for i=1:size(S_period,2)
071         S_period(S_period(:,i) < -1,i) = -1;
072         S_period(S_period(:,i) > +1,i) = +1;
073     end
074     mean_temp=mean(S_period);
075     std_temp=std(S_period);
076
077     S.mean(periode_courante,1:3) = mean_temp;
078     S.std(periode_courante,1:3) = std_temp;
079     S.points(periode_courante,1) = involved_points;
080
081     Phi.mean(periode_courante,1) = mean(acos(S_period(:,1)));
082     Phi.std(periode_courante,1) = std(acos(S_period(:,1)));
083     Phi.points(periode_courante,1) = involved_points;
084 end
085 S_period= [] ;

```

```
086     involved_points = 0;  
087  
088 end  
089  
090 output_struct.folder = mean_struct.folder;  
091 output_struct.files = mean_struct.files;  
092 output_struct.mean_struct = mean_struct;  
093 output_struct.S = S;  
094 output_struct.Phi = Phi;  
095 end
```

ANNEXE XI

SEGMENTATION TEMPORELLE DES MESURES DU POLARIMÈTRE

Le Script-A XI-1 assure la segmentation temporelle des mesures du polarimètre avant de procéder au calcul de la valeur moyenne des paramètres de Stokes.

Entrée

Ce script prend cinq paramètres :

- **tsColumn**, contenant l'ensemble des valeurs d'horodatage des fichiers de mesure du polarimètre à moyenner;
- **initColumn**, contenant l'instant de fin de la période transitoire de mesure pour chaque fichier;
- **lengthFile**, le nombre de point de mesure par fichier du polarimètre (2×10^5 points);
- **periodInstrument**, la période d'échantillonnage de l'instrument;
- **periodGroup**, la durée sur laquelle on souhaite moyenner les paramètres.

Sortie

Le script retourne deux matrices :

- **periods**, contenant pour chaque intervalle de longueur *PeriodGroup* l'instant de début et de fin d'intervalle;
- **groups**, contenant pour chaque intervalle défini précédemment les fichiers et les points de mesure qui ont permis de calculer la moyenne des paramètres.

Script-A XI-1 Segmentation temporelle des mesures du polarimètre

```
001 function [ periods, groups ] = segmentation_temporelle_pola( ...
002     TsColumn, InitColumn, LengthFile, PeriodInstrument, PeriodGroup )
003
004 info=0;
005 verbose=0;
```

```

006
007 [TsColumn, signature] = sort(TsColumn);
008 InitColumn = InitColumn(signature);
009
010 nMeas = size(TsColumn, 1);
011 groups = [];
012
013 file_set = setdiff(1:nMeas, find(TsColumn==0));
014 % Indice des fichiers traites (excluant dont les erreurs de lecture)
015 TsColumn = TsColumn(file_set);
016 signature = signature(file_set);
017 InitColumn = InitColumn(file_set);
018 nFiles = size(TsColumn,1);
019
020 file_Id = 1;
021 initialisation = InitColumn(file_Id);
022 TS_begin_file=TsColumn(file_Id)+(initialisation-1)*PeriodInstrument;
023 TS_end_file = TsColumn(file_Id)+(LengthFile-1)*PeriodInstrument;
024
025 TS_begin_period = TsColumn(1)+(initialisation-1)*PeriodInstrument;
026 num_periods=ceil((TsColumn(end) ...
027 +LengthFile*PeriodInstrument-TS_begin_period)/PeriodGroup);
028
029 if info
030     fprintf('%d periodes\n',num_periods); %#ok<*UNRCH>
031 end
032
033 periods=zeros(num_periods,2);
034 for i = 1:num_periods
035     periods(i,:)=[TS_begin_period+(i-1)*...
036                 PeriodGroup TS_begin_period+i*PeriodGroup];
037 end
038
039 for period_Id = 1:num_periods
040     if info&&(mod(period_Id,10000)==0)
041         fprintf('periodes actuelle : %4.2f\n',...

```

```

043             period_Id/num_periods*100);
044         end
045
046         TS_begin_subperiod = periods(period_Id,1);
047         TS_end_subperiod = periods(period_Id,2);
048
049         skip_periode=0;
050
051         while (~ (TS_begin_file>TS_end_subperiod))&&(file_Id<=nFiles)&&...
052             (~skip_periode)
053             if verbose
054                 fprintf('New loop\n');
055             end
056
057             % Cas 1 : la période se trouve/finit avant le fichier
058             if (TS_begin_file>TS_end_subperiod)
059                 if verbose
060                     fprintf('Cas 1\n');
061                 end
062                 % Nouvelle période !
063                 skip_periode = 1;
064             end
065
066             % Cas 2 : la période est à cheval en début de fichier
067             if (TS_begin_subperiod<TS_begin_file)&&...
068                 (TS_end_subperiod>TS_begin_file)
069                 if verbose
070                     fprintf('Cas 2\n');
071                 end
072                 fin_locale = round((TS_end_subperiod-TS_begin_file)/...
073                     PeriodInstrument);
074                 groups(end+1,:)=[period_Id signature(file_Id),...
075                     initialisation, initialisation+fin_locale 2];
076                 % Nouvelle période
077                 skip_periode = 1;
078             end
079

```

```

080      % Cas 3 : la période est incluse dans le fichier (sens large)
081      if (TS_begin_subperiod>=TS_begin_file) &&...
082          (TS_end_subperiod<=TS_end_file)
083          if verbose
084              fprintf('Cas 3\n');
085          end
086          initialisation_locale=max(ceil((TS_begin_subperiod-...
087              TS_begin_file)/PeriodInstrument),1);
088          fin_locale=ceil((TS_end_file-TS_end_subperiod)/...
089              PeriodInstrument);
090          groups(end+1,:)=[period_Id signature(file_Id),...
091              initialisation_locale, ...
092              LengthFile-initialisation-fin_locale 3];
093
094      % Nouvelle période
095      skip_periode = 1;
096  end
097
098
099  if ((TS_begin_subperiod<TS_end_file)&&(TS_end_subperiod>...
100      TS_end_file))||(TS_begin_subperiod>TS_end_file))
101      % Cas 4 : la période est à cheval en fin de fichier
102      if ((TS_begin_subperiod<TS_end_file)&&...
103          (TS_end_subperiod>TS_end_file))
104          if verbose
105              fprintf('Cas 4\n');
106          end
107          initialisation_locale = round((TS_end_file-...
108              TS_begin_subperiod)/PeriodInstrument);
109          groups(end+1,:)=[period_Id, signature(file_Id), ...
110              LengthFile-initialisation-
111          initialisation_locale, ...
112              LengthFile, 4];
113      else
114          % Cas 4 : la période se trouve après le fichier
115          if verbose
116              fprintf('Cas 5\n');

```

```
117         end
118     end
119     file_Id=file_Id+1;
120     if file_Id<=nFiles
121         initialisation = InitColumn(file_Id);
122         TS_begin_file = TsColumn(file_Id)+...
123             (initialisation-1)*PeriodInstrument;
124         TS_end_file = TsColumn(file_Id)+...
125             (LengthFile-1)*PeriodInstrument;
126     end
127 end
128 end
129 end
130 end
```


ANNEXE XII

CALCUL ET TRACÉ DE LA FONCTION D'AUTOCORRÉLATION DES MESURES

Le Script-A XII-1 calcule dans un premier temps la fonction d'autocorrélation du paramètre demandé après avoir appliqué un filtre super gaussien d'ordre 4. La fonction d'autocorrélation est ensuite tracée.

Entrée

Le script prend en paramètre la structure de données *DataSetGlobal* obtenue après l'opération de moyennage des mesures à l'aide du Script-A X-1.

Script-A XII-1 Calcul et tracé de la fonction d'autocorrélation des mesures

```
001 function [] = compute_autocorrelation(DataSetGlobal)
002
003 plot_drift = 0;
004
005 nom_colonnes = {'inst_PMD','inst_PDL','PreFECBER','S1','Phi'};
006
007 parameter_id = 5;
008
009 units_of = 'min';
010
011 parameter = nom_colonnes{parameter_id};
012
013 switch parameter
014     case 'inst_PMD'
015         parameter_legend = 'DGD';
016     case 'Phi'
017         parameter_legend = '\phi';
018     otherwise
019         parameter_legend = parameter;
020 end
021
```

```

022 timestamp_vect = DataSetGlobal.(parameter).(units_of) (:,1);
023 dataset = DataSetGlobal.(parameter).(units_of) (:,2);
024
025 PAS=datenum(0,0,0,0,1,0);
026
027 new_timestamps = min(timestamp_vect):PAS:max(timestamp_vect);
028 dataset=interp1(timestamp_vect,dataset,new_timestamps)';
029 timestamp_vect = new_timestamps';
030
031 PMD_filtre = filtre_super_gaussien(dataset, 500);
032
033 ACF_PMD = xcov(PMD_filtre,floor(length(PMD_filtre)/2), 'biased');
034
035 % Normalization à 1...
036 ACF_PMD = ACF_PMD/var(PMD_filtre);
037
038 % Detection coherente des temps de correlations
039 correlation_time_fwhm = 0.5;
040
041 % DETECTION TEMPS CORRELATION PMD
042 [ACVmax,loc] = max(ACF_PMD);
043
044 for ind=loc:(length(ACF_PMD)-1)
045
046 if( ACF_PMD(ind,1)>correlation_time_fwhm &&...
047 ACF_PMD(ind+1,1)<correlation_time_fwhm )
048 ind_correlation_fwhm=ind-loc;
049 end
050 end
051
052 t_HWHM = (timestamp_vect(ind_correlation_fwhm)-...
053 min(timestamp_vect))*24*60;
054
055 TS_plot = timestamp_vect(1:end,1)-min(timestamp_vect(:,1));
056
057 plot_y = ACF_PMD(loc:end);
058 plot_x = TS_plot(1:(length(plot_y)));

```

```

059
060 id_max = 2*60;
061
062 fit_x = plot_x(1:id_max);
063 fit_y = plot_y(1:id_max);
064
065 fit_x(1)=1e-12;
066
067 drift_fun = @(p,t) (1-exp(-abs(t)/p))./(abs(t)/p);
068
069 options_NLM = statset('nlinfit');
070 options_NLM.Display = 'Iter';
071 options_NLM.TolFun = 1e-250;
072 options_NLM.TolX = 1e-250;
073 options_NLM.MaxFunEvals = 500000;
074 options_NLM.MaxIter = 500000 ;
075 options_NLM.FunValCheck = 'off' ;
076
077 NLM_lin_drift = NonLinearModel.fit(fit_x,fit_y,drift_fun, 1, ...
078      'Options',options_NLM);
079 t_drift = double(NLM_lin_drift.Coefficients(1,1));
080
081 drift_y = drift_fun(t_drift, fit_x);
082
083 figure('units','normalized','outerposition',[0 0 1 1],...
084      'visible','on');
085
086 k1 = plot(plot_x,ACF_PMD(loc:end));
087
088 hold on;
089 if plot_drift
090     k2 = plot(fit_x, drift_y, 'g', 'linewidth',2);
091 end
092 y_min = -0.2:0.05:1;
093 x_min = ones(length(y_min))*datenum(0,0,0,0,10,0); % 10-min
094 k4 = plot(x_min,y_min,'--');
095 set(k4,'color','r');

```

```
096 set(k4,'linewidth',2);
097 set(k4,'marker','none');
098 text(x_min(1)-1e-3,-.15,'10 minutes','Rotation',90,'FontName',...
099      'Calibri','FontSize',15)
100 hold on;
101
102 y_min = -0.2:0.05:1;
103 x_min = ones(length(y_min))*datenum(0,0,0,0,30,0); % 30-min
104 k5 = plot(x_min,y_min,'--');
105 set(k5,'color','r');
106 set(k5,'linewidth',2);
107 set(k5,'marker','none');
108 text(x_min(1)-3e-3,-.15,'30 minutes','Rotation',90,'FontName',...
109      'Calibri','FontSize',15)
110 hold on;
111
112 y_hour = -0.2:0.05:1;
113 x_hour = ones(length(y_hour))*datenum(0,0,0,1,0,0); % 1-hour
114 k6 = plot(x_hour,y_hour,'--');
115 set(k6,'color','r');
116 set(k6,'linewidth',2);
117 set(k6,'marker','none');
118 text(x_hour(1)-5e-3,-.15,'1 heure','Rotation',90,'FontName',...
119      'Calibri','FontSize',15)
120 hold on;
121
122 y_hour = -0.2:0.05:1;
123 x_hour = ones(length(y_hour))*datenum(0,0,1,0,0,0); % 1-jour
124 k6 = plot(x_hour,y_hour,'--');
125 set(k6,'color','r');
126 set(k6,'linewidth',2);
127 set(k6,'marker','none');
128 text(x_hour(1)-2e-1,.2,'1 jour','Rotation',90,'FontName',...
129      'Calibri','FontSize',15)
130 hold on;
131
132
```

```

133 set(gca,'XLim',[1e-3 plot_x(end,1)]);
134 set(gca,'XTick',[1e-3 1e-2 1e-1 1e0 1e1]);
135 set(gca,'XScale','log');
136
137 % M=k1;
138
139 legend_string_1 = strrep(...%
140     sprintf('\tau_HWHM = %2.0f minutes',t_HWHM),...
141     '.', ',');
142 legend_string_2 = strrep(...%
143     sprintf('t_{drift} = %2.0f minutes',t_drift*24*60),...
144     '.', ',');
145
146 if plot_drift
147     legend([k1 k2],legend_string_1,legend_string_2,...%
148         'Location',[.7 .75 .2 .15]);
149 else
150     legend(k1,legend_string_1,'Location',[.7 .75 .2 .15]);
151 end
152
153 set(gca,'FontSize',12,'FontName','Calibri');
154 xlabel('Temps de corrélation (jour)', 'FontName', 'Calibri',...
155     'FontSize', 24);
156 ylabel(sprintf('ACF de <%s>',parameter_legend),...
157     'FontName', 'Calibri' , 'FontSize', 24,'color','k');
158 set(gca,'YLim',[-0.2 1]);
159 set(gca,'YTick',(-0.3:0.1:1));
160 grid on;
161
162 hold off;
163
164 export_fig('figure.png','-transparent');
165 end
166
167 function [ filtered_data ] = filtre_super_gaussien( data, sigma )
168
169 FFT_data = fft(data,length(data));

```

```
170
171 x_pmd = 0:(length(data)-1);
172
173 Filtre_Super_Gauss = exp((-4)*(x_pmd/sigma).^4);
174
175 Filtre_Super_Gauss2 = transpose(Filtre_Super_Gauss);
176
177 Signal_filtre = zeros(length(data),1);
178
179 for i=1:length(data)
180     Signal_filtre(i,1) = FFT_data(i,1)*Filtre_Super_Gauss2(i,1);
181 end
182
183 IFFT_signal_filtre_pmd = ifft(Signal_filtre(:,1),length(data));
184
185 filtered_data = real(IFFT_signal_filtre_pmd);
186
187 end
```

ANNEXE XIII

COMPARAISON ENTRE LES MESURES DU TRANSPONDEUR COHÉRENT ET CELLES DU POLARIMÈTRE

Le Script-A XIII-1 permet de comparer les mesures du transpondeur cohérent et celles du transpondeur cohérent. La première étape est l'alignement des mesures des deux instruments, la seconde est le tracé de l'évolution temporelle des mesures par les deux instruments et la dernière est le tracé de la corrélation croisée entre les mesures des deux instruments.

Entrée

Le script prend en paramètre deux structures : *ocld* et *pola*, contenant le résultat du moyennage des paramètres réalisés respectivement avec le Script-A IV-1 et le Script-A X-1.

Script-A XIII-1 Comparaison entre les mesures du transpondeur cohérent et celles du transpondeur cohérent

```
001 function [] = comparison_ocld_pola(ocld, pola)
002
003 %% Chargement des donnees utiles au trace
004 dataset_ocld = ocld.Phi.sec(:,1:2);
005 dataset_pola = [pola.Phi.TS(pola.Phi.mean>0), ...
006     pola.Phi.mean(pola.Phi.mean>0)];
007
008 PAS_tgrid = datenum(0,0,0,0,0,1);
009
010 %% Calibration des temps de debut et de fin
011 start_hour_temp = max([min(dataset_ocld(:,1)), ...
012     min(dataset_pola(:,1))]);
013
014 end_hour_temp = min([max(dataset_ocld(:,1)), ...
015     max(dataset_pola(:,1))]);
016
017 offset = datenum(0,0,0,4,0,0);
018
```

```

019 nb_intervals = floor((end_hour_temp-start_hour_temp)/PAS_tgrid);
020 new_interval =(start_hour_temp:PAS_tgrid:...
021     (start_hour_temp+nb_intervals*PAS_tgrid))';
022
023 dataset_ocld_rect =...
024     interp1(dataset_ocld(:,1),dataset_ocld(:,2),new_interval);
025 dataset_pola_rect =...
026     interp1(dataset_pola(:,1),dataset_pola(:,2),new_interval);
027
028 dataset_merge = [new_interval dataset_ocld_rect dataset_pola_rect];
029 fprintf('from %s\n to %s\n',...
030     datestr(start_hour_temp,'mmmm dd, yyyy HH:MM:SS.FFF AM'),...
031     datestr(end_hour_temp,'mmmm dd, yyyy HH:MM:SS.FFF AM'));
032
033 %% Filtrage du trace (moyenne glissante)
034 dataset_merge_time(:,1)=dataset_merge(:,1);
035 windowSize = 60;
036 dataset_merge_time(:,2)=...
037     filter(ones(1,windowSize)/windowSize,1,dataset_merge(:,2));
038 dataset_merge_time(:,3)=...
039     filter(ones(1,windowSize)/windowSize,1,dataset_merge(:,3));
040
041 difference = dataset_merge_time(:,2)-dataset_merge_time(:,3);
042 ratio = dataset_merge_time(:,2)./dataset_merge_time(:,3);
043
044
045 %% Definition du pas de grille
046 PAS = datenum(0,0,0,12,0,0);
047
048 fig1 = figure('units','normalized',...
049     'outerposition',[0 0 1 1],'visible','on');
050
051 % [left bottom width height]
052
053 ax1 = subplot('Position',[.12 .625 .83 .325]);
054
055 plot(dataset_merge_time(:,1),dataset_merge_time(:,2),'Color','r');

```

```

056
057 grid on;
058
059 set(gca,'XLim',[start_hour_temp end_hour_temp]);
060 set(gca,'XTick',start_hour_temp:PAS:end_hour_temp);
061 set(gca,'XTickLabel',[])
062
063 set(gca,'YLim',[-.1 pi+.1]);
064 set(gca,'FontName','Symbol','fontsize',14);
065 set(gca,'YTick',0:pi/4:pi);
066 set(gca,'FontName','Symbol')
067 legend=cell(0);
068 legend{1}='0';
069 legend{2}='';
070 legend{3}='p/2';
071 legend{4}='';
072 legend{5}='p';
073 set(gca,'YTickLabel',legend)
074 ylabh1 = ylabel('\phi_{OCLD} (rad)', 'FontName', 'Calibri', ...
075     'FontSize', 14, 'FontWeight', 'bold', 'color', 'k');
076 position1=get(ylabh1,'Position');
077 set(ylabh1,'Position',[start_hour_temp-offset, ...
078     position1(2), position1(3)]);
079
080 ax2 = subplot('Position',[0.12 .25 .83 .325]);
081
082 plot(dataset_merge_time(:,1),dataset_merge_time(:,3), 'Color', 'b');
083
084 grid on;
085
086 set(gca,'XLim',[start_hour_temp end_hour_temp]);
087 set(gca,'XTick',start_hour_temp:PAS:end_hour_temp);
088 set(gca,'XTickLabel',[])
089
090 set(gca,'YLim',[-.1 pi+.1]);
091 set(gca,'FontName','Symbol','fontsize',14);
092 set(gca,'YTick',0:pi/4:pi);

```

```

093 set(gca,'FontName','Symbol')
094 legend=cell(0);
095 legend{1}='0';
096 legend{2}='';
097 legend{3}='p/2';
098 legend{4}='';
099 legend{5}='p';
100 set(gca,'YTickLabel',legend)
101 ylabh2=ylabel('\phi_{polarimeter} (rad)', 'FontName', 'Calibri', ...
102     'FontSize', 14, 'FontWeight', 'bold', 'color', 'k');
103 position2=get(ylabh2,'Position');
104 set(ylabh2,'Position',[start_hour_temp-offset, ...
105     position2(2), position2(3)]);
106
107 set(gca,'XLim',[start_hour_temp end_hour_temp]);
108 set(gca,'XTick',start_hour_temp:PAS:end_hour_temp);
109 set(gca,'XTickLabel',[])
110 XTick = get(gca,'XTick');
111 XTick = XTick(:);
112 for k=1:length(XTick)
113     t=text(XTick(k),-.1, ...
114         [datestr(start_hour_temp+(k-1)*PAS,'ddd mm/dd','local'), ...
115             ' ', ...
116             datestr(start_hour_temp+(k-1)*PAS,'HH','local'),'h']);
117     set(t,'fontsize',14,'fontname','calibri',...
118         'HorizontalAlignment','right','VerticalAlignment','top',...
119         'Rotation',45);
120 end
121
122 linkaxes([ax1,ax2], 'x')
123
124 dataset_merge_cov(:,1) = dataset_merge(:,1);
125 dataset_merge_cov(:,2)=filtre_super_gaussien(dataset_merge(:,2),500);
126 dataset_merge_cov(:,3)=filtre_super_gaussien(dataset_merge(:,3),500);
127
128 [correlation_croisee, lags]=xcov(dataset_merge_cov(:,2), ...
129     dataset_merge_cov(:,3), 'biased');

```

```
130 correlation_croisee=correlation_croisee/...
131     (var(dataset_merge_cov(:,2))*var(dataset_merge_cov(:,3)));
132 [maximum, indice]=max(correlation_croisee);
133 lags(indice);
134
135 fig2 = figure('units','normalized',...
136     'outerposition',[0 0 1 1],'visible','on');
137
138 plot(lags/3600,correlation_croisee)
139 ylabel('Amplitude','FontName', 'Calibri',...
140     'FontSize', 14,'FontWeight','bold','color','k');
141 xlabel('Décalage temporel (heure)','FontName', 'Calibri',...
142     'FontSize', 14,'FontWeight','bold','color','k');
143 set(gca,'FontSize',16)
144 xlim([0 max(lags/3600)]);
145 set(gca,'XTick',0:12:floor(max(lags/3600)))
146 grid on;
147 end
```


ANNEXE XIV

TRACÉ SUR UN MÊME GRAPHIQUE DE L'ÉVOLUTION DE LA TEMPÉRATURE ET D'UN PARAMÈTRE MESURÉ

Le Script-A XIV-1 effectue sur un même graphique le tracé de l'évolution temporelle de la température et d'un paramètre mesuré.

Entrée

Ce script prend deux paramètres :

- **dataset_time**, matrice contenant les valeurs d'horodatage des différentes mesures à tracer;
- **dataset_y**, un matrice contenant les valeurs du paramètre mesuré aux instants indiqués dans *dataset_time*;
- **WeatherData**, contenant la structure de données retournée par le Script-A XV-1.

Script-A XIV-1 Tracé sur un même graphique de l'évolution de la température de la température te d'un paramètre mesuré

```
001 function [] =...
002     plot_temperature_parameter(dataset_time, dataset_y, WeatherData)
003
004 start_hour_temp = min(dataset_time);
005 end_hour_temp = max(dataset_time);
006
007 % Preparation des donnees meteo
008 abscisse_meteo=start_hour_temp:datenum(0,0,0,0,1,0):end_hour_temp;
009 mesures_meteo = get_weather_subset...
010     (WeatherData.data,start_hour_temp,end_hour_temp);
011 moyenne_temperature_global =...
012     zeros(length(abscisse_meteo),length(mesures_meteo));
013 for i=1:length(mesures_meteo)
014     moyenne_temperature_global(:,i)=interp1...
015         mesures_meteo(i).TS, ...
016         mesures_meteo(i).temperature,abscisse_meteo)';
```

```

017 end
018 moyenne_temperature = mean(moyenne_temperature_global,2);
019
020 C = linspecer(2);
021
022 figure('units','normalized',...
023     'outerposition',[0 0 1 1], 'visible','on');
024 hold on
025
026 [ax,h1,h2] = plotyy(dataset_time,...
027     dataset_y, abscisse_meteo, moyenne_temperature);
028
029 set(h1,'linewidth',1,'color',C(1,:))
030 set(ax(1), 'ycolor',C(1,:))
031
032 set(h2,'linewidth',2,'color',C(2,:))
033 set(ax(2), 'ycolor',C(2,:))
034
035 plot(dataset_time,dataset_y);
036
037 grid on
038
039 PAS = datenum(0,0,0,6,0,0); % Pas des graduation de l'axe du temps
040
041 set(ax(1), 'XLim',[start_hour_temp end_hour_temp]);
042 set(ax(1), 'XTick',start_hour_temp:PAS:end_hour_temp);
043 axes(ax(1)); ylabel('< \phi >', 'FontSize', 14);
044
045 set(ax(2), 'XLim',[start_hour_temp end_hour_temp]);
046 set(ax(2), 'XTick',start_hour_temp:PAS:end_hour_temp);
047 axes(ax(2)); ylabel('Temperature (°C)', 'FontSize', 14);
048
049 XTick = get(ax(1), 'XTick');
050 XTick = XTick(:);
051 for k=2:length(XTick)
052     l=text(XTick(k),1e-6,...
053         [datestr(start_hour_temp+(k-1)*PAS, 'mm/dd'), ' ', ...

```

```
054         datestr(start_hour_temp+(k-1)*PAS,'HH'), 'h'], ...
055         'fontsize',12,'fontname','calibri');
056     set(l,'HorizontalAlignment','right',...
057         'VerticalAlignment','top', 'Rotation',45);
058 end
059
060 set(ax(1), 'XTickLabel', []);
061 set(ax(2), 'XTickLabel', []);
062 end
063
064 function [ weather_struct_out ] =...
065     get_weather_subset( weather_struct, start_time, end_time )
066
067 weather_struct_out = struct();
068
069 for i=1:length(weather_struct)
070
071     weather_struct_out(i,1).station_code =...
072         weather_struct(i).station_code;
073     id_temp = intersect(...
074         find(weather_struct(i).TS>=start_time),...
075         find(weather_struct(i).TS<=end_time));
076     weather_struct_out(i,1).TS = weather_struct(i).TS(id_temp);
077     weather_struct_out(i,1).temperature =...
078         weather_struct(i).temperature(id_temp);
079
080 end
081
082 end
```


ANNEXE XV

CONVERSION DES FICHIERS CSV DU NOAA AU FORMAT MATLAB

Le Script-A XV-1 assure la conversion des fichiers CSV du NOAA en format MATLAB.

Les fichiers bruts du NOAA sont obtenus à partir du serveur FTP de l'institution et sont ensuite converties en CSV avec le script "ishJava.java" fourni par le NOAA à la même adresse. Adresse du serveur : <ftp://ftp.ncdc.noaa.gov/pub/data/noaa/>

Entrée

Ce script prend deux paramètres :

- **chemin_dossier**, chaîne de caractères contenant le chemin du dossier contenant les fichiers (CSV) à traiter;
- **gmt_offset**, valeur entière correspondant au décalage horaire entre l'heure GMT et le fuseau horaire du lieu dont les mesures proviennent (les mesures du NOAA sont datées en GMT).

Sortie

Le script retourne une structure de données **mesures_meteo** dont le nombre d'éléments est égal au nombre de fichiers contenus dans le dossier *chemin_dossier*. Chacun de ces éléments contient trois variables :

- **station_code**, contenant le code (propre au NOAA) de la station météorologique de mesure;
- **ts**, un vecteur Matlab contenant les valeurs d'horodatage des différentes mesures contenues dans les fichiers du NOAA;
- **temperature**, un vecteur Matlab contenant les mesures de température réalisées à chaque instant décrit dans le vecteur *TS*.

Script-A XV-1 Conversion des fichiers CSV du NOAA en format MATLAB

```
036     measures_meteo(i,1).temperature =...
037         celsius_from_fahrenheit(temperature_temp(:,2));
038
039 end
040
041 end
042
043 function [ content, nFiles ] = list_files( ext, folder, output )
044
045 filter=strcat('*.',ext);
046 dirContents = dir([folder filter]);
047 nFiles = length(dirContents);
048 content=cell(0);
049 switch output
050     case 'short'
051         for i=1:nFiles
052             content{i}=dirContents(i).name;
053         end
054     otherwise
055         for i=1:nFiles
056             content{i}=strcat(folder,dirContents(i).name);
057         end
058     end
059 if max(size(content))>0, content=char(content); end;
060 end
```


ANNEXE XVI

TRACÉ DE L'HISTOGRAMME DES VITESSES DE ROTATION DU VECTEUR DE STOKES

Le Script-A XVI-1 permet de tracer l'histogramme des vitesses de rotation du vecteur de Stokes.

Entrée

Le script prend en paramètre la structure de données obtenues à l'aide du Script-A VI-1.

Script-A XVI-1 Tracé de l'histogramme des vitesses de rotation du vecteur de Stokes

```
001 function [] = plot_hist_arr(FileData)
002
003 stats_array = zeros(size(FileData.mean_ARR,1),5);
004 stats_array(:,2)=FileData.mean_ARR;
005 stats_array(:,3)=FileData.std_ARR;
006 stats_array(:,4)=FileData.glitch_tab(:,1);
007 stats_array(:,5)=2e5*ones(size(FileData.mean_ARR,1),1);
008 [ mean_ARR, std_ARR ] = get_global_stats_pola( stats_array );
009
010 raw_data=struct();
011 raw_data.x=FileData.parametres.Bins';
012 raw_data.y_abs=FileData.hist_ARR;
013 raw_data.y_rel = raw_data.y_abs/sum(raw_data.y_abs);
014 raw_data.y_pdf = raw_data.y_rel/trapz(raw_data.x,raw_data.y_rel);
015 id_pos = find(raw_data.y_abs>0);
016
017 y_min_power = floor(log10(min(raw_data.y_pdf(id_pos))));
018 y_max_power = ceil(log10(max(raw_data.y_pdf(id_pos))));
019 y_min = 10^y_min_power;
020 y_max = 10^y_max_power;
021 x_min = 0;
022 x_max = ceil(max(raw_data.x(id_pos))));
```

```

023 x_lim_global = [x_min x_max];
024 y_lim_global = [y_min/10 y_max];
025
026 y_tick_global =...
027     logspace(y_min_power,y_max_power,y_max_power-y_min_power+1);
028
029 legend_string = sprintf('$\bar{x} = $%6.4f rad/ms, $\sigma = $%6.4f
030 rad/ms',...
031 mean_ARR, ...
032 std_ARR);
033
034 label_x = 'Vitesse de rotation du vecteur de Stokes (rad/ms)';
035 label_y = 'Distribution de probabilité normalisée';
036
037 %% Affichage de l'histogramme seul
038 figure('units','normalized','outerposition',[0 0 1 1],...
039 'visible','on');
040
041 bar(raw_data.x(id_pos),raw_data.y_pdf(id_pos),'b',...
042 'BaseValue',y_min/10);
043
044 l = legend(legend_string);
045
046 set(l,'Interpreter','latex','FontName','Calibri','FontSize',12,...
047 'position',[.45 .8 .4 .1]);
048
049 xlabel(label_x,'FontName','Calibri','fontsize',16,'fontweight','b');
050 ylabel(label_y,'FontName','Calibri','fontsize',16,'fontweight','b');
051
052 xlim(x_lim_global);
053 ylim(y_lim_global);
054
055 set(gca,'YTick',y_tick_global);
056 set(gca,'YScale','log');
057 set(gca,'YMinorTick','on');
058 set(gca,'XTick',0:1:max(raw_data.x(id_pos)));
059 set(gca,'FontSize',15,'FontName','Calibri');
```

```
060
061  grid on;
062
```


ANNEXE XVII

AJUSTEMENT DE L'HISTOGRAMME DES VITESSES DE ROTATION DU VECTEUR DE STOKES À UNE DISTRIBUTION DE RAYLEIGH ET À UNE FONCTION EXPONENTIELLE

Le Script-A XVII-1 permet d'ajuster l'histogramme des vitesses de rotation du vecteur de Stokes à deux distributions.

Entrée

Ce script prend en paramètre la structure de données obtenue à l'aide du Script-A VI-1.

Script-A XVII-1 Ajustement de l'histogramme des vitesses de rotation du vecteur de stokes à une distribution de Rayleigh et à une fonction exponentielle

```
001 function [] = fit_hist_arr(FileData)
002
003 stats_array = zeros(size(FileData.mean_ARR,1),5);
004 stats_array(:,2)=FileData.mean_ARR;
005 stats_array(:,3)=FileData.std_ARR;
006 stats_array(:,4)=FileData.glitch_tab(:,1);
007 stats_array(:,5)=2e5*ones(size(FileData.mean_ARR,1),1);
008 [ mean_ARR, std_ARR ] = get_global_stats_pola( stats_array );
009
010 raw_data=struct();
011 raw_data.x=FileData.parametres.Bins';
012 raw_data.y_abs=FileData.hist_ARR;
013 raw_data.y_rel = raw_data.y_abs/sum(raw_data.y_abs);
014 raw_data.y_pdf = raw_data.y_rel/trapz(raw_data.x,raw_data.y_rel);
015 id_pos = find(raw_data.y_abs>0);
016
017 y_min_power = floor(log10(min(raw_data.y_pdf(id_pos)))); 
018 y_max_power = ceil(log10(max(raw_data.y_pdf(id_pos)))); 
019 y_min = 10^y_min_power;
020 y_max = 10^y_max_power;
021 x_min = 0;
```

```

022 x_max = ceil(max(raw_data.x(id_pos)));
023
024 x_lim_global = [x_min x_max];
025 y_lim_global = [y_min/10 y_max];
026
027 y_tick_global = ...
028 logspace(y_min_power,y_max_power,y_max_power-y_min_power+1);
029
030 legend_string = sprintf('$\bar{x} = $%6.4f rad/ms, $\sigma = $%6.4f
031 rad/ms',...
032 mean_ARR, ...
033 std_ARR);
034
035 label_x = 'Vitesse de rotation du vecteur de Stokes (rad/ms)';
036 label_y = 'Distribution de probabilité normalisée';
037
038 %% Calcul de l'ajustement de Rayleigh de l'histogramme
039 modelFun_fG_linear = @(sigma,x) ...
040 (1/sigma^2) * x.* exp(-(x.^2)/(2*sigma^2));
041 sigma_th = .1;
042
043 % Extraction des données utiles au fit
044 data_fit = [(raw_data.x)', raw_data.y_pdf];
045
046 % Calcul du fit
047 options_NLM = statset('nlinfit');
048 options_NLM.Display = 'Iter';
049 options_NLM.TolFun = 1e-250;
050 options_NLM.TolX = 1e-250;
051 options_NLM.MaxFunEvals = 500000;
052 options_NLM.MaxIter = 500000 ;
053 options_NLM.FunValCheck = 'off' ;
054
055 NLM_lin_rayleigh = NonLinearModel.fit(...%
056 data_fit(id_pos,1),data_fit(id_pos,2),...
057 modelFun_fG_linear, sigma_th, ...
058 'Options',options_NLM);

```

```
059 sigma_final_linear = double(NLM_lin_rayleigh.Coefficients(1,1));
060
061 % Extraction des coefficients d'incertitudes
062 CI_fG = coefCI(NLM_lin_rayleigh);
063 sigma_low = CI_fG(1);
064 sigma_high = CI_fG(2);
065
066 % Calcul des valeurs pour le fit, le fit par defaut et le fit par
067 exces
068 fit_rayleigh_lin =
069 modelFun_fG_linear(sigma_final_linear,(raw_data.x)');
070 fit_rayleigh_lin_low = modelFun_fG_linear(sigma_low,(raw_data.x)');
071 fit_rayleigh_lin_high = modelFun_fG_linear(sigma_high,(raw_data.x)');
072
073 % Calcul du coefficient de Pearson de l'échantillon
074 numerator = sum((data_fit(:,2)-mean(data_fit(:,2))).*...
075     (fit_rayleigh_lin(:,1)-mean(fit_rayleigh_lin(:,1))));
076 denominator = sqrt(sum((data_fit(:,2)-mean(data_fit(:,2))).^2)).*...
077     sqrt(sum((fit_rayleigh_lin(:,1)-
078 mean(fit_rayleigh_lin(:,1))).^2));
079 Pearson_rayleigh = (numerator/denominator);
080
081 %% Affichage de l'histogramme accompagné de l'ajustement de Rayleigh
082 figure('units','normalized','outerposition',[0 0 1 1],...
083     'visible','on');
084
085 bar(raw_data.x(id_pos),raw_data.y_pdf(id_pos),'b',...
086     'BaseValue',1e-10);
087
088 hold on;
089
090 h2 = plot((raw_data.x)', fit_rayleigh_lin(:,1));
091 set(h2,'color','r');
092 set(h2,'linewidth',2);
093 set(h2,'marker','none');
094 hold on
095
```

```

096 h3 = plot((raw_data.x)', fit_rayleigh_lin_low(:,1), '--');
097 set(h3, 'color','r');
098 set(h3, 'linewidth',2);
099 set(h3, 'marker','none');
100 hold on
101
102 h4 = plot((raw_data.x)', fit_rayleigh_lin_high(:,1), '--');
103 set(h4, 'color','r');
104 set(h4, 'linewidth',2);
105 set(h4, 'marker','none');
106 hold on
107
108 legend_string_2 = sprintf('\\textbf{Ajustement de Rayleigh} : 
109 $\\frac{x}{\\alpha^2}e^{-\\frac{x^2}{2\\alpha^2}}$\\n$\\alpha=%6.4f
110 rad/ms',sigma_final_linear);
111
112 legend_string_3 = sprintf('\\textbf{Intervalles de confiance
113 (95\%)}\\n$\\alpha_{inf}=%6.4f rad/ms, $\\alpha_{sup}=%6.4f
114 rad/ms\\n\\n\\textbf{Coefficient de Pearson} $R_p=%6.4f',...
115 sigma_low,sigma_high,Pearson_rayleigh);
116
117 legend(legend_string,legend_string_2,legend_string_3)
118
119 set(legend, 'Interpreter','latex','FontName','Calibri',...
120      'FontSize',16,'Position',[.45 .55 .45 .35]);
121
122 xlabel(label_x,'FontName','Calibri','fontsize',16,'fontweight','b');
123 ylabel(label_y,'FontName','Calibri','fontsize',16,'fontweight','b');
124
125 xlim(x_lim_global);
126 ylim(y_lim_global);
127
128
129 set(gca,'YTick',y_tick_global);
130 set(gca,'YScale','log');
131 set(gca,'YMinorTick','on');
132 set(gca,'XTick',0:1:max((raw_data.x(id_pos))));

```

```
133 set(gca, 'FontSize',15, 'FontName', 'Calibri');
134
135 grid on;
136
137 %% Calcul de l'ajustement Exponentiel de l'histogramme
138 rate_begin_tail = 2;
139 coeff_bound = 95;
140
141 indice_fit=find(raw_data.x>=rate_begin_tail,1);
142 x_fit_left=raw_data.x(1:(indice_fit-1))';
143 y_fit_left=raw_data.y_pdf(1:(indice_fit-1));
144 x_fit=raw_data.x(indice_fit:end)';
145 y_fit=raw_data.y_pdf(indice_fit:end);
146 x_fit_full=[x_fit_left; x_fit];
147
148 modelFun_exp1 = @(p,x) p(1)*exp(p(2)*x);
149
150 ft = fittype('exp1');
151 opts = fitoptions( ft );
152 opts.Algorithm = 'Levenberg-Marquardt';
153 opts.Display = 'Iter';
154 opts.TolX = 1e-250;
155 opts.TolFun = 1e-250;
156 opts.MaxFunEvals = 500000;
157 opts.MaxIter = 500000 ;
158
159 f2 = fit(x_fit,y_fit,ft,opts);
160
161 coefExp2=coeffvalues(f2);
162 coefCiExp2 = confint(f2,coeff_bound/100);
163 coefCiExp2min=coefCiExp2(1,:);
164 coefCiExp2max=coefCiExp2(2,:);
165
166 fit_exp = modelFun_exp1(coefExp2,x_fit);
167
168 % Calcul du coefficient de Pearson de l'echantillon
169 numerator = sum((y_fit(:,1)-mean(y_fit(:,1))).*...
```

```

170      (fit_exp(:,1) - mean(fit_exp(:,1)))) ;
171 denominator = sqrt(sum((y_fit(:,1) - mean(y_fit(:,1))).^2)).*...
172      sqrt(sum((fit_exp(:,1) - mean(fit_exp(:,1))).^2));
173 Pearson_exp = (numerator/denominator);
174
175
176 legend_string_1 = sprintf('$\bar{x} = \%6.4f \text{ rad/ms}, \sigma = '
177 '$ \%6.4f \text{ rad/ms}', ...
178     mean_ARR, ...
179     std_ARR);
180 legend_string_2=sprintf(' \textbf{Ajustement Exponentiel} : '
181 '$y(x)=\alpha e^{\beta x}$ $\alpha=%4.2d,$
182 '$\beta=%4.2d$',coefExp2(1),coefExp2(2));
183 legend_string_3=sprintf(' \textbf{Intervalles de confiance (95\%)} '
184 ': $\alpha_{\min}=%4.2d,$
185 '$\alpha_{\max}=%4.2d$ $\beta_{\min}=%4.2d,$
186 '$\beta_{\max}=%4.2d$ \n \textbf{Coefficient de Pearson}'
187 '$R_p=%6.4f', ...
188     coefCiExp2min(1), ...
189     coefCiExp2min(2), ...
190     coefCiExp2max(1), ...
191     coefCiExp2max(2), ...
192     Pearson_exp);
193
194 y_min_global = ...
195     10^(floor(log10(min(raw_data.y_pdf(raw_data.y_pdf>0)))))-1;
196
197 %% Affichage de l'histogramme accompagné de l'ajustement Exponentiel
198 figure('units','normalized','outerposition',[0 0 1 1],...
199     'visible','on');
200 hold on
201 grid on
202 bar(x_fit_left,y_fit_left,'BaseValue',y_min_global,...
203     'FaceColor',[.75,.75,.75],'EdgeColor',[.75,.75,.75]);
204 h1 = bar(x_fit,y_fit,'BaseValue',y_min_global,...
205     'FaceColor',[0,0,1],'EdgeColor',[0,0,1]);
206 h2 = plot(x_fit_full,modelFun_exp1(coefExp2,x_fit_full),...

```

```
207      'r','LineWidth',2);
208 h3 = plot(x_fit_full,modelFun_exp1(coefCiExp2min,x_fit_full),...
209      '--r','LineWidth',2);
210 plot(x_fit_full,modelFun_exp1(coefCiExp2max,x_fit_full),...
211      '--r','LineWidth',2);
212 M=[h1 h2 h3];
213
214 legend(M,legend_string_1,legend_string_2,legend_string_3);
215 set(legend,'Interpreter','latex','FontName','Calibri',...
216      'FontSize',14,'Position',[.39 .5 .5 .38]);
217
218 xlabel(label_x,'FontName','Calibri','fontsize',16,'fontweight','b');
219 ylabel(label_y,'FontName','Calibri','fontsize',16,'fontweight','b');
220
221 xlim(x_lim_global);
222 ylim(y_lim_global);
223
224 set(gca,'YTick',y_tick_global);
225 set(gca,'YScale','log');
226 set(gca,'YMinorTick','on');
227 set(gca,'XTick',0:1:max(raw_data.x(id_pos)));
228 set(gca,'FontSize',15,'FontName','Calibri');
229 grid on;
```


ANNEXE XVIII

CARACTÉRISATION DE LA DISTRIBUTION DU BRUIT DU POLARIMÈTRE

Le Script-A XVIII-1 permet de caractériser la distribution du bruit du polarimètre.

Entrée

Ce script prend deux paramètres obligatoires :

- **folder**, chaîne de caractères contenant le chemin du dossier contenant les fichiers à traiter;
- **fileList**, contenant la liste des fichiers à traiter (sans chemin absolu). Cette liste doit être de type *Character Array*.

Ce script accepte d'autres paramètres facultatifs :

- **bins**, pour définir manuellement la largeur des colonnes de l'histogramme des vitesses de rotation du vecteur de Stokes;
- **filterFrequencies**, matrice à une dimension contenant la liste des fréquences de filtre passe haut à utiliser pour le traitement.

Sortie

Le script retourne une variable, **resultats**, qui est une structure Matlab contenant contenant neuf variables :

- **output_struct_stats.TS**, matrice à une dimension contenant la valeur d'horodatage de l'ensemble des fichiers traités. Ce vecteur a autant d'éléments que de fichiers traités;
- **output_struct_stats.file_error_list**, variable de type Cell contenant la liste des fichiers corrompus (c'est-à-dire qui n'ont pas pu être ouverts par le script);
- **output_struct_stats.mean_S**, matrice à une dimension contenant la valeur moyenne de chaque composante Stokes par fichier;
- **output_struct_stats.std_S**, matrice à une dimension contenant l'écart type de chaque composante Stokes par fichier;

- **output_struct_stats.mean_Snorm**, matrice à une dimension contenant la valeur moyenne de chaque composante Stokes normalisée par fichier;
- **output_struct_stats.std_Snorm**, matrice à une dimension contenant l'écart type de chaque composante Stokes normalisée par fichier;
- **output_struct_stats.hist_S**, matrice à une dimension contenant les valeurs de l'histogramme des valeurs des trois composantes de Stokes pour chaque mesure du polarimètre. Les colonnes correspondent à la valeur du vecteur *Bins* donné comme paramètre àu script;
- **output_struct_stats.hist_Snorm**, matrice à une dimension contenant les valeurs de l'histogramme des valeurs des trois composantes de Stokes normaliséespour chaque mesure du polarimètre. Les colonnes correspondent à la valeur du vecteur *Bins* donné comme paramètre àu script.

Dépendances

Le script dépend de la fonction *partitionnement_parallel*, présentée au Script-A V-1.

Script-A XVIII-1 Caractérisation de la distribution du bruit du polarimètre

```

001 function [resultat] = characterization_noise_pola(varargin)
002
003 p = inputParser;
004 p.addRequired('Folder', @isstr);
005 p.addRequired('FileList', @ischar);
006 p.addValue('Bins', -1:.0001:1,@(x) isreal(x) &&...
007     issorted(x) && x(1)<x(end));
008 p.addValue('FilterFrequencies', [.1,.5, 1, 2, 5], @isreal);
009 p.parse(varargin{:});
010 opts = p.Results; % Structure of input options
011
012 folder = opts.Folder;
013 file_list = opts.FileList;
014 Bins = opts.Bins;
015 filter_frequencies = opts.FilterFrequencies;

```

```
016
017 nbWorker = matlabpool('size');
018 if nbWorker==0, nbWorker=1; end
019
020 fprintf('Processing folder : %s with %d workers\n',folder,nbWorker);
021
022 nFiles=size(file_list,1);
023 resultat=struct();
024 file_error_list=cell(0);
025 taille_dataset=length(Bins);
026
027 resultat.TS=zeros(nFiles,1);
028 resultat.threshold_freqs=filter_frequencies;
029 resultat.Bins=Bins;
030 resultat.mean_S=zeros(nFiles,3);
031 resultat.std_S=zeros(nFiles,3);
032 resultat.mean_S_norm=zeros(nFiles,3);
033 resultat.std_S_norm=zeros(nFiles,3);
034 resultat.hist_S=zeros(taille_dataset,3);
035 resultat.hist_Snorm=zeros(taille_dataset,3);
036 resultat.file_error_list=cell(0);
037
038
039 for i = 1:length(filter_frequencies)
040     resultat.filter(i).hist_S = zeros(taille_dataset,3);
041     resultat.filter(i).hist_Snorm = zeros(taille_dataset,3);
042 end
043
044 D=partitionnement_2(1:nFiles,nbWorker);
045
046 spmd
047     [resultat_comp] =...
048         characterization_noise_pola_subworker(folder, ...
049             file_list(D{labindex},:), Bins, filter_frequencies);
050 end
051
052 for j=1:nbWorker
```

```
053     resultat_worker_temp=resultat_comp{j};  
054  
055     resultat.TS(D{j},1)=resultat_worker_temp.TS;  
056  
057     resultat.mean_S(D{j},1:3)=resultat_worker_temp.mean_S;  
058     resultat.std_S(D{j},1:3)=resultat_worker_temp.std_S;  
059     resultat.mean_S_norm(D{j},1:3)=...  
060         resultat_worker_temp.mean_S_norm;  
061     resultat.std_S_norm(D{j},1:3)=...  
062         resultat_worker_temp.std_S_norm;  
063  
064     resultat.hist_S=resultat.hist_S+...  
065         resultat_worker_temp.hist_S;  
066     resultat.hist_Snorm=resultat.hist_Snorm+...  
067         resultat_worker_temp.hist_Snorm;  
068  
069     resultat.file_error_list=cell(0);  
070  
071     for i=1:length(filter_frequencies)  
072         resultat.filter(i).frequency=filter_frequencies(i);  
073         resultat.filter(i).hist_S=...  
074             resultat.filter(i).hist_S+...  
075             resultat_worker_temp.filter(i).hist_S;  
076         resultat.filter(i).hist_Snorm=...  
077             resultat.filter(i).hist_Snorm+...  
078             resultat_worker_temp.filter(i).hist_Snorm;  
079     end  
080  
081     for i=1:length(resultat_worker_temp.file_error_list)  
082         resultat.file_error_list{end+1}=...  
083             resultat_worker_temp.file_error_list{i};  
084     end  
085 end  
086  
087 fprintf('Folder completed with %d errors\n',length(file_error_list));  
088  
089 end
```

```
090
091 function [resultat] = characterization_noise_pola_subworker(...
092     folder, file_list, Bins, filter_frequencies)
093
094 resultat=struct();
095 resultat.filter=struct();
096 nFiles = size(file_list,1);
097 taille_dataset=length(Bins);
098
099 for i = 1:length(filter_frequencies)
100     resultat.filter(i).hist_S = zeros(taille_dataset,3);
101     resultat.filter(i).hist_Snorm = zeros(taille_dataset,3);
102 end
103
104 resultat.TS=zeros(nFiles,1);
105 resultat.threshold_freqs=filter_frequencies;
106 resultat.Bins=Bins;
107 resultat.mean_S=zeros(nFiles,3);
108 resultat.std_S=zeros(nFiles,3);
109 resultat.mean_S_norm=zeros(nFiles,3);
110 resultat.std_S_norm=zeros(nFiles,3);
111 resultat.hist_S=zeros(taille_dataset,3);
112 resultat.hist_Snorm=zeros(taille_dataset,3);
113 resultat.file_error_list=cell(0);
114
115 for indFichier = 1:nFiles
116     if ((mod(indFichier,10)==0) && (labindex==1))
117         percentDone = indFichier/nFiles*100;
118         fprintf('%.1f %%\n',percentDone);
119     end
120     file_name=file_list(indFichier,:);
121
122     try
123         file_path = strcat(folder,file_name);
124         SORTIE=load(file_path);
125     catch
126         resultat.file_error_list{end+1}=file_name;
```

```

127         continue;
128     end
129
130     parsed_name=cell2mat(textscan(file_name,'%f %f %f %f %f %f %f'));
131     TS_file = datenum(parsed_name(2),parsed_name(3),...
132                         parsed_name(4),parsed_name(5),parsed_name(6),parsed_name(7));
133     resultat.TS(indFichier,1)=TS_file;
134
135     S=SORTIE.sop;
136     Snorm = bsxfun(@rdivide, S, sqrt(sum(S.^2,2)));
137
138     % Recherche des glitches
139     initialisation = 250;
140
141     S=S(initialisation:end,:);
142     Snorm = Snorm(initialisation:end,:);
143
144     resultat.mean_S(indFichier,:)=mean(S);
145     resultat.std_S(indFichier,:)=std(S);
146     resultat.mean_S_norm(indFichier,:)=mean(Snorm);
147     resultat.std_S_norm(indFichier,:)=std(Snorm);
148
149     for colonneId = 1:3
150         resultat.hist_S(:,colonneId)=...
151             resultat.hist_S(:,colonneId)+...
152             histc(S(:,colonneId),Bins);
153         resultat.hist_Snorm(:,colonneId)=...
154             resultat.hist_Snorm(:,colonneId)+...
155             histc(Snorm(:,colonneId),Bins);
156     end
157
158     for i = 1:length(filter_frequencies)
159         threshold_freq=filter_frequencies(i);
160         S_filter=high_pass_filter(S,threshold_freq);
161         Snorm_filter=high_pass_filter(Snorm,threshold_freq);
162         for colonneId=1:3
163             resultat.filter(i).hist_S(:,colonneId)=...

```

```
164         resultat.filter(i).hist_S(:,colonneId)+...
165             histc(S_filter(:,colonneId),Bins);
166         resultat.filter(i).hist_Snorm(:,colonneId)=...
167             resultat.filter(i).hist_Snorm(:,colonneId)+...
168                 histc(Snorm_filter(:,colonneId),Bins);
169     end
170 end
171 end
172 end
```


ANNEXE XIX

GÉNÉRATION D'UN POINT ALEATOIRE SUR UNE SPHÈRE

Le Script-A XIX-1 assure la génération d'un point aléatoire uniformément réparti sur une sphère.

Sortie

Le script retourne trois réels **x**, **y** et **z** correspondant aux coordonnées cartésiennes d'un point aléatoire dans une sphère unitaire.

Script-A XIX-1 Génération d'un point aléatoire sur une sphère

```
001 function [ x, y, z ] = generate_random_sphere_point()
002
003 temp_u = rand(1,1)*2-1;
004 temp_phi = rand(1,1)*2*pi;
005
006 x = sqrt(1-temp_u.^2) .* cos(temp_phi);
007 y = sqrt(1-temp_u.^2) .* sin(temp_phi);
008 z = temp_u;
009
010 end
```


ANNEXE XX

SIMULATION DU BRUIT DU POLARIMÈTRE

Le Script-A XX-1 permet de simuler le bruit du polarimètre.

Entrée

Ce script prend deux paramètres obligatoires :

- **nFilesPerWorker**, nombre de fichiers à traiter par moteur de calcul;
- **nPointsPerFile**, contenant la liste des fichiers à traiter (sans chemin absolu). Cette liste doit être de type *Character Array*;
- **mu**, réel correspondant à la valeur moyenne du bruit gaussien à ajouter aux mesures aléatoires;
- **sigma**, réel correspondant à l'écart type du bruit gaussien à ajouter aux mesures aléatoires.

Ce script accepte d'autres paramètres facultatifs :

- **bins**, pour définir manuellement la largeur des colonnes de l'histogramme des vitesses de rotation du vecteur de Stokes;
- **minDOP**, correspondant à la valeur minimale du degré de polarisation du signal à générer aléatoirement entre *minDOP* et 1 pour chaque fichier généré.

Sortie

Le script retourne une variable, **output_struct_stats**, qui est une structure Matlab contenant trois variables :

- **output_struct_stats.mean_ARR**, matrice à une dimension contenant la valeur moyenne de la vitesse de rotation du vecteur de Stokes observée par fichier généré;
- **output_struct_stats.std_ARR**, matrice à une dimension contenant l'écart type de la vitesse de rotation du vecteur de Stokes observée par fichier généré;
- **output_struct_stats.hist_ARR**, matrice à une dimension contenant les valeurs de l'histogramme de la vitesse de rotation du vecteur de Stokes pour chaque mesure du

polarimètre. Les colonnes correspondent à la valeur du vecteur *Bins* donné comme paramètre àu script.

Script-A XX-1 Simulation du bruit du polarimètre

```

001 function [ output_struct ] = pola_noise_simulator( varargin )
002
003 p = inputParser;
004 p.addRequired('nFilesPerWorker',@(x)((x >= 0) && (x == fix(x))));
005 p.addRequired('nPPointsPerFile',@(x)((x >= 0) && (x == fix(x))));
006 p.addRequired('mu',@isreal);
007 p.addRequired('sigma',@(x)(x >= 0));
008 p.addValue('minDOP', 1, @(x)((x >= 0) && (x <= 1)));
009 p.addValue('Bins', [0:.01:20 inf], @(x) isreal(x) &&...
010      issorted(x) && x(1)<x(end));
011 p.parse(varargin{:});
012 opts = p.Results;
013
014 rng('shuffle');
015
016 % Acquisition du nombre de workers en vue du calcul parallele
017 nbWorker = matlabpool('size');
018 if nbWorker==0, nbWorker=1; end
019
020 nFiles=opts.nFilesPerWorker*nbWorker;
021
022 fprintf('%d files to generate, with %d worker(s)\n',nFiles,nbWorker);
023
024 %% Initialisation des variables
025
026 mean_transient = zeros(nFiles,1);
027 std_transient = zeros(nFiles,1);
028
029 size_countMatrix_ARR=1;
030 hist_ARR = zeros(length(opts.Bins),1);
031

```

```

032 spmd
033     [output_struct_comp]=...
034         pola_noise_simulator_subworker(...
035             opts.nFilesPerWorker, ...
036             opts.nPointsPerFile, ...
037             opts.mu, ...
038             opts.sigma, ...
039             'minDOP',opts.minDOP, ...
040             'Bins',opts.Bins);
041 end
042 %mean_transient_comp,std_transient_comp,hist.ARR_comp
043 %% Rassemblement des valeurs
044 for worker=1:nbWorker
045     output_struct_temp = output_struct_comp{worker};
046     % Gestion ARR
047     mean_transient_comp_temp=output_struct_temp.mean_transient;
048     std_transient_comp_temp=output_struct_temp.std_transient;
049     mean_transient((opts.nFilesPerWorker*(worker-1)+1):...
050                     (opts.nFilesPerWorker*worker))=mean_transient_comp_temp;
051     std_transient((opts.nFilesPerWorker*(worker-1)+1):...
052                     (opts.nFilesPerWorker*worker))=std_transient_comp_temp;
053
054     size_hist_ARR_comp=length(output_struct_temp.hist_ARR);
055     if size_hist_ARR_comp>size_countMatrix_ARR
056         %      hist.ARR(size_hist_ARR_comp,1:3)=[0,0,0];
057         size_countMatrix_ARR=size_hist_ARR_comp;
058     end
059     hist.ARR=hist.ARR+output_struct_temp.hist_ARR;
060 end
061
062 output_struct = struct();
063 output_struct.mean_ARR=mean_transient;
064 output_struct.std_ARR=std_transient;
065 output_struct.hist_ARR=hist.ARR;
066 opts.nbWorker=nbWorker;
067 output_struct.parameters=opts;
068

```

```

069 end

070

071 function [ output_struct ] = pola_noise_simulator_subworker(varargin)
072
073 p = inputParser;
074 p.addRequired('nFiles',@(x)((x >= 0) && (x == fix(x))));
075 p.addRequired('nPPointsPerFile',@(x)((x >= 0) && (x == fix(x))));
076 p.addRequired('mu',@isreal);
077 p.addRequired('sigma',@(x)(x >= 0));
078 p.addValue('minDOP', 1, @(x)((x >= 0) && (x <= 1)));
079 p.addValue('Bins', [0:.01:20 inf], @(x) isreal(x) &&...
080 issorted(x) && x(1)<x(end));
081 p.parse(varargin{:});
082 opts = p.Results;
083
084 mean_transient = zeros(opts.nFiles,1);
085 std_transient = zeros(opts.nFiles,1);
086 size_countCol_ARR=1;
087 hist_ARR = zeros(size_countCol_ARR,1);
088
089 for fileIx=1:opts.nFiles
090 if (mod(fileIx,50)==0) && (labindex==1)
091 fprintf('%d files (%.2f%%)\n',fileIx,fileIx/opts.nFiles*100);
092 end
093
094
095 dop=(opts.minDOP+(1-opts.minDOP)*rand());
096
097 [ temp_x, temp_y, temp_z ] = generate_random_sphere_point();
098
099 rand_component_x = opts.sigma.*randn(opts.nPointsPerFile,1) +
100 opts.mu;
101 rand_component_y = opts.sigma.*randn(opts.nPointsPerFile,1) +
102 opts.mu;
103 rand_component_z = opts.sigma.*randn(opts.nPointsPerFile,1) +
104 opts.mu;
105

```

```

106     S = [ ...
107         (dop*temp_x+rand_component_x), ...
108         (dop*temp_y+rand_component_y), ...
109         (dop*temp_z + rand_component_z) ...
110     ];
111
112     Snorm = bsxfun(@rdivide, S, sqrt(sum(S.^2,2)));
113     angular_velocity=pola_angular_velocity_matrix(Snorm);
114     initialisation = 1;
115
116     mean_transient_temp = mean(angular_velocity(initialisation:...
117                                 length(angular_velocity(:,1)),1));
118
119     std_transient_temp = std(angular_velocity(initialisation:...
120                                 length(angular_velocity(:,1)),1));
121
122     taille_mesure_transient = length(angular_velocity(:,1));
123
124     hist_ARR_temp=histc(angular_velocity(initialisation:...
125                                 taille_mesure_transient),opts.Bins);
126
127     size_hist_ARR_temp=length(hist_ARR_temp);
128
129     if size_hist_ARR_temp<size_countCol_ARR
130         hist_ARR(1:size_hist_ARR_temp)=...
131             hist_ARR(1:size_hist_ARR_temp)+hist_ARR_temp;
132     elseif size_hist_ARR_temp>size_countCol_ARR
133         hist_ARR((end+1):(end+size_hist_ARR_temp-...
134             size_countCol_ARR),1)=...
135             zeros(size_hist_ARR_temp-size_countCol_ARR,1);
136         size_countCol_ARR=size_hist_ARR_temp;
137         hist_ARR=hist_ARR+hist_ARR_temp;
138     else
139         hist_ARR=hist_ARR+hist_ARR_temp;
140     end
141
142     mean_transient(fileIx)=mean_transient_temp;

```

```
143     std_transient(fileIx)=std_transient_temp;
144
145 end
146
147 output_struct = struct();
148 output_struct.mean_transient=mean_transient;
149 output_struct.std_transient=std_transient;
150 output_struct.hist_ARR=hist_ARR;
151
152 end
153
```

ANNEXE XXI

TRACÉ DE L'EVOLUTION TEMPORELLE DES PARAMETRES DE STOKES MESURÉS PAR LE POLARIMÈTRE

Le Script-A XXI-1 assure le tracé de l'évolution temporelle des paramètres de Stokes mesurés par le polarimètre (un tracé par paramètre de Stokes).

Entrée

Ce script prend deux paramètres obligatoires :

- **folder**, chaîne de caractères contenant le chemin du dossier contenant les fichiers à traiter;
- **fileList**, contenant la liste des fichiers à traiter (sans chemin absolu). Cette liste doit être de type *Character Array*.

Dépendances

Le script dépend :

- de la fonction *pola-angular-velocity-matrix*, présentée au Script-A VII-1.

Script-A XXI-1 Tracé de l'évolution temporelle des paramètres de Stokes
mesurés par le polarimètre

```
001 function [ ] = plot_SOP_reatime( varargin )
002
003 p = inputParser;
004 p.addRequired('Folder', @ischar);
005 p.addRequired('FileList', @ischar);
006 p.parse(varargin{:});
007 opts = p.Results;
008
009 folder = opts.Folder;
010 files_list = opts.FileList;
011 nFiles=size(files_list,1);
012 S = [];
```

```

013 TS = [] ;
014
015 for i = 1:nFiles
016
017     name = strtrim(files_list(i,:));
018     file_name = sprintf('%s%s%s', folder, name);
019     FichierPol=load(file_name);
020
021     parsed_name=cell2mat(textscan(name, ...
022         '%f_%f_%f_%f_%f_%f'));
023     TS_file = datenum(parsed_name(2),parsed_name(3), ...
024         parsed_name(4),parsed_name(5),parsed_name(6),parsed_name(7));
025
026     fprintf('%s : %s\n',name,datestr(TS_file));
027
028     S_file = FichierPol.sop;
029     Snorm = bsxfun(@rdivide, S_file, sqrt(sum(S_file.^2,2)));
030     angular_velocity=pola_angular_velocity_matrix(Snorm);
031     [~, loc_glitch] = max(angular_velocity(:,1));
032     initialisation = min(loc_glitch+50,size(angular_velocity,1));
033
034     S_temp = S_file(initialisation:end,:);
035     S((end+1):(end+size(S_temp,1)),:) = S_temp;
036     ech_pola_en_jour = 1/(24*3600*20e3);
037     TS_temp = TS_file+ech_pola_en_jour*(initialisation-1)+...
038         ech_pola_en_jour*(0:(size(S_temp,1)-1))';
039     TS((end+1):(end+size(TS_temp,1)),1) = TS_temp;
040
041 end
042
043 [TS, sig] = sort(TS);
044 S = S(sig,:);
045
046 C = linspecer(3);
047
048 TS=(TS-min(TS))*24*60;
049 Snorm = bsxfun(@rdivide, S, sqrt(sum(S.^2,2)));

```

```
050
051 Stokes_Components = Snorm(:,:, :) ;
052 id_plot = ~isnan(Stokes_Components(:, 1)) ;
053 TS = TS(id_plot) ;
054 Stokes_Components=...
055 Stokes_Components(~isnan(Stokes_Components(:, 1)), :) ;
056
057 id_jump = find(diff(TS)>(1/60)) ;
058 id_jump = [0 ; id_jump ; length(TS)] ;
059
060 intervals = cell(length(id_jump)-1, 1) ;
061 for i = 1:(length(id_jump)-1)
062     intervals{i, 1} = (id_jump(i)+1):(id_jump(i+1)) ;
063 end
064
065 for composanteStokes = 1:3
066
067     figure('units','normalized','outerposition',[0 0 1 1], ...
068             'visible','on') ;
069     hold on;
070     for lot = 1:length(intervals)
071         plot(TS(intervals{lot}), ...
072               Stokes_Components(intervals{lot}, composanteStokes), ...
073               'Color', C(composanteStokes, :), ...
074               'LineWidth', 1) ;
075     end
076
077     ylim([-1 1]) ;
078     xlim([min(TS) max(TS)]) ;
079     set(gca, 'XTick', 0:10:(80)) ;
080     set(gca, 'FontSize', 18) ;
081     xlabel('Temps (min)', 'FontSize', 24) ;
082     ylabel(sprintf('Composante de Stokes : S_%d', ...
083                 composanteStokes), 'FontSize', 24) ;
084     grid on
085
086 end
```

087
088 end
089
090
091

ANNEXE XXII

ANALYSE DE LA DENSITÉ SPECTRALE DES MESURES DU POLARIMÈTRE

Le Script-A XXII-1 permet de tracer la fonction de densité spectrale de puissance pour un fichier de mesure du polarimètre.

Entrée

Le script prend en paramètre une structure **FileData** contenant un fichier de mesure du polarimètre. Cette structure est obtenue typiquement en exécutant la commande :

```
FileData = load(['Chemin du fichier de mesure'])
```

Script-A XXII-1 Analyse de la densité spectrale des mesures du polarimètre

```
001 function [ ] = plot_harmonics_pola(FileData)
002
003 S = FileData.sop;
004 Snorm = bsxfun(@rdivide, S, sqrt(sum(S.^2,2)));
005 angular_velocity=pola_angular_velocity_matrix(Snorm);
006 [~, max_loc] = max(angular_velocity);
007
008 s1_brut = S((max_loc+50):end,1);
009
010 N=length(s1_brut);
011 Fs = 20e3;
012
013 figure('units','normalized','outerposition',[0 0 1 1],...
014     'visible','on');
015 xdft = fft(s1_brut);
016 xdft = xdft(1:N/2+1);
017 psdx = 1/N * abs(xdft).^2;
018 psdx(2:end-1) = 2*psdx(2:end-1);
019 freq = 0:Fs/length(s1_brut):Fs/2;
020 plot(freq,10*log10(psdx))
021
```

```
022 xlim([0 60*4+10])
023 set(gca,'XTick',0:60:60*4)
024 set(gca,'FontSize',20)
025 xlabel('Fréquence (Hz)', 'FontSize', 20)
026 ylabel('Spectre de puissance (dB)', 'FontSize', 20)
027
028 grid on
029
030 end
```

ANNEXE XXIII

DISTRIBUTIONS DES EFFETS RAPIDES DE POLARISATION

Le Script-A XXIII-1 trace trois histogrammes correspondant, pour chaque événement de polarisation enregistré, à :

- la distribution de la longueur de chaque événement;
- la distribution de la déflexion angulaire finale observée pour chaque événement;
- la distribution de la vitesse de rotation du vecteur de Stokes maximale observée pour chaque événement.

Entrée

Le script prend en entrée la structure de donnée **FileData** retournée par le Script-A VI-1.

Script-A XXIII-1 Distribution des effets rapides de polarisation

```
001 function [ ] = events_distributions( FileData )  
002  
003 % Histogramme de la longueur des événements  
004 bins_Duration = (0:.1:12)';  
005 Duration = (FileData.events_SOP(:,3)-FileData.events_SOP(:,2))...  
006 / (20e3)*1e3;  
007  
008 mean_elas = mean(Duration);  
009 std_elas = std(Duration);  
010  
011 legend_string_1 = ...  
012 sprintf('$\bar{x} = $%6.4f ms, $\sigma = $%6.4f ms', ...  
013 mean_elas, std_elas);  
014 legend_string_1=strrep( legend_string_1, '.', ',');  
015  
016 hist_Duration = histc(Duration, bins_Duration);  
017
```

```

018 hist_Duration = hist_Duration/sum(hist_Duration);
019 hist_Duration = hist_Duration/trapz(bins_Duration, hist_Duration);
020
021 y_min_global =...
022     10^(floor(log10(min(hist_Duration(hist_Duration>0)))))-1;
023
024 figure('units','normalized','outerposition',[0 0 1 1],...
025     'visible','on');
026
027 bar(bins_Duration, hist_Duration,'BaseValue',y_min_global)
028 xlim([min(bins_Duration) max(bins_Duration)])
029 set(gca,'YScale','log')
030 set(gca,'FontSize',18)
031 legend(legend_string_1)
032 set(legend,'Interpreter','latex','FontName','Calibri',...
033     'FontSize',14,'Position',[.6 .8 .3 .1]);
034 xlabel('Durée de l''événement de polarisation (ms)', 'FontSize',24);
035 ylabel('Distribution de probabilité normalisée', 'FontSize',24);
036 grid on
037
038 % Histogramme de la déflexion angulaire finale
039 bins_FinalDeflectionection = (0:.01:3)';
040 FinalDeflection = FileData.events_SOP(:,6);
041
042 mean_elas = mean(FinalDeflection);
043 std_elas = std(FinalDeflection);
044
045 legend_string_1 =...
046     sprintf('$\bar{x} = $%6.4f rad, $\sigma = $%6.4f rad',...
047     mean_elas, std_elas);
048 legend_string_1=strrep( legend_string_1,'.',',');
049
050 hist_Elas = histc(FinalDeflection, bins_FinalDeflectionection);
051 hist_Elas = hist_Elas/sum(hist_Elas);
052 hist_Elas = hist_Elas/trapz(bins_FinalDeflectionection, hist_Elas);
053
054 y_min_global = 10^(floor(log10(min(hist_Elas(hist_Elas>0)))))-1);

```

```

055
056 figure('units','normalized','outerposition',[0 0 1 1],...
057     'visible','on');
058
059 bar(bins_FinalDeflectionection, hist_Elas,'BaseValue',y_min_global)
060 xlim([min(bins_FinalDeflectionection),...
061         max(bins_FinalDeflectionection)])
062 set(gca,'YScale','log')
063 set(gca,'FontSize',18)
064 legend(legend_string_1)
065 set(legend,'Interpreter','latex','FontName','Calibri',...
066     'FontSize',14,'Position',[.6 .8 .3 .1]);
067 xlabel('Déflexion angulaire finale de l''évnement (rad)',...
068     'FontSize',24);
069 ylabel('Distribution de probabilité normalisée','FontSize',24);
070 grid on
071
072 % Histogramme de la vitesse de rotation du vecteur de Stokes maximale
073 bins_MaxRR = (0:.1:16)';
074 MaxARR = FileData.events_SOP(:,4); % Units of [rad/ms]
075
076 mean_amplitude = mean(MaxARR);
077 std_amplitude = std(MaxARR);
078
079 legend_string_1 =...
080     sprintf('$\bar{x} = $%6.4f rad/ms, $\sigma = $%6.4f rad/ms',...
081     mean_amplitude, std_amplitude);
082 legend_string_1=strrep( legend_string_1,'.',',');
083
084 hist_Amplitude = histc(MaxARR, bins_MaxRR);
085
086 hist_Amplitude = hist_Amplitude/sum(hist_Amplitude);
087 hist_Amplitude = hist_Amplitude/trapz(bins_MaxRR, hist_Amplitude);
088
089 y_min_global =...
090     10^(floor(log10(min(hist_Amplitude(hist_Amplitude>0)))))-1);
091

```

```
092 figure('units','normalized','outerposition',[0 0 1 1],...
093     'visible','on');
094
095 bar(bins_MaxRR, hist_Amplitude,'BaseValue',y_min_global)
096 xlim([min(bins_MaxRR) max(bins_MaxRR)])
097 set(gca,'YScale','log')
098 set(gca,'FontSize',18)
099 legend(legend_string_1)
100 set(legend,'Interpreter','latex','FontName','Calibri',...
101     'FontSize',14,'Position',[.5 .8 .35 .1]);
102 xlabel(...,
103     'Vitesse maximale de rotation du vecteur de Stokes (rad/ms)',...
104     'FontSize',24);
105 ylabel('Distribution de probabilité normalisée','FontSize',24);
106 grid on
107
108 end
```

ANNEXE XXIV

HISTOGRAMME 3D DES DÉFLEXIONS ANGULAIRES MAXIMALES ET DES VITESSES DE ROTATION DU VECTEUR DE STOKES MAXIMALES

Le Script-A XXIV-1 permet de tracer l'histogramme 3D des déflexions angulaires maximales et des vitesses de rotation du vecteur de Stokes maximales.

Entrée

Le script prend en entrée la structure de donnée FileData retournée par le Script-A VI-1.

Script-A XXIV-1 Histogramme 3D des déflexions angulaires maximales et des vitesses de rotation du vecteur de Stokes maximales

```
001 function [ ] = hist_3d_pola(FileData)
002
003 events_SOP = FileData.events_SOP;
004
005 taille_events_SOP = size(events_SOP,1);
006
007 offset_xlim=10;
008 offset_ylim=10;
009
010 min_ARR=.9;
011 width_ARR=.1;
012 max_ARR=16;
013
014 min_Def=0;
015 width_Def=.0125;
016 max_Def=3;
017
018 bins_ARR = min_ARR:width_ARR:max_ARR;
019 bins_Def = min_Def:width_Def:max_Def;
020
021 edges = { bins_ARR ; bins_Def };
```

```

022
023 dataset_hist = events_SOP(:,[4 5]);
024
025 countMatrix_ARR_Def = hist3(dataset_hist,edges)';
026
027 %% Mise en place de la matrice PDF
028 Sum_Prob = sum(sum(countMatrix_ARR_Def));
029 matrix_PDF_3D = countMatrix_ARR_Def/Sum_Prob;
030
031 %% Construction dynamique des ticks d'axes
032 TickDefl = .5;
033 TicksDefl=1:(TickDefl/width_Def):length(bins_Def);
034 ULabelDefl=min_Def+(TicksDefl-1)*width_Def;
035 LabelDefl=cell(1,length(ULabelDefl));
036
037 for i=1:length(ULabelDefl)
038     LabelDefl{i}=ULabelDefl(i);
039 end
040
041 TickARR = 3;
042 TicksARR=[1 12:(TickARR/width_ARR):length(bins_ARR)];
043 ULabelARR=min_ARR+(TicksARR-1)*width_ARR;
044 LabelARR=cell(1,length(ULabelDefl));
045
046 for i=1:length(ULabelARR)
047     LabelARR{i}=ULabelARR(i);
048 end
049
050 %% Creation de l'histogramme 3D
051 figure('units','normalized','outerposition',[0 0 1 1],...
052     'visible','on');
053 h1 = bar3(matrix_PDF_3D);
054 caxis([min(min(log(matrix_PDF_3D))) 0]);
055
056 myColorMap=[
057     0.86,0.86,0.86; 0.88,0.88,0.88; 0.89,0.89,0.89; 0.91,0.91,0.91;
058     0.93,0.93,0.93; 0.94,0.94,0.94; 0.78,0.95,0.95; 0.63,0.96,0.96;

```

```

059      0.47,0.97,0.97; 0.31,0.98,0.98; 0.16,0.99,0.99; 0.00,1.00,1.00;
060      0.04,1.00,0.96; 0.07,1.00,0.93; 0.11,1.00,0.89; 0.14,1.00,0.86;
061      0.18,1.00,0.82; 0.21,1.00,0.79; 0.25,1.00,0.75; 0.29,1.00,0.71;
062      0.32,1.00,0.68; 0.36,1.00,0.64; 0.39,1.00,0.61; 0.43,1.00,0.57;
063      0.46,1.00,0.54; 0.50,1.00,0.50; 0.54,1.00,0.46; 0.57,1.00,0.43;
064      0.61,1.00,0.39; 0.64,1.00,0.36; 0.68,1.00,0.32; 0.71,1.00,0.29;
065      0.75,1.00,0.25; 0.79,1.00,0.21; 0.82,1.00,0.18; 0.86,1.00,0.14;
066      0.89,1.00,0.11; 0.93,1.00,0.07; 0.96,1.00,0.04; 1.00,1.00,0.00;
067      1.00,0.94,0.00; 1.00,0.89,0.00; 1.00,0.83,0.00; 1.00,0.78,0.00;
068      1.00,0.72,0.00; 1.00,0.67,0.00; 1.00,0.61,0.00; 1.00,0.56,0.00;
069      1.00,0.50,0.00; 1.00,0.44,0.00; 1.00,0.39,0.00; 1.00,0.33,0.00;
070      1.00,0.28,0.00; 1.00,0.22,0.00; 1.00,0.17,0.00; 1.00,0.11,0.00;
071      1.00,0.06,0.00; 1.00,0.00,0.00; 0.97,0.03,0.00; 0.95,0.05,0.00;
072      0.92,0.08,0.00; 0.90,0.11,0.00; 0.87,0.13,0.00; 0.85,0.16,0.00;
073      ];
074
075 colormap(myColorMap);
076
077 %% Parade pour le probleme d'affichage des couleurs en log
078 for n=1:numel(h1)
079     cdata=get(h1(n), 'zdata');
080     cdata = log10(cdata);
081     set(h1(n), 'cdata',cdata,'facecolor','interp')
082 end
083
084 set(gca, 'ZScale', 'log');
085 set(gca, 'ZMinorTick', 'on');
086 set(gca, 'YDir', 'normal');
087 view(gca, [115 25]);
088
089 llim = 1/taille_events_SOP;
090
091 h2 = get(gca, 'Children');
092
093 for i = 1:length(h2)
094     ZData = get(h2(i), 'ZData');
095     ZData(ZData==0) = llim/3;

```

```

096     set(h2(i), 'ZData', ZData);
097 end
098
099 xmin_3d=0;
100 xmax_3d=length(bins_ARR)+offset_xlim;
101 ymin_3d=0;
102 ymax_3d=length(bins_Def)+offset_ylim;
103 zmin_3d=5e-6;
104 zmax_3d=1e0;
105
106 xlim([xmin_3d xmax_3d])
107 ylim([ymin_3d ymax_3d])
108 zlim([zmin_3d zmax_3d])
109
110 set(gca,'XTick',TicksARR);
111 set(gca,'XTickLabel',LabelARR,'FontName','Calibri','fontsize',15);
112 set(gca,'YTick',TicksDefl);
113 set(gca,'YTickLabel',LabelDefl,'FontName','Calibri','fontsize',15);
114 set(gca,'ZTick',[1e-6 1e-5 1e-4 1e-3 1e-2 1e-1 1e0]);
115
116 xlabel(sprintf('Vitesse de rotation maximale\n(rad/ms)'),...
117     'FontName','Calibri','fontsize',16,'fontweight','b',...
118     'Position',[260 320 1e-4],'Rotation',45);
119 ylabel('Déflexion maximale (rad)', 'FontName','Calibri',...
120     'fontsize',16,'fontweight','b','Position',[260 225 4e-5],...
121     'Rotation',-11);
122 zlabel('Distribution de probabilité normalisée',...
123     'FontName','Calibri','fontsize',16,'fontweight',...
124     'b','Position',[250 10 .15],'Rotation',90);
125
126 %% Creation de l'histogramme 2D
127
128 figure('units','normalized','outerposition',[0 0 1 1],...
129     'visible','on');
130
131 for i=1:length(bins_Def)
132     for j=1:length(bins_ARR)

```

```

133         if matrix_PDF_3D(i,j) == 0
134             matrix_PDF_3D(i,j) = llim/3;
135         end
136     end
137 end
138
139 % Pour avoir le diagramme sur fond blanc : contourf -> contour
140 contourf(log(matrix_PDF_3D));
141 caxis([min(min(log(matrix_PDF_3D))) log(1e0)])
142
143 contourf_myColorMap=
144     0.00,0.00,0.80; 0.04,0.52,0.78; 0.04,0.55,0.79; 0.04,0.57,0.81;
145     0.04,0.60,0.82; 0.03,0.63,0.83; 0.03,0.66,0.84; 0.03,0.69,0.86;
146     0.03,0.72,0.87; 0.02,0.74,0.88; 0.02,0.77,0.90; 0.02,0.80,0.91;
147     0.02,0.83,0.92; 0.01,0.86,0.94; 0.01,0.89,0.95; 0.01,0.91,0.96;
148     0.01,0.94,0.97; 0.00,0.97,0.99; 0.00,1.00,1.00; 0.06,1.00,0.94;
149     0.11,1.00,0.89; 0.17,1.00,0.83; 0.22,1.00,0.78; 0.28,1.00,0.72;
150     0.33,1.00,0.67; 0.39,1.00,0.61; 0.44,1.00,0.56; 0.50,1.00,0.50;
151     0.56,1.00,0.44; 0.61,1.00,0.39; 0.67,1.00,0.33; 0.72,1.00,0.28;
152     0.78,1.00,0.22; 0.83,1.00,0.17; 0.89,1.00,0.11; 0.94,1.00,0.06;
153     1.00,1.00,0.00; 1.00,0.94,0.00; 1.00,0.89,0.00; 1.00,0.83,0.00;
154     1.00,0.78,0.00; 1.00,0.72,0.00; 1.00,0.67,0.00; 1.00,0.61,0.00;
155     1.00,0.56,0.00; 1.00,0.50,0.00; 1.00,0.44,0.00; 1.00,0.39,0.00;
156     1.00,0.33,0.00; 1.00,0.28,0.00; 1.00,0.22,0.00; 1.00,0.17,0.00;
157     1.00,0.11,0.00; 1.00,0.06,0.00; 1.00,0.00,0.00; 0.94,0.00,0.00;
158     0.89,0.00,0.00; 0.83,0.00,0.00; 0.78,0.00,0.00; 0.72,0.00,0.00;
159     0.67,0.00,0.00; 0.61,0.00,0.00; 0.56,0.00,0.00; 0.50,0.00,0.00;
160 ];
161
162 colormap(contourf_myColorMap); % Apply the colormap
163
164 xlim([xmin_3d xmax_3d])
165 ylim([ymin_3d ymax_3d])
166
167 set(gca,'XTick',TicksARR);
168 set(gca,'XTickLabel',LabelARR,'FontName','Calibri','fontsize',15);
169 set(gca,'YTick',TicksDefl);

```

```
170 set(gca,'YTickLabel',LabelDef1,'FontName','Calibri','fontsize',15);
171
172 xlabel('Maximum rotation rate (rad/ms)', 'FontName','Calibri',...
173     'fontsize',16,'fontweight','b');
174 ylabel('Maximum angular deflection (rad)', 'FontName','Calibri',...
175     'fontsize',16,'fontweight','b');
176
177 labels = cell(64);
178 for i = 64:-1:1
179     if mod(64-i,5)==0
180         labels{i}=sprintf('10^-%d',(64-i)/5);
181     else
182         labels{i}='';
183     end
184 end
185
186 lcolorbar(labels)
187
188 end
```

BIBLIOGRAPHIE

- Abuzariba, Suad Mohamed. 2010. « Theoretical Calculation of System Performance of Fiber Optic Network with Chromatic Dispersion, Polarization Mode Dispersion, Polarization Dependent Loss, and Amplifier Spontaneous Emission Noise. » University of Ottawa (Canada). En ligne. <<http://www.ruor.uottawa.ca/handle/10393/30040>>. Consulté le 2 mars 2015.
- Agilent Technologies. 2013. *Agilent N7781B Polarization Analyzer*.
- Bao, Xiaoyi, David S. Waddy et Liang Chen. 2004. « Polarization fluctuations in field fibers. » Dans *The 17th Annual Meeting of the IEEE Lasers and Electro-Optics Society, 2004. LEOS 2004.*, vol. 1, p. 82-83. IEEE. En ligne.
<<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1363121>>.
- Blaise Barney, Lawrence Livermore National Laboratory. 2014. « Introduction To Parallel Computing. » Dans *High Performance Computing Training*. En ligne.
<https://computing.llnl.gov/tutorials/parallel_comp/>.
- Boroditsky, M., M. Brodsky, N.J. Frigo, P. Magill et H. Rosenfeldt. 2005. « Polarization Dynamics in Installed Fiberoptic Systems. » Dans *2005 IEEE LEOS Annual Meeting Conference Proceedings*, p. 414-415. IEEE. En ligne.
<<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1548054>>. Consulté le 4 février 2015.
- Bruhat, Georges, Alfred Kastler et Patrick Bouchareine. 2005. « Optique. » Paris : Dunod.
- Bulow, H., W. Baumert, H. Schmuck, F. Mohr, T. Schulz, F. Kuppers et W. Weiershausen. 1999. « Measurement of the maximum speed of PMD fluctuation in installed field fiber. » Dans *OFC/IOOC . Technical Digest. Optical Fiber Communication Conference, 1999, and the International Conference on Integrated Optics and Optical Fiber Communication*, vol. 2, p. 83-85. IEEE. En ligne.
<<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=766343>>. Consulté le 4 février 2015.
- Chan, Calvin C K. 2010. « Optical performance monitoring advanced techniques for next-generation photonic networks. » Burlington, MA : Academic Press/Elsevier. En ligne.
<<http://app.knovel.com/web/toc.v/cid:kPOMATNG1>>.
- Crawford, Travis H. R., Michael Reimer, Maurice O'Sullivan et Michel Bélanger. 2014. « Correspondance privée. » Ciena corp.

- Dumas, Derek. 2009. « Polarization Effects of Mechanical Impacts on Dispersion Compensating Modules. » En ligne.
<https://uwspace.uwaterloo.ca/handle/10012/4446>. Consulté le 4 février 2015.
- Ednay, Richard. 2009. « Field trial of pmd test methods and investigation into the dynamics of polarisation effects in a variety of installed cable environments. » *Optical Technology Training Ltd*, vol. 44, no 0.
- Hui, Rongqing et Maurice O'Sullivan. 2009. *Fiber Optic Measurement Techniques*. Elsevier. Elsevier. En ligne. <<http://app.knovel.com/web/toc.v/cid:kpFOMT0001>>. Consulté le 10 mars 2015.
- Hui, Zang, Zhao Guiliang et Dejung Wang. 1991. « Super-Gaussian window function and its applications. » Dans *1991 International Conference on Circuits and Systems*, p. 595-598. IEEE. En ligne.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=184426>. Consulté le 1 juillet 2015.
- Karlsson, O., J. Brentel et P.A. Andrekson. 2000. « Long-term measurement of PMD and polarization drift in installed fibers. » *Journal of Lightwave Technology*, vol. 18, no 7, p. 941-951. En ligne.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=850739>. Consulté le 3 juillet 2015.
- Keiser, Gerd. 2010. « Optical fiber communications. » New York : McGraw-Hill.
- Kohns, P. et Y. Macheckhin. 2002. « Measurement of polarization and applications. » Dans *Proceedings of LFNFM 2002. 4th International Workshop on Laser and Fiber-Optical Networks Modeling (IEEE Cat. No.02EX549)*, p. 188-196. IEEE. En ligne.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1014163>. Consulté le 6 août 2015.
- Kreyszig, Erwin. 2011. *Advanced engineering mathematics*. Sous la dir. de Herbert Kreyszig et E J (Edward J.) Norminton. 10th ed. Hoboken, N.J. : Wiley.
- Krummrich, P.M., E.-D. Schmidt, W. Weiershausen et A. Mattheus. 2005. « Field trial results on statistics of fast polarization changes in long haul WDM transmission systems. » Dans *OFC/NFOEC Technical Digest. Optical Fiber Communication Conference, 2005.*, vol. 4, p. 3 pp. Vol. 4. IEEE. En ligne.
<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1501559>. Consulté le 4 février 2015.
- Leeson, Jesse, Xiaoyi Bao et Alain Côté. 2009. « Polarization dynamics in optical ground wire network. » *Applied optics*, vol. 48, no 12, p. 2214-2219.

- Leo, P.J., G.R. Gray, G.J. Simer et K.B. Rochford. 2003. « State of polarization changes: classification and measurement. » *Journal of Lightwave Technology*, vol. 21, no 10, p. 2189-2193. En ligne.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1236487>. Consulté le 27 avril 2015.
- Matsuda, T., T. Kawasaki, T. Kotanigawa, A. Naka et K. Oda. 2009. « Field trial of 43-Gbit/s RZ-DQPSK transmission in aerial fiber with rapidly changing SOP. »
- Mortelette, Aurélien. 2014. « Monitoring et caractérisation des effets de polarisation de deux liaisons optiques cohérentes. » École de technologie supérieure.
- Nelson, L.E. 2012. « Polarization effects in coherent systems. »
- Nelson, Lynn E., Martin Birk, Sheryl L. Woodward et Peter Magill. 2011. « Field measurements of polarization transients on a long-haul terrestrial link. » Dans *IEEE Photonic Society 24th Annual Meeting*, p. 833-834. IEEE. En ligne.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6110816>. Consulté le 9 février 2015.
- Orfanidis, Sophocles J. 2007. *Optimum Signal Processing . An Introduction*. 2nd éd. McGraw-Hill Publishing Company. En ligne.
<http://eceweb1.rutgers.edu/~orfanidi/osp2e/>.
- Peterson, D.L., P.J. Leo et K.B. Rochford. 2004. « Field measurements of state of polarization and PMD from a tier-1 carrier. »
- Peterson, Daniel, Bryan Ward, Kent Rochford, Paul Leo et Greg Simer. 2002. « Polarization mode dispersion compensator field trial and field fiber characterization. » *Optics Express*, vol. 10, no 14, p. 614. En ligne.
<http://www.opticsinfobase.org/viewmedia.cfm?uri=oe-10-14-614&seq=0&html=true>. Consulté le 27 avril 2015.
- Peterson Jr, Daniel L. 2004. « Estimation of outage statistics due to polarization mode dispersion. » *ProQuest Dissertations and Theses*. Ann Arbor : The University of Texas at Dallas. En ligne.
<http://search.proquest.com/docview/305097610?accountid=27231>.
- Poirrier, Julien, Suzanne Salaun, Gerard Buxeres, Bruno Raguenes, Frederic Neddam et Maryse Moignard. 2008. « Field demonstration of 10 Gbit/s transmission over a 37 ps PMD cable using electronic mitigation. » Dans *OFC/NFOEC 2008 - 2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference*, p. 1-3. IEEE. En ligne.
<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4528210>. Consulté le 30 mars 2015.

- Roberge, Raymond, Francis Audet et Larry Bernier. 2009. « Case study: SOP and PMD measurements on aerial fiber under typical wind gust conditions. » EXFO. En ligne. <metrotek.ru/lib/pdf/EXFO/appnotes/tnote040-ang.pdf>.
- Roberts, K., M. O'Sullivan, A. Awadalla, D.J. Krause et C. Laperle. 2009. « Performance of Dual-Polarization QPSK for Optical Transport Systems. » *Journal of Lightwave Technology*, vol. 27, no 16, p. 3546-3559. En ligne. <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4912371>>. Consulté le 4 février 2015.
- Roberts, Kim, Douglas Beckett, David Boertjes, Joseph Berthold et Charles Laperle. 2010. « 100G and beyond with digital coherent signal processing. » *IEEE Communications Magazine*, vol. 48, no 7, p. 62-69. En ligne. <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5496879>>. Consulté le 12 février 2015.
- Salaun, S., F. Neddam, J. Poirrier, B. Raguenes et M. Moignard. 2009. « Fast SOP variation measurement on WDM systems are the OPMDC fast enough? »
- Scott, David W. 1992. *Multivariate density estimation : theory, practice, and visualization*. New York : Wiley.
- Sun, Han, Kuang-Tsan Wu et Kim Roberts. 2008. « Real-time measurements of a 40 Gb/s coherent system. » *Optics Express*, vol. 16, no 2, p. 873-879. En ligne. <<http://www.opticsexpress.org/abstract.cfm?URI=oe-16-2-873>>.
- Szafraniec, Bogdan, Todd S. Marshall et Bernd Nebendahl. 2013. « Performance Monitoring and Measurement Techniques for Coherent Optical Systems. » *Journal of Lightwave Technology*, vol. 31, no 4, p. 648-663. En ligne. <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6263258>>. Consulté le 23 avril 2015.
- Walpole, Ronald E. 2011. « Probability & statistics for engineers & scientists. » *Probability and statistics for engineers and scientists*. Boston : Pearson Education.
- Weisstein, Eric W. 2015. « Sphere Point Picking. » MathWorld--A Wolfram Web Resource.
- Woodward, S.L., L.E. Nelson, M.D. Feuer, X. Zhou, P.D. Magill, S. Foo, D. Hanson, H. Sun, M. Moyer et M. O'Sullivan. 2008. « Characterization of Real-Time PMD and Chromatic Dispersion Monitoring in a High-PMD 46-Gb/s Transmission System. » *IEEE Photonics Technology Letters*, vol. 20, no 24, p. 2048-2050. En ligne. <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4663558>>. Consulté le 9 février 2015.

- Woodward, S.L., L.E. Nelson, C.R. Schneider, L.A. Knox, M. O'Sullivan, C. Laperle, M. Moyer et S. Foo. 2012. « Field observations of optical parameters using integrated performance monitoring. »
- Woodward, Sheryl L., Lynn E. Nelson, Charles R. Schneider, Laurie A. Knox, Maurice O'Sullivan, Charles Laperle, Michael Moyer et Sik Foo. 2014. « Long-Term Observation of PMD and SOP on Installed Fiber Routes. » *IEEE Photonics Technology Letters*, vol. 26, no 3, p. 213-216. En ligne.
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6663615>>. Consulté le 9 février 2015.
- Xu, Li. 2010. « All-order polarization mode dispersion sensing and compensation in optical fiber communication systems. » *ProQuest Dissertations and Theses*. Ann Arbor : Purdue University. En ligne.
<<http://search.proquest.com/docview/750353401?accountid=27231>>.

