

Université de Montréal

Optimisation du traitement numérique de signaux générés dans un cristal de ICs

par
Simon Lagrange

Département de physique
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en physique

novembre, 2016

© Simon Lagrange, 2016.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Optimisation du traitement numérique de signaux générés dans un cristal de ICs

présenté par:

Simon Lagrange

a été évalué par un jury composé des personnes suivantes:

Viktor Zacek,	président-rapporteur
Paul Taras,	directeur de recherche
Sjoerd Roorda,	membre du jury

Mémoire accepté le:

RÉSUMÉ

Ce travail a pour but l'optimisation du traitement numérique de signaux générés dans le cristal de ICs d'un calorimètre électromagnétique, dans le cadre d'expériences à haut taux de comptage telle que Belle II. La scintillation du cristal est convertie en signal électronique par une photopentode de Hamamatsu avec un gain de 150 ± 2 électrons par photoélectron, pour être amplifiée par un préamplificateur. Le *shaper*, un filtre actif de type CR-(RC)⁴, effectue ensuite une mise en forme du signal, qu'un ADC numérise alors pour qu'il soit traité numériquement par le DSP. À partir de formes de référence, le DSP peut extraire l'énergie déposée par une particule incidente et le temps d'occurrence de son passage.

Pour définir les formes de référence, on a déterminé que sur les trois approches étudiées, soit des formes moyennes expérimentales, soit à partir de la fonction analytique pour un filtre actif ou soit une somme de 3 gaussiennes, la plus optimale est l'ajustement de cette dernière fonction, que ce soit avec un générateur d'impulsions ou un cristal de ICs. De plus, la règle pour convertir l'amplitude des signaux mesurés en énergie déposée dans le cristal a été établie, malgré une gamme dynamique étroite.

On a aussi observé des temps moyens de propagation de la scintillation dans le cristal de 47 ± 4 ns et $3,4 \pm 0,5$ μ s, liés aux deux états d'excitation accessibles aux atomes du ICs, mais aussi à la géométrie du cristal, aux réflexions de photons sur les parois et au temps de réaction des circuits électroniques, qui allongent ces temps de scintillation mesurés.

Mots clés: Calorimètre électromagnétique, ICs, shaper, DSP, traitement numérique, Belle II.

ABSTRACT

The goal of this work is to optimize the digital processing of signals generated in the CsI crystal of an electromagnetic calorimeter, in the context of high counting rate experiments like Belle II. The scintillation from the crystal is converted into an electronic signal by a Hamamatsu photopentode with a gain of 150 ± 2 electrons per photoelectron, to then be amplified by a preamplifier. The signal is then shaped by the shaper, a CR-(RC)⁴ active filter, before it is digitized by an ADC, to be processed by the DSP. Using reference shapes, the DSP can extract the energy deposited by an incident particle and the time of occurrence of it going through the crystal.

To define the reference shapes, we determined that of the three approaches studied, which are experimental average shapes, the analytical function for an active filter and the sum of 3 gaussian functions, the best results were achieved using the last one, both with the pulse generator and the CsI crystal. Also, a conversion formula has been established to convert the measured signals' amplitudes into deposited energy in the crystal, despite a narrow dynamic range.

We also observed average propagation times of the scintillation through the crystal of 47 ± 4 ns and $3,4 \pm 0,5$ μ s, related to the two accessible excitation states of the CsI atoms, but also to the crystal's geometry, the reflections of photons on its surface and the reaction time of the electronics circuits, which make those scintillation times appear longer.

Keywords: Electromagnetic calorimeter, CsI, shaper, DSP, digital processing, Belle II.

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
Liste des Tableaux	viii
Liste des Figures	ix
Liste des Annexes	xxi
REMERCIEMENTS	xxiii
CHAPITRE 1 : INTRODUCTION	1
CHAPITRE 2 : MOTIVATIONS PHYSIQUES DE BELLE II	3
2.1 Étude des saveurs lourdes	3
2.2 Violation CP et symétries	3
2.3 Modèle Standard et nouvelle physique	6
CHAPITRE 3 : SUPERKEKB ET DÉTECTEURS	8
3.1 Production des faisceaux	8
3.1.1 Contrôle de la trajectoire des faisceaux	9
3.1.2 Anneau d’amortissement	9
3.2 Luminosité	10
3.3 Région d’interaction	12
3.3.1 Point d’interaction	14
3.4 Sources de bruit de fond	15
3.4.1 Radiation synchrotron	15
3.4.2 Diffusion Touschek	16

3.4.3	Diffusion faisceau-gaz	16
3.4.4	Nuages d'électrons dans le LER	17
3.4.5	Radiation Bhabha	18
3.4.6	Production de paires via 2 photons	19
3.4.7	Interactions faisceau-faisceau	19
3.5	Détecteurs de Belle II	20
3.5.1	Calorimètre électromagnétique	21
CHAPITRE 4 : ÉLÉMENTS DU CIRCUIT ICS-DAQ		23
4.1	Cristal de ICs	23
4.1.1	Dépôt d'énergie	23
4.1.2	Excitation du milieu	25
4.1.3	Cristaux de ICs	26
4.1.4	Muons	27
4.2	Photomultiplicateur	31
4.3	Préamplificateur	31
4.3.1	Amplification	32
4.3.2	Forme du signal sortant	32
4.3.3	Piédestal et seuil de détection	33
4.3.4	Ajustements	35
4.3.5	Conversion énergie-amplitude	43
4.4	Shaper	44
4.4.1	Filtres passe-haut et passe-bas	44
4.4.2	Filtres CR-(RC) ⁴	45
4.4.3	Temps de montée	46
4.4.4	Gamme dynamique	49
4.5	ADC	51
4.6	DSP	51
4.6.1	Extraction des valeurs d'intérêt	53
4.7	DAQ	61

CHAPITRE 5 : FORMES DE RÉFÉRENCE	62
5.1 Générateur d'impulsions	63
5.1.1 Formes moyennes expérimentales	63
5.1.2 Filtres actifs	67
5.1.3 Somme de 3 gaussiennes	76
5.1.4 Comparaison	84
5.2 Cristal de ICs	85
5.2.1 Formes moyennes expérimentales	87
5.2.2 Filtres actifs	89
5.2.3 Somme de 3 gaussiennes	95
5.2.4 Comparaison	102
5.3 Qualité des formes de référence	103
5.3.1 Générateur d'impulsions	103
5.3.2 Cristal de ICs	115
CHAPITRE 6 : CONCLUSION	128
BIBLIOGRAPHIE	131

LISTE DES TABLEAUX

VI.I	Table de coefficients pour les formes moyennes expérimentales, pour les décalages dans le temps de 0 à -9 ns, pour des signaux produits par un générateur d'impulsions.	xlvi
VI.II	Table de coefficients pour l'équation (5.1) pour les filtres actifs, pour les décalages dans le temps de 0 à -9 ns, pour des signaux produits par un générateur d'impulsions.	liii
VI.III	Table de coefficients pour la somme de 3 gaussiennes, pour les décalages dans le temps de 0 à -9 ns, pour des signaux produits par un générateur d'impulsions.	lix
VI.IV	Table de coefficients pour les formes moyennes expérimentales, pour les décalages dans le temps de 0 à -9 ns, pour des signaux générés dans un cristal de ICs.	lxvi
VI.V	Table de coefficients pour l'équation (5.1) pour les filtres actifs, pour les décalages dans le temps de 0 à -9 ns, pour des signaux générés dans un cristal de ICs.	lxxii
VI.VI	Table de coefficients pour la somme de 3 gaussiennes, pour les décalages dans le temps de 0 à -9 ns, pour des signaux générés dans un cristal de ICs.	lxxviii

LISTE DES FIGURES

2.1	Triangle d'unitarité de la matrice CKM et les contraintes expérimentales sur celui-ci [23].	4
2.2	Diagrammes de Feynman pour les désintégrations (2.1a) et (2.1b), qui permettent de distinguer les mésons $B^0 (d\bar{b})$ et $\bar{B}^0 (b\bar{d})$ par la charge du muon produit [23].	5
2.3	Deux mécanismes de mélange permettant aux mésons $B^0 (d\bar{b})$ et $\bar{B}^0 (b\bar{d})$ d'osciller d'une forme à l'autre [23].	5
2.4	Exemple de processus au-delà du Modèle Standard, $b \rightarrow s\bar{s}s$, prédit par la SuperSymétrie, avec un squark gluino-down \bar{g} [6].	6
3.1	Définition de la taille longitudinale d et de l'angle de croisement 2ϕ entre les paquets des faisceaux. L'axe du cylindre de détecteurs est représenté en pointillés.	11
3.2	En haut, les paquets de particules se croisent selon un angle. En bas, les cavités de crabe font subir une rotation à ces paquets afin qu'ils se présentent de face [23].	12
3.3	Schéma du point d'interaction, vu du haut et de côté. Les électrons se dirigent vers l'avant et les positrons vers l'arrière. Les longueurs indiquées sont en mm [6].	14
3.4	Tuyau en cuivre avec des antichambres pour réduire l'importance des nuages d'électrons et de la radiation synchrotron [6].	17
3.5	Diagrammes de Feynman pour la diffusion Bhabha radiative.	18

3.6	Disposition des différents détecteurs utilisés dans Belle II en haut et Belle en bas. <i>IP Chamber</i> : Chambre du point d'interaction. <i>PXD</i> : Détecteurs à pixels. <i>SVD</i> : Détecteurs de vertex au silicium. <i>CDC</i> : Chambre à dérive centrale. <i>PID</i> : Détecteurs pour l'identification des particules. <i>ECL</i> : Calorimètres électromagnétiques. <i>KLM</i> : Détecteurs à K_L^0 et à muons. <i>ACC</i> : Compteurs en aérogel à effet Cherenkov. <i>TOF</i> : Compteurs de temps-de-vol. <i>QCS</i> : Aimants quadrupolaires pour la focalisation finale du faisceau [6].	20
3.7	Chaîne qui constitue le calorimètre électromagnétique.	21
4.1	Perte d'énergie par ionisation selon l'équation de Bethe-Block (4.1), pour différents atomes [16].	24
4.2	Énergie perdue selon la distance parcourue dans un milieu (ici de l'eau), pour des rayons X, des protons, des électrons et des photons émis par du ^{60}Co [11]. Une réponse de 100% correspond à une perte d'énergie maximale.	24
4.3	Pics principaux d'émission du ICs, du ICs(Tl) et du ICs(Na) dans le spectre visible [12].	27
4.4	Distribution en énergie des muons cosmiques, selon l'altitude [17].	29
4.5	Pouvoir d'arrêt du muon selon sa quantité de mouvement [16]. . .	30
4.6	Exemple de signal sortant du préamplificateur, avant élimination du piédestal.	33
4.7	Ajustement de l'équation (4.11) sur un signal sortant du préamplificateur.	36
4.8	Distribution d'amplitude pour des particules de la radiation ambiante, obtenue par l'ajustement de l'équation (4.11) sur les signaux sortant du préamplificateur.	36
4.9	Distribution du paramètre a de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.	37

4.10	Distribution du paramètre b de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.	37
4.11	Distribution du paramètre τ_1 de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.	38
4.12	Distribution du paramètre τ_2 de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.	38
4.13	Distribution du paramètre τ_3 de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.	39
4.14	Distribution des χ^2 par degré de liberté, ou $\widetilde{\chi^2}$, pour l'ajustement de l'équation (4.11) sur les signaux sortant du préamplificateur.	39
4.15	Distribution du temps d'atteinte du pic pour les signaux sortant du préamplificateur.	40
4.16	Distribution de l'amplitude selon le temps d'atteinte du pic pour des signaux sortant du préamplificateur.	42
4.17	Distribution de l'amplitude selon le ratio b/a pour des signaux sortant du préamplificateur.	42
4.18	Distribution du temps d'atteinte du pic selon le ratio b/a pour des signaux sortant du préamplificateur.	43
4.19	Filtre passe-haut à gauche et filtre passe-bas à droite. Les signaux entrent par la gauche et sortent à droite.	45
4.20	Filtre passe-bas de Sallen-Key de second ordre, où un signal entre à gauche et sort à droite [8].	46
4.21	Exemple de signal, produit par un générateur d'impulsions, sortant du <i>shaper</i> , après numérisation par un ADC et enregistré par le DAQ.	47
4.22	Distribution des temps de montée pour des signaux produits par un générateur d'impulsions, à la sortie du <i>shaper</i> , après numérisation par un ADC.	48
4.23	Distribution des temps de montée pour des signaux en provenance du ICs, à la sortie du <i>shaper</i> , après numérisation par un ADC.	48

4.24	Montage pour déterminer la conversion de l'amplitude d'un signal qui entre dans le <i>shaper</i> vers celle d'un signal qui en sort.	49
4.25	Schéma d'un diviseur de tension.	49
4.26	Amplitude qui sort du <i>shaper</i> selon l'amplitude qui y entre.	50
4.27	Représentation schématique du traitement numérique d'un signal par le DSP.	56
4.28	Extraction du paramètre A par la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence obtenues avec la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	58
4.29	Extraction du paramètre B par la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence obtenues avec la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	58
4.30	Paramètre t_{0_2} dans la région d'ajustement optimal avec la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	59
4.31	Extraction du paramètre P par la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence obtenues avec la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	59
5.1	Montage pour déterminer les formes de référence pour des signaux produits par un générateur d'impulsions.	63
5.2	Distribution d'amplitudes des points dans la région du piédestal pour des signaux produits par un générateur d'impulsions.	64
5.3	Distribution d'amplitudes maximales pour l'ajustement autour du sommet d'un polynôme de puissance 2 sur des signaux produits par un générateur d'impulsions.	66

5.4	Distribution du décalage dans le temps pour des signaux produits par un générateur d'impulsions.	66
5.5	Formes de référence normalisées moyennes expérimentales pour des impulsions produites par un générateur.	67
5.6	Exemple de l'ajustement de l'expression (5.1) des filtres actifs sur un signal produit par un générateur d'impulsions, après numérisation par l'ADC.	69
5.7	Distribution d'amplitudes pour l'ajustement de l'expression (5.1) des filtres actifs sur des signaux en provenance d'un générateur d'impulsions.	70
5.8	Distribution du paramètre A_{amp} de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.	71
5.9	Distribution du paramètre A_s de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions. . .	72
5.10	Distribution du paramètre τ_s de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions. . .	72
5.11	Distribution du paramètre $1/\beta$ de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.	73
5.12	Distribution du paramètre $1/\omega_0$ de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.	73
5.13	Distribution du paramètre $1/\omega_d$ de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.	74
5.14	Distribution des $\widetilde{\chi^2}$ individuels pour l'ajustement de l'expression (5.1) des filtres actifs sur des signaux en provenance d'un générateur d'impulsions.	74
5.15	Formes de référence normalisées pour l'expression (5.1) des filtres actifs sur des signaux en provenance d'un générateur d'impulsions.	75
5.16	Exemple de l'ajustement d'une somme de 3 gaussiennes sur un signal produit par un générateur d'impulsions, après numérisation par l'ADC.	75

5.17	Distribution d'amplitudes pour les ajustements individuels d'une somme de 3 gaussiennes sur des signaux en provenance d'un générateur d'impulsions.	77
5.18	Distribution du paramètre A_1	78
5.19	Distribution du paramètre A_2	78
5.20	Distribution du paramètre A_3	79
5.21	Distribution du paramètre μ_1	79
5.22	Distribution du paramètre μ_2	80
5.23	Distribution du paramètre μ_3	80
5.24	Distribution du paramètre σ_1	81
5.25	Distribution du paramètre σ_2	81
5.26	Distribution du paramètre σ_3	82
5.27	Distribution de la différence $\mu_2 - \mu_1$	82
5.28	Distribution de la différence $\mu_3 - \mu_2$	83
5.29	Distribution des $\tilde{\chi}^2$ individuels pour l'ajustement d'une somme de 3 gaussiennes.	83
5.30	Formes de référence normalisées pour la somme de 3 gaussiennes.	84
5.31	Montage pour déterminer les formes de référence pour des signaux générés dans un cristal de ICs.	85
5.32	Distribution d'amplitudes des points dans la région du piédestal pour des signaux générés dans un cristal de ICs.	86
5.33	Distribution d'amplitudes estimées par ajustement d'un polynôme de puissance 2 sur des impulsions en provenance d'un cristal de ICs.	87
5.34	Distribution des décalages dans le temps estimés par ajustement d'un polynôme de puissance 2 sur des impulsions en provenance d'un cristal de ICs.	88
5.35	Formes de référence normalisées moyennes expérimentales pour des impulsions en provenance d'un cristal de ICs.	88
5.36	Exemple de l'ajustement de l'équation (5.1) sur un signal généré dans un cristal de ICs, après numérisation par l'ADC.	89

5.37	Distribution d'amplitudes maximales pour les ajustements de l'équation (5.1).	90
5.38	Distribution du paramètre A_{amp} pour des signaux en provenance d'un cristal de ICs.	91
5.39	Distribution du paramètre A_s pour des signaux en provenance d'un cristal de ICs.	91
5.40	Distribution du paramètre τ_s pour des signaux en provenance d'un cristal de ICs.	92
5.41	Distribution du paramètre $1/\beta$ pour des signaux en provenance d'un cristal de ICs.	92
5.42	Distribution du paramètre $1/\omega_0$ pour des signaux en provenance d'un cristal de ICs.	93
5.43	Distribution du paramètre $1/\omega_d$ pour des signaux en provenance d'un cristal de ICs.	93
5.44	Distribution des $\widetilde{\chi^2}$ individuels pour l'ajustement de l'expression (5.1) des filtres actifs sur des signaux en provenance d'un cristal de ICs.	94
5.45	Formes de référence normalisées pour l'expression (5.1) des filtres actifs sur des signaux en provenance d'un cristal de ICs.	94
5.46	Exemple de l'ajustement d'une somme de 3 gaussiennes sur un signal généré dans un cristal de ICs, après numérisation par l'ADC.	95
5.47	Distribution du paramètre A_1	96
5.48	Distribution du paramètre A_2	97
5.49	Distribution du paramètre A_3	97
5.50	Distribution du paramètre μ_1	98
5.51	Distribution du paramètre μ_2	98
5.52	Distribution du paramètre μ_3	99
5.53	Distribution du paramètre σ_1	99
5.54	Distribution du paramètre σ_2	100
5.55	Distribution du paramètre σ_3	100

5.56	Distribution des $\widetilde{\chi}^2$ individuels pour l'ajustement d'une somme de 3 gaussiennes.	101
5.57	Distribution d'amplitudes maximales pour les ajustements d'une somme de 3 gaussiennes.	101
5.58	Formes de référence normalisées pour la somme de 3 gaussiennes.	102
5.59	Exemple d'ajustement de l'équation (4.17b) sur un signal produit par un générateur d'impulsions, avec la forme moyenne expérimentale optimale et les paramètres A , B et P correspondants. . . .	104
5.60	Distribution du piédestal, évalué avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.	105
5.61	Distribution d'amplitudes évaluées avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.	105
5.62	Distribution du ratio d'amplitudes aux première et deuxième évaluations, évalué avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.	106
5.63	Distribution de t_{0_2} , évalué avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.	106
5.64	Distribution du $\widetilde{\chi}^2$ individuel, évalué avec la forme moyenne expérimentale optimale pour chacun des signaux produits par un générateur d'impulsions.	107
5.65	Exemple d'ajustement de l'équation (4.17b) sur un signal produit par un générateur d'impulsions, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir de l'expression (5.1) des filtres actifs.	108
5.66	Distribution du piédestal, évalué avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.	108

5.67	Distribution d'amplitudes évaluées avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.	109
5.68	Distribution du ratio d'amplitudes aux première et deuxième évaluations, évalué avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.	109
5.69	Distribution de t_{0_2} , évalué avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.	110
5.70	Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de l'expression (5.1) des filtres actifs pour chacun des signaux produits par un générateur d'impulsions.	110
5.71	Exemple d'ajustement de l'équation (4.17b) sur un signal produit par un générateur d'impulsions, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir d'une somme de 3 gaussiennes.	111
5.72	Distribution du piédestal, évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	112
5.73	Distribution d'amplitudes évaluées avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	112
5.74	Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenue avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	113
5.75	Distribution de t_{0_2} , évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.	113

5.76	Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de la somme de 3 gaussiennes pour chacun des signaux produits par un générateur d'impulsions.	114
5.77	Exemple d'ajustement de l'équation (4.17b) sur un signal généré dans un cristal de ICs, avec la forme moyenne expérimentale optimale et les paramètres A , B et P correspondants.	116
5.78	Distribution du piédestal, évalué avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.	116
5.79	Distribution d'amplitudes évaluées avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.	117
5.80	Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenu avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.	117
5.81	Distribution de t_{02} , évalué avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.	118
5.82	Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme moyenne expérimentale optimale pour chacun des signaux générés dans un cristal de ICs.	118
5.83	Exemple d'ajustement de l'équation (4.17b) sur un signal généré dans un cristal de ICs, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir de l'expression (5.1) des filtres actifs.	119
5.84	Distribution du piédestal, évalué avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.	120
5.85	Distribution d'amplitudes évaluées avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.	120

5.86	Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenue avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.	121
5.87	Distribution de t_{0_2} , évalué avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.	121
5.88	Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de l'expression (5.1) des filtres actifs pour chacun des signaux générés dans un cristal de ICs.	122
5.89	Exemple d'ajustement de l'équation (4.17b) sur un signal généré dans un cristal de ICs, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir d'une somme de 3 gaussiennes.	123
5.90	Distribution du piédestal, évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.	123
5.91	Distribution d'amplitudes évaluées avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.	124
5.92	Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenue avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.	124
5.93	Distribution de t_{0_2} , évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.	125
5.94	Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de la somme de 3 gaussiennes pour chacun des signaux générés dans un cristal de ICs.	125

IV.1	Montage pour sélectionner des muons cosmiques par coïncidence entre un scintillateur de plastique et un cristal de ICs.	xxxii
IV.2	Temps qui sépare les signaux du plastique et du ICs pour les candidats au passage de muons cosmiques.	xxxiv
IV.3	Exemple d'empilement dû à la désintégration d'un muon selon la désintégration faible IV.1.	xxxv
IV.4	Exemple de signal lissé, pour un empilement dû à la désintégration d'un muon selon la désintégration faible IV.1.	xxxvi

LISTE DES ANNEXES

Annexe I :	Régression linéaire	xxiv
I.1	Cas particulier d'une droite	xxv
Annexe II :	Algorithme de Levenberg-Marquardt	xxvi
II.1	Problème à résoudre	xxvi
II.2	Méthode de Gauss-Newton	xxvi
II.3	Méthode du gradient	xxvii
II.4	Algorithme de Levenberg-Marquardt	xxviii
II.5	Critères de convergence	xxix
Annexe III :	Énergie d'un muon produit par désintégration du pion	xxxii
Annexe IV :	Temps de vie et masse du muon	xxxiii
IV.1	Coïncidences	xxxiii
IV.2	Temps de vie du muon	xxxiii
IV.3	Masse du muon	xxxvii
Annexe V :	Calcul d'incertitudes	xxxix
Annexe VI :	Coefficients pour les formes de référence	xl
VI.1	Générateur d'impulsions	xl
VI.1.1	Formes moyennes expérimentales	xl
VI.1.2	Filtres actifs	xlvii
VI.1.3	Somme de 3 gaussiennes	liv
VI.2	ICs	lx
VI.2.1	Formes moyennes expérimentales	lx
VI.2.2	Filtres actifs	lxvii
VI.2.3	Somme de 3 gaussiennes	lxxiii

Annexe VII :	Code pour la lecture des fichiers binaires de l'oscilloscope LeCroy	lxxix
Annexe VIII :	Code pour l'ajustement non-linéaire d'une somme de 3 gaussiennes sur des signaux produits par un générateur d'impulsions	lxxxv
Annexe IX :	Code pour l'extraction des paramètres A, B et P avec la méthode du DSP	cvii

REMERCIEMENTS

J'ai eu la chance de travailler avec trois physiciens pour qui j'ai un grand respect. Tout d'abord, je suis reconnaissant envers Paul Taras, mon directeur de recherche, pour m'avoir donné l'opportunité de réaliser un de mes grands objectifs de vie. Ce que je retiens de vous, c'est l'importance de la rigueur à chaque étape d'un projet. Je veux remercier Jean-Pierre Martin, une véritable encyclopédie, avec qui j'ai découvert à quel point l'électronique est un domaine riche et intéressant. Je veux aussi remercier Nikolai Starinsky, pour son expertise approfondie dans le domaine expérimental et pour toutes les discussions stimulantes et réalistes sur le domaine de la physique.

Je veux aussi remercier mes correcteurs Viktor Zacek et Sjoerd Roorda, pour leurs commentaires pertinents et leur patience.

J'ai une pensée pour Triumph, où j'ai eu l'occasion de passer une semaine très stimulante. J'ai adoré l'expérience.

Enfin, je me dois de remercier ma famille pour leur présence et leurs encouragements. C'est un peu grâce à vous si je me suis rendu jusqu'ici.

CHAPITRE 1

INTRODUCTION

L'expérience Belle II est le prolongement naturel de Belle, expérience à forte luminosité qui s'est déroulée au Japon de 1999 à 2009 avec l'accélérateur KEKB et qui exploitait la collision de faisceaux d'électrons et de positrons asymétriques en énergie, soit 8,0 GeV et 3,5 GeV, respectivement. Cette expérience détient le record de la plus grande luminosité produite, soit $2 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ pour une luminosité intégrée de 1000 fb^{-1} , tandis que pour Belle II on prévoit augmenter ce pic record d'un facteur 40 environ pour atteindre $8 \times 10^{35} \text{ cm}^{-2} \text{ s}^{-1}$ [6]. Cette augmentation implique davantage de particules à détecter et à distinguer les unes des autres, mais aussi un bruit de fond plus important. Il en découle un problème d'empilement des signaux générés dans les détecteurs par les particules qui les traversent. En 2016, après améliorations et ajustements techniques pour pallier à ces difficultés, l'accélérateur pourra rentrer en fonction sous le nom de Super-KEKB, avec des faisceaux un peu plus symétriques de 7 GeV et 4 GeV, respectivement [6].

Le type de détecteur concerné par ce travail est un calorimètre électromagnétique, conçu dans l'optique d'une expérience à haut taux de comptage. Il y a deux approches possibles pour résoudre le problème d'empilement. L'une d'elles consiste à modifier les circuits électroniques pour que les signaux qui en ressortent aient un temps de montée et un temps de descente plus courts, ainsi qu'une réduction du temps de latence/temps mort dû au traitement et au transfert des signaux. L'autre méthode est d'utiliser un scintillateur dont le temps de scintillation est plus rapide. Pour Belle, des cristaux d'iodure de césium dopés au thallium (ICs(Tl)) ont été utilisés. Même si Belle II réutilisera ces cristaux, il semble que des cristaux d'iodure de césium pur (ICs) permettraient de concevoir un calorimètre électromagnétique plus adapté pour des expériences à taux de comptage élevé [6]. Ainsi, ce travail se concentrera sur l'optimisation du traitement numérique de signaux générés dans un cristal de ICs pur. Le traitement numérique en tant que tel est un ajustement de formes de référence sur les signaux électroniques obtenus. Il s'agit donc

de définir des formes de référence fiables pour les signaux générés dans des ICs purs.

Nous aborderons au chapitre 2 les motivations physiques de Belle II, soit l'étude des saveurs lourdes et de la violation CP, ainsi qu'aller au-delà du Modèle Standard. Au chapitre 3, nous ferons un survol de l'accélérateur SuperKEKB et de l'ensemble des détecteurs utilisés pour l'expérience. Nous décrirons au chapitre 4 comment des signaux lumineux sont générés dans les ICs purs, puis convertis en signaux électroniques qui sont amplifiés. On expliquera ensuite le rôle du *shaper* pour la mise en forme de ces signaux avant leur numérisation par un ADC, ainsi que la méthode utilisée par le DSP pour en faire le traitement numérique. Enfin, nous déterminerons les formes de référence au chapitre 5, de trois façons différentes. Nous verrons aussi comment les formes de référence sont utilisées pour extraire les mesures que l'on obtient d'un calorimètre électromagnétique.

CHAPITRE 2

MOTIVATIONS PHYSIQUES DE BELLE II

Les accélérateurs KEKB et SuperKEKB sont des usines à mésons B^1 , où la production de mésons contenant des quarks ou antiquarks bottom (b ou \bar{b}) est favorisée. En effet, avec des faisceaux d'électrons et de positrons de 8,0 GeV et 3,5 GeV respectivement pour Belle ou 7 GeV et 4 GeV pour Belle II, l'énergie dans le centre de masse est de 10,58 GeV pour les deux expériences. Ceci mène à la production de la résonance $\Upsilon(4S)$, composée d'une paire de quarks $b\bar{b}$ et qui se désintègre ensuite en mésons $B^0\bar{B}^0$ ou B^+B^- [6][23].

2.1 Étude des saveurs lourdes

Un des objectifs de l'expérience Belle II est l'étude des saveurs lourdes, d'où le choix de la résonance Υ dans le but d'étudier plus en détails la troisième génération de quarks. L'avantage de produire des quarks b plutôt que des quarks top (t) est que ces derniers ont un temps de vie très court et une masse beaucoup plus élevée [6][23]. Ainsi, il faut une quantité d'énergie beaucoup plus grande pour les produire, mais surtout ils changent de saveurs par interaction faible avant même d'avoir pu interagir par interaction forte, ce qui n'est pas le cas des quarks b . Lorsqu'un quark b se désintègre par interaction faible en émettant un boson W^\pm , il en ressort un quark charmé (c), soit le quark le plus lourd de la deuxième génération, dont on peut alors étudier les propriétés. L'étude des saveurs lourdes couvre également la physique du lepton τ [6][23].

2.2 Violation CP et symétries

L'asymétrie entre la matière et l'antimatière dans l'Univers s'explique en partie par la violation CP, qui est possible dans le Modèle Standard par les phases complexes dans

¹*B-factories*

les matrices unitaires CKM et PMNS² [6][23]. Ces matrices décrivent les probabilités d'interactions entre saveurs pour les quarks et les neutrinos, respectivement. Si la phase complexe a été mesurée avec précision pour les quarks, la violation CP n'a pas encore été observée pour les neutrinos [6][23]. De plus, l'unitarité de la matrice CKM équivaut à un triangle dans le plan complexe, illustré sur la figure 2.1 avec les contraintes obtenues expérimentalement. On voudrait aussi savoir pourquoi cette matrice a une structure presque diagonale, ce qui pourrait être dû à une symétrie de saveurs inconnues [6][23].

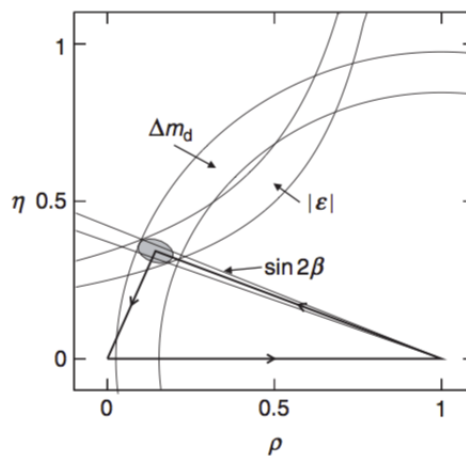


Figure 2.1 : Triangle d'unitarité de la matrice CKM et les contraintes expérimentales sur celui-ci [23].

L'étude de la violation CP peut se faire à KEKB et son successeur grâce à l'interaction forte qui brise le Y et produit une paire de mésons électriquement neutres qui incluent le quark down (d), soient B^0 ($d\bar{b}$) et \bar{B}^0 ($b\bar{d}$) [6][23]. Ces deux mésons sont interchangeables par conjugaison de charge. Ainsi, par intrication quantique au moment de leur formation, il est impossible de les distinguer. Cependant, le premier qui se désintègre peut le faire entre autres selon un processus interdit à l'autre méson, ce qui permet d'identifier lequel s'est désintégré et lequel reste [6][23]. La figure 2.2 montre un

²CKM : Cabibbo-Kobayashi-Maskawa ; PMNS : Pontecorvo-Maki-Nakagawa-Sakata.

exemple d'un tel processus, soient les interactions faibles :

$$B^0 \rightarrow D^- + \mu^+ + \nu_\mu \quad (2.1a)$$

$$\bar{B}^0 \rightarrow D^+ + \mu^- + \bar{\nu}_\mu \quad (2.1b)$$

où l'identification du méson désintégré se fait avec la charge du muon produit. Les mésons D^- et D^+ sont constitués des paires de quarks $d\bar{c}$ et $c\bar{d}$, respectivement [23]. Cependant, il existe deux mécanismes principaux faisant intervenir des quarks t virtuels

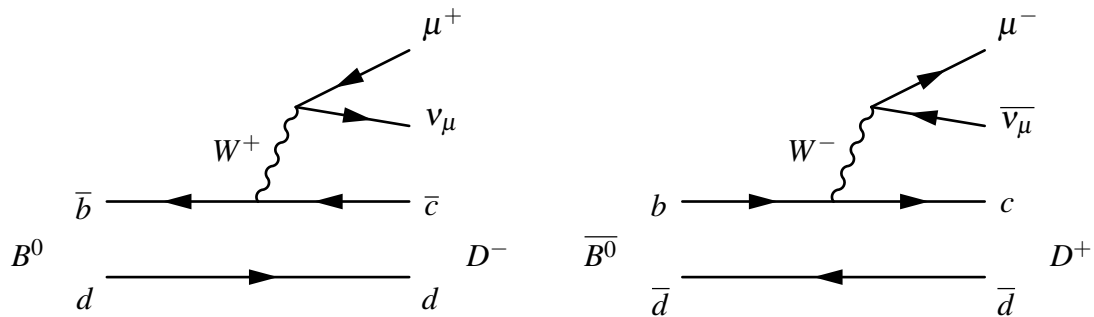


Figure 2.2 : Diagrammes de Feynman pour les désintégrations (2.1a) et (2.1b), qui permettent de distinguer les mésons B^0 ($d\bar{b}$) et \bar{B}^0 ($b\bar{d}$) par la charge du muon produit [23].

permettant aux mésons d'osciller d'une forme à l'autre, comme il est montré à la figure 2.3. En principe, des diagrammes avec des quarks u ou c virtuels pourraient aussi être considérés, mais comme l'élément V_{tb} de la matrice CKM est dominant pour les quarks b , les processus avec des quarks t virtuels sont les plus significatifs [23]. Le méson res-

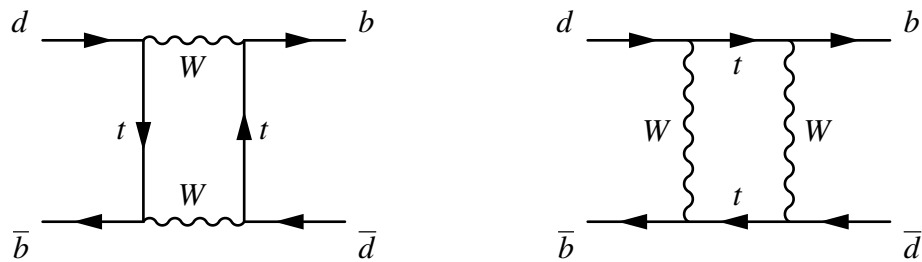


Figure 2.3 : Deux mécanismes de mélange permettant aux mésons B^0 ($d\bar{b}$) et \bar{B}^0 ($b\bar{d}$) d'osciller d'une forme à l'autre [23].

tant peut donc osciller et se désintégrer malgré tout suivant le même processus que le premier. La violation CP s'observe par cette oscillation qui se fait plus facilement dans

un sens que dans l'autre, c'est-à-dire que l'on observe les produits d'un des mésons plus souvent que ceux de l'autre. On peut aussi observer la violation CP par des interférences entre des états finaux communs aux deux mésons [6][23].

2.3 Modèle Standard et nouvelle physique

Dans le cadre du Modèle Standard (MS), il importe de déterminer expérimentalement les couplages et les paramètres de mélange des bosons et des fermions, ainsi que leurs masses. Ces résultats permettent de contraindre davantage le MS et d'avoir des valeurs plus précises de ses paramètres, d'autant plus qu'il compte trop de paramètres libres [6][23]. En effet, il y a 12 paramètres de couplage de Yukawa entre les fermions et le champ de Higgs, autrement dit les masses des quarks et des leptons. Il y a aussi 3 constantes de couplage liées aux jauges d'interactions, 2 paramètres pour décrire le potentiel de Higgs et 8 angles de mélange dans les matrices CKM et PMNS, ce qui donne 25 paramètres libres [23] dans le MS. Un paramètre supplémentaire peut apparaître dans le Lagrangien en chromodynamique quantique (QCD) et qui est lié à la violation CP en interaction forte. Cependant, il est souvent négligé du fait de sa valeur extrêmement faible [23].

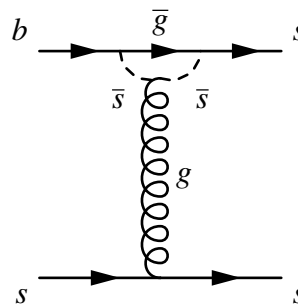


Figure 2.4 : Exemple de processus au-delà du Modèle Standard, $b \rightarrow \bar{s}s$, prédit par la SuperSymétrie, avec un squark gluino-down \bar{g} [6].

Aussi, les motivations de Belle II dépendent de l'observation d'un plus grand nombre d'événements pour avoir un meilleur ensemble statistique [6]. On gagne alors en précision sur les rapports d'embranchement des différentes réactions possibles, plus particulièrement les processus rares et difficiles à distinguer. On souhaite aussi observer des processus prédits par des modèles au-delà du MS ou interdits par celui-ci. Par exemple,

la figure 2.4 présente le processus $b \rightarrow s\bar{s}s$, prédit par la SuperSymétrie avec l'apparition d'un squark gluino-down \bar{g} [6]. De plus, si on trouve des déviations par rapport au MS, ce pourrait être le signe d'une nouvelle physique [6].

CHAPITRE 3

SUPERKEKB ET DÉTECTEURS

L'accélérateur de particules qu'utilisera Belle II a la forme d'un carré aux coins arrondis, c'est-à-dire 4 sections linéaires de 540 m de long reliés par 4 arcs, pour une longueur totale de 3016 m. Ces sections se situent géographiquement à Tsukuba, Fuji, Oho et Nikko au Japon. Quant à l'ensemble de détecteurs, il se trouve à Tsukuba et mesure 10 m de long par 10 m de haut pour un poids de 1500 tonnes [6]. L'expérience consiste en l'étude des produits de désintégration de collisions électron-positron, avec des énergies de 7 et 4 GeV, respectivement. Chaque faisceau circule dans son propre anneau, que l'on désigne par HER pour les électrons et LER pour les positrons¹, avec 2503 paquets de particules par anneau, comparativement à 1584 pour Belle. De plus, le courant produit par le passage des électrons et des positrons dans les anneaux passe de 1,64 à 3,6 A dans le HER et de 1,19 à 2,6 A dans le LER [6].

3.1 Production des faisceaux

Tout d'abord, une cathode froide en cuivre de 7 mm² est irradiée par de courtes impulsions lasers dont la longueur d'onde est de 205 nm [6]. On génère ainsi des électrons par effet photoélectrique, sous la forme d'impulsions d'une longueur de l'ordre de 10 ps. La petite taille de la photocathode a pour but de réduire l'émission de ces électrons, qu'on accélère à l'aide de cavités radio-fréquences (RF) [6]. Deux préinjecteurs sont utilisés, soit un par anneau. Les électrons qui sont destinés au HER sont accélérés jusqu'à 7 GeV, avec une charge de 5 nC pour 2 paquets [6]. Pour le LER, des électrons dont l'énergie est de 4 GeV sont dirigés vers une cible de tungstène pour produire des positrons. Avec une charge de 10 nC pour 2 paquets d'électrons, on obtient 4 nC pour 2 paquets de positrons. Ces derniers sont alors accélérés à 4 GeV avant d'être injectés dans le LER [6]. L'injection dans le HER et le LER se fait en alternance, impulsion

¹HER et LER : *High and Low Energy Rings* ou Anneaux à haute et faible énergie

par impulsion, afin de maintenir les courants de faisceaux constants [6]. Aussi, comme SuperKEKB aura des courants plus élevés et que l'effet Touschek (voir section 3.4.2) raccourcira grandement le temps de vie des faisceaux (10 minutes), l'injection devra être plus intense [6].

3.1.1 Contrôle de la trajectoire des faisceaux

La trajectoire des faisceaux est contrôlée par un ensemble d'aimants dipolaires qui appliquent une force magnétique sur les particules permettant de les diriger. Ces aimants passeront de 5,9 à 3,8 m pour le HER, ce qui laissera plus d'espace pour d'autres dispositifs et permettra d'installer davantage d'aimants pour un meilleur contrôle du faisceau dans les arcs [6]. Dans le LER, ces aimants gagneront en puissance en passant de 0,89 à 4 m [6].

Si Belle adoptait une approche en haute intensité, Belle II optera en plus pour la méthode des nano-faisceaux [6]. Plutôt que de simplement maximiser le nombre de particules par paquet comme auparavant, on cherchera à condenser ces particules dans un paquet aussi petit que possible. Cette densité plus élevée accroît la possibilité de collisions quand les paquets se croisent dans la région d'interaction [6]. Pour générer des nano-faisceaux, des aimants quadrupolaires assurent la focalisation des paquets du faisceau [6]. Cependant, il en résulte un étalement en énergie des particules. Des aimants sextupolaires ont pour but de maintenir la chromaticité des faisceaux, quoique des effets non-linéaires doivent être considérés [6].

Des aimants onduleurs², constitués d'aimants dipolaires, serviront à atténuer les oscillations betatron et synchrotron. En fait, la période de ces oscillations sera diminuée de moitié [6].

3.1.2 Anneau d'amortissement

Les positrons étant des particules secondaires produites par collision sur une cible, leur émittance de $2,1 \mu\text{m}$ est plus grande que celle des électrons et même trop pour

²*Wiggler magnets*

le concept de nano-faisceaux qui requiert 4 nm [6]. C'est pourquoi on utilise un anneau d'amortissement afin de réduire l'émittance des positrons par radiation synchrotron, pour ensuite les accélérer jusqu'à 4 GeV et les injecter dans le LER [6]. De plus, cet anneau assure une certaine monochromaticité du faisceau, puisque l'étalement en énergie est réduit de $\pm 5\%$ à $\pm 1,5\%$ [6].

3.2 Luminosité

La luminosité peut être calculée avec l'équation suivante [6] :

$$L = \frac{\gamma_{\pm}}{2er_e} \left(\frac{I_{\pm} \xi_{y\pm}}{\beta_{y\pm}^*} \right) \left(\frac{R_L}{R_{\xi_y}} \right) \left(1 + \frac{\sigma_y^*}{\sigma_x^*} \right) \quad (3.1)$$

L'indice \pm dénote un positron avec $+$ et un électron avec $-$, γ est le facteur de Lorentz (environ 13699 pour le HER et 7828 pour le LER), e est la charge élémentaire, $r_e = 2,8179 \times 10^{-15}$ m est le rayon classique de l'électron dans un atome [4], I est le courant dans l'anneau (2,62 et 3,60 A), ξ_y est l'émittance en y (0,088 et 0,090), β_y^* est la fonction beta liée à l'optique du faisceau (0,41 et 0,27 mm) et σ_x^* (7,75 et 10,2 μm) et σ_y^* (59 et 59 nm) sont les tailles du faisceau en x et en y [6]. Les astérisques signifient qu'on utilise les valeurs au point d'interaction. Quant à R_L et R_{ξ_y} , ce sont des facteurs de réduction de luminosité en lien avec les paramètres d'interaction faisceau-faisceau et leur ratio est près de l'unité [6]. De plus, le facteur γ_{\pm} aura pratiquement la même valeur que dans Belle, puisque la vitesse des particules est environ $\beta = v/c \approx 1$ dans les deux cas.

Dans l'optique de la méthode des nano-faisceaux, la fonction β_y^* verticale sera comprimée au point d'interaction en minimisant la taille longitudinale $d = \frac{\sigma_x^*}{\phi}$ des régions qui se recouvrent [6], illustrée à la figure 3.1. Pour minimiser d autant que possible, il est préférable de réduire σ_x^* , car plus l'angle ϕ entre les faisceaux est grand, plus il est difficile d'obtenir le paramètre faisceau-faisceau ξ_y prévu pour Belle II [6]. Avec un angle ϕ de 41,5 mrad, σ_x^* , σ_y^* et σ_z doivent respectivement être de l'ordre des μm , des nm et des mm [6]. Ce grand angle, en ajoutant de l'espace entre les conduits au point

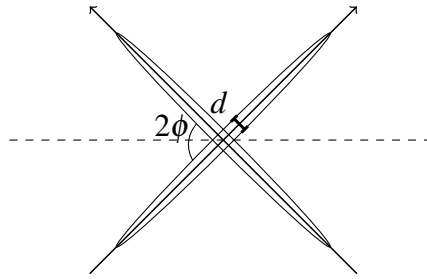


Figure 3.1 : Définition de la taille longitudinale d et de l'angle de croisement 2ϕ entre les paquets des faisceaux. L'axe du cylindre de détecteurs est représenté en pointillés.

d'interaction, permettra aussi aux aimants quadrupolaires servant au focus final d'être indépendants pour chaque faisceau. Ceci réduira le bruit dû à la radiation synchrotron. De plus, ces aimants pourront être plus près du point d'interaction, ce qui élargit l'ouverture dynamique [6]. Si celle-ci est étroite, le temps de vie du faisceau est raccourci par effet Touschek, soit la diffusion et la perte de particules chargées dans un anneau de stockage [6].

Étant donné que par rapport à Belle la fonction β_y^* sera diminuée d'un facteur 20, que les courants des faisceaux seront doublés et que l'émittance ξ_y aura sensiblement la même valeur, on en déduit de l'équation (3.1) que la luminosité augmentera d'un facteur 40 par rapport à Belle pour atteindre $8 \times 10^{35} \text{ cm}^{-2}\text{s}^{-1}$ [6].

Les cavités de crabe sont une contribution astucieuse de Belle qui a augmenté la luminosité de 15% environ et des simulations prévoient une augmentation de 10% pour Belle II [6]. Étant donné que, dans la région d'interaction, les faisceaux se croisent selon un angle, du point de vue du référentiel du centre de masse les paquets se rencontrent selon le même angle, tel qu'illustré sur la figure 3.2. Les cavités de crabe ont pour but de faire subir une rotation aux paquets afin qu'ils se rencontrent face à face [23]. Ainsi, le volume instantané de la région où les paquets se superposent est en moyenne plus grand, ce qui augmente la probabilité de collisions. Cependant, ces cavités ne sont pas nécessaires pour atteindre les objectifs de Belle II, sans compter certains effets non-linéaires qu'elles apportent [6]. Elles pourront donc servir de solution de rechange ou d'amélioration.

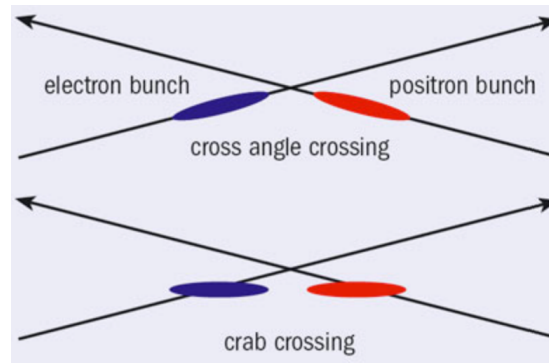


Figure 3.2 : En haut, les paquets de particules se croisent selon un angle. En bas, les cavités de crabe font subir une rotation à ces paquets afin qu’ils se présentent de face [23].

3.3 Région d’interaction

Les anneaux HER et LER se croisent dans la région d’interaction pour permettre la rencontre des particules. Si l’angle 2ϕ entre les faisceaux était de 22 mrad pour Belle dans cette région, il sera de 83 mrad avec Belle II [6]. L’augmentation de cet angle réduit les interactions entre les faisceaux. Cela permet aussi d’installer les aimants quadrupolaires pour le focus final le plus près possible du point d’interaction afin d’assurer que les paquets de particules soient aussi petits que possible au moment du croisement [6]. Il y aura 5 aimants quadrupolaires supraconducteurs (QC1LP, QC1LE, QC1RP, QC1RE et QC2RP)³ et 3 permanents (QC2LP, QC2LE et QC2RE). Les aimants QC1 appliquent un focus vertical, tandis que les QC2 s’occupent de l’axe horizontal [6]. Les aimants QC1LP et QC1RP sont les plus rapprochés du point d’interaction et profitent de 4 types de bobines correctrices : des dipôles normaux et des dipôles de travers pour aligner les axes de leurs champs magnétiques, des quadrupôles de travers pour aligner leurs plans médians et enfin des octupôles pour améliorer l’ouverture dynamique [6]. Puisque les flux magnétiques des aimants QC1LP et QC1RP traversent le HER près de la région d’interaction, d’autres bobines correctrices ont pour but d’éliminer ce champ de fuite, qui peut nuire à la stabilité du faisceau d’électrons du HER. En tout, 32 bobines correctrices supraconductrices seront utilisées [6].

³L et R : gauche (*left*) et droite (*right*). E et P : électron et positron

Pour éviter que les aimants quadrupolaires supraconducteurs surchauffent, ils sont contenus dans un cryostat, avec deux solénoïdes de compensation pour compenser les 1,5 T générés par le solénoïde de Belle II [6]. Ces deux solénoïdes sont segmentés en petites bobines pour limiter les effets de bord, qui augmentent l'émittance verticale. Ainsi, chacune de ces bobines peut être placée au plus près des faisceaux tout en ayant son axe aligné avec celui de la région d'interaction. De plus, le nombre de tours de fil diminue graduellement à mesure qu'on s'éloigne du point d'interaction pour adoucir le gradient au bord de chaque solénoïde [6].

Dans la région d'interaction, la surface interne des tuyaux qui conduisent les faisceaux vers le point d'interaction a un diamètre de 2 cm et est plaquée d'une couche d'or de 50 μm d'épaisseur pour limiter la contribution de la paroi en contaminants due aux collisions [6]. Les tuyaux sont dans un cryostat et sont faits de cuivre sans oxygène⁴, recouvert d'une couche d'acier inoxydable à l'extérieur [6]. Une petite couche isolante de vide sépare ensuite ces tuyaux de l'hélium liquide [6]. Il est essentiel d'avoir dans ces tuyaux des unités de soufflerie⁵ qui permettent des mouvements indépendants les uns des autres et qui atténuent les erreurs d'ingénierie à la fabrication, ainsi que les déformations à l'assemblage et dues à l'opération de la machine [6]. Des pompes pour la pression locale permettent un meilleur vide dans la région d'interaction que globalement dans les anneaux [6]. Il importe aussi d'utiliser une structure facilitant le refroidissement pour éliminer la chaleur accumulée sur la paroi. Les ondes électromagnétiques produites par la structure non-uniforme des tuyaux constituent une source supplémentaire de chaleur, tandis que la pression interne est principalement due au gaz induit par la radiation synchrotron sur la paroi [6]. Des masques serviront à empêcher les photons diffusés d'atteindre le point d'interaction. Ainsi, la pression au point d'interaction est d'environ 6×10^{-5} Pa, tandis qu'aux pompes elle est de 6×10^{-6} Pa [6].

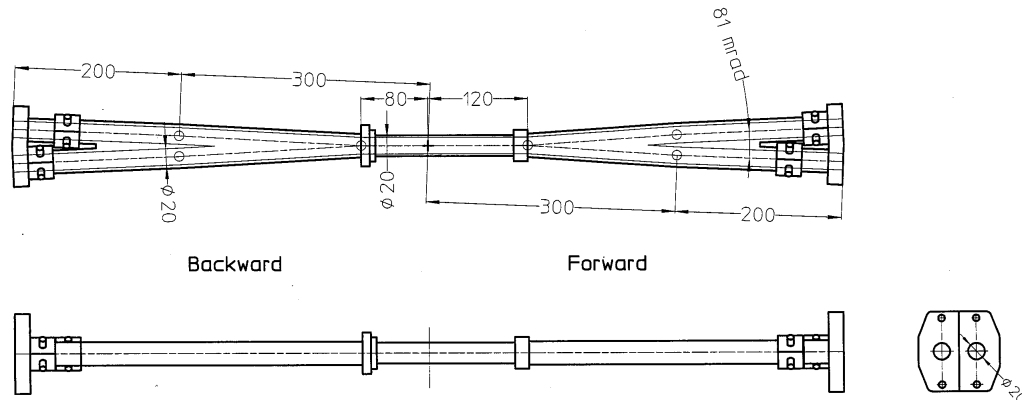


Figure 3.3 : Schéma du point d'interaction, vu du haut et de côté. Les électrons se dirigent vers l'avant et les positrons vers l'arrière. Les longueurs indiquées sont en mm [6].

3.3.1 Point d'interaction

La chambre du point d'interaction, illustrée sur la figure 3.3, a un rayon interne de 10 mm [6]. Les tuyaux de cuivre de la région d'interaction dirigent les particules vers le point d'interaction, dont les fourches sont faites de tantale, tandis que la partie centrale longue de 200 mm est faite d'une double épaisseur de béryllium, entre lesquelles circule le liquide de refroidissement. Le tantale et le béryllium ont été choisis pour limiter l'effet des gerbes de particules dues au bruit de fond [6]. Aussi, la fourche de la partie avant mesure 380 mm de long et celle à l'arrière mesure 420 mm, ce qui fait que la partie centrale est décalée de 20 mm vers l'avant. Ces dimensions reflètent l'asymétrie en énergie des électrons et des positrons, dont les produits de désintégration se dirigent majoritairement vers l'avant [6]. Si de l'eau est utilisée pour refroidir les fourches, le béryllium est quant à lui très réactif à cette substance. C'est pourquoi on choisit plutôt la paraffine pour refroidir la partie centrale. Des calculs prévoient des contributions de chaleur de 217 W/m pour le LER et de 149 W/m pour le HER. Puisque la partie centrale fait 20 cm de long, la contribution de chaleur y sera donc de 81 W. La capacité de chaleur de la paraffine étant de 2 J/K/g, elle doit circuler avec un flux de 300 cm³/min pour assurer des variations de température sous la barre de 10°C. Pour se donner une marge de manoeuvre, ce

⁴Oxygen free copper

⁵Bellows

flux sera de $1000 \text{ cm}^3/\text{min}$. De plus, le liquide de refroidissement devrait être turbulent pour maximiser l'efficacité d'échange de chaleur. [6] Aussi, des modes d'ordres élevés peuvent conduire à d'importants dépôts de chaleur en des endroits particuliers [6]. Enfin, comme les faisceaux sont très petits, il faut minimiser autant que possible les vibrations qui pourraient déstabiliser le faisceau [6].

3.4 Sources de bruit de fond

3.4.1 Radiation synchrotron

La radiation synchrotron est causée par l'accélération (centripète) des particules dans les anneaux, c'est-à-dire que la déviation d'une particule par la force magnétique induite par les aimants dipolaires entraîne une perte d'énergie sous forme de photons [23]. Ce phénomène se produit principalement dans les coins arrondis de l'accélérateur et les trajectoires plus tordues dans la région d'interaction qui assurent la rencontre des faisceaux. Ainsi, il est important de connaître cette radiation, autant pour assurer que chaque faisceau a la bonne énergie, mais aussi pour protéger les détecteurs à pixels (PXD) et les détecteurs de vertex au silicium, qui sont les détecteurs les plus près du point d'interaction [6]. Le rayonnement produit dépend de l'optique du faisceau, de la géométrie des composants de la région d'interaction et du courant dans le HER, l'anneau le plus énergétique [6]. Cette source de bruit est générée avant le point d'interaction (amont vers aval) [6].

Il y a une autre source de radiation synchrotron juste après le point d'interaction (depuis aval⁶) due aux aimants qui font la séparation des faisceaux [6]. SuperKEKB aura 2 aimants quadrupôlares de chaque côté dédiés à la séparation, tandis que KEKB utilisait le même aimant que pour le focus final du faisceau de positrons [6]. On s'attend à ce que la radiation produite soit 800 fois plus petite que pour KEKB [6].

⁶*Backscattering*

3.4.2 Diffusion Touschek

La diffusion Touschek est due aux interactions entre particules d'un même paquet du faisceau, soit une répulsion électrique entre des particules de même charge [6]. Cette diffusion change la quantité de mouvement des particules et certaines quittent le faisceau. Lorsqu'elles frappent la paroi, des gerbes électromagnétiques sont détectées et des atomes de la paroi peuvent être arrachés et contaminer le faisceau [6]. Ce phénomène dépend du courant d'un paquet du faisceau ⁷, du nombre de paquets et de l'inverse de la taille du faisceau [6]. La contribution du LER domine celle du HER, car la diffusion Touschek a une dépendance en E^{-3} sur l'énergie des particules et la densité de courant de paquet est plus grande dans le LER [6].

L'importance de cet effet est évaluée par la mesure du temps de vie du faisceau, qui est proportionnelle à sa taille. Puisque la taille des faisceaux sera diminuée d'un facteur 20 comparativement à Belle, on s'attend à ce que la diffusion Touschek soit la source principale de bruit de fond de SuperKEKB [6].

3.4.3 Diffusion faisceau-gaz

Dans les tuyaux, il est important que la pression soit aussi basse que possible. Même avec une pompe idéale, il est impossible de produire un vide parfait. La vitesse linéaire⁸ de fonctionnement des pompes sera de $0,1 \text{ m}^3\text{s}^{-1}\text{m}^{-1}$ en supposant que 1 molécule par 10^6 photons se détache de la paroi [6]. La pression visée et requise dans la région d'interaction est d'environ 10^{-7} Pa en moyenne [6]. Par ailleurs, les photons et électrons secondaires qui entrent en collision avec la paroi contribuent à générer un nuage d'électrons qui s'ajoutent à ce gaz résiduel. Les électrons de ce nuage qui sont en phase avec les cavités RF sont accélérés et peuvent générer davantage d'électrons secondaires. Les interactions entre les particules d'un faisceau et celles du gaz résiduel dans l'anneau correspondent à la diffusion de Coulomb et au bremsstrahlung [23]. La quantité de mouvement de particules étant modifiée, celles-ci frappent alors la paroi et produisent des gerbes électromagnétiques [6]. Cette diffusion dépend du courant de faisceau, de la

⁷*Beam bunch current*

⁸Les unités correspondent à un volume de gaz par unité de temps et par unité de longueur du tuyau.

pression dans les anneaux et de la puissance des aimants [6]. Pour Belle II, la pression dans une région de plus ou moins 2 m autour du point d'interaction sera de 100 à 1000 fois plus faible que pour Belle et les aimants de focalisation du faisceau seront beaucoup plus puissants [6].

3.4.4 Nuages d'électrons dans le LER

Les nuages d'électrons sont encore plus problématiques dans le LER, où ces particules peuvent s'annihiler avec des positrons et ainsi générer des photons qui s'ajoutent au bruit détecté [6]. Ces photons produisent aussi de nouveaux électrons secondaires par collision avec la paroi, ce qui entretient le nuage d'électrons tout en déstabilisant le faisceau. Des simulations prévoient une densité moyenne de 10^{11} m^{-3} pour ce nuage [6]. Dans l'anneau d'amortissement, on s'attend à une densité effective d'électrons de $0,7 \times 10^{14} \text{ m}^{-2}$ près du faisceau, ce qui respecte le seuil d'instabilités qui est de $1,57 \times 10^{14} \text{ m}^{-2}$ [6].

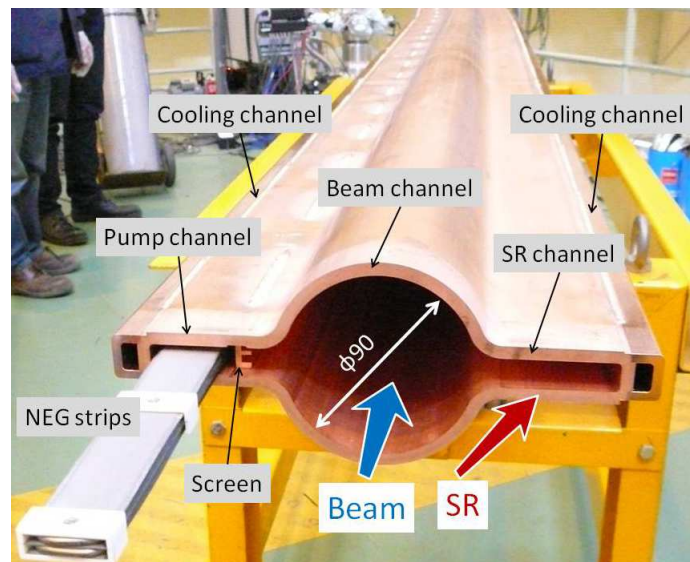


Figure 3.4 : Tuyau en cuivre avec des antichambres pour réduire l'importance des nuages d'électrons et de la radiation synchrotron [6].

Pour réduire cette densité, des antichambres sont ajoutées de part et d'autre du tuyau, telles qu'illustrées à la figure 3.4, afin d'y piéger les photoélectrons, qui sont les germes

du nuage, ainsi que diluer la radiation synchrotron [6]. Cependant, dans le régime de courant de paquet élevé⁹, les électrons secondaires ont plus d'importance dans l'entretien du nuage. Ainsi, les tuyaux seront généralement faits de cuivre, sauf dans les arcs et les chambres complexes et spéciales qui seront faits d'un alliage d'aluminium [6]. Les taux d'émission d'électrons secondaires de ces matériaux sont relativement bas. Comme cette mesure n'est pas suffisante, l'intérieur des tuyaux sera tapissé d'un matériau dont le taux d'émission est encore plus faible, par exemple du TiN, du NEG ou du graphite (ceci est indispensable pour des tuyaux faits d'un alliage d'aluminium) [6]. De plus, le champ magnétique généré par un solénoïde diminuera les possibilités de décharges auto-entretenues¹⁰ et ce de plusieurs ordres de grandeur. Par contre, cette dernière méthode ne peut être utilisée que dans des régions de dérive libres de champ [6]. Une autre approche est d'utiliser des électrodes de compensation¹¹ qui réduisent le taux d'émission d'électrons secondaires de façon géométrique [6]. Enfin, des rainures sur la paroi augmentent la surface d'absorption des électrons du nuage [6].

3.4.5 Radiation Bhabha

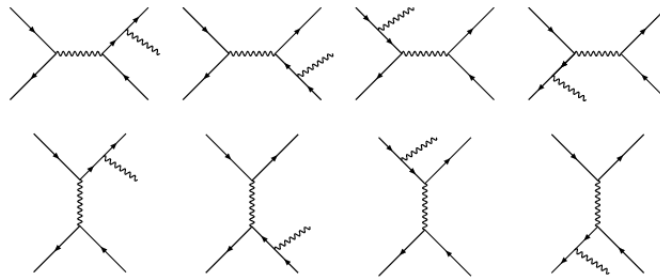


Figure 3.5 : Diagrammes de Feynman pour la diffusion Bhabha radiative.

La diffusion Bhabha radiative, dont le taux d'occurrence est proportionnel à la luminosité, est la diffusion d'un électron et d'un positron accompagné d'une émission de photon(s) [6], telle qu'illustrée sur la figure 3.5. Ceux-ci se propagent vers la paroi en suivant l'axe du faisceau et interagissent avec le fer des aimants, entraînant la production

⁹High bunch-current

¹⁰Multipactoring

¹¹Clearing electrodes

de neutrons par le mécanisme de résonance photo-nucléaire géante ¹² [6]. Ces neutrons sont la principale source de bruit dans le détecteur de K_L et de muons (KLM) [6].

SuperKEKB utilisera des aimants quadrupolaires afin de mieux centrer les faisceaux dans la région d'interaction, contrairement à KEKB où l'aimant utilisé dirigeait les particules diffusées vers les parois, ce qui produisait des gerbes. Ainsi, on s'attend ce que cette contribution au bruit soit réduite d'un facteur 40 [6].

3.4.6 Production de paires via 2 photons

Il se peut que la collision d'une paire électron-positron donne lieu à la production d'une nouvelle paire, par le processus $e^-e^+ \rightarrow e^-e^+e^-e^+$ via 2 photons virtuels [6]. Des simulations Monte-Carlo prévoient de 900 à 14000 paires produites dans chaque événement dans la première couche de détecteurs à pixel (PXD), jusqu'à 1,3 cm du point d'interaction [6]. Le taux de production de paires diminue avec la distance r depuis ce point selon $1/r^2$ [6]. Il est important de connaître ce bruit de fond afin de ne pas être aveuglé dès la première couche de PXD, puisque c'est le point d'entrée pour tous les scénarios de détection possibles [6].

3.4.7 Interactions faisceau-faisceau

Cet effet n'a d'importance que près du point d'interaction, où les faisceaux sont suffisamment près l'un de l'autre. Lorsque les faisceaux se rapprochent, les particules qui n'entrent pas en collision interagissent malgré tout entre elles. Chaque faisceau agit sur l'autre comme une lentille non-linéaire et en affecte l'optique du faisceau [6]. L'utilisation d'un angle de croisement permet de réduire l'importance de cet effet, c'est d'ailleurs pour cette raison que Belle a obtenu une luminosité 2 fois plus grande que l'expérience Babar. C'est aussi pourquoi Belle II utilisera un angle presque 4 fois plus grand que Belle, soit 41,5 mrad plutôt que 11 mrad [6].

¹²*Giant photo-nuclear resonance mechanism*

3.5 Détecteurs de Belle II

Différents types de détecteurs sont nécessaires afin de bien reconstruire le fil des événements après la collision initiale au point d'interaction. Ces détecteurs sont disposés selon un ordre bien précis autour de la région d'interaction en fonction de leurs rôles respectifs, suivant la forme d'un baril et incluant les embouts [6], tel que montré à la figure 3.6.

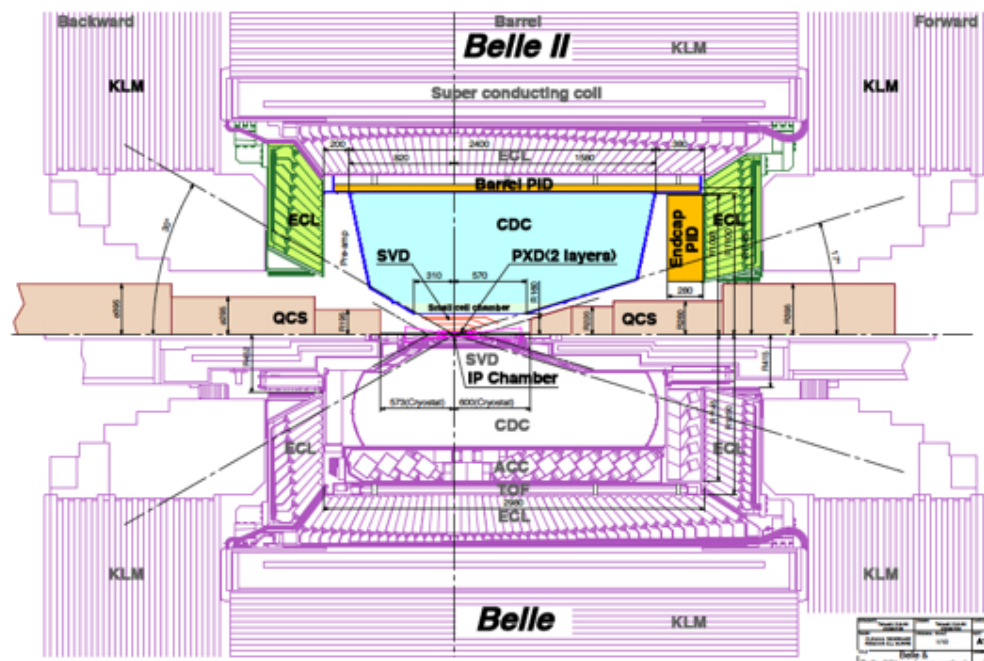


Figure 3.6 : Disposition des différents détecteurs utilisés dans Belle II en haut et Belle en bas. *IP Chamber* : Chambre du point d'interaction. PXD : Détecteurs à pixels. SVD : Détecteurs de vertex au silicium. CDC : Chambre à dérive centrale. PID : Détecteurs pour l'identification des particules. ECL : Calorimètres électromagnétiques. KLM : Détecteurs à K_L^0 et à muons. ACC : Compteurs en aérogel à effet Cherenkov. TOF : Compteurs de temps-de-vol. QCS : Aimants quadrupolaires pour la focalisation finale du faisceau [6].

Tout d'abord, il y a les détecteurs à pixels (PXD). Ceux-ci forment la première couche de détecteurs et permettent de déterminer le point de départ de la trajectoire d'une particule produite suite à la collision entre un électron et un positron [6]. Ensuite viennent plusieurs couches de détecteurs de vertex au silicium (SVD). Chaque couche

traversée permet de déterminer un des points de la trajectoire suivie, tandis que si la particule se désintègre, plusieurs trajectoires partiront du même point de départ, soit la position du vertex de désintégration [6]. Le détecteur suivant est la chambre à dérive centrale (CDC), qui utilise un champ magnétique pour dévier les trajectoires de particules possédant une charge électrique. Ainsi, on peut déterminer si la particule a une charge électrique positive ou négative selon que la trajectoire courbe dans une direction ou dans l'autre. De plus, cette courbure permet d'estimer la quantité de mouvement de la particule [6][11][23]. Après on a des détecteurs pour l'identification des particules (PID), qui précèdent les calorimètres électromagnétiques (ECL) dont le rôle est d'estimer l'énergie déposée par des photons et des électrons principalement, mais aussi n'importe quelle particule ayant une charge électrique non-nulle [6][23]. Enfin, les détecteurs de K_L^0 et de muons (KLM) sont constitués d'une série de plaques résistives séparées par des scintillateurs et utilisent le champ magnétique de retour du solénoïde de Belle II pour faire dévier les muons et estimer leur quantité de mouvement [6][23].

3.5.1 Calorimètre électromagnétique

Ce travail a pour but l'optimisation d'un calorimètre électromagnétique qui utilise un cristal d'iodure de césium pur (ICs). Cette section introduit la chaîne qui relie le passage d'une particule dans le détecteur jusqu'à l'extraction des données d'intérêt, telle qu'illustrée à la figure 3.7. Cette chaîne sera étudiée plus en détails au chapitre suivant.

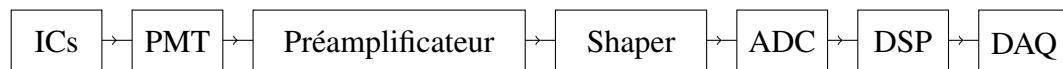


Figure 3.7 : Chaîne qui constitue le calorimètre électromagnétique.

Lorsqu'un électron, un positron, un photon de haute énergie ou un pion neutre (par sa désintégration électromagnétique en deux photons) traverse le ICs, une gerbe électromagnétique est produite. Elle est entretenue par le bremsstrahlung et les pertes d'énergie par ionisation pour les électrons et les positrons, tandis que les photons contribuent par la production de paires, l'effet Compton et l'effet photoélectrique [6][11][23]. Cette gerbe se développe jusqu'à ce que les électrons et les positrons ne puissent perdre leur énergie

que par collisions, en excitant les atomes du cristal, et que les photons se contentent de se propager dans le cristal [11][23]. Alors ces photons et ceux émis par la désexcitation des atomes se rendent jusqu'à un photomultiplicateur (PMT), où ces signaux lumineux seront convertis en signaux électroniques. Après amplification par un préamplificateur, ces signaux traversent un module de mise en forme (*shaper*) pour ensuite être numérisés (ADC) en vue de leur traitement numérique dans le DSP¹³. Enfin, les valeurs obtenues de ce traitement sont transmises au système d'acquisition de données (DAQ¹⁴).

¹³*Digital Signal Processing*

¹⁴*Data Acquisition System*

CHAPITRE 4

ÉLÉMENTS DU CIRCUIT ICS-DAQ

4.1 Cristal de ICs

4.1.1 Dépôt d'énergie

Les calorimètres électromagnétiques exploitent les propriétés des scintillateurs. Ces matériaux sont constitués d'atomes très lourds et donc qui ont des nuages électroniques imposants autour de leurs noyaux. Cette maximisation du nombre d'électrons dans un matériau augmente les probabilités d'interaction entre un photon ou une particule électriquement chargée qui le traverse et ces électrons [11][23].

Le passage d'une de ces particules au travers d'un scintillateur y génère une gerbe électromagnétique. Si c'est un photon, l'origine de la gerbe semble ponctuelle, puisque soit il se crée une paire électron-positron en traversant un champ coulombien, soit le photon est absorbé par un électron qui devient le point de départ de la gerbe [11][23]. Si c'est plutôt une particule massive, celle-ci parcourt une trajectoire dans le scintillateur, tout au long de laquelle elle perd de l'énergie, alimentant ainsi la gerbe produite [11][23]. Cette perte d'énergie est dominée par l'ionisation des atomes du milieu et est décrite par l'équation de Bethe-Bloch [13][21] :

$$-\frac{dE}{dx} = \frac{4\pi n z_e^2 Z_e}{m_e c^2 \beta^2} \left(\frac{e^2}{4\pi\epsilon_0} \right)^2 \left[\ln \left(\frac{2m_e c^2 \beta^2 \gamma^2}{I} \right) - \beta^2 - \frac{\delta(\gamma)}{2} \right] \quad (4.1)$$

où n est la densité d'atomes, z_e et Z_e sont respectivement les charges électriques de la particule incidente et des atomes du milieu, m_e est la masse d'un électron, e est la charge électrique élémentaire, ϵ_0 est la permittivité du vide, $\gamma = (1 - \beta^2)^{-\frac{1}{2}}$ est le facteur de Lorentz avec $\beta = \frac{v}{c}$ la fraction de la vitesse lumineuse, I est le potentiel d'ionisation du milieu et δ est une correction due à l'amortissement diélectrique par le milieu. Le comportement de l'équation (4.1) est illustré sur la figure 4.1 pour différents atomes. Cette équation justifie l'utilisation d'atomes lourds pour le scintillateur, afin d'avoir une

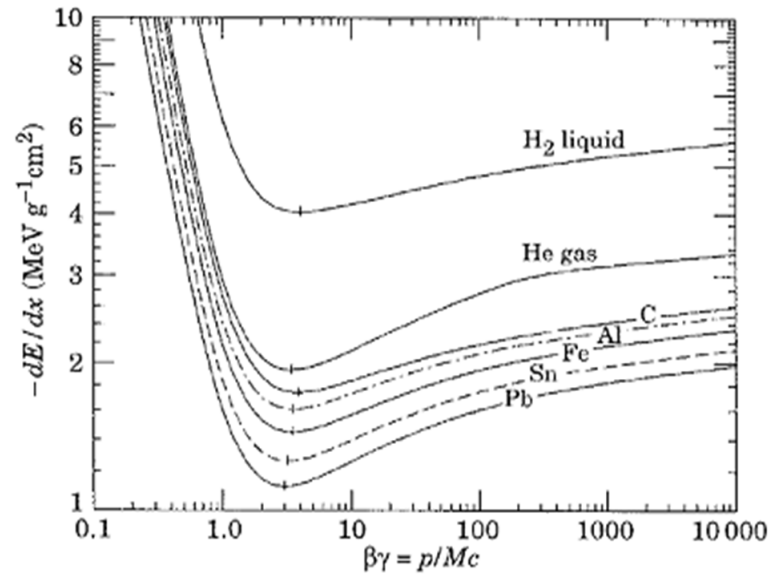


Figure 4.1 : Perte d'énergie par ionisation selon l'équation de Bethe-Block (4.1), pour différents atomes [16].

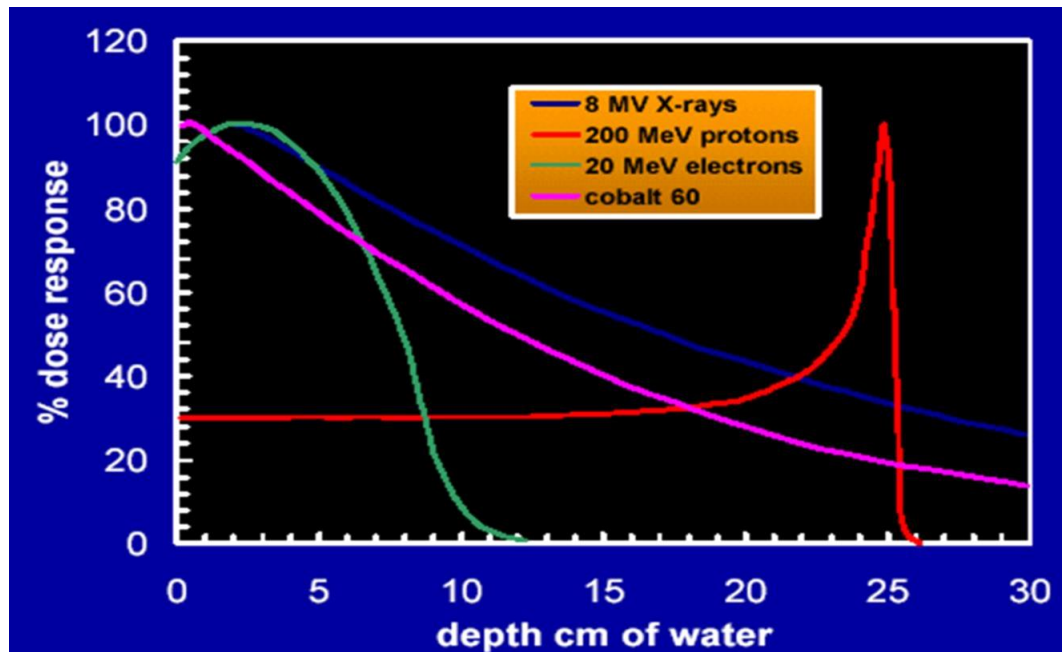


Figure 4.2 : Énergie perdue selon la distance parcourue dans un milieu (ici de l'eau), pour des rayons X, des protons, des électrons et des photons émis par du ^{60}Co [11]. Une réponse de 100% correspond à une perte d'énergie maximale.

charge Z_e aussi grande que possible. Quant à la particule incidente, plus elle a une charge électrique z_e importante, est légère ou est lente, plus elle perd rapidement de son énergie. Tandis qu'au-delà de la vitesse associée au minimum du potentiel d'ionisation (MIP), le taux de pertes suit une croissance lente monotone à mesure que $\beta \rightarrow 1$ et que les effets relativistes prennent de l'importance; l'espace étant contracté, le milieu semble plus dense. Aussi, la distance que peut parcourir une particule avant de s'arrêter est définie comme étant la longueur de pénétration, qui augmente avec la masse de la particule, ainsi qu'avec l'énergie [11][13][21][23].

Pour une particule qui termine sa course au repos dans le scintillateur, toute son énergie est déposée et ce suivant la courbe de Bragg [11], illustrée pour les protons à la figure 4.2. La courbe particulière de l'électron est due à sa faible masse qui le rend plus vulnérable aux diffusions multiples, ce qui diminue de beaucoup son pouvoir de pénétration [21]. Même si l'électron parcourt une grande distance, ce sera généralement à l'intérieur d'une région restreinte. Par contre, les muons ont une masse environ 200 fois plus grande que les électrons [23] et sont donc beaucoup moins affectés par le nuage électronique des atomes du scintillateur. C'est justement cette capacité des muons de traverser profondément la matière qui justifie qu'un détecteur, le KLM, lui soit dédié et que son emplacement soit le plus éloigné du point d'interaction [6].

4.1.2 Excitation du milieu

À mesure que les gerbes électromagnétiques générées dans le scintillateur évoluent, les collisions avec les atomes du milieu en excitent la structure moléculaire. Cette énergie est ensuite réémise sous forme de photons lors de sa désexcitation, dans le spectre visible. C'est là que les scintillateurs prouvent leur utilité, de par leur transparence à ces longueurs d'onde [21]. Ceci permet à la scintillation de se propager jusqu'à être captée par une photocathode située à l'interface du scintillateur pour ensuite être convertie en signal électronique, ce que nous aborderons plus en détails à la section 4.2.

L'étalement dans le temps de la scintillation émise dépend du temps de vie des états d'excitation occupés par les atomes, ainsi que de la trajectoire de la particule [11]. En effet, les photons doivent parcourir une certaine distance dans le scintillateur à partir

de l'endroit où la particule est passée. Plus la trajectoire de la particule incidente est longue ou se rapproche de la perpendiculaire par rapport à la photocathode, plus l'effet de marche devient important, c'est-à-dire l'effet de la différence de distance à parcourir entre les photons émis au début et à la fin de cette trajectoire. Cependant, ces dernières considérations sont de moindre importance pour les photons et les électrons, qui ne voyagent pas aussi loin dans le scintillateur. Le seul paramètre sur lequel on peut avoir un certain contrôle est le temps de vie des états d'excitation, que l'on prend aussi court que possible de par notre choix du matériau utilisé. Ainsi, les signaux lumineux produits durent moins longtemps, ce qui réduit les problèmes d'empilement [14].

4.1.3 Cristaux de ICs

Des cristaux d'iodure de césium dopés au thalium (ICs(Tl)) ont été utilisés pour l'expérience Belle [6][14]. Ce matériau émet principalement des photons dont la longueur d'onde tourne autour de 550 nm et possède deux composantes de scintillation, une rapide dont le temps de vie est de 600 ns et une lente avec un temps de 3500 ns [18]. Pour Belle II, il était initialement prévu d'utiliser un scintillateur différent afin d'améliorer la performance du calorimètre électromagnétique et les cristaux d'iodure de césium pur (ICs) font partie des options envisagées [6]. Ce matériau non dopé possède lui aussi une composante rapide et une lente, respectivement de 16 et 1000 ns. Cependant, ces deux composantes correspondent à deux états d'excitation dont les longueurs d'onde d'émission sont de 315 et 500 nm [18]. La figure 4.3 compare les spectres d'émission du ICs pur avec ceux du ICs(Tl) et du ICs(Na) dopé au sodium. On y voit clairement que le ICs a un pic d'émission principal et un secondaire. L'effet de ces deux composantes est que la montée des signaux est constituée d'une partie rapide suivie d'une partie lente, ce qui correspond à la désexcitation rapide de l'état associée à la plus petite longueur d'onde et à la désexcitation lente de l'état associé au pic secondaire. Par ailleurs, le ICs émet une scintillation environ 10 fois moins importante que le ICs(Tl) [6].

Dans le contexte d'une expérience à haute luminosité comme Belle II, les cristaux de ICs doivent faire preuve d'une grande résistance à la radiation. Des études ont montré que des cristaux de ICs pur frappés par des neutrons et des rayons γ peuvent recevoir

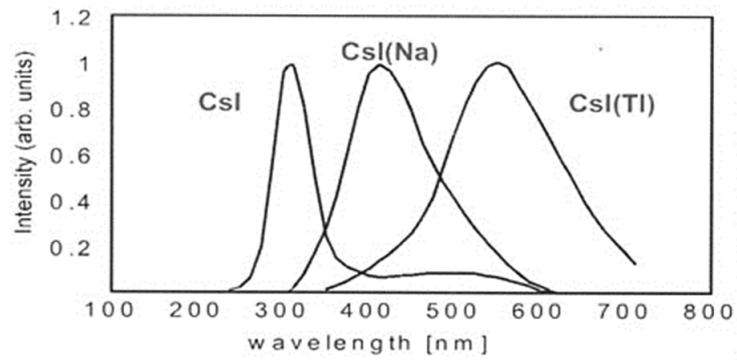


Figure 4.3 : Pics principaux d'émission du ICs, du ICs(Tl) et du ICs(Na) dans le spectre visible [12].

une radiation totale de 10 krad sans qu'il n'y ait de dégradations majeures et ce avec un flux de particules pouvant aller jusqu'à 10^{12} cm^{-2} [6].

4.1.4 Muons

Comme mentionné plus haut, les muons peuvent parcourir de grandes distances dans la matière, comparativement à d'autres particules à la même énergie. Ceci s'explique par la masse du muon $m_\mu = 105,66 \text{ MeV}/c^2$, grande comparée à celle d'un électron $m_e = 0,511 \text{ MeV}/c^2$ qui est facilement dévié de sa trajectoire, et petite par rapport à un proton $m_p = 938,272 \text{ MeV}/c^2$ qui se déplace plus lentement et interagit donc davantage avec le milieu, en plus de la possibilité d'interaction forte [23]. De plus, il y a toujours de la radiation dans l'environnement, qu'elle soit émise par les objets qui s'y trouvent ou qu'elle soit d'origine cosmique. C'est de cette deuxième source que proviennent majoritairement les muons qui traverseront le cristal. Il est possible de trier ces muons des autres particules incidentes en entourant le cristal de briques de plomb, bloquant ainsi la majorité de ces particules, qui n'ont pas une aussi grande capacité de pénétration. Les rayons cosmiques constituent donc une source fiable et bien connue de muons. D'ailleurs, le laboratoire RJAL¹ se situe environ 1 mètres sous le roc, ce qui élimine plusieurs particules chargées, ne laissant principalement que les neutrons et les muons.

¹René J. A. Lévesque

Ainsi, il est possible de déterminer le temps de vie et la masse du muon, par la méthode décrite à l'annexe IV.

4.1.4.1 Rayons cosmiques

Les rayons cosmiques primaires sont constitués de particules chargées à haute énergie, avec 90% de protons, 9% de particules alpha et des noyaux plus lourds pour le 1% restant. D'ailleurs, le record pour la particule la plus énergétique observée est de 10^{21} eV [1]. En passant dans l'atmosphère, ces rayons primaires génèrent des particules secondaires, incluant des hadrons (nucléons, anti-nucléons, pions, kaons...), mais on observe principalement des μ^\pm , des e^\pm , des neutrinos et des rayons gamma une fois au niveau de la mer [1]. La majorité des muons qui nous atteignent sont créés à une altitude de 15 km environ [1], principalement par la désintégration faible de pions chargés [9] :

$$\pi^+ \rightarrow \mu^+ + \nu_\mu \quad (4.2a)$$

$$\pi^- \rightarrow \mu^- + \bar{\nu}_\mu \quad (4.2b)$$

Ceci étant une désintégration à deux corps, l'énergie de chacune des particules produites prend une valeur bien définie dans le centre de masse. Pour le muon, on a :

$$E_\mu = \frac{m_\pi^2 + m_\mu^2}{2m_\pi} \quad (4.3)$$

où $m_\pi = 140 \text{ MeV}/c^2$ pour un π^\pm [23]. Il est démontré à l'annexe III comment on arrive à cette expression. L'énergie totale E_μ dans le centre de masse vaut alors 110,13 MeV, ce qui est seulement 4,47 MeV de plus que la masse au repos du muon. Les kaons peuvent aussi produire des muons selon le même processus [9], mais dans une plus faible mesure, ainsi que par deux mécanismes de désintégration en pions chargés [13][16] :

$$K^\pm \rightarrow \pi^\pm + \pi^0 \quad (4.4)$$

$$K^\pm \rightarrow \pi^\pm + \pi^+ + \pi^- \quad (4.5)$$

4.1.4.2 Distance moyenne dans le cristal

Les muons cosmiques qui traversent les ICs arrivent selon une distribution angulaire proportionnelle à $\cos^2\theta$, où θ est l'angle d'incidence par rapport à la verticale [9]. Puisqu'un muon parcourra dans le cristal une distance d'environ $p/\cos\theta$, où p est l'épaisseur du cristal, la distance moyenne sera donnée par

$$\begin{aligned} \langle x \rangle &= \frac{\int_0^\theta \frac{p}{\cos\theta} \cos^2\theta d\theta}{\int_0^\theta \cos^2\theta d\theta} \\ &= \frac{2p \sin\theta}{\theta + \sin\theta \cos\theta} \end{aligned} \quad (4.6)$$

Avec $\theta = 90^\circ$, soit l'horizontale, on obtient une valeur de $\langle x \rangle = 4p/\pi \approx 1,27p$ environ. Connaissant cette grandeur, on peut la multiplier par le taux de perte d'énergie par unité de distance pour un muon dans les ICs pour obtenir l'énergie qui y est déposée en moyenne :

$$\langle E \rangle = \left(\frac{dE}{dx}\right)_\mu \langle x \rangle = 1,27p \left(\frac{dE}{dx}\right)_\mu \quad (4.7)$$

4.1.4.3 Dépôt d'énergie par les muons

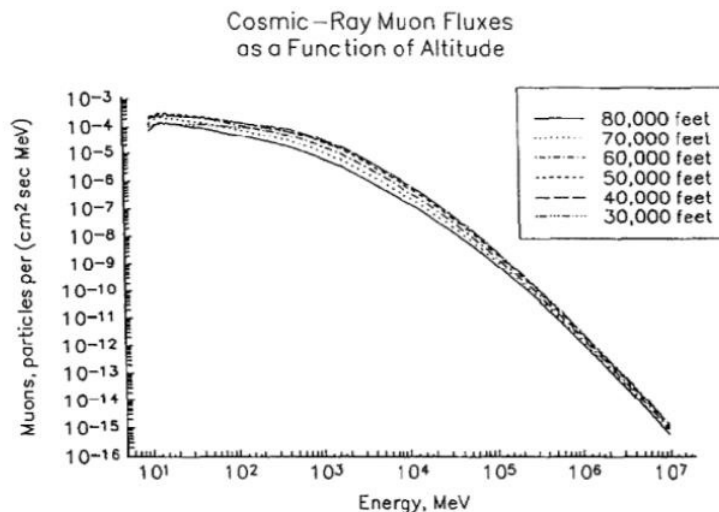


Figure 4.4 : Distribution en énergie des muons cosmiques, selon l'altitude [17].

La figure 4.4 montre que la distribution en énergie des muons cosmiques est relativement plate jusqu'à environ 1 GeV et suit au-delà la relation approximative

$$n(E)dE \propto E^{-\alpha}dE \quad (4.8)$$

où $n(E)$ est le flux de muons par unité d'énergie E dans un intervalle entre E et $E + dE$ et $\alpha \approx (7,9 \pm 9,1) \times 10^{-4}$ est un paramètre constant [17]. La majorité des muons qui nous parviennent ont une énergie d'environ 1 GeV [17]. Ceci correspond, au vu de la figure 4.5 qui représente le taux moyen de perte d'énergie $\frac{1}{\rho} \frac{dE}{dx}$ pour les muons (Bethe-Bloch), au régime des MIPs avec un minimum d'environ $1,5 \text{ MeV cm}^2/\text{g}$. Comme un cristal de ICs a une densité de $4,51 \text{ g/cm}^3$ [16], un muon y perd $6,77 \text{ MeV/cm}$ en moyenne. Comme les dimensions des cristaux sont de $30 \times 8 \times 6 \text{ cm}^3$, un muon incident à 0° qui traverse de part et d'autre l'épaisseur verticale de 6 cm doit déposer au moins $6,77 \cdot 6 = 40,62 \text{ MeV}$, puisqu'il peut arriver avec un angle par rapport à la verticale, tel que décrit par l'expression $\cos^2\theta$ [9]. À la section précédente, on a déterminé que la distance moyenne parcourue est à peu près 127% de l'épaisseur du milieu, soit 7,62 cm. Ainsi, pour le cristal, le dépôt moyen d'énergie est d'environ 51,59 MeV.

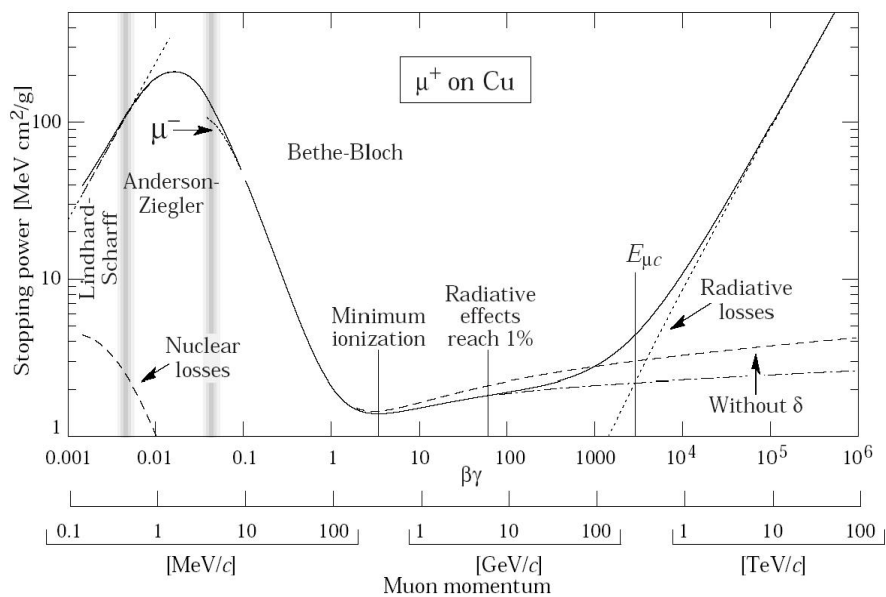


Figure 4.5 : Pouvoir d'arrêt du muon selon sa quantité de mouvement [16].

4.2 Photomultiplicateur

Un photon émis par le cristal de ICs se propage jusqu'à atteindre la cathode d'une photopentode de Hamamatsu, qui est un photomultiplicateur constitué de 3 dynodes entre la cathode et l'anode et qu'on soumet à une tension de 970 V en polarité négative [14]. Par effet photoélectrique, chaque photon qui atteint la photopentode arrache un électron de la cathode, qui est ensuite accéléré sous l'effet de la tension. Lorsque cet électron entre en collision avec la première dynode, il arrache plusieurs électrons qui se partagent son énergie. Le processus se répète avec ces électrons qui sont alors accélérés vers la deuxième, puis la troisième dynode. Enfin, les électrons arrachés de cette dernière dynode sont accélérés vers l'anode où ils sont collectés, générant une impulsion électronique [11]. On a ainsi converti une impulsion lumineuse en signal électronique.

La photopentode de Hamamatsu a un gain de 150 ± 2 électrons par photoélectron. Elle a été conçue pour conserver un gain satisfaisant lorsqu'elle est soumise à un champ magnétique important, considérant qu'elle devra supporter un champ de 1,5 T dans l'expérience Belle II. Dans de telles conditions, le gain tombe à $50 \pm 1 e^-$ /photoélectron, soit une diminution d'un facteur 3 [14].

4.3 Préamplificateur

Pour chacune des impulsions sortant de la photopentode, quelques centaines d'électrons seulement sont produits, ce qui n'est pas suffisant pour exploiter le signal, qui doit donc être amplifié. Le préamplificateur est un circuit électronique alimenté par deux sources de tension, une de 9 V et une de -3 V. Un fil conduit les impulsions sortant de la photopentode vers l'entrée du préamplificateur, qui a été conçu pour avoir une large gamme dynamique, avec saturation à 4 V et une large bande passante de 10 Hz à 30 MHz, avec un bruit faible de $118 \pm 7 \mu\text{V}$ [14]. Les impulsions sortantes sont guidées par une ligne de transmission jusqu'au module électronique suivant, le *shaper*, qu'on abordera à la section 4.4.

4.3.1 Amplification

Le principe de base pour l'amplification d'un signal électronique consiste à utiliser un circuit alimenté par une source de tension plus élevée que le voltage caractéristique des signaux. Le courant qui circule dans ce circuit peut être contrôlé ou modulé avec un transistor, par la tension appliquée sur sa grille². Un signal entrant fait varier la tension sur la grille, ainsi le courant dans le circuit est modifié de façon à en reproduire la forme à la sortie avec une plus grande amplitude. Le préamplificateur n'ayant qu'un fil de sortie, la fluctuation dans le temps du courant induit une variation de potentiel correspondante sur ce fil, ce qui génère une impulsion se propageant le long de celui-ci [2].

À l'entrée du préamplificateur, un transistor de type JFET est utilisé [14], dont l'ouverture du canal de conduction entre la source et le drain est contrôlé par la tension sur la grille [2]. Dans le reste du circuit on utilise plutôt des transistors à jonction bipolaire [14], contrôlés par le courant qui entre dans la base. Ces derniers amplifient les courants circulant dans le préamplificateur et leurs fluctuations [2].

4.3.2 Forme du signal sortant

Un condensateur de 2 pF à l'entrée du préamplificateur joue le rôle d'intégrateur [14], c'est-à-dire que les charges électriques portées par l'impulsion s'y accumulent [8], faisant augmenter la tension à ses bornes. C'est par cette tension que le contrôle de la grille du JFET est assuré. À mesure que les charges s'accumulent, l'amplitude du signal de sortie augmente.

Aussi, une résistance de 10 M Ω a pour but de décharger ce condensateur, afin d'éviter qu'il sature [14]. Une fois qu'il n'y a plus de charges qui s'accumulent, soit lorsqu'il n'y a plus de scintillation en provenance du cristal, la queue des signaux suit une décroissance exponentielle avec une constante de temps dont la valeur attendue est de $\tau_3 = RC = 20 \mu s$ [14]. Un exemple de signal sortant est présenté à la figure 4.6.

²Gate

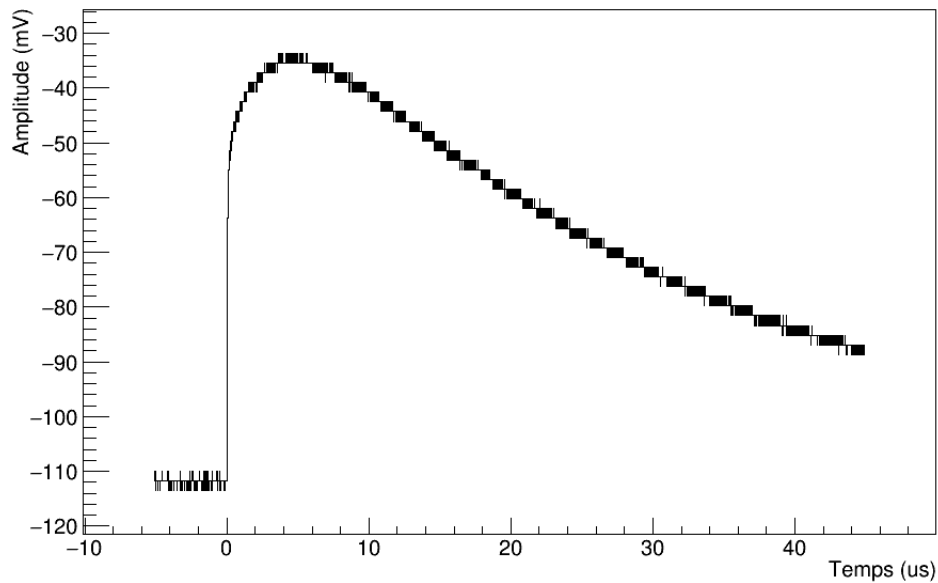


Figure 4.6 : Exemple de signal sortant du préamplificateur, avant élimination du piédestal.

4.3.3 Piédestal et seuil de détection

Les signaux produits par le cristal sont envoyés dans un oscilloscope LeCroy Wave-Pro, qui nous permet d'enregistrer l'amplitude selon le temps dans un fichier binaire de format *.trc*. L'oscilloscope s'enclenche et enregistre les signaux lorsqu'ils dépassaient le seuil d'enclenchement, d'environ 10 mV.

Un code C++ a été développé pour la lecture de ces fichiers, que l'on peut retrouver à l'annexe IX et qui est en fait une traduction d'un code Matlab [3]. Pour les valeurs de temps t_j , il y a trois valeurs à extraire : le nombre N de valeurs enregistrées, le temps Δt entre chacune de ces valeurs et un temps de décalage négatif t_0 par rapport à l'instant $t = 0$, qui correspond à l'instant où l'amplitude du signal dépasse la valeur d'enclenchement de l'oscilloscope, que l'on choisit aussi faible que possible. Si cette valeur est trop faible, l'oscilloscope enclenche sur du bruit. De plus, si l'amplitude d'un signal dépasse à peine ce seuil d'enclenchement, son sommet sera aligné sur l'instant $t = 0$, tandis qu'un signal de très haute amplitude dépassera rapidement le seuil et l'instant $t = 0$ sera près

du début du signal. L'extraction de la j -ème valeur de temps t_j se fait selon l'opération suivante :

$$t_j = t_0 + j\Delta t \quad (4.9)$$

avec $j = \{0, 1, 2, \dots, N - 1\}$. Le plus petit temps Δt possible avec l'oscilloscope est de 25 ps. Avec cette résolution maximale de 25 ps, un total de $N = 2000000$ de valeurs d'amplitudes y_j sont enregistrées sur un intervalle de temps de 50 μ s. Ainsi, on peut voir une bonne partie de la queue des signaux, dont on s'attend à ce que la décroissance suive une constante de temps de 20 μ s [14]. Ces valeurs d'amplitudes y_j en mV sont enregistrées dans le fichier sous forme d'entiers Y_j en représentation binaire. Ces entiers doivent être multipliés par un certain facteur K avant d'ajouter un piédestal P , qui sont deux valeurs à extraire du fichier, pour obtenir les amplitudes y_j :

$$y_j = Y_j K + P \quad (4.10)$$

La valeur du facteur K , qui est de 1,78 mV, est l'amplitude associée à une unité de Y_j . Le facteur K correspond donc aussi à la limite de précision de l'oscilloscope, puisqu'une unité en représentation binaire équivaut à 1,78 mV. Cette limite est plus apparente pour des signaux de faibles amplitudes et on peut en remarquer l'effet sur la figure 4.6, par la descente semblable à une fonction *step* qui décroît graduellement, par sauts de 1,78 mV.

Pour déterminer l'amplitude d'un signal, il faut d'abord tenir compte du piédestal et définir un seuil de détection. Le piédestal est donné par la valeur moyenne des amplitudes sur l'intervalle de temps qui précède l'instant $t = 0$, d'une durée d'environ 5 μ s. Ainsi, le piédestal à la sortie du préamplificateur est de $-111,5 \pm 0,3$ mV.

Les fluctuations autour du piédestal causées par le bruit suivent une distribution gaussienne. Ainsi, le seuil de détection devrait être défini comme étant au moins 3 fois l'incertitude du piédestal, ou 3σ , puisque 99,7% de la distribution se trouve à l'intérieur de cet intervalle, de part et d'autre de la valeur moyenne. Cependant, avec $\sigma = 0,3$ mV, le seuil de 0,9 mV est plus petit que la limite de précision K de l'oscilloscope. Ceci est dû au fait que les fluctuations ne s'étalent que sur très peu de valeurs discrètes d'amplitudes, sur un intervalle de $2K$ centré sur le piédestal. Ainsi, on définit plutôt le seuil

comme étant 6 mV, soit un peu plus que $3K$. L'amplitude d'un signal doit alors dépasser les 3 premières valeurs discrètes possibles au-dessus du piédestal³.

4.3.4 Ajustements

La scintillation produite dans le ICs dépend de deux états d'excitation aux temps de vie différents, $\tau_1 = 16$ ns et $\tau_2 = 1000$ ns [18]. En considérant que le signal sortant du préamplificateur correspond à l'intégration de cette scintillation combinée à une décroissance exponentielle suivant une constante de temps $\tau_3 = 20$ μ s [14], on peut s'attendre à ce que l'amplitude $f(t)$ à la sortie selon le temps t écoulé soit donnée par l'expression suivante :

$$f(t) = e^{-\frac{t}{\tau_3}} \int_0^t (ae^{-\frac{t}{\tau_1}} + be^{-\frac{t}{\tau_2}}) dt \quad (4.11a)$$

$$= e^{-\frac{t}{\tau_3}} [a\tau_1(1 - e^{-\frac{t}{\tau_1}}) + b\tau_2(1 - e^{-\frac{t}{\tau_2}})] \quad (4.11b)$$

où les paramètres a et b sont proportionnels à l'énergie déposée par la particule qui traverse le cristal et reflètent les populations relatives de chaque état d'excitation. L'intégrale converge vers $a\tau_1 + b\tau_2$, quand tous les atomes se sont désexcités et qu'ils n'émettent plus de scintillation. De plus, $f(t) = 0$ au début du signal quand $t = 0$. L'ajustement de l'équation (4.11) sur un signal sortant produit par le passage d'une particule de radiation ambiante est présenté à la figure 4.7 et l'accord est si bon que l'ajustement est à peine visible. La méthode utilisée pour les ajustements non-linéaires est présentée en détails à l'annexe II.

Sur la distribution d'amplitudes obtenues par l'ajustement de l'équation (4.11), montrée à la figure 4.8, on observe deux pics d'amplitude pour la radiation ambiante, à environ 20 et 60 mV. Ces pics peuvent s'expliquer par les deux sources de radiation naturellement disponibles, soient les muons cosmiques et l'environnement immédiat. Les figures 4.9 à 4.13 montrent les distributions pour les paramètres a , b , τ_1 , τ_2 et τ_3 , obtenues pour

³On a la même situation pour des signaux sortant d'un photomultiplicateur utilisé avec un scintillateur de plastique, avec un piédestal de $1,0 \pm 0,2$ mV qui mène à un seuil de 0,6 mV. On utilise donc la même convention de $3K \approx 6$ mV que pour la sortie du préamplificateur. Le scintillateur de plastique a servi à établir des coïncidences ; voir l'annexe IV.

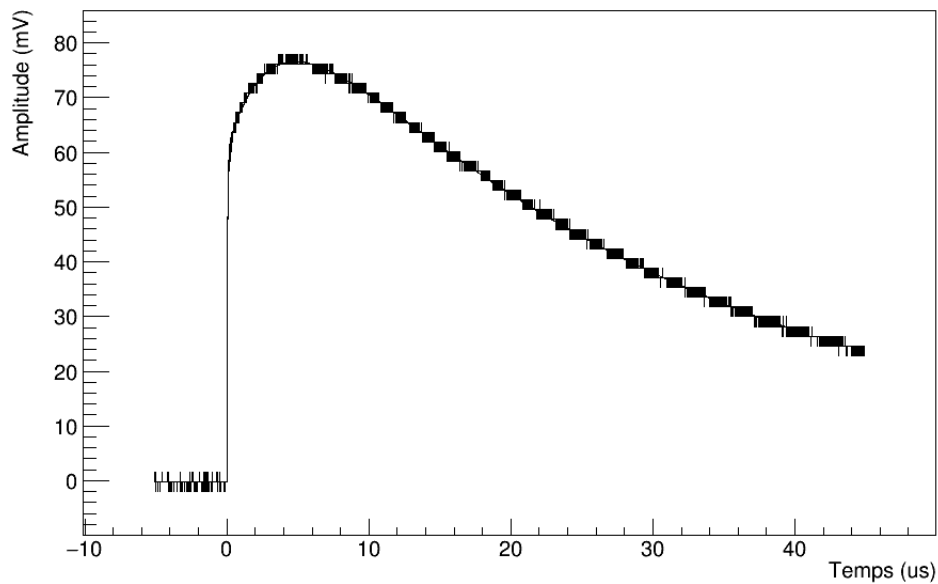


Figure 4.7 : Ajustement de l'équation (4.11) sur un signal sortant du préamplificateur.

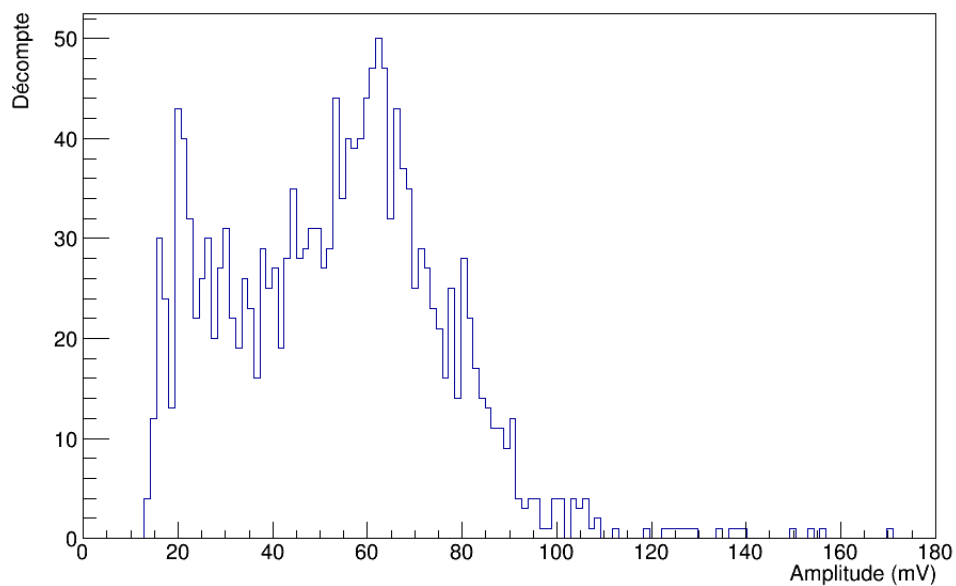


Figure 4.8 : Distribution d'amplitude pour des particules de la radiation ambiante, obtenue par l'ajustement de l'équation (4.11) sur les signaux sortant du préamplificateur.

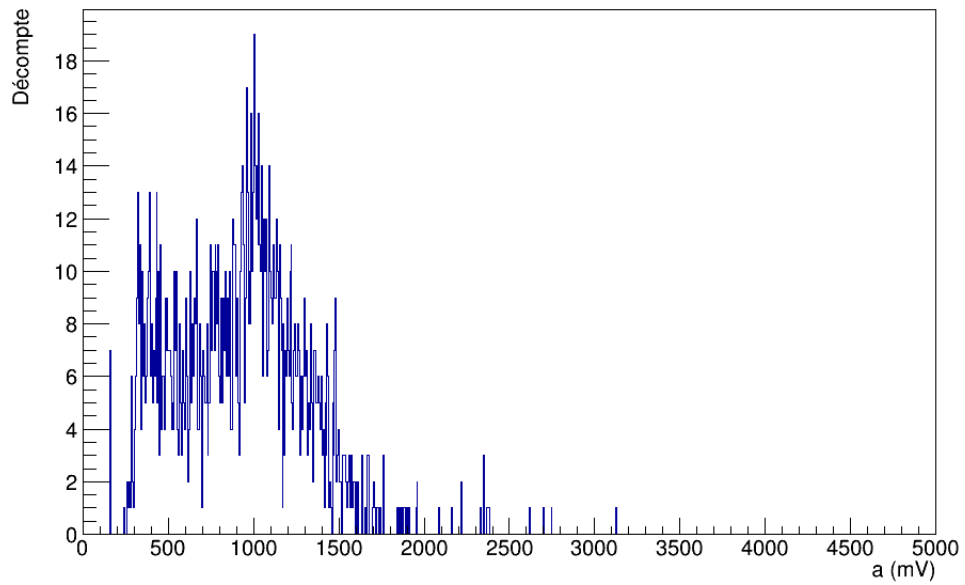


Figure 4.9 : Distribution du paramètre a de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.

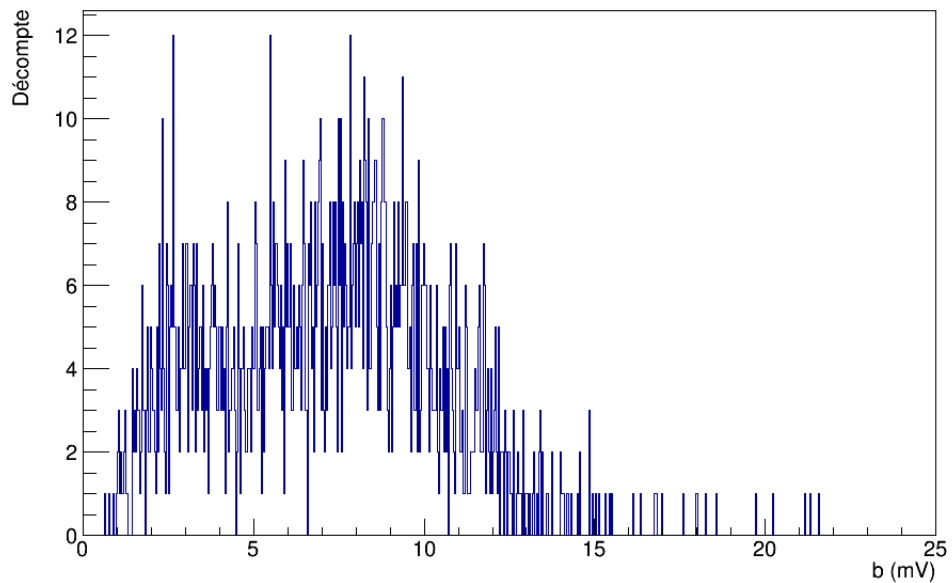


Figure 4.10 : Distribution du paramètre b de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.

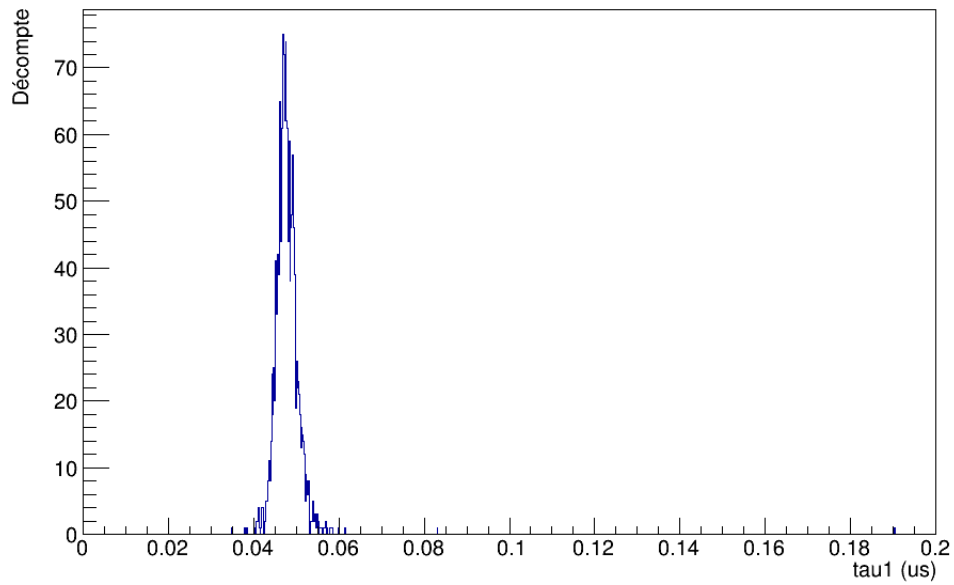


Figure 4.11 : Distribution du paramètre τ_1 de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.

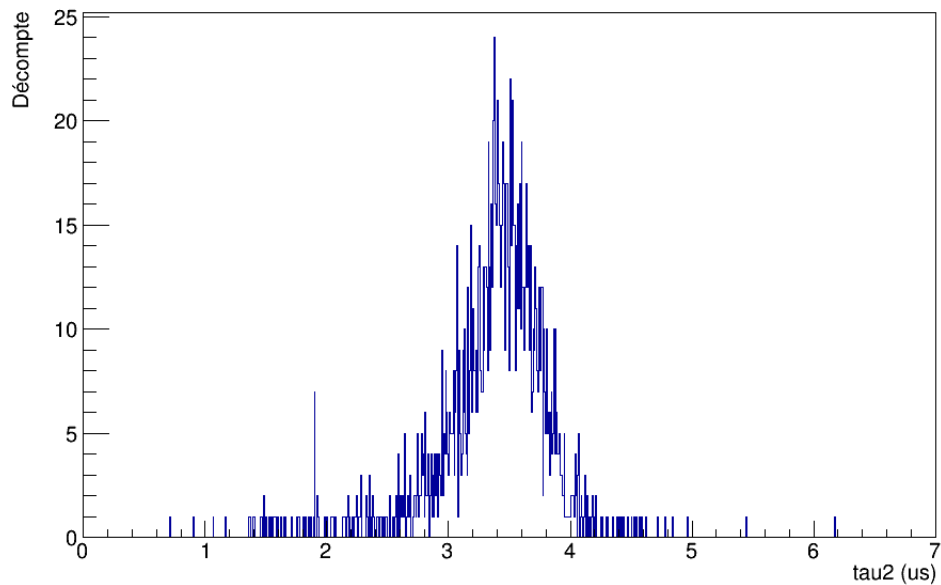


Figure 4.12 : Distribution du paramètre τ_2 de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.

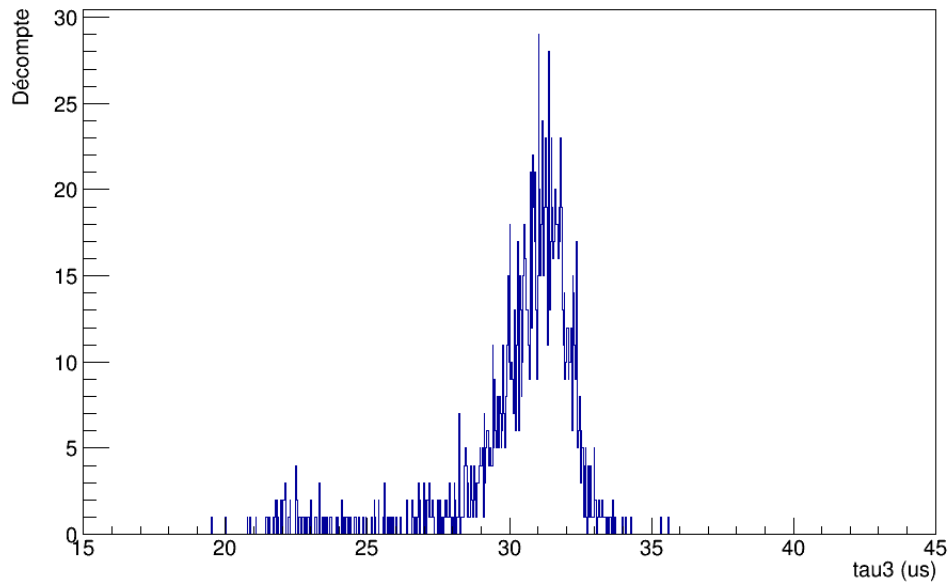


Figure 4.13 : Distribution du paramètre τ_3 de l'équation (4.11) après ajustement sur les signaux sortant du préamplificateur.

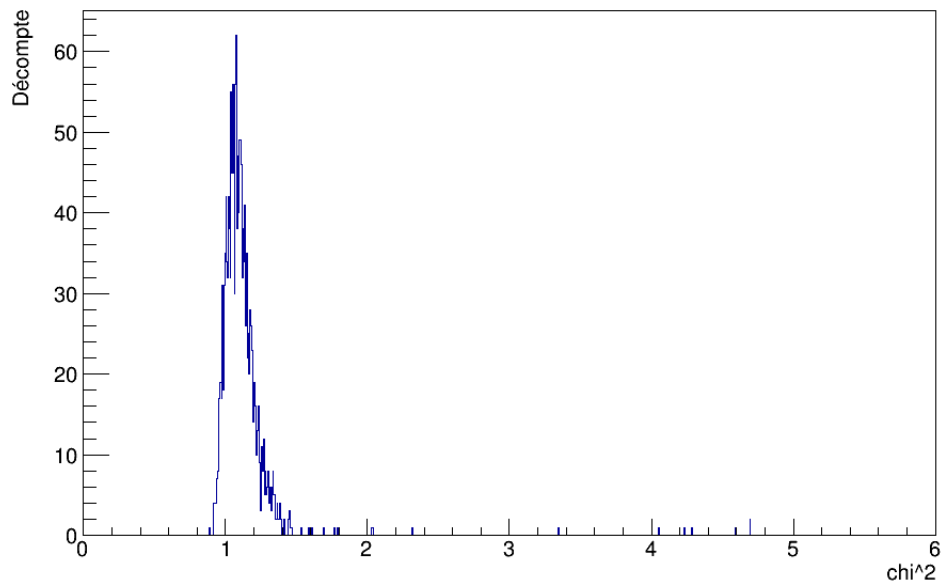


Figure 4.14 : Distribution des χ^2 par degré de liberté, ou $\widetilde{\chi^2}$, pour l'ajustement de l'équation (4.11) sur les signaux sortant du préamplificateur.

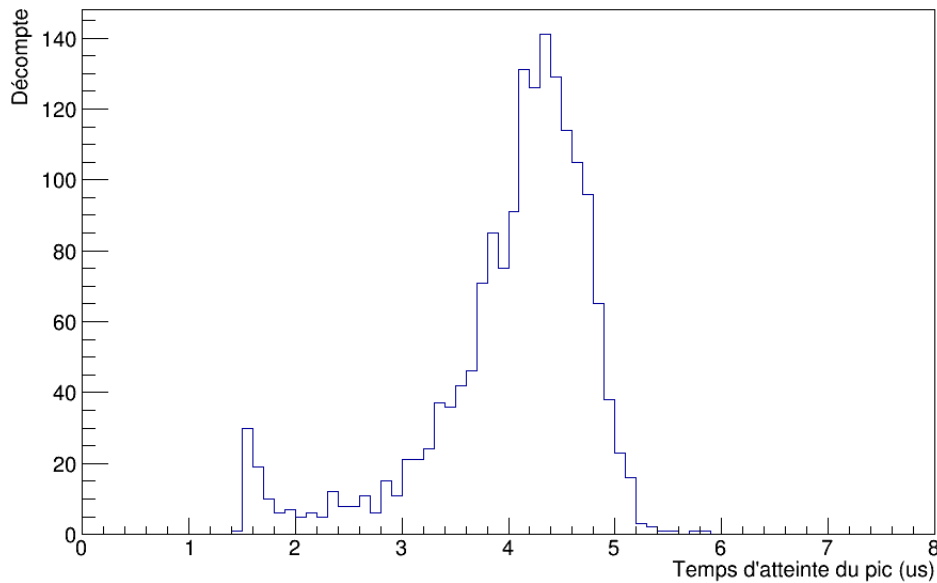


Figure 4.15 : Distribution du temps d’atteinte du pic pour les signaux sortant du préamplificateur.

des signaux générés par de la radiation ambiante. Les paramètres a et b prennent respectivement les valeurs moyennes 900 ± 400 mV et 7 ± 3 mV. Leurs distributions présentent deux pics, ce qui reflète la distribution d’amplitudes de la figure 4.8. Quant aux autres paramètres, on obtient respectivement 47 ± 4 ns et $3,4 \pm 0,5$ μ s pour les temps de vie τ_1 et τ_2 des états d’excitation du ICs et 30 ± 2 μ s pour la constante de temps τ_3 de la décroissance exponentielle, alors qu’on s’attendait à 16 ns [18], 1 μ s [18] et 20 μ s [14]. Les temps obtenus sont plus longs, surtout pour la scintillation en raison de réflexions à la surface du cristal, de la géométrie allongée de ce dernier et du temps d’intégration du condensateur à l’entrée du préamplificateur. Ces effets ajoutent 31 ns à τ_1 et 2,4 μ s à τ_2 et ces paramètres valent environ le triple de leurs valeurs attendues. Pour τ_3 , il faut aussi savoir que la capacitance d’un condensateur a une incertitude relative de 25% et celle d’une résistance est de 1%, ce qui implique que la constante de temps réelle pour la décharge du condensateur pourrait en réalité être aussi grande que 25,25 μ s en raison de cette incertitude. De plus, pour un signal, les dernières contributions de scintillation arrivent durant la descente, ce qui ralentit cette dernière et donc allonge le temps estimé

pour τ_3 . C'est ainsi que sa valeur optimale peut être $10 \mu\text{s}$ plus grande que la valeur attendue. La distribution des χ^2 par degré de liberté, ou $\widetilde{\chi^2}$, pour l'ajustement de chacun des signaux est présentée à la figure 4.14, avec une valeur moyenne de $1,1 \pm 0,2$, ce qui implique un très bon accord entre les ajustements et les signaux mesurés, ainsi qu'une stabilité dans les valeurs optimales pour les paramètres constants τ_1 , τ_2 et τ_3 , malgré le désaccord avec les valeurs attendues.

On observe des fluctuations pour τ_2 et τ_3 vers des temps plus courts. Il faut savoir que deux formes de signaux générés dans le ICs ont été observées à la sortie du préamplificateur et qui diffèrent par le temps d'atteinte du pic⁴, comme l'atteste la figure 4.15. En effet, dans la majorité des cas, la montée est due à une composante de scintillation rapide et une lente, mais dans certains cas, cette dernière, représentée par le terme avec b et τ_2 dans l'équation (4.11), est absente. Ainsi, les fluctuations peuvent être reliées à ces quelques occurrences. De plus, l'étalement du temps d'atteinte du pic autour de sa valeur moyenne peut dépendre de l'angle d'incidence de la particule dans le cristal, puisque la scintillation émise dans le ICs doit parcourir une certaine distance avant d'atteindre la photopentode, distance qui dépend du point de départ le long de la trajectoire de la particule.

La figure 4.16 semble montrer une dépendance de l'amplitude maximale de l'ajustement sur le temps d'atteinte du pic, surtout en bas de 60 mV environ. Plus haut, ce temps semble se stabiliser autour de $4,6 \mu\text{s}$, quoique l'ensemble statistique pour des amplitudes élevées n'est pas suffisant pour confirmer. On peut aussi voir le pic associé à des temps très courts d'environ $1,6 \mu\text{s}$ et qui correspondent à une absence de la composante lente. Le ratio $\frac{b}{a}$, qui exprime les populations relatives de chaque état d'excitation dans la scintillation, prend une valeur moyenne de $0,008 \pm 0,003$. L'état qui vit le moins longtemps domine, ce qui est une bonne qualité pour une situation à haut taux de comptage. La figure 4.17 montre que ce ratio prend généralement la même valeur, sans dépendance sur l'amplitude. Quant à la figure 4.18, elle semble montrer une légère dépendance du ratio sur le temps d'atteinte du pic, qui augmente pour une plus grande population relative dans l'état d'excitation au temps de vie le plus long. Cependant, en l'absence de

⁴Peaking time

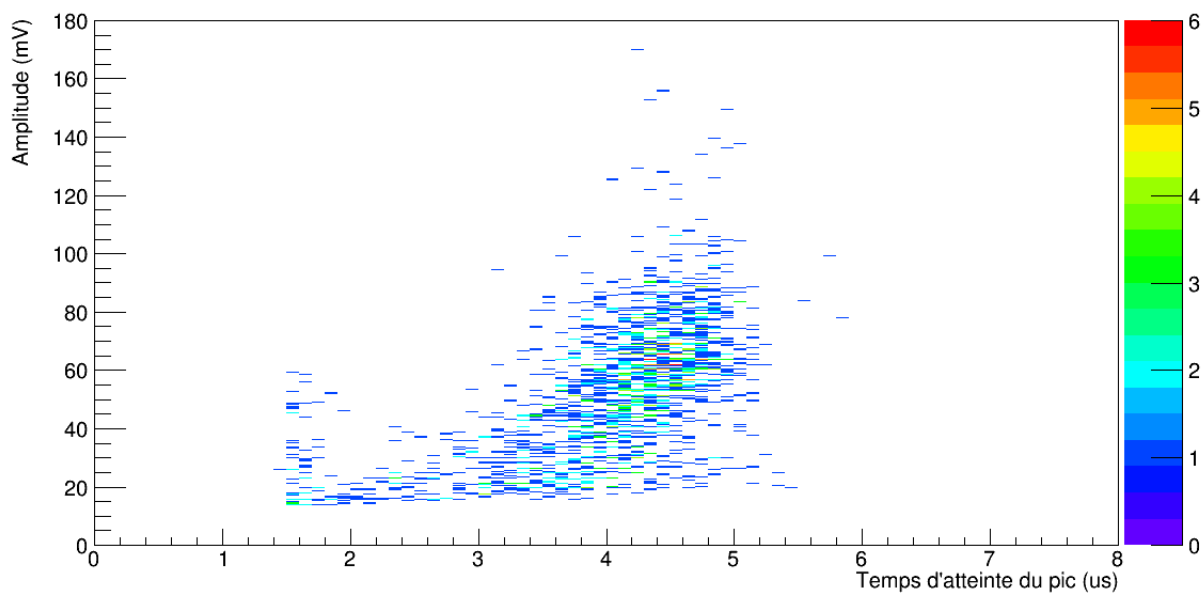


Figure 4.16 : Distribution de l'amplitude selon le temps d'atteinte du pic pour des signaux sortant du préamplificateur.

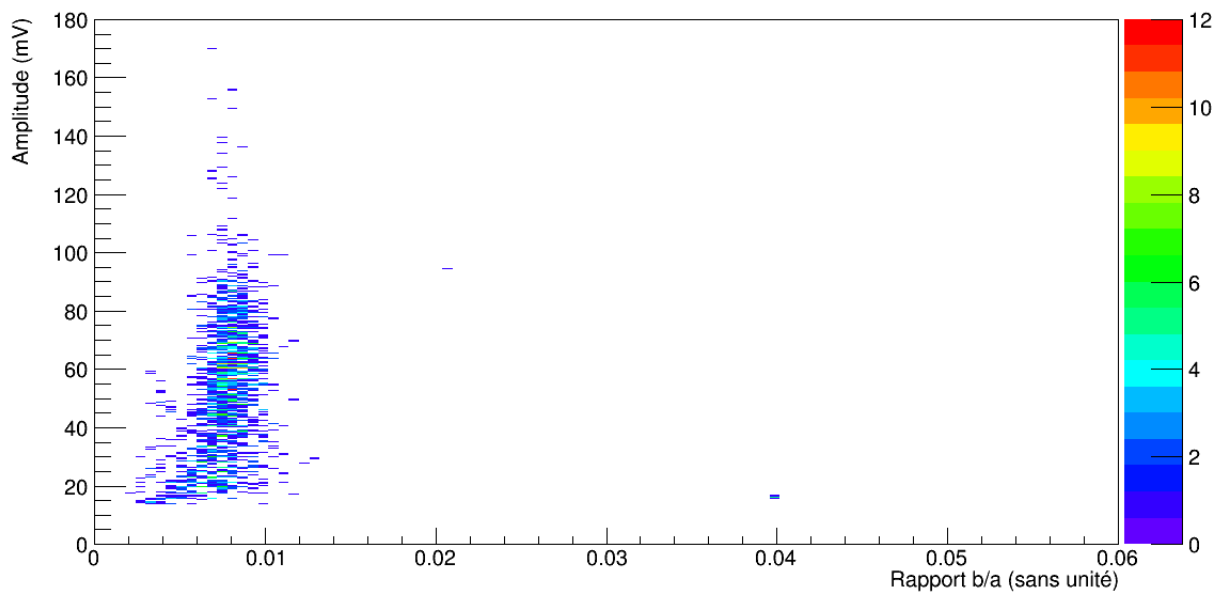


Figure 4.17 : Distribution de l'amplitude selon le ratio b/a pour des signaux sortant du préamplificateur.

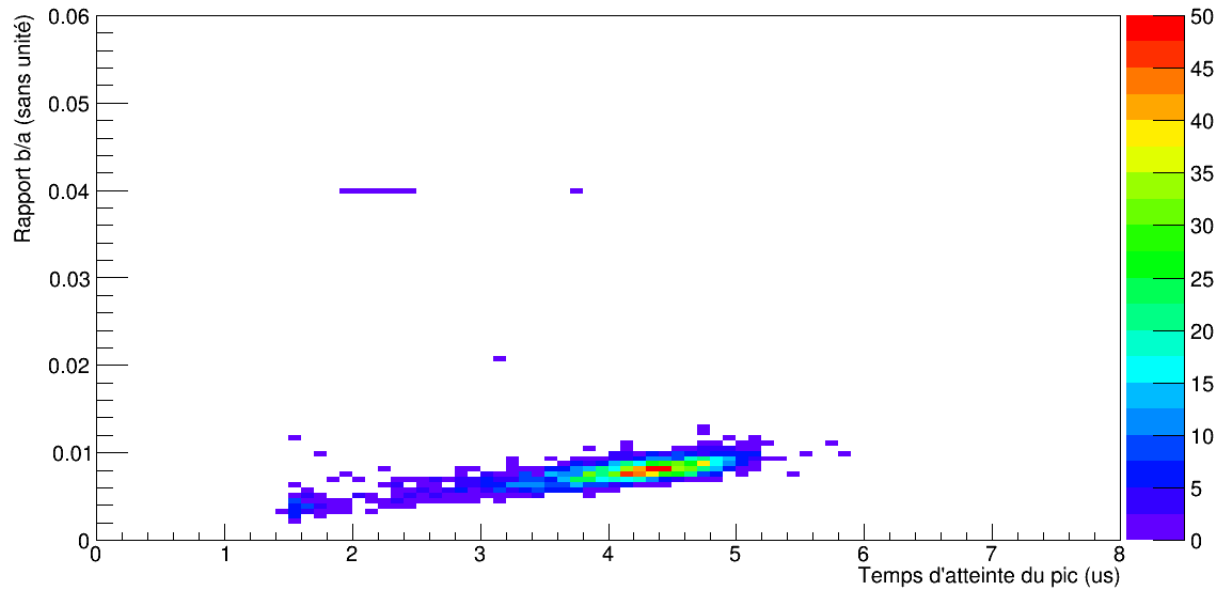


Figure 4.18 : Distribution du temps d'atteinte du pic selon le ratio b/a pour des signaux sortant du préamplificateur.

dépendance claire entre l'amplitude, le ratio $\frac{b}{a}$ et le temps d'atteinte du pic, on est forcé de conclure que la présence ou non de la composante lente dans la montée des signaux est due à une cause non identifiée.

4.3.5 Conversion énergie-amplitude

Pour convertir l'amplitude d'un signal mesuré en énergie déposée par une particule, on utilise deux sources de radiation connues, par exemple le ^{137}Cs qui émet des photons à 662 keV et le ^{22}Na à 1,275 MeV. Pour chacune de ces sources, on collecte des signaux dont les amplitudes sont d'environ 3 et 5,6 mV, respectivement. En faisant passer une droite par ces deux valeurs, on obtient une règle de conversion entre l'amplitude V d'un signal sortant du préamplificateur et l'énergie E déposée par une particule :

$$E = 0,236V - 0,045 \quad (4.12)$$

où les unités correspondent à une conversion de mV vers MeV. Avec un générateur d'impulsions, des ondes carrées, dont on peut contrôler l'amplitude, sont envoyées dans le préamplificateur, afin de vérifier la linéarité entre la tension qui y entre et celle qui en sort.

4.4 Shaper

Le signal sortant du préamplificateur se dirige ensuite vers le *shaper*, où il subit une mise en forme. Ceci permettra éventuellement dans le DSP de comparer les signaux à une même forme de référence. Nous verrons dans cette section comment se fait cette mise en forme. Étant donnée la présence de transistors, le *shaper* est un filtre actif, puisque les filtres passifs ne contiennent rien d'autre que des résistances R , des condensateurs C et des solénoïdes L [8].

Étant donné qu'il y a un décalage en courant continu⁵ à la sortie du préamplificateur, on ajoute un filtre passe-haut avec un condensateur de 3000 pF entre le préamplificateur et le *shaper* pour éviter le régime de saturation.

4.4.1 Filtres passe-haut et passe-bas

Un filtre passe-haut CR passif [8] est constitué d'un condensateur suivi d'une résistance, tel que montré à gauche sur la figure 4.19. La tension de sortie $V_{out} = RI$ dépend du courant qui traverse la résistance. Le courant I qui traverse le condensateur est donné par l'équation :

$$I = C \frac{dV}{dt} \quad (4.13)$$

où V est la tension aux bornes du condensateur C . Ainsi, la variation rapide de l'amplitude d'un signal induit un courant plus élevé qui traverse le filtre, tandis qu'une variation trop lente produit un courant trop faible à la sortie. Un filtre passe-haut ne laisse donc passer que les composantes à hautes fréquences d'un signal. Dans le cas limite d'une fréquence nulle, ou un courant continu, aucun courant ne sort du filtre une fois que le condensateur est complètement chargé.

⁵DC offset.

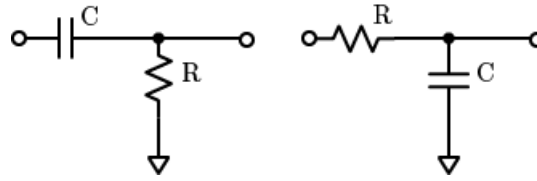


Figure 4.19 : Filtre passe-haut à gauche et filtre passe-bas à droite. Les signaux entrent par la gauche et sortent à droite.

Sur un filtre passe-bas RC passif [8], à droite sur la figure 4.19, la résistance vient avant le condensateur. La tension de sortie étant la même qu'aux bornes du condensateur, celui-ci doit se charger pour qu'il y ait un courant de sortie. Cependant, la charge du condensateur ne se fait pas instantanément et suit une constante de temps RC . Donc si la fréquence d'un signal est trop grande, le condensateur n'a pas le temps de se charger et la tension de sortie est nulle. De plus, la décharge du condensateur définit la longueur de la queue du signal en suivant la même constante de temps RC .

La fréquence de coupure f_c , liée au temps de réponse d'un filtre, satisfait la même expression pour les filtres passe-haut et passe-bas [8] :

$$f_c = \frac{1}{2\pi RC} \quad (4.14)$$

Cette fréquence ne correspond pas à une coupure abrupte de la tension de sortie des filtres, mais plutôt à une atténuation de 3 dB. Ensuite, cette tension est atténuée de 20 dB chaque fois que la fréquence varie d'un facteur 10, dépendamment du type de filtre utilisé. C'est aussi pourquoi le condensateur du filtre passe-haut entre le préamplificateur et le *shaper* a une capacité élevée de 3000 pF, afin d'avoir une fréquence de coupure aussi petite que possible, pour ne couper que le décalage en courant continu.

4.4.2 Filtres CR-(RC)⁴

La mise en forme du signal se fait principalement à l'aide d'un filtre CR-(RC)⁴, composé d'un filtre passe-haut CR en entrée, suivi de quatre étages de filtres passe-bas RC disposés en deux structures de type Sallen-Key de second ordre. Cette structure est

montrée à la figure 4.20 et consiste en deux filtres passe-bas l'un après l'autre, dont le condensateur du premier filtre est connecté à la sortie plutôt que sur la mise à la terre. Un amplificateur assure que la structure a un gain de 1 et qu'il n'y a donc pas d'atténuation.

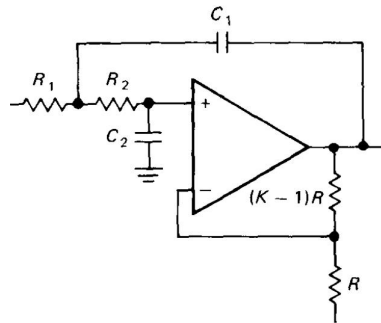


Figure 4.20 : Filtre passe-bas de Sallen-Key de second ordre, où un signal entre à gauche et sort à droite [8].

Si le gain d'un filtre passif est toujours plus petit que 1, on s'attend à ce que le *shaper*, actif, ait un gain de 20. Le filtre CR actif à l'entrée du *shaper* amplifie le signal et permet d'éliminer les composantes lentes des signaux, dues au temps de vie plus long d'un des deux états d'excitation possibles des atomes de ICs, soit 1000 ns comparé à 16 ns environ. Ainsi, cette composante ne constituera pas un problème majeur pour l'ajustement par rapport à une forme de référence. Chacun des quatre étages suivants de filtres RC contribuent à la mise en forme elle-même du signal, avec les condensateurs qui jouent le rôle d'intégrateurs de la charge entrante, ce qui définit la tension de sortie. La figure 4.21 montre un exemple de signal, produit par un générateur d'impulsions, sortant du *shaper*, après numérisation par un ADC. On remarque un piédestal important par rapport à l'amplitude, ce qui réduit la largeur de la gamme dynamique.

4.4.3 Temps de montée

Le temps de montée d'un signal électronique est habituellement défini comme le temps nécessaire pour que l'amplitude passe de 10 à 90% de l'amplitude maximale du signal. Pour ce faire, on prend le premier point dont l'amplitude dépasse 10% de ce

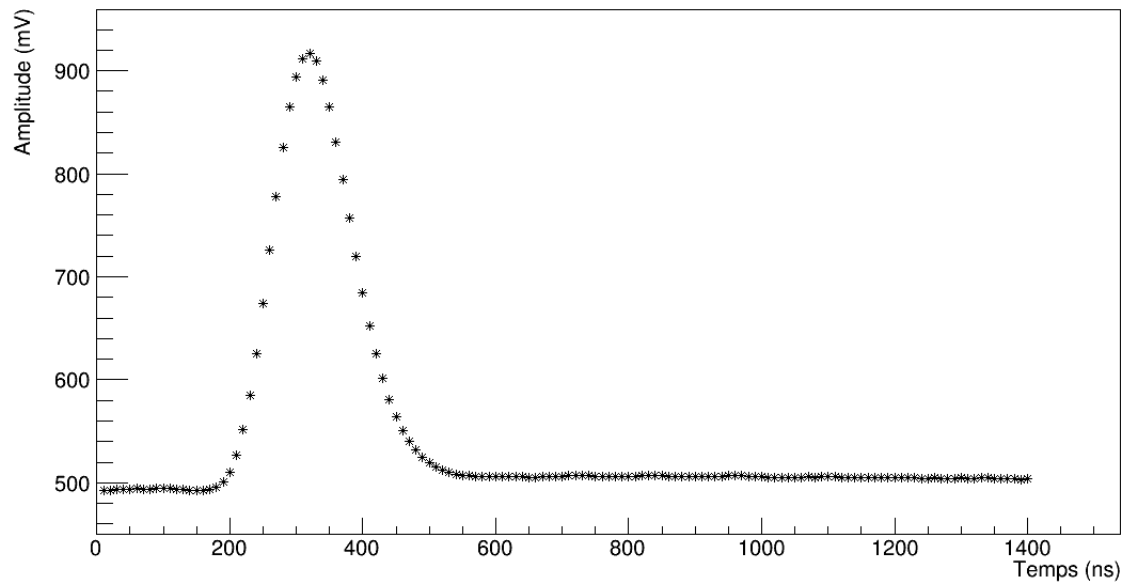


Figure 4.21 : Exemple de signal, produit par un générateur d’impulsions, sortant du *shaper*, après numérisation par un ADC et enregistré par le DAQ.

maximum, ainsi que le point précédent. On détermine la droite qui les traverse pour extrapoler l’instant où l’amplitude valait exactement 10% du maximum. En appliquant la même méthode pour 90% de l’amplitude maximale, on obtient le temps de montée en prenant la différence entre ces deux instants. Le temps de montée moyen obtenu avec un générateur d’impulsions est de $78,7 \pm 0,3$ ns, tandis qu’avec des signaux en provenance du ICs, on a plutôt 71 ± 2 ns. Leurs distributions sont illustrées sur les figures 4.22 et 4.23. La différence peut s’expliquer par le fait que même si les signaux produits par le générateur et dans le ICs ont des formes très semblables, ils n’ont pas nécessairement le même étalement dans le temps. De plus, pour le ICs, l’angle d’incidence de la trajectoire de la particule peut affecter le temps de collecte de la scintillation, donc le temps de montée, d’où une plus grande incertitude.

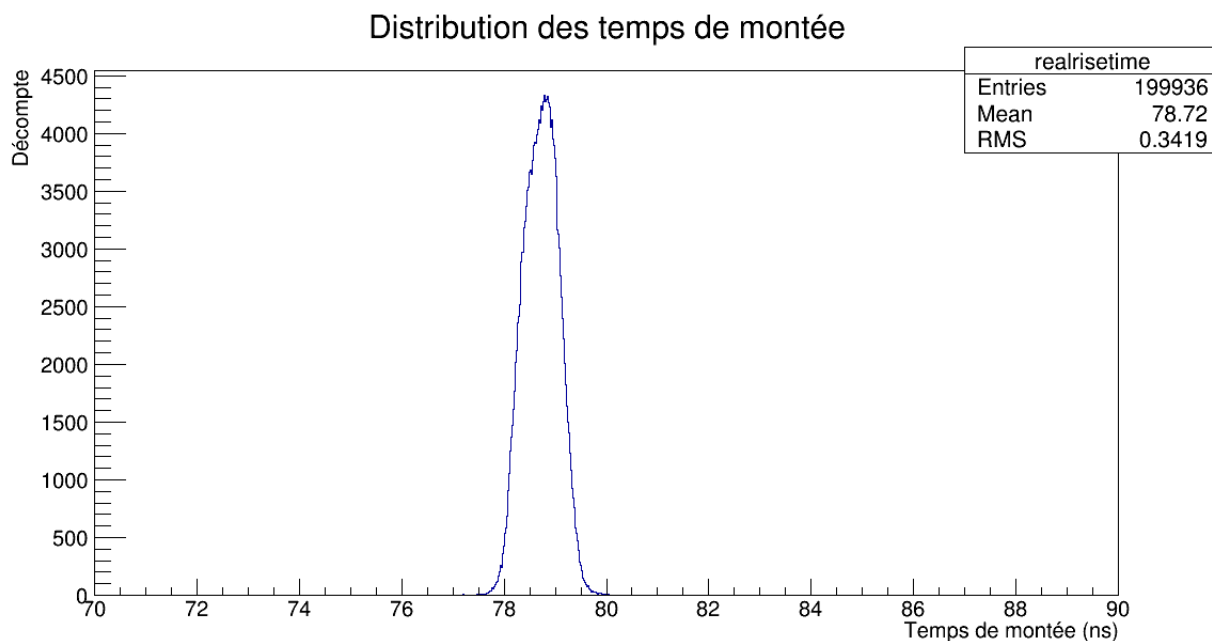


Figure 4.22 : Distribution des temps de montée pour des signaux produits par un générateur d'impulsions, à la sortie du *shaper*, après numérisation par un ADC.

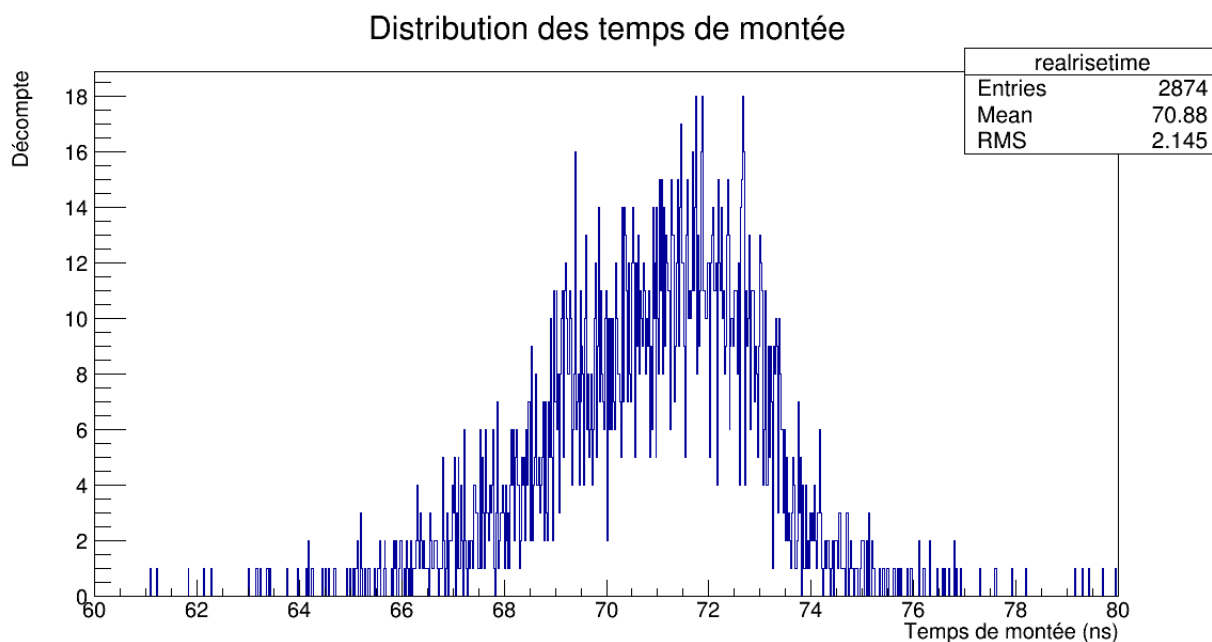


Figure 4.23 : Distribution des temps de montée pour des signaux en provenance des ICs, à la sortie du *shaper*, après numérisation par un ADC.

4.4.4 Gamme dynamique

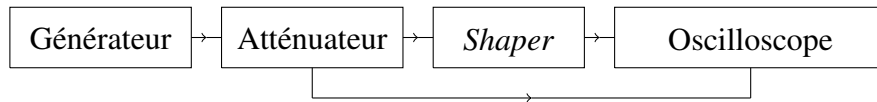


Figure 4.24 : Montage pour déterminer la conversion de l'amplitude d'un signal qui entre dans le *shaper* vers celle d'un signal qui en sort.

Pour faire la conversion de l'amplitude d'un signal sortant du *shaper* en énergie déposée par une particule, on peut difficilement utiliser les muons cosmiques ou une source de radiation comme le ^{60}Co dont on connaît l'énergie des photons émis, puisque la gamme dynamique est trop étroite pour observer une linéarité entre les amplitudes entrante et sortante. On peut cependant utiliser un générateur d'impulsions pour envoyer des ondes carrées d'amplitude bien définie, ce qui permet de savoir combien de mV sortent pour chaque mV qui entre. Le montage est illustré à la figure 4.24. Étant donné que la limite inférieure du générateur pour l'amplitude du signal est d'environ 11,8 mV, soit dans le régime de saturation du *shaper*, on ajoute un atténuateur de 40 dB, donc d'un facteur 100. Un atténuateur est essentiellement un diviseur de tension, soit deux résistances avec le fil de sortie connecté entre elles. Le schéma d'un diviseur de tension est illustré sur la figure 4.25.

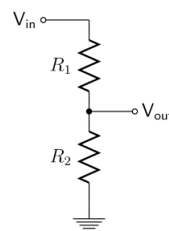


Figure 4.25 : Schéma d'un diviseur de tension.

La figure 4.26 montre la relation entre l'amplitude x à l'entrée du *shaper* et celle à la sortie y , ainsi qu'une droite de régression linéaire qui satisfait la règle :

$$y = 702,7x - 6,7 \quad (4.15)$$

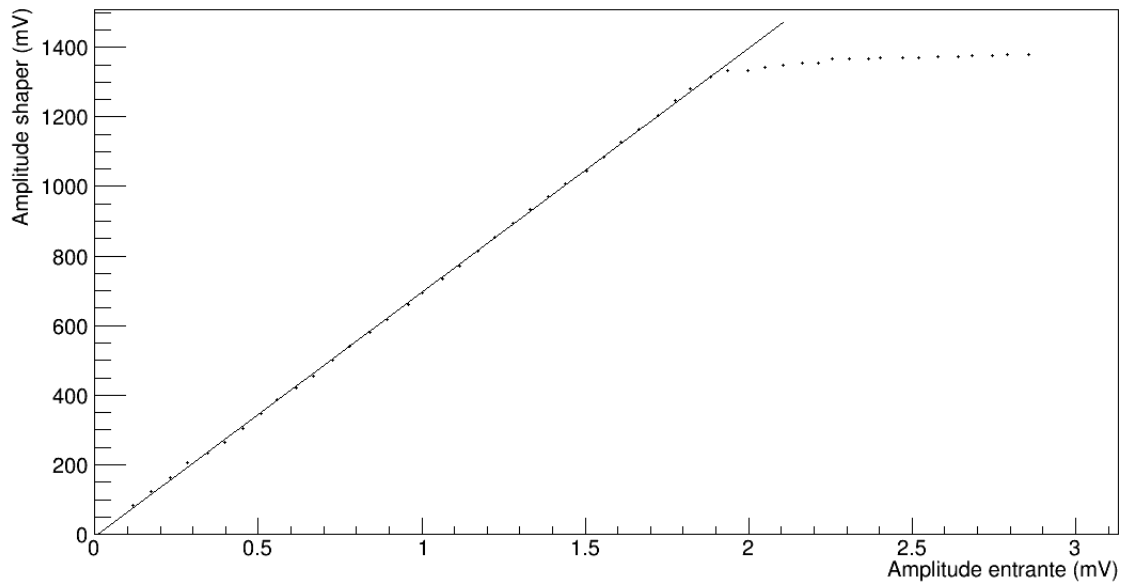


Figure 4.26 : Amplitude qui sort du *shaper* selon l'amplitude qui y entre.

qui devient notre taux de conversion en mV sortant, pour chaque mV entrant. On voit clairement que la gamme dynamique est très étroite, puisque le *shaper* commence à saturer à partir d'une amplitude très faible d'environ 1,9 mV à l'entrée, avec une tension maximale d'environ 1350 mV à la sortie. Malgré tout, il est possible de faire l'étude de la forme des signaux sortant.

À la section 4.3.5, on a déterminé la règle de conversion entre l'énergie E en MeV déposée dans le cristal et l'amplitude V en mV à la sortie du préamplificateur, donnée par l'équation (4.12). L'amplitude V correspond à l'amplitude x qui entre dans le *shaper*. Cependant, elle peut avoir été diminuée d'un facteur k par un atténuateur placé après la sortie du préamplificateur, auquel cas l'amplitude qui entre dans le *shaper* équivaut à $x = V/k$. Alors la conversion entre l'amplitude en mV d'un signal sortant du *shaper* et l'énergie en MeV déposée par une particule dans le cristal est donnée par :

$$E = k(3,36y + 22,50) \times 10^{-4} - 0,045 \quad (4.16)$$

avec un facteur k d'une valeur de 100 dans notre montage.

4.5 ADC

Avant d'entrer dans le DSP⁶, le signal sortant du *shaper* doit passer par un ADC⁷ afin qu'il soit numérisé. Alors seulement le DSP sera en mesure de faire le traitement du signal, puisqu'il ne traite que des informations binaires. De plus, le format binaire est pratique pour l'acquisition de données avec un ordinateur. Cette numérisation se fait à une fréquence de 100 MHz ou à toutes les 10 ns.

4.6 DSP

Le développement du DSP se fait avec la technologie des FPGA⁸ qui se fonde sur les transistors, principalement de type MOSFET [2]. Le grand intérêt de cette technologie, qui peut contenir des millions de transistors, est sa flexibilité, qui permet le développement de circuits électroniques en décrivant leurs comportements par des codes de programmation, par exemple Verilog ou VHDL, sans avoir à continuellement fabriquer de nouveaux prototypes.

Les FPGA sont des circuits intégrés constitués de centaines de milliers de cellules logiques, à l'intérieur desquelles se trouvent toute une gamme de fonctions logiques de base. Différents ports/terminaux assurent la communication de la puce FPGA avec l'extérieur, que ce soit pour des signaux en entrée qui seront traités par le circuit implémenté ou pour des signaux en sortie, qui peuvent être enregistrés par un système d'acquisition de données.

Un réseau très complexe d'interconnexions relie entre elles toutes les fonctions logiques à l'intérieur d'une cellule, ainsi que toutes les cellules entre elles et avec les ports/terminaux. Chaque interconnexion est contrôlée par un transistor, dont l'ouverture ou la fermeture du canal de conduction détermine si un courant peut passer ou non par cette connection particulière. Ainsi, l'implémentation d'un circuit électronique sur un FPGA exige que l'état de chaque transistor soit spécifié. Ceci se fait lors de la compi-

⁶*Digital Signal Processing*

⁷*Analog to Digital Converter*

⁸*Field-Programmable Gate Array*

lation du code décrivant la logique du dispositif. Un fichier de configuration contenant cette information est généré, selon un format compris par la carte électronique particulière que l'on utilise, dans ce cas-ci une carte Cyclone V de marque Altera. D'ailleurs, l'ADC est implémenté sur une carte secondaire qui se connecte à la Cyclone V. Quand le fichier de configuration est injecté dans le FPGA, à l'aide du programme QuartusII, celui-ci est emmagasiné dans une mémoire vive, à partir de laquelle la carte peut traiter cette information pour générer et maintenir les tensions nécessaires à l'implémentation du circuit. La présence d'une mémoire Flash permet la sauvegarde d'une copie du fichier, oublié sinon par la mémoire vive aussitôt que l'alimentation de 12 V de la carte est coupée. Le circuit s'implémente alors automatiquement chaque fois que la carte est mise sous tension.

Plus concrètement, à chaque coup d'horloge dans le DSP *hardware*, deux séquences de 32 points consécutifs sont utilisés, avec un chevauchement d'environ 16 points. Ainsi, en temps réel, ces séquences, qui représentent un segment de la forme du signal, couvrent ensemble un intervalle de moins de 64 points (640 ns). À chaque coup d'horloge, les points qui constituent ce segment se décalent d'une position, avec l'introduction à toutes les 10 ns d'une nouvelle valeur d'amplitude numérisée par l'ADC.

Ces deux séquences de 32 points permettent une meilleure précision dans l'extraction des valeurs d'intérêt, puisque la méthode utilisée par le DSP est appliquée une deuxième fois lorsqu'une séquence respecte certains critères de sélection (voir section 4.6.1.2). Comme le signal se déplace dans le temps, il faut considérer un intervalle de temps plus large que 320 ns afin que le DSP ne l'oublie pas avant de réappliquer la méthode.

Pour ce travail, seuls les signaux qui satisfont les critères de sélection à la première évaluation dans le DSP sont enregistrés, sous la forme de 140 valeurs d'amplitude numérisées consécutives (1400 ns) transmises vers le DAQ. Pour chaque signal enregistré, ces 140 valeurs sont traitées à l'aide d'un code C++, d'abord pour définir les formes de référence nécessaires à l'implémentation de la méthode du DSP, puis pour vérifier la qualité de celles-ci en reproduisant cette méthode. De plus, que 140 points consécutifs soient enregistrés est utile pour bien voir et analyser la forme complète des signaux, dont une partie de la queue sera coupée dans les formes de références.

4.6.1 Extraction des valeurs d'intérêt

Deux mesures peuvent être obtenues avec le calorimètre électromagnétique, soient l'énergie déposée par une particule qui traverse le calorimètre et le début du temps d'occurrence de son passage. Ce temps d'occurrence est très important pour bien reconstruire la suite d'événements à travers les détecteurs depuis le point d'interaction. Pour Belle II, ceci permet d'éliminer le bruit en provenance de l'extérieur, par exemple d'origine cosmique. D'ailleurs, tous les détecteurs de l'expérience sont synchronisés par rapport à un même instant de référence, soit lorsqu'un paquet de particules du HER et un du LER entrent en collision au point d'interaction. Quant à l'énergie déposée dans chaque scintillateur, on l'obtient en mesurant l'amplitude des signaux électroniques, qui y est proportionnelle et qu'on insère dans l'équation (4.16).

4.6.1.1 Méthode utilisée

La méthode utilisée dans le DSP par Belle II sera la même que pour son prédécesseur, c'est-à-dire un ajustement linéaire sur une séquence de 32 valeurs d'amplitude consécutives d'un signal $F(t_i - T_0)$ sortant du *shaper* après numérisation par un ADC, dont un exemple est montré sur la figure 4.21, selon l'équation suivante [6][15] :

$$F(t_i - T_0) = Af(t_i - t_{ref}) - At_0 f'(t_i - t_{ref}) + P \quad (4.17a)$$

$$= Af(t_i - t_{ref}) + Bf'(t_i - t_{ref}) + P \quad (4.17b)$$

où $f(t_i - t_{ref})$ est une forme de référence de 32 points dont l'amplitude est normalisée à 1, le paramètre A est notre estimation de l'amplitude du signal, P est le piédestal électronique, t_i est l'instant qui correspond au i -ème des 32 points du signal F utilisés dans l'ajustement, T_0 est l'instant d'occurrence réel du signal, soit quand l'amplitude, sous forme analogue, monte au-dessus du piédestal et t_{ref} est l'approximation de ce temps d'occurrence, soit l'instant qui correspond au premier point de la forme de référence au-dessus du piédestal. Le décalage dans le temps entre le signal mesuré et la forme de référence correspond à $t_0 = T_0 - t_{ref}$. Dans l'équation (4.17a), l'effet de ce décalage sur

l'amplitude est pris en compte avec le deuxième terme, qui est une approximation linéaire où en chaque point on ajoute la variation due à un déplacement de $-t_0$ dans le temps suivant une pente donnée par $Af'(t_i - t_{ref})$, où f' est la dérivée temporelle de la forme de référence f évaluée au temps t_i . Cette dérivée est déterminée en calculant la pente entre deux points consécutifs de $f(t_i - t_{ref})$ séparés par 10 ns, c'est-à-dire :

$$f'(t_i - t_{ref}) = \frac{f(t_{i+1} - t_{ref}) - f(t_i - t_{ref})}{10} \quad (4.18)$$

Ainsi, il faudra considérer 33 points consécutifs pour définir la forme de référence et sa dérivée sur les 32 points nécessaires pour l'ajustement linéaire de l'équation (4.17). Les paramètres à extraire de cet ajustement sont A , B et P . Nous verrons dans les sections qui suivent comment l'ajustement est implémenté. Les 7 premiers points sur les 32 points de la forme de référence sont choisis nuls, comme sur la figure 5.30 par exemple⁹, pour permettre une évaluation fiable du piédestal P . En considérant que l'amplitude de la forme de référence f est normalisée à 1, l'amplitude du signal F est évaluée avec le paramètre A , tandis que le décalage dans le temps est obtenu avec la relation $t_0 = -B/A$, ce qui modifie le temps d'occurrence estimé :

$$T_0 \approx t_{ref} \rightarrow t_{ref} + t_0 \quad (4.19)$$

4.6.1.2 Implémentation dans le DSP

Dans le cas d'une synchronicité parfaite, t_0 est nul et l'amplitude du signal est donnée par A avec une bonne précision. Cependant, les signaux produits par des particules sont généralement asynchrones avec l'ADC, c'est-à-dire qu'il peut y avoir un décalage entre le temps d'occurrence T_0 d'un signal entrant et la numérisation à toutes les 10 ns par l'ADC. Il y a alors une contribution du deuxième terme de l'équation (4.17) et l'amplitude n'est plus strictement donnée par A . Pour améliorer la précision sur le temps d'occurrence au nanoseconde près, on veut générer 10 formes de référence pour des dé-

⁹Sur la figure, 33 points sont affichés, pour les considérations de l'équation (4.18). La numérotation des points commence à 0.

calages de -9 à 0 ns, par pas de 1 ns. Si en principe on peut générer autant de formes de référence qu'on veut, les incertitudes sur les valeurs d'amplitude imposent une limite sur la précision en temps qu'on peut atteindre avec l'ajustement de l'équation (4.17), sans compter qu'une partie des ressources du FPGA sert à mémoriser les coefficients associés à ces formes. C'est pour ces raisons que le pas entre les décalages associés aux formes de référence est de 1 ns. Nous verrons comment obtenir les coefficients à la section 4.6.1.3. Avec un ensemble de formes de référence qui permet l'évaluation d'un décalage t_0 plus petit que 1 ns plutôt que 10 ns, la contribution du deuxième terme de l'équation (4.17) est plus petite et le paramètre A donne alors une valeur plus précise de l'amplitude.

Ainsi, dans le DSP, l'extraction des valeurs d'intérêt se fait en deux évaluations pour en optimiser la précision. La représentation schématique du processus est montrée sur la figure 4.27. À la première évaluation, sur 32 points consécutifs, on suppose que le signal est synchronisé avec l'ADC. On fait donc l'ajustement linéaire de l'équation (4.17b) avec la forme de référence associée à un décalage de 0 ns. Ceci équivaut à estimer le temps d'occurrence comme étant $T_0 \approx t_{ref}$, où t_{ref} est l'instant qui correspond au premier point de la forme de référence au-dessus du piédestal (donc le 8^e point). À toutes les 10 ns, le DSP effectue cette première évaluation sur les 32 valeurs d'amplitude les plus récentes numérisées par l'ADC. Lorsque la superposition de la forme de référence sur un signal est optimale, le paramètre A atteint un maximum, comme sur la figure 4.28 avec la 11^e séquence de 32 points possible avec 140 valeurs d'amplitude enregistrées (1400 ns / 10 ns), soit de 110 ns à 430 ns sur la figure 4.21. L'alignement des signaux sur cet ensemble de 140 valeurs, incluant celles réservées pour l'évaluation du piédestal, est la raison pourquoi on obtient A_{max} avec la 11^e séquence. Si cette valeur A_{max} dépasse un seuil de 375 mV, pour éviter d'enclencher sur du bruit, alors une deuxième évaluation des paramètres A , B et P est faite sur la séquence de 32 points associée à ce maximum, mais cette fois-ci avec la forme de référence qui correspond au décalage $t_{0_1} = -B/A$ obtenu à la première évaluation et arrondi à l'unité près, ce qui équivaut à estimer le temps d'occurrence comme étant $T_0 \approx t_{ref} + t_{0_1}$. De cette deuxième évaluation, un autre décalage en temps $t_{0_2} = -B/A$ est obtenu et l'estimation finale du temps d'occurrence

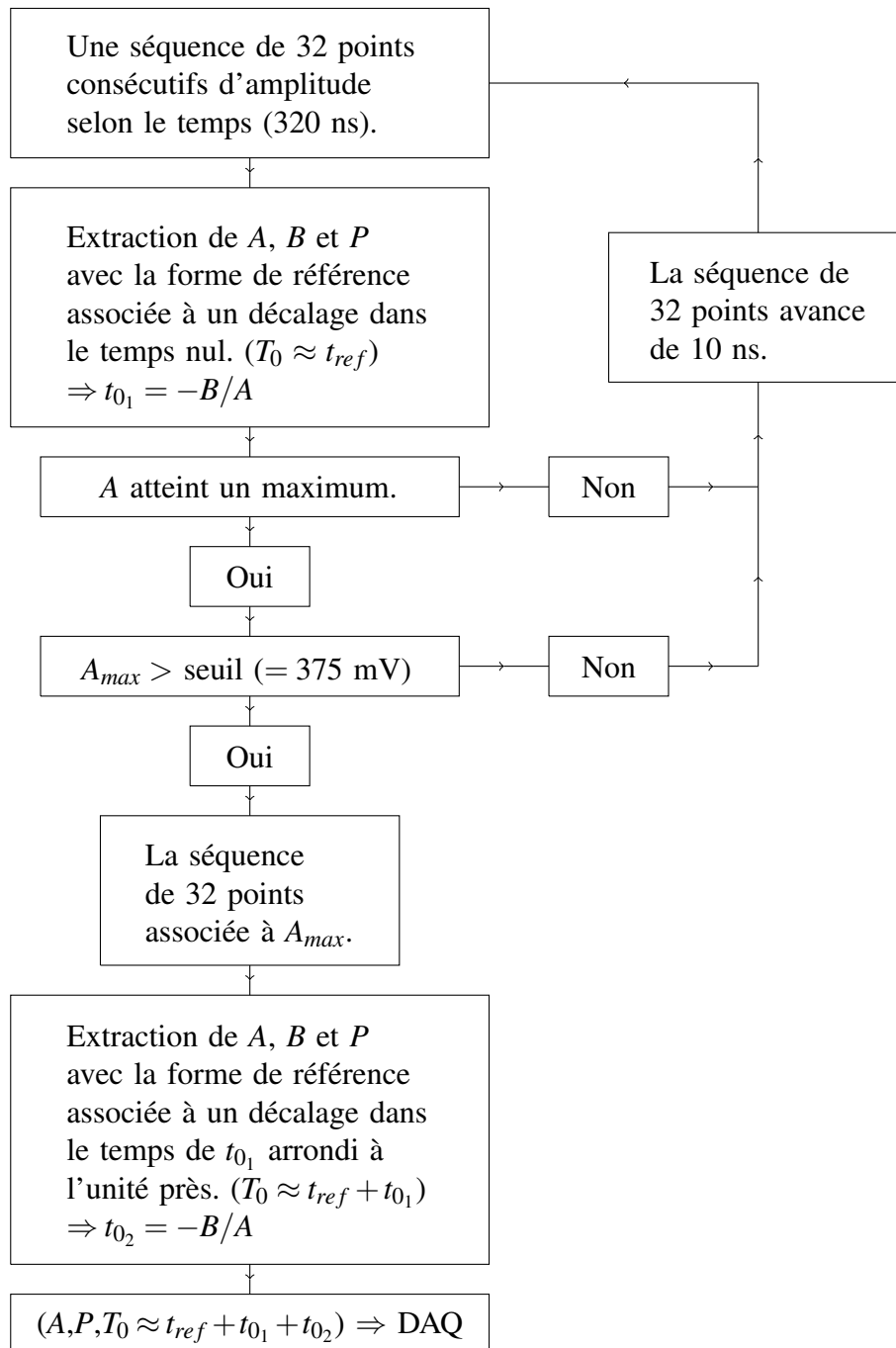


Figure 4.27 : Représentation schématique du traitement numérique d'un signal par le DSP.

devient :

$$T_0 \approx t_{ref} + t_{0_1} + t_{0_2} \quad (4.20)$$

où on s'attend à ce que t_{0_1} soit plus petit que 10 ns et que t_{0_2} soit plus petit que 1 ns et tende vers 0 ns, comme sur la figure 4.30 avec la 11^e séquence de 32 points. De façon équivalente, le paramètre B doit aussi tendre vers 0, comme sur la figure 4.29 avec la 11^e séquence de 32 points. L'évaluation du paramètre P en fonction du temps est montrée sur la figure 4.31. La valeur extraite pour le piédestal est celle obtenue avec l'ajustement optimal, avec la 11^e séquence de 32 points. On voit aussi sur cette figure que la valeur du piédestal P doit atteindre un minimum lorsque l'amplitude vaut A_{max} .

Les figures 4.28 à 4.31 ont été obtenues en appliquant la méthode du DSP sur les 140 valeurs d'amplitudes enregistrées en fonction du temps t_i pour un signal produit par un générateur d'impulsions, en utilisant les ensembles de 3×32 coefficients des formes de référence définies par l'ajustement non-linéaire d'une somme de 3 gaussiennes (voir section 5.1.3). Avec 140 valeurs d'amplitudes, en commençant avec la première séquence de 32 points possible, puis en la déplaçant de 10 ns jusqu'à la fin, il y a en tout 109 séquences de 32 points consécutifs possibles ($140 - 32 + 1 = 109$). On peut donc faire 109 évaluations des paramètres A , B et P , comme sur les figures 4.28 à 4.31, où l'ajustement optimal correspond à A_{max} , soit dans ce cas-ci la 11^e séquence de 32 points. Les résultats sur ces figures sont obtenus à la première évaluation, sauf pour la figure 4.30 où t_{0_2} est extrait à la deuxième évaluation. De plus, un agrandissement a été fait sur cette dernière figure pour mieux voir le comportement dans la région optimale autour de la 11^e séquence de 32 points.

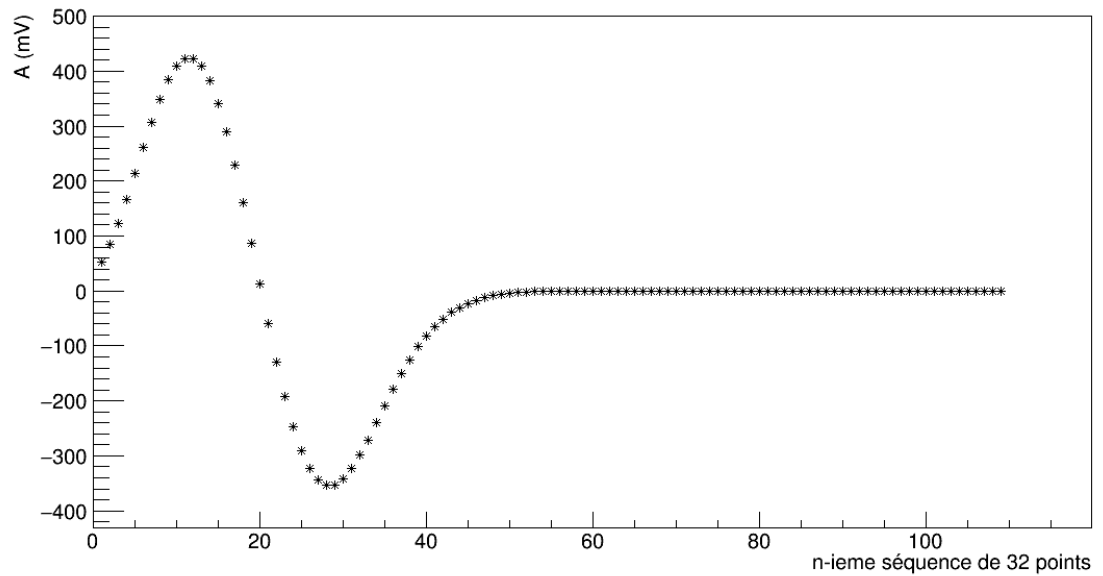


Figure 4.28 : Extraction du paramètre A par la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence obtenues avec la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

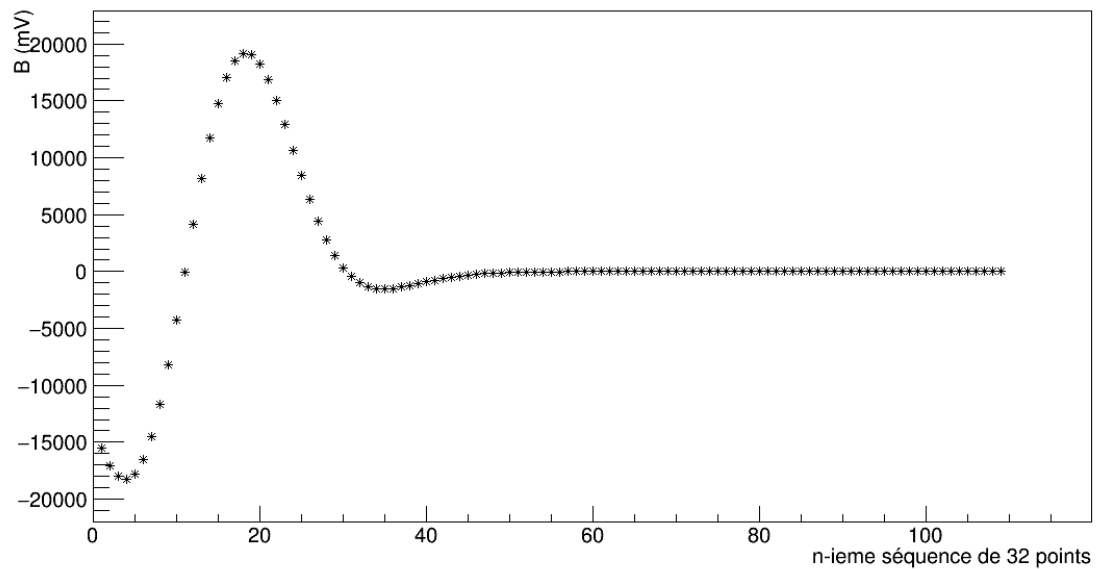


Figure 4.29 : Extraction du paramètre B par la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence obtenues avec la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

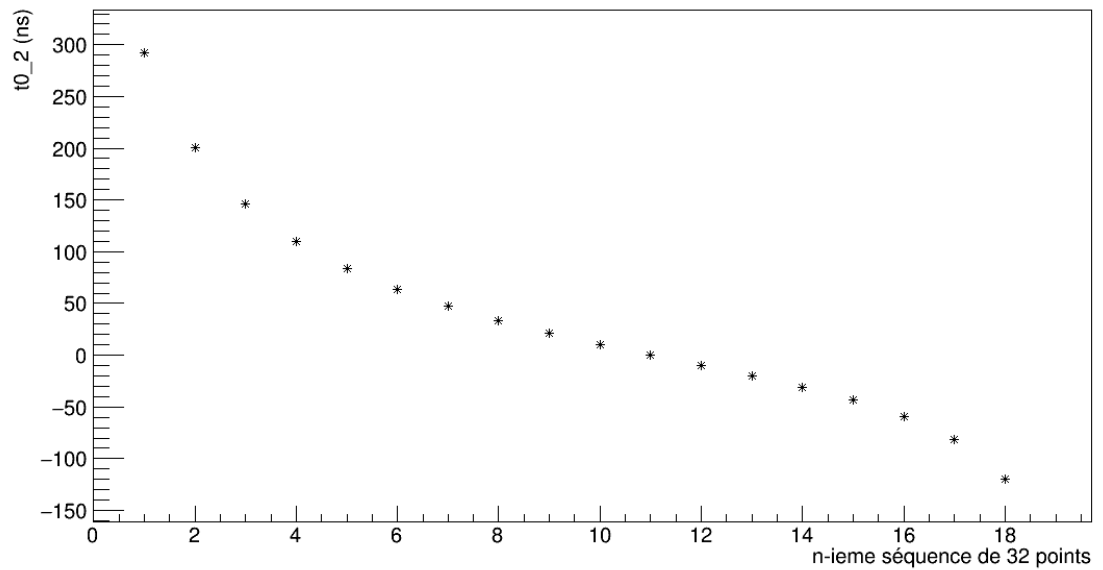


Figure 4.30 : Paramètre t_{0_2} dans la région d'ajustement optimal avec la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

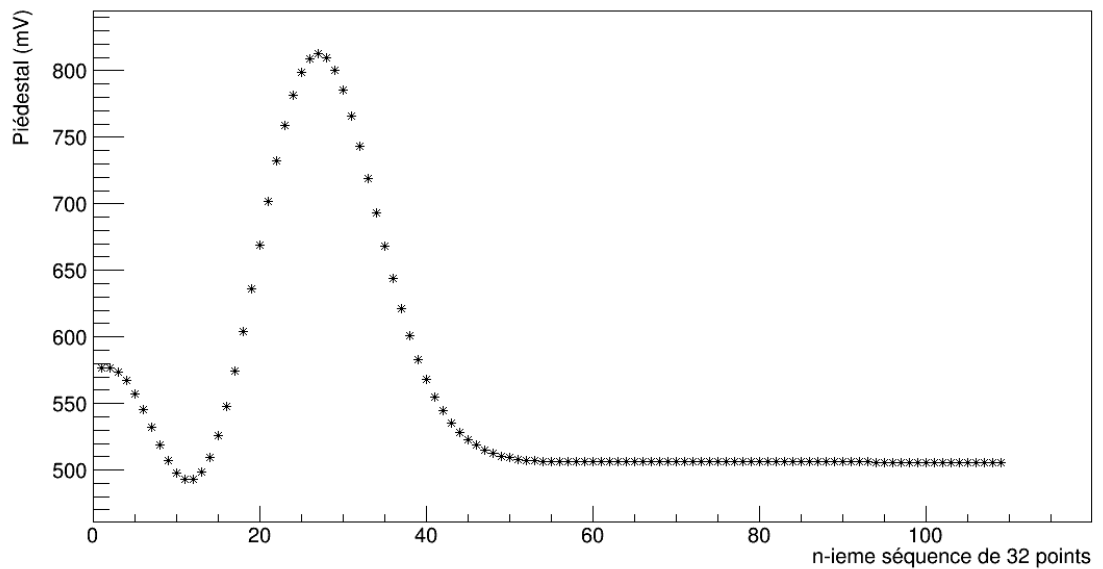


Figure 4.31 : Extraction du paramètre P par la méthode du DSP selon la séquence de 32 points utilisée, avec les coefficients des formes de référence obtenues avec la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

4.6.1.3 Calcul des coefficients

Le problème consiste à évaluer les paramètres A , B et P . Pour ce faire, on cherche à minimiser par régression linéaire le résidu suivant :

$$\chi^2 = \sum_{i=1}^m \left(\frac{y_i - Af(t_i - t_{ref}) - Bf'(t_i - t_{ref}) - P}{\sigma_i} \right)^2 \quad (4.21)$$

où y_i est une mesure de l'amplitude du signal au temps t_i et σ_i est son incertitude. Plus la valeur de χ^2 est faible, plus le modèle utilisé est en accord avec la forme du signal mesuré. En particulier, on définit le χ^2 par degré de liberté, ou $\widetilde{\chi}^2$, comme :

$$\widetilde{\chi}^2 = \frac{\chi^2}{m - n} \quad (4.22)$$

où m est le nombre de points utilisés et n est le nombre de paramètres du modèle. Si le $\widetilde{\chi}^2$ tourne autour de 1, l'accord est excellent.

En supposant que les m valeurs y_i sont parfaitement prédites par le modèle dans l'équation (4.21), on se retrouve avec un système de m équations à 3 paramètres, que l'on peut réécrire sous forme matricielle :

$$FA = Y \quad (4.23)$$

$$\begin{pmatrix} f(t_1 - t_{ref}) & f'(t_1 - t_{ref}) & 1 \\ f(t_2 - t_{ref}) & f'(t_2 - t_{ref}) & 1 \\ \vdots & \vdots & \vdots \\ f(t_m - t_{ref}) & f'(t_m - t_{ref}) & 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ P \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \quad (4.24)$$

Ce système d'équations peut se résoudre en lui appliquant d'abord une diagonalisation, ce qui donne :

$$F^T S^{-1} FA = F^T S^{-1} Y \quad (4.25)$$

où F^T est la transposée de F et S^{-1} est la matrice de déviation covariante. Cette dernière est diagonale si les incertitudes sur les mesures ne sont pas corrélées, avec des éléments

valant $1/\sigma_i$. De plus, nous considérerons que tous les points mesurés sont sujets à la même incertitude, donc que les σ_i ont toutes la même valeur absolue. En définissant la matrice $M = F^T S^{-1} F$, on obtient la solution en multipliant le système sur la gauche par l'inverse de M , avec pour forme finale :

$$A = M^{-1} F^T S^{-1} Y = CY \quad (4.26)$$

où $C = M^{-1} F^T S^{-1}$, de dimensions $3 \times m$, est la matrice de coefficients qui définit une forme de référence et qui sert dans le DSP pour extraire les mesures d'intérêt d'un signal à l'autre. Cette matrice ne dépend que de la forme de référence et de l'incertitude sur chacun des m points. Puisque 32 points sont utilisés pour les ajustements linéaires, chacune des 10 formes de référence a sa propre matrice C de 3×32 coefficients. Les coefficients sont inscrits dans un fichier de format *.mif* qui peut être lu lors de l'implémentation du code sur le FPGA.

4.7 DAQ

Dans notre montage, le DSP effectue un premier tri des signaux selon l'amplitude. Si le seuil de 375 mV est dépassé, alors le signal est transféré vers le DAQ, sous la même forme qu'à la sortie de l'ADC. Ainsi, 140 points d'amplitude par signal sont transférés. Les 16 premiers points servent à l'évaluation du piédestal juste avant l'occurrence du signal.

Le logiciel ZOC7 Terminal est utilisé pour établir le port de communication entre la carte électronique Cyclone V et un ordinateur. Ce dernier sert aussi à implémenter le circuit du DSP sur cette carte électronique via le programme QuartusII avec un code développé en Verilog (par J.-P. Martin). Le logiciel ZOC7 enregistre les signaux sur un fichier de format *.log*. Le contrôle sur l'enregistrement des signaux qui traversent le port de communication se fait à l'aide d'un code C (aussi développé par J.-P. Martin).

Dans l'expérience Belle II, les valeurs qui seront acquises par le DAQ sont l'amplitude estimée avec A , le temps d'occurrence T_0 donné par l'équation (4.20) et le piédestal P .

CHAPITRE 5

FORMES DE RÉFÉRENCE

Les 10 formes de référence utilisées par le DSP pour l'ajustement des signaux peuvent être obtenues soit en prenant une moyenne expérimentale pour chaque décalage dans le temps de 0 à -9 ns, soit en ajustant une fonction analytique sur les signaux, dont les valeurs moyennes des paramètres définissent la forme pour un décalage de 0 ns. Les formes pour les autres décalages sont obtenues en déplaçant la fonction de 1 ns plus tôt, jusqu'à -9 ns. Les deux fonctions analytiques étudiées dans ce travail sont une expression théorique qui décrit la forme des signaux sortant d'un filtre actif et une somme de 3 gaussiennes. L'exercice a été fait pour des signaux de mêmes forme et amplitude produits par un générateur d'impulsions, ainsi que pour des signaux générés dans le ICs par de la radiation ambiante. Chaque forme de référence est une suite de 32 points, dont les 7 premiers sont de valeurs nulles pour permettre une évaluation précise du piédestal. Cette considération est importante lorsqu'il y a possibilité d'empilements comme pour Belle II et peut donner une idée du bruit de fond ambiant. Aussi, chaque forme de référence doit être normalisée à une amplitude de 1 en divisant tous les points par la plus grande valeur d'amplitude, afin que le paramètre A extrait de l'ajustement de l'équation (4.17) soit une mesure fiable de l'amplitude des signaux. De plus, avant de faire les ajustements non-linéaires sur les signaux afin de déterminer les formes de référence, on soustrait la valeur mesurée pour le piédestal, autant dans le cas du générateur d'impulsions que du cristal de ICs.

5.1 Générateur d'impulsions

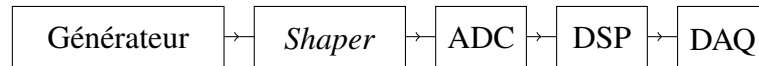


Figure 5.1 : Montage pour déterminer les formes de référence pour des signaux produits par un générateur d'impulsions.

Le montage avec un générateur d'impulsions est illustré sur la figure 5.1. L'utilité d'avoir, à l'entrée du *shaper*, des signaux tous à la même amplitude, fixée à 30 mV et tous de la même forme, semblable à celle des signaux sortant du préamplificateur, est de vérifier la qualité des ajustements des fonctions analytiques et que la méthode implémentée dans le DSP fonctionne correctement. Les impulsions sont générées avec des ondes carrées de 3 V émises à une fréquence de 2 Hz et atténuées d'un facteur 100, avec un temps de descente de 20 μ s qui est ce à quoi on s'attendait à la sortie du préamplificateur. De plus, il n'y a aucun risque d'empilement de ces impulsions.

Puisque la fréquence de 2 Hz est un sous-multiple de la fréquence de 100 MHz de l'ADC, on pourrait s'attendre à ce que les impulsions et le taux de numérisation de l'ADC soient synchronisés. Mais en réalité, ces fréquences sont légèrement différentes. Ainsi, le décalage dans le temps entre les impulsions et le taux de numérisation de l'ADC devrait plutôt varier de façon uniforme avec le temps, dans un intervalle de 10 ns. Les impulsions devraient donc se distribuer de façon uniforme selon le décalage en temps.

Étant donné la grande stabilité des signaux produits par un générateur d'impulsions, la figure 5.2 montre un seul pic sur la distribution d'amplitudes dans la région du piédestal, dont la valeur moyenne est de $493,4 \pm 0,6$ mV. Le seuil de détection utilisé vaut $5\sigma \approx 3$ mV, où σ est l'incertitude du piédestal. Les analyses qui suivent sont toutes faites sur le même ensemble de 10000 signaux.

5.1.1 Formes moyennes expérimentales

Une forme moyenne expérimentale est une forme de référence où chaque point correspond à la moyenne des points associés au même instant sur un ensemble statistique de signaux. Pour obtenir des formes moyennes expérimentales, il faut d'abord séparer

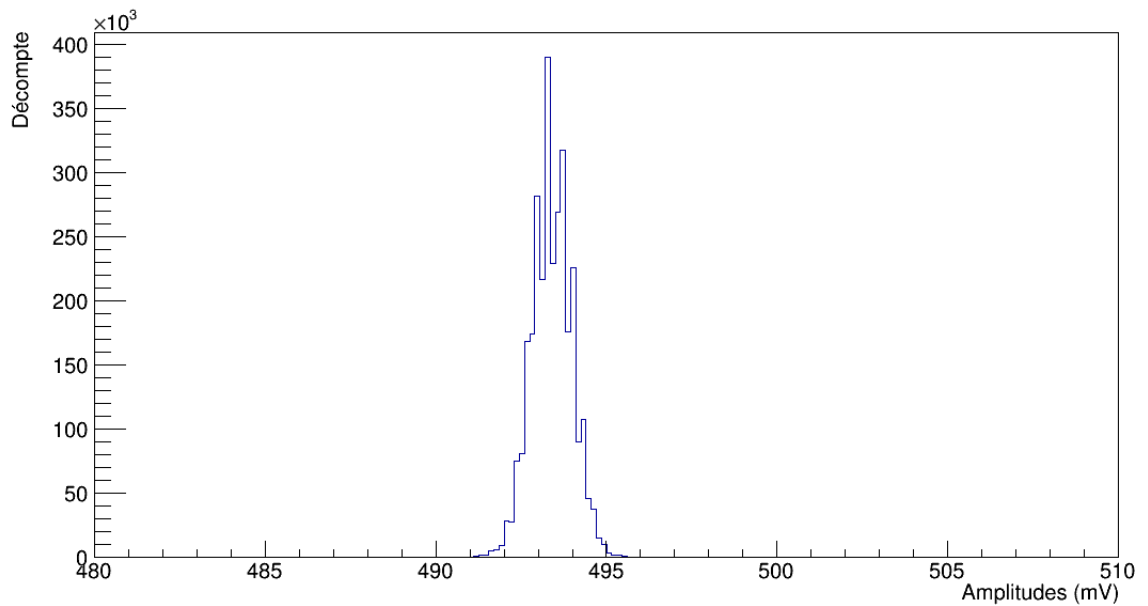


Figure 5.2 : Distribution d'amplitudes des points dans la région du piédestal pour des signaux produits par un générateur d'impulsions.

les impulsions en 10 groupes, qui correspondent chacun à un des décalages en temps de 0 à -9 ns. Pour ce faire, on ajuste un polynôme de puissance 2 autour du sommet du signal avec le point ayant la plus haute amplitude et un de chaque côté, soit le minimum de 3 points nécessaires pour cette puissance (voir annexe I). La fonction quadratique a été choisie, puisque l'amplitude et le décalage estimés avec le minimum de points nécessaires sont indépendants de la puissance choisie, d'où qu'on prend la moins exigeante à évaluer numériquement. Pour ce qui est de décrire la forme complète d'un signal avec un polynôme, les ailes remontent rapidement de part et d'autre de la fenêtre de 32 points utilisée pour l'ajustement, ce qui rend le polynôme final irréaliste dans ces régions. La figure 5.3 montre la distribution d'amplitudes estimées avec l'ajustement d'un polynôme de puissance 2, avec une valeur moyenne d'environ 425 ± 1 mV. En considérant que les signaux entrent dans le *shaper* avec une amplitude de 30 mV, on en conclut que le gain à la sortie est d'environ 14 plutôt que de 20 comme on s'attendait. L'amplitude plus petite que les $493,4 \pm 0,6$ mV du piédestal reflète la gamme dynamique étroite du *shaper*.

Le maximum réel du signal, estimé avec le polynôme de puissance 2, devrait se

trouver dans un intervalle de $[-5, 5[$ ns de part et d'autre du point correspondant à la plus haute amplitude enregistrée pour ce signal. En effet, au sommet, le cas limite est d'avoir deux points de même amplitude, avec le maximum réel au milieu de l'intervalle de 10 ns qui les sépare. Sinon, le maximum réel est plus près du point de plus grande amplitude. On peut donc utiliser ce décalage pour faire la séparation des impulsions en 10 groupes de décalage dans le temps. La distribution de ces décalages est montrée à la figure 5.4. Elle est plutôt uniforme et reste à l'intérieur d'un intervalle de $[-5, 5[$ ns, comme on s'attendait pour le générateur.

Pour ensuite obtenir la forme moyenne expérimentale de chaque groupe de décalage dans le temps, il faut un point de départ commun pour les signaux. Le premier point au-dessus du seuil de détection n'est pas suffisant, puisque des fluctuations juste sous le seuil amènent à sélectionner le point suivant pour certains signaux correspondant au décalage en temps de 0 ns, ce qui les fait paraître 10 ns en avance. Ainsi, à partir d'un deuxième seuil, propre à chaque décalage dans le temps, on détermine le point considéré comme étant le troisième au-dessus du seuil, ou 20 ns après le point de départ. En effet, à ce point, l'écart d'amplitude entre deux points consécutifs est plus grand, ce qui facilite la distinction des signaux qui semblent en avance afin de les réaligner. Une fois que l'alignement est fait, pour chacun des 10 groupes d'impulsions, on peut obtenir la forme expérimentale en prenant la moyenne des points associés au même instant.

Chaque forme de référence expérimentale est une suite de 32 points dont les valeurs sont les moyennes des points correspondant au même instant pour un groupe d'impulsions associées au même décalage temporel. Les formes que l'on obtient sont montrées à la figure 5.5. Il n'y a aucun croisement autre qu'au sommet et qui est dû au décalage entre chaque forme, ce qui est une caractéristique importante à respecter. Contrairement à ce qu'il n'y paraît, les courbes ne se croisent pas toutes au même point. De plus, les sommets sont dans un intervalle d'environ 10 ns, comme il se doit.

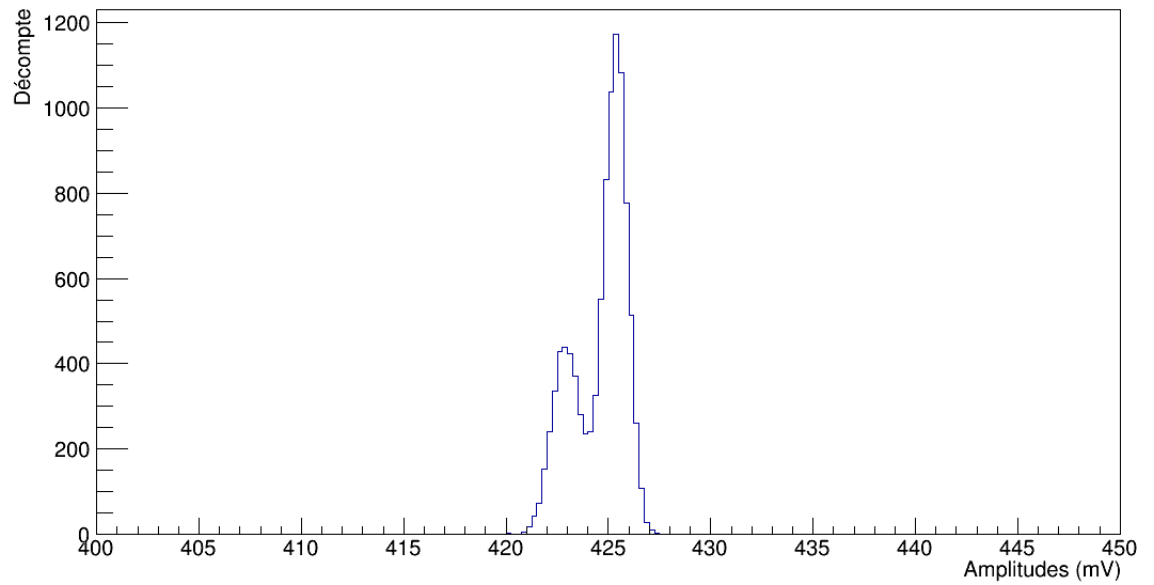


Figure 5.3 : Distribution d'amplitudes maximales pour l'ajustement autour du sommet d'un polynôme de puissance 2 sur des signaux produits par un générateur d'impulsions.

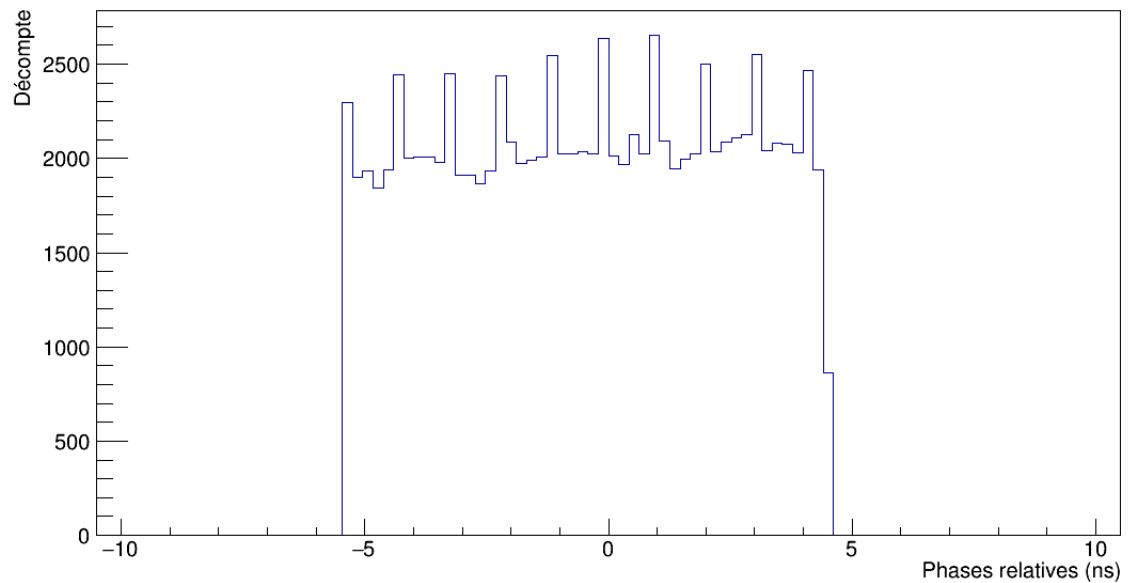


Figure 5.4 : Distribution du décalage dans le temps pour des signaux produits par un générateur d'impulsions.

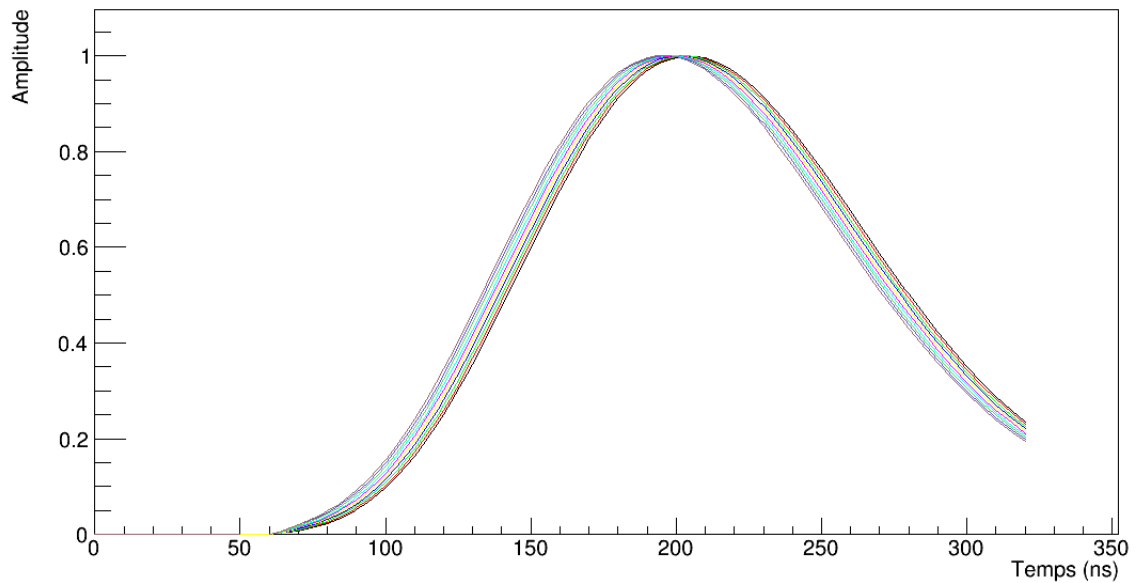


Figure 5.5 : Formes de référence normalisées moyennes expérimentales pour des impulsions produites par un générateur.

5.1.2 Filtres actifs

La forme des signaux sortant d'un filtre actif comme le *shaper* est théoriquement décrite par l'expression suivante [20][24][25] :

$$y(t) = A_{amp} \left[G(t) + A_s e^{-\frac{t}{\tau_s}} \left(1 - e^{-\omega_0 t} \sum_{n=0}^5 \frac{(\omega_0 t)^n}{n!} \right) \right] \quad (5.1)$$

où y est l'amplitude en fonction du temps t en ns écoulé depuis le début du signal avec $y(t=0) = 0$. Le paramètre A_{amp} est un coefficient de proportionnalité pour l'amplitude et son ordre de grandeur détermine l'ordre de grandeur de l'amplitude y , donnée ici en mV. Le deuxième terme, d'amplitude A_s et sans unité, sert à décrire la longue queue du signal, dont la décroissance suit éventuellement la constante de temps τ_s , qui représente le temps de scintillation du cristal. On s'attend à une valeur approximative de 30 ns pour τ_s , soit un peu plus que les 16 ns attendues pour le temps de scintillation de la composante rapide, puisqu'il y a une petite contribution de la composante lente même si la majorité

de celle-ci est coupée par le filtre passe-haut à l'entrée du *shaper*. Le courant qui traverse ce filtre passe-haut dépend de la variation rapide de la tension sur son condensateur à l'entrée, comme décrit à la section 4.4.1. Cette variation rapide de tension est causée par la montée du signal entrant, qui dépend de la scintillation du cristal, dominée par la composante rapide. C'est ce qui détermine la longueur du signal qui sort du *shaper*, d'où que la queue dépend de τ_s . De plus, la fonction $G(t)$ est donnée par [24][25] :

$$G(t) = \frac{a^2}{\tau^4} [\alpha e^{-\omega_d t} + (\cos(\beta t)(\gamma t + \delta) + \sin(\beta t)(\mu t + \chi)) e^{-\omega_0 t}] \quad (5.2)$$

où α , γ , δ , μ et χ sont des paramètres décrits par les expressions :

$$\alpha = \frac{2\pi}{((\omega_d - \omega_0)^2 + \beta^2)^2} \quad (5.3a)$$

$$\gamma = \frac{\pi(\omega_d - \omega_0)}{\beta^2((\omega_d - \omega_0)^2 + \beta^2)} \quad (5.3b)$$

$$\delta = -\alpha \quad (5.3c)$$

$$\mu = \frac{-\pi}{\beta((\omega_d - \omega_0)^2 + \beta^2)} \quad (5.3d)$$

$$\chi = \frac{-\pi(3(\omega_d - \omega_0)\beta^2 + (\omega_d - \omega_0)^3)}{\beta^3((\omega_d - \omega_0)^2 + \beta^2)^2} \quad (5.3e)$$

tandis que le temps de montée τ et le paramètre a sont donnés par :

$$\tau = \sqrt{\frac{1}{\omega_0^2 + \beta^2}} \quad (5.4a)$$

$$a = 3 - 2\tau\omega_0 \quad (5.4b)$$

Ainsi, A_{amp} , A_s , τ_s , ω_0 , ω_d et β sont des paramètres suffisants pour décrire $y(t)$. Ces trois derniers paramètres sont liés à l'électronique du *shaper*, avec entre autres $\omega_d = 1/R_d C_d = 1/RC$ qui est l'inverse de la constante de temps du filtre passe-haut à l'entrée, qui agit comme un dérivateur tel que l'exprime l'équation (4.13). Le point de départ commun pour l'ajustement des signaux est défini de la même façon que pour les formes moyennes expérimentales, mais si pour ces dernières le point de référence déterminé à partir de l'ensemble de deuxièmes seuils est 20 ns après le point de départ,

c'est plutôt 40 ns pour l'expression d'un filtre actif pour permettre à la fonction de mieux s'ajuster par rapport au piédestal.

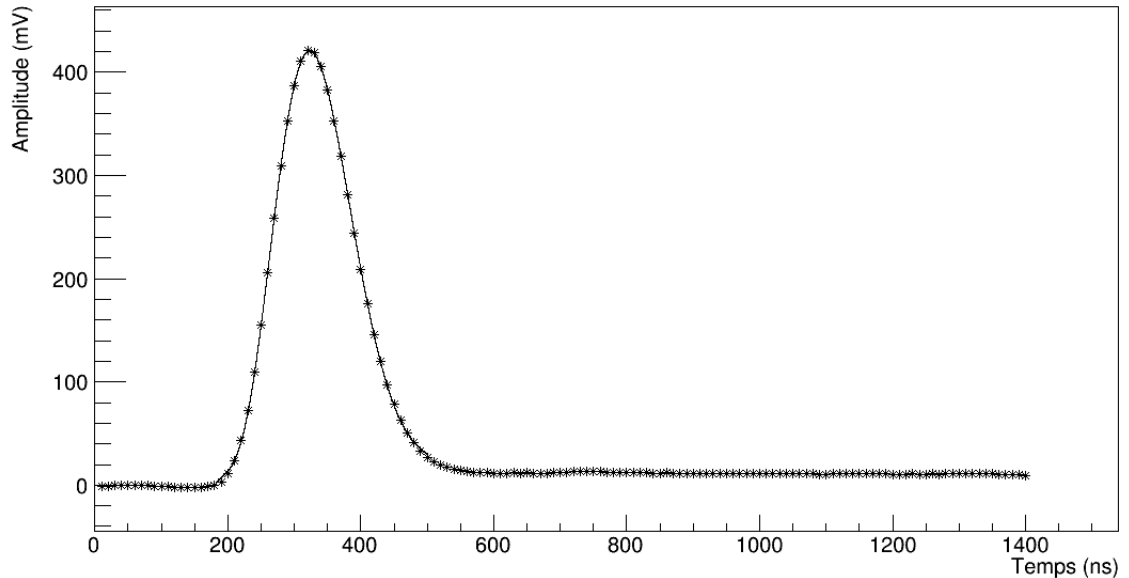


Figure 5.6 : Exemple de l'ajustement de l'expression (5.1) des filtres actifs sur un signal produit par un générateur d'impulsions, après numérisation par l'ADC.

Un exemple d'ajustement de la fonction (5.1) sur un signal du générateur d'impulsions est montré sur la figure 5.6. La distribution d'amplitudes obtenue est montrée sur la figure 5.7, avec une valeur moyenne de 424 ± 1 mV. Les figures 5.8 à 5.13 illustrent les distributions pour les 6 paramètres de l'expression (5.1). On remarque tout d'abord qu'il ne semble pas y avoir de solution stable. Pour A_{amp} , on obtient une valeur moyenne de 41 ± 9 mV, cependant l'amplitude totale de la fonction dépend aussi de la contribution de la queue avec A_s , dont la valeur moyenne de 700 ± 700 ne correspond pas au pic de sa distribution en raison d'un étalement très large vers de plus hautes valeurs. Le temps de scintillation τ_s , en moyenne de 58 ± 6 ns, a une distribution qui reflète l'étalement de celle de A_s , puisque la forme de la queue dépend de ces deux paramètres. On observe le même comportement pour $\beta \approx 1/(43 \pm 1) \text{ ns}^{-1}$, $\omega_0 \approx 1/(53,2 \pm 3,0) \text{ ns}^{-1}$ et $\omega_d \approx 1/(52,5 \pm 2,9) \text{ ns}^{-1}$, qui peuvent aussi affecter la courbure de la queue. De plus, ces 3 derniers paramètres, ainsi que τ_s et A_{amp} , ont tous un pic secondaire qui ressort

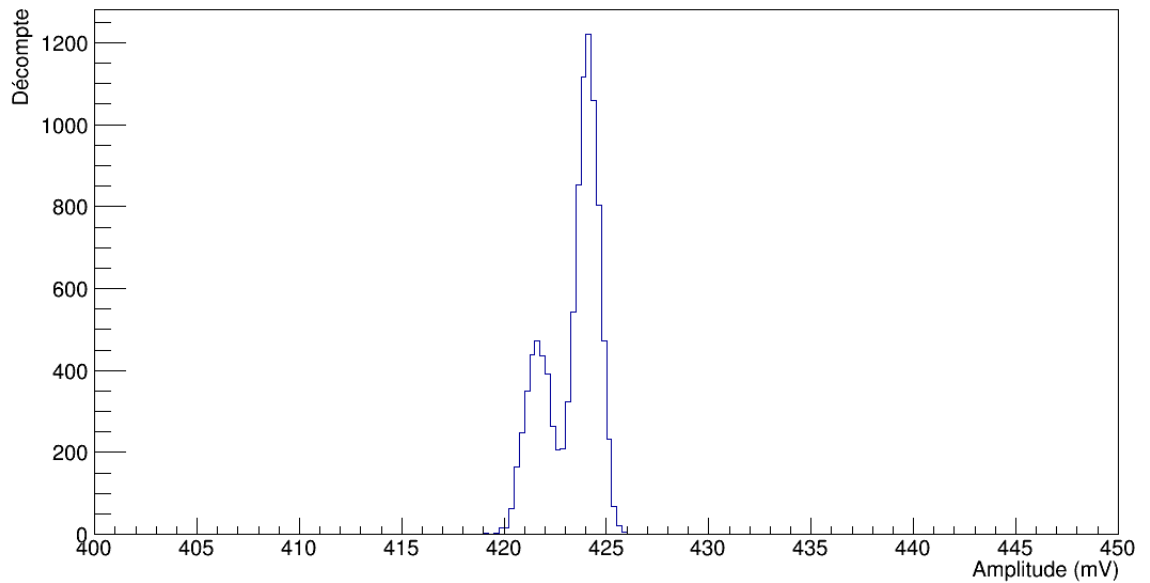


Figure 5.7 : Distribution d'amplitudes pour l'ajustement de l'expression (5.1) des filtres actifs sur des signaux en provenance d'un générateur d'impulsions.

de la distribution, tous avec à peu près la même contribution. Pour τ_s , ce pic secondaire est au centre de la distribution. Sans ces pics, les distributions ont des formes très semblables. Avec 6 paramètres pour la fonction (5.1) et 32 points utilisés pour l'ajustement, le $\widetilde{\chi^2}$ est évalué avec 26 degrés de liberté. La figure 5.14 montre sa distribution. Il y a un deuxième pic de moindre importance, associé aux pics secondaires des distributions des paramètres β , ω_0 , ω_d , τ_s et A_{amp} . Aussi, le manque de stabilité dans les distributions rend les valeurs moyennes obtenues pour les paramètres peu fiables pour définir des formes de référence, malgré un $\widetilde{\chi^2}$ moyen de $1,1 \pm 0,3$. Il est à noter que le temps de montée τ , donné par l'expression (5.4a), vaut 33 ± 2 ns, ce qui est moins de la moitié du vrai temps de montée mesuré, qui est de $78,7 \pm 0,3$ ns.

Pour comprendre les instabilités sur les distributions, il faut savoir que le comportement de l'expression (5.1) est très sensible aux variations de ses paramètres, dont certains peuvent avoir des effets semblables comme A_s , τ_s , β , ω_0 et ω_d , qui peuvent entre autres affecter la forme de la queue. De plus, une légère variation de β peut produire des variations importantes dans le comportement global de la fonction. Il en va de même pour les

paramètres ω_0 et ω_d , qui doivent aussi prendre des valeurs rapprochées, mais différentes l'une de l'autre. Ainsi, il est possible que l'ajustement de cette expression avec la méthode de Levenberg-Marquardt converge vers une solution erronée, soit l'atteinte d'un minimum local par le calcul de variations de paramètres qui semblent optimiser l'ajustement sur le signal, mais dans la mauvaise direction. Le manque de stabilité dans les distributions est donc dû à la sensibilité de la fonction aux variations de ses paramètres. Les formes de référence définies en insérant les valeurs moyennes des paramètres dans l'équation (5.1) sont illustrées sur la figure 5.15.

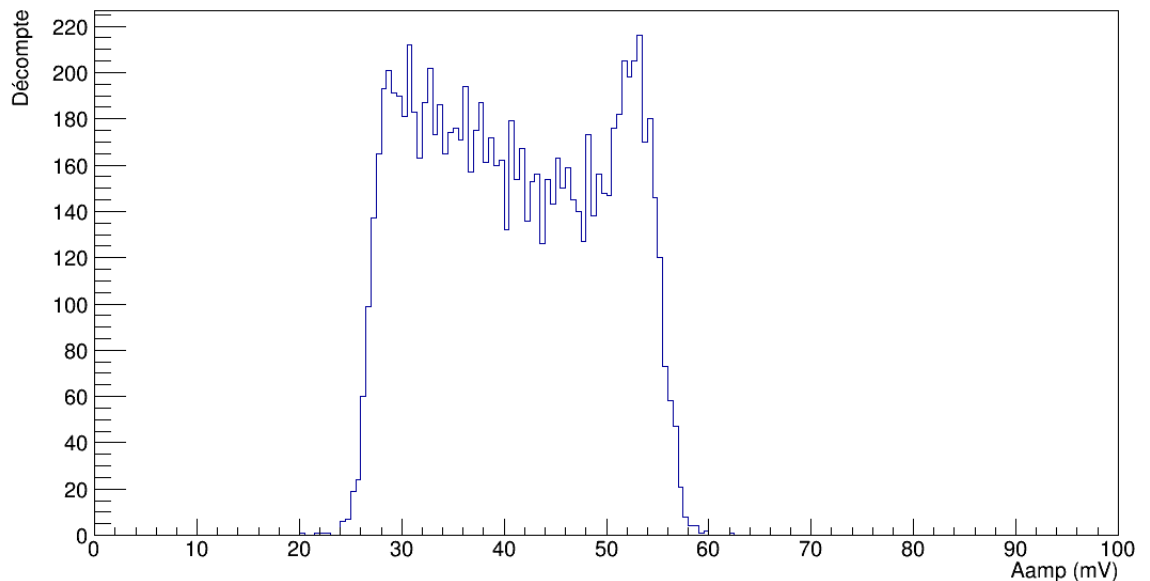


Figure 5.8 : Distribution du paramètre A_{amp} de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.

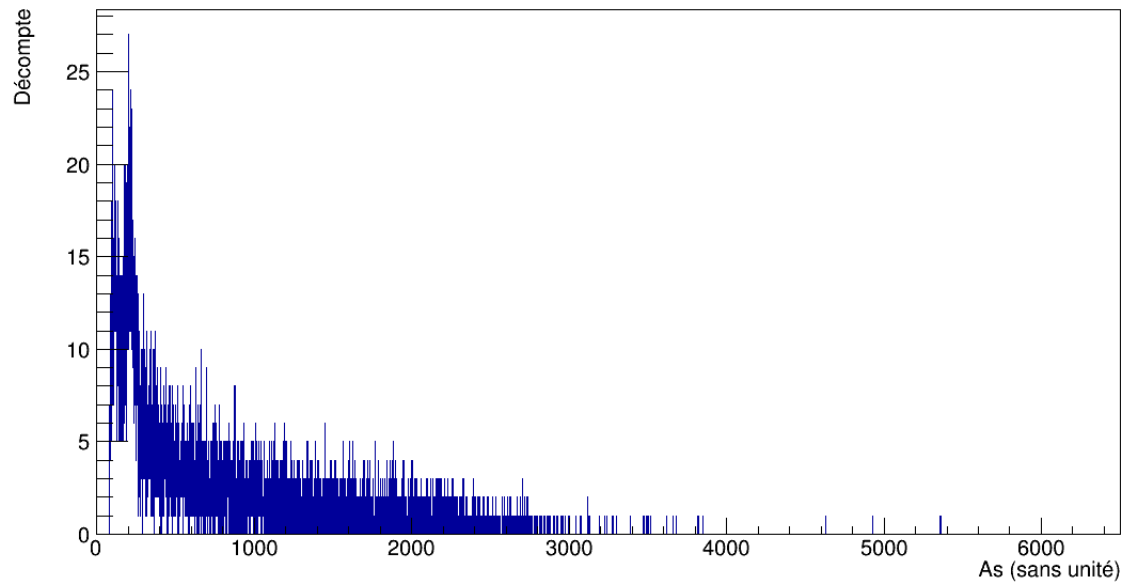


Figure 5.9 : Distribution du paramètre A_s de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.

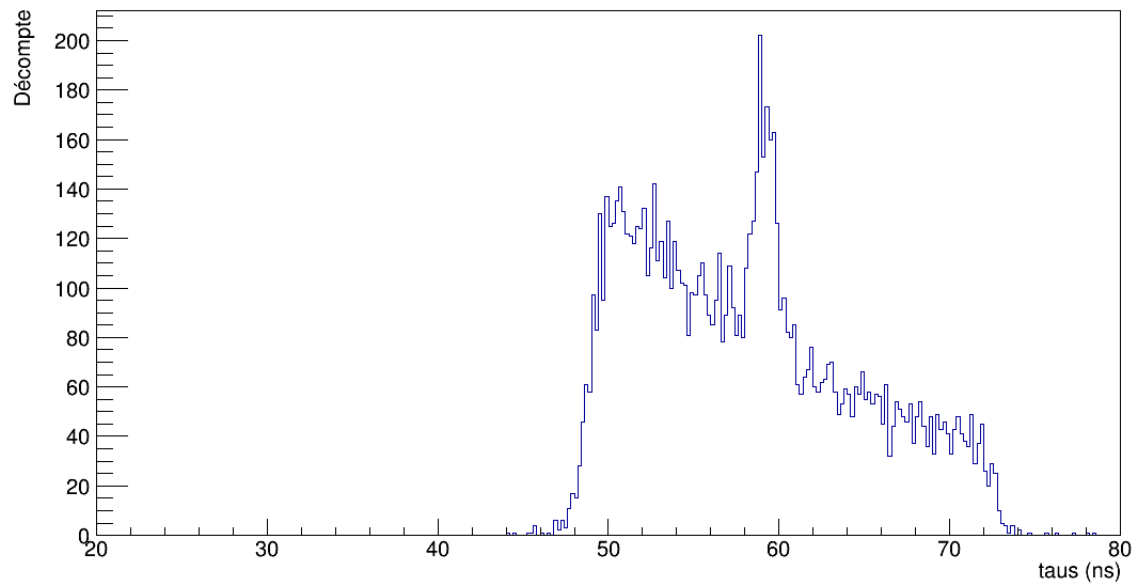


Figure 5.10 : Distribution du paramètre τ_s de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.

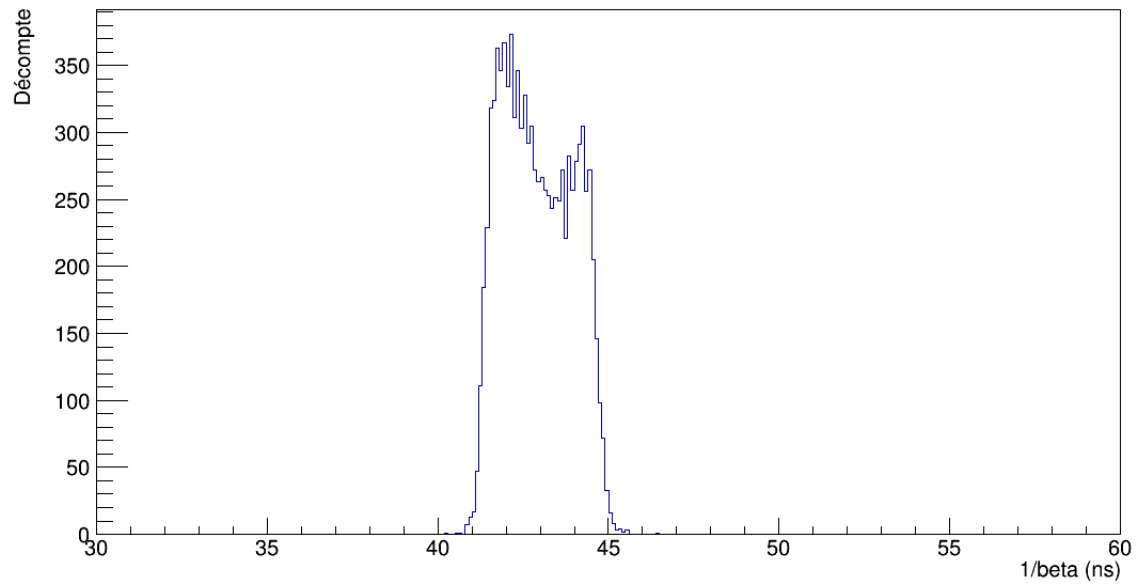


Figure 5.11 : Distribution du paramètre $1/\beta$ de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.

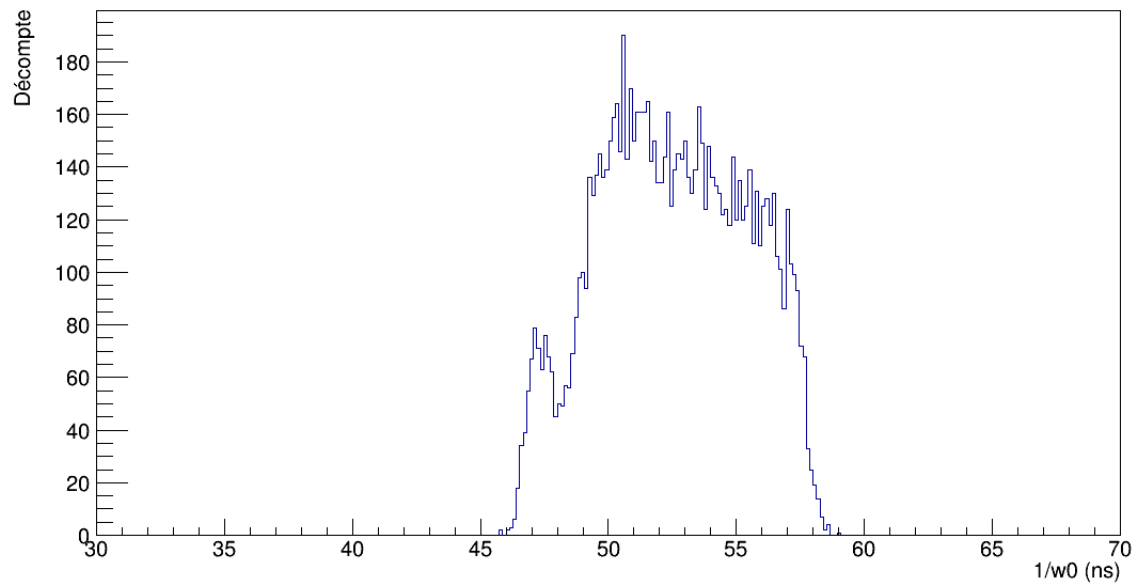


Figure 5.12 : Distribution du paramètre $1/\omega_0$ de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.

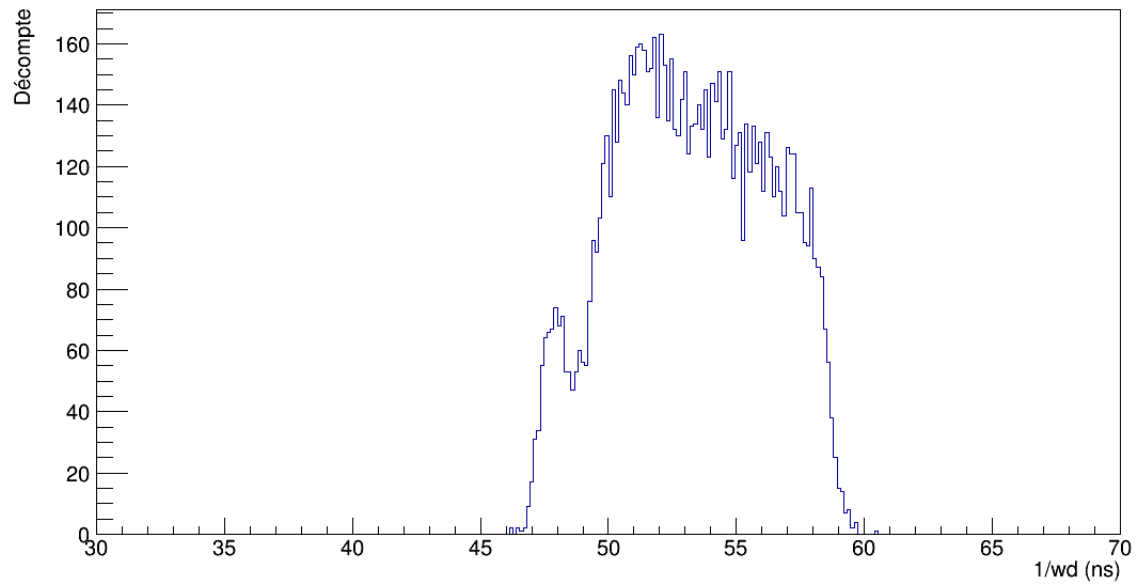


Figure 5.13 : Distribution du paramètre $1/\omega_d$ de l'expression (5.1) des filtres actifs pour des signaux en provenance d'un générateur d'impulsions.

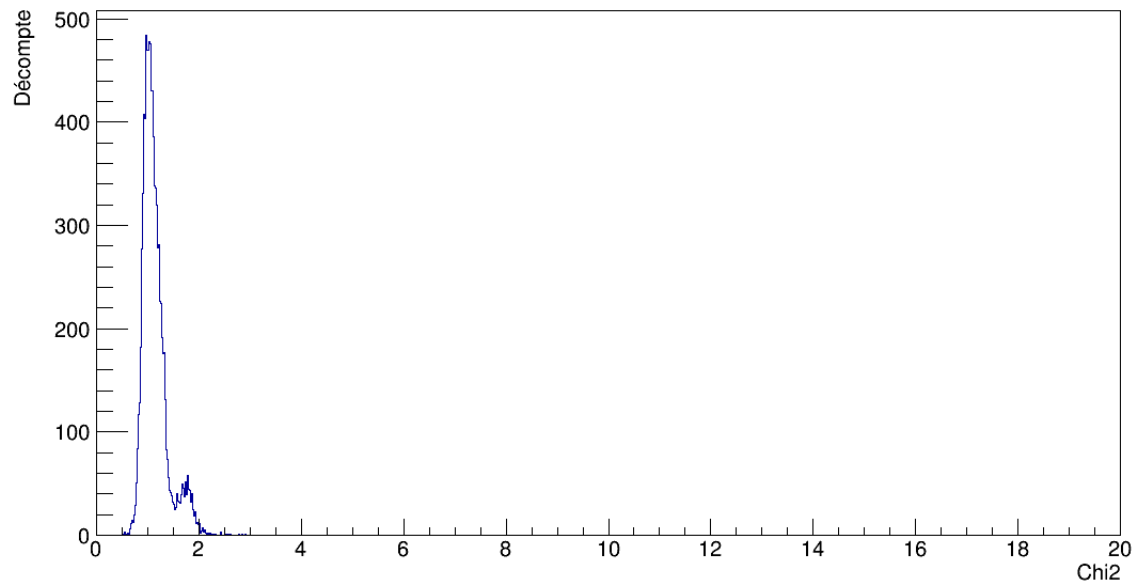


Figure 5.14 : Distribution des $\widetilde{\chi^2}$ individuels pour l'ajustement de l'expression (5.1) des filtres actifs sur des signaux en provenance d'un générateur d'impulsions.

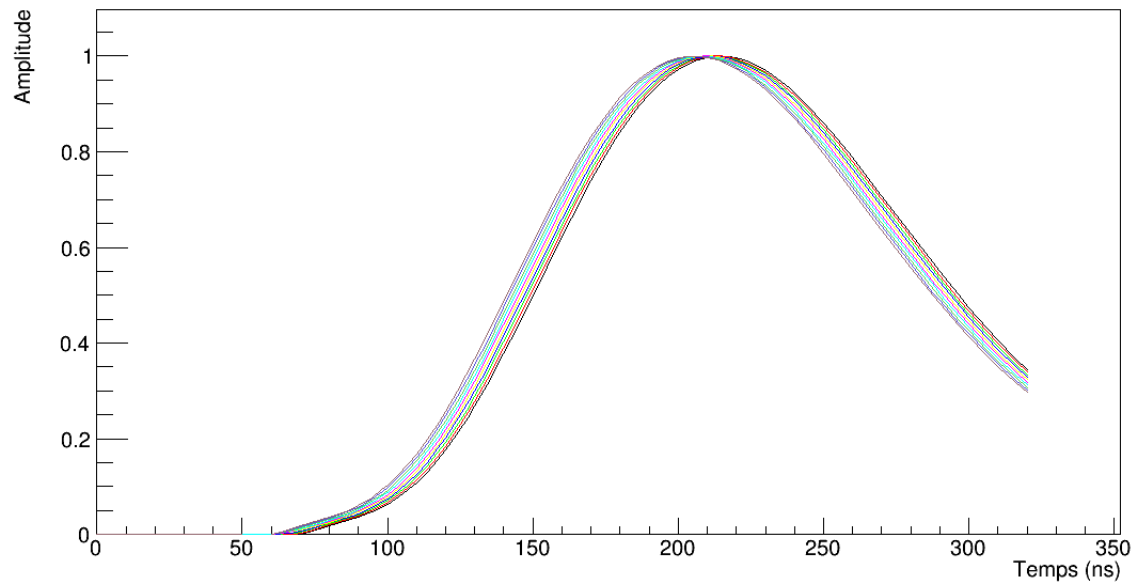


Figure 5.15 : Formes de référence normalisées pour l'expression (5.1) des filtres actifs sur des signaux en provenance d'un générateur d'impulsions.

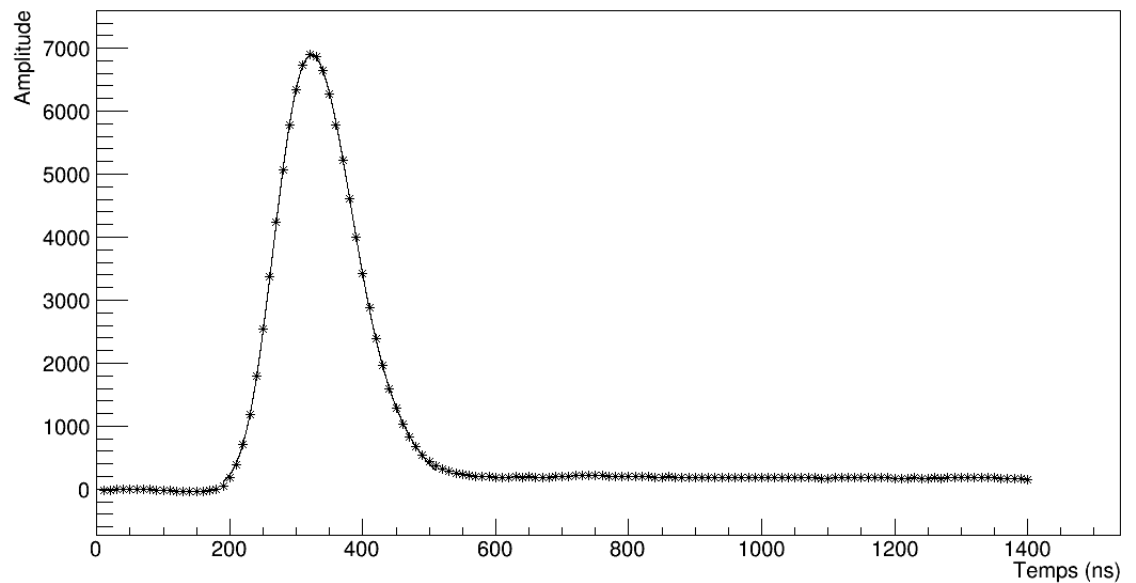


Figure 5.16 : Exemple de l'ajustement d'une somme de 3 gaussiennes sur un signal produit par un générateur d'impulsions, après numérisation par l'ADC.

5.1.3 Somme de 3 gaussiennes

On peut faire l'ajustement d'une fonction exprimée comme une somme de trois gaussiennes sur un signal sortant du *shaper*, telle que

$$y(t) = \sum_{i=1}^3 A_i e^{-\frac{(t-\mu_i)^2}{2\sigma_i^2}} \quad (5.5)$$

où y est l'amplitude en fonction du temps t , A_i est l'amplitude de la i -ème gaussienne, centrée sur le temps μ_i et d'écart-type σ_i . Il y a donc 9 paramètres en tout. La première gaussienne permet de décrire la montée du signal, la deuxième décrit la région du sommet et la troisième décrit la queue. Pour définir le jacobien qu'on insérera dans l'algorithme de Levenberg-Marquardt, les dérivées partielles par rapport à chacun des paramètres pour le j -ème point sont calculées selon les expressions suivantes

$$\frac{dy_j(t_j)}{dA_i} = e^{-\frac{(t_j-\mu_i)^2}{2\sigma_i^2}} \quad (5.6)$$

$$\frac{dy_j(t_j)}{d\mu_i} = A_i \frac{(t_j - \mu_i)}{\sigma_i^2} e^{-\frac{(t_j-\mu_i)^2}{2\sigma_i^2}} \quad (5.7)$$

$$\frac{dy_j(t_j)}{d\sigma_i} = A_i \frac{(t_j - \mu_i)^2}{\sigma_i^3} e^{-\frac{(t_j-\mu_i)^2}{2\sigma_i^2}} \quad (5.8)$$

Un exemple d'ajustement est illustré à la figure 5.16, pour lequel 32 points ont été utilisés à partir du premier au-dessus du seuil. La distribution d'amplitudes totales est montrée sur la figure 5.17. En considérant que l'amplitude totale moyenne estimée avec une somme de 3 gaussiennes est de 422 ± 1 mV, les valeurs pour les paramètres A_i après normalisation des signaux sont respectivement $0,36 \pm 0,01$ (fig. 5.18), $0,705 \pm 0,005$ (fig. 5.19) et $0,21 \pm 0,01$ (fig. 5.20). Pour les μ_i on a 283 ± 2 ns (fig. 5.21), 333 ± 2 ns (fig. 5.22) et 396 ± 3 ns (fig. 5.23) et le plateau d'environ 10 ns de large qu'on observe sur leurs distributions est dû à la fréquence de numérisation de l'ADC et reflète la distribution uniforme des décalages dans le temps obtenus avec les impulsions d'un générateur. Par contre, μ_3 est moins affecté, puisque ce paramètre s'attache à décrire la

queue du signal. De plus, les différences $\mu_2 - \mu_1$ et $\mu_3 - \mu_2$, et donc le comportement global de la fonction ajustée, sont plus stables que les paramètres eux-mêmes avec les valeurs $50,0 \pm 0,3$ ns et 63 ± 1 ns, telles qu'on le voit sur les figures 5.27 et 5.28. Ceci indique une flexibilité dans l'ajustement d'une somme de 3 gaussiennes. Quant aux paramètres σ_i , les valeurs moyennes sont $36,6 \pm 0,4$ ns (fig. 5.24), $46,8 \pm 0,4$ ns (fig. 5.25) et $63,6 \pm 0,8$ ns (fig. 5.26). À la figure 5.29, on voit que le $\widetilde{\chi^2}$ individuel moyen, évalué avec 23 degrés de liberté, est de $1,1 \pm 0,1$, ce qui signifie un très bon accord.

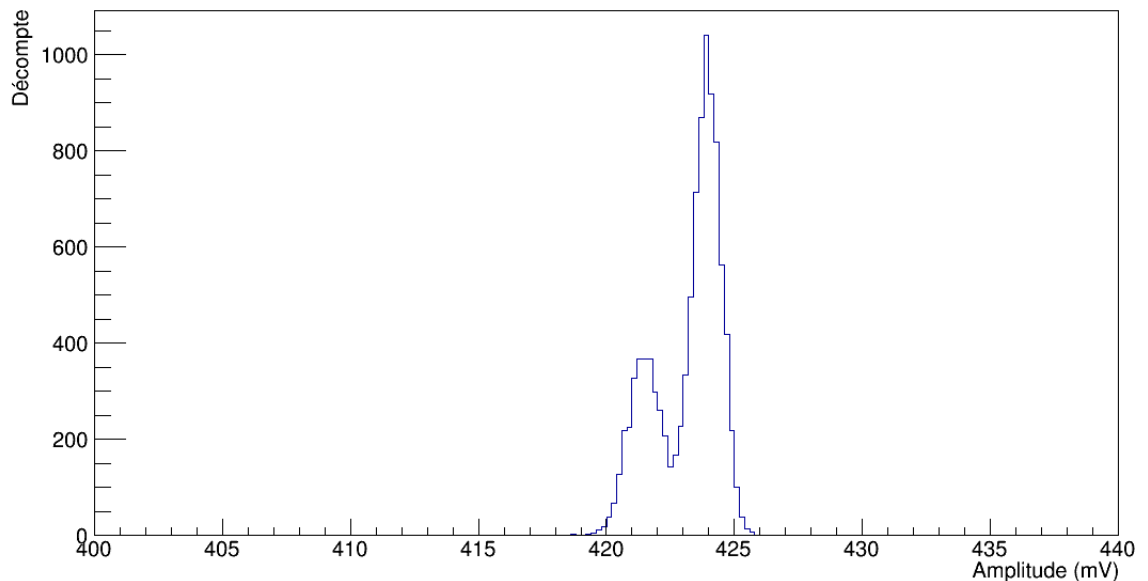
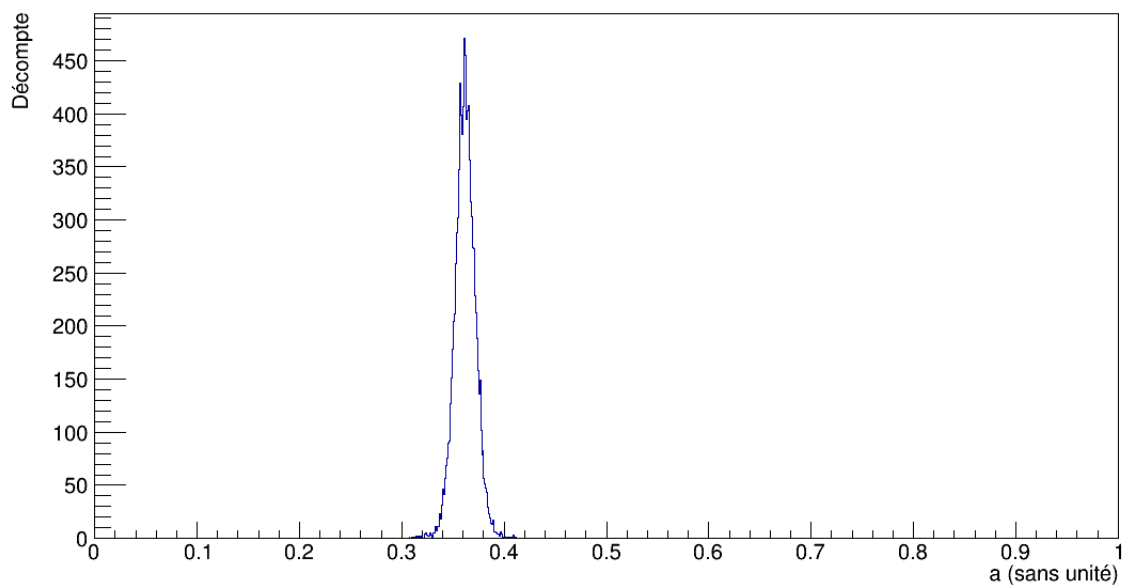
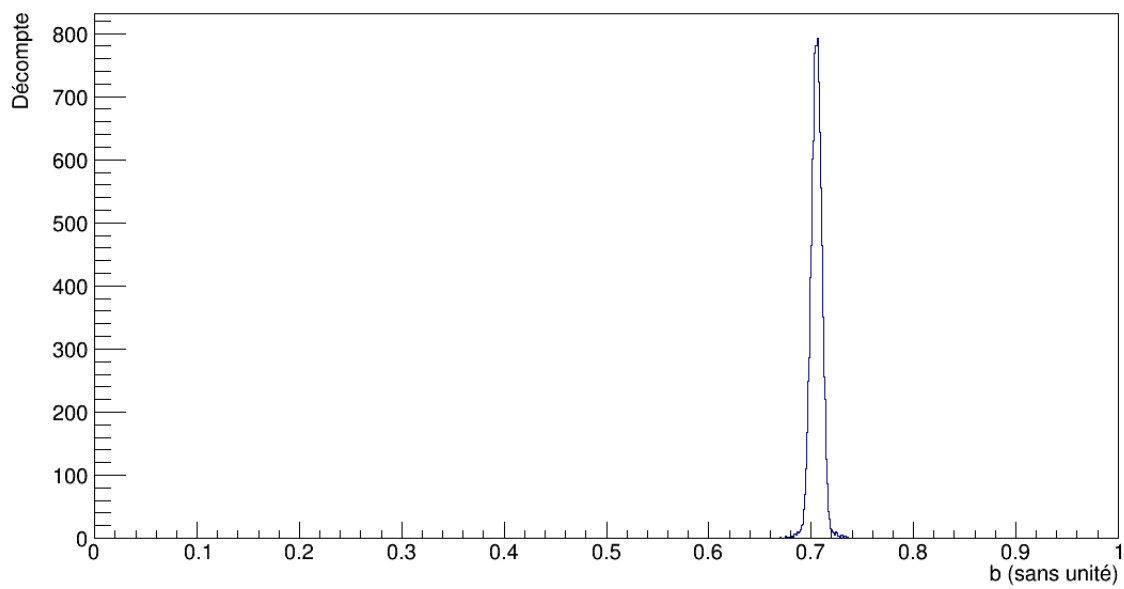
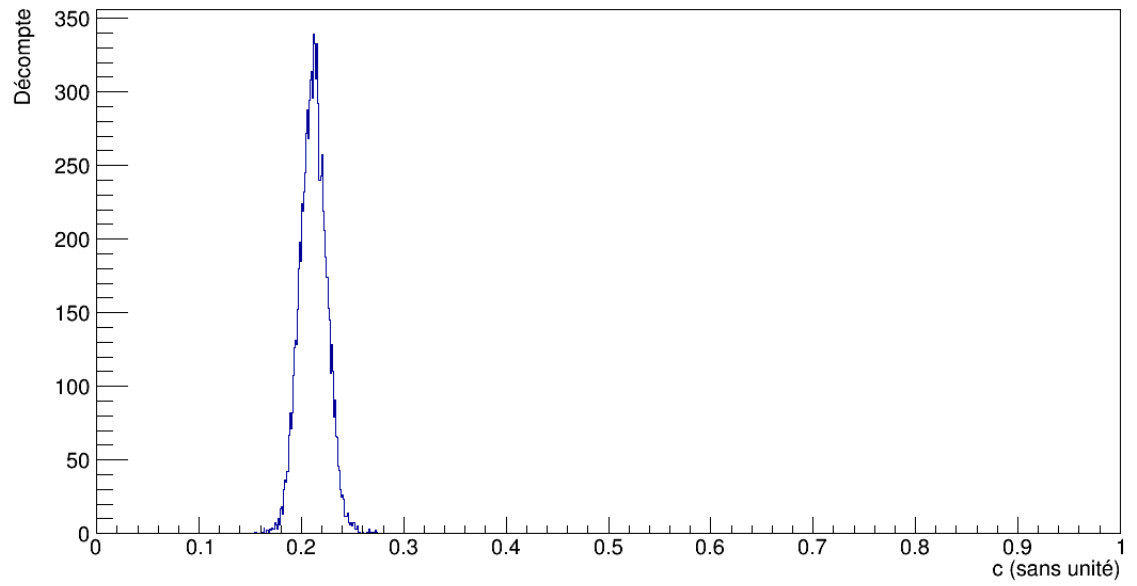
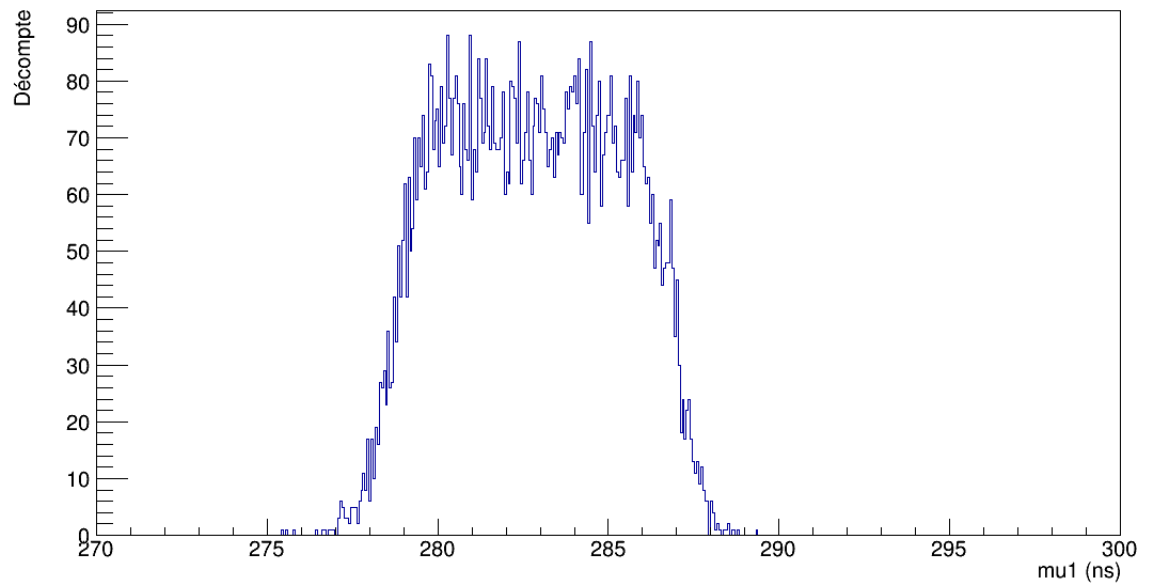
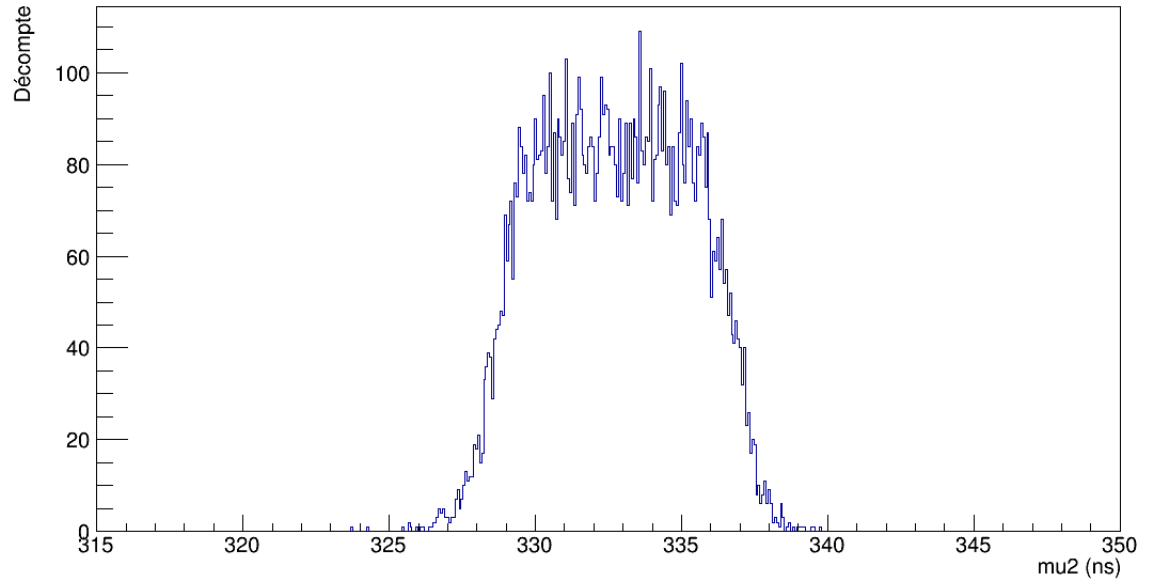
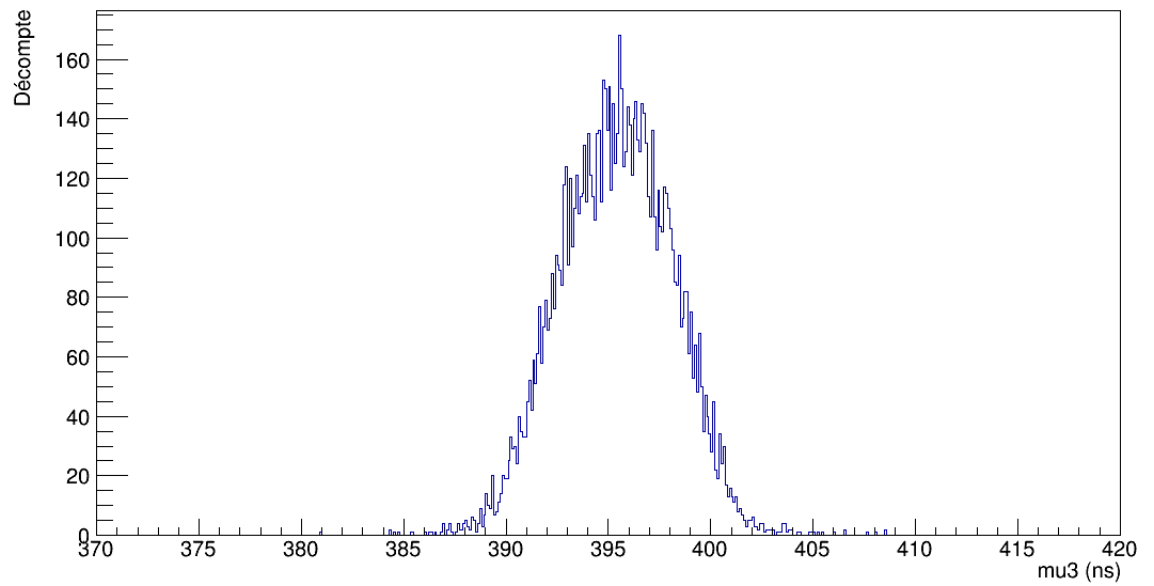


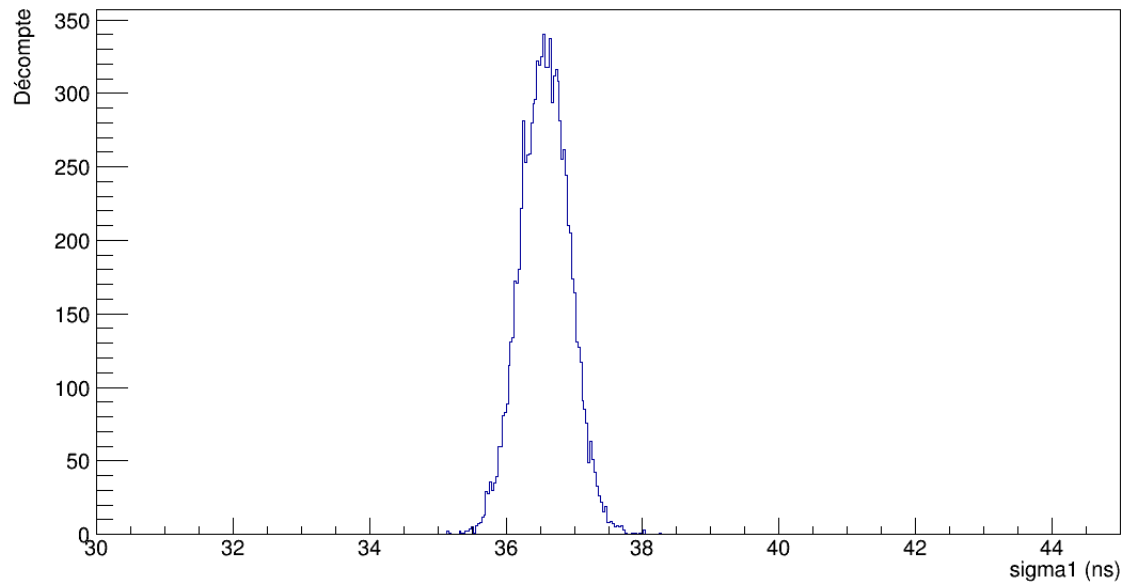
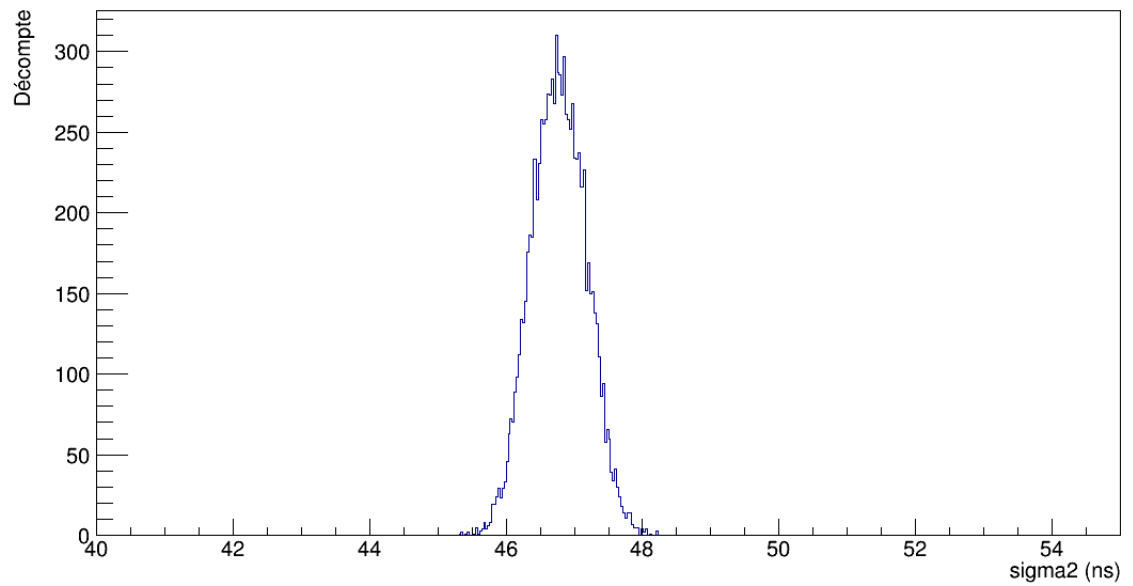
Figure 5.17 : Distribution d'amplitudes pour les ajustements individuels d'une somme de 3 gaussiennes sur des signaux en provenance d'un générateur d'impulsions.

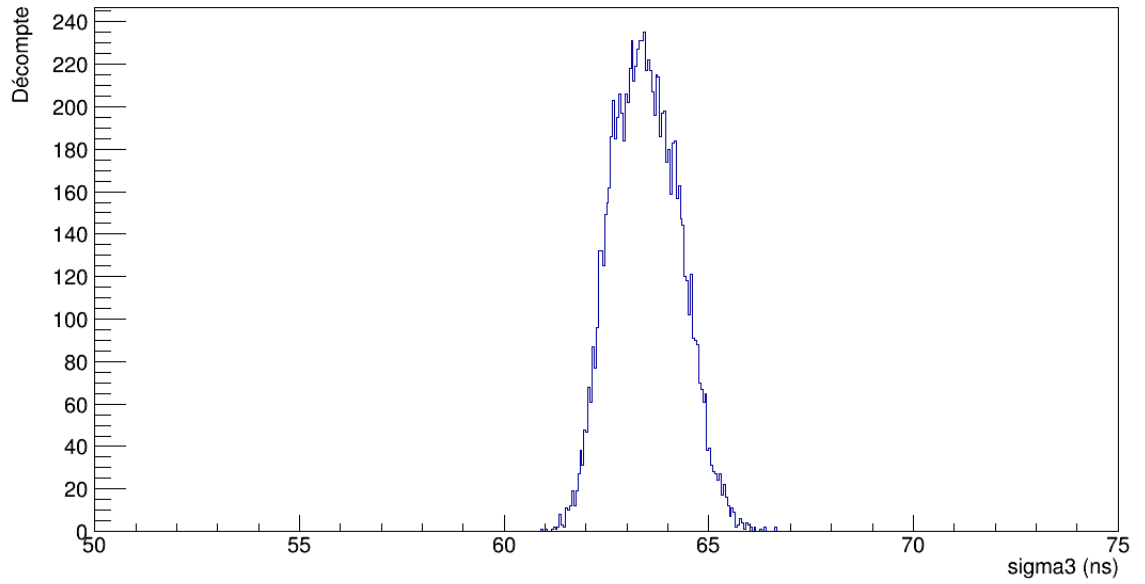
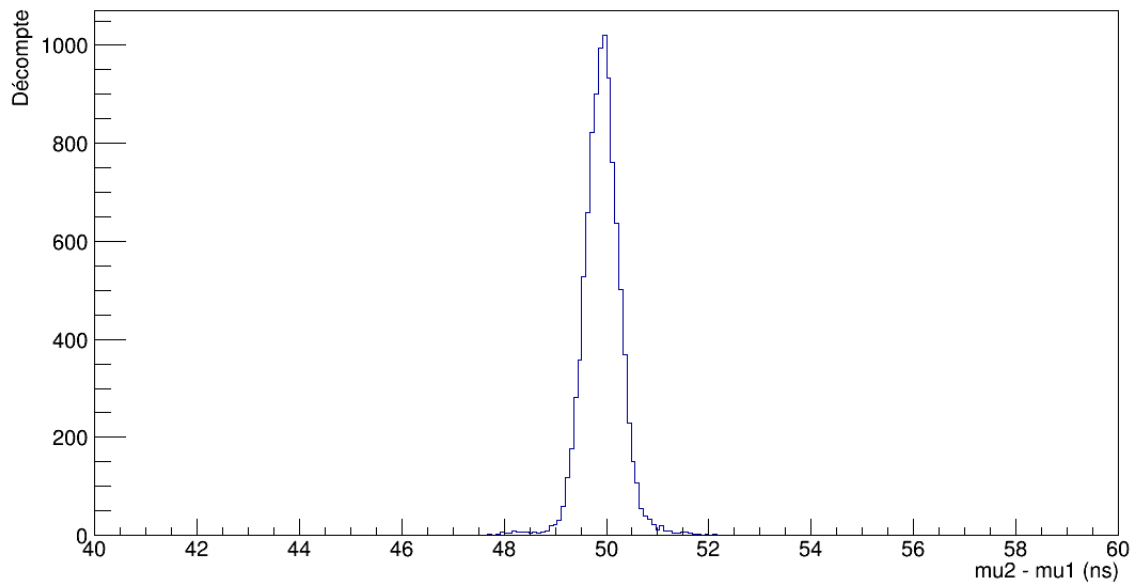
Avec les valeurs optimales moyennes obtenues pour les 9 paramètres après ajustement de l'équation (5.5) sur tous les signaux, on peut définir les formes de référence en insérant ces valeurs dans la même équation. La difficulté avec la somme de 3 gaussiennes est de lui définir un début, puisqu'une fonction gaussienne n'atteint jamais 0. Pour s'assurer que le début de ces formes soit lisses, on choisit un point de départ qui correspond à 1% de l'amplitude maximale. On peut alors sélectionner 32 points, par intervalles de 10 ns. On répète ensuite l'opération en déplaçant le point de départ de 1 ns et ainsi de suite jusqu'à ce qu'on ait 10 ensembles de 32 points. Ces formes de références

Figure 5.18 : Distribution du paramètre A_1 .Figure 5.19 : Distribution du paramètre A_2 .

Figure 5.20 : Distribution du paramètre A_3 .Figure 5.21 : Distribution du paramètre μ_1 .

Figure 5.22 : Distribution du paramètre μ_2 .Figure 5.23 : Distribution du paramètre μ_3 .

Figure 5.24 : Distribution du paramètre σ_1 .Figure 5.25 : Distribution du paramètre σ_2 .

Figure 5.26 : Distribution du paramètre σ_3 .Figure 5.27 : Distribution de la différence $\mu_2 - \mu_1$.

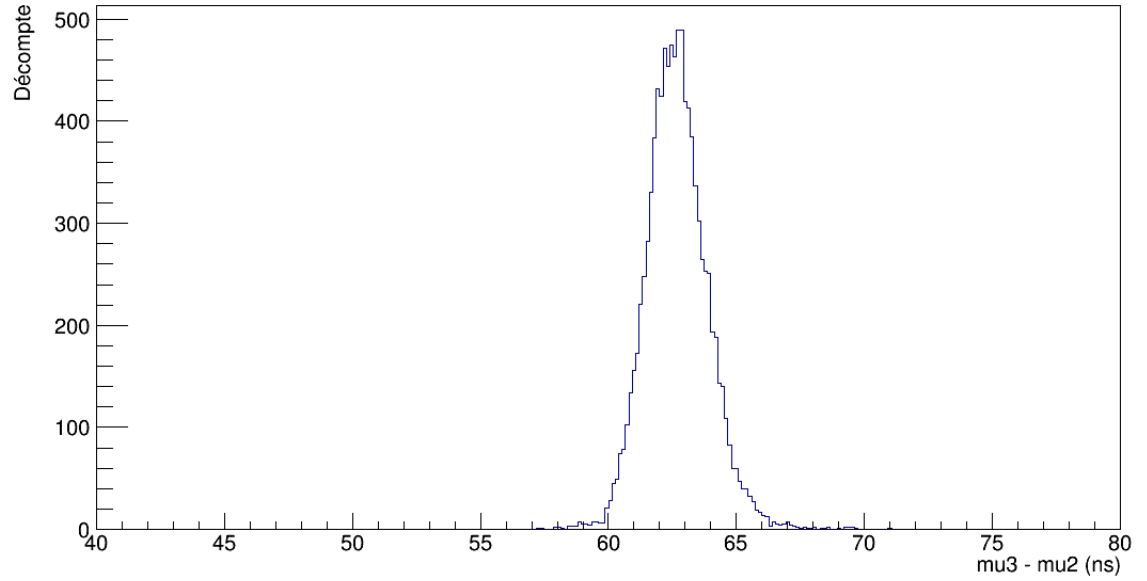


Figure 5.28 : Distribution de la différence $\mu_3 - \mu_2$.

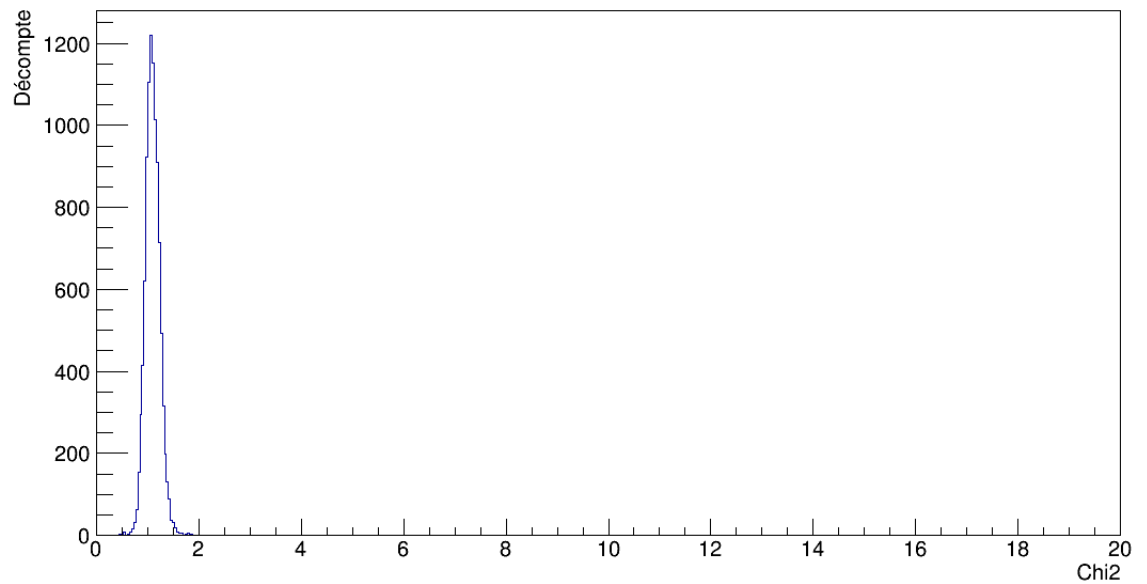


Figure 5.29 : Distribution des $\tilde{\chi}^2$ individuels pour l'ajustement d'une somme de 3 gaussiennes.

sont présentées à la figure 5.30.

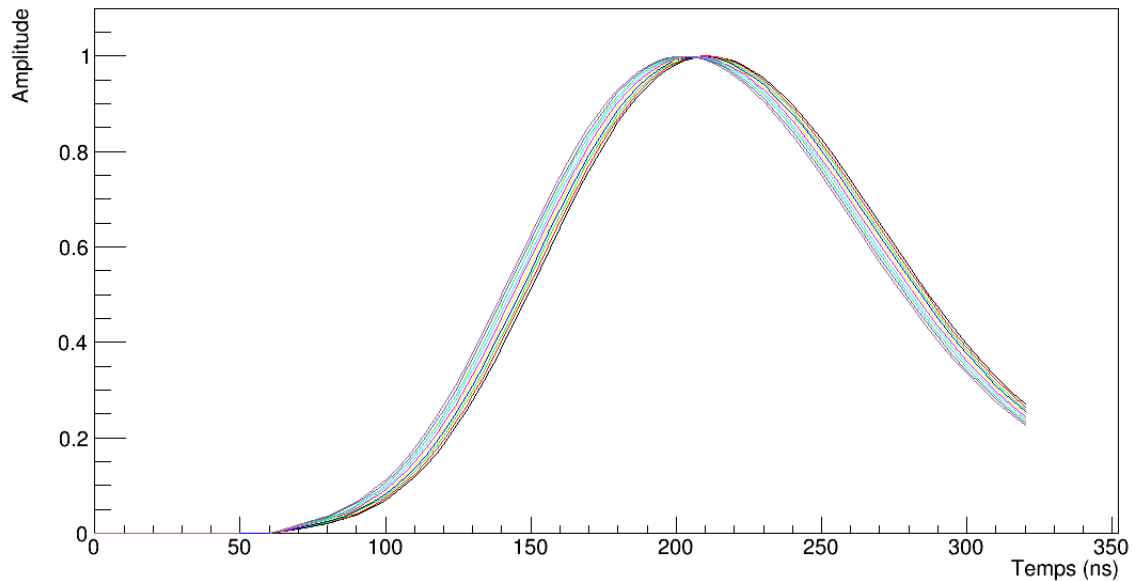


Figure 5.30 : Formes de référence normalisées pour la somme de 3 gaussiennes.

5.1.4 Comparaison

Les formes moyennes expérimentales sont les plus rapides à obtenir. Pour une forme moyenne expérimentale, chaque point est la moyenne des points qui correspondent au même instant pour un groupe d'impulsions associé à un des décalages dans le temps de 0 à -9 ns. Quant aux fonctions analytiques, les formes de référence pour un filtre actif sont les plus longues à obtenir, en raison du plus grand nombre d'itérations nécessaires pour les ajustements individuels, soit environ 126 en moyenne contre 90 pour la somme de 3 gaussiennes. Cependant, la qualité des ajustements est d'un plus grand intérêt, puisque c'est à partir des valeurs moyennes qu'on obtient pour les paramètres que sont définies les formes de référence. La valeur du $\widetilde{\chi^2}$ de $1,1 \pm 0,1$ pour la somme de 3 gaussiennes est semblable à celle de $1,1 \pm 0,3$ pour la fonction du filtre actif, mais son incertitude est plus petite. Aussi, les paramètres obtenus pour les 3 gaussiennes sont beaucoup plus stables que pour le filtre actif. Ainsi, pour les impulsions produites par un

générateur, les formes de référence pour la somme de 3 gaussiennes sont les plus fiables. Pour ce qui est des formes moyennes expérimentales, elles semblent tout aussi fiables, mais leur qualité sera vérifiée à la section 5.3, puisqu'elles n'ont pas été définies à partir d'ajustements individuels comme pour les deux autres méthodes. Cependant, la somme de 3 gaussiennes a l'avantage de s'ajuster au comportement de tout le signal.

5.2 Cristal de ICs

Le montage avec un cristal de ICs est illustré sur la figure 5.31. Les signaux obtenus avec des particules de la radiation ambiante qui traverse un cristal de ICs pur n'ont évidemment pas tous la même amplitude. On s'attend par contre à ce qu'ils aient tous une forme semblable après le *shaper*. Étant donné que Belle II ne distinguera pas les photons des électrons ou des positrons qui traversent le cristal, il n'est pas nécessaire d'utiliser un circuit de coïncidence pour sélectionner un type de particule (annexe IV). Aussi, un atténuateur de 10 dB (facteur d'environ 3,16) [8] a été ajouté entre le préamplificateur et le *shaper* afin de limiter les problèmes de saturation. Malgré tout, comme le *shaper* a une gamme dynamique très étroite, la majorité des signaux enregistrés avec le ICs sont dans ce régime de saturation, sans toutefois aveugler le *shaper*. On peut donc quand même étudier la forme des signaux qui en sortent. Ces derniers sont normalisés en divisant par l'amplitude maximale obtenue par ajustement.

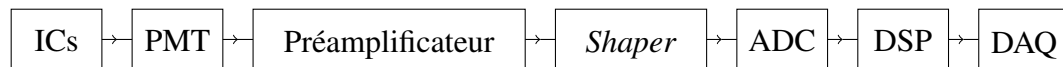


Figure 5.31 : Montage pour déterminer les formes de référence pour des signaux générés dans un cristal de ICs.

Le piédestal pour les signaux en provenance du ICs est de 1013 ± 7 mV. La distribution est illustrée sur la figure 5.32. Ainsi, le seuil de détection, défini à partir de l'étalement du bruit autour du piédestal, est de $5\sigma = 35$ mV, où σ est l'incertitude du piédestal. Afin de couper de l'ensemble statistique les interférences qui se sont rendues jusqu'au DAQ, deux autres seuils ont été utilisés dans le code C++ développé pour l'analyse des signaux. L'amplitude ne devait jamais descendre en-dessous de -100 mV et devait toujours dépasser 330 mV au sommet. Le même ensemble statistique de 15296 signaux a été utilisé pour les trois approches étudiées.

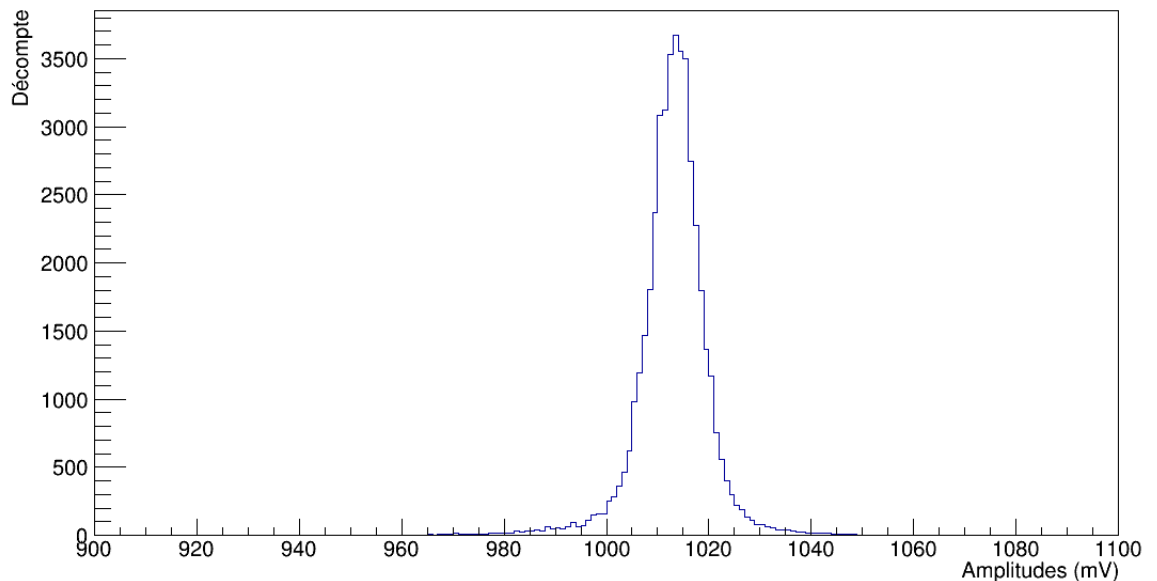


Figure 5.32 : Distribution d'amplitudes des points dans la région du piédestal pour des signaux générés dans un cristal de ICs.

5.2.1 Formes moyennes expérimentales

Pour le ICs, les distributions d'amplitudes et de décalages dans le temps estimés avec un polynôme de puissance 2 sont illustrées sur les figures 5.33 et 5.34. L'amplitude a une valeur moyenne de 600 ± 100 mV. La grande incertitude témoigne du régime de saturation. Quant au décalage, sa distribution n'est pas aussi uniforme que ce à quoi l'on pourrait s'attendre d'une source aléatoire comme la radiation ambiante, avec un pic centré sur la valeur moyenne de $0,1 \pm 0,8$ ns sur l'intervalle de $[-5, 5]$ ns. Ceci est aussi une conséquence du régime de saturation. Les formes de références, à la figure 5.35, ne sont absolument pas fiables, puisque qu'elles s'entrecroisent dans la montée et la descente et leurs sommets s'étalent sur un intervalle plus large que 10 ns. Ceci est dû à la distribution non-uniforme du décalage dans le temps, ainsi qu'aux fluctuations dans la forme de la queue des signaux.

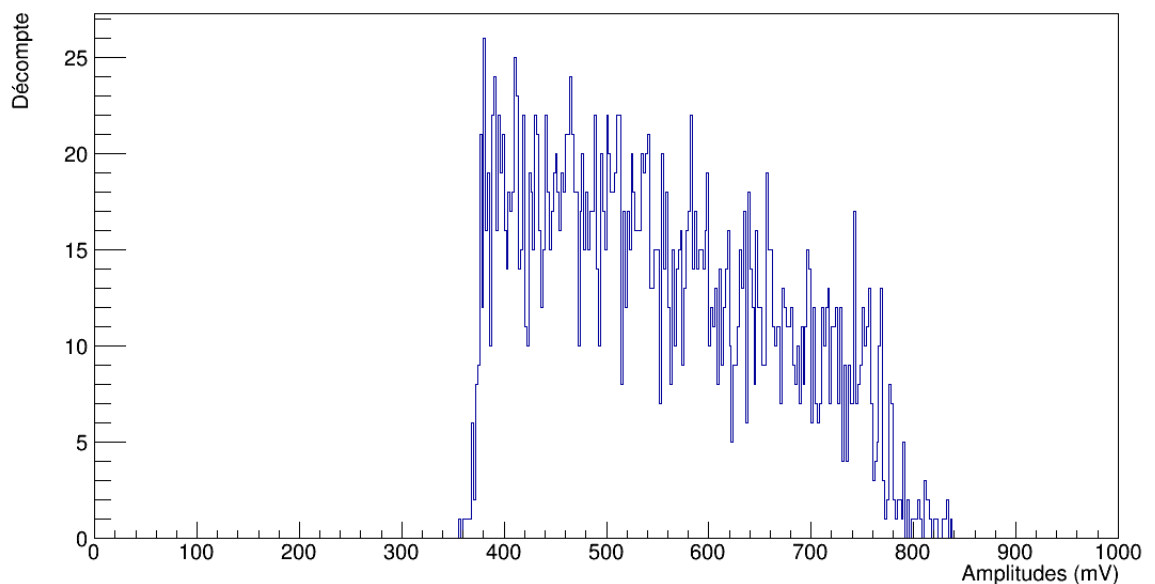


Figure 5.33 : Distribution d'amplitudes estimées par ajustement d'un polynôme de puissance 2 sur des impulsions en provenance d'un cristal de ICs.

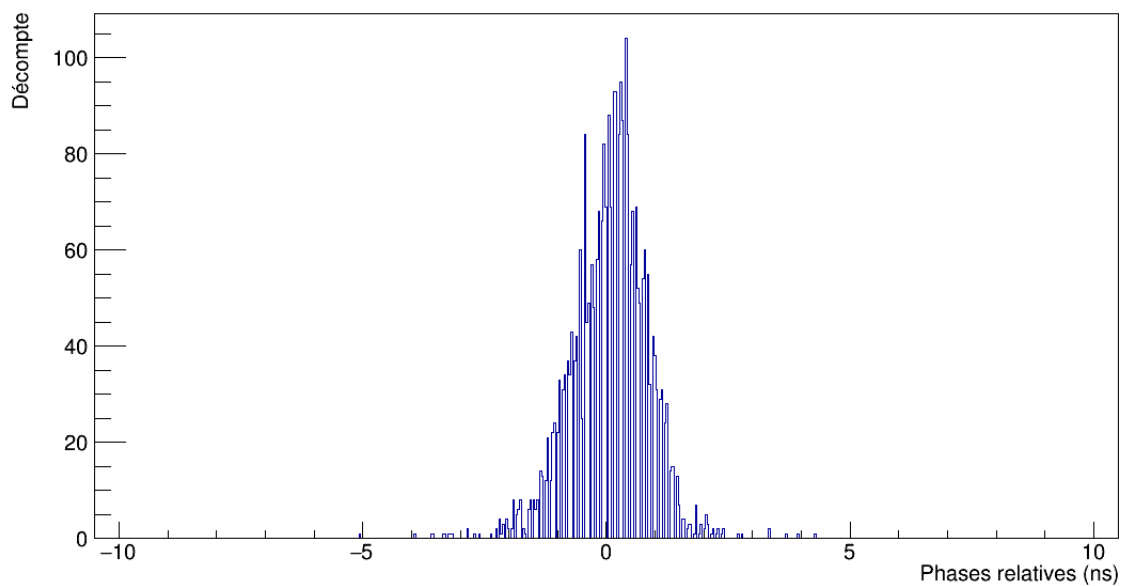


Figure 5.34 : Distribution des décalages dans le temps estimés par ajustement d'un polynôme de puissance 2 sur des impulsions en provenance d'un cristal de ICs.

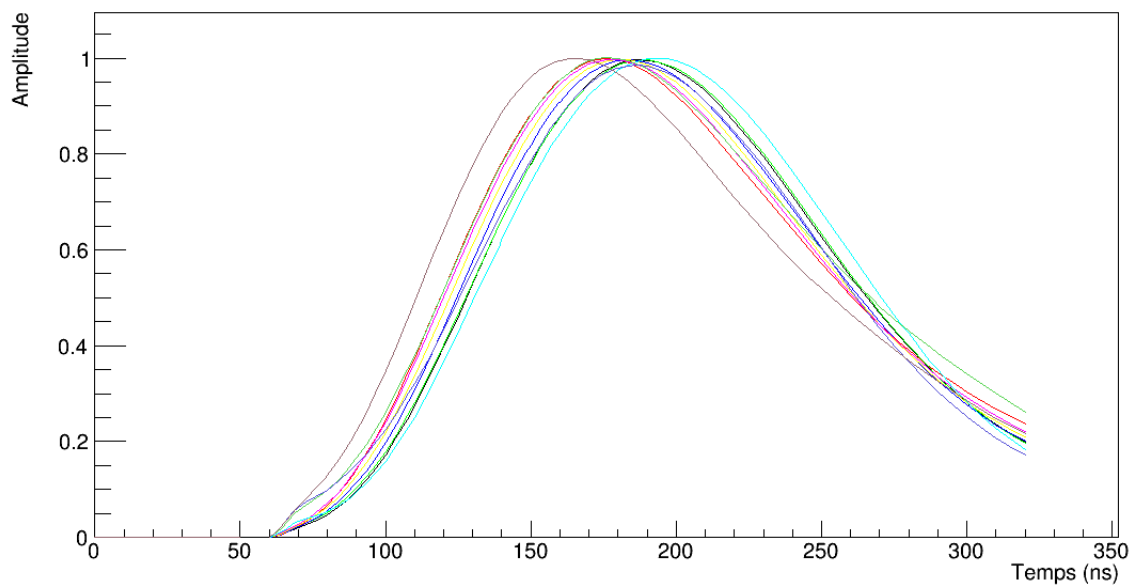


Figure 5.35 : Formes de référence normalisées moyennes expérimentales pour des impulsions en provenance d'un cristal de ICs.

5.2.2 Filtres actifs

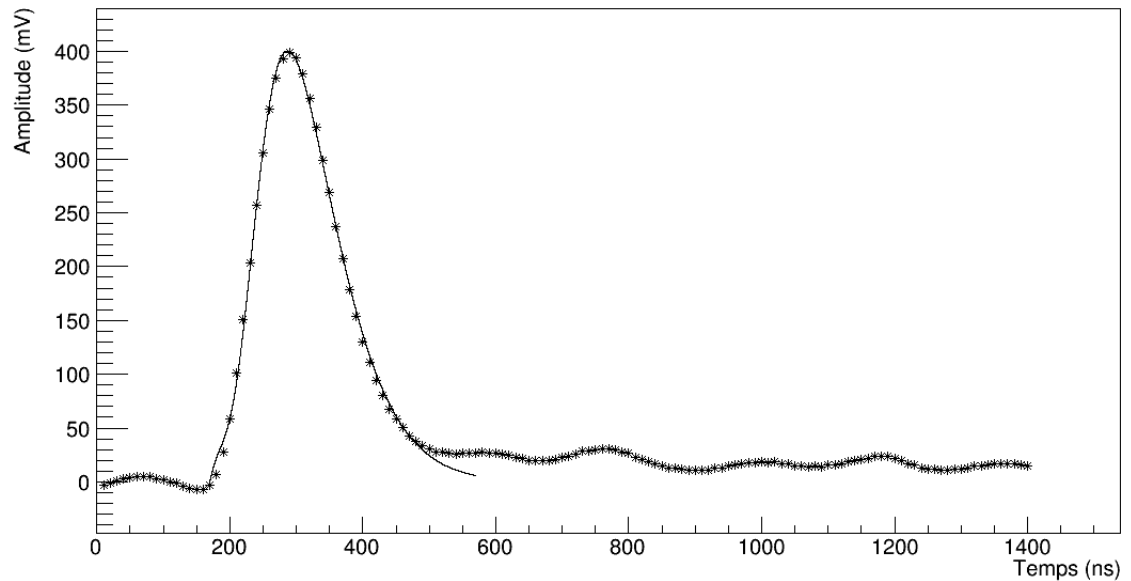


Figure 5.36 : Exemple de l'ajustement de l'équation (5.1) sur un signal généré dans un cristal de ICs, après numérisation par l'ADC.

Sur la figure 5.36, on peut voir un exemple d'ajustement de l'équation (5.1) sur un signal en provenance d'un cristal de ICs. La distribution d'amplitudes est montrée sur la figure 5.37. On voit sur la figure 5.38 la distribution de A_{amp} , dont la valeur moyenne est de 100 ± 50 mV. Pourtant, le pic principal semble plus près de 80 avec une incertitude plus étroite. Ceci est dû à la longue traînée vers de hautes amplitudes sur la distribution après le pic principal, ce qui correspond à un régime de saturation. Quant à A_s , montré sur la figure 5.39, les deux petits pics d'amplitude avec un étalement très large reflètent les fluctuations dans la courbure de la longue queue d'un signal à l'autre. Sur la figure 5.40, le temps de scintillation τ_s a un pic principal plutôt étroit centré autour de 30 ns, soit la valeur prévue pour un cristal de ICs, mais sa distribution s'étale vers des temps plus longs, sans convergence stable apparente dans cette région. Pour ce qui est de $1/\omega_0$ et $1/\omega_d$, illustrés sur les figures 5.42 et 5.43, tous les deux ont un pic principal à environ $(1/43)$ ns avec un étalement vers de plus petites valeurs, tandis que $1/\beta$ s'étale plutôt

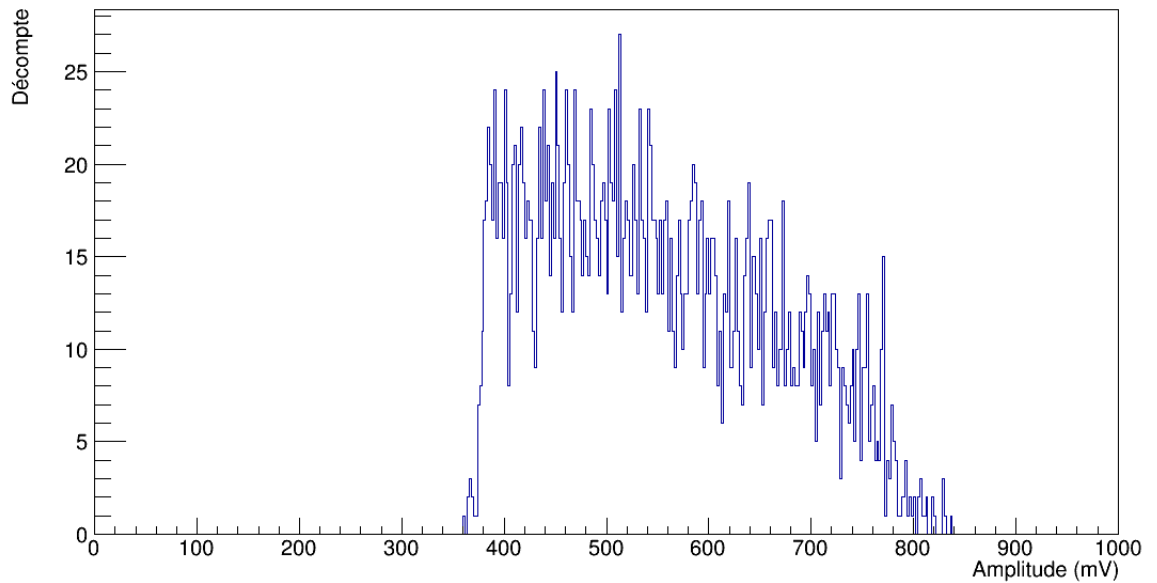


Figure 5.37 : Distribution d’amplitudes maximales pour les ajustements de l’équation (5.1).

vers des valeurs plus élevées, comme on le voit sur la figure 5.41. Le pic principal de ce dernier paramètre est d’environ $(1/44)$ ns. La figure 5.44 montre que le $\widetilde{\chi^2}$ individuel moyen tourne autour de 1 pour certaines impulsions, soit un excellent accord, mais la majorité des ajustements se répartissent de façon plutôt uniforme sur des valeurs plus élevées. Ces résultats s’expliquent par les fluctuations sur la courbure et la longueur de la queue des signaux en provenance du ICs, en plus des problèmes de stabilité dans les distributions des paramètres dûs à la sensibilité de la fonction (5.1) aux variations de ses paramètres, calculées avec la méthode de Levenberg-Marquardt. Pour le temps de montée τ , on obtient 30 ± 4 ns, ce qui est plus petit que le temps de montée mesuré de 71 ± 2 ns pour le ICs à la section 4.4.3. La distribution d’amplitudes maximales est montrée à la figure 5.37 et les formes de référence obtenues avec les valeurs moyennes des paramètres sont illustrées sur la figure 5.45. Aussi, les ajustements auront nécessité environ 70 itérations chacun.

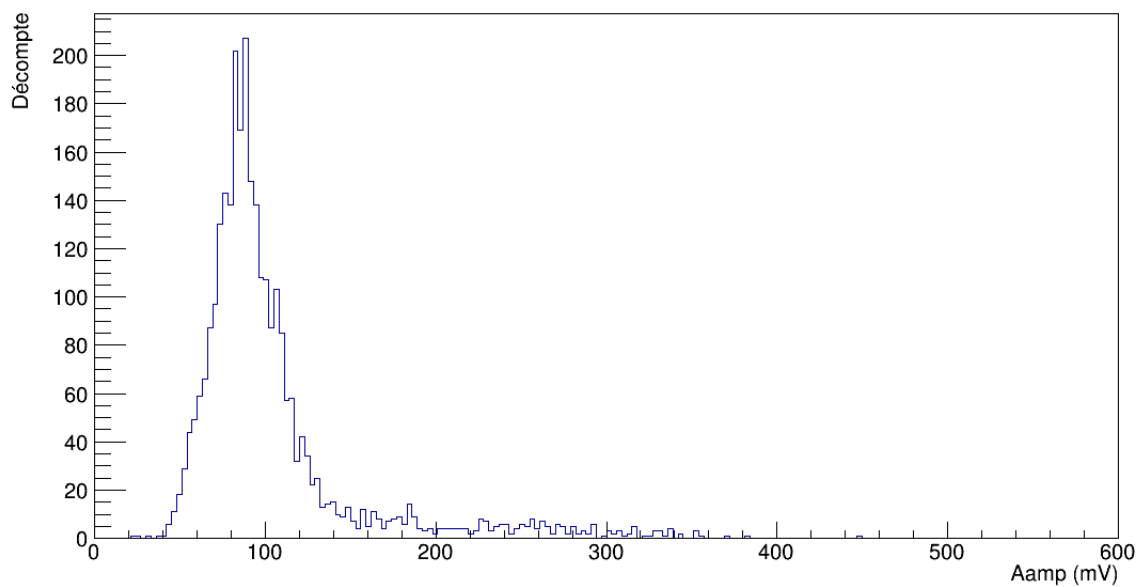


Figure 5.38 : Distribution du paramètre A_{amp} pour des signaux en provenance d'un cristal de ICs.

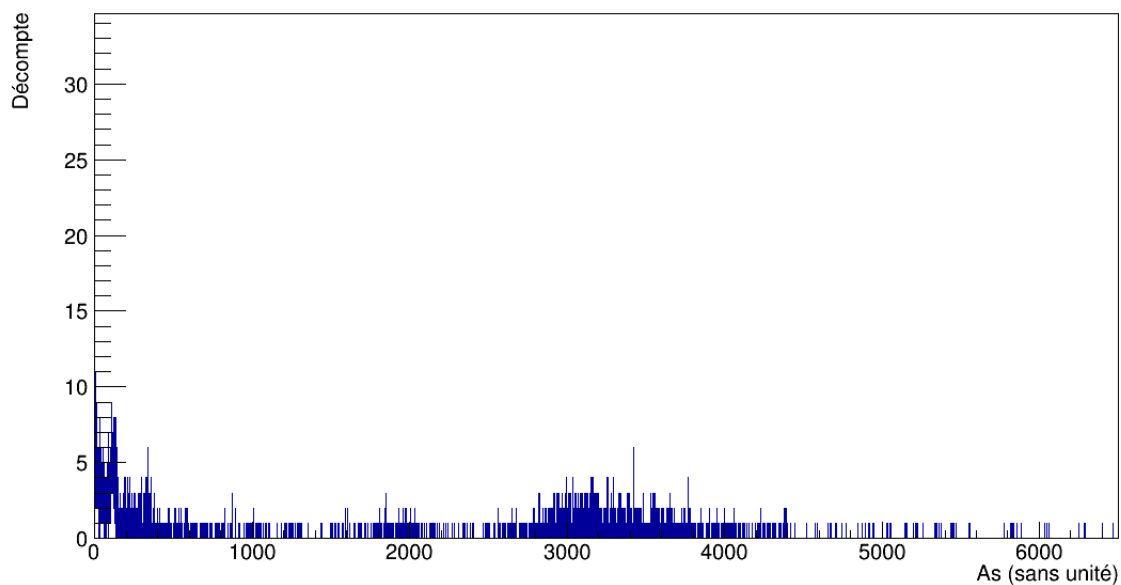


Figure 5.39 : Distribution du paramètre A_s pour des signaux en provenance d'un cristal de ICs.

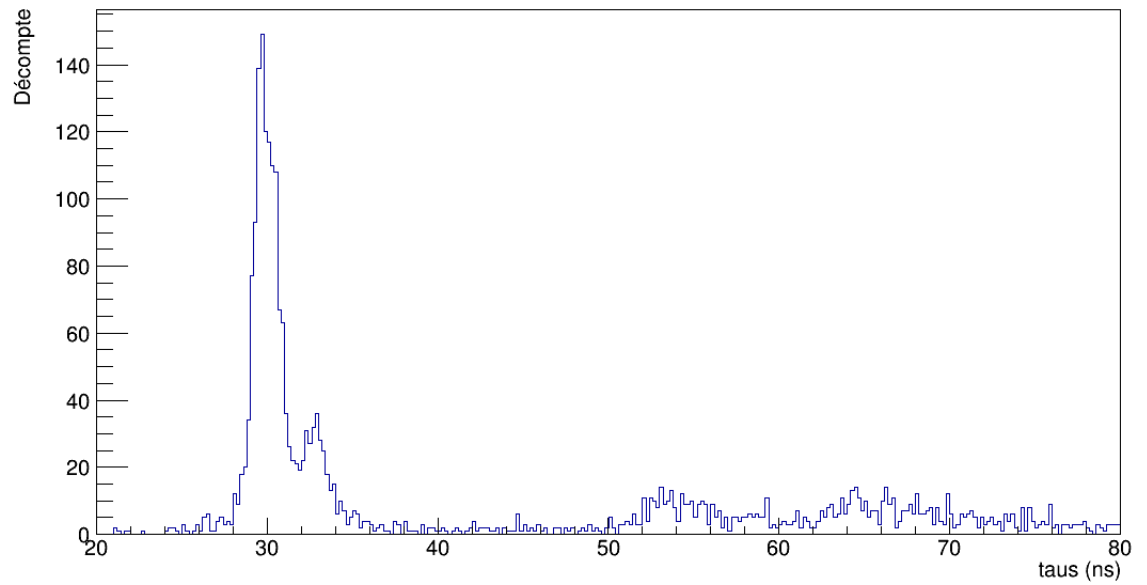


Figure 5.40 : Distribution du paramètre τ_s pour des signaux en provenance d'un cristal de ICs.

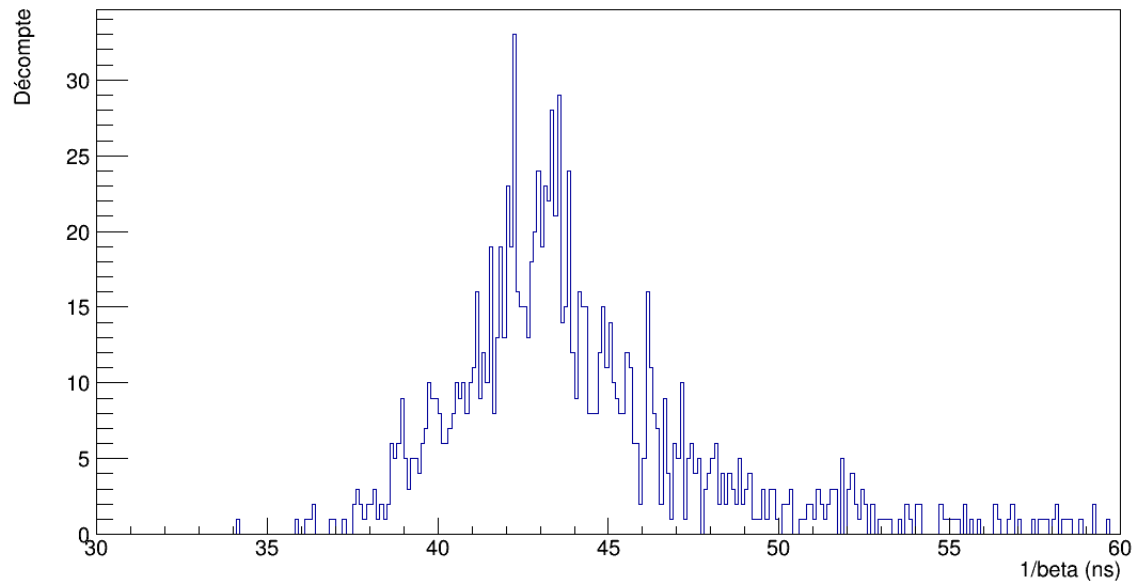


Figure 5.41 : Distribution du paramètre $1/\beta$ pour des signaux en provenance d'un cristal de ICs.

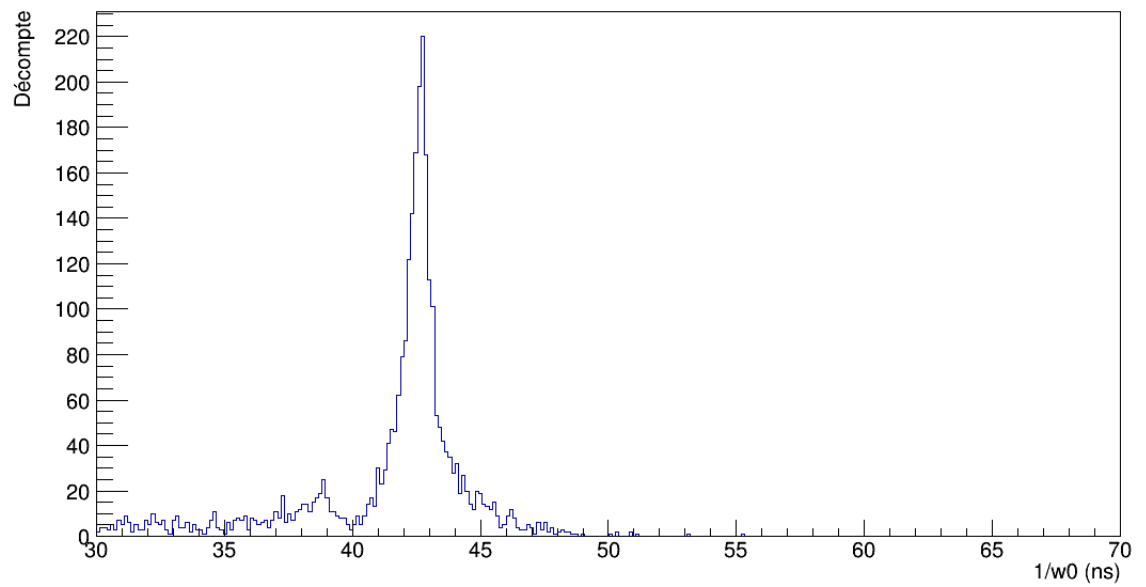


Figure 5.42 : Distribution du paramètre $1/w_0$ pour des signaux en provenance d'un cristal de ICs.

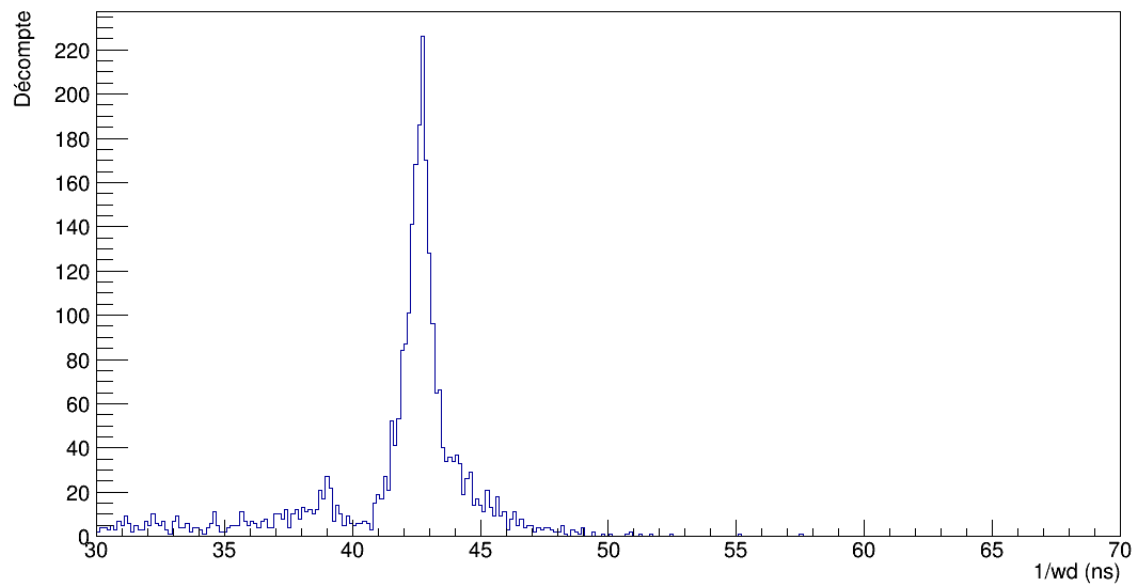


Figure 5.43 : Distribution du paramètre $1/w_d$ pour des signaux en provenance d'un cristal de ICs.

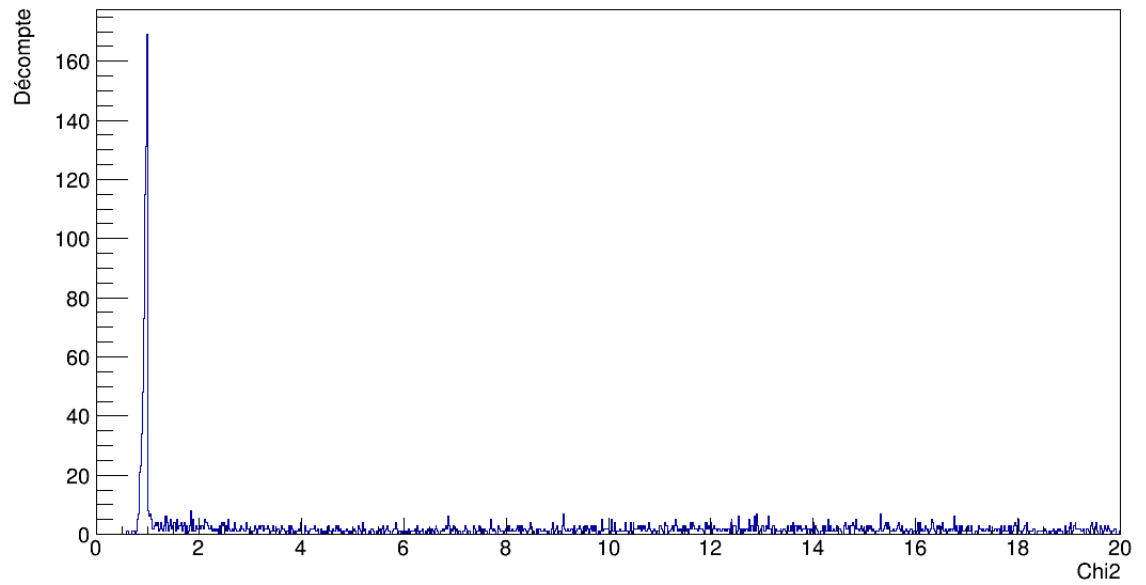


Figure 5.44 : Distribution des $\tilde{\chi}^2$ individuels pour l'ajustement de l'expression (5.1) des filtres actifs sur des signaux en provenance d'un cristal de ICs.

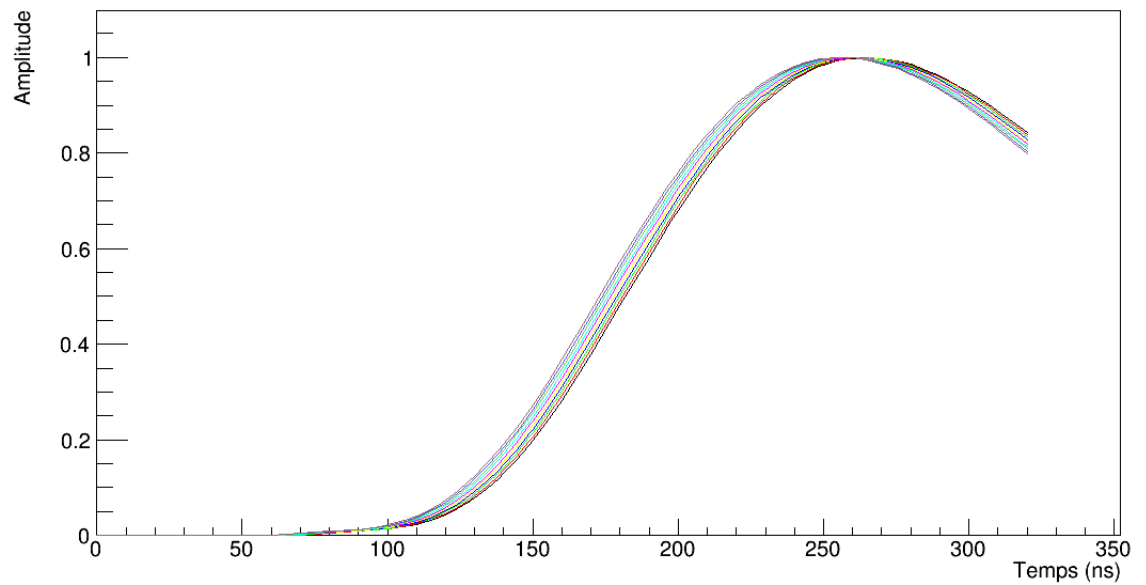


Figure 5.45 : Formes de référence normalisées pour l'expression (5.1) des filtres actifs sur des signaux en provenance d'un cristal de ICs.

5.2.3 Somme de 3 gaussiennes

Un exemple d'ajustement est illustré à la figure 5.46, pour lequel 32 points ont été utilisés à partir du premier au-dessus du seuil de détection de 35 mV pour le ICs.

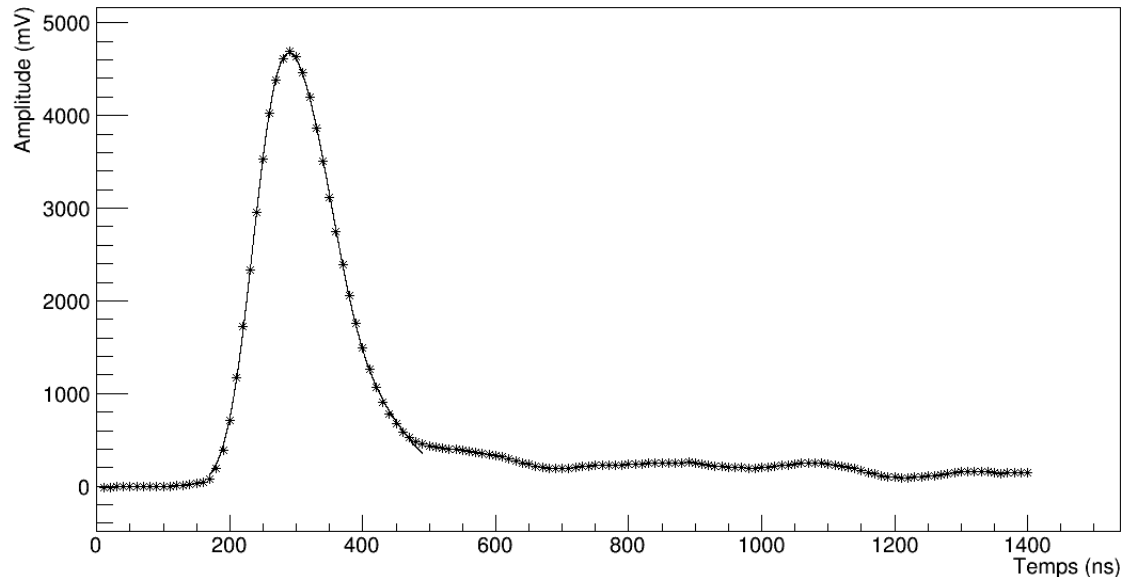


Figure 5.46 : Exemple de l'ajustement d'une somme de 3 gaussiennes sur un signal généré dans un cristal de ICs, après numérisation par l'ADC.

Les valeurs optimales que l'on obtient après normalisation des signaux en provenance du ICs pour les paramètres A_i sont, respectivement, $0,45 \pm 0,04$ (fig. 5.47), $0,69 \pm 0,02$ (fig. 5.48) et $0,20 \pm 0,04$ (fig. 5.49). Pour les μ_i , on a 260 ± 3 ns (fig. 5.50), 311 ± 3 ns (fig. 5.51) et 390 ± 20 ns (fig. 5.52) et pour les σ_i , on a 34 ± 1 ns (fig. 5.53), 46 ± 2 ns (fig. 5.54) et 71 ± 7 ns (fig. 5.55). Les paramètres A_i ont de belles distributions stables. Pour les μ_i , on devrait s'attendre à un étalement sur environ 10 ns en raison du taux d'échantillonnage de l'ADC. Cependant, la distribution de décalages dans le temps n'est pas uniforme, comme le montrait la figure 5.34 à la section 5.2.1. De plus, pour les paramètres μ_i et σ_i , on observe un étalement de plus en plus large avec i qui augmente et les distributions de μ_3 et σ_3 ont 2 pics. Ceci est dû aux fluctuations de la courbure et de la longueur de la queue des signaux, qui peut varier d'un à l'autre. En effet, la première

gaussienne est bien définie, puisqu'elle décrit la montée du signal. Mais il arrive parfois que la queue soit plus courte et abrupte. Alors les deux autres gaussiennes, principalement la troisième, subiront un décalage vers des temps plus petits et auront un étalement σ_i plus étroit. Les fluctuations de la queue des signaux ne sont pas expliquées, mais on suppose que ce peut être une conséquence soit des signaux sortant du préamplificateur, qui n'incluent pas tous la composante lente dans la montée, soit du régime de saturation du *shaper* ou encore des interférences captées par le *shaper*, qui n'était pas blindé. Aussi, le $\widetilde{\chi^2}$ moyen sur les ajustements individuels est de 3 ± 2 , ce qui représente un accord plus ou moins satisfaisant avec cette fonction. La distribution du $\widetilde{\chi^2}$ est illustrée sur la figure 5.56. Le pic centré sur 1 correspond à ces cas où la courbure de la queue est plus douce, comme sur la figure 5.44 pour les ajustements de la fonction pour un filtre actif. Quant à la figure 5.57, on y voit la distribution d'amplitudes obtenues avec la somme de 3 gaussiennes, pour une valeur moyenne de 600 ± 100 , en accord avec les formes moyennes expérimentales. Les formes de référence obtenues sont illustrées sur la figure 5.58. De plus, il aura fallu en moyenne 108 itérations pour chaque ajustement.

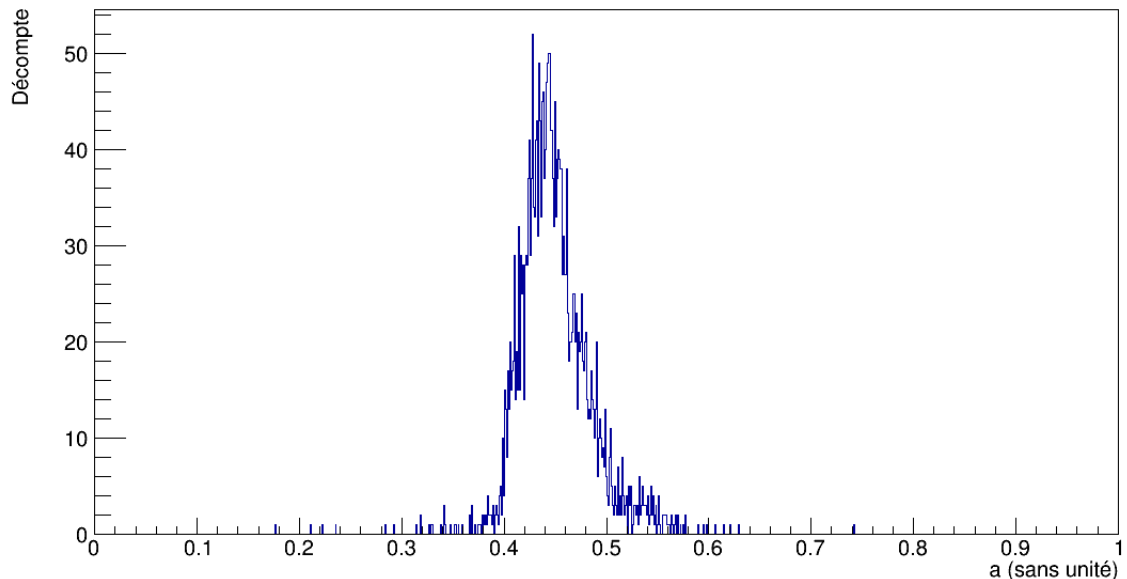
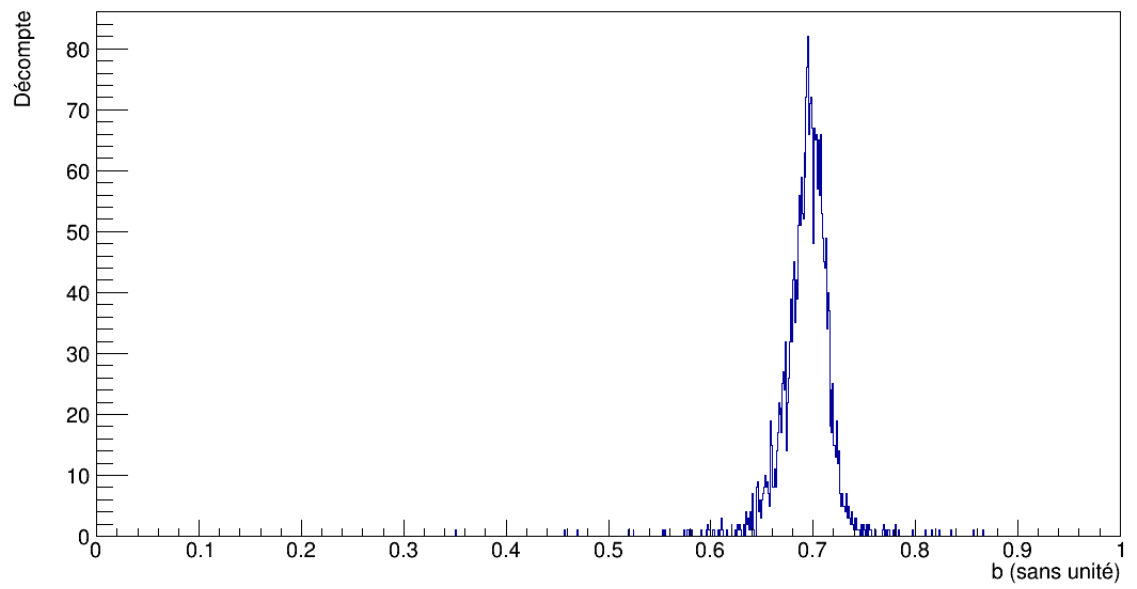
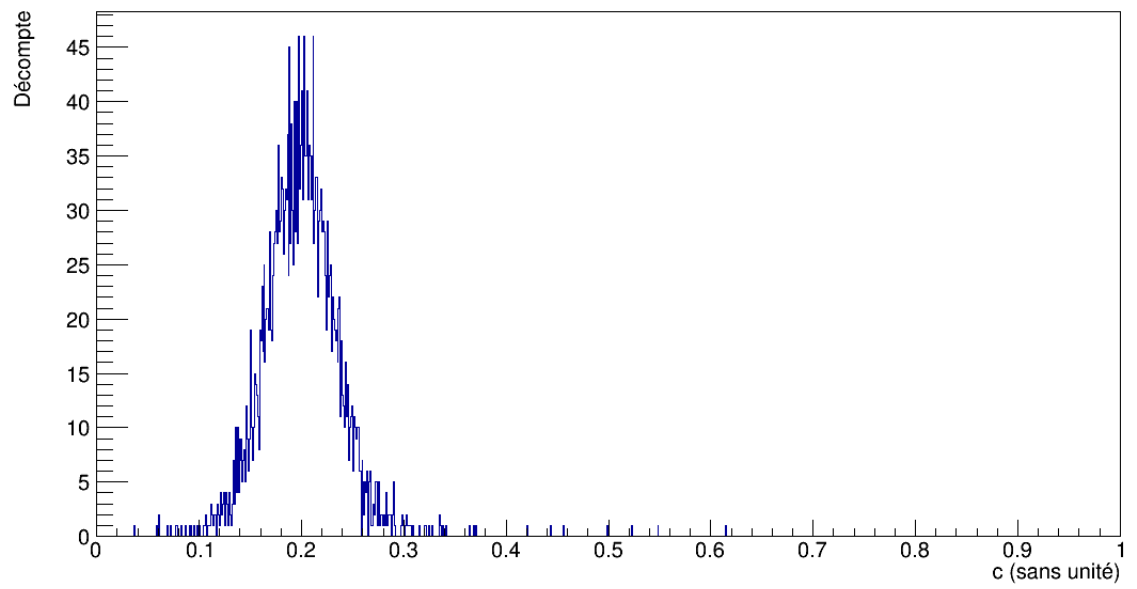
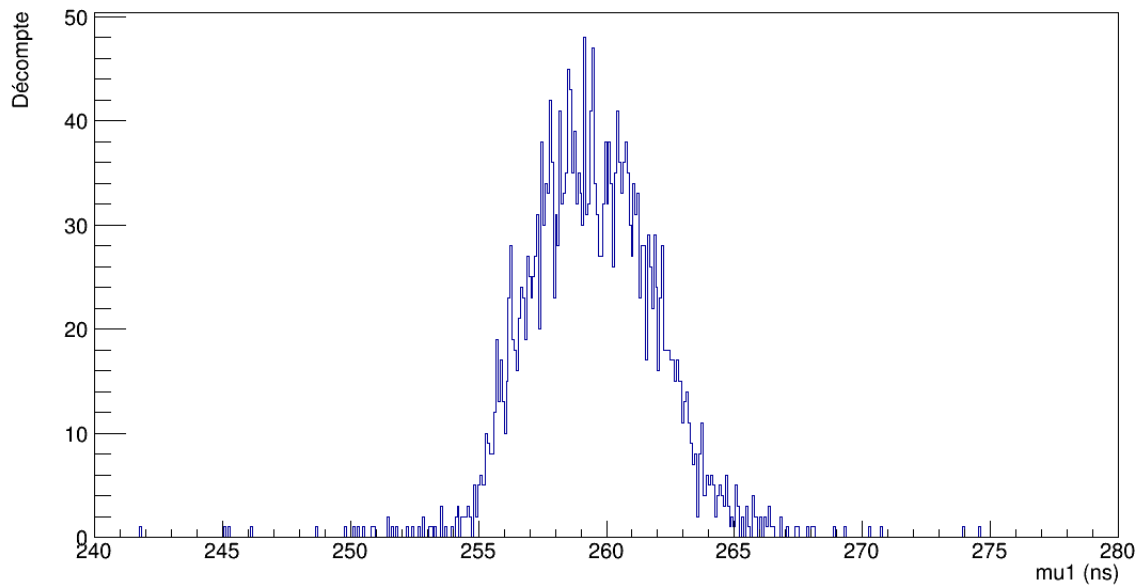
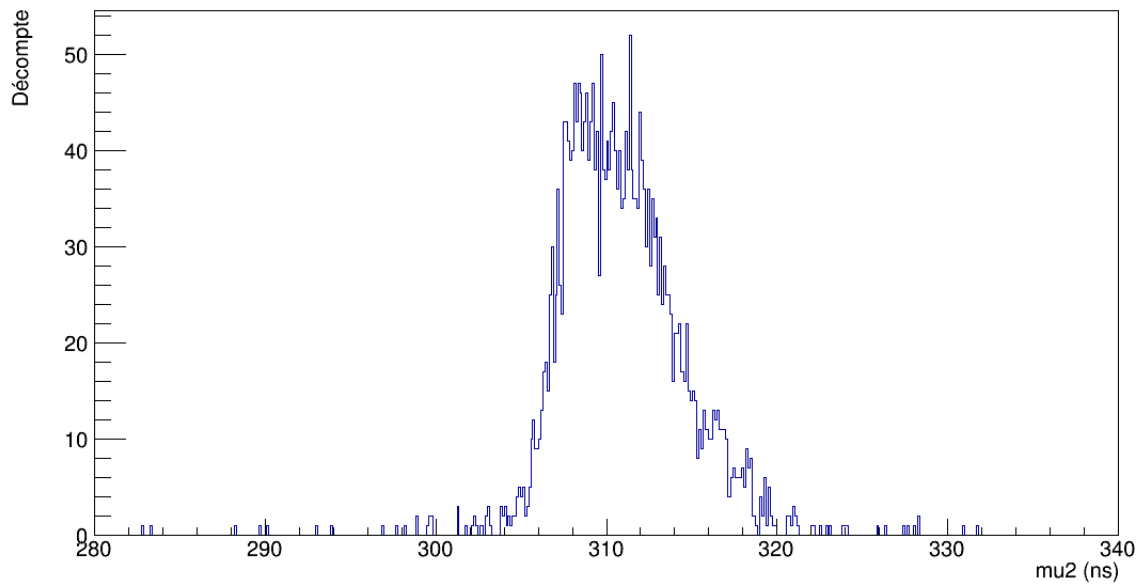
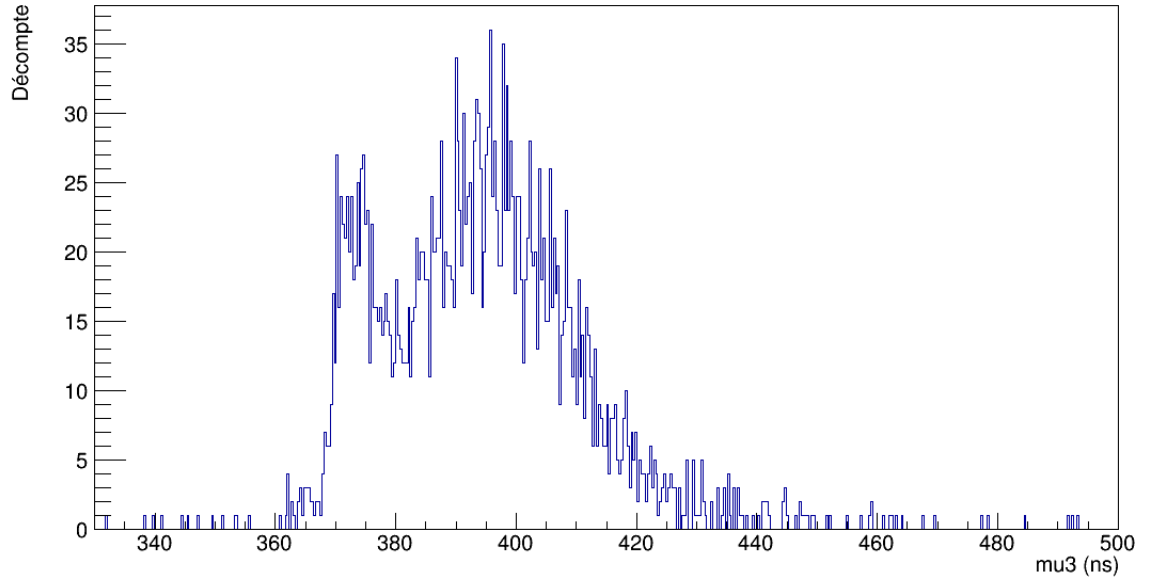
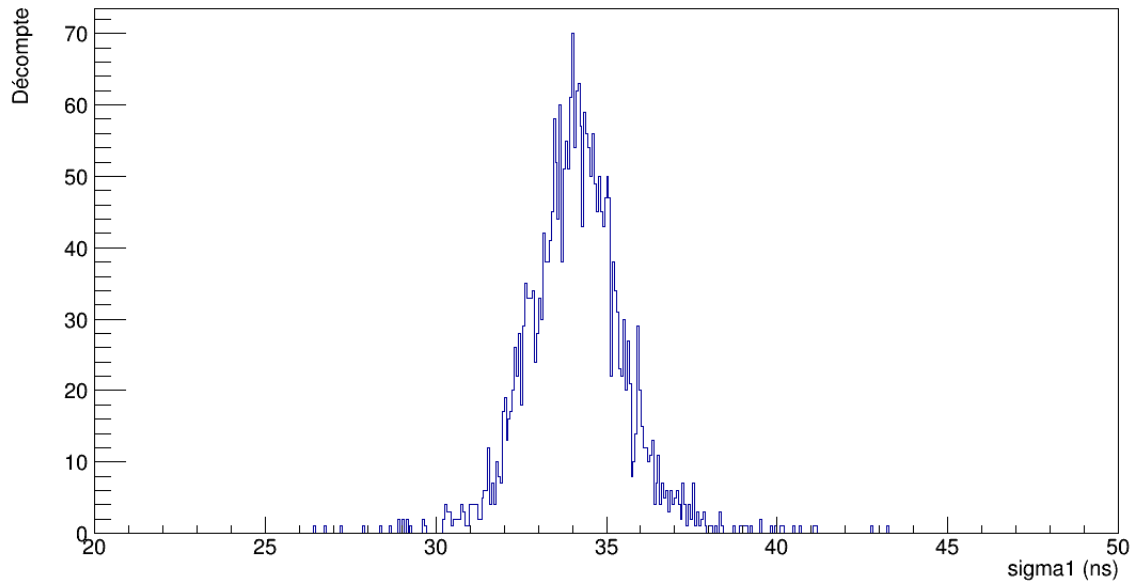
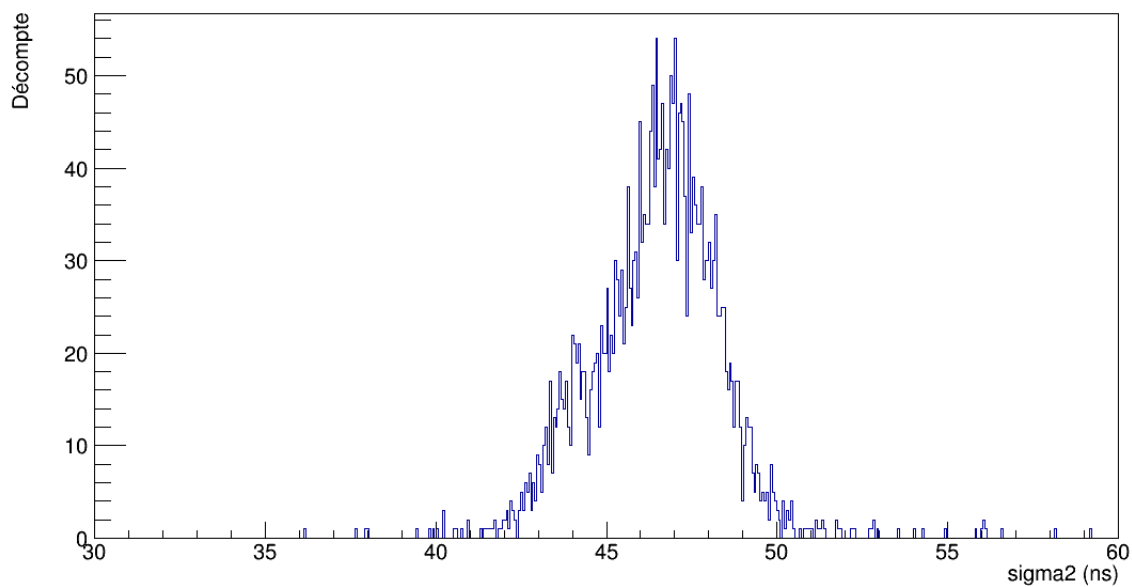
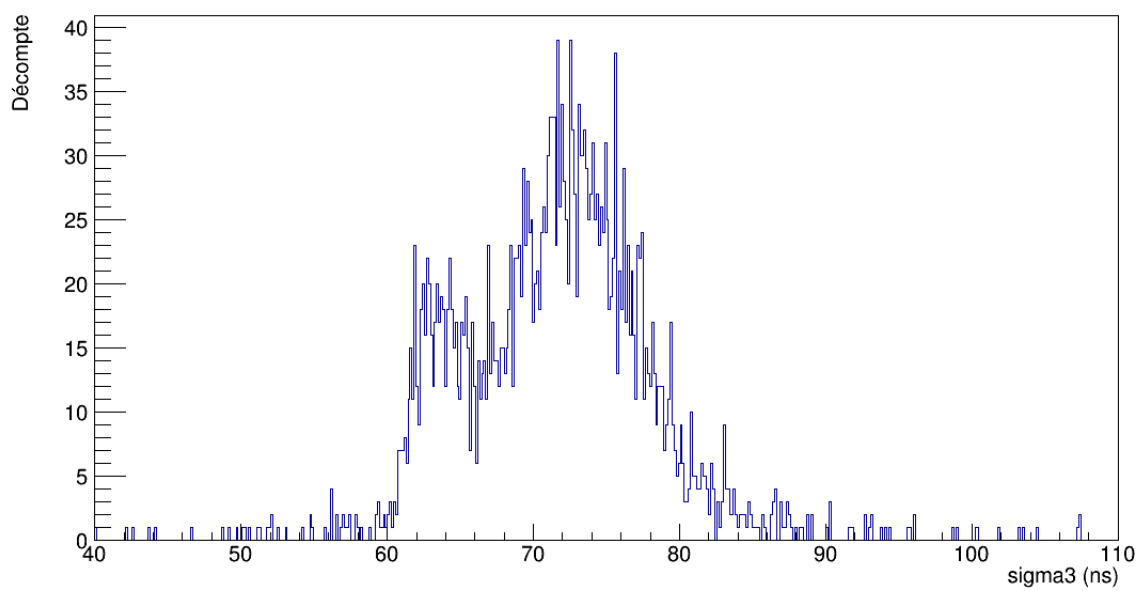


Figure 5.47 : Distribution du paramètre A_1 .

Figure 5.48 : Distribution du paramètre A_2 .Figure 5.49 : Distribution du paramètre A_3 .

Figure 5.50 : Distribution du paramètre μ_1 .Figure 5.51 : Distribution du paramètre μ_2 .

Figure 5.52 : Distribution du paramètre μ_3 .Figure 5.53 : Distribution du paramètre σ_1 .

Figure 5.54 : Distribution du paramètre σ_2 .Figure 5.55 : Distribution du paramètre σ_3 .

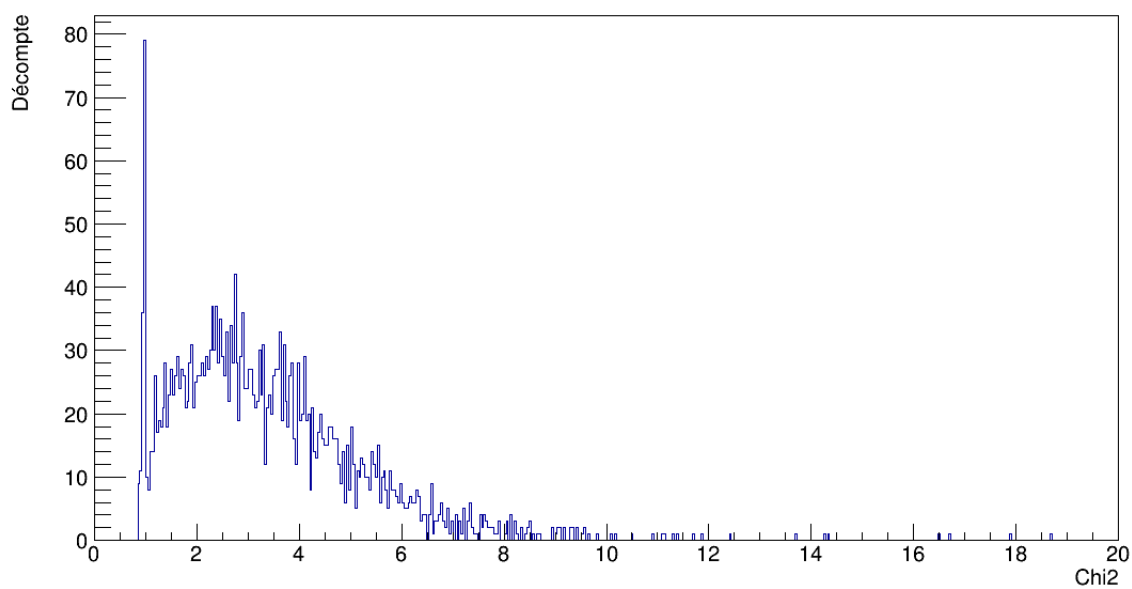


Figure 5.56 : Distribution des $\widetilde{\chi^2}$ individuels pour l'ajustement d'une somme de 3 gaussiennes.

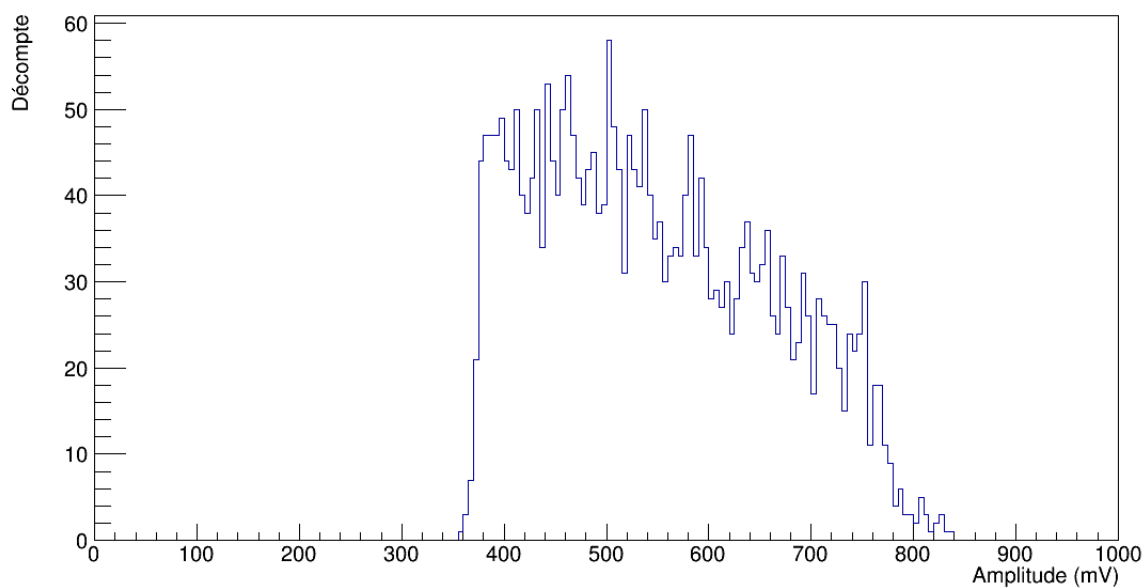


Figure 5.57 : Distribution d'amplitudes maximales pour les ajustements d'une somme de 3 gaussiennes.

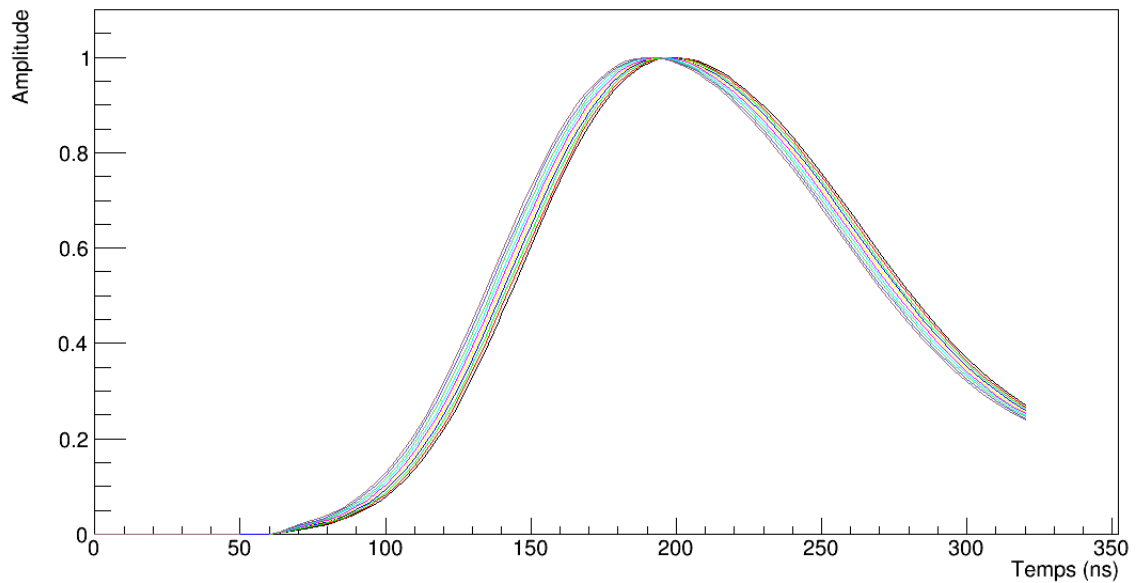


Figure 5.58 : Formes de référence normalisées pour la somme de 3 gaussiennes.

5.2.4 Comparaison

Dans le cas du ICs, on peut s'attendre à ce que les formes de références obtenues avec la somme de 3 gaussiennes soient les plus optimales pour le traitement numérique des signaux par le DSP. En effet, les formes moyennes expérimentales manquent de régularité en raison des fluctuations sur la courbure de la longue queue des signaux, tandis que les ajustements de la fonction pour un filtre actif sur les signaux ne convergent pas vers une solution bien définie pour les différents paramètres, car la fonction a une trop grande sensibilité sur les variations calculées pour ceux-ci à chaque itération. Quant à la somme de 3 gaussiennes, cette sensibilité est beaucoup moins importante, puisqu'on additionne 3 fonctions indépendantes les unes des autres, dont on fait graduellement varier l'amplitude, la position dans le temps et la largeur de chacune. On remarque que les trois méthodes donnent malgré tout des estimations semblables des amplitudes des signaux, comme le montrent les figures 5.33, 5.37 et 5.57, quoique les distributions pour les formes moyennes expérimentales et la somme de 3 gaussiennes se ressemblent davantage. De plus, même si le nombre d'itérations nécessaires pour l'ajustement d'une

somme de 3 gaussiennes est plus grand que dans le cas de l'expression des filtres actifs, les formes de référence obtenues sont beaucoup plus fiables.

5.3 Qualité des formes de référence

Dans cette section, nous vérifions la qualité des formes de référence pour chacune des trois approches étudiées en appliquant la méthode du DSP, qui utilise les matrices de coefficients telles qu'exposées à la section 4.6.1.3 pour extraire les paramètres A , B et P . Nous avons utilisé les mêmes ensembles statistiques que ceux ayant servis pour définir les formes de référence, soient 10000 signaux pour le générateur d'impulsions et 15296 signaux pour le cristal de ICs. Pour chaque signal, enregistré comme une suite de 140 valeurs d'amplitudes, on fait une première évaluation des paramètres A , B et P sur chacune des 109 séquences possibles de 32 points consécutifs, avec l'ensemble de coefficients associé à un décalage dans le temps de 0 ns. Sur la séquence qui donne une valeur d'amplitude A_{max} , si le seuil de 375 mV est dépassé, on applique une deuxième évaluation des paramètres A , B et P , avec un second ensemble de coefficients associé au décalage dans le temps $t_{0_1} = -B/A$ déterminé à la première évaluation. Pour chaque approche étudiée, on peut vérifier les distributions d'amplitudes A , de piédestaux P et de décalages dans le temps $t_{0_2} = -B/A$. Pour ce dernier, on s'attend à ce que le décalage soit plus petit que 1 ns à la deuxième évaluation. Il est aussi possible de comparer les amplitudes obtenues à chacune des deux évaluations en prenant leur ratio. De plus, chaque signal ayant une forme de référence optimale combinée aux paramètres A , B et P optimaux, on peut vérifier la qualité des formes de référence de chaque méthode étudiée en faisant directement le calcul du $\widetilde{\chi^2}$ à partir de l'équation (4.21).

5.3.1 Générateur d'impulsions

5.3.1.1 Formes moyennes expérimentales

Un exemple d'ajustement sur un signal de l'équation (4.17b) avec la forme de référence optimale et les paramètres A , B et P associés est présenté sur la figure 5.59, avec un très bon accord. Le piédestal évalué avec les formes moyennes expérimentales

est de $493,9 \pm 0,4$ mV et l'amplitude moyenne est de 425 ± 2 mV, avec le générateur d'impulsions. L'ADC étant bipolaire, ce qui lui permet de considérer des amplitudes positives et négatives, le code pour 0 V correspond au milieu de la gamme dynamique de l'ADC¹, ce qui explique le piédestal élevé par rapport à l'amplitude. Leurs distributions sont montrées sur les figures 5.60 et 5.61. Sur la figure 5.62 on voit la distribution du ratio de l'amplitude A_1 à la première évaluation sur l'amplitude A_2 à la deuxième et dont la valeur moyenne est de $1,003 \pm 0,001$. Aussi, le décalage dans le temps t_{0_2} à la deuxième évaluation se trouve dans un intervalle de $[-0,5; 0,5]$ ns, comme on peut voir sur la figure 5.63, soit l'incertitude à laquelle on peut s'attendre pour une précision au nanoseconde près. L'ajustement de ces formes de référence donne un $\widetilde{\chi^2}$ moyen de $1,5 \pm 0,9$, ce qui signifie un accord assez bon. La distribution est sur la figure 5.64.

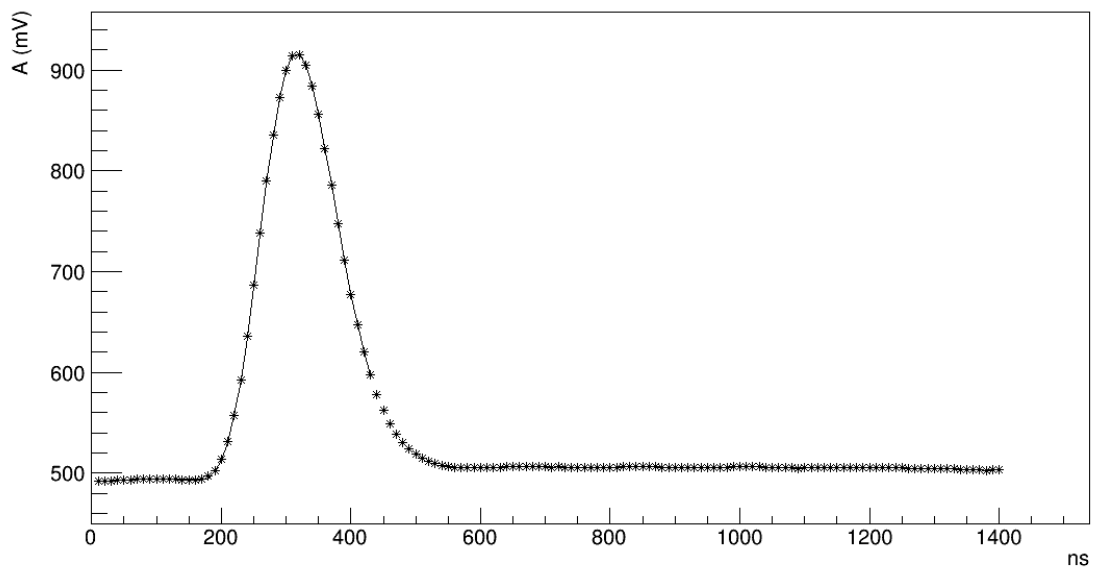


Figure 5.59 : Exemple d'ajustement de l'équation (4.17b) sur un signal produit par un générateur d'impulsions, avec la forme moyenne expérimentale optimale et les paramètres A , B et P correspondants.

¹Mode *Offset Binary*

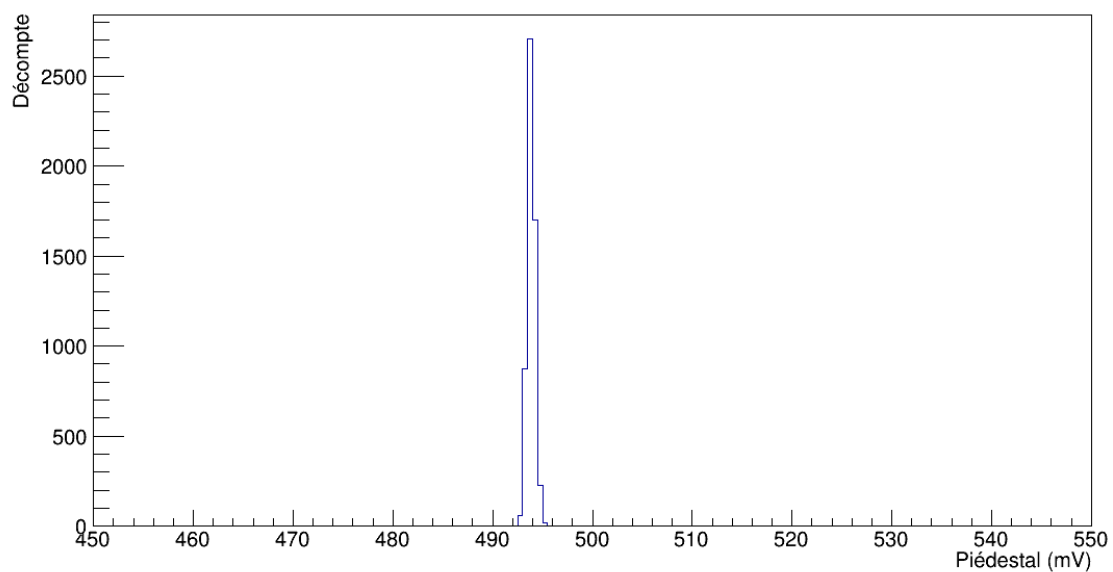


Figure 5.60 : Distribution du piédestal, évalué avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.

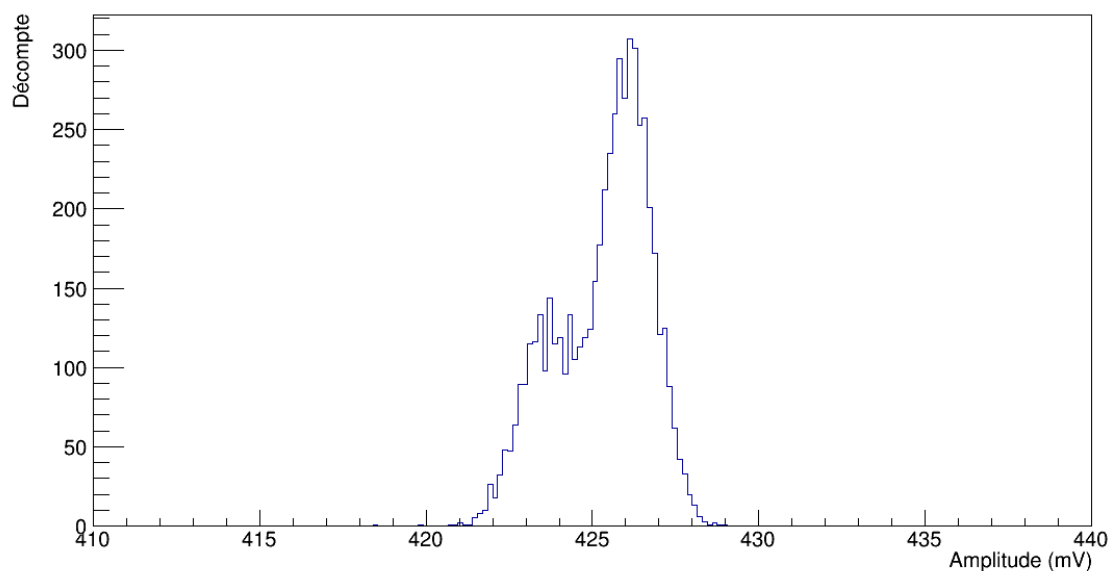


Figure 5.61 : Distribution d'amplitudes évaluées avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.

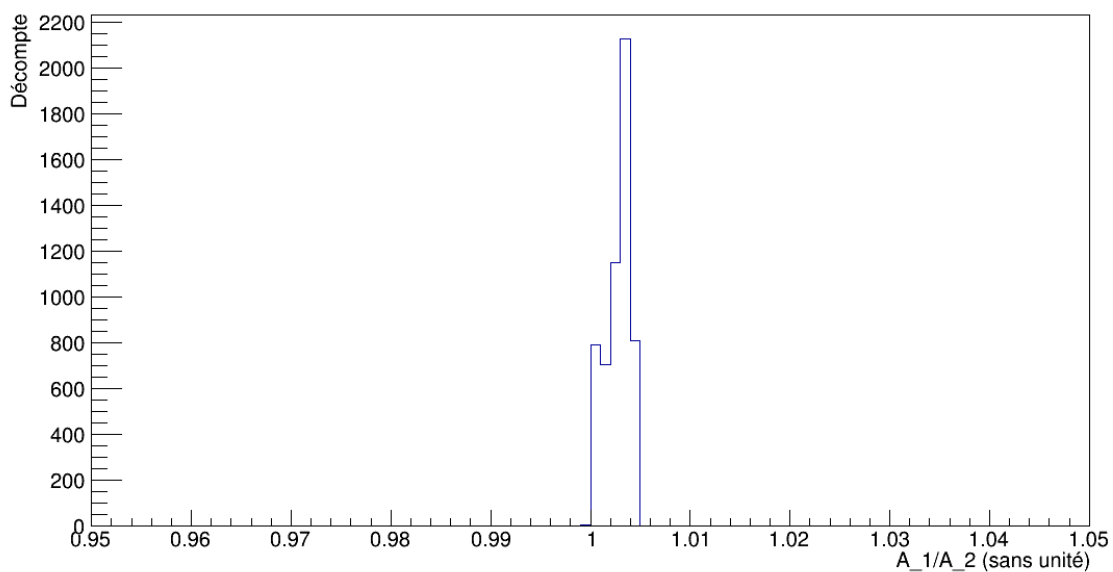


Figure 5.62 : Distribution du ratio d'amplitudes aux première et deuxième évaluations, évalué avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.

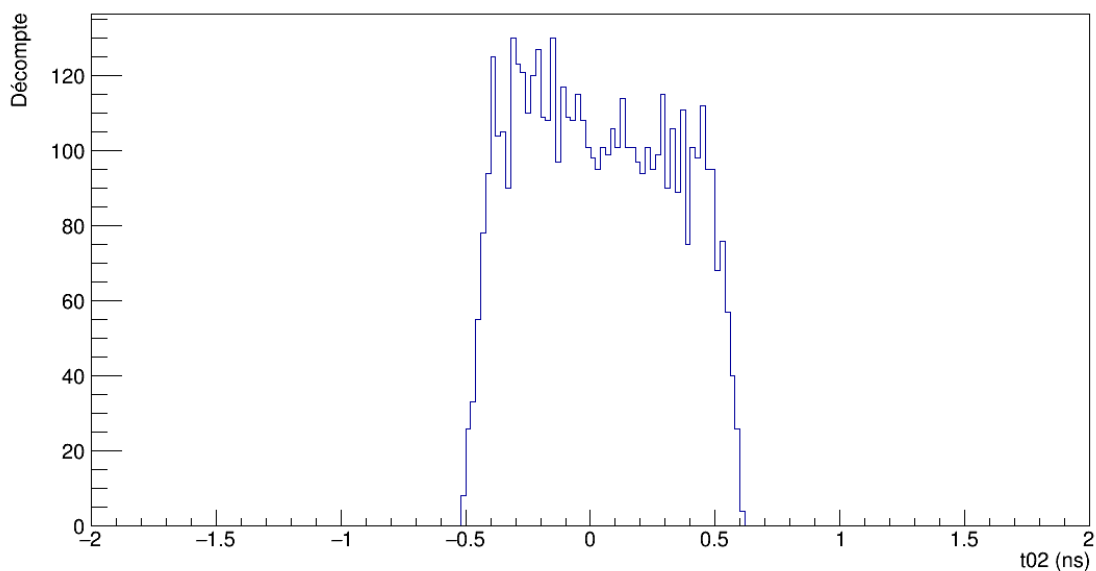


Figure 5.63 : Distribution de t_{0_2} , évalué avec les coefficients des formes moyennes expérimentales pour des signaux produits par un générateur d'impulsions.

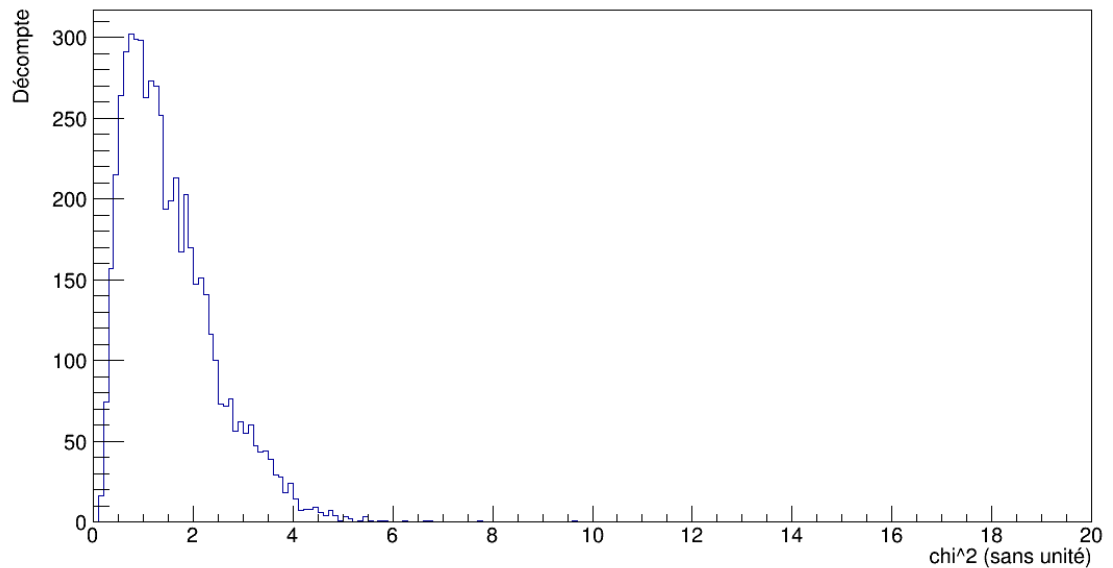


Figure 5.64 : Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme moyenne expérimentale optimale pour chacun des signaux produits par un générateur d'impulsions.

5.3.1.2 Filtres actifs

La figure 5.65 montre un exemple d'ajustement de l'équation (4.17b) avec la forme de référence optimale et les paramètres A , B et P associés et l'accord n'est pas vraiment satisfaisant. Dans le cas de l'expression (5.1) pour les filtres actifs, on obtient un piédestal moyen de $486,8 \pm 0,5$ mV, avec une distribution illustrée sur la figure 5.66. L'amplitude, dont la distribution est sur la figure 5.67, est estimée à 423 ± 2 mV, tandis que le ratio A_1/A_2 est de $1,00 \pm 0,01$. La figure 5.68 montre la distribution de ce ratio. On voit sur la figure 5.69 que la distribution du décalage dans le temps t_{0_2} n'est pas complètement inclus dans l'intervalle de $[-0,5; 0,5]$ ns, mais demeure à l'intérieur de ± 1 ns. Quant au $\widetilde{\chi^2}$, sa valeur moyenne est de $8,0 \pm 0,9$ et donc l'accord n'est pas satisfaisant. Sa distribution est montrée sur la figure 5.70,

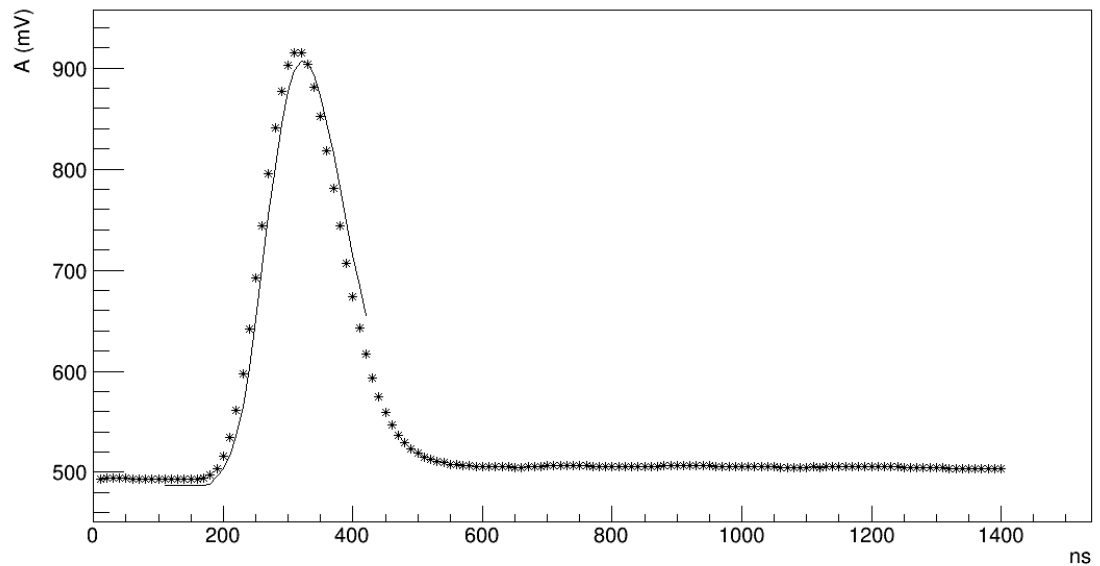


Figure 5.65 : Exemple d'ajustement de l'équation (4.17b) sur un signal produit par un générateur d'impulsions, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir de l'expression (5.1) des filtres actifs.

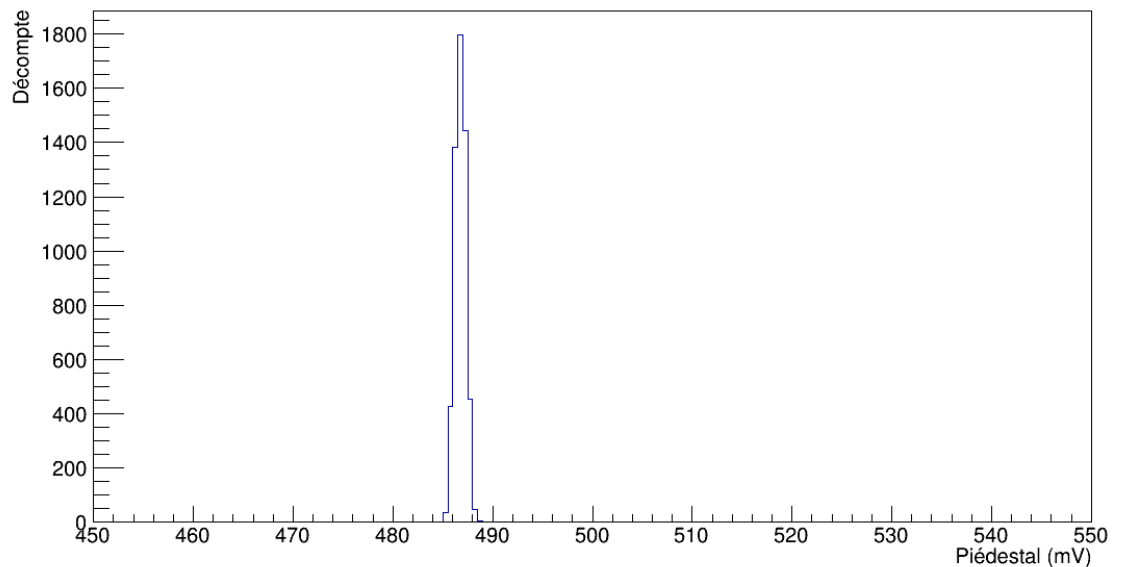


Figure 5.66 : Distribution du piédestal, évalué avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.

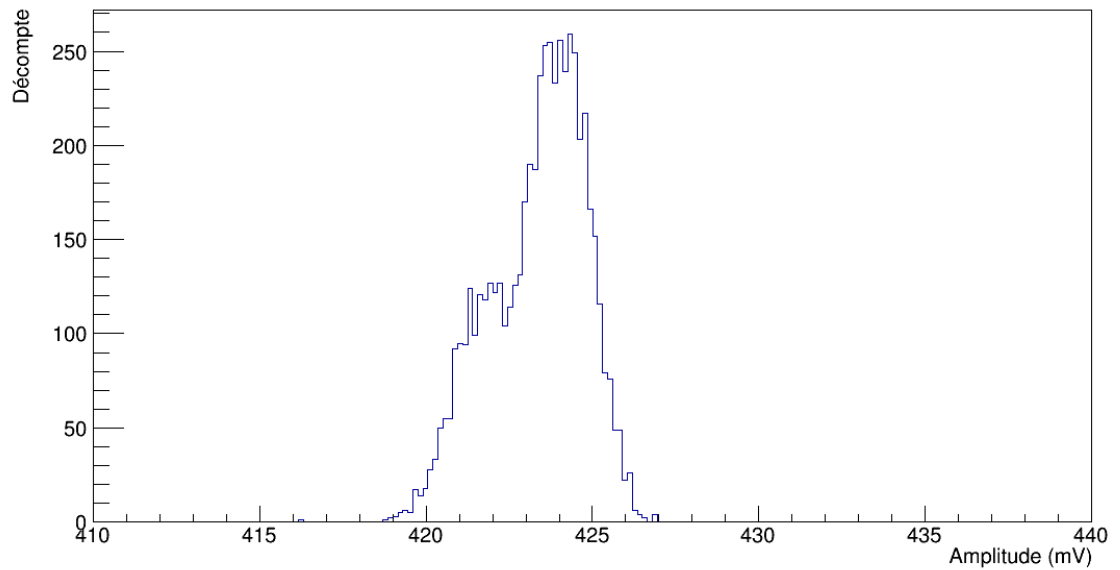


Figure 5.67 : Distribution d'amplitudes évaluées avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.

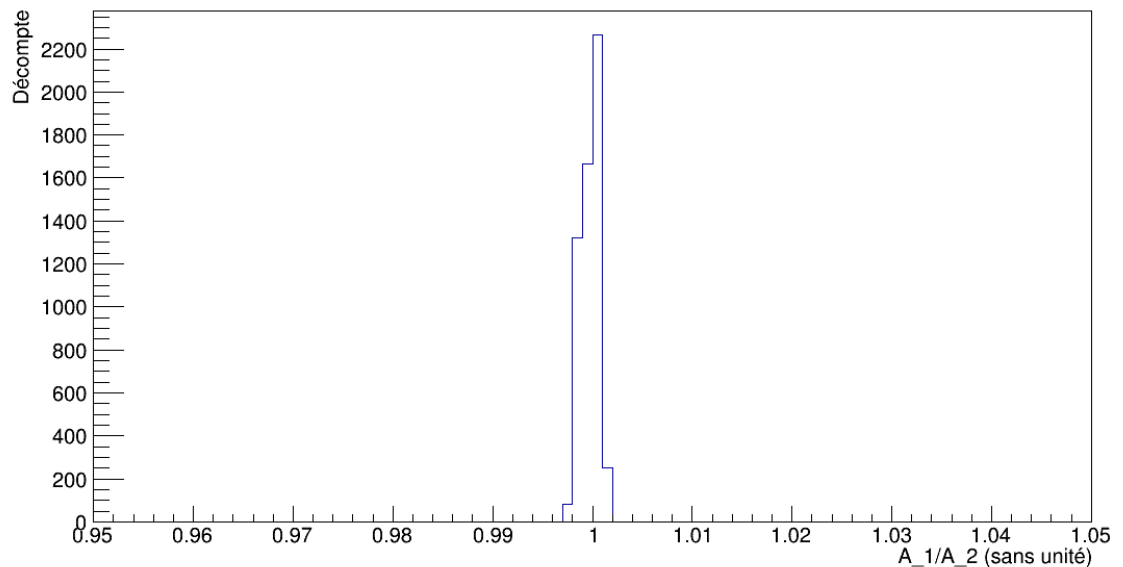


Figure 5.68 : Distribution du ratio d'amplitudes aux première et deuxième évaluations, évalué avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.

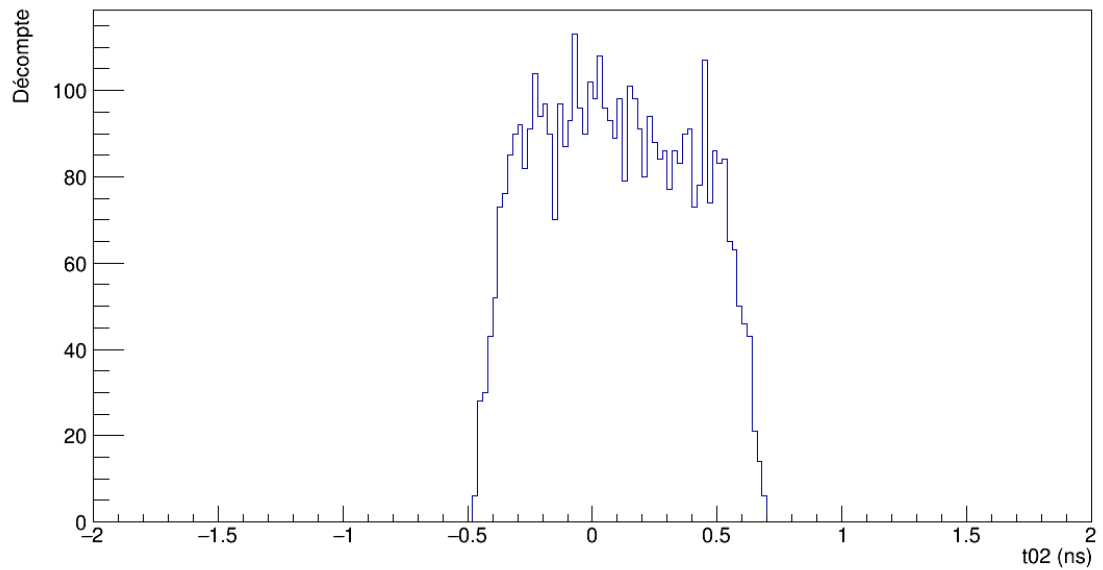


Figure 5.69 : Distribution de t_{02} , évalué avec les coefficients des formes de références pour l'expression (5.1) des filtres actifs pour des signaux produits par un générateur d'impulsions.

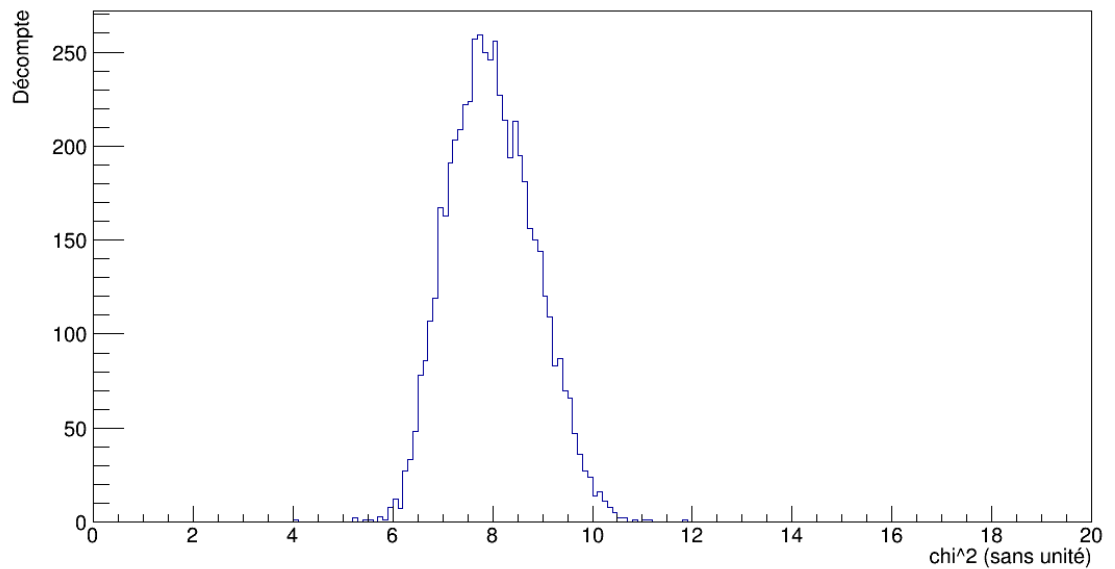


Figure 5.70 : Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de l'expression (5.1) des filtres actifs pour chacun des signaux produits par un générateur d'impulsions.

5.3.1.3 Somme de 3 gaussiennes

Avec les formes de référence correspondant à une somme de 3 gaussiennes sur les impulsions du générateur, on peut voir un exemple d'ajustement sur un signal avec la forme de référence optimale et les paramètres A , B et P associés sur la figure 5.71, avec un très bon accord. La distribution du piédestal est montrée sur la figure 5.72, avec une valeur moyenne de $493,1 \pm 0,4$ mV. Quant à l'amplitude, dont la distribution est illustrée sur la figure 5.73, elle est évaluée à 425 ± 2 mV. De plus, le ratio A_1/A_2 est en moyenne de $1,003 \pm 0,001$. La distribution de ce ratio est sur la figure 5.74. Le décalage dans le temps t_{0_2} est bien dans un intervalle de $[-0,5; 0,5]$ ns, comme on peut voir sur la figure 5.75. Le $\widetilde{\chi^2}$ moyen est de $1,5 \pm 0,4$ et la distribution est sur la figure 5.76. Ainsi, ces formes de référence donnent de bons ajustements sur les signaux du générateur.

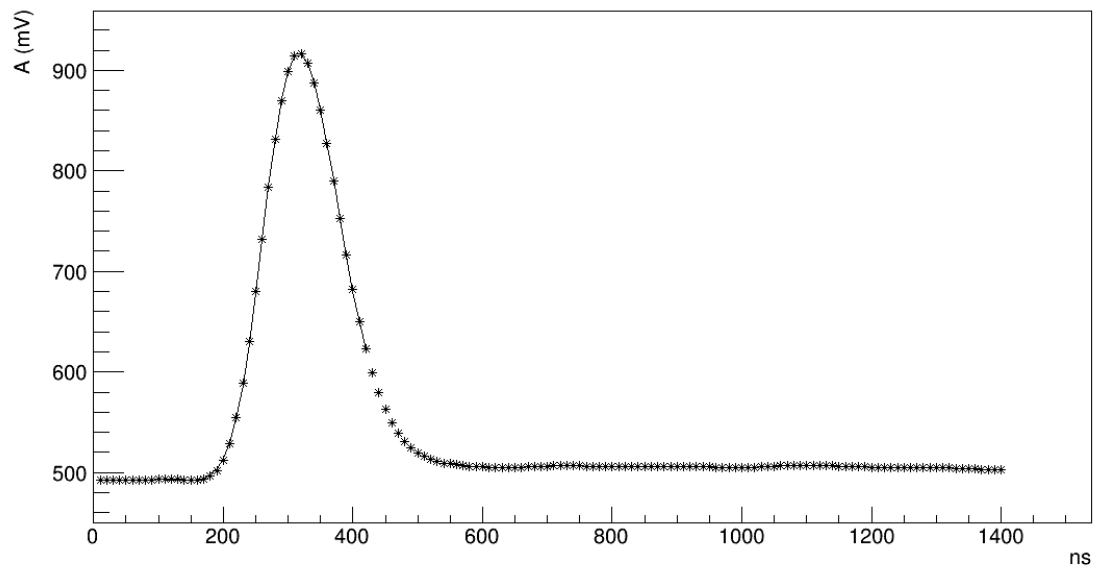


Figure 5.71 : Exemple d'ajustement de l'équation (4.17b) sur un signal produit par un générateur d'impulsions, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir d'une somme de 3 gaussiennes.

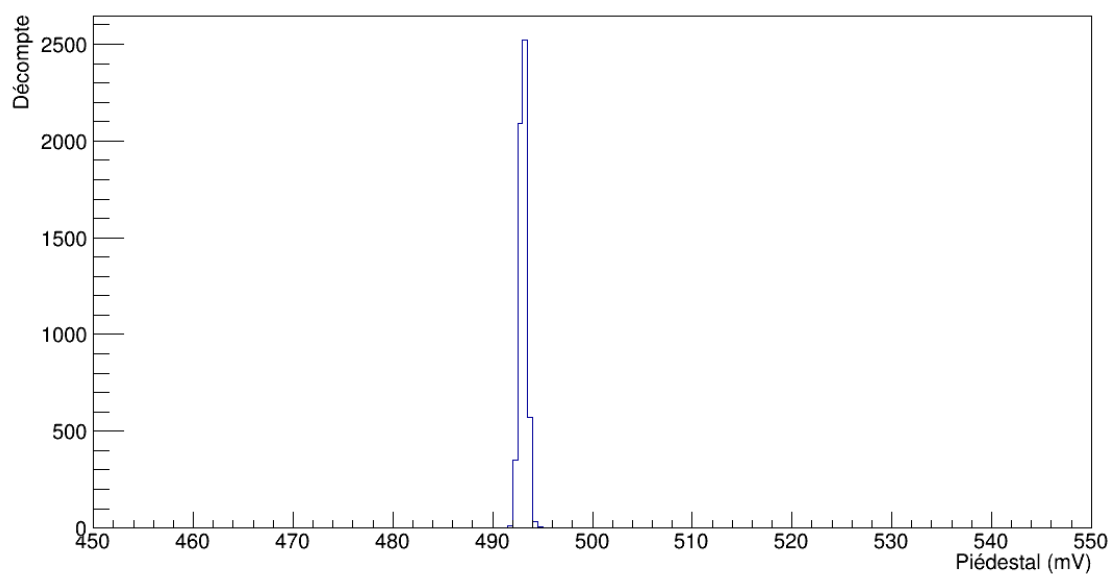


Figure 5.72 : Distribution du piédestal, évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

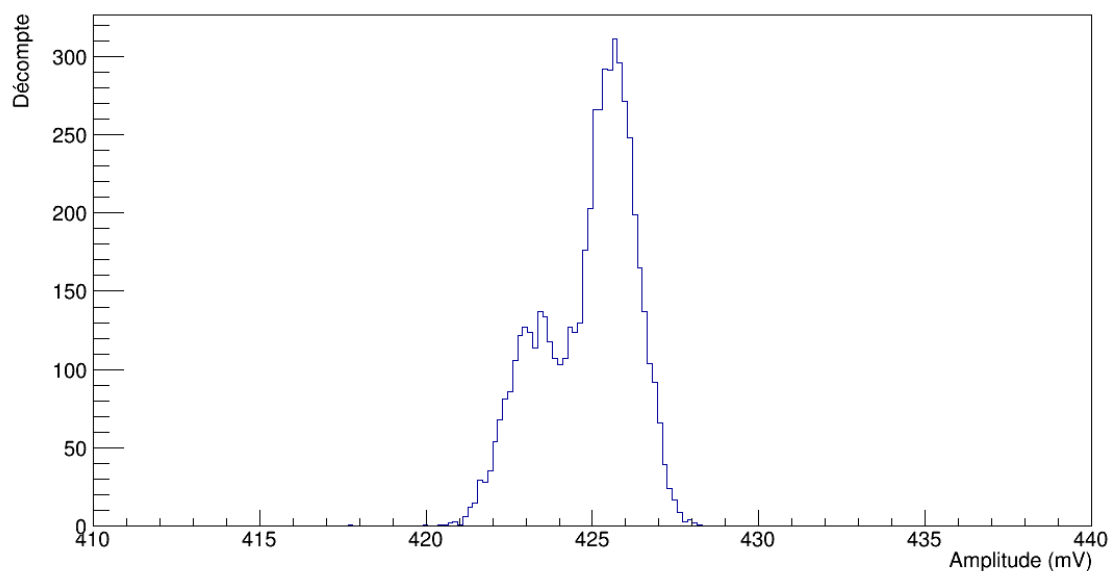


Figure 5.73 : Distribution d'amplitudes évaluées avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

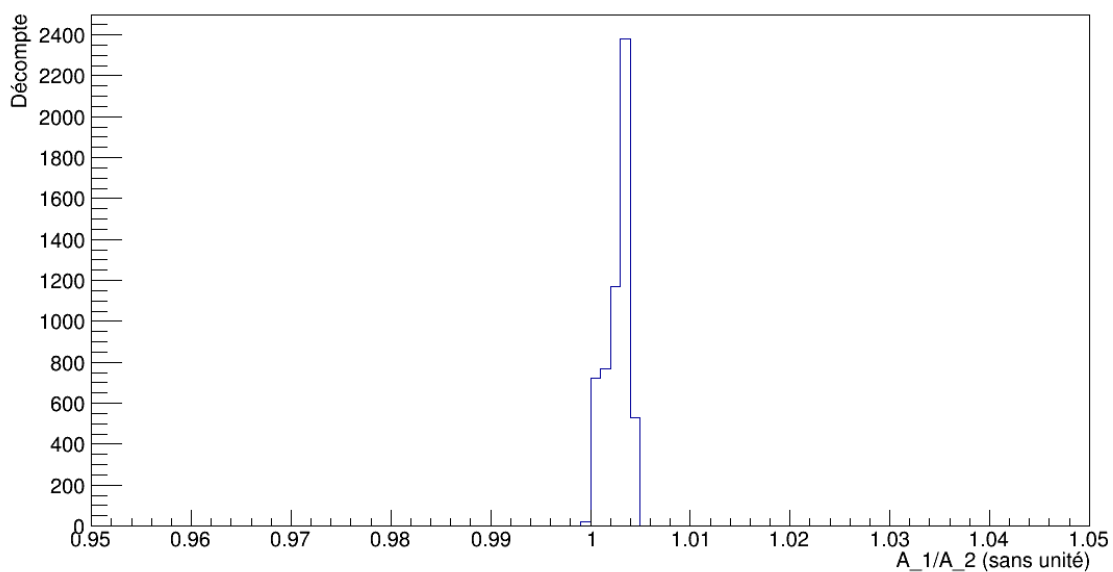


Figure 5.74 : Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenue avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

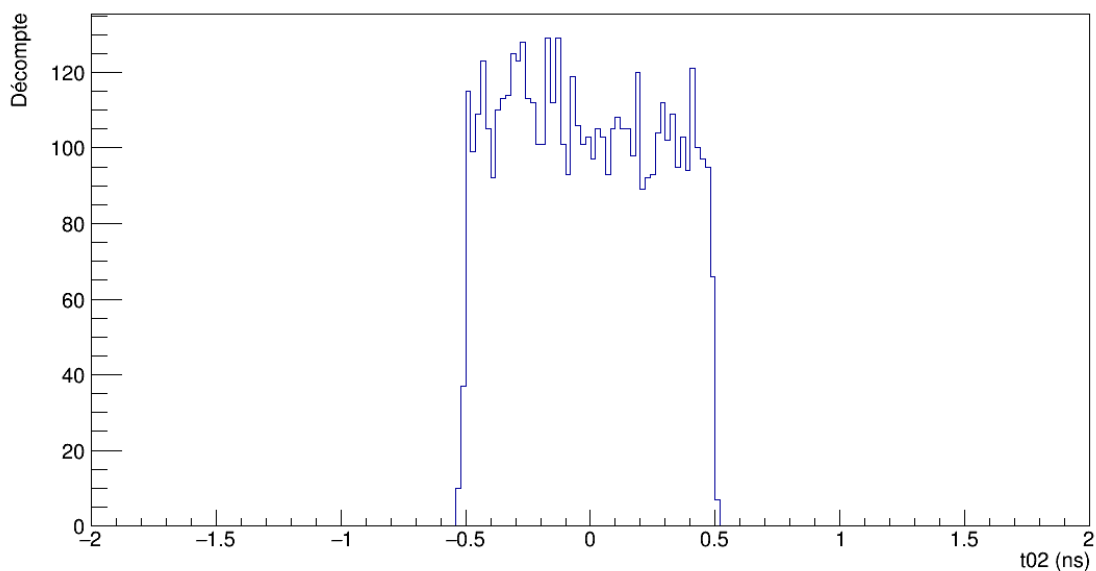


Figure 5.75 : Distribution de t_{02} , évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux produits par un générateur d'impulsions.

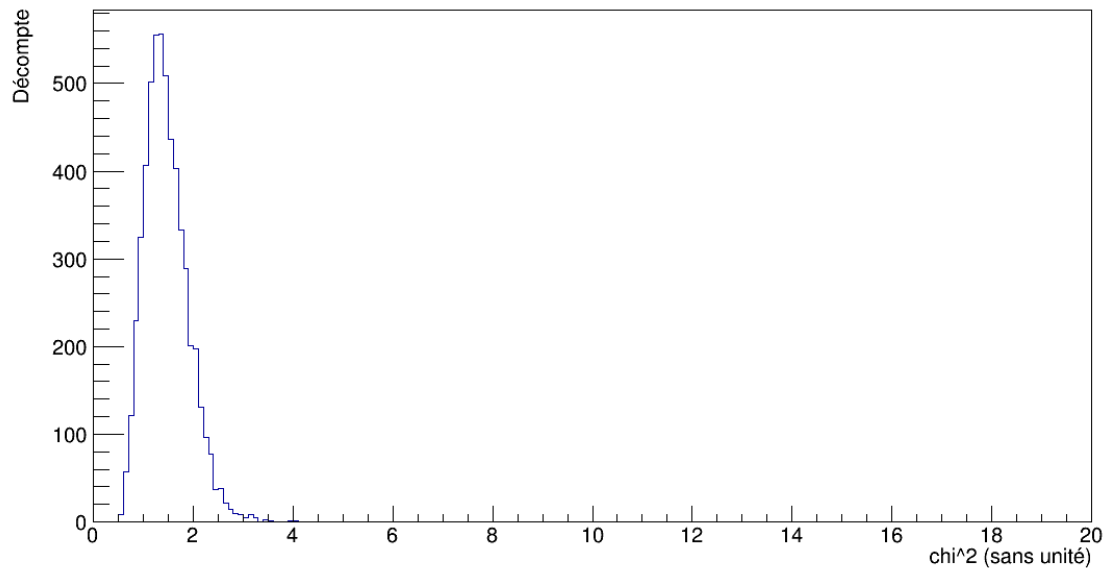


Figure 5.76 : Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de la somme de 3 gaussiennes pour chacun des signaux produits par un générateur d'impulsions.

5.3.1.4 Comparaison

Les coefficients pour les formes moyennes expérimentales et la somme de 3 gaussiennes sont en accord sur tous les paramètres. Pour les deux, l'amplitude est évaluée à 425 ± 2 mV et le ratio A_1/A_2 vaut $1,003 \pm 0,001$, impliquant que l'amplitude est légèrement surestimée à la première évaluation par le DSP. Le $\widetilde{\chi^2}$ est aussi le même pour les deux, respectivement $1,5 \pm 0,9$ et $1,5 \pm 0,4$, quoique l'incertitude est plus petite pour les 3 gaussiennes. Les formes moyennes expérimentales estiment le piédestal à $493,9 \pm 0,4$ mV, tandis que les 3 gaussiennes donnent $493,1 \pm 0,4$ mV. Ceci est en très bon accord avec le piédestal mesuré de $493,4 \pm 0,6$ mV (section 5.1). Quant aux résultats pour l'expression (5.1) du filtre actif, on obtient un piédestal et une amplitude un peu plus faibles de $486,8 \pm 0,5$ mV et 423 ± 2 mV, respectivement. Le ratio A_1/A_2 de $1,00 \pm 0,01$ signifie qu'avec ces coefficients, l'amplitude est bien estimée dès la première évaluation. Pour ce qui est du $\widetilde{\chi^2}$ de $8,0 \pm 0,9$, l'accord n'est tout simplement pas acceptable, contrairement aux deux autres ensembles de coefficients. Malgré tout, l'amplitude semble assez bien

estimée avec les trois méthodes quand on compare avec les résultats de la section 5.1. Quant à t_{0_2} , ce paramètre se trouve à l'intérieur d'un intervalle de $[-1; 1]$ ns pour les trois méthodes, généralement dans un intervalle de $[-0,5; 0,5]$ ns. Ceci implique une bonne estimation du décalage dans le temps dès la première évaluation. Cependant, seuls les décalages dans le temps pour les coefficients pour la somme de 3 gaussiennes ne sortent pas de cet intervalle plus restreint.

Ainsi, comme à la section 5.1.4, les coefficients pour l'expression (5.1) du filtre actif donnent des résultats insatisfaisants avec les signaux du générateur d'impulsions, tandis que ceux pour la somme de 3 gaussiennes sont les plus fiables, même si les formes moyennes expérimentales peuvent aussi donner des résultats acceptables.

5.3.2 Cristal de ICs

5.3.2.1 Formes moyennes expérimentales

Pour les signaux en provenance du ICs, on montre un exemple d'ajustement sur un signal avec la forme moyenne expérimentale optimale et les paramètres A , B et P associés sur la figure 5.77 et l'accord n'est pas satisfaisant. Avec ces formes de référence, le piédestal est évalué autour de 1000 ± 20 mV, tandis que l'amplitude est d'environ 600 ± 200 mV. Leurs distributions sont montrées sur les figures 5.78 et 5.79. Le piédestal est moins stable qu'avec le générateur d'impulsions en raison de fluctuations beaucoup plus importantes dues au bruit. La distribution pour le décalage dans le temps t_{0_2} , sur la figure 5.81, montre qu'elle est plus petite que 1 ns. Quant au ratio A_1/A_2 , il a une valeur moyenne de $1,001 \pm 0,007$, mais la distribution sur la figure 5.80 montre deux pics de part et d'autres de cette valeur. De plus, le $\widetilde{\chi^2}$ prend une valeur de 13 ± 4 et sa distribution est illustrée sur la figure 5.82. Les formes moyennes expérimentales donnent donc un très mauvais accord avec les signaux provenant du ICs.

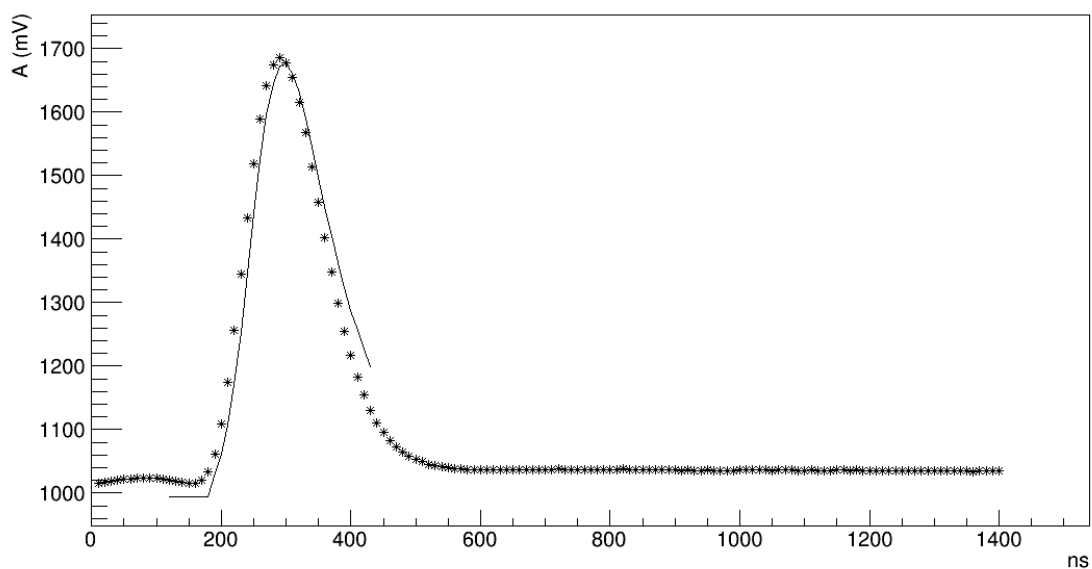


Figure 5.77 : Exemple d'ajustement de l'équation (4.17b) sur un signal généré dans un cristal de ICs, avec la forme moyenne expérimentale optimale et les paramètres A , B et P correspondants.

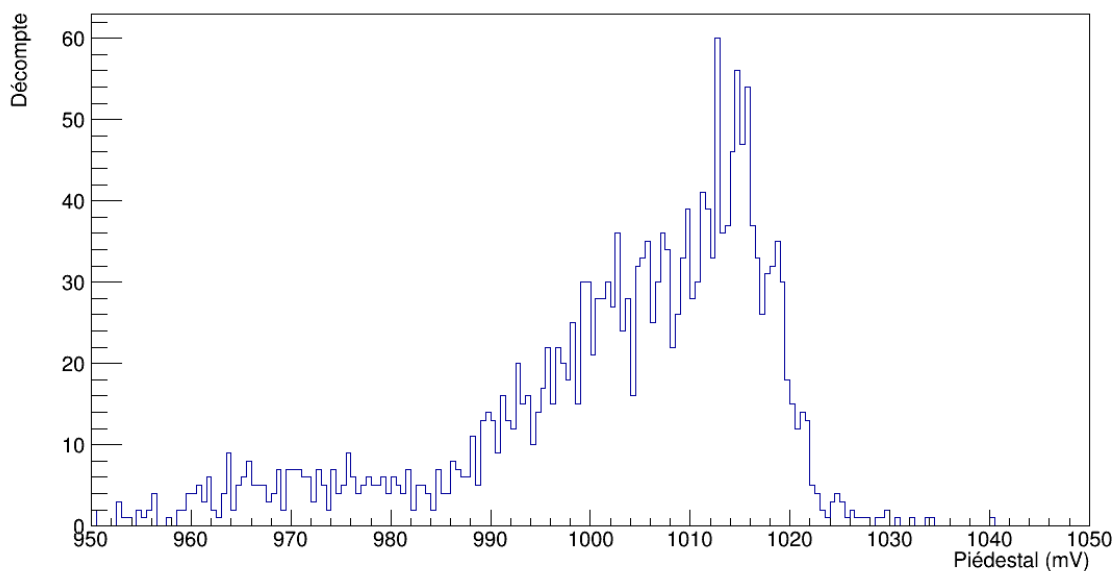


Figure 5.78 : Distribution du piédestal, évalué avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.

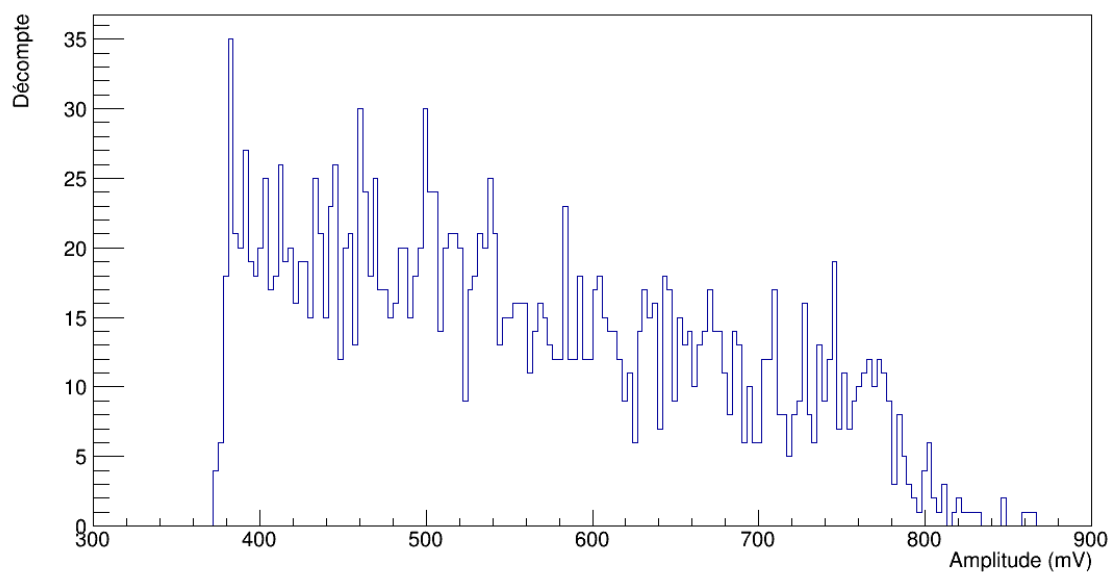


Figure 5.79 : Distribution d'amplitudes évaluées avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.

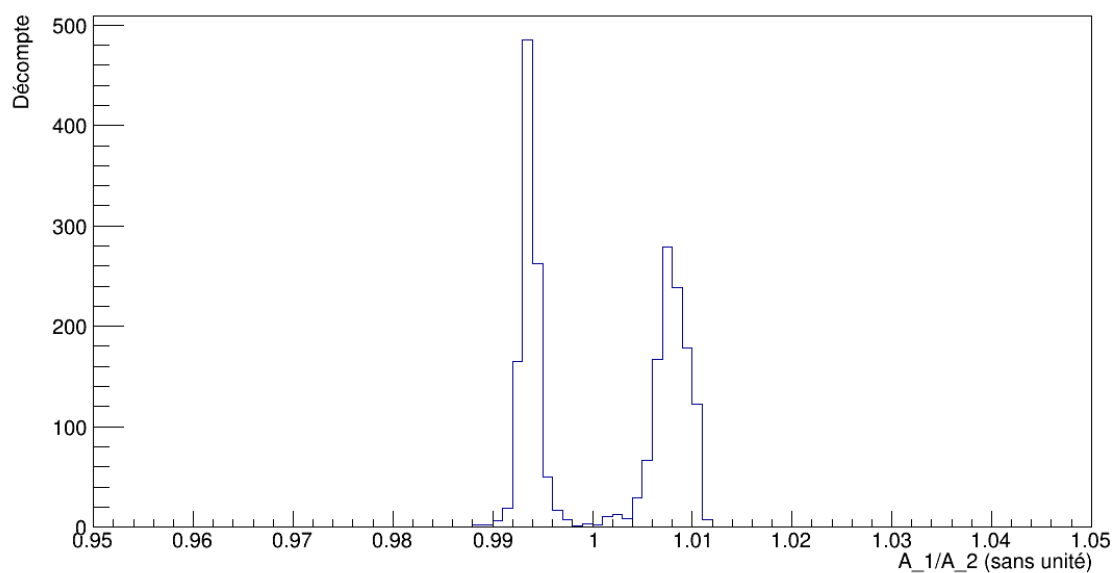


Figure 5.80 : Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenu avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.

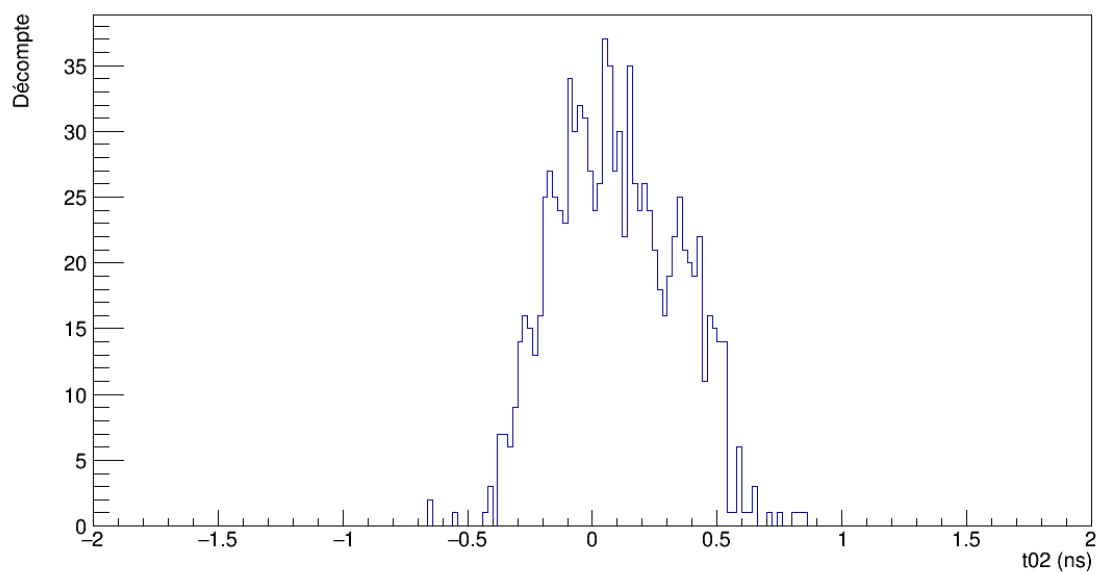


Figure 5.81 : Distribution de t_{0_2} , évalué avec les coefficients des formes moyennes expérimentales pour des signaux générés dans un cristal de ICs.

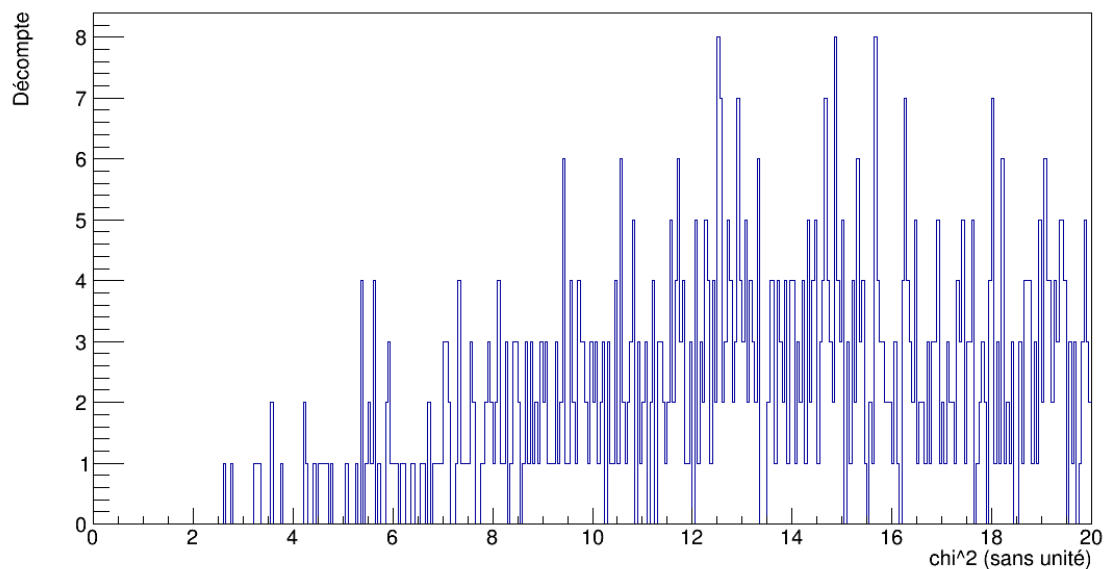


Figure 5.82 : Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme moyenne expérimentale optimale pour chacun des signaux générés dans un cristal de ICs.

5.3.2.2 Filtres actifs

Avec l'expression pour les filtres actifs, un exemple d'ajustement avec la forme de référence optimale et les paramètres A , B et P associés est montré sur la figure 5.83, avec un mauvais accord. La valeur obtenue pour le piédestal est de 998 ± 7 mV. La distribution est illustrée sur la figure 5.84. La figure 5.85 montre la distribution d'amplitudes, dont la moyenne est de 550 ± 200 mV. Quant au ratio A_1/A_2 , on obtient $1,003 \pm 0,003$ en moyenne et la distribution est illustrée sur la figure 5.86. Pour ce qui est du décalage dans le temps t_{0_2} , on voit sur la figure 5.87 qu'il n'est pas nécessairement plus petit que 1 ns. De plus, le $\tilde{\chi}^2$, illustré sur la figure 5.88, tourne autour de 70 ± 30 , ce qui signifie un très mauvais accord entre ces formes de référence et les signaux générés dans le ICs.

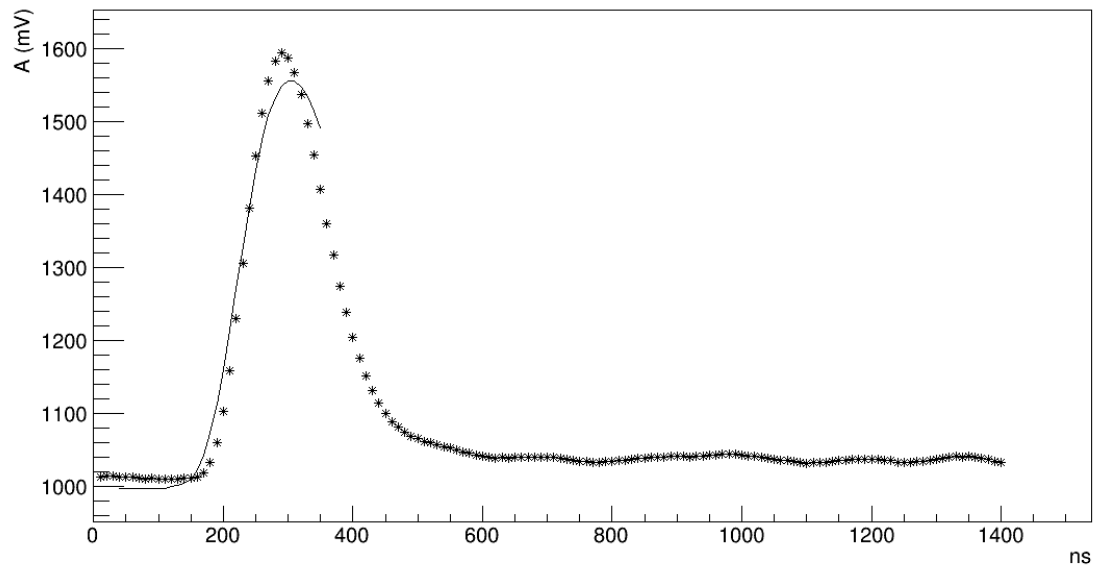


Figure 5.83 : Exemple d'ajustement de l'équation (4.17b) sur un signal généré dans un cristal de ICs, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir de l'expression (5.1) des filtres actifs.

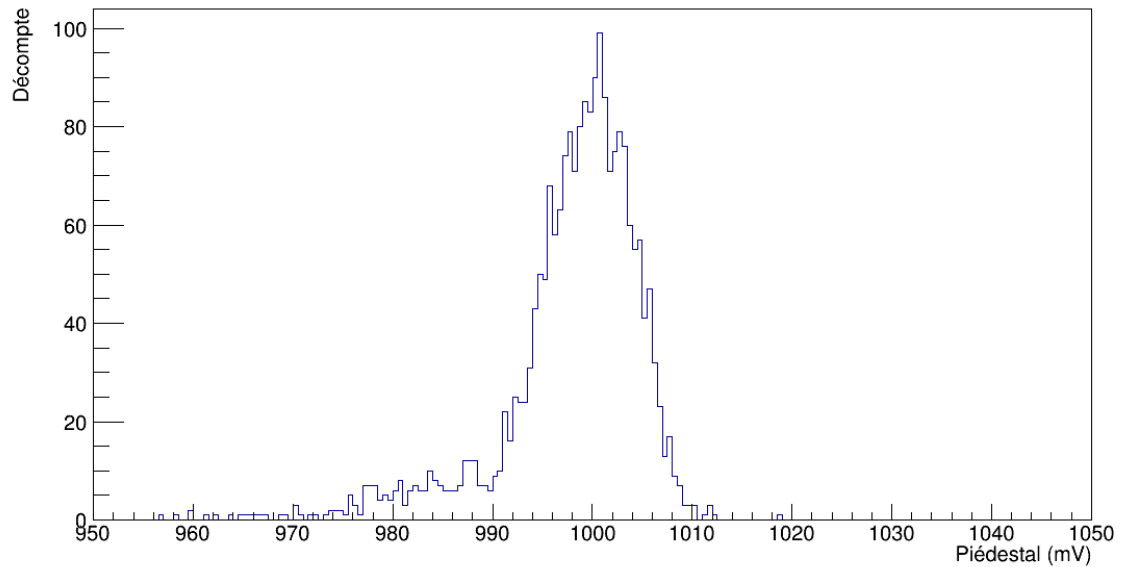


Figure 5.84 : Distribution du piédestal, évalué avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.

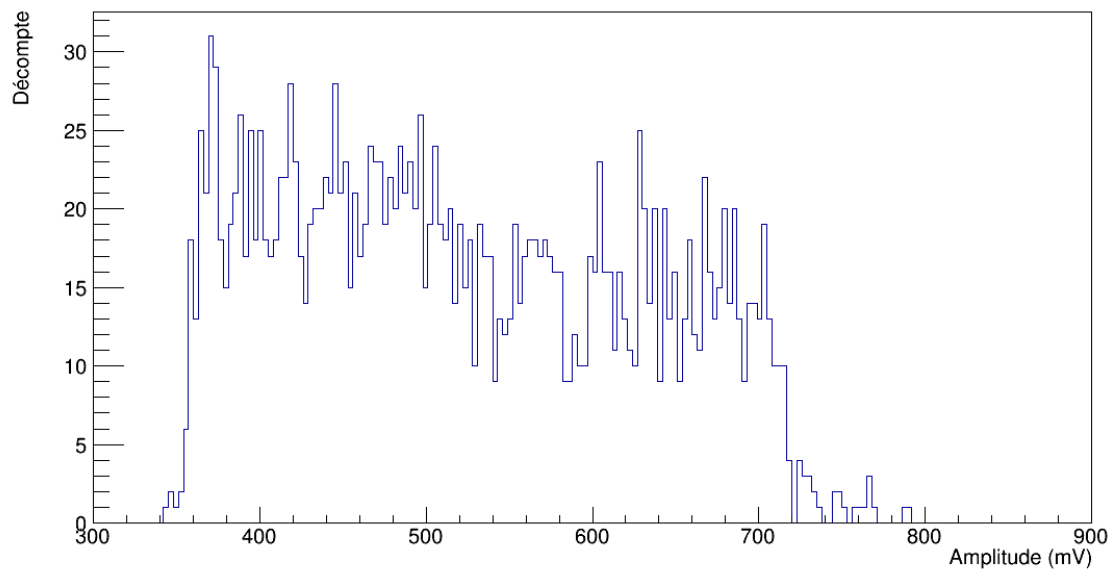


Figure 5.85 : Distribution d'amplitudes évaluées avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.

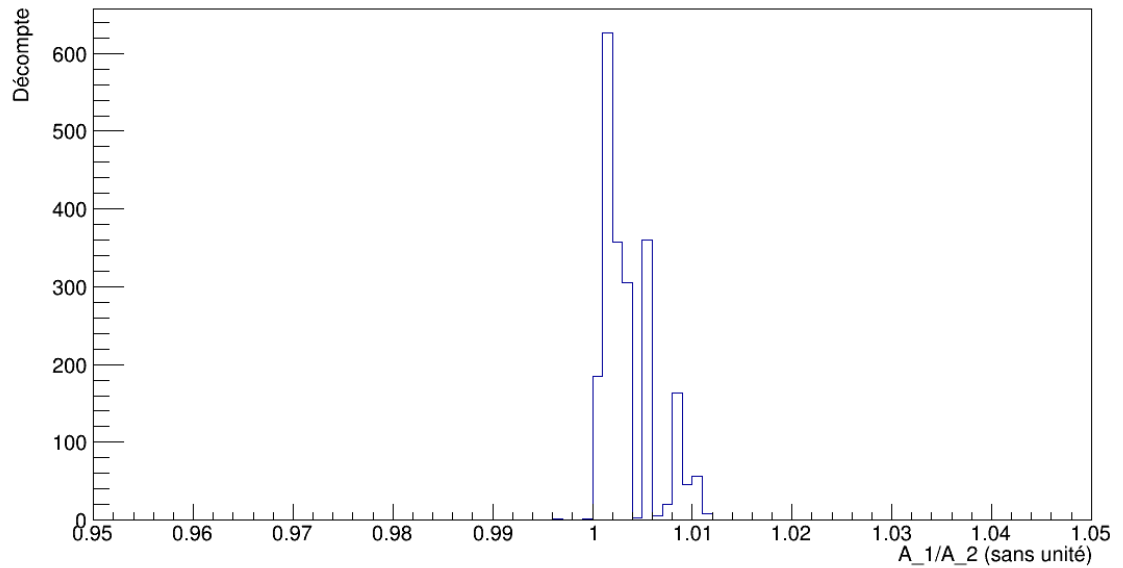


Figure 5.86 : Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenue avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.

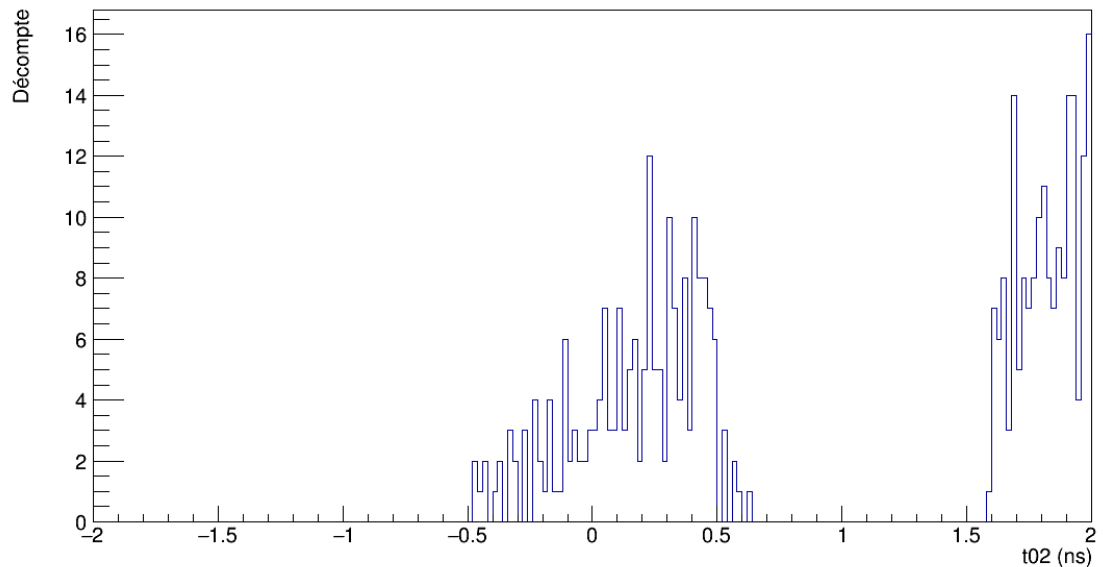


Figure 5.87 : Distribution de t_{02} , évalué avec les coefficients des formes de référence pour l'expression (5.1) des filtres actifs pour des signaux générés dans un cristal de ICs.

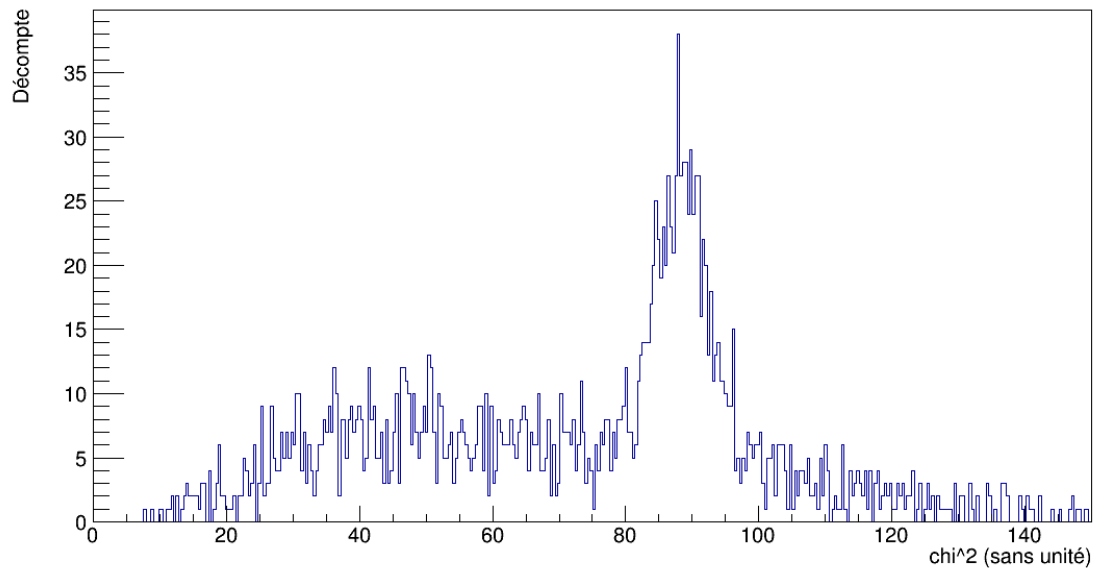


Figure 5.88 : Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de l'expression (5.1) des filtres actifs pour chacun des signaux générés dans un cristal de ICs.

5.3.2.3 Somme de 3 gaussiennes

Dans le cas d'une somme de 3 gaussiennes, la figure 5.89 illustre un exemple d'ajustement sur un signal avec la forme de référence optimale et les paramètres A , B et P associés, avec un très bon accord. Le piédestal vaut en moyenne 1012 ± 9 mV et l'amplitude vaut 600 ± 200 mV. Leurs distributions sont illustrées sur les figures 5.90 et 5.91. Sur la figure 5.92, on voit la distribution pour le ratio A_1/A_2 , dont la valeur moyenne est de $1,001 \pm 0,001$. Pour ce qui est du décalage dans le temps t_{0_2} , il reste à l'intérieur d'un intervalle de $[-0,5; 0,5]$ ns, comme on peut le voir sur la figure 5.93. Aussi, le $\widetilde{\chi^2}$ moyen est de 8 ± 4 . Sa distribution est plutôt large, comme le montre la figure 5.94. Ceci implique que l'accord n'est pas si bon, mais c'est principalement causé par des fluctuations dues au bruit, surtout dans les régions du piédestal et de la queue.

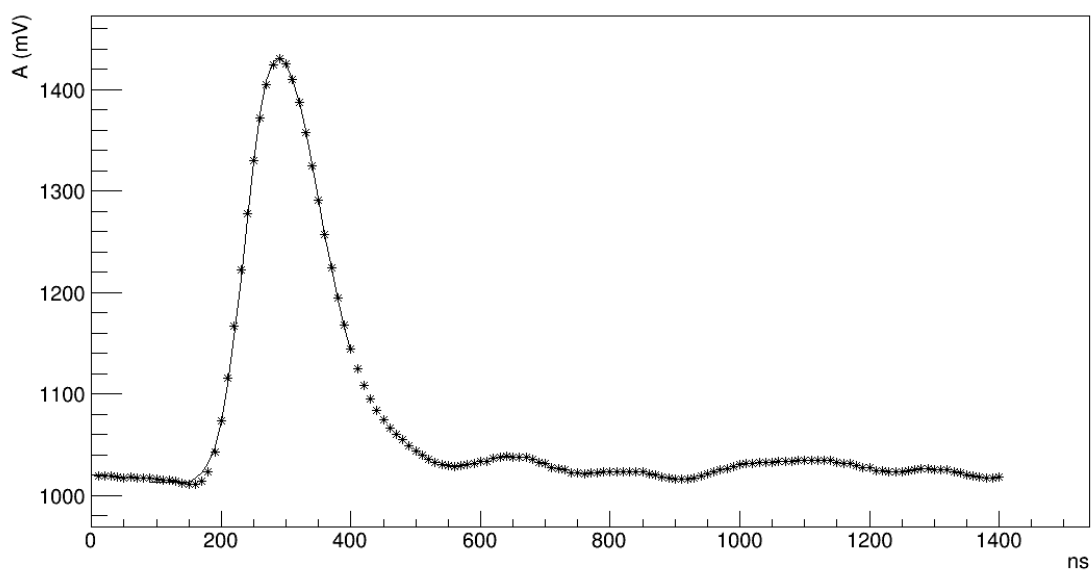


Figure 5.89 : Exemple d'ajustement de l'équation (4.17b) sur un signal généré dans un cristal de ICs, avec la forme de référence optimale et les paramètres A , B et P correspondants, à partir d'une somme de 3 gaussiennes.

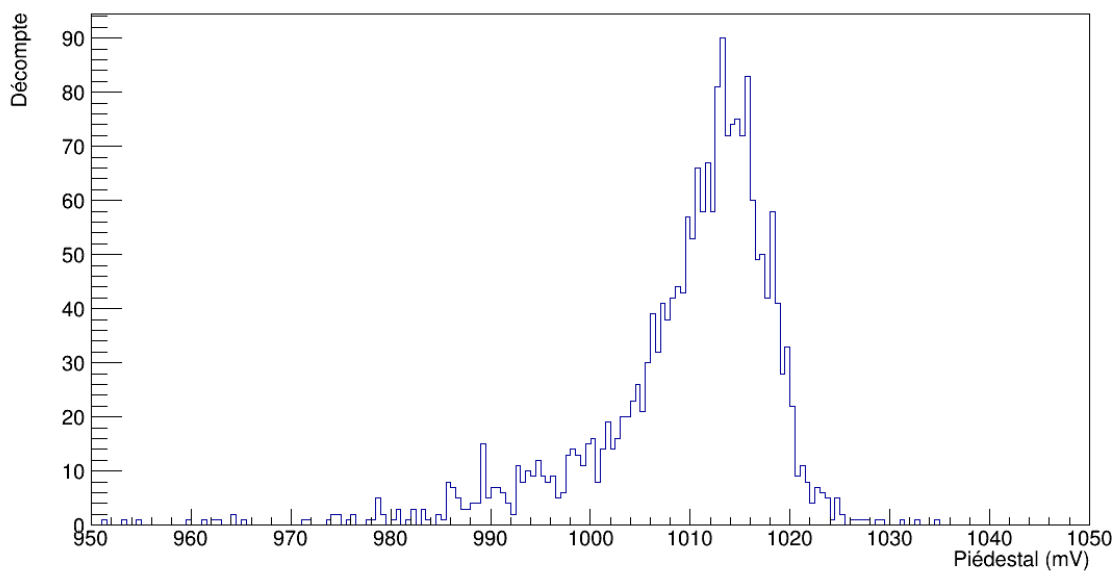


Figure 5.90 : Distribution du piédestal, évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.

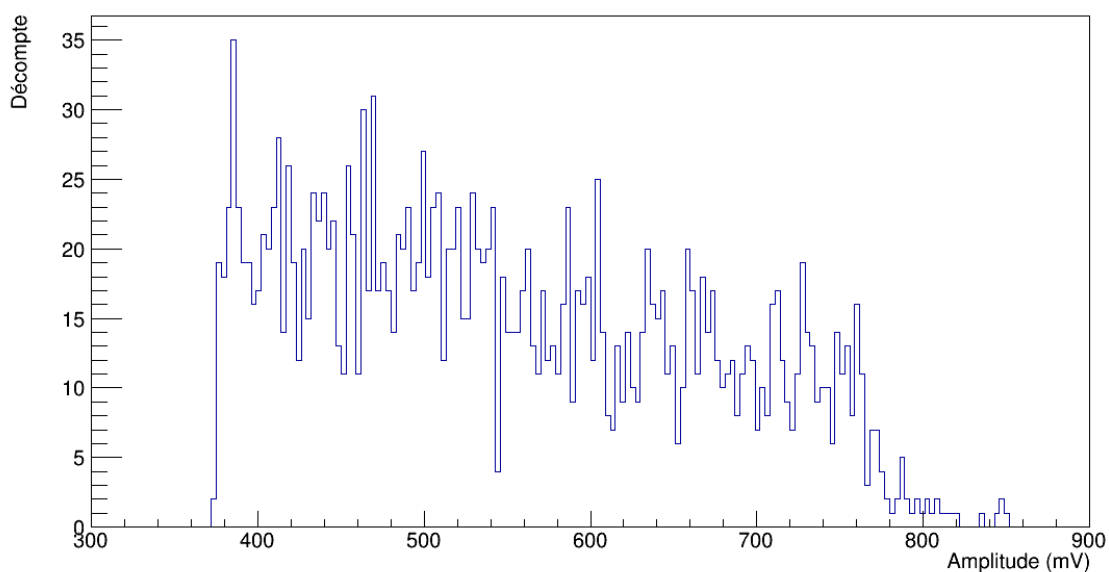


Figure 5.91 : Distribution d'amplitudes évaluées avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.

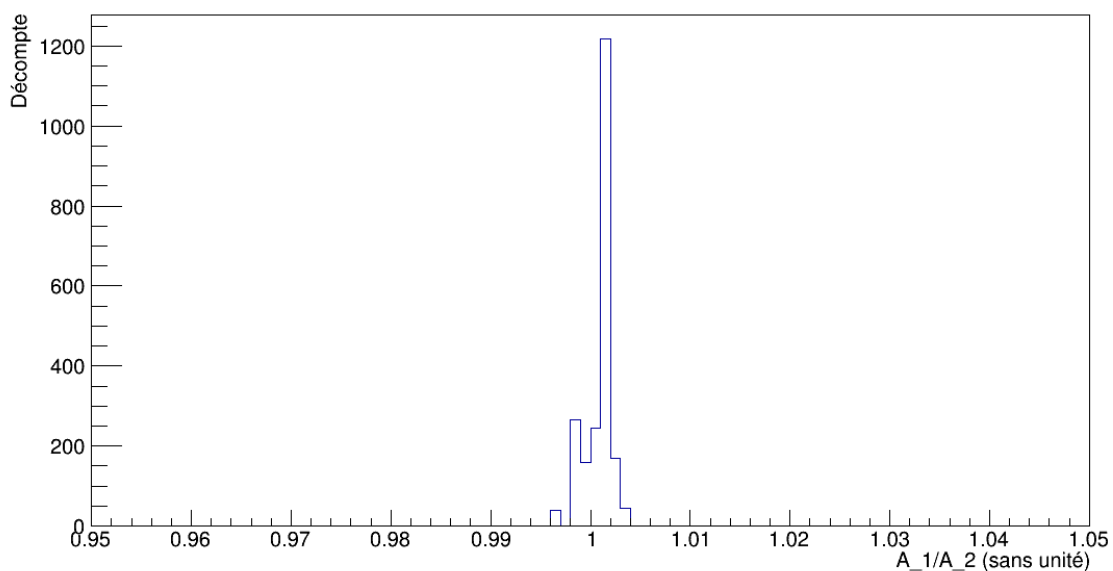


Figure 5.92 : Distribution du ratio d'amplitudes aux première et deuxième évaluations, obtenue avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.

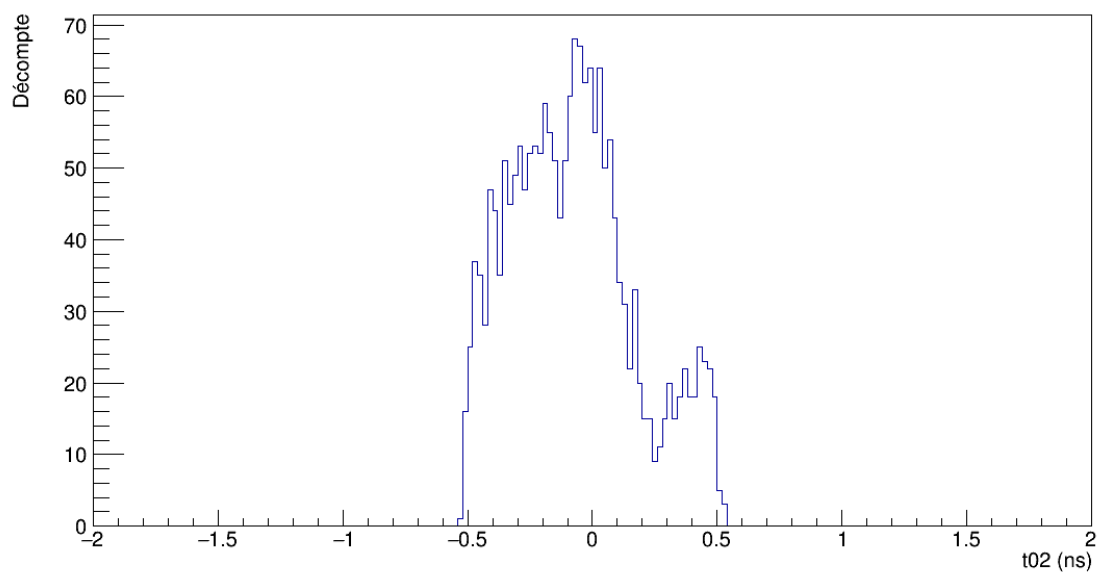


Figure 5.93 : Distribution de t_{02} , évalué avec les coefficients des formes de référence pour la somme de 3 gaussiennes pour des signaux générés dans un cristal de ICs.

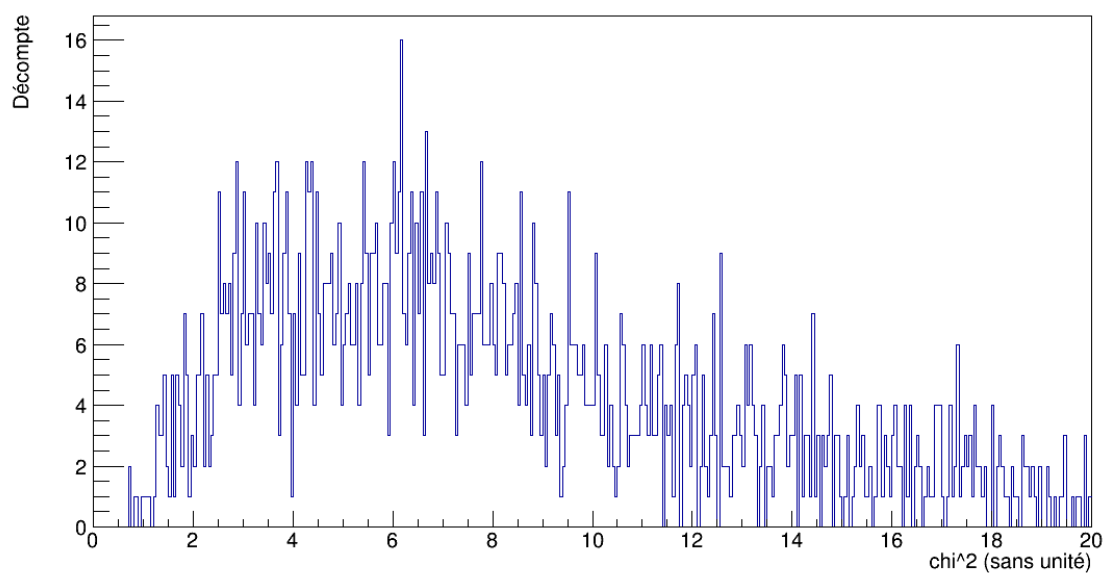


Figure 5.94 : Distribution du $\widetilde{\chi^2}$ individuel, évalué avec la forme de référence optimale à partir de la somme de 3 gaussiennes pour chacun des signaux générés dans un cristal de ICs.

5.3.2.4 Comparaison

Les valeurs obtenues pour le piédestal, soient 1000 ± 20 mV avec les formes moyennes expérimentales et 1012 ± 9 mV avec la somme de 3 gaussiennes, sont en accord avec la valeur mesurée de 1013 ± 7 mV à la section 5.2. Par contre, la valeur de 998 ± 7 mV obtenue avec l'expression (5.1) des filtres actifs n'est pas en accord avec ce piédestal mesuré, puisque la différence entre ces deux valeurs est de 15 ± 10 mV. On remarque d'ailleurs plus de fluctuations sur les distributions des figures 5.78, 5.84 et 5.90 pour le piédestal du ICs que sur les distributions des figures 5.60, 5.66 et 5.72 pour le piédestal avec le générateur d'impulsions. Pour ce qui est de l'amplitude, les distributions sur les figures 5.79, 5.85 et 5.91 ont la même forme et sont en accord avec les distributions sur les figures 5.33, 5.37 et 5.57, quoique celles pour les coefficients pour l'expression (5.1) des filtres actifs donnent des valeurs plus faibles. Quant au ratio A_1/A_2 , il prend la même valeur pour les coefficients des formes moyennes expérimentales et de la somme de 3 gaussiennes, quoique ce dernier a une plus petite incertitude, avec $1,001 \pm 0,007$ et $1,001 \pm 0,001$, respectivement. Cependant, on observe deux pics pour les formes moyennes expérimentales sur la figure 5.80, de part et d'autre de la valeur moyenne, indiquant un manque de stabilité dans l'évaluation des paramètres. Par contre, la distribution de la figure 5.92 pour la somme de 3 gaussiennes est beaucoup mieux définie. Avec les coefficients pour l'expression (5.1) des filtres actifs, on a un ratio de $1,003 \pm 0,003$. L'amplitude est donc légèrement surestimée à la première évaluation A_1 , plus qu'avec les deux autres ensembles de coefficients, mais la distribution de la figure 5.86 n'est pas aussi bien définie que celle de la figure 5.92 pour la somme de 3 gaussiennes. En ce qui a trait au décalage dans le temps, il est estimé avec une bonne précision dès la première évaluation, comme le montre les distributions sur les figures 5.81 et 5.93 pour les formes moyennes expérimentales et la somme de 3 gaussiennes, où le décalage t_0 à la deuxième évaluation se trouve généralement dans un intervalle de $[-0,5; 0,5]$ ns. Toutefois, avec l'expression (5.1) des filtres actifs, t_0 peut prendre des valeurs plus grandes que 1 ns et alors la précision désirée de 1 ns sur le temps d'occurrence n'est plus possible. Pour ce qui est du $\widetilde{\chi^2}$, les ajustements faits avec les coefficients pour

l'expression (5.1) des filtres actifs est très mauvais, avec une valeur moyenne de 70 ± 30 . Les coefficients pour la somme de 3 gaussiennes donnent les meilleurs résultats, avec un $\widetilde{\chi^2}$ de 8 ± 4 , tandis qu'avec les coefficients pour les formes moyennes expérimentales, on a 13 ± 4 . Il ne faut pas oublier que si ces valeurs ne sont pas aussi près de 1 qu'on le souhaiterait, c'est en raison de fluctuations dues au bruit beaucoup plus importantes sur les signaux en provenance du ICs. Ainsi, comme on pouvait s'y attendre avec les résultats de la section 5.2, les coefficients pour la somme de 3 gaussiennes sont les plus fiables. Quant au problème de fluctuations, peut-être suffirait-il de blinder le *shaper*.

CHAPITRE 6

CONCLUSION

Les motivations de Belle II que sont l'étude des saveurs lourdes, de la violation CP et de la physique au-delà du Modèle Standard requièrent l'acquisition d'un ensemble statistique très grand, d'où la luminosité record prévue pour cette expérience. Autrement dit, les détecteurs doivent être adaptés pour un taux de comptage plus élevé, surtout dans un contexte de bruit de fond important, comme pour Belle II où ce bruit sera dominé par l'effet Touschek.

Ce travail se concentrant sur un calorimètre électromagnétique utilisant un cristal de ICs pur, les particules qui peuvent être détectées sont principalement des photons, des électrons et des positrons, mais aussi des pions neutres par leur désintégration en paires de photons. Toutes ces particules déposent complètement leur énergie dans le cristal, sauf les muons qui peuvent le traverser de part et d'autre du fait de sa grande masse par rapport aux électrons atomiques. Comme les muons perdent leur énergie principalement par collisions et ionisation, on a calculé que ceux dans le régime du minimum du potentiel d'ionisation, les plus probables dans nos conditions d'expérimentation, déposent en moyenne 51,59 MeV dans le ICs. Les photons, électrons et positrons quant à eux perdent toute leur énergie par la formation de gerbes électromagnétiques. Éventuellement, lorsque les particules d'une gerbe n'ont plus assez d'énergie pour l'entretenir, les atomes du cristal sont excités par l'énergie de cette gerbe pour ensuite la réémettre sous forme de scintillation. On a tenté d'estimer les temps de scintillation pour les deux états d'excitation accessibles, cependant les valeurs de 47 ± 4 ns et $3,4 \pm 0,5$ μ s sont en désaccord avec les valeurs attendues de 16 ns et 1 μ s. Cette différence peut s'expliquer par des réflexions à la surface du cristal, ainsi que par sa géométrie allongée, qui peuvent augmenter le temps d'arrivée de la scintillation. De plus, ces temps peuvent être modifiés par l'intégration de la charge sur les condensateurs.

La scintillation du ICs est convertie en signal électronique par une photopentode de Hamamatsu dont le gain est de 150 ± 2 électrons par photoélectron. Ce signal est amplifié

par un préamplificateur, contrôlé à l'entrée par un condensateur de 2 pF et une résistance de 10 M Ω . Alors qu'on s'attendait à une constante de temps de $RC = 20 \mu\text{s}$, on obtient plutôt $30 \pm 2 \mu\text{s}$. Le signal passe ensuite par le shaper, un filtre actif de type CR-(RC)⁴ qui fait une mise en forme des signaux, avec un gain mesuré de 14, plutôt que la valeur attendue de 20. Ces signaux sont ensuite numérisés par un ADC à un taux de 100 MHz, permettant leur traitement numérique par le DSP, implémenté sur un FPGA. Une règle a été établie pour convertir l'amplitude des signaux sortant du préamplificateur en énergie déposée par une particule. Une autre règle a été établie pour la conversion de l'amplitude des signaux sortant du shaper vers celle des signaux entrant. La combinaison de ces deux règles pour convertir l'amplitude des signaux numérisés en énergie déposée dépend de la présence ou non d'une atténuation par un facteur k entre le préamplificateur et le *shaper*, ce qui mène à l'équation (4.16). De plus, le shaper a une gamme dynamique très étroite, d'à peine 1,9 mV environ d'après la figure 4.26.

Des trois approches étudiées pour définir les formes de référence utilisées par le DSP, les ajustements non-linéaires individuels les plus optimaux étaient obtenus avec une somme de 3 gaussiennes, avec un $\widetilde{\chi}^2$ de $1,1 \pm 0,1$ avec les signaux produits par un générateur et 3 ± 2 avec ceux en provenance du ICs. Cette fonction est plus flexible pour la convergence vers un minimum global que celle pour un filtre actif, peu importe si les signaux sont produits par un générateur d'impulsions ou par une particule dans un cristal de ICs. De plus, avec l'expression pour un filtre actif, le $\widetilde{\chi}^2$ est de $1,1 \pm 0,3$ avec le générateur d'impulsions, mais il est très instable et mal défini avec les signaux en provenance du ICs. Quant aux formes moyennes expérimentales, elles sont affectées par le bruit sur les signaux de l'ensemble statistique, plus particulièrement avec le ICs pour lequel les formes de références sont inutilisables.

Dans l'application de la méthode du DSP pour vérifier la qualité des formes de référence obtenues, avec les coefficients obtenus avec chacune des trois méthodes étudiées, la somme de 3 gaussiennes s'avère être la plus fiable pour l'évaluation de l'amplitude et du temps d'occurrence, autant avec le générateur d'impulsions qu'avec le ICs, avec lesquels le $\widetilde{\chi}^2$ est de $1,5 \pm 0,4$ et de 8 ± 4 , respectivement. Ceci est dû au manque de régularité dans les formes moyennes expérimentales, surtout avec le ICs, et au manque

de stabilité dans la détermination des paramètres qui permettent de définir les formes de référence pour l'équation (5.1) pour les filtres actifs, tandis que la somme de 3 gaussiennes a l'avantage de la flexibilité. De plus, avec un ensemble de coefficients fiables, on a vu que le décalage dans le temps est estimé avec une bonne précision dès la première évaluation par le DSP, puisque la deuxième évaluation demeure généralement dans un intervalle de $[-0,5; 0,5]$ ns. Cependant, l'amplitude peut être légèrement surestimée, d'au plus 0,3% environ, d'où l'utilité de la deuxième évaluation. Aussi, avec l'expression pour les filtres actifs, le piédestal de 998 ± 7 mV extrait avec la méthode du DSP est en désaccord avec la valeur moyenne mesurée de 1013 ± 7 mV, avec une différence de 15 ± 10 mV. Enfin, avec le générateur d'impulsions et le cristal de ICs, respectivement, le $\widetilde{\chi^2}$ avec l'expression pour les filtres actifs est de $8,0 \pm 0,9$ et de 70 ± 30 , tandis qu'avec les formes moyennes expérimentales, on obtient $1,5 \pm 0,9$ et 13 ± 4 . Les formes moyennes expérimentales donnent de meilleurs résultats que l'expression pour les filtres actifs, mais on obtient les formes de référence les plus fiables avec la somme de 3 gaussiennes.

BIBLIOGRAPHIE

- [1] The Speed and Decay of Cosmic-Ray Muons : Experiments in the Relativistic Kinematics of the Universal Speed Limit and Time Dilation. *MIT - Department of Physics*, 2014.
- [2] S. K. Banerjee et B. G. Streetman. *Solid State Electronic Devices, Seventh Edition*. Pearson Education, 2014.
- [3] A. Blankman. Read LeCroy Binary Waveform : Matlab Code. *LeCroy Corporation*, 2007.
- [4] C. Cohen-Tannoudji, B. Diu et F. Laloë. *Mécanique quantique II*. Hermann, éditeurs des sciences et des arts, 1973.
- [5] A. Abashian et al. The Belle detector. *Nuclear Instruments & Methods in Physics Research : Section A*, 479(1):117–232, 2002.
- [6] T. Abe et al. *Belle II Technical Design Report*. High Energy Accelerator Research Organization, 2010.
- [7] H. P. Gavin. The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering - Duke University*, 3, 2016.
- [8] P. Horowitz et W. Hill. *The Art Of Electronics, 2nd Edition*. Cambridge University Press, 1989.
- [9] Y. H. Kuo. B. Sc. Thesis : Determination of the Angular Distribution of Cosmic Rays at Sea Level. *Massachusetts Institute of Technology : Department of Physics*, 2010.
- [10] D. C. Lay. *Linear Algebra and its applications, 3rd Edition*. Pearson Education, 2003.

- [11] C. Leroy. PHY 6638 Physique Subatomique Instrumentale : Notes de cours. 2015.
- [12] John Caunt Scientific Limited. Scintillation Detectors : Physical Properties. Repéré à <http://www.johncaunt.com/detectors/scintillation-detectors/scintillator-properties/>.
- [13] B. R. Martin et G. Shaw. *Particle Physics : Third Edition*. John Wiley and Sons, 2008.
- [14] J.-P. Martin, N. A. Starinski et P. Taras. Fast charge-sensitive preamplifier for pure CsI crystals. *Nuclear Instruments and Methods*, 778:120–125, 2014.
- [15] J. P. Martin. An improved method for the autocalibration of the GRIF-4G modules : TIGRESS-GRIFFIN collaboration meeting. 2015.
- [16] K. A. Olive et al. Particle Data Group. *Chinese Physics C*, 38(9):399, 2014.
- [17] G. Remmen et E. McCreary. Measurement of the Speed and Energy Distribution of Cosmic Ray Muons. *Journal of Undergraduate Research in Physics*, 25, 2012.
- [18] Saint-Gobain. CsI(pure) Cesium Iodide Scintillation Material. *Saint-Gobain Ceramics & Plastics Inc.*, .
- [19] Saint-Gobain. CsI(Tl), CsI(Na) Cesium Iodide Scintillation Material. *Saint-Gobain Ceramics & Plastics Inc.*, .
- [20] V. Shebalin. ECL DAQ status : B2GM meeting. 2015.
- [21] P. Taras. *Concepts de Physique Nucléaire*. Université de Montréal, 2008.
- [22] J.R. Taylor. *An Introduction To Error Analysis : Second Edition*. University Science Books, 1997.
- [23] M. Thomson. *Modern Particle Physics*. Cambridge University Press, 2013.
- [24] U. Tietze, C. Schenk et E. Gamm. *Electronics Circuits : Handbook for Design and Application*. Springer, 2008.

- [25] A.N. Vinokurova. B. Sc. Thesis : Reconstruction of Time and Energy In Super-Belle Calorimeter. *Novosibirsk State University : Faculty of Physics*, page 10, 2004.
- [26] H. Winick, G. Brown, K. Halbach et J. Harris. Wiggler and Undulator Magnets. *Physics Today*, 34(5):50–63, Mai 1981.

Annexe I

Régression linéaire

On peut faire la régression linéaire d'un polynôme de puissance n de la forme [22] :

$$f_i = c_n x_i^n + c_{n-1} x_i^{n-1} + \dots + c_2 x_i^2 + c_1 x_i + c_0 \quad (\text{I.1})$$

où f_i est la valeur estimée par le modèle en fonction de x_i et c_j est le j -ème des $n + 1$ paramètres du polynôme. Pour déterminer les valeurs optimales des paramètres, il faut minimiser la différence entre les N mesures y_i et les prédictions f_i du polynôme à partir du résidu :

$$S = \sum_{i=0}^N (y_i - f_i)^2 \quad (\text{I.2})$$

On calcule donc les dérivées partielles $\frac{\partial S}{\partial c_j}$ par rapport à chacun des paramètres et on obtient :

$$\sum_{i=0}^N (y_i - f_i) x_i^j = 0 \quad (\text{I.3})$$

qui peut se réécrire sous la forme :

$$\sum_{i=0}^N (y_i x_i^j) = \sum_{i=0}^N (f_i x_i^j) \quad (\text{I.4})$$

Le système d'équations qu'on obtient pour toutes ces dérivées partielles peut s'écrire sous la forme matricielle $MA = Y$:

$$\begin{pmatrix} \sum x_i^{2n} & \sum x_i^{2n-1} & \dots & \sum x_i^{n+1} & \sum x_i^n \\ \sum x_i^{2n-1} & \sum x_i^{2n-2} & \dots & \sum x_i^n & \sum x_i^{n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sum x_i^{n+1} & \sum x_i^n & \dots & \sum x_i^2 & \sum x_i^1 \\ \sum x_i^n & \sum x_i^{n-1} & \dots & \sum x_i^1 & N \end{pmatrix} \begin{pmatrix} c_n \\ c_{n-1} \\ \vdots \\ c_1 \\ c_0 \end{pmatrix} = \begin{pmatrix} \sum y_i x_i^n \\ \sum y_i x_i^{n-1} \\ \vdots \\ \sum y_i x_i \\ \sum y_i \end{pmatrix}$$

La matrice A des paramètres du polynôme peut être déterminée en multipliant la matrice Y par l'inverse de la matrice M , qu'on obtient par la méthode de Gauss-Jordan [10].

I.1 Cas particulier d'une droite

Dans le cas particulier d'une droite $y = Ax + B$, où la puissance du polynôme est de 1, le système d'équations peut facilement se résoudre analytiquement pour donner les paramètres A et B selon les expressions suivantes [22] :

$$A = \frac{N(\sum x_i y_i) - (\sum x_i)(\sum y_i)}{\Delta} \quad (\text{I.5})$$

$$B = \frac{(\sum x_i^2)(\sum y_i) - (\sum x_i)(\sum x_i y_i)}{\Delta} \quad (\text{I.6})$$

$$\Delta = N(\sum x_i^2) - (\sum x_i)^2 \quad (\text{I.7})$$

où N est le nombre d'échantillons dans un signal.

Annexe II

Algorithme de Levenberg-Marquardt

II.1 Problème à résoudre

La méthode de Levenberg-Marquardt est utile pour les cas non-linéaires, c'est-à-dire les fonctions qui n'ont pas de dépendance linéaire sur chacun de ses paramètres. Le problème à résoudre consiste à minimiser le résidu suivant pour un cas non-linéaire :

$$S(\beta) = \sum r_i^2(\beta) \quad (\text{II.1})$$

où $\beta = \{\beta_1, \beta_2, \dots, \beta_n\}$ est un vecteur dont chaque composante correspond à un des paramètres du modèle f utilisé et $r_i(\beta) = y_i - f(t_i, \beta)$ est la différence entre une mesure y_i et la valeur prédite par le modèle au temps t_i correspondant.

Les méthodes de régression non-linéaire sont itératives. À partir d'un ensemble de valeurs initiales pour les paramètres β du modèle testé, on se rapproche graduellement des valeurs optimales selon l'expression linéaire suivante :

$$\beta^{s+1} = \beta^s + \Delta\beta \quad (\text{II.2})$$

où $\Delta\beta$ sont les variations qu'on appliquera sur les paramètres β_i^s pour cette itération, qui deviendront β_i^{s+1} pour la prochaine.

II.2 Méthode de Gauss-Newton

La méthode de Gauss-Newton [7] permet de linéariser le problème, avec $\Delta\beta$ qui doit satisfaire les équations normales :

$$(J_r^T J_r) \Delta\beta = -J_r^T r \quad (\text{II.3})$$

où J_r et J_r^T sont le jacobien et sa transposée, définis selon chacun des paramètres β_i avec les résidus $r_i = y_i - f(x_i, \beta)$. Les éléments du jacobien étant définis comme $\frac{\partial r_i}{\partial \beta_j} = -\frac{\partial f_i}{\partial \beta_j}$, où i désigne la ligne de J_r et j la colonne, l'équation (II.3) peut se réécrire

$$(J_f^T J_f) \Delta \beta = J_f^T r \quad (\text{II.4})$$

Alors, l'équation (II.2) devient

$$\beta^{s+1} = \beta^s + (J_f^T J_f)^{-1} J_f^T r \quad (\text{II.5})$$

De plus, il est possible que même si le vecteur de variations pour les paramètres de la fonction ajustée pointe dans la bonne direction, que les variations soient trop grandes. Il est alors préférable d'utiliser une version modifiée, soit

$$\beta^{s+1} = \beta^s + \alpha \Delta \beta \quad (\text{II.6})$$

où α est le facteur dans l'intervalle de $[0, 1]$ qui induit la plus petite somme (II.1) pour les variations $\Delta \beta$ calculées à cette itération. Le facteur α est réévalué à chaque itération.

L'efficacité de cette méthode est sensible à la complexité de la fonction ajustée. Si les valeurs initiales pour les paramètres sont choisies trop loin des valeurs optimales, la méthode peut converger vers une solution erronée ou tout simplement diverger. Ceci repose sur la supposition d'une convergence locale quadratique.

II.3 Méthode du gradient

Le gradient de l'équation (II.1) est donné par l'expression [7] :

$$\frac{\partial S(\beta)}{\partial \beta} = -2J_f^T r \quad (\text{II.7})$$

Ainsi, la méthode du gradient minimise S par un déplacement dans la direction opposée. Les variations de paramètres sont alors obtenues selon :

$$\Delta\beta = \kappa J_f^T r \quad (\text{II.8})$$

où κ est un facteur qui détermine la longueur du pas d'itération pour les paramètres β . Même si le déplacement se fait nécessairement dans une direction de minimisation, ce facteur assure que le déplacement n'est pas trop grand. Un énorme avantage de la méthode du gradient est que la convergence sur un minimum, global ou local, est certaine. Cependant, le temps requis pour une convergence satisfaisante peut être très long.

II.4 Algorithme de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt [7] combine les avantages de ces deux méthodes pour pallier aux défauts de chacune et se présente sous la forme :

$$((J_f^T J_f) + \lambda I)\Delta\beta = J_f^T r \quad (\text{II.9})$$

où I est la matrice identité et λ est un facteur qui détermine la dominance relative entre les deux méthodes. Initialement, λ a une valeur élevée, afin d'exploiter la méthode du gradient qui converge lentement, mais sûrement. Chaque fois que l'on obtient un meilleur accord entre les mesures et le modèle, on garde ces nouvelles valeurs de paramètres et on divise λ par une constante $\nu > 1$. À mesure que λ diminue, la méthode de Gauss-Newton domine de plus en plus, ce qui accélère la convergence. Dans le cas où les nouvelles valeurs de paramètres ne donnent pas un meilleur accord, on les rejette et on multiplie λ par le facteur ν . Dans l'implémentation développée pour ce travail, on a choisi $\nu = 1,2$. Pour des valeurs de ν plus élevées, la méthode de Gauss-Newton prend le dessus trop rapidement et empêche une convergence fiable.

La contribution de Marquardt fut de modifier l'équation (II.9) en remplaçant la matrice identité par la diagonale de $J^T J$, ce qui permet d'étalonner le gradient en fonction de la courbure du modèle à ajuster. Ainsi, la convergence peut se faire plus rapidement

dans le cas où le gradient est petit et le facteur λ devient moins dépendant des spécificités du problème. La méthode utilisée prend donc la forme :

$$((J_f^T J_f) + \lambda \text{diag}(J_f^T J_f)) \Delta \beta = J_f^T r \quad (\text{II.10})$$

II.5 Critères de convergence

Les variations de paramètres calculées à chaque itération ne sont acceptées que si la condition suivante est satisfaite [7] :

$$\rho = \frac{S(\beta) - S(\beta + \Delta \beta)}{\Delta \beta^T (\lambda \text{diag}(J_f^T J_f) \Delta \beta + J_f^T r(\beta))} > 0 \quad (\text{II.11})$$

Cette condition requiert que les résidus diminuent d'une itération à l'autre. Elle compare aussi l'amélioration des résidus relativement à celle des variations calculées, qui se base sur l'approximation linéaire :

$$f(x_i, \beta + \Delta \beta) \approx f(x_i, \beta) + J \Delta \beta \quad (\text{II.12})$$

La satisfaction d'un seul de trois critères détermine la convergence du modèle [7]. Tout d'abord, si les variations de paramètres correspondent toutes à une proportion des paramètres plus petite qu'une limite prédéfinie ε_1 :

$$\max \left| \frac{\Delta \beta}{\beta_i} \right| \leq \varepsilon_1 \quad (\text{II.13})$$

Il en va de même si le gradient passe au-dessous d'une certaine limite ε_2 :

$$\max |J_f^T r| \leq \varepsilon_2 \quad (\text{II.14})$$

Les limites ε_1 et ε_2 prennent toutes les deux la valeur de 10^{-6} , avec laquelle on a une précision acceptable sur les valeurs des paramètres β et une convergence suffisamment

rapide. Le troisième critère correspond au $\widetilde{\chi}^2$, lorsqu'il s'approche de 1 :

$$\widetilde{\chi}^2 = \frac{S}{\sigma^2(N-M)} \leq 1 \quad (\text{II.15})$$

où σ est l'incertitude sur les mesures individuelles et N est le nombre de points utilisés pour l'ajustement d'un modèle à M paramètres. Enfin, on fixe le nombre maximal d'itérations pour limiter le temps d'exécution dans le cas d'une convergence lente. Dans ce travail, on fait au plus 200 itérations pour un seul ajustement.

Annexe III

Énergie d'un muon produit par désintégration du pion

Un pion chargé peut se désintégrer en un muon et un neutrino muonique, selon l'interaction électrodynamique quantique (QED) suivante [23] :

$$\pi^+ \rightarrow \mu^+ + \nu_\mu \quad (\text{III.1a})$$

$$\pi^- \rightarrow \mu^- + \bar{\nu}_\mu \quad (\text{III.1b})$$

L'énergie du muon est bien définie dans le référentiel du centre de masse du pion initial. La conservation de l'énergie et de la quantité de mouvement requiert que les quadriverbeurs P de chaque particule respectent l'expression :

$$\begin{aligned} P_\pi^2 &= (P_\mu + P_{\nu_\mu})^2 \\ &= P_\mu^2 + P_{\nu_\mu}^2 + 2P_\mu \cdot P_{\nu_\mu} \end{aligned} \quad (\text{III.2})$$

qui se réécrit :

$$m_\pi^2 = m_\mu^2 + m_{\nu_\mu}^2 + 2(E_\mu E_{\nu_\mu} - \vec{p}_\mu \cdot \vec{p}_{\nu_\mu}) \quad (\text{III.3})$$

Sachant que les lois de conservation impliquent que $m_\pi = E_\mu + E_{\nu_\mu}$ et que $\vec{p}_\mu = -\vec{p}_{\nu_\mu}$ et en considérant que la masse du neutrino est négligeable, on obtient :

$$\begin{aligned} m_\pi^2 &= m_\mu^2 + 2(E_\mu E_{\nu_\mu} + p_\mu^2) \\ &= m_\mu^2 + 2(E_\mu(m_\pi - E_\mu) + E_\mu^2 - m_\mu^2) \end{aligned} \quad (\text{III.4})$$

En réarrangeant cette équation, on obtient l'énergie d'un muon produit par la désintégration d'un pion chargé, dans le référentiel du centre de masse :

$$E_\mu = \frac{m_\pi^2 + m_\mu^2}{2m_\pi} \quad (\text{III.5})$$

Annexe IV

Temps de vie et masse du muon

IV.1 Coïncidences

Puisqu'il est important de bien sélectionner les muons pour déterminer avec précision leur temps de vie et que des briques de plomb ne suffisent pas pour que seuls ces particules traversent le cristal, on place un scintillateur de plastique de 1 cm d'épaisseur parallèlement au ICs. La sélection se fait alors par coïncidence, soit quand un signal produit dans le plastique précède un signal du ICs dans un temps suffisamment court, suggérant le passage d'un muon en provenance du ciel. Le montage utilisé est illustré sur la figure IV.1. On y voit que la scintillation du plastique est convertie par un photomultiplicateur, alimenté par une tension de 1,2 kV, tandis que celle du cristal passe par une photopentode suivie d'un préamplificateur. Les signaux sont envoyés dans l'oscilloscope LeCroy, qui s'enclenche et enregistre séparément les deux signaux lorsque celui en provenance du ICs dépasse le seuil d'enclenchement, d'environ 10 mV.

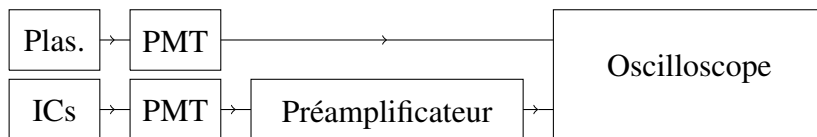


Figure IV.1 : Montage pour sélectionner des muons cosmiques par coïncidence entre un scintillateur de plastique et un cristal de ICs.

Un muon dont l'énergie cinétique équivaut à 1% de sa masse, soit environ 1 MeV, se déplace à 14% de la vitesse de la lumière, c'est-à-dire 4,2 cm/ns. À cette vitesse, un muon devrait généralement parcourir au moins 63 cm en 15 ns. En considérant que nos deux scintillateurs sont séparés de 20 cm, une distance de parcours de 63 cm entre eux correspond à un angle d'incidence d'environ $71,5^\circ$ par rapport à la verticale. Ainsi, on peut s'attendre à ce que les coïncidences ne surviennent majoritairement qu'à l'intérieur d'une fenêtre de 15 ns, d'autant plus que les muons cosmiques qui traversent le ICs arrivent selon une distribution angulaire proportionnelle à $\cos^2\theta$, où θ est l'angle

d'incidence par rapport à la verticale [9].

La figure IV.2 montre le temps qui sépare le signal du plastique de celui du cristal pour des candidats aux coïncidences. Pour déterminer ce temps, on cherche l'instant qui correspond à la première valeur d'amplitude qui dépasse un seuil dans le ICs, que l'on considère comme le début du signal. Ensuite, dans une fenêtre de 50 ns de part et d'autre de cet instant, si l'amplitude dans le plastique dépasse un seuil de 6 mV, on considère qu'il y a coïncidence entre les scintillateurs. Les temps négatifs indiquent que la particule a traversé le plastique en premier, ce qui est le cas pour la majorité des candidats. Au vu de cet histogramme, une fenêtre de 15 ns avant le signal dans le cristal est suffisante, ce qui corrobore la discussion ci-dessus. Sur 3785 paires de fichiers, on a alors 1840 coïncidences, ce qui correspond à une proportion de 48,6% qui passent les critères de sélection. Pour comprendre cette proportion, il faut savoir que le cristal n'était pas complètement entouré de briques de plomb. Ainsi, l'oscilloscope peut s'être enclenché sur des signaux générés par d'autres particules de la radiation ambiante que les muons cosmiques. Par exemple des photons à haute énergie absorbés par le ICs n'atteindront pas le scintillateur de plastique et ne seront donc pas sélectionnés par coïncidences. De plus, il faut se méfier des fausses coïncidences, où des signaux sont générés dans les deux scintillateurs à l'intérieur de la fenêtre de temps utilisée, mais par deux particules différentes. C'est pourquoi on utilise une fenêtre de 15 ns, physiquement raisonnable et pas trop large.

IV.2 Temps de vie du muon

Avec les muons identifiés par coïncidence entre les signaux à la sortie du préamplificateur et ceux provenant du scintillateur de plastique, il est possible de déterminer le temps de vie du muon, qui peut se désintégrer par interaction faible [23] :



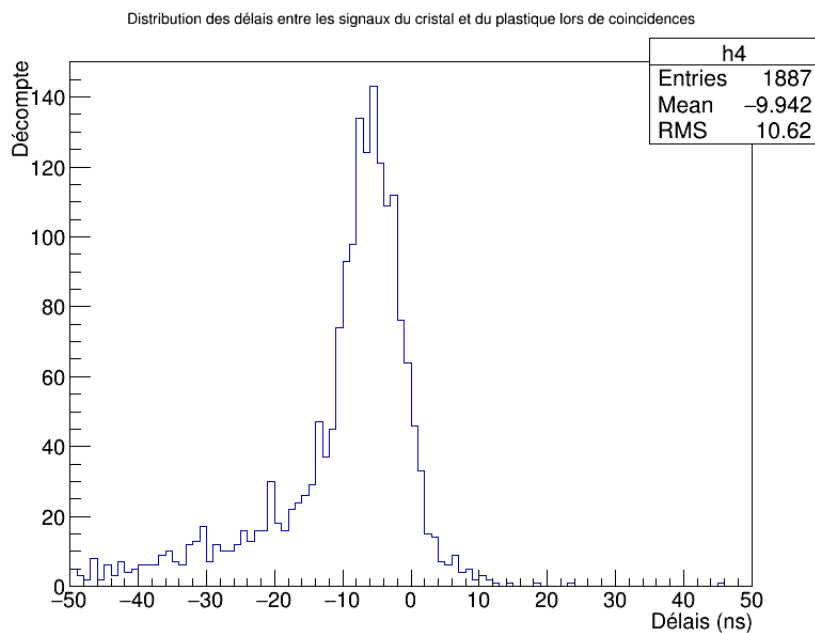


Figure IV.2 : Temps qui sépare les signaux du plastique et du ICs pour les candidats au passage de muons cosmiques.

Les électrons et positrons produits ne sortent pas du cristal et ne traversent donc pas le scintillateur de plastique. Ils perdent ainsi toute leur énergie dans le cristal, y générant un deuxième signal après celui du muon. On sélectionne donc ces électrons et ces positrons par des signaux dans le cristal seulement et qui viennent après une coïncidence indiquant le passage d'un muon. Un exemple est illustré à la figure IV.3, où une deuxième montée associée à un électron ou un positron se superpose à la première, associée à un muon qui s'est désintégré. Ces montées sont séparées d'environ $2 \mu\text{s}$. En faisant l'histogramme du nombre d'événements selon l'intervalle de temps Δt entre le signal du μ^\pm et celui de e^\pm , il est possible de déterminer le temps de vie du muon à partir de la décroissance exponentielle attendue [1] :

$$y(\Delta t) = A e^{-(\Delta t)/\tau_\mu} + B \quad (\text{IV.2})$$

où $y(\Delta t)$ est le décompte du nombre d'événements, A est une amplitude qui dépend de la grandeur de l'ensemble statistique utilisé, B est un décalage vertical dû au bruit et τ_μ est le temps de vie du muon, pour lequel on s'attend à une valeur d'environ $2,197 \mu\text{s}$.

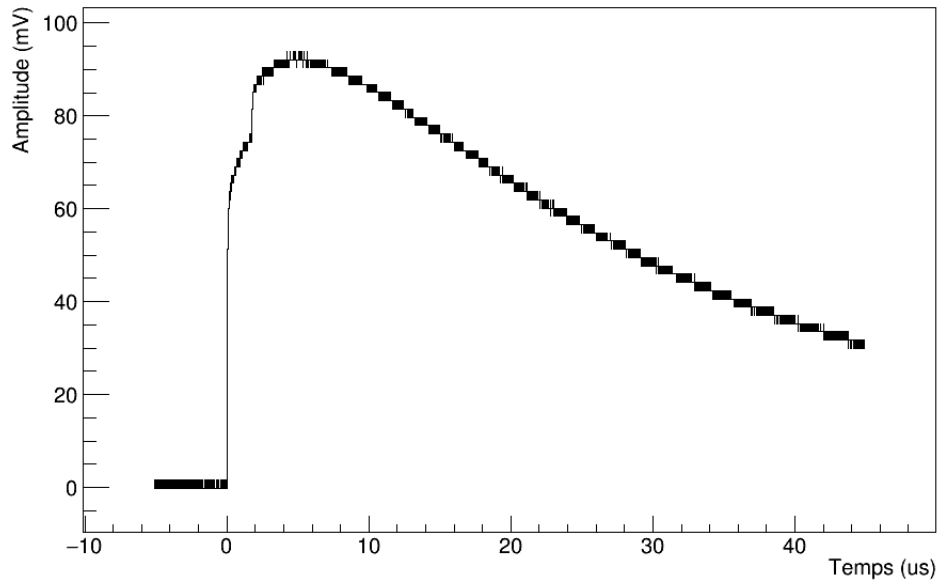
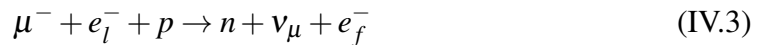


Figure IV.3 : Exemple d'empilement dû à la désintégration d'un muon selon la désintégration faible IV.1.

Un deuxième processus de moindre importance peut être confondu avec la désintégration du muon, soit la capture d'un muon négatif par un atome. En effet, un μ^- incident peut éjecter un e^- d'un atome pour prendre sa place autour du noyau. Comme le muon est environ 207 fois plus lourd que l'électron, il tourne beaucoup plus près du noyau, ce qui augmente la probabilité d'une absorption par un proton du noyau selon le mécanisme d'absorption suivant :



où e_l^- est un électron lié et e_f^- est un électron libre. Dans les deux processus, un muon est détruit et un électron libre est créé, la principale différence étant la modification d'un atome dans le mécanisme d'absorption, avec la charge de son noyau passant de Z à $(Z - 1)^*$. Au final, le temps de vie mesuré sera légèrement plus court que la vraie valeur τ_μ .

La distinction entre électron et positron ne peut se faire que si le positron s'annihile

avec l'un des électrons du cristal, auquel cas l'émission d'un photon génère un troisième signal dans le cristal seulement. Ainsi, on ne peut distinguer que les μ^+ et les e^+ si on observe deux signaux dans le cristal qui suivent une coïncidence avec le scintillateur de plastique.

Pour sélectionner les coïncidences où il semble y avoir eu la désintégration d'un muon, il faut d'abord s'affranchir autant que possible des fluctuations dues au bruit. Pour ce faire, sur un signal brut de $45 \mu\text{s}$, on utilise une fenêtre de 15000 points, ou $0,375 \mu\text{s}$, dont on fait la moyenne. Cette valeur devient l'amplitude d'un point du signal affranchi de bruit, associé à un temps centré sur la fenêtre. On répète l'opération en déplaçant la fenêtre point par point sur toute la longueur du signal brut pour obtenir une forme lissée de ce signal. La largeur de la fenêtre est un compromis; trop petite et la queue des signaux de faibles amplitudes ressemble à une fonction *step*, trop large et la montée du signal subit un trop grand étalement, qui déplace le début du signal plus tôt qu'en réalité. La figure IV.4 montre un exemple de signal lissé pour un muon désintégré.

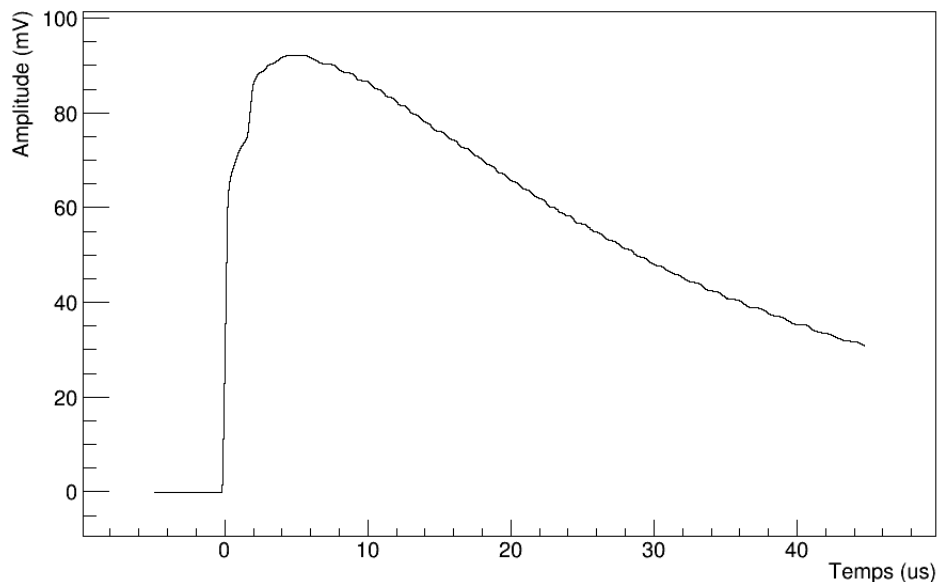


Figure IV.4 : Exemple de signal lissé, pour un empilement dû à la désintégration d'un muon selon la désintégration faible IV.1.

Lorsqu'un muon se désintègre, le signal enregistré correspond à une superposition de deux signaux, dont les débuts sont séparés par le temps de vie du muon dans le cristal. Pour déterminer ce temps, il faut trouver l'instant où commence le deuxième signal, dû à l'électron produit. Le premier point au-dessus du seuil de détection de 6 mV est considéré comme le début du premier signal, dû au muon. Normalement, à partir de ce point, la pente diminue graduellement à mesure qu'on s'approche de l'amplitude maximale. Cependant, cette pente augmente brusquement lorsqu'un deuxième signal se superpose. Ainsi, on déplace deux fenêtres consécutives de 20000 points de large, ou $0,5 \mu\text{s}$, jusqu'à ce que la pente qui traverse les extrémités de la deuxième fenêtre soit plus grande que la pente de la première fenêtre, sans dépasser l'instant d'occurrence de l'amplitude maximale de la superposition. On estime que le signal de l'électron débute à la fin de la deuxième fenêtre, qui commence à peine à traverser la montée abrupte du deuxième signal lorsque sa pente devient plus grande. La largeur de $0,5 \mu\text{s}$ est un compromis ; trop petite et on obtient en majorité de fausses détections dues à des fluctuations, trop grande et les désintégrations ne sont pas détectées.

IV.3 Masse du muon

Quant à la masse du muon, il est possible de la déterminer à partir de la théorie de Fermi en mesurant son temps de vie moyen τ_μ [23] :

$$\tau_\mu = \frac{192\pi^3\hbar^7}{G_F^2 m_\mu^5 c^4} \quad (\text{IV.4})$$

où il ne reste qu'à isoler m_μ , avec $G_F = 1,16 \times 10^{-5} (\hbar c)^3 \text{ GeV}^{-2}$. Cette théorie fut une première explication approximative de ce qui se produit lors d'une désintégration et qui date d'avant la découverte des bosons de jauge W^\pm et Z^0 . Avec $\tau = 2,197 \mu\text{s}$, on s'attend à $m_\mu = 105,66 \text{ MeV}$ environ.

Malheureusement, même s'il est possible de quantifier le temps de vie et la masse du muon à l'aide des photomultiplicateurs et du préamplificateur, cet objectif a dû être abandonné en raison de la trop grande quantité de données qu'il faudrait accumuler pour

avoir un ensemble statistique satisfaisant. En effet, sur 14 Go de données, 7 candidats à la désintégration de muon ont été identifiés. Ainsi, pour avoir au moins 1000 candidats, il faudrait accumuler environ 2 To de données, d'autant plus que l'oscilloscope LeCroy est limité à 80 Go.

Annexe V

Calcul d'incertitudes

Les valeurs finales des différents paramètres mesurés correspondent aux moyennes \bar{x} de leurs ensembles statistiques :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (\text{V.1})$$

où N est le nombre de mesures x_i d'un paramètre. L'incertitude, ou écart-type, σ d'une distribution est donnée par l'expression [22] :

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (\text{V.2})$$

L'incertitude est calculée plus efficacement durant l'exécution d'un code informatique avec la forme suivante :

$$\sigma = \sqrt{\frac{1}{N-1} \left(\sum_{i=1}^N (x_i)^2 - N\bar{x}^2 \right)} \quad (\text{V.3})$$

Enfin, l'incertitude $\bar{\sigma}$ sur la moyenne est obtenue avec l'expression [22] :

$$\bar{\sigma} = \frac{\sigma}{\sqrt{N}} \quad (\text{V.4})$$

Annexe VI

Coefficients pour les formes de référence

Le DSP fait l'ajustement de formes de référence, des ensembles de 32 points, sur les signaux pour en extraire l'amplitude et le temps d'occurrence. Chaque forme de référence est enregistrée sur le DSP sous la forme d'une matrice C de coefficients, de dimensions 3×32 telle que définie à la section 4.6.1.3, soient 32 coefficients pour chacun des 3 paramètres. Cette annexe présente ces coefficients, pour les décalages en temps de 0 à -9 ns pour chacune des trois méthodes étudiées, soient les formes moyennes expérimentales, les formes obtenues par l'ajustement de l'expression (5.1) pour les filtres actifs et la somme de 3 gaussiennes. Pour chacun des paramètres A , B et P , les 32 coefficients sont présentés de haut en bas dans l'ordre chronologique, de 0 à 310 ns. De plus, le décalage $t_0 = -B/A$ est évaluée en nanoseconde avec ces ensembles de coefficients.

VI.1 Générateur d'impulsions

VI.1.1 Formes moyennes expérimentales

Décalage de 0 ns			Décalage de -1 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.103882	-1.53921	0.0770133	-0.104365	-1.52621	0.077274
-0.103882	-1.53921	0.0770133	-0.104365	-1.52621	0.077274
-0.103882	-1.53921	0.0770133	-0.104365	-1.52621	0.077274
-0.103882	-1.53921	0.0770133	-0.104365	-1.52621	0.077274
-0.103882	-1.53921	0.0770133	-0.104365	-1.52621	0.077274
-0.103882	-1.53921	0.0770133	-0.104365	-1.52621	0.077274
-0.10213	-1.02301	0.0758823	-0.102341	-0.929303	0.0759743
-0.0983555	-0.407988	0.0738099	-0.0983247	-0.323457	0.0738105
-0.0915858	0.514271	0.0702253	-0.0911841	0.636154	0.0700477
-0.0807386	1.78622	0.0646325	-0.079871	1.93493	0.0642453

-0.0650648	3.32074	0.0567734	-0.0636778	3.483	0.0561631
-0.0443777	4.95933	0.0466839	-0.0424741	5.11744	0.0458619
-0.0192429	6.48167	0.0347686	-0.01691	6.61	0.0337853
0.00904338	7.66624	0.0217466	0.0116656	7.74874	0.0206663
0.0385563	8.3141	0.00859088	0.0412665	8.33304	0.00750386
0.0670477	8.31015	-0.00364828	0.069632	8.26265	-0.0046568
0.0923531	7.65622	-0.0140422	0.0946222	7.55105	-0.0149042
0.112596	6.41553	-0.0218308	0.114348	6.25059	-0.0224625
0.126431	4.72601	-0.0265372	0.127587	4.53914	-0.0269335
0.133129	2.75646	-0.0279721	0.133646	2.55985	-0.0281215
0.13273	0.726091	-0.0263128	0.132596	0.530121	-0.0262124
0.125833	-1.19857	-0.0219391	0.125134	-1.36429	-0.0216406
0.113394	-2.90647	-0.0153428	0.112225	-3.03933	-0.0148808
0.0968057	-4.25856	-0.00722445	0.0952647	-4.35744	-0.00663346
0.0773966	-5.25973	0.00184883	0.0756139	-5.31612	0.00251422
0.0565898	-5.86864	0.0112352	0.0546913	-5.88446	0.0119292
0.035568	-6.14192	0.020468	0.0336199	-6.13566	0.0211807
0.0152391	-6.14876	0.0292078	0.0133394	-6.11223	0.0298945
-0.00362077	-5.91919	0.0371432	-0.00541841	-5.866	0.037793
-0.0206325	-5.56253	0.0441914	-0.0222996	-5.50091	0.0447982
-0.0354926	-5.09755	0.0502357	-0.037012	-5.03132	0.0507947
-0.0481811	-4.60486	0.0553266	-0.0495506	-4.53891	0.0558391
Décalage de -2 ns			Décalage de -3 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.104847	-1.51245	0.0775291	-0.105317	-1.49799	0.0777804
-0.104847	-1.51245	0.0775291	-0.105317	-1.49799	0.0777804
-0.104847	-1.51245	0.0775291	-0.105317	-1.49799	0.0777804
-0.104847	-1.51245	0.0775291	-0.105317	-1.49799	0.0777804
-0.104847	-1.51245	0.0775291	-0.105317	-1.49799	0.0777804

-0.104847	-1.51245	0.0775291	-0.105317	-1.49799	0.0777804
-0.102532	-0.828698	0.0760502	-0.10271	-0.727028	0.0761246
-0.0982472	-0.231184	0.0737828	-0.0981384	-0.133606	0.073742
-0.0907272	0.760945	0.0698428	-0.0902081	0.894989	0.0696092
-0.0789493	2.08324	0.0638346	-0.0779287	2.2423	0.0633784
-0.0622127	3.65161	0.0555151	-0.0606356	3.82404	0.0548205
-0.0404734	5.28005	0.0449959	-0.0383617	5.44194	0.0440858
-0.014446	6.74686	0.032741	-0.0119	6.87454	0.0316691
0.0143838	7.81946	0.0195544	0.0171746	7.88652	0.0184124
0.0440448	8.34213	0.00639552	0.0468944	8.35011	0.00525372
0.0722625	8.20712	-0.00567892	0.0749354	8.14259	-0.00671872
0.0968973	7.43265	-0.015759	0.0991933	7.3078	-0.0166258
0.116114	6.087	-0.0231012	0.117861	5.91183	-0.0237321
0.128736	4.34572	-0.0273215	0.129829	4.14182	-0.0276869
0.134132	2.35618	-0.0282511	0.134536	2.14562	-0.0283495
0.132438	0.337201	-0.0261027	0.132181	0.139299	-0.0259537
0.124374	-1.54015	-0.0213061	0.123523	-1.71263	-0.0209411
0.110973	-3.1772	-0.014377	0.109645	-3.31031	-0.0138476
0.0936396	-4.45566	-0.00600363	0.0919323	-4.55477	-0.00534013
0.0737673	-5.3654	0.00320556	0.0718258	-5.41782	0.00393963
0.0526946	-5.9092	0.0126754	0.0506762	-5.91208	0.0134167
0.0316144	-6.11648	0.0219121	0.0295597	-6.105	0.0226725
0.0113779	-6.07602	0.030611	0.00937206	-6.03803	0.0313485
-0.00727409	-5.81088	0.0384687	-0.00914685	-5.74757	0.0391499
-0.0240044	-5.43267	0.0454182	-0.0257414	-5.36876	0.0460593
-0.0385587	-4.96132	0.0513638	-0.0401223	-4.88676	0.0519414
-0.0509419	-4.47061	0.0563596	-0.052346	-4.40111	0.056889
Décalage de -4 ns			Décalage de -5 ns		
C_A	C_B	C_P	C_A	C_B	C_P

-0.105793	-1.48531	0.0780359	-0.106254	-1.472	0.0782894
-0.105793	-1.48531	0.0780359	-0.106254	-1.472	0.0782894
-0.105793	-1.48531	0.0780359	-0.106254	-1.472	0.0782894
-0.105793	-1.48531	0.0780359	-0.106254	-1.472	0.0782894
-0.105793	-1.48531	0.0780359	-0.106254	-1.472	0.0782894
-0.105793	-1.48531	0.0780359	-0.106254	-1.472	0.0782894
-0.102843	-0.611736	0.0761716	-0.102941	-0.489011	0.0762055
-0.0979815	-0.0374499	0.0736811	-0.097773	0.0629631	0.0736014
-0.0896274	1.03133	0.0693466	-0.0890134	1.16261	0.0690793
-0.0768466	2.39959	0.0628946	-0.0757023	2.56547	0.0623842
-0.0590228	3.9873	0.0541125	-0.0573329	4.15452	0.0533733
-0.0362578	5.58985	0.0431818	-0.034054	5.74414	0.0422334
-0.00938566	6.99444	0.0306068	-0.00678215	7.11088	0.0295089
0.0199267	7.95134	0.0172768	0.0227497	8.01058	0.016112
0.0496844	8.3528	0.00412776	0.0525016	8.34252	0.00299353
0.0775245	8.0726	-0.00773284	0.0801196	7.99645	-0.00875335
0.101403	7.18468	-0.0174701	0.103591	7.05344	-0.0183092
0.119536	5.74351	-0.0243483	0.121162	5.56793	-0.0249498
0.130855	3.94235	-0.0280353	0.131789	3.73076	-0.028348
0.134891	1.94308	-0.0284373	0.135144	1.73807	-0.0284905
0.131894	-0.0502819	-0.0257994	0.13152	-0.235227	-0.0256201
0.122654	-1.88029	-0.0205692	0.121677	-2.05375	-0.0201548
0.108299	-3.44249	-0.0133055	0.10688	-3.56383	-0.0127449
0.0902351	-4.64607	-0.00467785	0.0884909	-4.72934	-0.00400451
0.0699306	-5.46065	0.00465782	0.067991	-5.50242	0.0053929
0.0486572	-5.93059	0.014181	0.0466012	-5.94684	0.0149598
0.0275112	-6.09407	0.0234428	0.0254704	-6.06976	0.0242027
0.00742768	-5.98683	0.0320628	0.00547866	-5.93922	0.0327849
-0.0109766	-5.69299	0.0398304	-0.0128061	-5.6369	0.0405138

-0.0274237	-5.30461	0.0466873	-0.0290918	-5.23634	0.0473113
-0.0416354	-4.81659	0.0525098	-0.0431336	-4.74625	0.0530769
-0.0536712	-4.32632	0.057389	-0.0550085	-4.25942	0.0579052
Décalage de -6 ns			Décalage de -7 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.106688	-1.45894	0.0785265	-0.107118	-1.44556	0.0787589
-0.106688	-1.45894	0.0785265	-0.107118	-1.44556	0.0787589
-0.106688	-1.45894	0.0785265	-0.107118	-1.44556	0.0787589
-0.106688	-1.45894	0.0785265	-0.107118	-1.44556	0.0787589
-0.106688	-1.45894	0.0785265	-0.107118	-1.44556	0.0787589
-0.106688	-1.45894	0.0785265	-0.107118	-1.44556	0.0787589
-0.103013	-0.366587	0.0762261	-0.103054	-0.234906	0.0762269
-0.0975205	0.167573	0.0734963	-0.0972272	0.274015	0.0733712
-0.0883486	1.29526	0.0687858	-0.0876256	1.43293	0.0684633
-0.0745424	2.72035	0.0618709	-0.0733066	2.87997	0.0613216
-0.0556603	4.31139	0.0526445	-0.0538822	4.47629	0.0518648
-0.0318891	5.8912	0.0413019	-0.0296278	6.0347	0.0403308
-0.00424785	7.21745	0.028442	-0.00163507	7.31801	0.0273443
0.0254959	8.06699	0.0149762	0.0282922	8.11431	0.0138234
0.0552159	8.32421	0.00190263	0.0579683	8.30163	0.000796067
0.0825988	7.91672	-0.00972739	0.0850936	7.82997	-0.010706
0.105642	6.91671	-0.0190907	0.107676	6.77011	-0.0198607
0.122665	5.39664	-0.0255038	0.124113	5.21511	-0.0260299
0.132626	3.52767	-0.0286254	0.133428	3.33066	-0.028893
0.135329	1.54272	-0.0285195	0.135467	1.34446	-0.0285267
0.131082	-0.418221	-0.0254108	0.130599	-0.599628	-0.0251826
0.120706	-2.20567	-0.0197522	0.119676	-2.36102	-0.01932
0.105496	-3.67657	-0.012198	0.104044	-3.79102	-0.0116174
0.0867748	-4.81477	-0.00333333	0.0850179	-4.89121	-0.00264709

0.0660802	-5.54403	0.00612469	0.0641382	-5.58239	0.00687171
0.0446391	-5.9444	0.0156949	0.0426339	-5.94499	0.0164549
0.0235122	-6.04978	0.0249386	0.0215101	-6.02982	0.0256973
0.00360262	-5.89511	0.0334858	0.00167966	-5.85187	0.0342113
-0.0145687	-5.58361	0.041177	-0.0163725	-5.52883	0.0418603
-0.0306813	-5.16562	0.0479054	-0.0323048	-5.09305	0.0485159
-0.0445766	-4.68404	0.0536298	-0.0460184	-4.61186	0.0541778
-0.0562883	-4.19281	0.0583997	-0.0575724	-4.1282	0.0588987
Décalage de -8 ns			Décalage de -9 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.107528	-1.43288	0.0789789	-0.107948	-1.42077	0.0792077
-0.107528	-1.43288	0.0789789	-0.107948	-1.42077	0.0792077
-0.107528	-1.43288	0.0789789	-0.107948	-1.42077	0.0792077
-0.107528	-1.43288	0.0789789	-0.107948	-1.42077	0.0792077
-0.107528	-1.43288	0.0789789	-0.107948	-1.42077	0.0792077
-0.107528	-1.43288	0.0789789	-0.107948	-1.42077	0.0792077
-0.103085	-0.106732	0.076224	-0.103075	0.0367602	0.0761995
-0.0969098	0.383267	0.0732309	-0.0965464	0.491681	0.0730755
-0.0868929	1.56516	0.0681368	-0.0860844	1.70558	0.0677762
-0.0720858	3.03131	0.0607802	-0.0707639	3.18918	0.0601937
-0.052125	4.63856	0.0510914	-0.0502857	4.79434	0.0502874
-0.0274288	6.16483	0.0393896	-0.0251301	6.29996	0.0384006
0.000889386	7.41053	0.0262833	0.00351405	7.50282	0.025175
0.0309861	8.15733	0.0127105	0.0337225	8.18351	0.011583
0.0605915	8.27135	-0.000257208	0.0632465	8.23919	-0.00133204
0.0874546	7.74271	-0.0116332	0.089821	7.64871	-0.012569
0.109602	6.63229	-0.0205952	0.111509	6.48894	-0.0213294
0.125468	5.03888	-0.0265225	0.126777	4.85641	-0.0270025
0.13414	3.13628	-0.0291248	0.134801	2.94243	-0.029345

0.135546	1.15687	-0.0285142	0.135559	0.966014	-0.028479
0.130053	-0.781528	-0.0249233	0.129447	-0.96005	-0.0246465
0.118628	-2.5024	-0.0188828	0.11754	-2.6366	-0.0184368
0.102596	-3.89895	-0.0110355	0.101089	-4.01246	-0.0104262
0.0833004	-4.95791	-0.00197525	0.0815361	-5.0205	-0.00128504
0.0622777	-5.60924	0.00758553	0.0603441	-5.64458	0.00833881
0.0406954	-5.95134	0.017199	0.0387307	-5.94433	0.0179502
0.0196135	-5.99937	0.0264128	0.0176943	-5.96951	0.0271436
-0.000109827	-5.80205	0.0348831	-0.00193116	-5.75628	0.0355771
-0.018047	-5.47681	0.0424963	-0.0197463	-5.42408	0.0431488
-0.0338264	-5.02967	0.0490941	-0.0353467	-4.9601	0.0496744
-0.0473728	-4.54508	0.054696	-0.048737	-4.48339	0.055228
-0.0587895	-4.07105	0.0593767	-0.0599962	-4.00904	0.0598532

Tableau VI.I: Table de coefficients pour les formes moyennes expérimentales, pour les décalages dans le temps de 0 à -9 ns, pour des signaux produits par un générateur d'impulsions.

VI.1.2 Filtres actifs

Décalage de 0 ns			Décalage de -1 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.101793	-2.02552	0.0778084	-0.10245	-2.01424	0.0781696
-0.101793	-2.02552	0.0778084	-0.10245	-2.01424	0.0781696
-0.101793	-2.02552	0.0778084	-0.10245	-2.01424	0.0781696
-0.101793	-2.02552	0.0778084	-0.10245	-2.01424	0.0781696
-0.101793	-2.02552	0.0778084	-0.10245	-2.01424	0.0781696
-0.101793	-2.02552	0.0778084	-0.10245	-2.01424	0.0781696
-0.101793	-2.02552	0.0778084	-0.101846	-1.84082	0.0777217
-0.0965924	-0.525447	0.0739234	-0.0968832	-0.55477	0.0741876
-0.0926165	-0.581389	0.0722506	-0.0927519	-0.536975	0.0723629
-0.0861622	0.164147	0.0686328	-0.0859036	0.287582	0.0684945
-0.075227	1.64317	0.0622704	-0.074412	1.82247	0.0618428
-0.0588257	3.57105	0.0530412	-0.0573993	3.7703	0.0523396
-0.0371207	5.58018	0.0414122	-0.0351355	5.76231	0.0404942
-0.0112777	7.3193	0.0282707	-0.00887356	7.45406	0.0272198
0.0168419	8.51872	0.014719	0.0194698	8.58693	0.0136285
0.0450508	9.02176	0.00187954	0.0476867	9.0165	0.000838688
0.0711872	8.7882	-0.00926188	0.0736252	8.71347	-0.0101779
0.0933931	7.87759	-0.0179596	0.0954609	7.74566	-0.0186957
0.110298	6.42154	-0.023758	0.11187	6.24967	-0.0242812
0.121101	4.59258	-0.0264945	0.122107	4.39988	-0.0267931
0.125573	2.57521	-0.0262678	0.125992	2.38003	-0.026348
0.123987	0.542597	-0.0233837	0.123842	0.360589	-0.0232659
0.117013	-1.35973	-0.0182915	0.116364	-1.51666	-0.0180053
0.105593	-3.02291	-0.0115193	0.104523	-3.14705	-0.0110993
0.0908152	-4.37761	-0.00361599	0.0894178	-4.46531	-0.00309796
0.0738008	-5.39204	0.00489521	0.0721733	-5.44316	0.00547739

0.0556126	-6.0666	0.0135515	0.0538469	-6.0838	0.0141678
0.0371884	-6.42681	0.0219715	0.0353658	-6.41468	0.0225973
0.0192997	-6.51548	0.0298652	0.0174871	-6.47977	0.0304814
0.00253241	-6.38534	0.037034	0.000781138	-6.33219	0.0376272
-0.0127123	-6.09254	0.0433637	-0.0143666	-6.02795	0.0439257
-0.0262041	-5.69148	0.0488124	-0.0277401	-5.62085	0.0493395
Décalage de -2 ns			Décalage de -3 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.1031	-2.00278	0.0785268	-0.103741	-1.99117	0.07888
-0.1031	-2.00278	0.0785268	-0.103741	-1.99117	0.07888
-0.1031	-2.00278	0.0785268	-0.103741	-1.99117	0.07888
-0.1031	-2.00278	0.0785268	-0.103741	-1.99117	0.07888
-0.1031	-2.00278	0.0785268	-0.103741	-1.99117	0.07888
-0.1031	-2.00278	0.0785268	-0.103741	-1.99117	0.07888
-0.101914	-1.66344	0.0776527	-0.101993	-1.49301	0.0776006
-0.0971618	-0.577917	0.0744373	-0.0974257	-0.594467	0.0746711
-0.0928498	-0.484243	0.0724497	-0.0929078	-0.42326	0.0725101
-0.0855871	0.41775	0.0683253	-0.0852115	0.554341	0.0681248
-0.0735288	2.00507	0.0613838	-0.0725772	2.1906	0.0608935
-0.0559056	3.96894	0.0516104	-0.0543453	4.16664	0.0508537
-0.0330939	5.94041	0.039555	-0.0309975	6.11424	0.0385952
-0.00643126	7.58239	0.0261557	-0.00395269	7.7042	0.0250791
0.0221139	8.64759	0.0125333	0.0247722	8.70074	0.0114339
0.0503164	9.00374	-0.000199016	0.052938	8.98363	-0.0012331
0.076037	8.63225	-0.0110842	0.0784211	8.54476	-0.0119804
0.0974867	7.60886	-0.0194172	0.0994693	7.46745	-0.0201237
0.11339	6.07484	-0.0247866	0.114857	5.8973	-0.0252738
0.123055	4.20612	-0.0270721	0.123945	4.01153	-0.0273316
0.126352	2.18545	-0.0264085	0.126655	1.99166	-0.0264493

0.123644	0.180506	-0.0231293	0.123393	0.00246533	-0.0229739
0.115669	-1.67077	-0.0177021	0.114928	-1.82201	-0.0173817
0.103415	-3.26791	-0.0106644	0.10227	-3.38547	-0.0102148
0.0879927	-4.54962	-0.00256754	0.0865406	-4.6306	-0.00202483
0.070528	-5.4911	0.00606943	0.0688655	-5.53595	0.00667123
0.0520727	-6.09824	0.0147917	0.0502903	-6.11003	0.0154228
0.0335426	-6.40036	0.0232283	0.0317191	-6.38393	0.0238646
0.0156804	-6.44246	0.0311009	0.0138798	-6.40366	0.0317238
-0.00095927	-6.27804	0.0382222	-0.00268879	-6.223	0.0388189
-0.0160066	-5.9629	0.0444883	-0.0176324	-5.89748	0.0450514
-0.0292598	-5.55023	0.0498662	-0.0307632	-5.47968	0.0503925
Décalage de -4 ns			Décalage de -5 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.104375	-1.97942	0.0792292	-0.105001	-1.96757	0.0795746
-0.104375	-1.97942	0.0792292	-0.105001	-1.96757	0.0795746
-0.104375	-1.97942	0.0792292	-0.105001	-1.96757	0.0795746
-0.104375	-1.97942	0.0792292	-0.105001	-1.96757	0.0795746
-0.104375	-1.97942	0.0792292	-0.105001	-1.96757	0.0795746
-0.104375	-1.97942	0.0792292	-0.105001	-1.96757	0.0795746
-0.102085	-1.32911	0.077564	-0.102188	-1.1712	0.0775416
-0.0976723	-0.604063	0.0748879	-0.0978991	-0.606409	0.0750867
-0.0929237	-0.354123	0.0725437	-0.0928955	-0.27696	0.0725499
-0.0847753	0.697032	0.0678929	-0.0842773	0.845494	0.0676295
-0.0715568	2.37866	0.0603721	-0.0704676	2.56887	0.0598198
-0.0527192	4.36305	0.0500702	-0.0510281	4.55784	0.0492601
-0.0288477	6.28357	0.0376152	-0.026646	6.44818	0.0366156
-0.00143976	7.8194	0.0239906	0.00110556	7.92792	0.0228907
0.0274427	8.74641	0.010331	0.0301234	8.78467	0.00922513
0.0555498	8.95633	-0.00226307	0.0581499	8.922	-0.00328841

0.080776	8.45123	-0.0128662	0.0831002	8.35187	-0.013741
0.101408	7.32169	-0.020815	0.103301	7.17181	-0.0214907
0.11627	5.71731	-0.0257428	0.117629	5.5351	-0.0261933
0.124777	3.81632	-0.0275713	0.125551	3.62068	-0.0277912
0.126901	1.79883	-0.0264703	0.127089	1.6071	-0.0264714
0.123088	-0.173424	-0.0227997	0.122732	-0.347064	-0.0226068
0.114142	-1.97033	-0.0170444	0.11331	-2.11567	-0.0166902
0.101089	-3.49975	-0.00975045	0.0998728	-3.61073	-0.00927151
0.0850621	-4.70827	-0.00146993	0.0835581	-4.78268	-0.000902954
0.0671865	-5.57776	0.00728269	0.0654914	-5.61661	0.00790366
0.0485003	-6.11925	0.0160612	0.0467031	-6.12598	0.0167068
0.0298957	-6.36552	0.0245059	0.0280726	-6.34519	0.0251521
0.0120855	-6.36348	0.0323498	0.0102976	-6.322	0.032979
-0.0044074	-6.16715	0.0394173	-0.00611506	-6.1106	0.0400173
-0.0192441	-5.83179	0.0456149	-0.0208418	-5.7659	0.0461788
-0.0322507	-5.40929	0.0509182	-0.0337224	-5.33912	0.0514434
Décalage de -6 ns			Décalage de -7 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.105619	-1.95563	0.079916	-0.10623	-1.94362	0.0802536
-0.105619	-1.95563	0.079916	-0.10623	-1.94362	0.0802536
-0.105619	-1.95563	0.079916	-0.10623	-1.94362	0.0802536
-0.105619	-1.95563	0.079916	-0.10623	-1.94362	0.0802536
-0.105619	-1.95563	0.079916	-0.10623	-1.94362	0.0802536
-0.105619	-1.95563	0.079916	-0.10623	-1.94362	0.0802536
-0.1023	-1.01873	0.0775319	-0.10242	-0.871057	0.0775332
-0.0981036	-0.601265	0.0752665	-0.0982833	-0.588441	0.0754262
-0.092821	-0.191922	0.0725282	-0.0926983	-0.0991884	0.0724781
-0.0837165	0.999386	0.0673346	-0.0830917	1.15836	0.0670079
-0.0693094	2.76084	0.0592368	-0.0680824	2.95421	0.0586234

-0.102546	-0.72751	0.0775437	-0.102675	-0.587377	0.0775617
-0.0984358	-0.567796	0.075565	-0.0985587	-0.539231	0.0756821
-0.0925255	0.00104222	0.0723992	-0.0923008	0.108549	0.0722911
-0.0824022	1.32207	0.0666495	-0.0816471	1.49015	0.0662594
-0.0667867	3.1486	0.0579798	-0.0654226	3.34363	0.0573062
-0.0455744	5.1293	0.0466753	-0.0436334	5.31445	0.0457638
-0.0197466	6.91172	0.0335053	-0.0173543	7.05549	0.0324334
0.00891603	8.21268	0.0195293	0.0115709	8.2938	0.0183904
0.0382062	8.8555	0.00589559	0.0409073	8.86465	0.00478392
0.0658621	8.77828	-0.00633129	0.0683974	8.71727	-0.0073326
0.0898749	8.02094	-0.0162957	0.0920624	7.90032	-0.0171223
0.108702	6.69978	-0.0234213	0.110406	6.53569	-0.0240313
0.121376	4.9773	-0.0274315	0.122514	4.78831	-0.0278056
0.127525	3.03311	-0.0283304	0.128067	2.83759	-0.0284695
0.127316	1.03996	-0.0263553	0.12728	0.854004	-0.0262765
0.121354	-0.853607	-0.0219156	0.120795	-1.0174	-0.0216479
0.110556	-2.53349	-0.0155268	0.109553	-2.66658	-0.0151057
0.0960185	-3.92401	-0.00774821	0.0946678	-4.02188	-0.00721207
0.0788994	-4.98675	0.000868953	0.0773002	-5.0485	0.0014827
0.0603168	-5.71601	0.00982216	0.0585641	-5.74361	0.0104796
0.0412735	-6.13206	0.0186844	0.0394526	-6.12959	0.0193567
0.022609	-6.27364	0.0271191	0.020791	-6.24648	0.0277835
0.00497442	-6.1906	0.0348835	0.00321434	-6.14471	0.0355235
-0.0111716	-5.93738	0.0418253	-0.0128347	-5.87867	0.0424301
-0.0255511	-5.56766	0.0478714	-0.0270931	-5.50156	0.0484354
-0.038044	-5.13044	0.0530147	-0.0394538	-5.06164	0.0535365

Tableau VI.II: Table de coefficients pour l'équation (5.1) pour les filtres actifs, pour les décalages dans le temps de 0 à -9 ns, pour des signaux produits par un générateur d'impulsions.

VI.1.3 Somme de 3 gaussiennes

Décalage de 0 ns			Décalage de -1 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.100734	-1.63468	0.0755352	-0.101232	-1.62134	0.0758009
-0.100734	-1.63468	0.0755352	-0.101232	-1.62134	0.0758009
-0.100734	-1.63468	0.0755352	-0.101232	-1.62134	0.0758009
-0.100734	-1.63468	0.0755352	-0.101232	-1.62134	0.0758009
-0.100734	-1.63468	0.0755352	-0.101232	-1.62134	0.0758009
-0.100734	-1.63468	0.0755352	-0.101232	-1.62134	0.0758009
-0.0983067	-0.923385	0.0739	-0.0986253	-0.857692	0.0740549
-0.0959028	-0.873874	0.0728343	-0.0960611	-0.813877	0.0729241
-0.0914994	-0.261554	0.070441	-0.0914072	-0.174398	0.0704075
-0.0841619	0.669985	0.0665281	-0.0837086	0.789377	0.0663224
-0.0728315	1.95553	0.0606152	-0.0719089	2.10586	0.0601945
-0.0566503	3.54836	0.0523764	-0.0551855	3.71832	0.0517197
-0.0353186	5.29138	0.041817	-0.0333098	5.45876	0.0409397
-0.00940974	6.92463	0.029401	-0.00695029	7.05993	0.0283615
0.0195167	8.14157	0.0160519	0.02224	8.21626	0.014944
0.0491109	8.68128	0.00299122	0.0518515	8.67742	0.00192332
0.0767122	8.41739	-0.00854106	0.0792175	8.33412	-0.00947187
0.0999355	7.39837	-0.0175701	0.102	7.25158	-0.0182977
0.117127	5.81692	-0.0235545	0.118623	5.63227	-0.0240496
0.127524	3.92629	-0.0263835	0.128402	3.72951	-0.0266453
0.131119	1.95161	-0.0262445	0.131386	1.76146	-0.0262868
0.128379	0.0419073	-0.0234622	0.128073	-0.131285	-0.0233032
0.120003	-1.72305	-0.0184022	0.11918	-1.87413	-0.0180621
0.106826	-3.29328	-0.0114621	0.105556	-3.41791	-0.010967
0.0898352	-4.61357	-0.00310902	0.0882081	-4.70616	-0.00249457
0.0702007	-5.61766	0.00610256	0.0683278	-5.67257	0.00679186

0.0492326	-6.25004	0.0155678	0.0472361	-6.26457	0.0162848
0.0282403	-6.49526	0.0247159	0.0262368	-6.47191	0.0254193
0.00835018	-6.39316	0.0331008	0.00643365	-6.33977	0.0337618
-0.00964189	-6.03116	0.0404575	-0.0114111	-5.95891	0.0410621
-0.0253296	-5.51941	0.0467061	-0.0269258	-5.43976	0.0472533
-0.0386578	-4.96173	0.0519109	-0.0400835	-4.88388	0.052408
Décalage de -2 ns			Décalage de -3 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.101726	-1.6081	0.0760644	-0.102216	-1.59495	0.0763256
-0.101726	-1.6081	0.0760644	-0.102216	-1.59495	0.0763256
-0.101726	-1.6081	0.0760644	-0.102216	-1.59495	0.0763256
-0.101726	-1.6081	0.0760644	-0.102216	-1.59495	0.0763256
-0.101726	-1.6081	0.0760644	-0.102216	-1.59495	0.0763256
-0.101726	-1.6081	0.0760644	-0.102216	-1.59495	0.0763256
-0.0989279	-0.788669	0.0742012	-0.0992137	-0.716129	0.0743385
-0.0961944	-0.751664	0.073002	-0.0963015	-0.687163	0.0730674
-0.0912796	-0.0843848	0.0703574	-0.0911152	0.00853004	0.0702902
-0.0832084	0.911824	0.0660955	-0.0826602	1.0373	0.0658466
-0.0709299	2.2586	0.059749	-0.0698934	2.41364	0.0592782
-0.05366	3.889	0.0510371	-0.0520734	4.06014	0.0503288
-0.0312444	5.62421	0.0400394	-0.0291233	5.78745	0.0391164
-0.0044484	7.19042	0.0273056	-0.00190578	7.31581	0.0262339
0.0249837	8.2839	0.0138288	0.0277456	8.34434	0.012707
0.054587	8.66578	0.000857396	0.057315	8.64636	-0.00020568
0.0816943	8.24409	-0.0103931	0.0841406	8.14744	-0.0113039
0.104019	7.10033	-0.0190109	0.105991	6.94485	-0.0197089
0.120064	5.44579	-0.0245278	0.12145	5.2577	-0.0249886
0.129222	3.53301	-0.0268892	0.129985	3.33694	-0.0271148
0.131597	1.57279	-0.0263108	0.131751	1.38566	-0.0263161

0.127715	-0.302516	-0.0231256	0.127304	-0.471767	-0.0229293
0.118309	-2.02302	-0.0177039	0.117391	-2.16968	-0.0173273
0.104245	-3.54001	-0.0104552	0.102893	-3.65952	-0.0099268
0.0865496	-4.79576	-0.00186615	0.0848604	-4.88232	-0.00122396
0.0664348	-5.72418	0.00749155	0.0645228	-5.77244	0.00820133
0.0452314	-6.27589	0.0170085	0.0432197	-6.284	0.0177385
0.0242359	-6.44597	0.0261263	0.0222383	-6.41751	0.0268364
0.00452762	-6.28484	0.0344242	0.00263259	-6.22842	0.0350876
-0.0131652	-5.88626	0.041667	-0.0149039	-5.81328	0.0422718
-0.0285054	-5.3607	0.0478004	-0.0300683	-5.28228	0.0483471
-0.0414931	-4.80729	0.052905	-0.0428867	-4.73193	0.0534016
Décalage de -4 ns			Décalage de -5 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.1027	-1.5819	0.0765845	-0.103178	-1.56893	0.0768407
-0.1027	-1.5819	0.0765845	-0.103178	-1.56893	0.0768407
-0.1027	-1.5819	0.0765845	-0.103178	-1.56893	0.0768407
-0.1027	-1.5819	0.0765845	-0.103178	-1.56893	0.0768407
-0.1027	-1.5819	0.0765845	-0.103178	-1.56893	0.0768407
-0.1027	-1.5819	0.0765845	-0.103178	-1.56893	0.0768407
-0.0994819	-0.639878	0.0744665	-0.0997318	-0.559715	0.0745847
-0.0963812	-0.620303	0.0731198	-0.0964325	-0.551012	0.0731586
-0.0909128	0.104387	0.0702052	-0.0906709	0.203221	0.0701019
-0.0820625	1.16579	0.0655754	-0.0814143	1.29723	0.0652812
-0.0687987	2.57083	0.0587817	-0.067645	2.73004	0.0582594
-0.0504259	4.23153	0.0495945	-0.0487173	4.40289	0.0488344
-0.0269474	5.94817	0.0381712	-0.0247176	6.10608	0.0372042
0.000675767	7.43583	0.0251473	0.00329436	7.55022	0.0240464
0.0305231	8.3974	0.0115796	0.033314	8.44296	0.0104474
0.0600331	8.61918	-0.00126504	0.0627388	8.58427	-0.00231978

0.0865543	8.04436	-0.0122036	0.0889336	7.935	-0.0130915
0.107915	6.78536	-0.0203913	0.109789	6.6221	-0.0210573
0.122779	5.06821	-0.0254315	0.124051	4.87751	-0.0258561
0.130689	3.1414	-0.0273217	0.131334	2.94653	-0.0275096
0.131849	1.20011	-0.0263024	0.13189	1.0162	-0.0262693
0.12684	-0.63902	-0.022714	0.126323	-0.804251	-0.0224793
0.116425	-2.31409	-0.0169324	0.115412	-2.45621	-0.0165191
0.101502	-3.77639	-0.00938179	0.100072	-3.89055	-0.00882028
0.0831415	-4.96576	-0.000568227	0.0813935	-5.04601	0.000100797
0.0625926	-5.81732	0.00892085	0.0606453	-5.85876	0.00964977
0.0412016	-6.28891	0.0184745	0.0391783	-6.29063	0.0192161
0.0202448	-6.38655	0.0275495	0.0182561	-6.35316	0.028265
0.000749055	-6.17059	0.0357519	-0.00112248	-6.11142	0.0364166
-0.0166271	-5.74002	0.0428764	-0.0183344	-5.66656	0.0434804
-0.0316144	-5.20452	0.0488932	-0.0331437	-5.12746	0.0494384
-0.0442644	-4.65781	0.0538976	-0.0456262	-4.58493	0.0543927

Décalage de -6 ns			Décalage de -7 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.103651	-1.55604	0.0770943	-0.104118	-1.54323	0.0773449
-0.103651	-1.55604	0.0770943	-0.104118	-1.54323	0.0773449
-0.103651	-1.55604	0.0770943	-0.104118	-1.54323	0.0773449
-0.103651	-1.55604	0.0770943	-0.104118	-1.54323	0.0773449
-0.103651	-1.55604	0.0770943	-0.104118	-1.54323	0.0773449
-0.103651	-1.55604	0.0770943	-0.104118	-1.54323	0.0773449
-0.0999625	-0.475435	0.0746926	-0.100173	-0.386825	0.0747899
-0.096454	-0.479223	0.0731833	-0.0964446	-0.404873	0.0731931
-0.0903882	0.30506	0.0699795	-0.0900635	0.409924	0.0698376
-0.0807142	1.43158	0.0649635	-0.0799611	1.56878	0.0646218
-0.0664315	2.89109	0.0577108	-0.0651576	3.05382	0.0571358

-0.100363	-0.293667	0.074876	-0.100532	-0.195739	0.0749506
-0.0964029	-0.327898	0.0731876	-0.0963277	-0.248241	0.0731662
-0.0896954	0.517826	0.0696755	-0.0892824	0.62877	0.0694925
-0.0791539	1.70874	0.0642557	-0.0782914	1.85138	0.0638646
-0.0638227	3.21803	0.056534	-0.0624263	3.38352	0.0559054
-0.0432266	4.91421	0.0464	-0.0412758	5.08278	0.0455377
-0.0177167	6.5598	0.034177	-0.0152835	6.70334	0.0331277
0.0113521	7.85701	0.020666	0.0140981	7.94634	0.0195163
0.0417412	8.53345	0.00703222	0.04456	8.54789	0.00589082
0.0707576	8.4337	-0.00544734	0.0733897	8.36849	-0.00647458
0.0958452	7.57121	-0.0156767	0.0980674	7.4387	-0.0165097
0.115101	6.11203	-0.0229524	0.116764	5.93601	-0.0235478
0.127519	4.30005	-0.0270155	0.128556	4.10639	-0.0273624
0.132917	2.36692	-0.0279547	0.133326	2.17573	-0.0280624
0.131674	0.474675	-0.0260506	0.131488	0.297713	-0.0259368
0.124459	-1.28755	-0.0216571	0.123733	-1.44442	-0.0213431
0.112094	-2.86832	-0.0151673	0.110897	-3.00076	-0.0146795
0.0955512	-4.21608	-0.00703809	0.0939703	-4.31868	-0.00641203
0.0759843	-5.26699	0.00218476	0.0741294	-5.33383	0.00290397
0.0547105	-5.96217	0.0118891	0.0527048	-5.98958	0.0126518
0.0330864	-6.27679	0.0214702	0.0310517	-6.2659	0.02223
0.012327	-6.23892	0.0304227	0.0103655	-6.19634	0.0311445
-0.00666004	-5.92653	0.0384104	-0.00847852	-5.86267	0.0390737
-0.0233591	-5.44545	0.0452864	-0.0250006	-5.37168	0.0458855
-0.0376303	-4.90069	0.0510664	-0.0390917	-4.82665	0.0516056
-0.0496164	-4.37363	0.0558706	-0.0509147	-4.3056	0.0563599

Tableau VI.III: Table de coefficients pour la somme de 3 gaussiennes, pour les décalages dans le temps de 0 à -9 ns, pour des signaux produits par un générateur d'impulsions.

VI.2 ICs

VI.2.1 Formes moyennes expérimentales

Décalage de 0 ns			Décalage de -1 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.109432	-1.43558	0.0798646	-0.117158	-1.687	0.0850389
-0.109432	-1.43558	0.0798646	-0.117158	-1.687	0.0850389
-0.109432	-1.43558	0.0798646	-0.117158	-1.687	0.0850389
-0.109432	-1.43558	0.0798646	-0.117158	-1.687	0.0850389
-0.109432	-1.43558	0.0798646	-0.117158	-1.687	0.0850389
-0.109432	-1.43558	0.0798646	-0.117158	-1.687	0.0850389
-0.104824	-0.0960793	0.0770268	-0.111412	-0.17863	0.0813495
-0.0981758	0.459794	0.0737835	-0.0996644	1.48779	0.0748519
-0.0862706	2.01978	0.0676268	-0.0805142	3.58553	0.0647162
-0.0678598	3.93291	0.0584145	-0.0534304	5.90626	0.0508579
-0.043019	5.86711	0.0463848	-0.0204155	7.78559	0.0346651
-0.0130659	7.48126	0.0323233	0.0160544	9.06449	0.0173654
0.0196606	8.48655	0.0174287	0.0521056	9.34767	0.000987781
0.0523301	8.73472	0.00302705	0.0843794	8.72292	-0.0130262
0.0820881	8.19963	-0.0096206	0.110349	7.3769	-0.0236809
0.10667	7.02592	-0.0196161	0.12858	5.57516	-0.0305287
0.124407	5.35484	-0.0263185	0.138109	3.43319	-0.0332229
0.134517	3.41931	-0.0295313	0.138825	1.16737	-0.0318731
0.136986	1.41964	-0.029372	0.132045	-0.843372	-0.0273498
0.132374	-0.489829	-0.0261799	0.119455	-2.48119	-0.0204955
0.121774	-2.1576	-0.0205259	0.103124	-3.62569	-0.0123268
0.106545	-3.50548	-0.0130508	0.0846656	-4.37257	-0.00349772
0.0881284	-4.50064	-0.00440331	0.0658165	-4.67642	0.00518006
0.0679178	-5.14429	0.00480913	0.0472064	-4.80896	0.0136244

0.0471117	-5.47947	0.0140906	0.0292096	-4.83698	0.0217165
0.0267377	-5.54105	0.0230144	0.0121426	-4.77349	0.0293241
0.00751878	-5.4052	0.0313125	-0.00336018	-4.53216	0.0360991
-0.0100197	-5.11734	0.0387836	-0.0171028	-4.21976	0.0420323
-0.0255799	-4.73559	0.0453339	-0.0290261	-3.87265	0.0471238
-0.0390333	-4.30252	0.0509337	-0.0391754	-3.51851	0.0514146
-0.0504269	-3.86522	0.0556324	-0.0482898	-3.34045	0.0553711
-0.0598985	-3.44766	0.0595052	-0.0567276	-3.25003	0.0590887
Décalage de -2 ns			Décalage de -3 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.109894	-1.4487	0.0803029	-0.111325	-1.46668	0.0810724
-0.109894	-1.4487	0.0803029	-0.111325	-1.46668	0.0810724
-0.109894	-1.4487	0.0803029	-0.111325	-1.46668	0.0810724
-0.109894	-1.4487	0.0803029	-0.111325	-1.46668	0.0810724
-0.109894	-1.4487	0.0803029	-0.111325	-1.46668	0.0810724
-0.109894	-1.4487	0.0803029	-0.111325	-1.46668	0.0810724
-0.104598	0.0947498	0.0770291	-0.105928	0.0740416	0.0777372
-0.097396	0.616637	0.073551	-0.0978323	0.816169	0.073716
-0.0851561	2.12771	0.0672544	-0.0838196	2.54658	0.0664771
-0.0666051	4.00482	0.0579662	-0.0628815	4.57118	0.0560116
-0.0418519	5.89285	0.0459532	-0.0354785	6.5178	0.042755
-0.0122751	7.44057	0.0320365	-0.0034073	8.02025	0.0277258
0.0199908	8.43354	0.0172837	0.0305594	8.80161	0.0123153
0.0520693	8.64612	0.00309483	0.0633466	8.76262	-0.00206345
0.0812856	8.12099	-0.00938443	0.0921136	7.94482	-0.0141886
0.105494	6.98311	-0.0192908	0.114698	6.51116	-0.0232122
0.12312	5.37065	-0.0260197	0.129788	4.67422	-0.0286913
0.133464	3.51477	-0.029407	0.136947	2.6591	-0.0305745
0.136398	1.56296	-0.0294884	0.1365	0.667128	-0.0291309

0.13249	-0.293041	-0.0266298	0.129322	-1.15215	-0.0248387
0.122445	-2.01638	-0.0211643	0.11673	-2.67208	-0.0183557
0.107588	-3.42301	-0.0137853	0.100141	-3.85767	-0.0103262
0.0892286	-4.51372	-0.00506631	0.0809981	-4.68933	-0.00139611
0.0689065	-5.20828	0.00426814	0.0606618	-5.17682	0.00784267
0.0478453	-5.58809	0.0137322	0.0402033	-5.38893	0.0169627
0.0271296	-5.67681	0.0228651	0.0205158	-5.36204	0.0255945
0.00761699	-5.52843	0.0313245	0.0022053	-5.16719	0.0335161
-0.0102216	-5.24402	0.0389663	-0.0143309	-4.85418	0.0405844
-0.0260946	-4.86552	0.0456887	-0.0288841	-4.47016	0.0467371
-0.0397518	-4.39965	0.0513861	-0.0414236	-4.06136	0.0519897
-0.0512291	-3.92706	0.0561241	-0.0520026	-3.64893	0.0563787
-0.0605275	-3.43324	0.0598941	-0.0607914	-3.26574	0.0599997
Décalage de -4 ns			Décalage de -5 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.113234	-1.52671	0.0824776	-0.11574	-1.61473	0.084234
-0.113234	-1.52671	0.0824776	-0.11574	-1.61473	0.084234
-0.113234	-1.52671	0.0824776	-0.11574	-1.61473	0.084234
-0.113234	-1.52671	0.0824776	-0.11574	-1.61473	0.084234
-0.113234	-1.52671	0.0824776	-0.11574	-1.61473	0.084234
-0.113234	-1.52671	0.0824776	-0.11574	-1.61473	0.084234
-0.107102	0.183013	0.0786417	-0.108638	0.309726	0.0797238
-0.0976592	1.1142	0.0738452	-0.0976446	1.43451	0.0740213
-0.081792	2.97095	0.0655947	-0.079631	3.44635	0.0645609
-0.0587219	5.08008	0.0539842	-0.0543868	5.5852	0.0517723
-0.0293169	7.01137	0.0396796	-0.0230632	7.48433	0.0364246
0.00427262	8.41192	0.023865	0.0118218	8.76209	0.0199092
0.0389833	9.01674	0.00807239	0.046923	9.1692	0.00389742
0.0715732	8.75361	-0.00621301	0.0791107	8.73069	-0.0102255

0.0992434	7.71594	-0.0178105	0.105624	7.51503	-0.0212692
0.120006	6.09897	-0.0259658	0.12458	5.733	-0.0285358
0.132884	4.15861	-0.0304124	0.135328	3.69896	-0.0319501
0.137658	2.0882	-0.0311791	0.137757	1.55263	-0.0315582
0.13496	0.116875	-0.0286954	0.133084	-0.346622	-0.0281543
0.125968	-1.60323	-0.0235841	0.122458	-2.00608	-0.0222476
0.112085	-3.00333	-0.0165119	0.107544	-3.27365	-0.0146896
0.0947745	-4.05432	-0.0081468	0.0897969	-4.19021	-0.00610387
0.0754564	-4.75432	0.000879677	0.0705597	-4.77119	0.00291822
0.0553893	-5.13449	0.0100297	0.0508388	-5.09444	0.0119793
0.0355227	-5.26739	0.0189294	0.0316973	-5.1346	0.0205855
0.0166588	-5.18234	0.027242	0.0136887	-5.01904	0.0285766
-0.000722396	-4.96018	0.0348075	-0.00279355	-4.78556	0.0358023
-0.0163413	-4.64954	0.0415334	-0.0174324	-4.45097	0.0421321
-0.0300602	-4.28566	0.0473818	-0.0303521	-4.13161	0.047702
-0.041908	-3.91244	0.0523939	-0.0415815	-3.80293	0.0525079
-0.0519876	-3.54955	0.0566285	-0.0512391	-3.48937	0.0566198
-0.0604238	-3.20342	0.060145	-0.0596086	-3.23703	0.0601967
Décalage de -6 ns			Décalage de -7 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.107828	-1.36786	0.0795113	-0.122766	-1.94447	0.0896796
-0.107828	-1.36786	0.0795113	-0.122766	-1.94447	0.0896796
-0.107828	-1.36786	0.0795113	-0.122766	-1.94447	0.0896796
-0.107828	-1.36786	0.0795113	-0.122766	-1.94447	0.0896796
-0.107828	-1.36786	0.0795113	-0.122766	-1.94447	0.0896796
-0.107828	-1.36786	0.0795113	-0.122766	-1.94447	0.0896796
-0.100285	0.907389	0.0748821	-0.106601	2.14007	0.0788573
-0.0950335	0.096497	0.0730361	-0.0960792	1.30354	0.0746693
-0.0859597	1.37141	0.0683082	-0.0778455	3.2698	0.0646203

-0.0705413	3.26654	0.0604302	-0.0522473	5.51888	0.0509303
-0.0488328	5.15547	0.0497864	-0.0202329	7.6316	0.0343811
-0.0216972	6.84759	0.0368663	0.015479	9.10726	0.0166405
0.00890909	8.02076	0.0227164	0.0511052	9.59719	-0.000254765
0.0399316	8.3518	0.00886744	0.0835563	9.2025	-0.0149571
0.0693338	8.14494	-0.00395892	0.109788	7.91416	-0.0260495
0.0943073	7.12093	-0.0143656	0.127612	5.91709	-0.0326699
0.113862	5.82411	-0.0222298	0.136851	3.71701	-0.0351497
0.126914	4.1694	-0.0270252	0.137048	1.29425	-0.033261
0.133278	2.42436	-0.0288237	0.130522	-0.65623	-0.0286454
0.132813	0.577971	-0.0275575	0.118155	-2.40173	-0.0214928
0.126076	-1.1603	-0.0235919	0.102562	-3.52324	-0.0133567
0.113973	-2.69712	-0.0173796	0.085065	-4.27757	-0.00463931
0.0975068	-4.0023	-0.00937966	0.0669552	-4.68646	0.00407969
0.0784307	-4.84242	-0.000497788	0.049325	-4.78918	0.0123264
0.057941	-5.38752	0.00883682	0.0319659	-4.90773	0.0204605
0.0371686	-5.63097	0.0181225	0.0157839	-4.79631	0.0278617
0.0164529	-5.80074	0.0273409	0.00100852	-4.59103	0.034535
-0.0033637	-5.72216	0.0360206	-0.0123592	-4.3616	0.0405367
-0.0218606	-5.52902	0.0440534	-0.0241078	-4.06213	0.0457316
-0.0381463	-5.06358	0.0509561	-0.0350571	-3.9463	0.0507064
-0.0517288	-4.45101	0.0565841	-0.0456834	-3.96205	0.0556393
-0.0624785	-3.78492	0.0609341	-0.0559734	-3.985	0.0604223
Décalage de -8 ns			Décalage de -9 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.115096	-1.42289	0.0822798	-0.123502	-1.65644	0.0886583
-0.115096	-1.42289	0.0822798	-0.123502	-1.65644	0.0886583
-0.115096	-1.42289	0.0822798	-0.123502	-1.65644	0.0886583
-0.115096	-1.42289	0.0822798	-0.123502	-1.65644	0.0886583

-0.115096	-1.42289	0.0822798	-0.123502	-1.65644	0.0886583
-0.115096	-1.42289	0.0822798	-0.123502	-1.65644	0.0886583
-0.100022	3.04462	0.0732993	-0.106137	2.75675	0.0777671
-0.0900432	1.27813	0.069889	-0.0893359	2.73768	0.0701224
-0.0765028	2.59443	0.0632691	-0.0647033	4.89624	0.0574394
-0.0579144	4.10597	0.0543398	-0.031074	7.40491	0.0404196
-0.0336375	5.83997	0.0428068	0.00743002	8.95208	0.0218264
-0.00478294	7.28136	0.0294316	0.0472702	9.65945	0.00319057
0.0268423	8.25967	0.0150948	0.0833471	9.09359	-0.0128714
0.0587481	8.56987	0.000994061	0.112825	7.78352	-0.0254235
0.0879284	8.05291	-0.0114726	0.132963	5.71956	-0.0332106
0.112118	6.91268	-0.0214251	0.142236	3.18257	-0.0357262
0.129832	5.33925	-0.028317	0.142693	1.06323	-0.0345054
0.139435	3.25614	-0.0313933	0.134928	-0.998147	-0.029576
0.140999	1.15931	-0.030951	0.12137	-2.52492	-0.0223669
0.135373	-0.764644	-0.0274617	0.104542	-3.47474	-0.014056
0.123341	-2.54415	-0.0212521	0.0855392	-4.18493	-0.0049159
0.107336	-3.67983	-0.0136527	0.0663703	-4.40933	0.00397246
0.0880362	-4.71288	-0.00466936	0.0474849	-4.52647	0.0126593
0.0668408	-5.37344	0.00494185	0.030242	-4.2966	0.0203634
0.0444712	-5.87317	0.0149796	0.0152982	-3.88731	0.0268987
0.0217839	-6.14544	0.0250339	0.00130635	-3.79938	0.0332168
0.000785239	-5.84706	0.0340447	-0.0121762	-3.81086	0.0393697
-0.0182584	-5.50481	0.042178	-0.0247979	-3.72678	0.0450659
-0.0346109	-4.92887	0.0490106	-0.0365548	-3.63415	0.0503621
-0.0482054	-4.32747	0.0546251	-0.0473043	-3.49674	0.055169
-0.0597742	-3.91792	0.0594578	-0.0569735	-3.33124	0.0594646
-0.0695425	-3.53729	0.0635197	-0.0657755	-3.20938	0.0633943

Tableau VI.IV: Table de coefficients pour les formes moyennes expérimentales, pour les décalages dans le temps de 0 à -9 ns, pour des signaux générés dans un cristal de ICs.

VI.2.2 Filtres actifs

Décalage de 0 ns			Décalage de -1 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.0775103	-4.79773	0.0756774	-0.0783447	-4.7973	0.0761593
-0.0775103	-4.79773	0.0756774	-0.0783447	-4.7973	0.0761593
-0.0775103	-4.79773	0.0756774	-0.0783447	-4.7973	0.0761593
-0.0775103	-4.79773	0.0756774	-0.0783447	-4.7973	0.0761593
-0.0775103	-4.79773	0.0756774	-0.0783447	-4.7973	0.0761593
-0.0775103	-4.79773	0.0756774	-0.0783447	-4.7973	0.0761593
-0.0775103	-4.79773	0.0756774	-0.0783447	-4.7973	0.0761593
-0.0775103	-4.79773	0.0756774	-0.0782766	-4.69354	0.0758595
-0.0770226	-3.95199	0.07325	-0.0777046	-3.9963	0.0737974
-0.0762501	-4.21112	0.0736156	-0.0769733	-4.21973	0.0740806
-0.0755075	-4.02317	0.072816	-0.0761844	-3.97347	0.07311
-0.0741579	-3.13148	0.0699141	-0.0746562	-3.00709	0.0699484
-0.0713621	-1.40928	0.0642317	-0.0715309	-1.21463	0.063964
-0.0662773	1.04463	0.0556833	-0.0659877	1.28417	0.0551316
-0.0582856	3.93723	0.0447873	-0.0574649	4.18436	0.0440178
-0.0471492	6.87777	0.0324752	-0.0457971	7.09442	0.0315798
-0.0330568	9.48	0.0198416	-0.0312428	9.63586	0.0189153
-0.0165723	11.4368	0.00792678	-0.0144207	11.5139	0.00705175
0.00147509	12.558	-0.00242851	0.00380731	12.5516	-0.00319158
0.0201236	12.7761	-0.0106518	0.0224695	12.6927	-0.0112661
0.0384045	12.1295	-0.0164471	0.0406059	11.9837	-0.0168975
0.0554416	10.7349	-0.0197621	0.0573634	10.5459	-0.020051
0.0705187	8.75633	-0.0207353	0.072057	8.54479	-0.0208771
0.0831144	6.37729	-0.019636	0.0841996	6.16265	-0.0196522
0.0929093	3.77844	-0.0168091	0.093505	3.57739	-0.016724
0.0997727	1.12207	-0.0126282	0.0998725	0.947595	-0.0124661
0.103736	-1.45707	-0.00746141	0.103358	-1.59598	-0.00724424

0.104961	-3.85572	-0.00164653	0.104141	-3.95387	-0.00139283
0.103701	-6.00194	0.00452258	0.102489	-6.05749	0.00479815
0.100273	-7.85246	0.0108026	0.0987246	-7.86625	0.0110892
0.0950214	-9.38831	0.0170013	0.0931973	-9.36325	0.0172915
0.0883023	-10.61	0.0229747	0.086261	-10.5504	0.023264
0.0804586	-11.5325	0.0286213	0.078256	-11.4434	0.0289077
Décalage de -2 ns			Décalage de -3 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.0791828	-4.79696	0.0766457	-0.0800248	-4.79673	0.0771367
-0.0791828	-4.79696	0.0766457	-0.0800248	-4.79673	0.0771367
-0.0791828	-4.79696	0.0766457	-0.0800248	-4.79673	0.0771367
-0.0791828	-4.79696	0.0766457	-0.0800248	-4.79673	0.0771367
-0.0791828	-4.79696	0.0766457	-0.0800248	-4.79673	0.0771367
-0.0791828	-4.79696	0.0766457	-0.0800248	-4.79673	0.0771367
-0.0790347	-4.5958	0.0760604	-0.0797863	-4.5042	0.0762797
-0.0784013	-4.03746	0.0743434	-0.0791116	-4.07542	0.0748875
-0.0777013	-4.22358	0.0745371	-0.0784324	-4.22251	0.0749842
-0.0768502	-3.91716	0.0733856	-0.077503	-3.85417	0.073642
-0.0751241	-2.8759	0.0699571	-0.0755596	-2.73802	0.0699399
-0.0716516	-1.01512	0.063669	-0.0717229	-0.81102	0.0633469
-0.0656392	1.52521	0.0545561	-0.0652317	1.76745	0.0539572
-0.0565842	4.4294	0.0432313	-0.0556443	4.67206	0.0424283
-0.0443933	7.30604	0.0306752	-0.0429396	7.51249	0.0297619
-0.0293931	9.78499	0.017987	-0.0275095	9.92736	0.0170571
-0.0122533	11.5839	0.00618032	-0.010072	11.6469	0.00531275
0.00613475	12.5388	-0.00394741	0.0084558	12.5198	-0.0046959
0.0247919	12.6044	-0.0118713	0.0270899	12.5113	-0.0124675
0.0427693	11.8349	-0.0173386	0.0448943	11.6832	-0.0177703
0.0592377	10.3558	-0.0203312	0.0610647	10.1647	-0.0206027

0.0735436	8.33378	-0.0210113	0.0749791	8.12341	-0.021138
0.0852337	5.94983	-0.0196621	0.0862178	5.73889	-0.0196658
0.0940545	3.37899	-0.0166338	0.094559	3.18325	-0.0165388
0.099934	0.776139	-0.0123	0.0999584	0.607681	-0.0121297
0.102952	-1.73191	-0.00702364	0.102518	-1.86489	-0.00679959
0.103304	-4.0494	-0.00113598	0.102451	-4.14237	-0.000875952
0.101271	-6.11101	0.00507687	0.100048	-6.16258	0.00535882
0.0971816	-7.87878	0.0113792	0.0956436	-7.89009	0.0116727
0.091388	-9.33774	0.0175856	0.0895932	-9.31184	0.0178836
0.0842432	-10.4911	0.0235578	0.0822481	-10.4323	0.023856
0.076084	-11.3554	0.029199	0.0739417	-11.2687	0.0294954
Décalage de -4 ns			Décalage de -5 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.0808709	-4.79661	0.0776324	-0.0817213	-4.79661	0.0781329
-0.0808709	-4.79661	0.0776324	-0.0817213	-4.79661	0.0781329
-0.0808709	-4.79661	0.0776324	-0.0817213	-4.79661	0.0781329
-0.0808709	-4.79661	0.0776324	-0.0817213	-4.79661	0.0781329
-0.0808709	-4.79661	0.0776324	-0.0817213	-4.79661	0.0781329
-0.0808709	-4.79661	0.0776324	-0.0817213	-4.79661	0.0781329
-0.0805331	-4.4184	0.0765169	-0.0812763	-4.33808	0.0767714
-0.0798345	-4.11011	0.0754295	-0.080569	-4.14144	0.0759688
-0.0791651	-4.21634	0.0754211	-0.0798977	-4.20492	0.0758469
-0.0781406	-3.78443	0.0738784	-0.078761	-3.70792	0.074094
-0.0759607	-2.59361	0.0698964	-0.0763255	-2.44283	0.0698262
-0.0717436	-0.602637	0.0629977	-0.0717126	-0.390265	0.0626217
-0.0647649	2.01054	0.0533355	-0.0642389	2.25417	0.0526913
-0.0546462	4.9121	0.0416095	-0.0535909	5.14925	0.0407754
-0.0414377	7.71362	0.0288405	-0.0398893	7.90929	0.0279116
-0.0255942	10.063	0.016126	-0.0236493	10.1918	0.0151942

-0.00787868	11.7029	0.00444927	-0.00567535	11.7521	0.0035901
0.0107689	12.4946	-0.00543698	0.0130725	12.4636	-0.0061706
0.0293625	12.4137	-0.0130548	0.0316088	12.3118	-0.0136332
0.0469806	11.5288	-0.0181929	0.049028	11.3719	-0.0186065
0.0628447	9.97279	-0.0208658	0.064578	9.78014	-0.0211206
0.0763644	7.91378	-0.0212575	0.0777001	7.70498	-0.0213698
0.0871529	5.52988	-0.0196635	0.08804	5.32284	-0.0196553
0.0950196	2.99018	-0.0164388	0.0954374	2.7998	-0.016334
0.0999467	0.4422	-0.0119555	0.0999001	0.279672	-0.0117773
0.102058	-1.99497	-0.00657208	0.101572	-2.12221	-0.00634109
0.101582	-4.23282	-0.000612677	0.100697	-4.32082	-0.000346104
0.0988188	-6.21224	0.00564409	0.0975853	-6.26005	0.00593275
0.0941106	-7.90024	0.0119698	0.0925825	-7.90929	0.0122706
0.0878126	-9.28558	0.0181857	0.0860459	-9.25899	0.0184919
0.0802752	-10.3739	0.0241589	0.0783238	-10.316	0.0244665
0.0718281	-11.1832	0.029797	0.0697422	-11.0988	0.0301039
Décalage de -6 ns			Décalage de -7 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.0825761	-4.79674	0.0786383	-0.0834357	-4.797	0.0791486
-0.0825761	-4.79674	0.0786383	-0.0834357	-4.797	0.0791486
-0.0825761	-4.79674	0.0786383	-0.0834357	-4.797	0.0791486
-0.0825761	-4.79674	0.0786383	-0.0834357	-4.797	0.0791486
-0.0825761	-4.79674	0.0786383	-0.0834357	-4.797	0.0791486
-0.0825761	-4.79674	0.0786383	-0.0834357	-4.797	0.0791486
-0.0820173	-4.26292	0.0770427	-0.0827572	-4.19257	0.0773302
-0.081314	-4.16931	0.076505	-0.0820683	-4.19359	0.0770374
-0.0806285	-4.18808	0.0762606	-0.0813558	-4.16566	0.0766613
-0.0793621	-3.6246	0.0742881	-0.0799419	-3.53447	0.0744598
-0.0766523	-2.28584	0.0697291	-0.0769393	-2.12285	0.0696048

-0.083497	-4.12671	0.0776332	-0.0842376	-4.065	0.077951
-0.0828307	-4.21416	0.0775654	-0.0836001	-4.23085	0.0780884
-0.0820778	-4.1375	0.0770481	-0.0827926	-4.10348	0.07742
-0.0804983	-3.43755	0.0746086	-0.0810292	-3.33386	0.0747337
-0.0771848	-1.95404	0.0694531	-0.0773874	-1.77964	0.0692739
-0.0713005	0.267765	0.0613341	-0.0710542	0.493056	0.0608527
-0.0623073	2.98515	0.0506294	-0.0615466	3.22782	0.0499006
-0.0500933	5.84104	0.0381873	-0.0488209	6.06435	0.037298
-0.0349825	8.46244	0.0250847	-0.0332656	8.63519	0.0241307
-0.0176556	10.5376	0.012398	-0.0156112	10.6394	0.0114669
0.000976952	11.8594	0.00104058	0.00320279	11.8821	0.000200714
0.0199133	12.3364	-0.00832603	0.0221659	12.2832	-0.00902921
0.038182	11.9816	-0.0153153	0.0403156	11.8639	-0.0158585
0.0549343	10.8878	-0.0197942	0.056824	10.7224	-0.0201728
0.0695007	9.1992	-0.0218365	0.0710506	9.00494	-0.0220594
0.0814172	7.0844	-0.021665	0.0825621	6.87973	-0.0217499
0.0904235	4.71399	-0.0195961	0.0911287	4.51524	-0.0195652
0.0964449	2.24473	-0.0159914	0.0967022	2.06506	-0.0158678
0.099561	-0.190471	-0.0112189	0.0993848	-0.341465	-0.0110249
0.0999713	-2.48726	-0.00562694	0.0993924	-2.60354	-0.00538172
0.0979591	-4.57064	0.000474004	0.09702	-4.64935	0.00075438
0.0938588	-6.39294	0.00681993	0.0926088	-6.43388	0.00712299
0.088028	-7.93018	0.0131963	0.0865194	-7.93521	0.013513
0.0808248	-9.17762	0.0194368	0.0791096	-9.15005	0.0197608
0.0725912	-10.1451	0.0254187	0.0707188	-10.0892	0.0257462
0.0636411	-10.853	0.0310574	0.0616565	-10.7734	0.0313866

Tableau VI.V: Table de coefficients pour l'équation (5.1) pour les filtres actifs, pour les décalages dans le temps de 0 à -9 ns, pour des signaux générés dans un cristal de ICs.

VI.2.3 Somme de 3 gaussiennes

Décalage de 0 ns			Décalage de -1 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.102627	-1.62938	0.0767286	-0.103161	-1.62214	0.0770232
-0.102627	-1.62938	0.0767286	-0.103161	-1.62214	0.0770232
-0.102627	-1.62938	0.0767286	-0.103161	-1.62214	0.0770232
-0.102627	-1.62938	0.0767286	-0.103161	-1.62214	0.0770232
-0.102627	-1.62938	0.0767286	-0.103161	-1.62214	0.0770232
-0.102627	-1.62938	0.0767286	-0.103161	-1.62214	0.0770232
-0.100189	-0.93707	0.0750938	-0.100527	-0.875394	0.0752648
-0.0975655	-0.834551	0.0738796	-0.0977152	-0.773749	0.0739693
-0.0925726	-0.131662	0.071138	-0.0924118	-0.0357477	0.0710688
-0.0839428	0.976921	0.0664895	-0.0833206	1.11583	0.0661917
-0.0702597	2.53901	0.059285	-0.0690307	2.71913	0.0586986
-0.0504457	4.47433	0.0491296	-0.0485252	4.67777	0.0482321
-0.0243542	6.52629	0.0361779	-0.0217784	6.71769	0.0350105
0.00674227	8.28391	0.0213253	0.00978235	8.41806	0.02
0.0400861	9.29768	0.00613822	0.0432659	9.33551	0.00481623
0.0719758	9.2554	-0.00752817	0.0749172	9.18144	-0.00868278
0.0987781	8.12798	-0.018088	0.101158	7.95884	-0.0189589
0.117966	6.19215	-0.0246903	0.119602	5.96966	-0.0252369
0.128673	3.90471	-0.0273506	0.129542	3.67701	-0.0276001
0.131534	1.69933	-0.0267089	0.131727	1.50108	-0.0267217
0.127986	-0.173039	-0.023596	0.127633	-0.329954	-0.023428
0.119512	-1.67629	-0.0186796	0.118721	-1.79806	-0.0183632
0.107215	-2.90186	-0.0123561	0.106055	-3.00012	-0.0119051
0.0918432	-3.94317	-0.00486772	0.0903672	-4.024	-0.00429425
0.0740692	-4.81724	0.00351086	0.0723408	-4.87791	0.00418113
0.0547479	-5.46652	0.0123636	0.0528581	-5.49955	0.0130893

0.0349639	-5.81499	0.02116	0.0330238	-5.8148	0.0218928
0.0158743	-5.82841	0.0293737	0.0139929	-5.79587	0.0300709
-0.00153352	-5.54221	0.0366106	-0.0032725	-5.48498	0.0372448
-0.0166337	-5.05043	0.0426816	-0.0181852	-4.9802	0.0432436
-0.0292274	-4.47156	0.0476018	-0.0305853	-4.40012	0.0480977
-0.0394836	-3.91242	0.0515343	-0.0406714	-3.84869	0.0519793
Décalage de -2 ns			Décalage de -3 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.103692	-1.61502	0.0773167	-0.10422	-1.60801	0.077609
-0.103692	-1.61502	0.0773167	-0.10422	-1.60801	0.077609
-0.103692	-1.61502	0.0773167	-0.10422	-1.60801	0.077609
-0.103692	-1.61502	0.0773167	-0.10422	-1.60801	0.077609
-0.103692	-1.61502	0.0773167	-0.10422	-1.60801	0.077609
-0.103692	-1.61502	0.0773167	-0.10422	-1.60801	0.077609
-0.100848	-0.809943	0.0754264	-0.101152	-0.740477	0.0755778
-0.097837	-0.710087	0.074045	-0.0979294	-0.643451	0.074106
-0.0922085	0.0640213	0.0709788	-0.0919608	0.167722	0.070867
-0.0826396	1.25896	0.0658658	-0.0818979	1.40626	0.0655109
-0.0677295	2.90246	0.0580788	-0.0663548	3.08883	0.0574252
-0.0465283	4.88168	0.047301	-0.0444551	5.08569	0.0463363
-0.0191366	6.90538	0.0338158	-0.0164305	7.08888	0.0325946
0.0128625	8.54414	0.0186599	0.0159796	8.66176	0.0173065
0.0464494	9.36259	0.00349509	0.049633	9.37876	0.00217633
0.0778257	9.09706	-0.00982277	0.0806979	9.00244	-0.0109469
0.103479	7.78261	-0.0198071	0.105737	7.59972	-0.0206317
0.121166	5.74494	-0.025759	0.12266	5.51842	-0.0262565
0.130343	3.45128	-0.0278279	0.131076	3.2278	-0.0280338
0.131862	1.30686	-0.0267163	0.13194	1.11674	-0.0266927
0.127232	-0.482969	-0.0232439	0.126784	-0.632186	-0.0230433

0.117889	-1.91719	-0.0180308	0.117016	-2.0338	-0.0176822
0.10486	-3.09681	-0.0114377	0.103627	-3.19196	-0.0109538
0.0888595	-4.10342	-0.00370527	0.0873206	-4.18137	-0.0031009
0.070588	-4.93658	0.0048642	0.0688114	-4.99315	0.00555973
0.050954	-5.52988	0.0138238	0.0490366	-5.55744	0.0145664
0.0310807	-5.81172	0.02263	0.0291355	-5.80573	0.023371
0.0121185	-5.76094	0.030769	0.0102518	-5.72368	0.0314675
-0.00499767	-5.42642	0.0378778	-0.00670853	-5.36661	0.0385092
-0.0197198	-4.90984	0.0438037	-0.0212376	-4.83944	0.0443616
-0.0319269	-4.32958	0.048592	-0.0332521	-4.25996	0.0490846
-0.0418455	-3.78646	0.0524236	-0.043006	-3.72574	0.0528671
Décalage de -4 ns			Décalage de -5 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.104746	-1.60109	0.0779	-0.105269	-1.59426	0.0781896
-0.104746	-1.60109	0.0779	-0.105269	-1.59426	0.0781896
-0.104746	-1.60109	0.0779	-0.105269	-1.59426	0.0781896
-0.104746	-1.60109	0.0779	-0.105269	-1.59426	0.0781896
-0.104746	-1.60109	0.0779	-0.105269	-1.59426	0.0781896
-0.104746	-1.60109	0.0779	-0.105269	-1.59426	0.0781896
-0.101437	-0.666748	0.0757186	-0.101702	-0.588497	0.075848
-0.0979908	-0.573729	0.0741515	-0.0980196	-0.500813	0.0741807
-0.0916668	0.275425	0.0707325	-0.0913247	0.387189	0.0705744
-0.0810937	1.55771	0.0651263	-0.0802254	1.71323	0.0647111
-0.0649057	3.27802	0.0567373	-0.063381	3.46977	0.0560148
-0.0423059	5.2894	0.0453382	-0.0400809	5.49242	0.044307
-0.013662	7.26773	0.0313478	-0.0108331	7.44145	0.0300764
0.0191304	8.77055	0.0159412	0.0223115	8.87017	0.0145655
0.0528127	9.38394	0.0008615	0.0559847	9.37804	-0.000447852
0.0835307	8.89779	-0.0120539	0.0863207	8.78335	-0.0131426

0.107933	7.41059	-0.0214321	0.110062	7.21567	-0.0222077
0.124082	5.29055	-0.026729	0.125431	5.06177	-0.0271764
0.131741	3.00685	-0.028218	0.132341	2.78866	-0.0283803
0.131961	0.930754	-0.0266507	0.131926	0.748926	-0.0265904
0.12629	-0.777713	-0.0228262	0.125749	-0.919664	-0.0225923
0.116103	-2.14802	-0.017317	0.115151	-2.25996	-0.0169352
0.102359	-3.28561	-0.0104533	0.101056	-3.37778	-0.00993625
0.0857508	-4.25778	-0.00248126	0.0841505	-4.33259	-0.00184655
0.0670118	-5.0475	0.00626737	0.0651901	-5.09954	0.00698674
0.0471067	-5.58214	0.0153169	0.0451655	-5.60392	0.0160745
0.0271893	-5.79682	0.0241154	0.0252431	-5.78499	0.0248628
0.00839378	-5.68414	0.032166	0.00654513	-5.64239	0.0328641
-0.0084046	-5.30564	0.0391387	-0.0100854	-5.24359	0.039766
-0.0227382	-4.76907	0.0449172	-0.0242215	-4.69881	0.0454703
-0.0345612	-4.19133	0.0495754	-0.0358541	-4.12372	0.0500642
-0.0441533	-3.66652	0.0533097	-0.0452874	-3.60882	0.0537513
Décalage de -6 ns			Décalage de -7 ns		
C_A	C_B	C_P	C_A	C_B	C_P
-0.105788	-1.58752	0.0784776	-0.106303	-1.58085	0.0787639
-0.105788	-1.58752	0.0784776	-0.106303	-1.58085	0.0787639
-0.105788	-1.58752	0.0784776	-0.106303	-1.58085	0.0787639
-0.105788	-1.58752	0.0784776	-0.106303	-1.58085	0.0787639
-0.105788	-1.58752	0.0784776	-0.106303	-1.58085	0.0787639
-0.105788	-1.58752	0.0784776	-0.106303	-1.58085	0.0787639
-0.101947	-0.505457	0.0759654	-0.102169	-0.417351	0.0760701
-0.0980142	-0.424595	0.0741928	-0.0979727	-0.344972	0.0741869
-0.0909326	0.503065	0.070392	-0.0904886	0.623093	0.0701842
-0.0792912	1.87276	0.0642646	-0.0782895	2.03619	0.0637862
-0.0617801	3.66384	0.0552574	-0.060102	3.85995	0.0544648

-0.102368	-0.323893	0.0761615	-0.102542	-0.224792	0.0762387
-0.0978936	-0.261845	0.0741622	-0.0977749	-0.175119	0.0741178
-0.0899908	0.7473	0.0699501	-0.0894372	0.8757	0.069689
-0.0772186	2.2034	0.0632751	-0.0760771	2.37426	0.0627307
-0.0583462	4.05779	0.0536369	-0.0565122	4.25702	0.0527735
-0.0329586	6.09298	0.0410191	-0.0304384	6.28881	0.0398601
-0.0020075	7.92716	0.0261259	0.00103868	8.07569	0.0247677
0.0320005	9.11065	0.0103914	0.0352663	9.17037	0.00898981
0.0654162	9.29353	-0.00432733	0.0685188	9.24314	-0.00559905
0.0944037	8.3841	-0.0162871	0.0969924	8.23344	-0.0172911
0.116043	6.6007	-0.0243791	0.117895	6.38718	-0.0250493
0.129043	4.37408	-0.0283662	0.130101	4.14572	-0.0287115
0.133744	2.1529	-0.028736	0.134082	1.94786	-0.0288107
0.131494	0.228369	-0.0262979	0.131243	0.0630773	-0.0261629
0.123853	-1.32523	-0.021788	0.123132	-1.45406	-0.0214851
0.112054	-2.58303	-0.0156878	0.110944	-2.68674	-0.0152375
0.0969321	-3.64538	-0.00828528	0.0954877	-3.73156	-0.00770181
0.0791725	-4.54647	0.000145818	0.077456	-4.61393	0.000838442
0.0596008	-5.24067	0.00921099	0.0576995	-5.28237	0.00997291
0.0392849	-5.65111	0.0183861	0.0373095	-5.6606	0.0191677
0.0194151	-5.73209	0.0271172	0.0174795	-5.7087	0.0278712
0.00106372	-5.50449	0.0349522	-0.000739237	-5.45454	0.0356448
-0.0150314	-5.05185	0.0416324	-0.0166465	-4.98635	0.0422483
-0.0285664	-4.4894	0.0471124	-0.0299792	-4.42027	0.0476534
-0.0396368	-3.92736	0.0515174	-0.0408659	-3.86415	0.0519968
-0.0486126	-3.44469	0.0550684	-0.0496958	-3.39295	0.0555046

Tableau VI.VI: Table de coefficients pour la somme de 3 gaussiennes, pour les décalages dans le temps de 0 à -9 ns, pour des signaux générés dans un cristal de ICs.

Annexe VII

Code pour la lecture des fichiers binaires de l'oscilloscope LeCroy

```
//lecture_lecroy.cpp
//Ecrit par Simon Lagrange le 24 fevrier 2015, modifie le 16 mars 2015
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <stdlib.h>
#include <stdio.h>

using namespace std ;

////////////////////////////////////
//          Definition des fonctions          //
////////////////////////////////////

/////    Pour trouver l'adresse du point de reference , indiquee par le string "WAVEDESC"    /////

int initpoint(char datatest[], int j){
    string teststr ;
    string wave = "WAVEDESC" ;
    for ( int k = 0 ; k < 8 ; k++){
        teststr += datatest[j+k] ;
    }
    return teststr != wave ;    //Retourne vrai ou faux
}

/////    Pour lire un ou deux octets d'info    /////

unsigned short int readbyte(char datatest[], int address){ //Pour 1 octet - 8 bits
    return datatest[address] ;
}

unsigned short int readword(char datatest[], int address, int comm_order){ //Pour 2 octets - 16 bits
    unsigned short int conc ;
    if (comm_order == 0){
        char datatemp [] = {datatest[address], datatest[address+1]} ;
        memcpy(&conc, &datatemp, sizeof(conc)) ;
    }
    else{
        char datatemp [] = {datatest[address+1], datatest[address]} ;
        memcpy(&conc, &datatemp, sizeof(conc)) ;
    }
    return conc ;
}

unsigned long int readlong(char datatest[], int address, int comm_order){ //Pour 4 octets - 32 bits
    unsigned long int conc ;
    if (comm_order == 0){
        char datatemp [] = {datatest[address], datatest[address+1], datatest[address+2], datatest[address+3]} ;
        memcpy(&conc, &datatemp, sizeof(conc)) ;
    }
    else{
        char datatemp [] = {datatest[address+3], datatest[address+2], datatest[address+1], datatest[address]} ;
        memcpy(&conc, &datatemp, sizeof(conc)) ;
    }
}
```

```

    }
    return conc ;
}

float readfloat(char datatest[], int address, int comm_order){ //Pour 4 octets - 32 bits
    float conc ;
    if (comm_order == 0){
        char datatemp [] = {datatest[address], datatest[address+1], datatest[address+2], datatest[address+3]} ;
        memcpy(&conc, &datatemp, sizeof(conc)) ;
    }
    else{
        char datatemp [] = {datatest[address+3], datatest[address+2], datatest[address+1], datatest[address]} ;
        memcpy(&conc, &datatemp, sizeof(conc)) ;
    }
    return conc ;
}

double readdouble(char datatest[], int address, int comm_order){ //Pour 8 octets - 64 bits
    double conc ;
    if (comm_order == 0){
        char datatemp [] = {datatest[address], datatest[address+1], datatest[address+2], datatest[address+3], datatest[
            address+4], datatest[address+5], datatest[address+6], datatest[address+7]} ;
        memcpy(&conc, &datatemp, sizeof conc) ;
    }
    else{
        char datatemp [] = {datatest[address+7], datatest[address+6], datatest[address+5], datatest[address+4], datatest
            [address+3], datatest[address+2], datatest[address+1], datatest[address]} ;
        memcpy(&conc, &datatemp, sizeof conc) ;
    }
    return conc ;
}

string readstring(char datatest[], int address, int comm_order){ //To read strings of 16 characters
    string readstr ;
    if (comm_order == 0 ){
        for (int j = 0 ; j < 16 ; j++){
            readstr += datatest[address+j] ;
        }
    }
    else{
        for (int j = 0 ; j < 16 ; j++){
            readstr += datatest[address+15-j] ;
        }
    }
    return readstr ;
}

string readtimestamp(char datatest[], int address, int comm_order){ //Timestamp ;
    char timestamp [50];
    long double seconds = readdouble(datatest, address, comm_order) ; //Read 8 bytes - 64 bits; precise to the
        nanosecond
    int minutes = readbyte(datatest, address+8) ; //Read 1 byte - 8 bits
    int hour = readbyte(datatest, address+9) ; //Read 1 byte - 8 bits
    int day = readbyte(datatest, address+10) ; //Read 1 byte - 8 bits
    int month = readbyte(datatest, address+11) ; //Read 1 byte - 8 bits
    int year = readword(datatest, address+12, comm_order) ; //Read 2 bytes - 16 bits
    sprintf(timestamp, "%i.%i.%i, %i:%i:%5.9f\n", day, month, year, hour, minutes, seconds) ;
    return timestamp ;
}

unsigned short int readtimebase(char datatest[], int address, int comm_order){

```

```

    unsigned short int ee = readword(datatest, address, comm_order);
    unsigned short int tmp[] = {1, 2, 5};
    unsigned short int mant = ee%3;
    unsigned short int ex = floor(ee/3) - 12;
    return mant*(10^ex);
}

unsigned short int readfixed_vert_gain(char datatest[], int address, int comm_order){
    unsigned short int ee = readword(datatest, address, comm_order);
    unsigned short int tmp[] = {1, 2, 5};
    unsigned short int mant = ee%3;
    unsigned short int ex = floor(ee/3) - 6;
    return mant*(10^ex);
}

string float_to_eng(float flo){
    //Convert floats to engineering format
    int ex = floor(log10(flo));
    int exeng = ex - (ex%3);
    if (exeng < -18){
        exeng = -18;
    }
    else if(exeng > 18){
        exeng = 18;
    }
    float mant = flo/(10^exeng);
    char prefix [] = "afpnum kMGPE"; //Prefixes (u=micro, m=milli, k=kilo, ...)
    char result [100];
    char temp = prefix[ (exeng+18)/3 + 1 ];
    sprintf(result, "%e%c", mant, temp);
    return result;
}

////////////////////////////////////
//***          ***** Main code *****          ***//
////////////////////////////////////

int main(){

    //////////////////////////////////////
    //          Initialisation of variables          //
    //////////////////////////////////////

    //***** Iterators //*****

    int j;

    //***** Related to reading and storing the informations from the files //*****

    streampos size; //How many bites in the file
    int sizedatacsi; //Number of samples from CsI
    int number_of_files; //How many files will be read
    char * donnees; //Pointer to dynamic memory allocation of bytes after reading of file
    double * yesi; //Pointer to the signal magnitude data for CsI
    double * xcsi; //Pointer to the time data
    int refpoint; //First address from which we find all other addresses; reference point
    char fname[100]; //To hold the name of the file to be read

    //***** String arrays to hold the possible settings used to take the data //*****

    string chan[5] = { "channel 1", "channel 2", "channel 3", "channel 4", "unknown " };
    string coupling[5] = {"DC_50_Ohms", "ground ", "DC 1MOhm ", "ground ", "AC 1MOhm " };

```

```

string onoff[2]      = {"off", "on " } ;
string record[10]   = {"single_sweep      ", "interleaved      ", "histogram      ", "graph      ", "
    filter_coefficient", "complex      ", "extrema      ", "sequence_obsolete ", "centered_RIS      ",
    "peak_detect      " } ;
string processing[8] = {"no_processing", "fir_filter      ", "interpolated ", "sparsed      ", "autoscaled      ", "
    no_result      ", "rolling      ", "cumulative      " } ;

/////////////////////////////////////////////////////////////////
//      Opening and reading of the files      //
/////////////////////////////////////////////////////////////////

int COMM_ORDER ;          //Endianness convention
COMM_ORDER = 0 ;         //Big-endian : set to 0
                        //Little-endian : set to 1

sprintf(fname1, "testCsIsignals\\firstgooddata\\CIMu00124.trc" ) ;
ifstream binarf1 ( fname1, ios::in | ios::binary | ios::ate ) ; //Open the CsI channel 4 file

if ( binarf1.is_open() ){
    size = binarf1.tellg() ;          //To determine the size of the file
    donnees = new char [size] ;      //Dynamic memory to hold the bytes; same size as the file
    binarf1.seekg ( 0, ios::beg) ;
    binarf1.read ( donnees, size ) ; //Assign to dynamic memory accessed with pointer donnees
    binarf1.close() ;
}

refpoint = 0 ;
while ( initpoint( donnees , refpoint )){ //To determine the address from which we determine all other addresses
    refpoint+=1 ;
}

/////////////////////////////////////////////////////////////////
// List of the addresses to reach particular informations //
/////////////////////////////////////////////////////////////////

//int aTEMPLATE_NAME      = refpoint+ 16;
int aCOMM_TYPE            = refpoint+ 32;
//int aCOMM_ORDER         = refpoint+ 34;
int aWAVE_DESCRIPTOR     = refpoint+ 36;          // Length of the descriptor block
int aUSER_TEXT           = refpoint+ 40;          // Length of the usertext block
int aTRIGTIME_ARRAY      = refpoint+ 48;
//int aWAVE_ARRAY_1       = refpoint+ 60;          // Length (in Byte) of the sample array
//int aINSTRUMENT_NAME    = refpoint+ 76;
//int aINSTRUMENT_NUMBER  = refpoint+ 92;
//int aTRACE_LABEL        = refpoint+ 96;
int aWAVE_ARRAY_COUNT     = refpoint+ 116;        // Number of samples
int aVERTICAL_GAIN        = refpoint+ 156;
int aVERTICAL_OFFSET     = refpoint+ 160;
//int aNOMINAL_BITS      = refpoint+ 172;
int aHORIZ_INTERVAL      = refpoint+ 176;
int aHORIZ_OFFSET        = refpoint+ 180;
//int aVERTUNIT           = refpoint+ 196;
//int aHORUNIT            = refpoint+ 244;
int aTRIGGER_TIME        = refpoint+ 296;
//int aRECORD_TYPE       = refpoint+ 316;
//int aPROCESSING_DONE   = refpoint+ 318;
//int aTIMEBASE          = refpoint+ 324;
//int aVERT_COUPLING     = refpoint+ 326;
//int aPROBE_ATT         = refpoint+ 328;
//int aFIXED_VERT_GAIN   = refpoint+ 332;
//int aBANDWIDTH_LIMIT   = refpoint+ 334;

```

```

//int aVERTICAL_VERNIER = reftype+ 336;
//int aACQ_VERT_OFFSET = reftype+ 340;
//int aWAVE_SOURCE      = reftype+ 344;

////////////////////////////////////
//           Accessing the data           //
////////////////////////////////////

/////////          Instrument          //////////

//string INSTRUMENT_NAME   = readstring( donnees , aINSTRUMENT_NAME, 0) ;
//int INSTRUMENT_NUMBER    = readlong( donnees , aINSTRUMENT_NUMBER, 0) ;
//string filename          = fname ;

/////////          Channels          //////////

//string WAVE_SOURCE       = chan[ readword( donnees , aWAVE_SOURCE, COMM_ORDER) ] ;
//string VERT_COUPLING     = coupling[ readword( donnees , aVERT_COUPLING, COMM_ORDER) ] ;
//string BANDWIDTH_LIMIT  = onoff[ readword( donnees , aBANDWIDTH_LIMIT, COMM_ORDER) ] ;
//string RECORD_TYPE      = record[ readword( donnees , aRECORD_TYPE, COMM_ORDER) ] ;
//string PROCESSING_DONE  = processing[ readword( donnees , aPROCESSING_DONE, COMM_ORDER) ] ;

/////////          Vertical settings          //////////

//int FIXED_VERT_GAIN      = readfixed_vert_gain( donnees , aFIXED_VERT_GAIN, COMM_ORDER) ;
//float PROBE_ATT          = readfloat( donnees , aPROBE_ATT, COMM_ORDER) ;
float VERTICAL_GAIN        = readfloat( donnees , aVERTICAL_GAIN, COMM_ORDER) ;
float VERTICAL_OFFSET     = readfloat( donnees , aVERTICAL_OFFSET, COMM_ORDER) ;
//int NOMINAL_BITS        = readword( donnees , aNOMINAL_BITS, COMM_ORDER) ;
//string GAIN_WITH_PROBE   = float_to_eng( FIXED_VERT_GAIN*PROBE_ATT) + "V/div" ;

/////////          Horizontal settings          //////////

float HORIZ_INTERVAL      = readfloat( donnees , aHORIZ_INTERVAL, COMM_ORDER) ;
double HORIZ_OFFSET       = readdouble( donnees , aHORIZ_OFFSET, COMM_ORDER) ;
//string TIMEBASE         = float_to_eng( readtimebase( donnees , aTIMEBASE, COMM_ORDER)) + "s/div" ;
//string SAMPLE_RATE      = float_to_eng( 1/HORIZ_INTERVAL) + "S/sec" ;
//float TS                 = HORIZ_INTERVAL ;
//float FS                 = 1/HORIZ_INTERVAL ;

///////// Read samples array (Plain binary ADC values) //////////

int COMM_TYPE              = readword( donnees , aCOMM_TYPE, COMM_ORDER);
long int WAVE_DESCRIPTOR   = readlong( donnees , aWAVE_DESCRIPTOR, COMM_ORDER);
long int USER_TEXT        = readlong( donnees , aUSER_TEXT, COMM_ORDER);
//long int WAVE_ARRAY_1    = readlong( donnees , aWAVE_ARRAY_1, COMM_ORDER);
long int WAVE_ARRAY_COUNT = readlong( donnees , aWAVE_ARRAY_COUNT, COMM_ORDER);
long int TRIGTIME_ARRAY   = readlong( donnees , aTRIGTIME_ARRAY, COMM_ORDER);

////////////////////////////////////
//           Extracting the data           //
////////////////////////////////////

ycsi = new double[WAVE_ARRAY_COUNT] ; //y (signal magnitude) and x (time) will hold the data
xcsi = new double[WAVE_ARRAY_COUNT] ;

short int temp ;
if ( COMM_TYPE == 0 ) { //Byte precision (1 byte)
for ( j = 0 ; j < WAVE_ARRAY_COUNT ; j++){
temp = readbyte( donnees , reftype + WAVE_DESCRIPTOR + USER_TEXT + TRIGTIME_ARRAY + j ) ;
ycsi[j] = ( VERTICAL_GAIN*temp - VERTICAL_OFFSET ) * (1e3) ;
}
}

```



```
        xcsi[j] = ( HORIZ_INTERVAL*j + HORIZ_OFFSET ) * (1e6) ;
    }
}
else{          //Word precision (2 bytes)
    for ( j = 0 ; j < WAVE_ARRAY_COUNT ; j++ ){
        temp = readword( donnees, refpoint + WAVE_DESCRIPTOR + USER_TEXT + TRIGTIME_ARRAY + (2*j), COMM_ORDER ) ;
        ycsi[j] = ( VERTICAL_GAIN*temp - VERTICAL_OFFSET ) * (1e3) ;
        xcsi[j] = ( HORIZ_INTERVAL*j + HORIZ_OFFSET ) * (1e6) ;
    }
}

//On fait ici ce qu'on veut avec les donnees ycsi, xcsi

delete[] ycsi ;
delete[] xcsi ;

return 0 ;
}
```

Annexe VIII

Code pour l'ajustement non-linéaire d'une somme de 3 gaussiennes sur des signaux produits par un générateur d'impulsions

```
//somme_3gaussiennes_generateur.cpp ;
//Ecrit par Simon Lagrange le 12 fevrier 2016, modifie le 24 aout 2016
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <stdlib.h>
#include <stdio.h>

#include "TROOT.h"
#include "TApplication.h"
#include "TCanvas.h"
#include "TPad.h"

#include "TH1D.h"
#include "TH2D.h"
#include "TGraph.h"
#include "TGraphEditor.h"
#include "TMultigraph.h"
#include "TAxis.h"
#include "TGaxis.h"
#include "TMarker.h"
#include "TLegend.h"
#include "TStyle.h"
#include "TColor.h"
#include "THistPainter.h"

using namespace std ;

#define pi 3.14159265358979323846

////////////////////////////////////
//          Declarations de fonctions          //
////////////////////////////////////

void gaussjordan( double M[][31], double Minv[][31], int n ){
    string quest ;
    int k, m, p, q, w ;
    double factor1 ;
    double factor2 ;
    double factor3 ;

    for ( k = 0 ; k < n ; k++ ){ //Initialisation de la matrice identite
        for ( m = 0 ; m < n ; m++ ){
            if ( k == m ){
                Minv[k][m] = 1 ;
            }
            else {
                Minv[k][m] = 0 ;
            }
        }
    }
}
```

```

}

for ( p = 0 ; p < n - 1 ; p++ ){ //Gauss part
  for ( q = p+1 ; q < n ; q++ ){
    factor1 = M[p][p] ;
    factor2 = M[q][p] ;
    if ( factor2 != 0 ){
      for ( w = 0 ; w < n ; w++ ){
        M[q][w] = M[q][w] * factor1 - M[p][w] * factor2 ;
        Minv[q][w] = Minv[q][w] * factor1 - Minv[p][w] * factor2 ;
      }
    }
    factor3 = M[q][p+1] ;
    if ( factor3 != 0 ){
      for ( w = 0 ; w < n ; w++ ){
        M[q][w] = M[q][w] / factor3 ;
        Minv[q][w] = Minv[q][w] / factor3 ;
      }
    }
  }
}

for ( p = n - 1 ; p > 0 ; p-- ){ //Jordan part
  for ( q = p-1 ; q >= 0 ; q-- ){
    factor1 = M[p][p] ;
    factor2 = M[q][p] ;
    if ( factor2 != 0 ){
      for ( w = n - 1 ; w >= 0 ; w-- ){
        M[q][w] = M[q][w] * factor1 - M[p][w] * factor2 ;
        Minv[q][w] = Minv[q][w] * factor1 - Minv[p][w] * factor2 ;
      }
    }
    factor3 = M[q][p-1] ;
    if ( factor3 != 0 ){
      for ( w = n - 1 ; w >= 0 ; w-- ){
        M[q][w] = M[q][w] / factor3 ;
        Minv[q][w] = Minv[q][w] / factor3 ;
      }
    }
  }
}

for ( p = 0 ; p < n ; p++ ){
  for ( q = 0 ; q < n ; q++ ){
    Minv[p][q] = Minv[p][q] / M[p][p] ;
  }
}

void fitlineaire( int n, int fitstart, double x0, double seuil, double donnees[], double xdonnees[], double M[][31],
  double Minv[][31], double A[], int nbpoint = 0 ){
  //n = puissance du polynome => n+1 parametres/points a utiliser
  string quest ;
  int k, m ;
  int fitpoint ;
  int fitstop ;
  double sums [61] = {} ;
  double Y [31] = {} ;
  fitpoint = fitstart ;
  if ( nbpoint == 0 ){
    fitstop = fitstart + n + 1 ;

```

```

}
else {
    fitstop = fitpoint + nbpoint ;
}
while ( fitpoint < fitstop && donnees[fitpoint] >= seuil ){ //Calcul des sommes pour le fit en x^n
    for ( k = 0 ; k < 2*n + 1 ; k++){
        sums[2*n - k] += pow( xdonnees[fitpoint] - x0, k ) ;
    }
    for ( k = 0 ; k < n + 1 ; k++){
        Y[n - k] += donnees[fitpoint] * pow( xdonnees[fitpoint] - x0, k ) ;
    }

    fitpoint += 1 ;
}

for ( k = 0 ; k < n + 1 ; k++){ //Matrice de base pour le calcul des parametres
    for ( m = k ; m < n + 1 ; m++){
        M[k][m] = sums[k + m] ;
        M[m][k] = M[k][m] ;
    }
}

///////// Calcul de l'inverse de la matrice M avec la methode de Gauss-Jordan ///////////

gaussjordan( M, Minv, n + 1 ) ;

for ( k = 0 ; k < n + 1 ; k++){
    A[k] = 0 ;
    for ( m = 0 ; m < n + 1 ; m++){
        A[k] += Minv[k][m] * Y[m] ;
    }
}

double positionfraction ( int m, double fraction , double donnees[], double xdonnees[], double ampprecis ){
    double pente ;
    double b ;
    pente = donnees[m + 1] - donnees[m] ;
    b = donnees[m] - pente * xdonnees[m] ;
    return ( (fraction * ampprecis) - b ) / pente ;
}

double incertitude ( double moy, double sommequad, int nbvaleur ){
    return sqrt( ( sommequad - pow( moy, 2)*nbvaleur ) / ( nbvaleur - 1 ) ) * pow( 2, -1.5) ;
}

double troisgauss( double a, double b, double c, double mu1, double mu2, double mu3, double sd1, double sd2, double
    sd3, double x ){
    return a*exp( -pow(x-mu1, 2) / (2*pow(sd1, 2)) ) + b*exp( -pow(x-mu2, 2) / (2*pow(sd2, 2)) ) + c*exp( -pow(x-mu3,
    2) / (2*pow(sd3, 2)) ) ;
}

void calculcoeff( char fname[50], double refer[10][33], double erreurmes ){
    int k, p, q, t ;
    double F[32][3] ;
    double S[32][32] ;
    double Sinv[32][32] ;
    double M[31][31] ;
    double Minv[31][31] ;
    double C[10][3][32] ;

```

```

ofstream coeff ( fname, ios::out | ios::trunc | ios::beg );

for ( t = 0 ; t < 10 ; t++){
  for ( p = 0 ; p < 32 ; p++){
    F[p][0] = refer[t][p] ;
    F[p][1] = refer[t][p + 1] - refer[t][p] ;
    F[p][2] = 1 ;
    S[p][p] = 1 / erreurmes ;
    Sinv[p][p] = erreurmes ;
  }

  for ( p = 0 ; p < 3 ; p++){
    for ( q = 0 ; q < 32 ; q++){
      M[p][q] = 0 ;
      for ( k = 0 ; k < 32 ; k++){
        M[p][q] += F[k][p] * F[k][q] * erreurmes ;
      }
    }
  }
}

gaussjordan( M, Minv, 3 ) ;

coeff << "Coefficients de la phase relative t0 = " << t << " ns " << endl ;
for ( p = 0 ; p < 3 ; p++){
  for ( q = 0 ; q < 32 ; q++){
    C[t][p][q] = 0 ;
    for ( k = 0 ; k < 3 ; k++){
      C[t][p][q] += Minv[p][k] * F[q][k] * erreurmes ;
    }
    coeff << C[t][p][q] << ", " ;
  }
  coeff << "\n" ;
}
coeff << "\n" ;
}
coeff.close() ;
}

////////////////////////////////////
//***          ***** Main code *****          ***//
////////////////////////////////////

int main(){

  TApplication theApp("App", 0, 0) ;          //Necessaire pour afficher les graphiques et histogrammes

  //////////////////////////////////////
  //          Definitions des variables          //
  //////////////////////////////////////

  int i, j, k, m, p, q ;                      //Iterateurs
  char fname[100] ;
  string ligne ;                               //Pour prendre les lignes du fichiers , une a la fois
  string mot ;                                 //Un des mots contenus dans une ligne
  string vramot ;                             //Seuls les 4 premiers caracteres d'un mot forme une mesure , un vrai mot
  string quest ;                              //Pour donner le temps de voir les graphiques produits
  double donnees [140] ;                     //Liste des donnees d'un signal particulier
  double donneesbrute[140] ;
  double xdonnees [140] ;                    //Contient les valeurs en x pour mettre les donnees en graphiques
  double xtemps [140] ;                      //Contient les valeurs en x convertis en temps ns pour mettre les donnees
  en graphiques

```

```

double temps ;
int marqueur = 0 ; //Indique qu'on extrait un signal des lignes suivantes dans la lecture
int index = 0 ; //Pour un signal, combien de valeurs sont deja extraites
int nbsignaux = 0 ; //Combien de signaux sont consideres
int nbutile[10] = {} ;
double piedestal ; //Valeur du piedestal
double seuil ; //Valeur du seuil
double seuil2 [10] = { 870, 800, 760, 710, 680, 650, 620, 590, 560, 530 } ;
double sigmaamp ;
double sigma, sigma2, sigma3, sigma4 ;
int pointseuil ;
int pointseuil3 ;
double pentetest ;
double pointinit ;
int maxpos ;
int fitstart = 0 ;
double M [31][31] = {} ;
double Minv [31][31] = {} ;
double A [31] ;
double Y [60] ;//[5] ;
double x ;
double maxprecis ;
double ampprecis ;
double ampprecismax = 0 ;
double amplimoy = 0, amplimoy2 = 0 ;
double nbampli = 0 ;
double phase ;
double phasefit ;
double risetime ;
double risemoy = 0, risemoy2 = 0 ;
double risemoypic1 = 0, risemoypic12 = 0 ;
double realrise ;
double realrisemoy = 0, realrisemoy2 = 0 ;
int nbrealrise = 0 ;
int nbrise = 0 ;
double refer [10][33] = {} ;
double refer1 [33] = {}, refer2 [33] = {}, refer3 [33] = {}, refer4 [33] = {}, refer5 [33] = {}, refer6 [33] = {},
refer7 [33] = {}, refer8 [33] = {}, refer9 [33] = {}, refer10 [33] = {} ;
double refertot [10][33] ;
double xrefer [33] ;
int referindex ;
int ordre [10] = {} ;
int discarte = 0 ;
string discartequi ;
int penteneg = 0 ;
int pentepos = 0 ;
int interference = 0 ;
int sousseuil = 0 ;
int nbutiletotale = 0 ;
double erreurmes ;
double C[10][3][32] = {} ;
double derivee ;
double x1, x2 ;
double cfd = 0.10 ;
double pentereal ;
double breal ;
double xcfd ;
double cfdrise ;
double cfdrisemoy = 0, cfdrisemoy2 = 0 ;
double jacob[60][10] ;
double jtj[31][31] ;

```

```

double jtjdiag[31][31] = {} ;
double jtjdelta[31] ;
int nblm = 0 ;
double lamb ;
double vlamb ;
int nbiter ;
double res ;
double resmin ;
double LM[31][31] ;
double LMPrec[31][31] ;
double LMinv[31][31] ;
double P[10] ;
double Pprec[10] ;
double deltaparam[10] ;
double tzero ;
int nbparam ;
int nbpoints ;
double eps1 , eps2 , eps3 , epsrho ;
double rho ;
double rhodenom ;
double Pmax ;
double retest , retestmoy = 0 , retestmoy2 = 0 ;
double past = 0.01 ;

double ampfit , a , b , c ;
double sd1 , sd2 , sd3 ;
double mu1 , mu2 , mu3 ;
double sigmaampfit , sigmaa , sigmab , sigmac , sigmaretest ;
double sigmaampratio ;
double sigmasd1 , sigmasd2 , sigmasd3 ;
double sigmamu1 , sigmamu2 , sigmamu3 ;
double ampfitmoy = 0 , almmoy = 0 , blmmoy = 0 , clmmoy = 0 ;
double sd1lmmoy = 0 , sd2lmmoy = 0 , sd3lmmoy = 0 ;
double mu1lmmoy = 0 , mu2lmmoy = 0 , mu3lmmoy = 0 ;
double ampfitmoy2 = 0 , almmoy2 = 0 , blmmoy2 = 0 , clmmoy2 = 0 ;
double sd1lmmoy2 = 0 , sd2lmmoy2 = 0 , sd3lmmoy2 = 0 ;
double mu1lmmoy2 = 0 , mu2lmmoy2 = 0 , mu3lmmoy2 = 0 ;
double ampratiomoy = 0 , ampratiomoy2 = 0 ;
double pentecorrection ;
double bcorrection ;
double pointzero ;
double shapefit[10][33] = {} ;

int debutref = 7 ;
int pic1 = 0 ;

int posdepart ; //Point de depart pour les ajustements
int nbpire = 0 ;
int trop_bas ;
double lambinit ;

int nblmused = 0 ;

for ( j = 0 ; j < 140 ; j++ ){
    xdonnees[j] = j + 1 ;
    xtemps[j] = xdonnees[j] * 10 ;
}

for ( j = 0 ; j < 10 ; j++ ){
    seuil2[j] = seuil2[j] * 1000 / pow( 2 , 14 ) ; //Conversion en mV
}

```

```

////////////////////////////////////
//          Lecture du fichier          //
////////////////////////////////////

//ifstream pedtemp ( "pedestal_CsI.txt", ios::in | ios::binary | ios::beg ); //Stocke les valeurs pour le
//calcul du pedestal et de son sigma
ifstream pedtemp ( "pedestal_generateur.txt", ios::in | ios::binary | ios::beg ); //Stocke les valeurs pour le
//calcul du pedestal et de son sigma
if ( pedtemp.is_open() ){
    getline( pedtemp, ligne );
    pedestal = stod( ligne );
    getline( pedtemp, ligne );
    sigma = stod( ligne );
    seuil = 5 * sigma * 1000 / pow( 2, 14 );
    erreurmes = sigma * 1000 / pow( 2, 14 );
    pedtemp.close();
    cout << "Pedestal : " << pedestal * 1000 / pow( 2, 14 ) << " mV " << endl;
    cout << "Seuil : " << seuil << " mV " << endl << endl;

//ifstream adcfile ( "data\\CsI_bon_output_trie.log" ); //On ouvre le fichier
ifstream adcfile ( "data\\generateur_impulsions\\pulser_phases_non_triees\\pulser_signals_0_10.log" ); //On
//ouvre le fichier

THID *phaserel = new THID("phaserel", /*"Distribution des phases relatives entre les donnees et l'echantillonnage
de l'ADC*/" ; Phases relatives (ns); Decompte", 21, -10.5, 10.5 );
THID *phaserel2 = new THID("phaserel2", /*"Distribution des phases relatives entre les donnees et l'
echantillonnage de l'ADC*/" ; Phases relatives (ns); Decompte", 100, -10.5, 10.5 );
THID *phaserel3 = new THID("phaserel3", /*"Distribution 2 des phases relatives entre les donnees et l'
echantillonnage de l'ADC*/" ; Phases relatives (ns); Decompte", 40, -20, 20 );
THID *phaserel4 = new THID("phaserel4", /*"Distribution 2 des phases relatives entre les donnees et l'
echantillonnage de l'ADC*/" ; Phases relatives (ns); Decompte", 2000, -20, 20 );
THID *phaserelgauss = new THID("phaserelgauss", /*"Distribution des phases relatives entre les donnees et l'
echantillonnage de l'ADC pour l'ajustement d'une gaussienne*/" ; Phases relatives (ns); Decompte", 21,
-10.5, 10.5 );
THID *phaserelgauss2 = new THID("phaserelgauss2", /*"Distribution des phases relatives entre les donnees et l'
echantillonnage de l'ADC pour l'ajustement d'une gaussienne*/" ; Phases relatives (ns); Decompte", 600,
-10.5, 10.5 );
THID *risegr = new THID("risegr", /*"Distribution des peaking times*/" ; Temps d'atteinte du pic (ns); Decompte
", 200, 110, 120 ) ;//1000, 100, 130 ) ;//
THID *realrisetime = new THID("realrisetime", /*"Distribution des temps de montee*/" ; Temps de montee (ns);
Decompte", 200, 70, 85 ) ;//1000, 70, 90 ) ;//
THID *cfdrisetime = new THID("cfdrise", /*"Distribution des temps de montee par Constant Fraction Discriminator
*/" ; Temps de montee (ns); Decompte", 200, 100, 110 ) ;//1000, 90, 120 ) ;//
THID *amplimes = new THID("amplimes", /*"Distribution des amplitudes mesurees*/" ; Amplitudes (mV); Decompte",
200, 0, 500 ) ;//200, 0, 10000 ) ;
THID *amplimes2 = new THID("amplimes2", /*"Distribution des amplitudes mesurees*/" ; Amplitudes (mV); Decompte",
200, 400, 450 ) ;//1000, 6700, 7100 ) ;
THID *amplimod = new THID("amplimod", /*"Distribution des amplitudes selon le polynome en x^2*/" ; Amplitudes (
mV); Decompte", 200, 0, 500 ) ;//200, 0, 10000 ) ;
THID *amplimod2 = new THID("amplimod2", /*"Distribution des amplitudes selon le polynome en x^2*/" ; Amplitudes
(mV); Decompte", 200, 400, 450 ) ;//1000, 6700, 7100 ) ;
THID *amplimodgauss = new THID("amplimodgauss", /*"Distribution des amplitudes pour l'ajustement d'une
gaussienne*/" ; Amplitudes (mV); Decompte", 200, 0, 500 ) ;//200, 0, 10000 ) ;
THID *amplimodgauss2 = new THID("amplimodgauss2", /*"Distribution des amplitudes pour l'ajustement d'une
gaussienne*/" ; Amplitudes (mV); Decompte", 200, 400, 450 ) ;//1000, 6700, 7100 ) ;
THID *ampfitgr = new THID("ampfitgr", /*"Distribution d'amplitude pour l'ajustement d'une somme de 3 gaussiennes
*/" ; Amplitude (mV); Decompte", 200, 400, 440 ) ;//200, 0, 10000 ) ;//5000, 6500, 7500 ) ;//
THID *alm = new THID("alm", /*"Distribution du parametre a pour l'ajustement d'une somme de 3 gaussiennes*/" ; a
(sans unite); Decompte", 1000, 0, 1 ) ;//3000, 2000, 3000 ) ;//

```



```

TH1D *blm = new TH1D("blm", /*"Distribution du parametre b pour l'ajustement d'une somme de 3 gaussiennes*/" ; b
(sans unite); Decompte", 1000, 0, 1 ) ;//4000, 4000, 5500 ) ;//
TH1D *clm = new TH1D("clm", /*"Distribution du parametre c pour l'ajustement d'une somme de 3 gaussiennes*/" ; c
(sans unite); Decompte", 1000, 0, 1 ) ;//4000, 800, 2200 ) ;//
TH1D *mullm = new TH1D("mullm", /*"Distribution du parametre mul pour l'ajustement d'une somme de 3 gaussiennes
*/" ; mu1 (ns); Decompte", 500, 270, 300 ) ;
TH1D *mu2lm = new TH1D("mu2lm", /*"Distribution du parametre mu2 pour l'ajustement d'une somme de 3 gaussiennes
*/" ; mu2 (ns); Decompte", 500, 315, 350 ) ;
TH1D *mu3lm = new TH1D("mu3lm", /*"Distribution du parametre mu3 pour l'ajustement d'une somme de 3 gaussiennes
*/" ; mu3 (ns); Decompte", 500, 370, 420 ) ;//1000, 360, 430 ) ;//
TH1D *sd1lm = new TH1D("sd1lm", /*"Distribution du parametre sigma1 pour l'ajustement d'une somme de 3
gaussiennes*/" ; sigma1 (ns); Decompte", 500, 30, 45 ) ;
TH1D *sd2lm = new TH1D("sd2lm", /*"Distribution du parametre sigma2 pour l'ajustement d'une somme de 3
gaussiennes*/" ; sigma2 (ns); Decompte", 500, 40, 55 ) ;
TH1D *sd3lm = new TH1D("sd3lm", /*"Distribution du parametre sigma3 pour l'ajustement d'une somme de 3
gaussiennes*/" ; sigma3 (ns); Decompte", 500, 50, 75 ) ;//1000, 50, 80 ) ;//
TH1D *restestgr = new TH1D("chi2fit", /*"Distribution du chi2 moyen pour l'ajustement d'une somme de 3
gaussiennes*/" ; Chi2; Decompte", 500, 0, 20 ) ;
TH1D *nbitergr = new TH1D("nbiter", /*"Distribution du nombre d'iterations pour l'ajustement d'une somme de 3
gaussiennes*/" ; Nb d'iterations; Decompte", 200, 0, 200 ) ;
TH1D *mu2lgr = new TH1D("mu2l", /*"Distribution des differences mu2 - mu1 pour l'ajustement d'une somme de 3
gaussiennes*/" ; mu2 - mu1 (ns); Decompte", 250, 40, 60 ) ;
TH1D *mu32gr = new TH1D("mu32", /*"Distribution des differences mu3 - mu2 pour l'ajustement d'une somme de 3
gaussiennes*/" ; mu3 - mu2 (ns); Decompte", 300, 40, 80 ) ;//800, 20, 110 ) ;//
TH1D *phasefitgr = new TH1D("phasefit", /*"Distribution des phases relatives entre les donnees et l'
echantillonnage de l'ADC, selon la somme de 3 gaussiennes*/" ; Phases relatives (ns); Decompte", 21, -10.5,
10.5 ) ;
TH1D *phasefitgr2 = new TH1D("phasefit2", /*"Distribution des phases relatives entre les donnees et l'
echantillonnage de l'ADC, selon la somme de 3 gaussiennes*/" ; Phases relatives (ns); Decompte", 100, -10.5,
10.5 ) ;
TH1D *ampratiogr = new TH1D("ampratiogr", /*"Distribution du ratio des amplitudes du fit par polynome et de la
somme de 3 gaussiennes*/" ; Ratio A_poly/A_3gauss; Decompte", 200, 0.99, 1.015 ) ;
TH1D *mul2somme = new TH1D("mul2somme", /*"Distribution des sommes mul + mu2 pour l'ajustement d'une somme de 3
gaussiennes*/" ; mul + mu2 (ns); Decompte", 300, 580, 650 ) ;
TH1D *mul23somme = new TH1D("mul23somme", /*"Distribution des sommes mul + mu2 + mu3 pour l'ajustement d'une
somme de 3 gaussiennes*/" ; mul + mu2 + mu3 (ns); Decompte", 400, 950, 1070 ) ;

TH2D *phaserise = new TH2D("phaserise", /*"Distribution des phases relatives selon les peaking time*/" ; Phase
relative (ns); Temps d'atteinte du pic (ns); Decompte", 21, -10.5, 10.5, 1000, 90, 115 ) ;//1000, 90, 130 )
;//
TH2D *ampphase = new TH2D("ampphase", /*"Distribution des phases relatives selon les amplitudes donnees par les
polynomes*/" ; Phase relative (ns); Amplitude (mV); Decompte", 21, -10.5, 10.5, 200, 0, 500 ) ;//200, 0,
10000 ) ;//800, 6700, 7100 ) ;//
TH2D *ampphasegauss = new TH2D("ampphasegauss", /*"Distribution des phases relatives selon les amplitudes pour l'
ajustement d'une gaussienne*/" ; Phase relative (ns); Amplitude (mV); Decompte", 21, -10.5, 10.5, 200,
400, 450 ) ;//800, 6700, 7100 ) ;
TH2D *amprise = new TH2D("amprise", /*"Distribution des peaking time selon les amplitudes donnees par les
polynomes*/" ; Temps d'atteinte du pic (ns); Amplitude (mV); Decompte", 1000, 90, 115, 200, 0, 500 )
;//200, 0, 10000 ) ;//1000, 90, 130, 800, 6700, 7100 ) ;//
TH2D *amprise2 = new TH2D("amprise2", /*"Distribution du temps de montee selon l'amplitude du premier point au-
dessus du seuil*/" ; Temps de montee (ns); Amplitude (mV); Decompte", 1000, 60, 100, 3000, 40, 500 ) ;
TH2D *amprise3 = new TH2D("amprise3", /*"Distribution du temps de montee selon l'amplitude du deuxieme point au-
dessus du seuil*/" ; Temps de montee (ns); Amplitude (mV); Decompte", 1000, 60, 100, 8000, 200, 800 ) ;
TH2D *phaserise2 = new TH2D("phaserise2", /*"Distribution des phases relatives selon l'amplitude du premier
point au-dessus du seuil*/" ; Phase relative (ns); Amplitude (mV); Decompte", 21, -10.5, 10.5, 3000, 40,
500 ) ;
TH2D *phaserise3 = new TH2D("phaserise3", /*"Distribution des phases relatives selon l'amplitude du deuxieme
point au-dessus du seuil*/" ; Phase relative (ns); Amplitude (mV); Decompte", 21, -10.5, 10.5, 8000, 200,
800 ) ;
TH2D *phaserise4 = new TH2D("phaserise4", /*"Distribution des phases relatives selon l'amplitude du troisieme
point au-dessus du seuil*/" ; Phase relative (ns); Amplitude (mV); Decompte", 21, -10.5, 10.5, 8000, 400,

```

```

1400 ) ;
TH2D *phaserearise = new TH2D("phaserearise", /*"Distribution des phases relatives selon les temps de montee
*/" ; Phase relative (ns); Temps de montee (ns); Decompte", 21, -10.5, 10.5, 1000, 60, 100 ) ;
TH2D *phasecfdrise = new TH2D("phasecfdrise", /*"Distribution des phases relatives selon les temps de montee par
Constant Fraction Discriminator*/" ; Phase relative (ns); Temps de montee (ns); Decompte", 21, -10.5,
10.5, 1000, 80, 120 ) ;
TH2D *ampligausspoint = new TH2D("ampligausspoint", /*"Distribution des amplitudes selon le nombre de points
utilises pour l'ajustement d'une gaussienne*/" ; Amplitude (mV); Nombre de points; Decompte", 200, 400,
450, 30, 2.5, 32.5 ) ;//800, 6700, 7100, 30, 2.5, 32.5 ) ;
TH2D *ampphaseru = new TH2D("ampphaseru", /*"Distribution des phases relatives selon les amplitudes, d'apres la
fonction russe*/" ; Phase relative (ns); Amplitude (mV); Decompte", 21, -10.5, 10.5, 200, 400, 450 )
; //800, 6700, 7100 ) ;
TH2D *phasevsphaseru = new TH2D("phasevsphaseru", /*"Distribution des phases relatives d'apres un polynome de
puissance 4 selon les phases relatives d'apres la fonction russe*/" ; Phase relative (ns); Phase relative
russe (ns); Decompte", 21, -10.5, 10.5, 21, -10.5, 10.5 ) ;
TH2D *mu1seuil = new TH2D("mu1seuil", /*"Distribution des parametres mu1 selon l'amplitude du premier point au-
dessus du seuil*/" ; mu1 (ns); Amplitude (mV); Decompte", 300, 240, 320, 800, 0, 0.2 ) ;//3000, 40, 500 )
; //
TH2D *mu2seuil = new TH2D("mu2seuil", /*"Distribution des parametres mu2 selon l'amplitude du premier point au-
dessus du seuil*/" ; mu2 (ns); Amplitude (mV); Decompte", 300, 300, 360, 800, 0, 0.2 ) ;//3000, 40, 500 )
; //
TH2D *mu3seuil = new TH2D("mu3seuil", /*"Distribution des parametres mu3 selon l'amplitude du premier point au-
dessus du seuil*/" ; mu3 (ns); Amplitude (mV); Decompte", 300, 360, 500, 800, 0, 0.2 ) ;//1000, 360, 430,
3000, 40, 500 ) ; //
TH2D *phasecompar = new TH2D("phasecompar", /*"Comparaison des phases relatives obtenues par polynomes avec les
phases selon la somme de 3 gaussiennes*/" ; Phase relative - polynomes (ns); Phase relative - gaussiennes (
ns); Decompte", 21, -10.5, 10.5, 21, -10.5, 10.5 ) ;
TH2D *ampcompar = new TH2D("ampcompar", /*"Comparaison des amplitudes obtenues par polynomes avec les amplitudes
selon la somme de 3 gaussiennes*/" ; Amplitude - polynomes (mV); Amplitude - gaussiennes (mV); Decompte",
1000, 0, 10000, 1000, 0, 10000 ) ;//1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitchi2 = new TH2D("amplitchi2", /*"Amplitudes selon les chi^2 individuels pour la somme de 3
gaussiennes*/" ; Chi^2 individuels; Amplitude (mV); Decompte", 40, 0, 20, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplita = new TH2D("amplita", /*"Amplitudes selon les valeurs du parametres a pour la somme de 3
gaussiennes*/" ; a (sans unite); Amplitude (mV); Decompte", 3000, 0, 1, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitb = new TH2D("amplitb", /*"Amplitudes selon les valeurs du parametres b pour la somme de 3
gaussiennes*/" ; b (sans unite); Amplitude (mV); Decompte", 3000, 0, 1, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitc = new TH2D("amplitc", /*"Amplitudes selon les valeurs du parametres c pour la somme de 3
gaussiennes*/" ; c (sans unite); Amplitude (mV); Decompte", 3000, 0, 1, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitmu1 = new TH2D("amplitmu1", /*"Amplitudes selon les valeurs du parametres mu1 pour la somme de 3
gaussiennes*/" ; mu1 (ns); Amplitude (mV); Decompte", 1000, 240, 320, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitmu2 = new TH2D("amplitmu2", /*"Amplitudes selon les valeurs du parametres mu2 pour la somme de 3
gaussiennes*/" ; mu2 (ns); Amplitude (mV); Decompte", 1000, 300, 360, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitmu3 = new TH2D("amplitmu3", /*"Amplitudes selon les valeurs du parametres mu3 pour la somme de 3
gaussiennes*/" ; mu3 (ns); Amplitude (mV); Decompte", 1000, 360, 500, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitsd1 = new TH2D("amplitsd1", /*"Amplitudes selon les valeurs du parametres sd1 pour la somme de 3
gaussiennes*/" ; sd1 (ns); Amplitude (mV); Decompte", 1000, 20, 50, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitsd2 = new TH2D("amplitsd2", /*"Amplitudes selon les valeurs du parametres sd2 pour la somme de 3
gaussiennes*/" ; sd2 (ns); Amplitude (mV); Decompte", 1000, 30, 60, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //
TH2D *amplitsd3 = new TH2D("amplitsd3", /*"Amplitudes selon les valeurs du parametres sd3 pour la somme de 3
gaussiennes*/" ; sd3 (ns); Amplitude (mV); Decompte", 1000, 50, 100, 200, 400, 450 ) ;//50, 0, 10000 )
; //1000, 6800, 7100, 1000, 6800, 7100 ) ; //

```

```

TH2D *amplitmu12 = new TH2D("amplitmu12", /*"Amplitudes selon la somme des parametres mu1 + mu2 pour la somme de
3 gaussiennes*/" ; mu1 + mu2 (ns); Amplitude (mV); Decompte", 500, 500, 700, 200, 400, 450 ) ;//50, 0,
10000 ) ;//1000, 6800, 7100, 1000, 6800, 7100 ) ;//
TH2D *amplitmu123 = new TH2D("amplitmu123", /*"Amplitudes selon la somme des parametres mu1 + mu2 + mu3 pour la
somme de 3 gaussiennes*/" ; mu1 + mu2 + mu3 (ns); Amplitude (mV); Decompte", 900, 800, 1200, 200, 400, 450
) ;//50, 0, 10000 ) ;//1000, 6800, 7100, 1000, 6800, 7100 ) ;//

if ( adcfile.is_open() ){
while ( getline( adcfile , ligne ) ){
if ( marqueur == 0 && ligne.length() == 54 && ligne[0] == '8' && ligne[9] == '9' && ligne[18] == 'a' &&
ligne[27] == 'b' && ligne[36] == 'b' && ligne[45] == 'c' ){
marqueur = 1 ; //On se prepare a lire les donnees pour un signal
}
else if ( marqueur == 1 ){
for ( i = 0 ; i < ligne.length() ; i++){
if ( ligne[i] != ' ' ){
mot += ligne[i] ;
}
else if ( ligne[i] == ' ' ){
if ( ( mot[0] != 'e' ) && ( mot[0] != 'c' ) ){ //&& mot[1] != ( '0' || '2' || '3' || '4' ) ){
for ( j = 0 ; j < 4 ; j++ ){ //Seuls les 4 premiers caracteres nous interessent
vraimot += mot[j] ;
}
donnees[index] = ( stoi( vraimot , nullptr , 16 ) - piedestal ) * 1000 / pow( 2 , 14 ) ;
donneesbrute[index] = donnees[index] ;
index += 1 ;
}
}
else if ( mot[0] == 'e' && mot[1] == '0' ){ //On atteint la fin du signal , on peut l'analyser dans
ce bloc de commandes
marqueur = 0 ;
index = 0 ;

nbsignaux += 1 ;

//////////////////////////////////////
// Scan du signal //
//////////////////////////////////////

pointseuil = -1 ; //Devient positif lorsqu'on trouve un signal
for ( j = 0 ; j < 130; j++ ){
if ( donnees[j] >= seuil && donnees[j+10] >= seuil && j > 0 ){ //On depasse le seuil
pointseuil = j ;

pentetest = donnees[pointseuil + 6] - donnees[pointseuil + 5] ;
if ( pentetest <= 0 ){
penteneg += 1 ;
}

if ( pentetest > 0 ){
pentepos += 1 ;

/*k = 0 ;
trop_bas = 0 ;
while ( donnees[k] > ( 7500 - piedestal ) * 1000 / pow( 2 , 14 ) && k < 15 ){
k += 1 ;
}
if ( k < 15 ){
trop_bas = 1 ; //Pour couper les interferences avec le CsI
}*/

while ( donnees[j] <= donnees[j + 1] && j < 140 ){ //On cherche le max

```

```

j += 1 ;
}

//if ( donnees[j] > 11000 - piedestal && trop_bas == 0 ){ //Seuils d'amplitude , CsI
if ( donnees[j] > 5000 * 1000 / pow( 2, 14 ) ){ //Seuil d'amplitude , generateur

////////////////////////////////////
// Calcul des parametres optimaux pour polynomes //
////////////////////////////////////

amplimes->Fill( donnees[j] ) ;
amplimes2->Fill( donnees[j] ) ;

maxpos = j ;
for ( k = 2 ; k < 3 ; k++){ //Boucle sur les differentes puissances k testees; fixe a 2
    fitstart = maxpos - ( k / 2 ) ;
    if ( fitstart < pointseuil ){
        fitstart = pointseuil ;
    }

    fitlineaire( k, fitstart, xdonnees[maxpos], seuil, donneesbrute, xdonnees, M, Minv, A ) ;

////////////////////////////////////
// Maximum et sa position selon le modele en x^2 //
////////////////////////////////////

x = xdonnees[maxpos - 1] ;
do {
    x += 0.005 ;
    derivee = 0 ;
    for ( p = 0 ; p < k ; p++){
        derivee += ( k - p ) * A[p] * pow( x - xdonnees[maxpos], k - p - 1 ) ;
    }
} while ( derivee > 0 && x < xdonnees[maxpos] + 1 ) ;

if ( derivee <= 0 ){
    maxprecis = x ;
    ampprecis = 0 ;
    for ( p = 0 ; p < k + 1 ; p++){
        ampprecis += A[p] * pow( x - xdonnees[maxpos], k - p ) ;
    }
    if ( ampprecis > ampprecismax ){
        ampprecismax = ampprecis ;
    }
}
amplimod->Fill( ampprecis ) ;
amplimod2->Fill( ampprecis ) ;
amplimoy += ampprecis ;
amplimoy2 += pow( ampprecis, 2 ) ;
nbampli += 1 ;

////////////////////////////////////
// Calcul de phases relatives //
////////////////////////////////////

phase = 10 * ( maxprecis - xdonnees[maxpos] ) ;
if ( phase < -5 ){
    phase += 10 ;
}
else if ( phase >= 5 ){
    phase -= 10 ;
}
}

```

```

referindex = floor( phase ) + 5 ;

phaserel->Fill( phase - 0.5 ) ;
phaserel2->Fill( phase - 0.5 ) ;
ampphase->Fill( phase - 0.5, ampprecis ) ;

////////////////////////////////////
//                               Temps de montee                               //
////////////////////////////////////

m = maxpos ;
while ( donneesbrute[m] > 0.90 * ampprecis ){
  m -= 1 ;
}
x2 = positionfraction ( m, 0.90, donneesbrute, xdonnees, ampprecis ) ;
while ( donneesbrute[m] > 0.10 * ampprecis ){
  m -= 1 ;
}
x1 = positionfraction ( m, 0.10, donneesbrute, xdonnees, ampprecis ) ;
nbrealrise += 1 ;
realrise = ( x2 - x1 ) * 10 ;
realrisemoy += realrise ;
realrisemoy2 += pow( realrise, 2 ) ;
realrisetime->Fill( realrise ) ;
phaserealrise->Fill( phase, realrise ) ;

////////////////////////////////////
//                               Constant fraction discriminator                               //
////////////////////////////////////

m = maxpos ;
while ( donneesbrute[m] > cfd * ampprecis ){
  m -= 1 ;
}

xcfd = positionfraction ( m, cfd, donneesbrute, xdonnees, ampprecis ) ;
cfdrise = ( maxprecis - xcfd ) * 10 ;
cfdrisemoy += cfdrise ;
cfdrisemoy2 += pow( cfdrise, 2 ) ;
cfdrisetime->Fill( cfdrise ) ;
phasecfdrise->Fill( phase, cfdrise ) ;

////////////////////////////////////
//                               Determiner le (3e) point seuil utilise pour les moyennes                               //
////////////////////////////////////

pointseuil3 = pointseuil ; //On cherche le 3e point au-dessus du seuil pour une phase
particuliere
while ( donneesbrute[pointseuil3] < seuil2[referindex] ){
  pointseuil3 += 1 ;
}

////////////////////////////////////
//                               Normalisation du signal et point initial                               //
////////////////////////////////////

for ( p = 0 ; p < 140 ; p++){
  donnees[p] = donneesbrute[p] / ampprecis ;
}

pentereal = 0.80 * ampprecis / ( x2 - x1 ) ;

```

```

breal = 0.90 * ampprecis - ( pentereal * x2 ) ;
pointinit = - breal / pentereal ;

phaserise2->Fill( phase - 0.5, donneesbrute[pointseuil3 - 2 ] ) ;
phaserise3->Fill( phase - 0.5, donneesbrute[pointseuil3 - 1 ] ) ;
phaserise4->Fill( phase - 0.5, donneesbrute[pointseuil3] ) ;

////////////////////////////////////
//                               Calcul du peaking time                               //
////////////////////////////////////

risetime = ( maxprecis - pointinit ) * 10 ;           //Exprime en ns
nbrise += 1 ;
risemoy += risetime ;
risemoy2 += pow( risetime , 2 ) ;
risegr->Fill( risetime ) ;

phaserise->Fill( phase - 0.5, risetime ) ;
amprise->Fill( risetime , ampprecis ) ;

////////////////////////////////////
// Ajustements non-lineaires Levenberg-Marquardt //
////////////////////////////////////

nblm += 1 ;
posdepart = pointseuil3 - 3 ;
tzero = xtemps[posdepart] ;           //Exprime en ns
a = 2500 * 1000 / pow( 2, 14 ) ;
b = 4900 * 1000 / pow( 2, 14 ) ;
c = 1500 * 1000 / pow( 2, 14 ) ;
sd1 = 35 ;
sd2 = 45 ;
sd3 = 60 ;
mu1 = 280 ;
mu2 = 330 ;
mu3 = 400 ;

lambinit = 100000 ;
lamb = lambinit ;
vlamb = 1.2 ;
eps1 = pow( 10, -6 ) ;
eps2 = pow( 10, -6 ) ;
eps3 = 1 ;           //Condition sur le chi^2 normalise
eprho = 0 ;
nbiter = 0 ;
nbpire = 0 ;
nbpoints = 32 ;
nbparam = 9 ;

do {           //Boucle pour l'algorithm de Levenverg-Marquardt
res = 0 ;

for ( p = 0 ; p < nbpoints ; p++ ){

////////////////////////////////////
//                               Somme de 3 gaussiennes                               //
////////////////////////////////////

//Parametres : a, b, c, mu1, mu2, mu3, sd1, sd2, sd3
jacob[p][0] = exp( - pow(xtemps[posdepart + p] - mu1, 2) / ( 2 * pow( sd1, 2 ) ) ) ;
jacob[p][1] = exp( - pow(xtemps[posdepart + p] - mu2, 2) / ( 2 * pow( sd2, 2 ) ) ) ;

```

```

jacob[p][2] = exp( - pow(xtemps[posdepart + p] - mu3, 2) / (2 * pow( sd3, 2 )) ) ;
jacob[p][3] = a * ( xtemps[posdepart + p] - mu1 ) * jacob[p][0] / pow( sd1, 2 ) ;
jacob[p][4] = b * ( xtemps[posdepart + p] - mu2 ) * jacob[p][1] / pow( sd2, 2 ) ;
jacob[p][5] = c * ( xtemps[posdepart + p] - mu3 ) * jacob[p][2] / pow( sd3, 2 ) ;
jacob[p][6] = ( xtemps[posdepart + p] - mu1 ) * jacob[p][3] / sd1 ;
jacob[p][7] = ( xtemps[posdepart + p] - mu2 ) * jacob[p][4] / sd2 ;
jacob[p][8] = ( xtemps[posdepart + p] - mu3 ) * jacob[p][5] / sd3 ;

Y[p] = donneesbrute[posdepart + p] - a*jacob[p][0] - b*jacob[p][1] - c*jacob[p][2] ;
res += pow( Y[p], 2 ) ;
}

for ( m = 0 ; m < nbparam ; m++){
  for ( q = 0 ; q < nbparam ; q++){
    jtj[m][q] = 0 ;
    for ( p = 0 ; p < nbpoints ; p++){
      jtj[m][q] += jacob[p][m] * jacob[p][q] ;
    }
    if ( m == q ){
      jtjdiag[m][q] = jtj[m][q] ;
    }
    LM[m][q] = ( 1 * jtj[m][q] ) + ( lamb * jtjdiag[m][q] ) ;
  }
}

for ( m = 0 ; m < nbparam ; m++){
  P[m] = 0 ;
  for ( p = 0 ; p < nbpoints ; p++){
    P[m] += jacob[p][m] * Y[p] ; //jT * Y
  }
}

if ( nbiter == 0 ){
  resmin = res ;
  lamb = lamb / vlamb ;
  for ( m = 0 ; m < nbparam ; m++){
    Pprec[m] = P[m] ;
    for ( q = 0 ; q < nbparam ; q++){
      LMprec[m][q] = LM[m][q] ;
    }
  }
}
else{
  for ( m = 0 ; m < nbparam ; m++){
    jtjdelta[m] = 0 ;
    for ( q = 0 ; q < nbparam ; q++){
      jtjdelta[m] += jtjdiag[m][q] * deltaparam[q] ;
    }
  }

  rhodenom = 0 ;
  for ( m = 0 ; m < nbparam ; m++){
    rhodenom += deltaparam[m] * ( lamb * jtjdelta[m] + Pprec[m] ) ;
  }
  rho = ( resmin - res ) / rhodenom ;

  if ( rho > epsrho ){
    for ( m = 0 ; m < nbparam ; m++){
      Pprec[m] = P[m] ;
      for ( q = 0 ; q < nbparam ; q++){
        LMprec[m][q] = LM[m][q] ;
      }
    }
  }
}

```

```

    }
  }
  resmin = res ;
  lamb = lamb / vlamb ;
  nbpire = 0 ;
}
else {
  a -= deltaparam[0] ;
  b -= deltaparam[1] ;
  c -= deltaparam[2] ;
  mu1 -= deltaparam[3] ;
  mu2 -= deltaparam[4] ;
  mu3 -= deltaparam[5] ;
  sd1 -= deltaparam[6] ;
  sd2 -= deltaparam[7] ;
  sd3 -= deltaparam[8] ;

  lamb = lamb * vlamb ;
  nbpire += 1 ;
  for ( m = 0 ; m < nbparam ; m++ ){
    P[m] = Pprec[m] ;
    for ( q = 0 ; q < nbparam ; q++ ){
      LM[m][q] = LMprec[m][q] ;
    }
  }
}
}

gaussjordan( LM, LMinv, nbparam ) ;

for ( m = 0 ; m < nbparam ; m++ ){
  deltaparam[m] = 0 ;
  for ( p = 0 ; p < nbparam ; p++ ){
    deltaparam[m] += LMinv[m][p] * P[p] ; //( jtj + lamb * jtjdiag )^-1 * jT * Y
  }
}

a += deltaparam[0] ;
b += deltaparam[1] ;
c += deltaparam[2] ;
mu1 += deltaparam[3] ;
mu2 += deltaparam[4] ;
mu3 += deltaparam[5] ;
sd1 += deltaparam[6] ;
sd2 += deltaparam[7] ;
sd3 += deltaparam[8] ;

for ( p = 0 ; p < nbparam ; p++ ){
  if ( p == 0 ){
    Pmax = P[p] ;
  }
  else if ( P[p] > Pmax ){
    Pmax = P[p] ;
  }
}

restest = resmin / (( nbpoints - nbparam ) * pow( 2.1*erreurmes , 2 )) ;

nbiter += 1 ;
} while ( abs(Pmax) > eps1 && ( abs(deltaparam[0] / a) > eps2 || abs(deltaparam[1] / b)
  > eps2 || abs(deltaparam[2] / c) > eps2 || abs(deltaparam[3] / mu1) > eps2 || abs(

```



```

deltaparam[4] / mu2) > eps2 || abs(deltaparam[5] / mu3) > eps2 || abs(deltaparam[6]
/ sd1) > eps2 || abs(deltaparam[7] / sd2) > eps2 || abs(deltaparam[8] / sd3) >
eps2 ) && /*retest > eps3 &&*/ nbiter < 200 && nbpire < 10 ) ;

if ( nbpire == 10 ){
    nbiter -= 10 ;
}
nbitergr->Fill( nbiter ) ;

a -= deltaparam[0] ; //Pour annuler la derniere variation de parametres ,
b -= deltaparam[1] ; //qui preparait pour la prochaine iteration
c -= deltaparam[2] ;
mu1 -= deltaparam[3] ;
mu2 -= deltaparam[4] ;
mu3 -= deltaparam[5] ;
sd1 -= deltaparam[6] ;
sd2 -= deltaparam[7] ;
sd3 -= deltaparam[8] ;

if ( retest > 0 ){
    nblmused += 1 ;
    //////////// Trouver l'amplitude du signal pour la somme de 3 gaussiennes ////////////

    x = xtemps[maxpos - 2] ; //Exprime en ns
    while ( troisgauss( a, b, c, mu1, mu2, mu3, sd1, sd2, sd3, x ) < troisgauss( a, b, c,
        mu1, mu2, mu3, sd1, sd2, sd3, x + past ) ){
        x += past ;
    }
    ampfit = troisgauss( a, b, c, mu1, mu2, mu3, sd1, sd2, sd3, x ) ;
    ampfitgr->Fill( ampfit ) ;
    ampfitmoy += ampfit ;
    ampfitmoy2 += pow( ampfit, 2 ) ;

    ampcompar->Fill( ampprecis, ampfit ) ;
    ampratiogr->Fill( ampprecis / ampfit ) ;
    ampratiomoy += ampprecis / ampfit ;
    ampratiomoy2 += pow( ampprecis / ampfit, 2 ) ;

    //////////// Trouver la phase relative pour la somme de 3 gaussiennes ////////////

    phasefit = x - xtemps[maxpos] ;
    if ( phasefit < -5 ){
        phasefit += 10 ;
    }
    else if ( phasefit >= 5 ){
        phasefit -= 10 ;
    }
    phasefitgr->Fill( phasefit - 0.5 ) ;
    phasefitgr2->Fill( phasefit - 0.5 ) ;
    phasecompar->Fill( phase - 0.5, phasefit - 0.5 ) ;

    //////////// Correction sur l'alignement des signaux pour la somme de 3 gaussiennes
    ////////////

    pentecorrection = -115.0 / 6.0 ;
    bcorrection = 300 - pentecorrection * 274 ;
    if ( donneesbrute[pointseuil3 - 2] < ( pentecorrection * mu1 + bcorrection ) * 1000 /
        pow( 2, 14 ) ){
        mu1 += 10 ;
        mu2 += 10 ;
        mu3 += 10 ;
    }
}

```

```

}

////////// Prise en note de valeurs importantes , pour la somme de 3 gaussiennes
//////////

a = a / ampfit ;
b = b / ampfit ;
c = c / ampfit ;

restestgr->Fill( restest ) ;
restestmoy += restest ;
restestmoy2 += pow( restest , 2 ) ;
amplitchi2->Fill( restest , ampfit ) ;
alm->Fill( a ) ;
almmoy += a ;
almmoy2 += pow( a , 2 ) ;
blm->Fill( b ) ;
blmmoy += b ;
blmmoy2 += pow( b , 2 ) ;
clm->Fill( c ) ;
clmmoy += c ;
clmmoy2 += pow( c , 2 ) ;
mu1lm->Fill( mu1 ) ;
mu1mmoy += mu1 ;
mu1mmoy2 += pow( mu1 , 2 ) ;
mu2lm->Fill( mu2 ) ;
mu2mmoy += mu2 ;
mu2mmoy2 += pow( mu2 , 2 ) ;
mu3lm->Fill( mu3 ) ;
mu3mmoy += mu3 ;
mu3mmoy2 += pow( mu3 , 2 ) ;
sd1lm->Fill( sd1 ) ;
sd1mmoy += sd1 ;
sd1mmoy2 += pow( sd1 , 2 ) ;
sd2lm->Fill( sd2 ) ;
sd2mmoy += sd2 ;
sd2mmoy2 += pow( sd2 , 2 ) ;
sd3lm->Fill( sd3 ) ;
sd3mmoy += sd3 ;
sd3mmoy2 += pow( sd3 , 2 ) ;
mu21gr->Fill( mu2 - mu1 ) ;
mu32gr->Fill( mu3 - mu2 ) ;
mu1seuil->Fill( mu1 , donneesbrute[pointseuil3 - 2] / ampfit ) ;
mu2seuil->Fill( mu2 , donneesbrute[pointseuil3 - 2] / ampfit ) ;
mu3seuil->Fill( mu3 , donneesbrute[pointseuil3 - 2] / ampfit ) ;
amplita->Fill( a , ampfit ) ;
amplitb->Fill( b , ampfit ) ;
amplitc->Fill( c , ampfit ) ;
amplitmu1->Fill( mu1 , ampfit ) ;
amplitmu2->Fill( mu2 , ampfit ) ;
amplitmu3->Fill( mu3 , ampfit ) ;
amplitsd1->Fill( sd1 , ampfit ) ;
amplitsd2->Fill( sd2 , ampfit ) ;
amplitsd3->Fill( sd3 , ampfit ) ;
mu12somme->Fill( mu1 + mu2 ) ;
mu123somme->Fill( mu1 + mu2 + mu3 ) ;
amplitmu12->Fill( mu1 + mu2 , ampfit ) ;
amplitmu123->Fill( mu1 + mu2 + mu3 , ampfit ) ;

//////////
//          Assignation aux formes de reference - moyennes experimentales          //

```



```

//          Moyennes et incertitude          //
////////////////////////////////////////////////////////////////////////////////////

if ( nbampli != 0 ){
    amplimoy = amplimoy / nbampli ;
    sigmaamp = incertitude( amplimoy, amplimoy2, nbampli ) ;
}
if ( nbrearise != 0 ){
    cfdrisemoy = cfdrisemoy / nbrearise ;
    sigma4 = incertitude( cfdrisemoy, cfdrisemoy2, nbampli ) ;
}
if ( nbrearise != 0 ){
    realrisemoy = realrisemoy / nbrearise ;
    sigma = incertitude( realrisemoy, realrisemoy2, nbrearise ) ;
}
if ( nbrise != 0 ){
    risemoy = risemoy / nbrise ;
    sigma2 = incertitude( risemoy, risemoy2, nbrise ) ;
}
if ( pic1 != 0 ){
    risemoypic1 = risemoypic1 / pic1 ;
    sigma3 = incertitude( risemoypic1, risemoypic12, pic1 ) ;
}
if ( nblmused != 0 ){
    nblm = nblmused ;

    almmoy = almmoy / nblm ;
    blmmoy = blmmoy / nblm ;
    clmmoy = clmmoy / nblm ;
    nullmmoy = nullmmoy / nblm ;
    mu2lmmoy = mu2lmmoy / nblm ;
    mu3lmmoy = mu3lmmoy / nblm ;
    sd1lmmoy = sd1lmmoy / nblm ;
    sd2lmmoy = sd2lmmoy / nblm ;
    sd3lmmoy = sd3lmmoy / nblm ;

    ampfitmoy = ampfitmoy / nblm ;
    ampratiomoy = ampratiomoy / nblm ;
    restestmoy = restestmoy / nblm ;

    sigmaa = incertitude( almmoy, almmoy2, nblm ) ;
    sigmab = incertitude( blmmoy, blmmoy2, nblm ) ;
    sigmac = incertitude( clmmoy, clmmoy2, nblm ) ;
    sigmamu1 = incertitude( nullmmoy, nullmmoy2, nblm ) ;
    sigmamu2 = incertitude( mu2lmmoy, mu2lmmoy2, nblm ) ;
    sigmamu3 = incertitude( mu3lmmoy, mu3lmmoy2, nblm ) ;
    sigmasd1 = incertitude( sd1lmmoy, sd1lmmoy2, nblm ) ;
    sigmasd2 = incertitude( sd2lmmoy, sd2lmmoy2, nblm ) ;
    sigmasd3 = incertitude( sd3lmmoy, sd3lmmoy2, nblm ) ;

    sigmaampfit = incertitude( ampfitmoy, ampfitmoy2, nblm ) ;
    sigmaampratio = incertitude( ampratiomoy, ampratiomoy2, nblm ) ;
    sigmaarestest = incertitude( restestmoy, restestmoy2, nblm ) ;
}

////////////////////////////////////////////////////////////////////////////////////
// Formes de references – somme de 3 gaussiennes //
////////////////////////////////////////////////////////////////////////////////////

//On cherche x tel que l'amplitude est de 550 (~a peu pres l'amplitude du 1er point de la phase 0
//          experimentale moyenne)

```

```

x = xtemps[pointseuil3 + 2] ; //Exprime en ns
while ( troisgauss( almoy, blmmoy, clmmoy, nullmmoy, mu2lmmoy, mu3lmmoy, sd1lmmoy, sd2lmmoy, sd3lmmoy, x ) <
    troisgauss( almoy, blmmoy, clmmoy, nullmmoy, mu2lmmoy, mu3lmmoy, sd1lmmoy, sd2lmmoy, sd3lmmoy, x + past
    ) ){
    x += past ;
}
ampfit = troisgauss( almoy, blmmoy, clmmoy, nullmmoy, mu2lmmoy, mu3lmmoy, sd1lmmoy, sd2lmmoy, sd3lmmoy, x ) ;

//On cherche x tel que l'amplitude est de 550 (~a peu pres l'amplitude du 1er point de la phase 0
    experimentale moyenne)
x = xtemps[pointseuil3] ; //Exprime en ns
while ( almoy * exp( - pow(x - nullmmoy, 2) / (2 * pow( sd1lmmoy, 2 )) ) + blmmoy * exp( - pow(x - mu2lmmoy,
    2) / (2 * pow( sd2lmmoy, 2 )) ) + clmmoy * exp( - pow(x - mu3lmmoy, 2) / (2 * pow( sd3lmmoy, 2 )) ) >
    0.01 ){//seuil ){// < 550 ){
    x -= 0.01 ;
}

pointzero = x + 0.01 ; //1er point de la phase 0 analytique

for ( p = 0 ; p < 10 ; p++ ){
    //sprintf( fname, "shapefit_gauss_phase%d-CsI.txt", p ) ;
    sprintf( fname, "shapefit_gauss_phase%d-generateur.txt", p ) ;
    ofstream shapefitfile ( fname, ios::out | ios::trunc | ios::beg ) ;
    for ( q = 0 ; q < 33 ; q++ ){
        if ( q < debutref ){
            shapefitfile << shapefit[p][q] << endl ;
        }
        else {
            temps = ( pointzero + p ) + ( q - debutref ) * 10 ;
            shapefit[p][q] = ( almoy * exp( - pow(temps - nullmmoy, 2) / (2 * pow( sd1lmmoy, 2 )) ) + blmmoy * exp(
                - pow(temps - mu2lmmoy, 2) / (2 * pow( sd2lmmoy, 2 )) ) + clmmoy * exp( - pow(temps - mu3lmmoy, 2)
                / (2 * pow( sd3lmmoy, 2 )) ) ) / ampfit ;
            shapefitfile << shapefit[p][q] << endl ;
        }
    }
    shapefitfile.close() ;
}
//calculcoeff( "shapefit_gauss_coeff-CsI.txt", shapefit, erreurmes ) ;
calculcoeff( "shapefit_gauss_coeff-generateur.txt", shapefit, erreurmes ) ;

////////////////////////////////////
// Ordonnancement pour les formes moyennes experimentales //
////////////////////////////////////

for ( q = 0 ; q < 33 ; q++ ){
    xrefer[q] = q * 10 ; //Exprime en ns
}

for ( p = 0 ; p < 10 ; p++ ){
    for ( q = 0 ; q < 10 ; q++ ){
        if ( refer[p][debutref] > refer[q][debutref] ){
            ordre[p] += 1 ;
        }
    }
}

ofstream phasefile1 ( "phase0_moy_exp-generateur.txt", ios::out | ios::trunc | ios::beg ) ;
ofstream phasefile2 ( "phase1_moy_exp-generateur.txt", ios::out | ios::trunc | ios::beg ) ;
ofstream phasefile3 ( "phase2_moy_exp-generateur.txt", ios::out | ios::trunc | ios::beg ) ;
ofstream phasefile4 ( "phase3_moy_exp-generateur.txt", ios::out | ios::trunc | ios::beg ) ;
ofstream phasefile5 ( "phase4_moy_exp-generateur.txt", ios::out | ios::trunc | ios::beg ) ;

```

```

ofstream phasefile6 ("phase5_moy_exp_generateur.txt", ios::out | ios::trunc | ios::beg );
ofstream phasefile7 ("phase6_moy_exp_generateur.txt", ios::out | ios::trunc | ios::beg );
ofstream phasefile8 ("phase7_moy_exp_generateur.txt", ios::out | ios::trunc | ios::beg );
ofstream phasefile9 ("phase8_moy_exp_generateur.txt", ios::out | ios::trunc | ios::beg );
ofstream phasefile10 ("phase9_moy_exp_generateur.txt", ios::out | ios::trunc | ios::beg );
for ( p = 0 ; p < 10 ; p++ ){
    cout << "p " << p << " et son ordre -> " << ordre[p] << endl ;
    for ( q = 0 ; q < 33 ; q++){
        if ( ordre[p] == 0 ){
            refer1[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile1 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 1 ){
            refer2[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile2 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 2 ){
            refer3[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile3 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 3 ){
            refer4[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile4 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 4 ){
            refer5[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile5 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 5 ){
            refer6[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile6 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 6 ){
            refer7[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile7 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 7 ){
            refer8[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile8 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 8 ){
            refer9[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile9 << refer[p][q] << endl ;
        }
        else if ( ordre[p] == 9 ){
            refer10[q] = refer[p][q] ;
            refer_tot[ordre[p]][q] = refer[p][q] ;
            phasefile10 << refer[p][q] << endl ;
        }
    }
}
phasefile1.close() ;

```

```
    phasefile2.close() ;
    phasefile3.close() ;
    phasefile4.close() ;
    phasefile5.close() ;
    phasefile6.close() ;
    phasefile7.close() ;
    phasefile8.close() ;
    phasefile9.close() ;
    phasefile10.close() ;

}
else {
    cout << "Le fichier n'ouvre pas" << endl << endl ;
}
}
else {
    cout << "Le fichier du piedestal et/ou du temps de montee ne s'ouvre(nt) pas." << endl ;
}
return 0 ;
}
```

Annexe IX

Code pour l'extraction des paramètres A , B et P avec la méthode du DSP

```
//extraction_ABP.cpp ;
//Ecrit par Simon Lagrange le 17 novembre 2016, modifie le 18 novembre 2016
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <stdlib.h>
#include <stdio.h>

#include "TROOT.h"
#include "TApplication.h"
#include "TCanvas.h"
#include "TPad.h"

#include "TH1D.h"
#include "TH2D.h"
#include "TGraph.h"
// #include "TGraphErrors.h"
#include "TGraphEditor.h"
#include "TMultigraph.h"
#include "TAxis.h"
#include "TGaxis.h"
#include "TMarker.h"
#include "TLegend.h"
#include "TStyle.h"
#include "TColor.h"
#include "THistPainter.h"

using namespace std ;

#define pi 3.14159265358979323846

////////////////////////////////////
//**          ***** Main code *****          **//
////////////////////////////////////

int main(){

    TApplication theApp("App", 0, 0) ;           //Necessaire pour afficher les graphiques et histogrammes

    //////////////////////////////////////
    //          Definitions des variables          //
    //////////////////////////////////////

    int i, j, k, m, q ;                          //Iterateurs
    char fname[100] ;
    string ligne ;                               //Pour prendre les lignes du fichiers , une a la fois
    string mot ;                                 //Un des mots contenus dans une ligne
    string vramot ;                             //Seuls les 4 premiers caracteres d'un mot forme une mesure , un vrai mot
    string quest ;                              //Pour donner le temps de voir les graphiques produits
    double donnees [140] ;                     //Liste des donnees d'un signal particulier
    double donneesbrute[140] ;
    int marqueur = 0 ;                          //Indique qu'on extrait un signal des lignes suivantes dans la lecture
    int index = 0 ;                            //Pour un signal , combien de valeurs sont deja extraites
```



```

int nbsignaux = 0 ; //Combien de signaux sont consideres
double piedestal ; //Valeur du piedestal
double seuil ; //Valeur du seuil
int nbligne = 0 ;
int pointseuil ;
double pentetest ;
int discarte = 0 ;
string discartequi ;
int penteneg = 0 ;
int pentepos = 0 ;
int seuildepasse = 0 ;
int interference = 0 ;
int sousseuil = 0 ;
int trop_bas ;
double erreurmes ;
double sigma ;

double Cref[10][3][32] ;
double Aextr[109] ;
double Bextr[109] ;
double Pextr[109] ;
double t0extr[109] ;
double Aextr2[109] ;
double Bextr2[109] ;
double Pextr2[109] ;
double t0extr2[109] ;
int Amaxpos ;
double Amax ;
int Amaxpos2 ;
double Amax2 ;
double t0 ;
double t02 ;
int phaseeval2, phaseevalfin ;
double xdonneesextr[109] ;
double xtempsextr[109] ;

double referexp[10][32] = {} ;
double shapefitgauss[10][32] = {} ;
double shapefitactif[10][32] = {} ;

double chi2, chi2min, chi22, chi22min ;
int chi2minpos, chi22minpos ;

for ( j = 0 ; j < 109 ; j++ ){
    xdonneesextr[j] = j + 1 ;
    xtempsextr[j] = xdonneesextr[j] * 10 ;
}

//ifstream coeffin ( "shapefit_gauss_coeff_generateur.txt", ios::in | ios::binary | ios::beg ) ; //Stocke les
//valeurs pour le calcul du piedestal et de son sigma
//ifstream coeffin ( "shapefit_gauss_coeff_CsI.txt", ios::in | ios::binary | ios::beg ) ; //Stocke les valeurs
//pour le calcul du piedestal et de son sigma
//ifstream coeffin ( "shapefit_filtreactif1_coeff_generateur.txt", ios::in | ios::binary | ios::beg ) ; //Stocke
//les valeurs pour le calcul du piedestal et de son sigma
ifstream coeffin ( "shapefit_filtreactif1_coeff_CsI.txt", ios::in | ios::binary | ios::beg ) ; //Stocke les
//valeurs pour le calcul du piedestal et de son sigma
//ifstream coeffin ( "coefficients_moy_exp_generateur.txt", ios::in | ios::binary | ios::beg ) ; //Stocke les
//valeurs pour le calcul du piedestal et de son sigma
//ifstream coeffin ( "coefficients_moy_exp_CsI.txt", ios::in | ios::binary | ios::beg ) ; //Stocke les valeurs
//pour le calcul du piedestal et de son sigma
j = 0 ;

```

```

while ( getline( coeffin, ligne ) ){
    k = 0 ;
    q = j / 5 ;
    if ( j % 5 == 1 ){
        for ( i = 0 ; i < ligne.length() ; i++ ){
            if ( ligne[i] != ',' ){
                mot += ligne[i] ;
            }
            else {
                Cref[q][0][k] = stod( mot ) ; //coeff pour A
                mot = "" ;
                k += 1 ;
            }
        }
    }
    else if ( j % 5 == 2 ){
        for ( i = 0 ; i < ligne.length() ; i++ ){
            if ( ligne[i] != ',' ){
                mot += ligne[i] ;
            }
            else {
                Cref[q][1][k] = 10 * stod( mot ) ; //coeff pour B
                mot = "" ;
                k += 1 ;
            }
        }
    }
    else if ( j % 5 == 3 ){
        for ( i = 0 ; i < ligne.length() ; i++ ){
            if ( ligne[i] != ',' ){
                mot += ligne[i] ;
            }
            else {
                Cref[q][2][k] = stod( mot ) ; //coeff pour P
                mot = "" ;
                k += 1 ;
            }
        }
    }
    j += 1 ;
}

for ( i = 0 ; i < 10 ; i++ ){
    // sprintf( fname, "phase%d_moy_exp_generateur.txt", i ) ;
    /* sprintf( fname, "phase%d_moy_exp_CsI.txt", i ) ;
    ifstream shapemoy ( fname ) ;
    if ( shapemoy.is_open() ){
        j = 0 ;
        while ( getline( shapemoy, ligne ) ){
            referexp[i][j] = stod( ligne ) ;
            j += 1 ;
        }
        shapemoy.close() ;
    }
    else {
        cout << "phase" << i << "_moy_exp n'ouvre pas" << endl ;
    }*/

    // sprintf( fname, "shapefit_gauss_phase%d_generateur.txt", i ) ;
    /* sprintf( fname, "shapefit_gauss_phase%d_CsI.txt", i ) ;
    ifstream shapefitfile ( fname ) ;

```

```

if ( shapefitfile.is_open() ){
    j = 0 ;
    while ( getline( shapefitfile , ligne ) ){
        shapefitgauss[i][j] = stod( ligne ) ;
        j += 1 ;
    }
    shapefitfile.close() ;
}
else {
    cout << fname << " n'ouvre pas" << endl ;
}*/

//sprintf( fname, "shapefit_filtreactif1_phase%d_generateur.txt", i ) ;
sprintf( fname, "shapefit_filtreactif1_phase%d_CsI.txt", i ) ;
ifstream shapefitfile2 ( fname ) ;
if ( shapefitfile2.is_open() ){
    j = 0 ;
    while ( getline( shapefitfile2 , ligne ) ){
        shapefitactif[i][j] = stod( ligne ) ;
        j += 1 ;
    }
    shapefitfile2.close() ;
}
else {
    cout << fname << " n'ouvre pas" << endl ;
}
}

//////////////////////////////////////////////////////////////////
//          Lecture du fichier          //
//////////////////////////////////////////////////////////////////

ifstream pedtemp ( "pedestal_CsI.txt", ios::in | ios::binary | ios::beg ) ; //Stocke les valeurs pour le calcul
    du pedestal et de son sigma
//ifstream pedtemp ( "pedestal_generateur.txt", ios::in | ios::binary | ios::beg ) ; //Stocke les valeurs pour
    le calcul du pedestal et de son sigma
if ( pedtemp.is_open() ){
    getline( pedtemp, ligne ) ;
    pedestal = stod( ligne ) ;
    getline( pedtemp, ligne ) ;
    sigma = stod( ligne ) ;
    seuil = 5 * sigma * 1000 / pow( 2, 13 ) ;
    erreurmes = sigma * 1000 / pow( 2, 13 ) ;
    pedtemp.close() ;
    cout << "Pedestal : " << pedestal * 1000 / pow( 2, 13 ) << " mV " << endl ;
    cout << "Seuil : " << seuil << " mV " << endl << endl ;

    ifstream adcfile ( "data\\CsI_bon_output_trie.log" ) ; //On ouvre le fichier
    //ifstream adcfile ( "data\\generateur_impulsions\\pulser_phases_non_triees\\pulser_signals_0_10.log" ) ; //On
        ouvre le fichier

    THID *t0gr = new THID("t0gr", /*""*/ ; t0 (ns); Decompte", 200, -20.5, 10.5 ) ;
    THID *phaseeval2gr = new THID("phaseeval2gr", /*""*/ ; phaseeval2 (ns); Decompte", 31, -10.5, 20.5 ) ;
    THID *t02gr = new THID("t02gr", /*""*/ ; t02 (ns); Decompte", 200, -2, 2 ) ;
    THID *phaseevalfingr = new THID("phaseevalfingr", /*""*/ ; phaseevalfin (ns); Decompte", 5, -2.5, 2.5 ) ;
    THID *ratioamp = new THID("ratioamp", /*""*/ ; A_1/A_2 (sans unite); Decompte", 100, 0.95, 1.05 ) ;
    THID *ampliteval = new THID("ampliteval", /*""*/ ; Amplitude (mV); Decompte", 200, 300, 900 ) ;//200, 410, 440
        ) ;
    THID *pedestalgr = new THID("pedestalgr", /*""*/ ; Pedestal (mV); Decompte", 200, 950, 1050 ) ;//200, 450,
        550 ) ;

```

```

TH1D *pedestalgr2 = new TH1D("pedestalgr2", /*""*/ ; Piedestal (mV); Decompte", 200, 950, 1050 ) ;//200, 450,
550 ) ;
TH1D *chi2gr = new TH1D("chi2gr", /*""*/ ; chi^2 (sans unite); Decompte", 200, 0, 150 ) ;
TH1D *chi2gr2 = new TH1D("chi2gr2", /*""*/ ; chi^2 (sans unite); Decompte", 400, 0, 150 ) ;

if ( adcfile.is_open() ){
while ( getline( adcfile , ligne ) ){ //Tant qu'on n'atteint pas la fin du fichier
nbligne += 1 ;
if ( nbligne % (100*26) == 1 ){
cout << nbligne << " — " << nbligne / 26 << endl ;
}

if ( marqueur == 0 && ligne.length() == 54 && ligne[0] == '8' && ligne[9] == '9' && ligne[18] == 'a' &&
ligne[27] == 'b' && ligne[36] == 'b' && ligne[45] == 'c' ){
marqueur = 1 ; //On se prepare a lire les donnees pour un signal
}
else if ( marqueur == 1 ){
for ( i = 0 ; i < ligne.length() ; i++){
if ( ligne[i] != ' ' ){
mot += ligne[i] ;
}
else if ( ligne[i] == ' ' ){
if ( ( mot[0] != 'e' ) && ( mot[0] != 'c' ) ){
for ( j = 0 ; j < 4 ; j++){ //Seuls les 4 premiers caracteres sur 8 nous interessent
vraimot += mot[j] ;
}
donnees[index] = ( stoi( vraimot , nullptr , 16 ) ) * 1000 / pow( 2, 13 ) ;
donneesbrute[index] = donnees[index] ;
index += 1 ;
}
}
else if ( mot[0] == 'e' && mot[1] == '0' ){ //On atteint la fin du signal , on en commence l'analyse
marqueur = 0 ;
index = 0 ;

nbsignaux += 1 ;

////////////////////////////////////
// Scan du signal //
////////////////////////////////////

pointseuil = -1 ; //Devient positif lorsqu'on trouve un signal
for ( j = 0 ; j < 130; j++){ //On arrete a 130, car inutile de chercher des signaux de 10 points
if ( donnees[j] >= seuil + piedestal * 1000 / pow( 2, 13 ) && donnees[j+10] >= seuil + piedestal *
1000 / pow( 2, 13 ) && j > 0 ){ //On depasse le seuil , Csl
//if ( donnees[j] >= seuil + piedestal * 1000 / pow( 2, 14 ) && donnees[j+10] >= seuil + piedestal
* 1000 / pow( 2, 14 ) && j > 0 ){ //On depasse le seuil , generateur
seuildepasse += 1 ;
pointseuil = j ;

pentetest = donnees[pointseuil + 6] - donnees[pointseuil + 5] ;
if ( pentetest <= 0 ){
penteneg += 1 ;
}

if ( pentetest > 0 ){
pentepos += 1 ;

k = 0 ;
trop_bas = 0 ;
while ( donnees[k] > ( 7500 ) * 1000 / pow( 2, 13 ) && k < 15 ){

```

```

    k += 1 ;
}
if ( k < 15 ){
    trop_bas = 1 ; //Pour couper les interferences avec le CsI
}

while ( donnees[j] <= donnees[j + 1] && j < 140 ){ //On cherche le max
    j += 1 ;
}

if ( donnees[j] > 11000 * 1000 / pow( 2, 13 ) && trop_bas == 0 ){ //Seuils d'amplitude , CsI
//if ( donnees[j] > 5000 * 1000 / pow( 2, 14 ) ){ //Seuil d'amplitude , generateur
    for ( q = 0 ; q < 109 ; q++){
        Aextr[q] = 0 ;
        Bextr[q] = 0 ;
        Pextr[q] = 0 ;
        for ( m = 0 ; m < 32 ; m++){
            Aextr[q] += Cref[0][0][m] * donnees[m + q] ;
            Bextr[q] += Cref[0][1][m] * donnees[m + q] ;
            Pextr[q] += Cref[0][2][m] * donnees[m + q] ;
        }
        if ( q == 0 ){
            Amax = Aextr[q] ;
            Amaxpos = q ;
        }
        else if ( Aextr[q] > Amax ){
            Amax = Aextr[q] ;
            Amaxpos = q ;
        }
        t0extr[q] = - Bextr[q] / Aextr[q] ;
    }

    for ( m = 0 ; m < 109 ; m++){
        chi2 = 0 ;
        for ( q = 0 ; q < 32 ; q++){
            //chi2 += pow( ( donnees[m + q] - piedestal * 1000 / pow( 2, 14 ) - Amax * referexp[0][q]
            // ) / ( erreurmes ), 2 ) / 32 ;
            //chi2 += pow( ( donnees[m + q] - piedestal * 1000 / pow( 2, 14 ) - Amax * shapefitgauss
            // [0][q] ) / ( erreurmes ), 2 ) / 32 ;
            //chi2 += pow( ( donnees[m + q] - piedestal * 1000 / pow( 2, 14 ) - Amax * shapefitactif
            // [0][q] ) / ( erreurmes ), 2 ) / 32 ;
            chi2 += pow( ( donnees[m + q] - piedestal * 1000 / pow( 2, 13 ) - Amax * referexp[0][q]
            // ) / ( erreurmes ), 2 ) / 32 ;
            //chi2 += pow( ( donnees[m + q] - piedestal * 1000 / pow( 2, 13 ) - Amax * shapefitgauss
            // [0][q] ) / ( erreurmes ), 2 ) / 32 ;
            //chi2 += pow( ( donnees[m + q] - piedestal * 1000 / pow( 2, 13 ) - Amax * shapefitactif
            // [0][q] ) / ( erreurmes ), 2 ) / 32 ;
        }
        if ( m == 0 ){
            chi2min = chi2 ;
            chi2minpos = q ;
        }
        else if ( chi2 < chi2min ){
            chi2min = chi2 ;
            chi2minpos = q ;
        }
    }
}
chi2gr->Fill( chi2min ) ;

t0 = - Bextr[Amaxpos] / Aextr[Amaxpos] ;
phaseval2 = floor( abs( t0 ) + 0.5 ) ;

```

```

if ( phaseeval2 >= 10 ){
    phaseeval2 -= 10 ;
}
if ( t0 < -10 ){
    t0 += 10 ;
}
t0gr->Fill( t0 ) ;
phaseeval2gr->Fill( phaseeval2 ) ;
pedestalgr->Fill( Pextr[Amxpos] ) ;
for ( q = 0 ; q < 109 ; q++){
    Aextr2[q] = 0 ;
    Bextr2[q] = 0 ;
    Pextr2[q] = 0 ;
    for ( m = 0 ; m < 32 ; m++){
        Aextr2[q] += Cref[phaseeval2][0][m] * donnees[m + q] ;
        Bextr2[q] += Cref[phaseeval2][1][m] * donnees[m + q] ;
        Pextr2[q] += Cref[phaseeval2][2][m] * donnees[m + q] ;
    }
    if ( q == 0 ){
        Amax2 = Aextr2[q] ;
        Amaxpos2 = q ;
    }
    else if ( Aextr2[q] > Amax2 ){
        Amax2 = Aextr2[q] ;
        Amaxpos2 = q ;
    }
    t0extr2[q] = - Bextr2[q] / Aextr2[q] ;
}

for ( m = 0 ; m < 109 ; m++){
    chi22 = 0 ;
    for ( q = 0 ; q < 32 ; q++){
        //chi22 += pow( ( donnees[m + q] - pedestal * 1000 / pow( 2, 14 ) - Amax2 * referexp[
            phaseeval2][q] ) / ( erreurmes ), 2 ) / 32 ;
        //chi22 += pow( ( donnees[m + q] - pedestal * 1000 / pow( 2, 14 ) - Amax2 *
            shapefitgauss[phaseeval2][q] ) / ( erreurmes ), 2 ) / 32 ;
        //chi22 += pow( ( donnees[m + q] - pedestal * 1000 / pow( 2, 14 ) - Amax2 *
            shapefitactif[phaseeval2][q] ) / ( erreurmes ), 2 ) / 32 ;
        chi22 += pow( ( donnees[m + q] - pedestal * 1000 / pow( 2, 13 ) - Amax2 * referexp[
            phaseeval2][q] ) / ( erreurmes ), 2 ) / 32 ;
        //chi22 += pow( ( donnees[m + q] - pedestal * 1000 / pow( 2, 13 ) - Amax2 *
            shapefitgauss[phaseeval2][q] ) / ( erreurmes ), 2 ) / 32 ;
        //chi22 += pow( ( donnees[m + q] - pedestal * 1000 / pow( 2, 13 ) - Amax2 *
            shapefitactif[phaseeval2][q] ) / ( erreurmes ), 2 ) / 32 ;
    }
    if ( m == 0 ){
        chi22min = chi22 ;
        chi22minpos = q ;
    }
    else if ( chi22 < chi22min ){
        chi22min = chi22 ;
        chi22minpos = q ;
    }
}
chi2gr2->Fill( chi22min ) ;

t02 = - Bextr2[Amxpos] / Aextr2[Amxpos] ;
phaseevalfin = floor( abs( t02 + 0.5 ) ) ;
t02gr->Fill( t02 ) ;
phaseevalfingr->Fill( phaseevalfin ) ;

```

```

        piedestalgr2 ->Fill( Pextr2[Amaxpos2] ) ;
        ampliteval ->Fill( Amax2 ) ;

        ratioamp ->Fill( Amax / Amax2 ) ;
    }
    else {
        discarte += 1 ;
        discartequi += to_string(nbsignaux) + ", " ;
    }
}
j = 1000 ; //Pour sauter au signal suivant

}
}

if ( pointseuil < 0 ){
    interference += 1 ;
    k = 0 ;
    while ( k < 140 && donnees[k] < seuil + piedestal * 1000 / pow( 2, 13 ) ){ //CsI
        //while ( k < 140 && donnees[k] < seuil + piedestal * 1000 / pow( 2, 14 ) ){ //generateur
        k += 1 ;
    }
    if ( k >= 140 ){
        sousseuil += 1 ;
    }
}
}
vraimot = "" ;
mot = "" ;
}
}

}
ligne = "" ;

}
adcfiler.close() ;

//////////////////////////////////////
//                               Graphiques                               //
//////////////////////////////////////

TCanvas * graphe = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
gStyle->SetOptStat(0) ;
graphe->SetFillColor(10) ;

t0gr ->Draw("HIST") ;

graphe->Modified() ;
graphe->Update() ;

graphe->SaveAs( "graphiques\\t0_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
//cin >> quest ;

delete graphe ;
delete t0gr ;

TCanvas * graphe3 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;

```

```

graphe3->SetFillColor(10) ;

phaseeval2gr->Draw("HIST") ;

graphe3->Modified() ;
graphe3->Update() ;

graphe3->SaveAs( "graphiques\\phaseeval2gr_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
// cin >> quest ;

delete graphe3 ;
delete phaseeval2gr ;

TCanvas * graphe4 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe4->SetFillColor(10) ;

t02gr->Draw("HIST") ;

graphe4->Modified() ;
graphe4->Update() ;

graphe4->SaveAs( "graphiques\\t02_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
// cin >> quest ;

delete graphe4 ;
delete t02gr ;

TCanvas * graphe5 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe5->SetFillColor(10) ;

phaseevalfingr->Draw("HIST") ;

graphe5->Modified() ;
graphe5->Update() ;

graphe5->SaveAs( "graphiques\\phaseevalfingr_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
// cin >> quest ;

delete graphe5 ;
delete phaseevalfingr ;

TCanvas * graphe6 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe6->SetFillColor(10) ;

ratioamp->Draw("HIST") ;

graphe6->Modified() ;
graphe6->Update() ;

graphe6->SaveAs( "graphiques\\ratioamp_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
// cin >> quest ;

delete graphe6 ;

```



```

delete ratioamp ;

TCanvas * graphe7 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe7->SetFillColor(10) ;

ampliteval->Draw("HIST") ;

graphe7->Modified() ;
graphe7->Update() ;

graphe7->SaveAs( "graphiques\\ampliteval_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
//cin >> quest ;

delete graphe7 ;
delete ampliteval ;

TCanvas * graphe8 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe8->SetFillColor(10) ;

pedestalgr->Draw("HIST") ;

graphe8->Modified() ;
graphe8->Update() ;

graphe8->SaveAs( "graphiques\\pedestal1_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
//cin >> quest ;

delete graphe8 ;
delete pedestalgr ;

TCanvas * graphe9 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe9->SetFillColor(10) ;

pedestalgr2->Draw("HIST") ;

graphe9->Modified() ;
graphe9->Update() ;

graphe9->SaveAs( "graphiques\\pedestal2_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
//cin >> quest ;

delete graphe9 ;
delete pedestalgr2 ;

TCanvas * graphe10 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe10->SetFillColor(10) ;

chi2gr->Draw("HIST") ;

graphe10->Modified() ;
graphe10->Update() ;

graphe10->SaveAs( "graphiques\\chi2_eval1_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique

```

```
//cin >> quest ;

delete graphe10 ;
delete chi2gr ;

TCanvas * graphe11 = new TCanvas(" ", " ", 100, 10, 1200, 640) ;
graphe11->SetFillColor(10) ;

chi2gr2->Draw("HIST") ;

graphe11->Modified() ;
graphe11->Update() ;

graphe11->SaveAs( "graphiques\\chi2_eval2_distribution.png" ) ;

cout << "Taper quelque chose" ; //Qu'on ait le temps de voir le graphique
//cin >> quest ;

delete graphe11 ;
delete chi2gr2 ;
}
else {
    cout << "Le fichier n'ouvre pas" << endl << endl ;
}
}
else {
    cout << "Le fichier du piedestal et/ou du temps de montee ne s'ouvre(nt) pas." << endl ;
}
}

return 0 ;
}
```