

SYLVAIN COMTOIS

**CONCEPTION D'UN MICROSTIMULATEUR OPTIQUE EN VUE D'UNE
DISSECTION DE L'INFORMATION VISUELLE ANIMALE**

Thèse
présentée
à la Faculté des études supérieures
de l'Université Laval
pour l'obtention
du grade de Philosophiae Doctor (Ph.D.)

Département de génie électrique et génie informatique
FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

JUILLET 1999

RÉSUMÉ COURT

La présente thèse décrit principalement la réalisation d'un stimulateur lumineux permettant d'effectuer une dissection visuelle fine du traitement de l'information d'un œil à facettes. Nous entendons par dissection fine la possibilité de stimuler optiquement une ou plusieurs cellules photoréceptrices tout en enregistrant l'activité électrique d'une quelconque cellule visuelle de l'animal. Le stimulateur est automatisé dans le but de simplifier le travail de l'expérimentateur. L'automatisation porte principalement sur le contrôle des paramètres des différents faisceaux lumineux. Il est possible de les déplacer, de contrôler leur diamètre, de modifier leur intensité, de sélectionner leur longueur d'onde, de fixer leur durée et de les synchroniser entre eux à l'aide d'un logiciel qui est simple d'utilisation. Le stimulateur est aussi muni d'un système vidéo permettant de choisir les cellules à stimuler. En dernier lieu, cette thèse présente une technique permettant de mesurer les paramètres d'un modèle représentant les interconnexions de six photorécepteurs d'une même ommatidie.

Sylvain Comtois

Simon Gagné

RÉSUMÉ LONG

La présente thèse décrit la réalisation et l'automatisation d'un stimulateur lumineux qui permet d'étudier les interconnexions neuronales d'un animal possédant des yeux à facettes et une technique permettant d'identifier les paramètres d'un modèle représentant les interconnexions entre six photorécepteurs dont les axones font connexion dans une même cartouche.

Afin de créer un stimulateur polyvalent et modulaire, il a été subdivisé en cinq parties, et chacune d'elles contient des modules mécaniques et électroniques ainsi qu'un logiciel.

La première partie permet de contrôler des moteurs électriques qui servent à changer les divers aspects des faisceaux lumineux ainsi que l'emplacement et l'orientation de l'animal.

La deuxième partie est responsable de l'acquisition de l'activité des senseurs et de l'activité électrique des cellules. Cette partie est principalement composée d'une carte d'acquisition et d'un logiciel. Le logiciel est spécialement conçu pour sauvegarder ou afficher automatiquement les signaux électriques. Il est aussi muni de quelques fonctions nécessaires à l'analyse ou au traitement des signaux électriques.

La troisième partie est similaire à la deuxième partie, à l'exception qu'elle utilise les données d'une caméra et que le logiciel manipule des images au lieu de signaux électriques.

La quatrième partie est responsable de la synchronisation et de la durée des différents événements du stimulateur. Un événement comprend une acquisition, l'injection d'un courant dans une cellule ou le contrôle de l'état d'un faisceau lumineux.

La dernière partie ne comprend qu'un logiciel qui communique avec les autres parties afin de représenter plus fidèlement les fonctionnalités du stimulateur. C'est aussi à partir de ce logiciel qu'il est possible de créer un protocole expérimental qui commandera les autres parties sans l'intervention de l'expérimentateur.

Il est à signaler que ce stimulateur permet la réalisation d'expériences qui n'avaient jusqu'à maintenant pas été envisageables d'effectuer. À titre d'exemple, nous pouvons obtenir les données nécessaires à l'évaluation des paramètres des interconnexions des six photorécepteurs dont les axones font connexion dans une même cartouche pour démontrer la validité du modèle proposé.

AVANT-PROPOS

Je voudrais profiter de l'occasion pour souligner l'apport important des gens de mon entourage à la réalisation de cette entreprise. En particulier, je remercie mon directeur de recherche, le docteur Simon Gagné, qui a su me prodiguer les conseils nécessaires et guider mes travaux. Merci au docteur Philippe ViaRouge pour ses conseils fort utiles en relation avec les circuits de puissance. Merci aux divers techniciens qui m'ont beaucoup aidé dans la réalisation matérielle du montage tels Jacques Lévesque (électronique), Yvon Chalifour (mécanique) et Gilles Brisson (mécanique). Merci à Christian Gourdeau pour les résultats expérimentaux qu'il m'a fournis. Merci aussi à Nathalie Bacon, qui m'a rendu un grand service en vérifiant la qualité du français de la présente thèse.

Pour terminer, je voudrais sincèrement remercier mes parents Jacques Comtois et Jacqueline Leclerc, qui m'ont supporté tout au long de mes études et sans qui je n'aurais pu atteindre un tel niveau de scolarité.

TABLE DES MATIÈRES

RÉSUMÉ COURT	ii
RÉSUMÉ LONG	iii
AVANT-PROPOS	v
TABLE DES MATIÈRES	vi
LISTE DES FIGURES	xiii
LISTE DES TABLEAUX	xx
LISTE DES SYMBOLES	xxi
 CHAPITRE I INTRODUCTION GÉNÉRALE	
 CHAPITRE II PHYSIOLOGIE DE LA MOUCHE DOMESTIQUE	
2.1 Introduction.....	3
2.2 L'organisation de l'œil de la mouche domestique.....	5
2.2.1 Rétine	6
2.2.2 Lamina.....	9
2.3 La superposition neuronale.....	10
2.4 Les pseudopupilles.....	12
2.4.1 La pseudopupille cornéenne.....	12
2.4.2 La pseudopupille cornéenne réduite.....	13
2.4.3 La pseudopupille profonde.....	13

2.5	Conclusion	14
-----	------------------	----

CHAPITRE III LE MONTAGE (PARTIE MATÉRIELLE)

3.1	Introduction.....	15
3.2	Technique de stimulation des photorécepteurs	18
3.3	Stimulateur optique.....	21
3.3.1	Les sources lumineuses	22
3.3.2	Les trajets lumineux	24
3.3.3	La détection de l'intensité lumineuse.....	25
3.3.4	La base	26
3.3.5	Le microscope	27
3.3.6	Discussion	28
3.4	Ordinateur	28
3.5	Contrôleur de moteurs	29
3.5.1	Le circuit	30
3.5.1.1	La carte maîtresse.....	30
3.5.1.1.1	Le microcontrôleur.....	31
3.5.1.1.2	Les mémoires	31
3.5.1.1.3	Les périphériques de communication.....	32
3.5.1.1.4	Le bus de communication	32
3.5.1.1.5	Discussion	33
3.5.1.2	La carte moteur	33
3.5.1.2.1	L'interface au bus.....	35
3.5.1.2.2	L'horloge.....	35
3.5.1.2.3	Le HCTL1100	36
3.5.1.2.4	Le convertisseur numérique analogique.....	36
3.5.1.2.5	L'amplificateur de puissance	36
3.5.1.2.6	L'index	37
3.5.1.2.7	Discussion	37
3.5.2	Discussion	37
3.6	Générateur de séquences.....	38
3.6.1	Le circuit du générateur de séquences	41

3.6.1.1	La carte maîtresse.....	41
3.6.1.2	La carte séquenceur.....	41
3.6.1.2.1	L'interface au bus de communication	43
3.6.1.2.2	Les horloges	43
3.6.1.2.3	Les registres de commande	44
3.6.1.2.4	La mémoire permanente.....	45
3.6.1.2.5	La mémoire temporaire	46
3.6.1.2.6	Le registre permanent et le registre temporaire.....	47
3.6.1.2.7	Le compteur.....	47
3.6.1.2.8	La machine à état primaire	48
3.6.1.2.9	La machine à état secondaire	51
3.6.1.2.10	Le système de démarrage	52
3.6.1.2.11	Le système de gâchette.....	53
3.6.1.2.12	L'état logique des lignes	53
3.6.2	Discussion	53
3.7	Obturateur	54
3.7.1	Le design mécanique.....	55
3.7.2	Le design électrique	56
3.7.2.1	Le circuit de contrôle	57
3.7.2.2	Le circuit de puissance	58
3.7.3	Discussion	59
3.8	Caméra.....	61
3.9	Acquisition et oscilloscope	62
3.10	Amplificateur.....	62
3.11	Compensation des microélectrodes	63
3.12	Synchronisation	64
3.12.1	Synchronisation avec l'ajout d'un délai.....	64
3.12.2	Synchronisation avec un photodétecteur.....	65
3.12.3	Multiplexage des signaux.....	67
3.13	Radiomètre.....	68
3.14	Lecture et contrôle de la température	69

3.15 Conclusion	70
-----------------------	----

CHAPITRE IV LE MONTAGE (PARTIE LOGICIELLE)

4.1 Introduction.....	72
4.2 L'application <i>Contrôleur de moteurs (68HC11)</i>	76
4.2.1 Les configurations de départ.....	76
4.2.2 Les queues de communication	78
4.2.3 La boucle principale.....	79
4.2.4 Les commandes.....	81
4.2.5 Discussion	82
4.3 L'application <i>Serveur de port RS-232</i>	83
4.4 L'application <i>Contrôleur de moteurs (x86)</i>	85
4.4.1 Le fonctionnement général de l'application.....	86
4.4.2 Les différentes façons de déplacer un moteur.....	88
4.4.3 Discussion	90
4.5 L'application <i>Générateur de séquences (68HC11)</i>	91
4.5.1 Les configurations de départ.....	91
4.5.2 Les queues de communication	92
4.5.3 La boucle principale.....	92
4.5.4 Les commandes.....	92
4.5.5 Discussion	94
4.6 L'application <i>Générateur de séquences (x86)</i>	94
4.6.1 Le fonctionnement général de l'application.....	94
4.6.2 Le contrôle des séquenceurs.....	96
4.6.3 La configuration des séquenceurs	97
4.6.4 Discussion	97
4.7 Les « librairies » dynamiques et les pilotes de la carte d'acquisition.....	97
4.7.1 La « librairie » dynamique	98
4.7.1.1 La configuration de la carte d'acquisition.....	99
4.7.1.2 Les opérations de la carte d'acquisition	100
4.7.2 Le pilote de bas niveau.....	100
4.7.3 Discussion	101

4.8	L'application <i>Acquisition et traitement des courbes</i>	102
4.8.1	Les projets	103
4.8.2	L'acquisition	106
4.8.3	L'interface usager.....	109
4.8.4	Les traitements	111
4.8.5	La communication DDE	112
4.8.6	Discussion	112
4.9	Les « librairies » dynamiques et les pilotes de la carte vidéo.....	113
4.10	L'application <i>Acquisition et traitement des images</i>	113
4.10.1	Le projet	115
4.10.2	L'acquisition	116
4.10.3	L'interface usager.....	116
4.10.4	Les traitements	118
4.10.5	La communication DDE	119
4.10.6	Discussion	120
4.11	L'application <i>Contrôle des expériences</i>	122
4.11.1	L'interface usager.....	123
4.11.2	La communication DDE	123
4.11.3	L'image	125
4.11.4	Le contrôle de la base.....	127
4.11.5	Le contrôle de la microélectrode.....	131
4.11.6	Le contrôle du trajet de fond et du polariseur	132
4.11.7	Le contrôle des trajets lumineux	133
4.11.8	Le protocole expérimental	134
4.11.9	Les algorithmes spécialisés.....	136
4.11.10	L'acquisition de courbes	139
4.11.11	Discussion	139
4.12	Conclusion	140

CHAPITRE V ÉTAPES PRÉLIMINAIRES D'UNE EXPÉRIENCE

5.1	Introduction.....	142
5.2	Installation de la préparation sur le stimulateur optique.....	142

5.3	Installation des microélectrodes sur le stimulateur optique.....	145
5.4	Recherche d'une cellule.....	146
5.5	Alignement de la cellule sous enregistrement	149
5.5.1	Alignement grossier de la cellule sous enregistrement	150
5.5.2	Alignement fin de la cellule sous enregistrement	151
5.6	Identification des facettes de la pseudopupille cornéenne réduite.....	152
5.6.1	Identification des facettes avec la réponse d'un photorécepteur.....	152
5.6.2	Identification des facettes avec la réponse d'une LMC	154
5.6.3	Identification des facettes avec la réponse d'une cartouche	154
5.7	Conclusion	155

**CHAPITRE VI ESTIMATION DES PARAMÈTRES D'UN MODÈLE
REPRÉSENTANT L'INTERCONNEXION ENTRE SIX
PHOTORÉCEPTEURS PARTICIPANT À LA MÊME
SUPERPOSITION NEURONALE**

6.1	Introduction.....	156
6.2	Le modèle électrique.....	157
6.2.1	Les paramètres R_L , δE et R_N	158
6.2.2	Le paramètre R_A	159
6.2.3	Le paramètre R_T	160
6.2.4	Le paramètre R_B	160
6.2.5	Le paramètre R_S	161
6.3	Extraction des équations selon différentes stimulations.....	161
6.3.1	Stimulation lumineuse d'un seul photorécepteur avec ou sans injection de courant dans ce même photorécepteur	162
6.3.2	Stimulation lumineuse de deux photorécepteurs.....	164
6.3.3	Stimulation lumineuse de trois photorécepteurs	165
6.3.4	Stimulation lumineuse des six photorécepteurs avec ou sans injection de courant dans un seul photorécepteur	166
6.3.5	Injection de courant dans un seul photorécepteur sans stimulation lumineuse	168

6.3.6	Discussion sur les équations obtenues	169
6.4	Résolution du modèle	171
6.4.1	Génération des équations	172
6.4.2	Résolution du système d'équations.....	173
6.5	Estimation des paramètres à partir des résultats expérimentaux de van Hateren ...	174
6.6	Comparaison entre une première série de résultats et ceux tirés de l'article de van Hateren.....	176
6.7	Conclusion	182

CHAPITRE VII CONCLUSION GÉNÉRALE

RÉFÉRENCES	187
-------------------------	------------

ANNEXE A	LA COMMUNICATION DDE SOUS WINDOWS NT	190
-----------------	---	------------

ANNEXE B	PROTOCOLE EXPÉRIMENTAL UTILISÉ POUR LA RÉSOLUTION DU MODÈLE DU CHAPITRE VI	193
-----------------	---	------------

ANNEXE C	RÉSOLUTION, ÉQUATIONS ET GRAPHS DE FLUENCE DU MODÈLE	202
-----------------	---	------------

C.1	Résolution des équations.....	202
C.2	Graphes de fluences	208
C.3	Équations du modèle	212
C.4	Équations des paramètres	220

LISTE DES FIGURES

Figure 2-1 :	Photographie d'une mouche domestique femelle (tirée de Hardie, 1985).	3
Figure 2-2 :	Schéma représentant différentes couches de l'œil de la mouche domestique (tirée de Strausfeld, 1989).	5
Figure 2-3 :	Vue en coupe de la rétine de l'œil de la mouche domestique (tirée de Rossel, 1989).	6
Figure 2-4 :	Vue en coupe de deux ommatidies (tirée de Hardi, 1985). a) Vue longitudinale des deux ommatidies. b) Trois vues transversales prises le long de la vue longitudinale. L'emplacement des vues transversales dans la vue longitudinale sont indiquées par les flèches. Les différentes parties des ommatidies sont : facette (f), pseudocône (pc), cellules pigmentaires primaires (cpp), cellules de Semper (cs), cellules pigmentaires secondaires (cps), segment interne (si), rhabdomère ou segment externe (r) et l'identification des huit photorécepteurs (R1 à R8).	7
Figure 2-5 :	Emplacement des photopigments sur les microvillies et des microvillies sur le photorécepteur (tirée de Rossel, 1989).	8
Figure 2-6 :	a) Réponse typique d'un photorécepteur. b) Stimulus lumineux (ordonnée non montrée).	8
Figure 2-7 :	Représentation des cellules d'une cartouche (tirée de van Hateren, 1987).	9
Figure 2-8 :	a) Réponse typique d'une LMC. b) Stimulus lumineux (ordonnée non montrée).	9
Figure 2-9 :	Liste des connexions synaptiques présentes dans lamina (tirée de Shaw, 1984).	10

Figure 2-10 : Principe de superposition. A) Vue en coupe de trois photorécepteurs (traits gras) et d'une cartouche (carré noir) participant au principe de superposition neuronale. B) Vue du dessus montrant les photorécepteurs (points noirs) de chaque facette (cercles blancs, F1 à F6) faisant connexions avec la cartouche (cercle noir). C) Vue de la numérotation des photorécepteurs (R1 à R8) dans une facette.....	11
Figure 2-11 : Connexions entre les parties rétine et lamina (tirée de Hardie, 1985).....	11
Figure 2-12 : Photographie d'une pseudopupille cornéenne.....	12
Figure 2-13 : Photographie d'une pseudopupille cornéenne réduite.....	13
Figure 2-14 : Photographie d'une pseudopupille profonde.....	13
Figure 3-1 : Organigramme du montage (partie matérielle).....	17
Figure 3-2 : Schéma optique d'un trajet lumineux.....	18
Figure 3-3 : Plan du stimulateur optique.....	21
Figure 3-4 : Organigramme du contrôleur de moteurs.....	30
Figure 3-5 : Organigramme de la carte maîtresse.....	30
Figure 3-6 : Organigramme de la carte moteur.....	35
Figure 3-7 : Diagramme temporel des séquences du générateur de séquences.....	39
Figure 3-8 : Diagramme temporel d'un événement du générateur de séquences.....	40
Figure 3-9 : Organigramme du générateur de séquences.....	41
Figure 3-10 : Organigramme de la carte séquenceur.....	42
Figure 3-11 : Organigramme de la génération des étapes de la machine à état secondaire.....	51
Figure 3-12 : Organigramme de la génération des événements de la machine à état secondaire.....	52
Figure 3-13 : Plan de l'obturateur.....	55
Figure 3-14 : Diagramme temporel du signal d'entrée et du signal de sortie du circuit de contrôle de l'obturateur.....	56
Figure 3-15 : Organigramme du circuit de contrôle de l'obturateur.....	56
Figure 3-16 : Diagramme temporel des signaux nécessaires au contrôle de l'obturateur.....	60
Figure 3-17 : Diagramme temporel des signaux électriques du circuit de compensation.....	63
Figure 3-18 : Organigramme du circuit de synchronisation.....	64

Figure 3-19 : Diagramme temporel des signaux du circuit de synchronisation.	66
Figure 3-20 : Conversion des signaux binaires en un signal analogique utilisé par le circuit de synchronisation.	67
Figure 3-21 : Photographie du montage (I).	71
Figure 3-22 : Photographie du montage (II).	71
Figure 3-23 : Photographie du montage (III).	71
Figure 4-1 : Organigramme de l'ordinateur principal.	75
Figure 4-2 : Algorithme de la boucle principale de l'application <i>Contrôleur de moteurs (68HC11)</i>	79
Figure 4-3 : Organigramme de l'application <i>Serveur de port RS-232</i>	83
Figure 4-4 : Algorithme d'une ligne de communication de l'application <i>Serveur de port RS-232</i>	84
Figure 4-5 : Organigramme de l'application <i>Contrôleur de moteurs (x86)</i>	86
Figure 4-6 : Fenêtre principale de l'application <i>Contrôleur de moteurs (x86)</i>	88
Figure 4-7 : Exemple de l'effet du jeu mécanique sur un déplacement linéaire.	90
Figure 4-8 : Algorithme de l'application <i>Générateur de séquences (68HC11)</i>	93
Figure 4-9 : Organigramme de l'application <i>Générateur de séquences (x86)</i>	95
Figure 4-10 : Fenêtre principale de l'application <i>Générateur de séquences (x86)</i>	96
Figure 4-11 : Fenêtre de configuration des séquenceurs de l'application <i>Générateur de séquences (x86)</i>	98
Figure 4-12 : Organigramme de l'application <i>Acquisition et traitement des courbes</i>	104
Figure 4-13 : Fenêtre de projet de l'application <i>Acquisition et traitements des courbes</i>	105
Figure 4-14 : Fenêtre de configuration d'un projet de l'application <i>Acquisition et traitement des courbes</i>	107
Figure 4-15 : Fenêtre principale de l'application <i>Acquisition et traitement des courbes</i>	110
Figure 4-16 : Fenêtre de traitement de l'application <i>Acquisition et traitement des courbes</i>	111
Figure 4-17 : Organigramme de l'application <i>Acquisition et traitement des images</i>	114
Figure 4-18 : Fenêtre de projet de l'application <i>Acquisition et traitement des images</i>	115
Figure 4-19 : Fenêtre de vue en coupe de l'application <i>Acquisition et traitement des images</i>	117

Figure 4-20 : Fenêtre de perspective de l'application <i>Acquisition et traitement des images</i>	117
Figure 4-21 : Fenêtre d'histogramme de l'application <i>Acquisition et traitement des images</i>	118
Figure 4-22 : Fenêtre de traitement de l'application <i>Acquisition et traitement des images</i>	119
Figure 4-23 : Fenêtre principale de l'application <i>Acquisition et traitement des images</i>	121
Figure 4-24 : Organigramme de l'application <i>Contrôle des expériences</i>	122
Figure 4-25 : Fenêtre principale de l'application <i>Contrôle des expériences</i>	124
Figure 4-26 : Représentation des contrôles relatifs à la manipulation de l'image de l'application <i>Contrôle des expériences</i>	125
Figure 4-27 : Exemple d'une relation entre le référentiel de l'image et le référentiel de la base où l'orientation et l'échelle sont différentes.	127
Figure 4-28 : Représentation des contrôles relatifs au déplacement de la base de l'application <i>Contrôle des expériences</i>	128
Figure 4-29 : Correspondance entre l'emplacement de la souris et le déplacement effectué par la base dans l'application <i>Contrôle des expériences</i>	130
Figure 4-30 : Représentation des contrôles du déplacement de la microélectrode de l'application <i>Contrôle des expériences</i>	131
Figure 4-31 : Représentation des contrôles du trajet de fond et du polariseur de l'application <i>Contrôle des expériences</i>	133
Figure 4-32 : Représentation des contrôles responsables des trajets lumineux de l'application <i>Contrôle des expériences</i>	133
Figure 4-33 : Représentation des contrôles responsables du protocole expérimental de l'application <i>Contrôle des expériences</i>	136
Figure 4-34 : Représentation de la recherche des facettes de l'application <i>Contrôle des expériences</i>	137
Figure 4-35 : Représentation de l'alignement d'une cellule. La courbe dentelée représente la réponse de la cellule, et la courbe continue le polynôme. Dans le présent cas, la réponse maximale se situe à -0.5°	138

- Figure 4-36 : Représentation des contrôles responsables de l'acquisition des courbes de l'application *Contrôle des expériences*.....139
- Figure 5-1 : Représentation de l'alignement de l'œil de l'insecte sur le support en tenant compte de l'orientation des microélectrodes (ligne pointillée), du trou (trait gras) et de la pseudopupille (point noir). A) Cas où l'insecte est correctement aligné. B) Cas où l'insecte est incorrectement aligné et où les microélectrodes ne peuvent passer par la pseudopupille cornéenne...143
- Figure 5-2 : Représentation de la modification de l'alignement de la pseudopupille cornéenne (ellipse) avec le centre de l'axe optique du microscope (flèche) et de la mise au point du microscope (ligne horizontale) à la suite d'une rotation de la base où le point noir indique le centre de rotation de la base. A) Cas avant rotation. B) Cas d'une rotation de 30° où le zéro de la base correspond au centre de l'œil. C) Cas d'une rotation de 30° où le zéro de la base diffère du centre de l'œil.....144
- Figure 5-3 : Représentation des cellules stimulées selon l'angle de convergence du faisceau lumineux. Les flèches représentent l'orientation des photorécepteurs, les lignes foncées, les photorécepteurs stimulés et les boîtes noires, les LMC et cartouches qui sont stimulés. A) Faisceaux lumineux générés par les trajets autres que le trajet de fond. B) Faisceau lumineux généré par le trajet de fond.147
- Figure 5-4 : Courbes obtenues lors de la compensation de la résistance de la microélectrode. A) Compensation insuffisante de la microélectrode. B) Bonne compensation de la microélectrode. C) Compensation exagérée de la microélectrode.148
- Figure 5-5 : Représentation des opérations de l'alignement grossier d'une cellule.....149
- Figure 5-6 : Représentation de la technique utilisée pour changer l'orientation de la base. Il ne s'agit que de pointer le centre de la pseudopupille cornéenne réduite avec la souris en ayant préalablement sélectionné un changement d'orientation de la base en relatif.....150

Figure 5-7 :	Représentation de la stimulation d'un photorécepteur, des LMC d'une même cartouche et d'une cartouche avec un faisceau lumineux de 20 μm de diamètre et un angle de convergence de $0,6^\circ$	152
Figure 5-8 :	Représentation du marquage des facettes participant à la superposition neuronale et faisant partie de la même pseudopupille cornéenne réduite.	153
Figure 6-1 :	Modèle électrique de l'organisation des photorécepteurs pour la stimulation lumineuse de deux d'entre eux.	157
Figure 6-2 :	Correspondance entre les paramètres du modèle électrique et le système visuel de la mouche.	158
Figure 6-3 :	Équivalent Thévenin du photorécepteur.....	158
Figure 6-4 :	Modèle simplifié pour la stimulation lumineuse d'un seul photorécepteur avec ou sans injection de courant dans ce même photorécepteur.....	163
Figure 6-5 :	Modèle simplifié pour la stimulation lumineuse de deux photorécepteurs. ...	165
Figure 6-6 :	Modèle simplifié pour la stimulation lumineuse de trois photorécepteurs....	166
Figure 6-7 :	Modèle simplifié pour la stimulation lumineuse des six photorécepteurs.....	167
Figure 6-8 :	Modèle simplifié pour la stimulation lumineuse des six photorécepteurs avec l'injection de courant dans un seul photorécepteur.	168
Figure 6-9 :	Modèle simplifié pour l'injection de courant dans un photorécepteur sans stimulation lumineuse.....	169
Figure 6-10 :	Modèle représentant la stimulation lumineuse d'un seul photorécepteur et comprenant les tensions et courants nécessaires à la réalisation du graphe de fluence.....	172
Figure 6-11 :	Graphe de fluence du modèle pour la stimulation lumineuse et l'injection de courant pour un seul photorécepteur.....	173
Figure C-1 :	Graphe de fluence pour la stimulation lumineuse d'un photorécepteur avec ou sans injection de courant dans ce photorécepteur.	208
Figure C-2 :	Graphe de fluence pour la stimulation lumineuse de deux photorécepteurs.....	209
Figure C-3 :	Graphe de fluence pour la stimulation lumineuse de trois photorécepteurs.....	209

Figure C-4 : Graphe de fluence pour la stimulation lumineuse de six photorécepteurs avec l'injection de courant dans un photorécepteur.	210
Figure C-5 : Graphe de fluence pour l'injection de courant dans un photorécepteur sans aucune stimulation lumineuse.....	211

LISTE DES TABLEAUX

Tableau 3-1 : Organisation des mémoires permanente et temporaire.....	45
Tableau 3-2 : Organisation des registres permanent et temporaire.....	47
Tableau 3-3 : Organigramme de la machine à état primaire.....	49
Tableau 3-4 : Interprétation des signaux de commande du courant dans le circuit de contrôle de l'obturateur.....	57
Tableau 4-1 : Statistiques concernant la réalisation des logiciels du projet.....	141
Tableau 6-1 : Notation des tensions du modèle, des équations de tension et des équations d'impédance.	162
Tableau 6-2 : Paramètres du segment interne.	170
Tableau 6-3 : Paramètres de l'axone.....	170
Tableau 6-4 : Paramètres de la cartouche.	171
Tableau 6-5 : Liste des tensions utilisées pour l'évaluation des paramètres à partir des résultats expérimentaux de van Hateren.	175
Tableau 6-6 : Liste des paramètres obtenus à partir des résultats expérimentaux de van Hateren.....	176
Tableau 6-7 : Valeurs expérimentales obtenues avec le montage.	177
Tableau 6-8 : Étude comparative entre une série de résultats obtenus avec notre montage et une série tirée des résultats expérimentaux de van Hateren (1986).....	178
Tableau 6-9 : Étude de sensibilité du modèle aux erreurs de mesures expérimentales.	181
Tableau C-1 : Données numériques utilisées pour compléter la résolution du modèle.....	207

LISTE DES SYMBOLES

° :	degré
°C :	degré Celsius
Ω :	ohms
A :	ampère
DMA :	Direct Memory Access
DLL :	Dynamic Link Library
COM :	Communication port
CTS :	Clear To Send
DDE :	Dynamic Data Exchange
DEL :	Diode Électroluminescence
GPIB :	General Purpose Interface Bus
LMC :	Large Monopolar Cell
M Ω :	mégaohms
MDI :	Multiple Document Interface
MHZ :	mégahertz
Hz :	hertz
KHz :	kilohertz
RTS :	Request To Send
UART :	Universal Asynchronous Receiver Transmitter
V :	volt
W :	watt
cm :	centimètre
ms :	milliseconde

mV :	millivolt
nm :	nanomètre
ns :	nanoseconde
μm :	micromètre
μs :	microseconde

CHAPITRE I

INTRODUCTION GÉNÉRALE

Beaucoup de recherches sont effectuées de par le monde pour tenter de comprendre le fonctionnement du système visuel des animaux. Les chercheurs doivent mener des expériences de plus en plus complexes et difficiles à réaliser. Pour ce faire, il devient important de mettre au point des montages automatisés qui pourront accélérer et simplifier la tâche des expérimentateurs.

Cette thèse de doctorat présente principalement la réalisation d'un stimulateur optique très sophistiqué qui permet d'automatiser en grande partie des expériences sur le système visuel d'insectes possédant des yeux à facettes. Nous y retrouvons aussi la solution théorique d'un modèle représentant les interconnexions de photorécepteurs dont les axones font des connexions entre eux. Ce modèle sera éventuellement validé ou invalidé par le stimulateur optique présenté dans la thèse.

Pour construire un stimulateur optique capable d'étudier le système visuel de la mouche, il faut préalablement comprendre la structure de ce système visuel, ce qui est fait dans le second chapitre. C'est en exploitant la structure et les propriétés optiques de l'œil de l'insecte qu'il

est possible d'implanter des fonctionnalités de haut niveau qui permettront de réaliser certaines étapes d'une expérience.

Dans le troisième chapitre, nous présentons l'aspect matériel du montage. Les différents modules mécaniques, électroniques et optiques y sont présentés. On pourra se rendre compte que ces modules sont très versatiles et qu'ils pourront être assignés à différentes tâches selon le bon vouloir de l'expérimentateur ou les exigences des logiciels de contrôle.

Le quatrième chapitre comprend l'organigramme des différents logiciels et montre les interrelations entre eux. La structure des logiciels repose sur quatre logiciels esclaves qui contrôlent les différents mécanismes du stimulateur optique et sur un cinquième logiciel maître qui communique avec ces quatre logiciels pour contrôler les différents aspects du montage. C'est en grande partie avec le logiciel maître que l'expérimentateur travaille et c'est aussi ce dernier logiciel qui réalise les algorithmes de haut niveau.

Lors de la réalisation d'une expérience, l'expérimentateur devra passer par une série d'opérations préliminaires qui sont décrites dans le cinquième chapitre. Nous y retrouvons les différentes opérations permettant de positionner l'animal sur le stimulateur, de trouver la cellule sous enregistrement, d'aligner la cellule sous enregistrement et d'identifier les facettes. Il est à souligner que certaines de ces étapes peuvent être esquivées selon le type d'expérience que l'expérimentateur désire effectuer.

Dans le sixième chapitre, nous présentons la résolution d'un modèle électrique d'une partie des photorécepteurs de l'œil de la mouche. La solution du modèle a été obtenue de façon à pouvoir le résoudre à l'aide de données expérimentales qui ne pourront que provenir d'un stimulateur optique aussi élaboré que celui présenté dans cette thèse.

Le dernier et septième chapitre comprend une conclusion générale et une discussion concernant les possibilités présentes et les améliorations futures du stimulateur optique.

CHAPITRE II

PHYSIOLOGIE DE LA MOUCHE DOMESTIQUE

2.1 Introduction

Une des premières décisions à prendre lors de l'étude du système visuel des animaux demeure le choix de l'animal proprement dit. Ce choix est très important et peut faire la différence entre la réussite et l'échec des recherches qui seront entreprises. Voici les critères qui nous ont conduit à la sélection de la mouche domestique comme sujet d'étude :



Figure 2-1 : Photographie d'une mouche domestique femelle (tirée de Hardie, 1985).

- Il fallait choisir un animal présentant les structures les plus simples tout en possédant un système visuel bien développé. Un tel système visuel permet à l'animal de prendre une image de l'espace et

d'en faire l'analyse pour ainsi identifier et localiser les composants de la scène comme la nourriture, les prédateurs, etc.

- L'utilisation d'un animal très étudié par divers groupes de recherche internationaux constitue un atout majeur. Cela permet de faire progresser plus rapidement la recherche et d'avoir accès à une banque de connaissances sur l'animal. Par exemple, il suffit de penser à tout le travail d'histologie effectué pour identifier et localiser les cellules nerveuses du système visuel de la mouche (Strausfeld, 1976).
- Quand on fait l'étude d'un animal, il est essentiel de le maintenir en vie lors d'une expérience. Les animaux possédant des yeux composés se prêtent bien à une étude *in vivo* et, contrairement aux animaux avec des yeux caméculaires, il n'est pas nécessaire de les disséquer pour atteindre les centres nerveux de l'œil ou du cerveau. Il en découle que la préparation de l'animal à l'expérience ne prend qu'une quinzaine de minutes, et l'expérience peut durer plusieurs heures.
- Contrairement à plusieurs expériences sur les vertébrés, il n'est pas nécessaire d'utiliser d'anesthésiques pour certains invertébrés comme les diptères. L'utilisation d'anesthésiques peut modifier le comportement des systèmes nerveux de façon imprévisible rendant ainsi l'analyse des données expérimentales très difficile.
- L'animal doit posséder des cellules nerveuses assez grosses pour que les techniques d'enregistrement intracellulaire fonctionnent. Il est aussi possible d'utiliser les techniques d'enregistrement extracellulaire sur les plus petites cellules, mais tel n'est pas toujours le cas.
- Pour les petits laboratoires, l'achat ou l'élevage d'animaux pour la recherche peut hypothéquer grandement le budget. La mouche demeure d'autant plus intéressante, car il n'en coûte que quelques dollars par semaine pour effectuer l'élevage, qui ne nécessite aucun personnel spécialisé. Ceci n'est pas le cas pour les laboratoires travaillant avec des animaux comme le chat, le singe, etc.

Puisque notre choix s'est arrêté sur la mouche domestique, le présent chapitre décrit certaines particularités de cet insecte qui sont liées au travail de notre thèse. La première section (2.2 L'organisation de l'œil de la mouche domestique) comprend une description des structures étudiées. La deuxième section (2.3 La superposition neuronale) présente le phénomène de superposition neuronale qui permet au montage de ne stimuler qu'un seul photorécepteur. La dernière section (2.4 Les pseudopupilles) montre un ensemble de phénomènes optiques utilisés lors de l'alignement de la mouche (voir section 5.5, page 149).

2.2 L'organisation de l'œil de la mouche domestique

La physiologie de l'œil de la mouche domestique fait voir une subdivision des structures nerveuses en différentes couches (Strausfeld, 1989) où chacune d'entre elles remplit un rôle très précis dans le traitement de l'information. L'information peut circuler d'une couche à l'autre et entre les structures d'une même couche. Plus les couches sont profondes, plus les structures et le traitement de l'information qui y est effectué sont

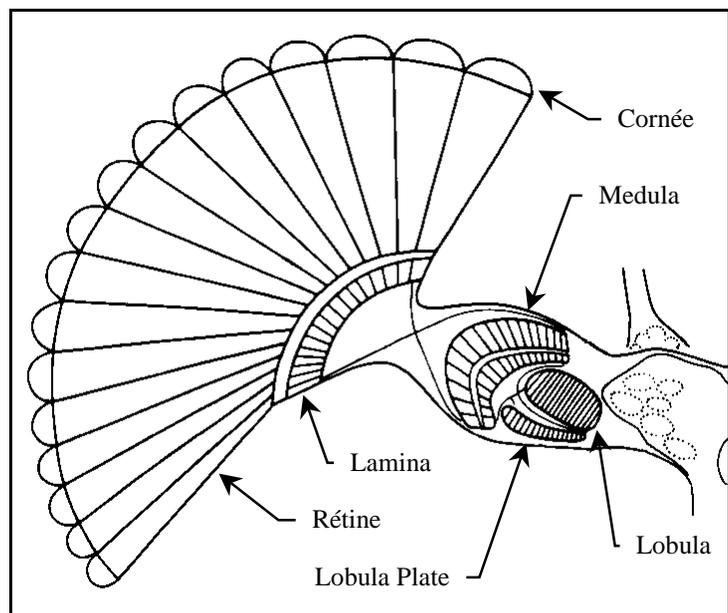


Figure 2-2 : Schéma représentant différentes couches de l'œil de la mouche domestique (tirée de Strausfeld, 1989).

complexes. La figure 2-2 présente un schéma des couches de l'œil de la mouche domestique ainsi que leur identification. La lumière provient de la partie gauche de la figure et passe à travers la cornée pour atteindre la première couche (rétine). Cette première couche a comme rôle principal de convertir l'intensité lumineuse en activité électrique. L'information électrique est par la suite transférée de couche en couche pour effectuer les différentes étapes du traitement de l'image. Dans certains cas, les connexions évitent certaines couches et peuvent aussi revenir en rétroaction dans des couches précédentes. Dans la présente thèse, seulement les niveaux rétine et lamina sont étudiés.

2.2.1 Rétine

La première couche de l'œil est subdivisée en 3000 structures presque identiques que l'on nomme ommatidies (voir figure 2-3). Leur emplacement sur un quart d'ellipsoïde permet un échantillonnage précis et uniforme de l'espace (Rossel, 1989).

L'ommatidie est une structure complexe comprenant plusieurs parties (voir figure 2-4, page suivante). Seules les parties jouant un rôle dans le traitement de l'information visuelle seront brièvement abordées. Pour obtenir une description plus complète, il suffit de se référer à la littérature (Hardie, 1985).

En premier lieu, la lumière pénètre dans l'œil en passant par une petite lentille que l'on nomme facette (f). Les facettes sont carrées

ou hexagonales, selon la région de l'œil. Les facettes font converger la lumière sur la région photosensible des photorécepteurs que l'on nomme rabdomère (r). Avant d'atteindre le rabdomère, la lumière passe dans une région aqueuse que l'on nomme pseudocône (pc). Le pseudocône est muni de murs formés par les cellules pigmentaires primaires (cpp) et d'un plancher formé des cellules de Semper (cs). Ces dernières sont munies de pigments visuels absorbant la lumière ne se rendant pas sur les rabdomères. On retrouve aussi des cellules pigmentaires séparant les ommatidies que l'on nomme cellules pigmentaires secondaires (cps). Les photorécepteurs contiennent aussi des pigments qui peuvent migrer près du rabdomère pour contrôler la quantité de lumière entrant dans les rabdomères (Howard et al., 1986).

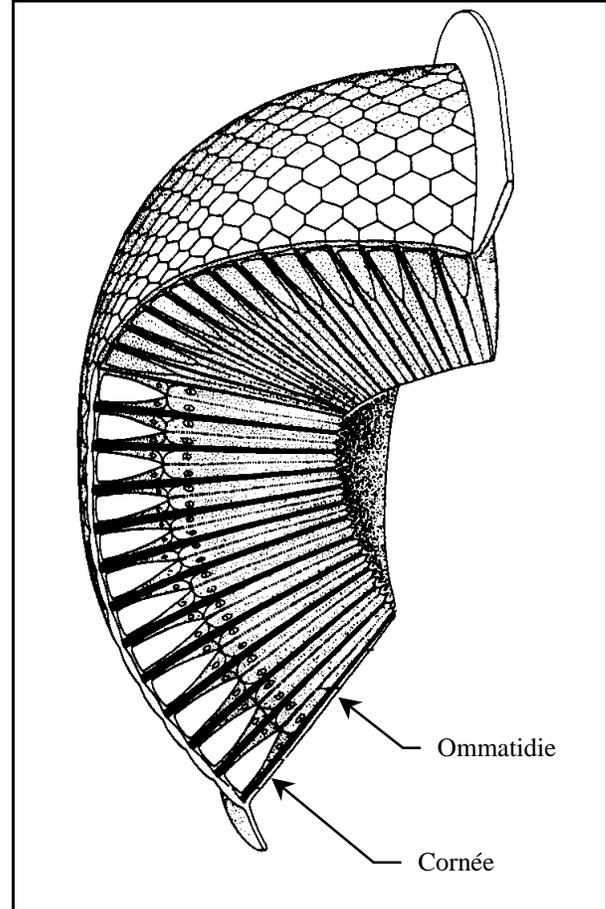


Figure 2-3 : Vue en coupe de la rétine de l'œil de la mouche domestique (tirée de Rossel, 1989).

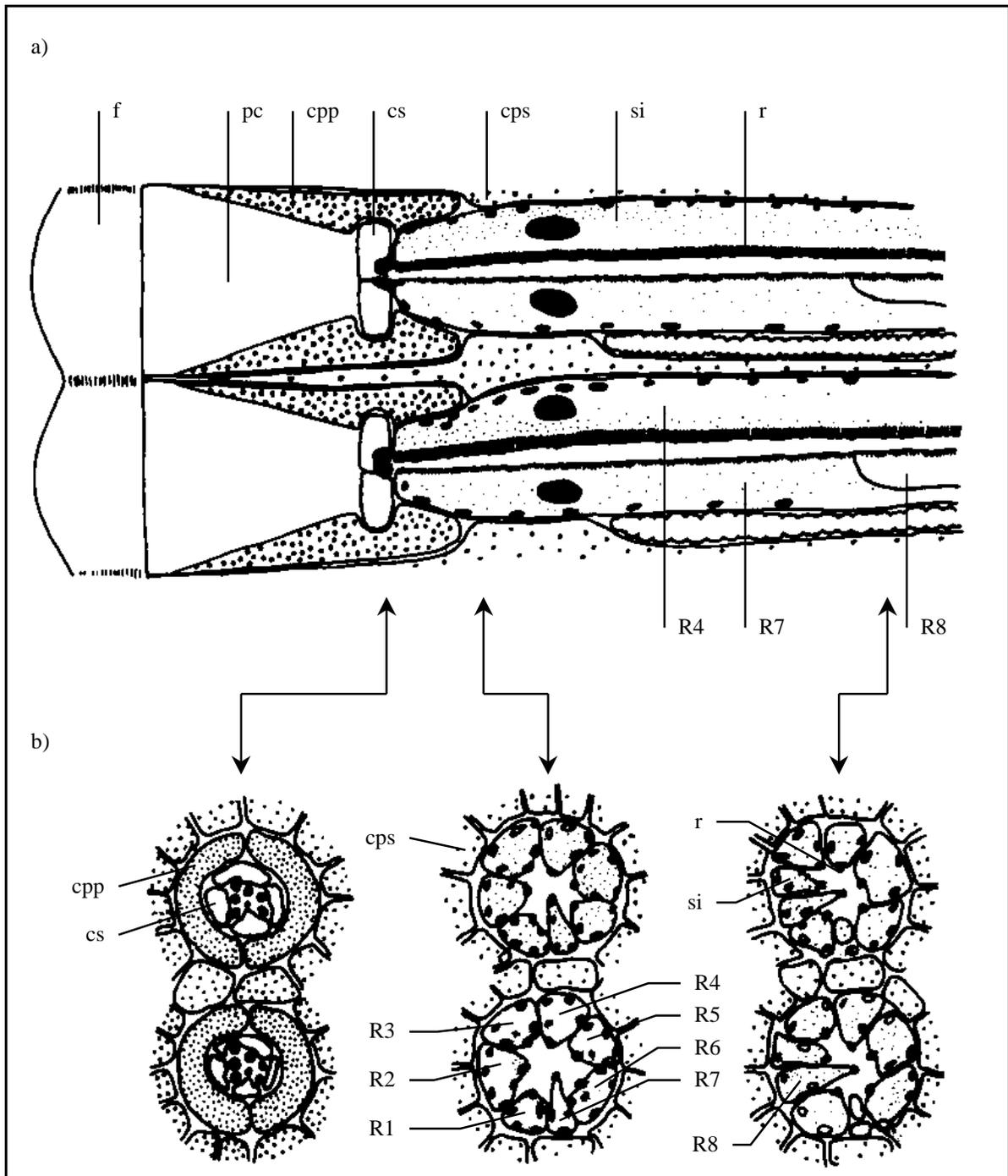


Figure 2-4 : Vue en coupe de deux ommatidies (tirée de Hardi, 1985). a) Vue longitudinale des deux ommatidies. b) Trois vues transversales prises le long de la vue longitudinale. L'emplacement des vues transversales dans la vue longitudinale sont indiquées par les flèches. Les différentes parties des ommatidies sont : facette (f), pseudocône (pc), cellules pigmentaires primaires (cpp), cellules de Semper (cs), cellules pigmentaires secondaires (cps), segment interne (si), rhabdomère ou segment externe (r) et l'identification des huit photorécepteurs (R1 à R8).

L'ommatidie est composée de huit photorécepteurs, numérotés R1 à R8, qui sont les cellules responsables de la conversion des photons en activité électrique. Plus précisément, la photoconversion est effectuée par les pigments visuels et, par la suite, la membrane de la cellule joue le rôle d'amplificateur pour générer des potentiels pouvant atteindre des déviations de 100 mV.

La figure 2-5 montre l'emplacement et l'orientation des photopigments sur les microvillis et l'emplacement des microvillis sur le photorécepteur qui forme le rhabdomère. Lorsque les photopigments sont orientés et que les microvillis possèdent tous la même orientation dans l'espace, le photorécepteur est sensible à la lumière polarisée (Rossel, 1989).

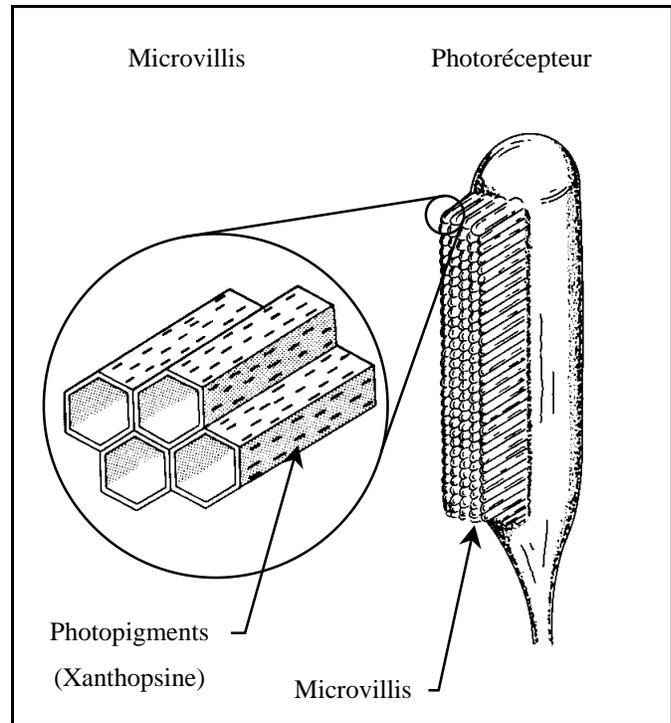


Figure 2-5 : Emplacement des photopigments sur les microvillis et des microvillis sur le photorécepteur (tirée de Rossel, 1989).

Enfin, la figure 2-6 montre une réponse électrique d'un photorécepteur à la suite d'une stimulation lumineuse de 200 ms. On peut constater que la cellule est normalement à la tension de repos de -30 mV et que la dépolarisation atteint 40 mV lorsque la lumière s'allume. Après quelques millisecondes, une hyperpolarisation de la cellule se produit, et la tension chute à 10 mV. Quand la lumière s'éteint, la tension revient rapidement à une tension de -20 mV puis décroît tranquillement vers la

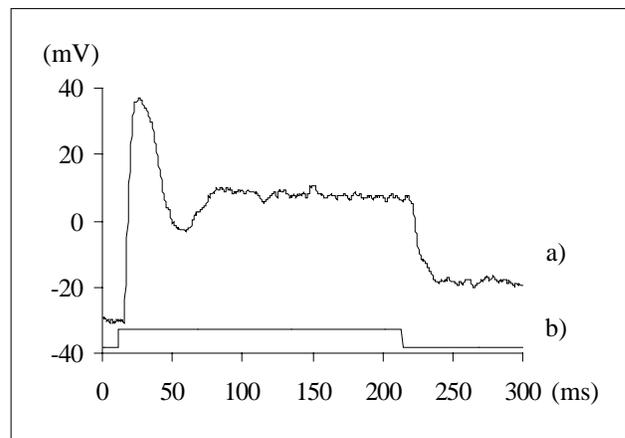


Figure 2-6 : a) Réponse typique d'un photorécepteur. b) Stimulus lumineux (ordonnée non montrée)

tension de repos initiale. La relation exacte entre la réponse électrique de la cellule et l'intensité lumineuse est très complexe.

2.2.2 Lamina

La seconde couche est beaucoup plus complexe que la première et comprend plusieurs types de cellules différents (voir figure 2-7). Tout comme pour la rétine, nous retrouvons une répétition régulière de structures que l'on nomme cartouche. Il y a approximativement une cartouche pour chaque ommatidie de la rétine. Parmi les cellules présentes dans une cartouche, on retrouve les grosses cellules monopolaires (L1 à L5), les amacrines (Am), les tangentielles (Tan1, Tan2), les fibres centrifugales (C2, C3) et la cellule en forme de panier (Basket Cells T1). De plus, pour parvenir à la seconde couche, les axones des photorécepteurs doivent traverser une membrane basale délimitant les deux couches.

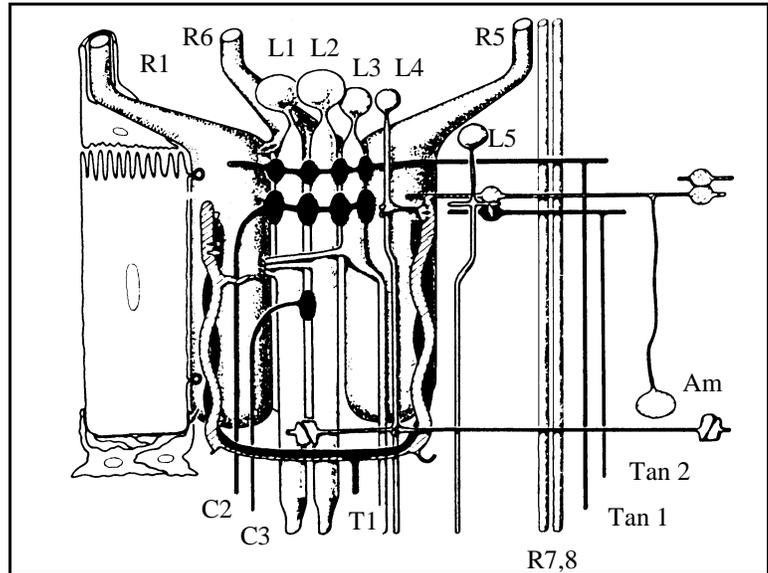


Figure 2-7 : Représentation des cellules d'une cartouche (tirée de van Hateren, 1987).

Vu la petitesse des cellules présentes dans la cartouche, il est très difficile de toutes les étudier. Par contre, les cellules L1 à L3 se prêtent bien à des techniques d'enregistrement intracellulaire, et les cellules L4 et L5 peuvent être étudiées à l'aide des techniques d'enregistrement extracellulaire au niveau de leur axone. La figure 2-8 montre la réponse typique d'une

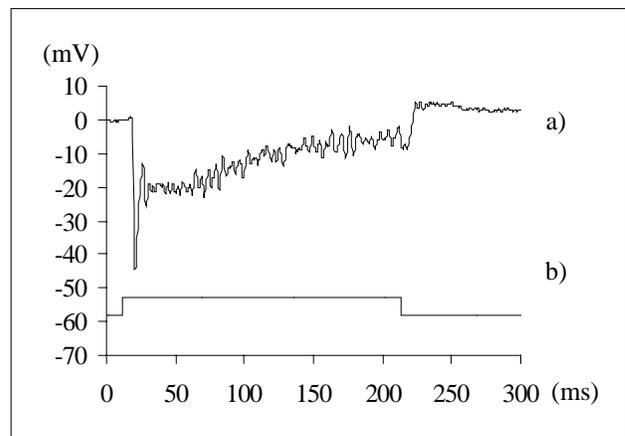


Figure 2-8 : a) Réponse typique d'une LMC. b) Stimulus lumineux (ordonnée non montrée)

LMC qui est obtenue par l'envoi d'une lumière sur les photorécepteurs qui font connexions avec la présente LMC. On peut remarquer que sa réponse est transitoire et qu'il y a hyperpolarisation lors de l'ouverture de la lumière et dépolarisation lors de la fermeture de la lumière. Il est aussi possible d'identifier les cellules L1, L2 et L3 selon la forme de leur réponse (Uusitalo et al., 1995).

La figure 2-7 de la page précédente montre aussi que ce ne sont pas tous les photorécepteurs qui sont connectés aux diverses cellules de lamina. On peut remarquer que les axones des photorécepteurs R7 et R8 passent directement dans une couche subséquente sans faire de connexion au niveau de lamina. Mais ce qui est le plus surprenant, c'est la façon dont les axones des photorécepteurs font connexion avec les différentes cellules de lamina. On verra qu'il y a superposition neuronale.

La figure 2-9 montre les différentes connexions que l'on retrouve dans lamina selon Shaw (1984). En regardant attentivement la figure, on remarque qu'il y a des rétroactions sur les photorécepteurs. Il s'ensuit que toute étude des photorécepteurs peut être modifiée selon l'importance de ces connexions. Aucune étude ne s'est encore attaquée à ce problème particulier.

2.3 La superposition neuronale

Contrairement à ce que l'on pourrait croire, les photorécepteurs de la mouche ne regardent pas tous dans une direction différente de l'espace. En réalité, pour chaque orientation donnée, il y a huit photorécepteurs qui sont alignés. Ces huit photorécepteurs (R1 à R8) ne font pas partie de la même facette. La figure 2-10-B de la page suivante montre les photorécepteurs regardant (points noirs d'où émerge une ligne) dans une même direction de l'espace et la facette où est localisé le photorécepteur. On peut voir que l'arrangement

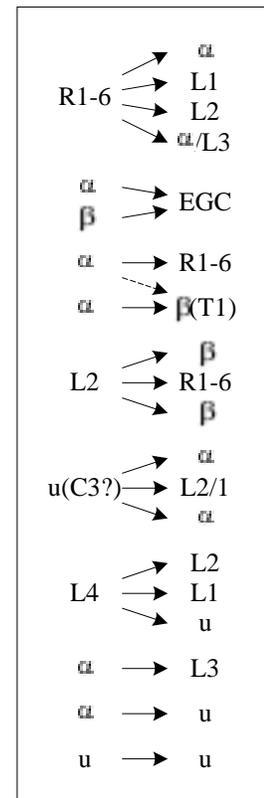


Figure 2-9 : Liste des connexions synaptiques présentes dans lamina (tirée de Shaw, 1984).

spatial des facettes est le même que celui des photorécepteurs. De plus, la position d'un photorécepteur par rapport à ses voisins dans une facette est la même que la position de la facette par rapport aux facettes voisines. Par exemple, le photorécepteur R1 de la facette F1 et le photorécepteur R3 de la facette F3 vont réunir leur information dans la même cartouche. La façon de numéroter les photorécepteurs est représentée à la figure 2-10-C, et la numérotation des facettes est représentée à la figure 2-10-B.

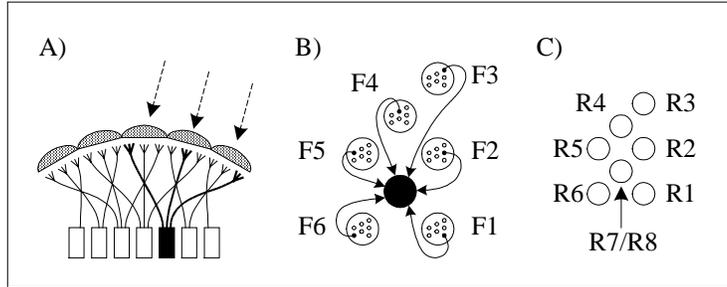


Figure 2-10 : Principe de superposition. A) Vue en coupe de trois photorécepteurs (traits gras) et d'une cartouche (carré noir) participant au principe de superposition neuronale. B) Vue du dessus montrant les photorécepteurs (points noirs) de chaque facette (cercles blancs, F1 à F6) faisant connexions avec la cartouche (cercle noir). C) Vue de la numérotation des photorécepteurs (R1 à R8) dans une facette.

Mais le plus surprenant, c'est que les axones des photorécepteurs regardant dans une même direction de l'espace vont faire connexion dans la même cartouche. La vue en coupe de la figure 2-10-A montre bien les photorécepteurs et la cartouche sollicités par une lumière parallèle. De son côté, la figure 2-11 montre le croisement effectué dans l'espace par les axones des photorécepteurs pour que la superposition neuronale se réalise.

Par contre, les photorécepteurs R8 ne participent pas à la superposition neuronale, et seulement les photorécepteurs R7 de type r (fluorescence rouge) qu'on retrouve seulement dans la région frontale de la mouche mâle sont impliqués dans la superposition neuronale (Hardie, 1985). La mouche domestique possède aussi des photorécepteurs R7 de type p (sans couleur) et y (fluorescence jaune) (voir Hardie, 1985).

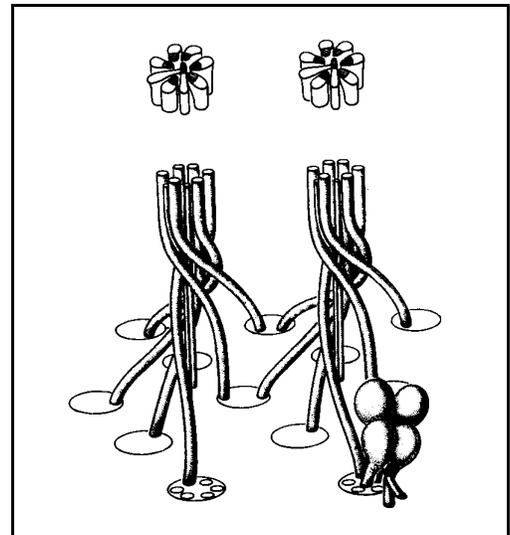


Figure 2-11 : Connexions entre les parties rétine et lamina (tirée de Hardie, 1985).

Lors de la réalisation d'une expérience, la connaissance de la superposition neuronale et l'apparition d'un phénomène optique (pseudopupille cornéenne réduite) permettent à l'expérimentateur de connaître l'emplacement des photorécepteurs et de les stimuler individuellement. La pseudopupille cornéenne réduite est en réalité un ensemble de facettes plus brillant que les autres et correspondant aux facettes participant à la superposition neuronale.

2.4 Les pseudopupilles

Le phénomène de pseudopupille est important, car il permet d'aligner la mouche et d'identifier l'emplacement des photorécepteurs. Puisqu'il existe plusieurs types de pseudopupilles, nous allons faire un bref survol de chacune d'entre elles tout en expliquant leur implication et la façon de les observer. (Pour plus d'information sur les pseudopupilles, voir Franceschini et Kirshfeld, 1971)

2.4.1 La pseudopupille cornéenne

La pseudopupille cornéenne (voir figure 2-12) est la plus facile à observer, elle nécessite seulement l'utilisation d'un microscope avec un épi-éclairage (éclairage dans l'axe d'observation). La pseudopupille cornéenne donne l'emplacement d'un groupe de facettes regardant dans l'axe optique du microscope. Le groupe de facettes contient quelques facettes de plus que le nombre nécessaire à la superposition neuronale. Par contre, avec des expériences d'électrophysiologie, il est très facile d'éliminer les facettes ne participant pas à la superposition neuronale. (voir section 5.6, page 152)

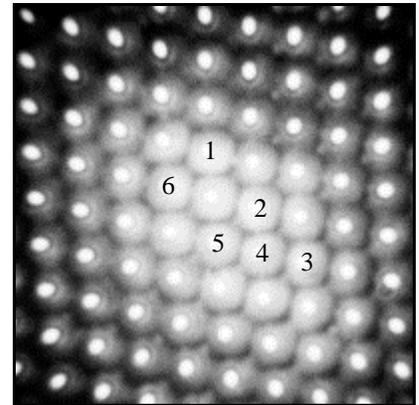


Figure 2-12 : Photographie d'une pseudopupille cornéenne.

Les numéros 1 à 6 de la figure 2-12 représentent les photorécepteurs et les facettes participant à la superposition neuronale. C'est donc la pseudopupille cornéenne qui sera principalement utilisée par le montage (voir section 5.6, page 152) pour identifier les cellules sous enregistrement.

2.4.2 La pseudopupille cornéenne réduite

La pseudopupille cornéenne réduite (voir figure 2-13) est très intéressante, car elle n'est composée que des facettes participant à la superposition neuronale. Cependant, elle est beaucoup plus difficile à percevoir. L'utilisation d'un épi-éclairage (éclairage dans l'axe d'observation) est suffisant, mais il faut que la lumière forme un faisceau parallèle et perpendiculaire à la pseudopupille cornéenne réduite.

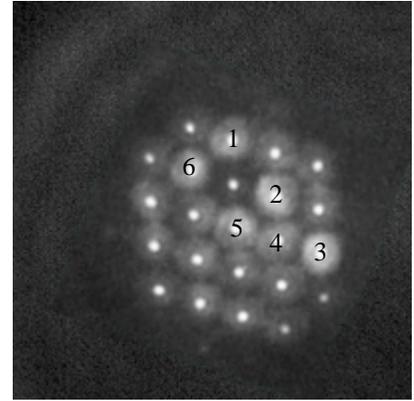


Figure 2-13 : Photographie d'une pseudopupille cornéenne réduite.

Tout comme dans la figure 2-12 de la page précédente, les numéros 1 à 6 de la figure 2-13 représentent les photorécepteurs et les facettes participant à la superposition neuronale. La présente pseudopupille est utilisée pour simplifier l'identification des facettes quand il est possible de la voir (voir section 5.6, page 152).

2.4.3 La pseudopupille profonde

Le dernier type de pseudopupille est très différent des deux autres. Il s'agit d'une image virtuelle créée par la convergence de tous les photorécepteurs en sept points au centre de la mouche. Pour la voir, il suffit de faire la mise au point du microscope au centre de l'oeil de la mouche. Un épi-éclairage peut être utilisé pour la percevoir, mais l'utilisation d'un éclairage antidromique la rend beaucoup plus visible. L'éclairage antidromique est créé en entrant une fibre optique par l'arrière de la tête de l'insecte. La figure 2-14 présente la pseudopupille profonde telle qu'elle est vue par le montage. Cette image n'est pas parfaite, car le montage n'a pas été construit dans le but d'observer la pseudopupille profonde.

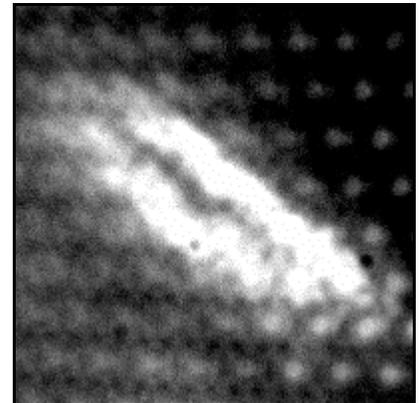


Figure 2-14 : Photographie d'une pseudopupille profonde.

Cette pseudopupille est utilisée pour trouver le centre de la tête de la mouche. Ce centre est important, car il s'agit du centre d'une sphère fictive formée par les photorécepteurs. Lorsque le centre de rotation de la base du montage est aligné avec le centre de la tête de la mouche,

une rotation de la mouche ne change pas la mise au point du microscope par rapport à la surface de l'œil de la mouche.

2.5 Conclusion

Le présent chapitre a permis de faire un très bref survol de la physiologie de la mouche domestique. Seuls les aspects physiologiques de la mouche qui sont nécessaires à la compréhension du montage ont été abordés. Par contre, le montage peut parfaitement être utilisé pour l'étude des autres centres nerveux de la mouche, comme les parties medula et lobula.

CHAPITRE III

LE MONTAGE (PARTIE MATÉRIELLE)

3.1 Introduction

Contrairement aux pratiques habituelles qui consistent à construire un montage très spécifique à l'expérience, le présent montage a été conçu afin de couvrir une assez grande gamme d'expériences. La polyvalence du montage doit permettre l'étude de plusieurs aspects du système visuel de certains animaux. La réalisation du montage a donc été faite dans l'optique de garder le plus de généralité possible.

Pour atteindre les objectifs de polyvalence, le montage doit être découpé en plusieurs modules interchangeables qui font chacun un travail bien précis et indépendant du type d'expérience à réaliser. Cependant, il doit y avoir un module principal qui commande tous les autres modules et qui permet la réalisation d'une expérience très spécifique. Le changement d'expérience se fait donc par une modification du module principal qui s'avère être un ordinateur. L'expérience effectuée dépendra donc du logiciel présent dans l'ordinateur. La réalisation de la partie matérielle du montage se limite à créer les modules nécessaires à l'ordinateur pour réaliser l'expérience proprement dite. L'interchangeabilité des modules permet de couvrir un plus grand nombre d'expériences tout en diminuant au minimum les changements à apporter au logiciel et en évitant de calibrer à nouveau le montage.

La structure du présent montage n'a pas été pensée dans le seul but d'en augmenter la polyvalence. Elle doit aussi permettre d'automatiser les expériences. L'automatisation permet de diminuer grandement le temps d'une expérience, de diminuer les risques d'erreur, de réaliser des expériences qui sont pratiquement impossibles à faire sans l'aide de l'ordinateur et de procéder à une vérification du montage avant chaque expérience pour s'assurer qu'il est toujours bien calibré. Ce dernier point est particulièrement important, car le montage comprend plusieurs pièces mécaniques ayant des ajustements sensibles et critiques au bon déroulement de l'expérience. Notons que l'automatisation de l'expérience n'est pas complète. Elle est entièrement autonome lors de la réalisation de l'expérience proprement dite, mais elle sert d'assistance lors des étapes préliminaires de l'expérience.

La gamme d'expériences visée par le présent montage consiste à faire l'étude du système visuel de divers insectes. Pour ce faire, le montage génère des stimuli lumineux et électriques tout en enregistrant l'activité électrique des cellules du système visuel. Dans la présente thèse, les expériences sont essentiellement pratiquées sur la mouche, mais rien n'empêche l'utilisation du présent montage pour étudier d'autres insectes ou d'autres animaux.

Plus concrètement, le montage permet de stimuler, avec des faisceaux lumineux, d'un à trois photorécepteurs de la rétine simultanément tout en contrôlant la lumière de fond de l'expérience. Pour chacun des faisceaux lumineux, il est possible de contrôler la position, le diamètre, la longueur d'onde, l'intensité lumineuse, l'angle du faisceau, la polarisation de la lumière et la durée de la stimulation. Il est aussi possible d'invertir les sources lumineuses. En effet, on peut utiliser des D.E.L., des lampes halogènes, des LASER ou des lampes éclair de très grande intensité. L'ajout d'une caméra au montage permet de voir en tout temps l'emplacement et le diamètre de chacun des faisceaux lumineux. L'enregistrement dans les différentes parties du système visuel de la mouche est possible grâce à l'utilisation de microélectrodes intracellulaires et extracellulaires. L'utilisation d'une carte d'acquisition permet à l'ordinateur d'enregistrer les différentes réponses obtenues à la suite d'une stimulation de même que l'état, ouvert ou fermé, de chacun des stimuli lumineux.

Puisque le montage est très polyvalent, trois versions différentes ont été construites pour étudier les divers aspects du système visuel de la mouche. Le présent chapitre décrit le premier des trois montages réalisés et les modifications qu'il est possible d'apporter à chacun des modules. La figure 3-1 donne un aperçu des différents modules et les interrelations entre chacun d'eux. Les flèches indiquent dans quelle direction circule l'information entre les modules. Les numéros inscrits dans les boîtes indiquent la section du chapitre où le module est présenté. Les boîtes contenant le symbole * possèdent des contrôles manuels qui ne sont pas accessibles par le module principal. Les boîtes contenant le symbole ° possèdent un ordinateur intégré qui exécute un ou plusieurs des logiciels du chapitre IV.

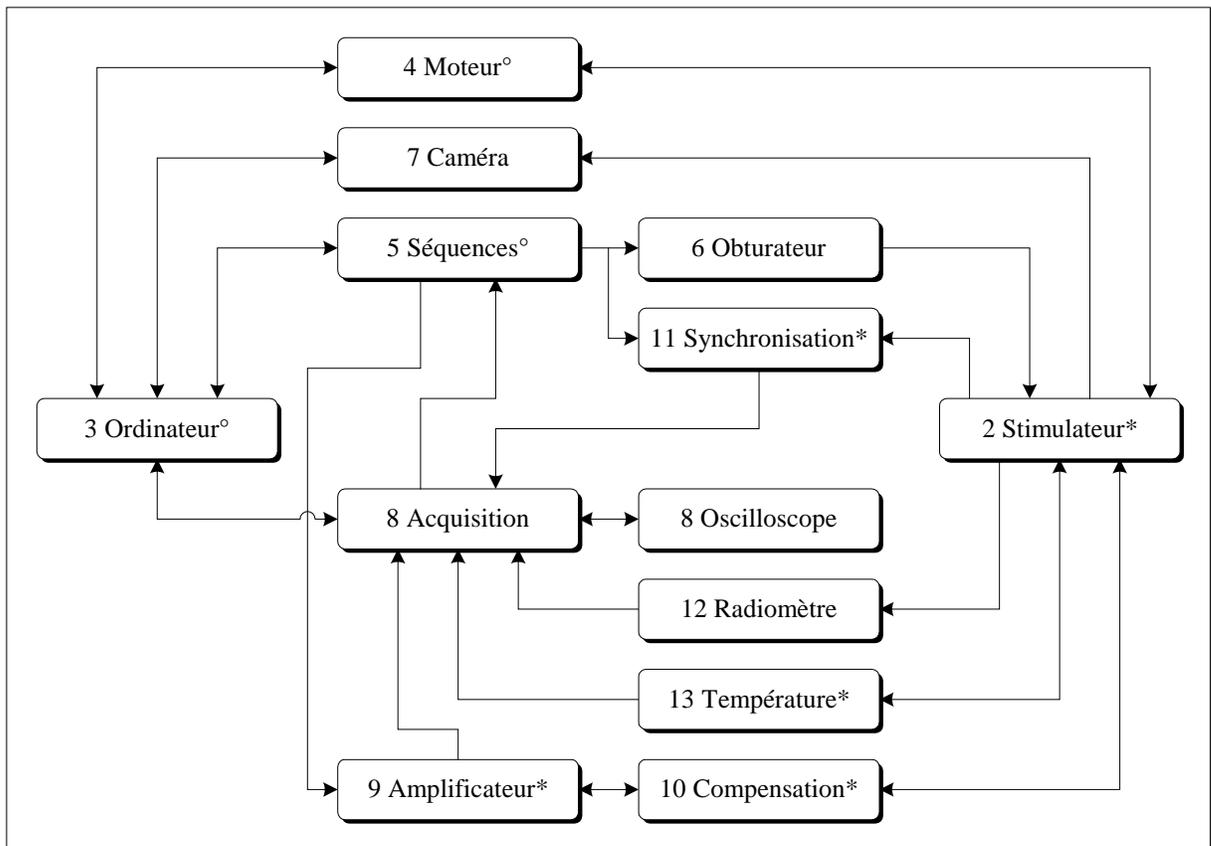


Figure 3-1 : Organigramme du montage (partie matérielle).

3.2 Technique de stimulation des photorécepteurs

La première et la seconde version du montage qui ont été réalisées, exploitent le phénomène de superposition neuronale en s'aidant des pseudopupilles cornéennes réduites (PPCR) pour parvenir à stimuler un seul photorécepteur. En contrôlant l'angle et la position du faisceau lumineux il est alors possible de stimuler un seul photorécepteur. Le choix d'une facette appartenant à une PPCR qui reçoit une lumière parallèle, permet de choisir un des huit photorécepteurs participant à la superposition neuronale (voir figure 2-10, page 11). Afin de stimuler une seule facette et un seul photorécepteur par facette, il faut générer un faisceau

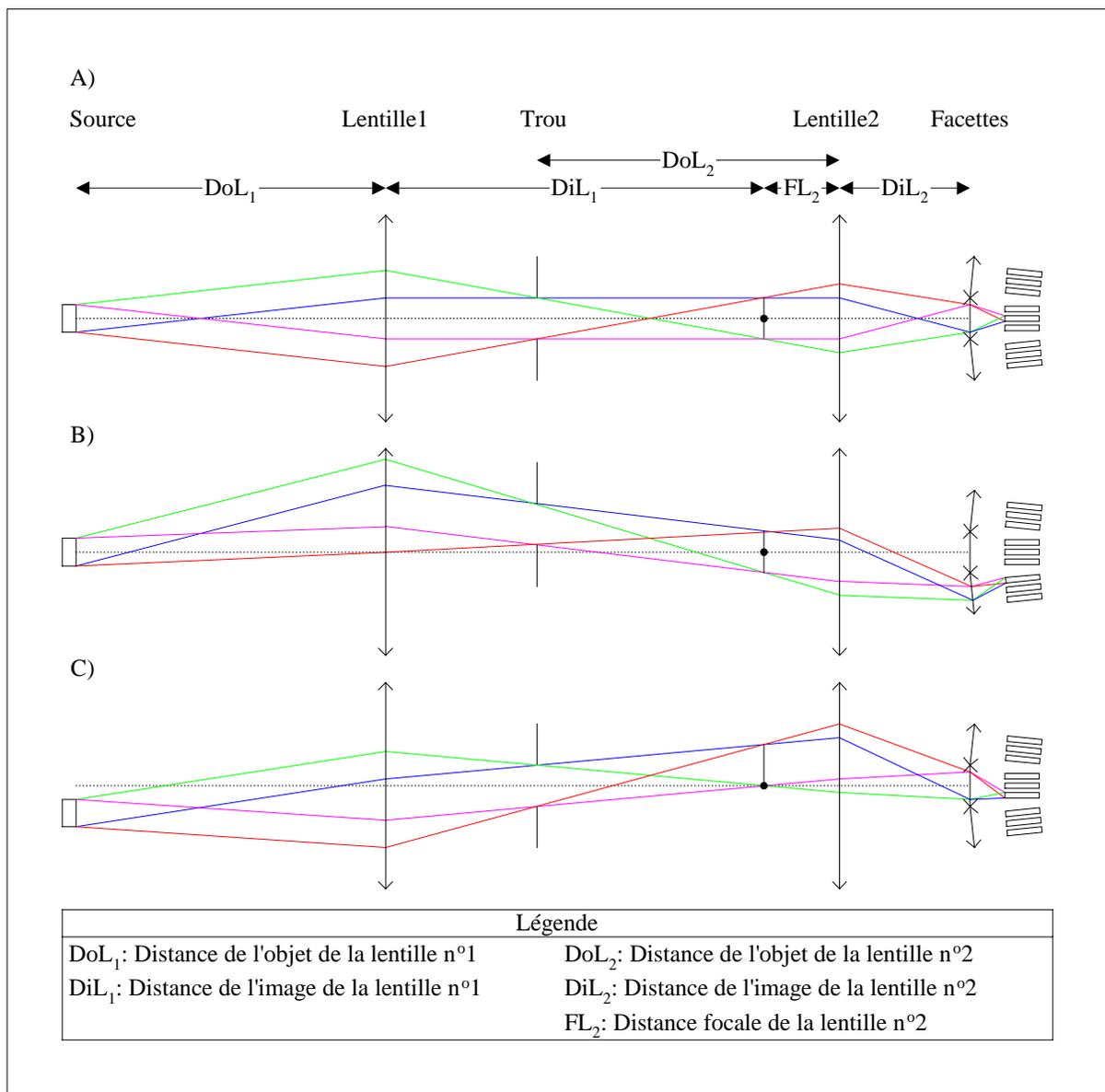


Figure 3-2 : Schéma optique d'un trajet lumineux.

lumineux ayant un diamètre inférieur à celui de la facette et possédant un angle de divergence inférieur à l'angle entre les photorécepteurs.

La figure 3-2-A de la page précédente illustre les caractéristiques du trajet lumineux qui permettent de remplir les deux conditions précédentes. Afin de générer un faisceau lumineux ayant un diamètre bien précis, qui soit inférieur à celui de la facette, il faut placer un trou dans le trajet lumineux de telle sorte qu'il soit projeté sur la facette. Le diamètre du faisceau lumineux est déterminé par les lois d'optique qui prédisent l'emplacement et le diamètre de l'image du trou en fonction de l'emplacement et du diamètre du trou selon la lentille utilisée. Les équations donnant l'emplacement de l'image et le diamètre du faisceau lumineux sont :

$$\text{Emplacement image } (DiL_2) = \frac{DoL_2 \times FL_2}{DoL_2 - FL_2}$$

$$\text{Diamètre faisceau} = \frac{DoL_2 - FL_2}{FL_2} \times \text{Diamètre trou}$$

La deuxième condition, soit l'angle de divergence, est déterminée par le diamètre de la source lumineuse. Plus le diamètre de la source lumineuse est grand, plus l'angle de divergence est grand. Puisqu'il est impossible d'avoir une source ponctuelle, il faut que le diamètre de la source lumineuse soit assez petit pour que l'angle de divergence soit plus petit que l'angle entre les photorécepteurs. D'un autre côté, le diamètre doit être assez grand pour que la quantité de lumière soit suffisante pour stimuler adéquatement le photorécepteur. Pour que l'image de la source sortant de la lentille F_2 soit projetée à l'infini, il faut que la première image de la source arrive au point focal de la lentille F_2 . L'angle de divergence est donné par l'équation suivante :

$$\text{Angle} = \tan \left(\frac{DoL_1 - FL_1}{FL_1 \times FL_2} \times \text{Diamètre source} \right)$$

Cette technique permet aussi de modifier séparément l'emplacement et l'angle du faisceau lumineux. Il suffit de déplacer verticalement respectivement le trou ou la source lumineuse. Il est à noter qu'un déplacement du trou ne change pas l'angle du faisceau lumineux (voir figure 3-2-B, page 18) et qu'un changement de l'angle ne modifie pas l'emplacement du trou (voir figure 3-2-C, page 18). Les équations permettant de lier l'emplacement du faisceau lumineux à l'emplacement du trou ainsi que l'angle du faisceau lumineux à l'emplacement de la source lumineuse sont :

$$Position\ faisceau = \frac{DoL_2 - FL_2}{FL_2} \times Position\ trou$$

$$Angle = \tan\left(\frac{DoL_1 - FL_1}{FL_1 \times FL_2} \times Position\ source\right)$$

Une troisième version du montage a aussi été réalisée en utilisant le principe de neutralisation optique (van Hateren, 1984). Dans ce montage, l'optique est différente mais les logiciels et l'électronique demeurent les mêmes. L'utilisation de l'un ou l'autre des montages dépend du type d'expérience que l'on veut effectuer. Certaines expériences peuvent être réalisées sur les deux types de montages, ce qui permet de mieux déceler la présence d'artefacts et de comparer les résultats expérimentaux selon deux techniques différentes.

Les deux premiers montages permettent d'utiliser des intensités lumineuses beaucoup plus grandes qu'avec le montage de neutralisation optique. La principale difficulté d'utilisation réside dans la production rapide de la pseudopupille cornéenne réduite pour une cellule sous enregistrement. Le montage utilisant le principe de neutralisation optique permet de sélectionner rapidement les photorécepteurs car on les voit directement. La stimulation lumineuse est toutefois de moindre qualité car l'optique de la mouche qui est de meilleure qualité que notre microscope est neutralisée.

3.3 Stimulateur optique

Le stimulateur optique, qui comprend les parties optiques et mécaniques du montage, est le module qui contient tous les mécanismes nécessaires au contrôle des faisceaux lumineux, au déplacement et à l'orientation de la préparation, au déplacement des microélectrodes, l'optique nécessaire à la visualisation de la préparation et tous les détecteurs nécessaires au suivi de l'expérience. La figure 3-3 présente un schéma simplifié du stimulateur optique. Les sections qui suivent décrivent chacune des parties. Mais avant, voici une brève description du fonctionnement du stimulateur.

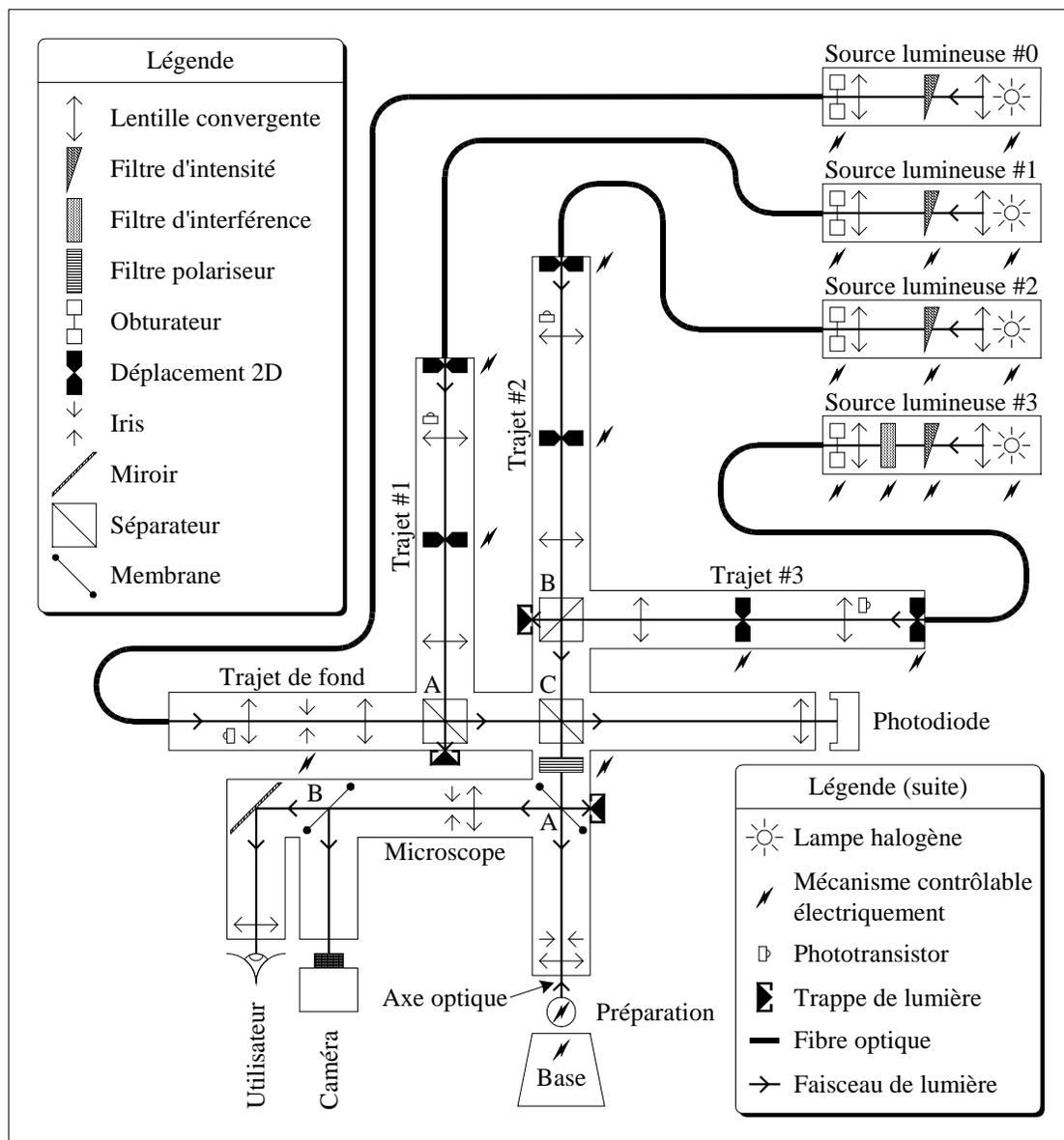


Figure 3-3 : Plan du stimulateur optique.

Le stimulateur est muni de sources lumineuses permettant de contrôler l'intensité, la longueur d'onde et la durée des stimuli lumineux. Les sources lumineuses injectent leur lumière dans les trajets lumineux à travers des fibres optiques. Les trajets lumineux génèrent les faisceaux lumineux et les dirigent vers la préparation avec une position, un diamètre et un angle donnés. Les faisceaux lumineux sont aussi présentés sur une photodiode afin d'en connaître l'intensité lumineuse. Il est aussi possible de contrôler la polarisation de la lumière des quatre trajets avec un polariseur. La préparation est installée sur une base permettant de la positionner et de l'orienter correctement dans l'espace. La base est munie de mécanismes permettant de faire pénétrer les microélectrodes dans la préparation. Il est aussi possible d'ajouter sur la base un système permettant de contrôler la température de la préparation. Enfin, le stimulateur optique est muni d'un microscope permettant de voir la préparation ainsi que la position des faisceaux lumineux par l'intermédiaire d'un oculaire ou d'une caméra.

3.3.1 Les sources lumineuses

Dans le présent montage, une source lumineuse est définie comme un ensemble permettant de générer de la lumière, tout en contrôlant l'intensité et la longueur d'onde. Elle est aussi munie d'un dispositif lui permettant de s'allumer ou de s'éteindre rapidement. La lumière émise par la source alimente le montage par l'intermédiaire d'une fibre optique.

Lors de la génération de la lumière, il est possible d'utiliser différents types de lampes. Dans le présent montage, l'utilisation de lampes halogènes de 20 watts et 100 watts est suffisante, mais, dans certaines expériences nécessitant une intensité lumineuse très grande, il est possible d'utiliser une lampe éclair de 1000 watts.

Afin de bien contrôler l'intensité lumineuse, les sources possèdent un filtre neutre variable contrôlé manuellement pour la source #0 et à l'aide de moteurs électriques pour les autres sources. Les moteurs électriques permettent à l'ordinateur de changer l'intensité lumineuse lors d'une expérience. À l'aide de la photodiode et des caractéristiques des filtres de densité neutre, l'ordinateur peut connaître l'intensité lumineuse appliquée à la préparation et ainsi calibrer chacun des trajets pour qu'ils aient tous la même intensité lumineuse.

Il est possible de mesurer la réponse spectrale d'une cellule, car une des quatre sources lumineuses est munie d'une série de filtres d'interférence. Il y a trente et un filtres d'interférence qui couvrent une plage allant de 400 à 700 nm par sauts de 10 nm. Les filtres sont montés sur une roue que l'ordinateur tourne pour choisir le filtre d'interférence désiré. Évidemment, l'ordinateur connaît l'emplacement de chacun des filtres sur la roue et leurs caractéristiques.

Finalement, les sources sont munies d'obturateurs mécaniques permettant l'ouverture et la fermeture des lumières. Le design des obturateurs a spécialement été fait pour que les temps d'ouverture et de fermeture soient très rapides, c'est-à-dire inférieurs à 1 ms. Les obturateurs sont contrôlés par le générateur de séquences qui permet d'obtenir une synchronisation très précise entre l'ouverture et la fermeture des faisceaux lumineux.

Les sources lumineuses auraient pu être installées directement au bout de chacun des trajets lumineux évitant ainsi l'utilisation de fibres optiques. Toutefois, les raisons qui nous ont persuadé de choisir des fibres optiques sont :

- L'intégration des sources lumineuses aux trajets lumineux aurait fait en sorte qu'il aurait été impossible de modifier la source lumineuse sans calibrer à nouveau le trajet lumineux en entier. Dans le même ordre d'idées, il aurait fallu calibrer le trajet au complet lorsqu'on aurait eu à changer une lampe brûlée.
- Lors du fonctionnement des obturateurs, beaucoup de vibrations sont générées. Ces vibrations ont tendance à modifier la calibration de l'ensemble du montage avec le temps et elles font vibrer la préparation. La vibration de la préparation peut faire avorter une expérience en déplaçant les microélectrodes. Cela va de soi que les sources lumineuses sont mécaniquement isolées du montage et qu'elles sont construites pour être insensibles aux vibrations.
- De plus, la construction des trajets lumineux est faite de telle sorte que les trajets doivent contrôler l'angle du faisceau lumineux arrivant sur la mouche. Or, s'il n'y avait pas de

fibres, le trajet serait obligé de déplacer l'ensemble des mécanismes présents dans la source. Il serait toutefois possible de ne déplacer que la lampe, mais cela résulterait en une modification de l'intensité lumineuse selon l'emplacement de la lampe.

- Enfin, l'utilisation de fibres optiques, contrairement à l'utilisation directe d'une lampe, permet d'obtenir plus facilement une source lumineuse uniforme à l'entrée du trajet. L'uniformité de la source lumineuse permet d'obtenir une intensité lumineuse constante sur toute la surface du faisceau lumineux.

3.3.2 Les trajets lumineux

Afin de générer des faisceaux lumineux ayant un diamètre très précis et une orientation donnée par rapport à la surface de la préparation, il est nécessaire d'utiliser plusieurs pièces d'optique. L'ensemble de ces pièces constitue un trajet lumineux. Les trajets lumineux ont des parties exclusives et des parties communes.

Au tout début, la lumière, arrivant par les fibres optiques, entre dans la partie exclusive des trajets. Après avoir passé le premier séparateur de faisceaux, les faisceaux lumineux entrent dans une première partie commune. Le faisceau lumineux du trajet de fond et celui du trajet #1 passent dans la première partie commune et les faisceaux lumineux des deux autres trajets passent dans l'autre partie commune. Après que les faisceaux aient traversé un second séparateur de faisceaux, tous se retrouvent dans une même partie commune.

Les séparateurs de faisceaux ont comme objectif de réunir ou de diviser les faisceaux lumineux. Lorsqu'un faisceau entre dans un séparateur, il y a un pourcentage de la lumière qui est réfléchi et le reste du pourcentage est transmis. Dans le cas où les pourcentages sont de 50%, l'intensité lumineuse est diminuée de 50%. Puisqu'il y a deux séparateurs de faisceaux dans un trajet, l'intensité du faisceau lumineux est diminuée de 75%.

Dans les parties exclusives des trajets, il y a différents contrôles permettant d'orienter et de déplacer un faisceau lumineux. Dans le cas du trajet de fond, il est seulement possible de changer le diamètre du faisceau lumineux à l'aide d'un iris contrôlé par l'ordinateur. Le

présent trajet est voué à l'éclairage global de la mouche. Dans les trois autres trajets, il est possible de contrôler l'angle des faisceaux lumineux par l'intermédiaire du déplacement 2D situé à l'extrémité du trajet. En réalité, la fibre est installée sur le déplacement et c'est le déplacement de la fibre qui contrôle l'angle du faisceau arrivant sur la préparation. Plus au centre du trajet, il y a un second déplacement 2D qui permet de déplacer le faisceau sur la préparation. Le déplacement possède deux trous, de diamètres différents, placés l'un à côté de l'autre. Les diamètres des trous sont tels qu'ils permettent d'éclairer une facette ou une pseudopupille cornéenne réduite. Le déplacement des faisceaux se fait en déplaçant les trous qui sont suffisamment éloignés pour qu'ils n'apparaissent pas simultanément sur la préparation. Le contrôle des déplacements 2D est fait par l'ordinateur.

Dans la partie commune, il est possible d'insérer un polariseur pour contrôler l'angle de polarisation des faisceaux lumineux. L'avantage de placer le filtre dans la région commune des trajets réside dans l'utilisation d'un seul filtre pour tous les trajets. L'inconvénient est qu'on ne peut avoir plusieurs faisceaux lumineux simultanés avec des angles de polarisation différents. Le contrôle de l'angle du polariseur est fait par l'ordinateur.

Enfin, chacun des trajets lumineux est muni d'un phototransistor permettant de connaître le moment exact où la source lumineuse s'allume ou s'éteint. (voir section 3.12, page 64.)

3.3.3 La détection de l'intensité lumineuse

Lors d'une expérience, il est essentiel de pouvoir ajuster les intensités de tous les faisceaux lumineux pour qu'elles soient identiques. Pour ce faire, il est nécessaire de posséder un élément permettant de mesurer la quantité de lumière arrivant sur la préparation pour chacun des trajets.

Pour bien mesurer l'intensité des faisceaux lumineux, il serait approprié d'installer un photodétecteur au même endroit que la préparation. L'impossibilité de placer la préparation et le photodétecteur au même endroit et en même temps nécessite une autre approche. Il serait toujours possible de faire une calibration de l'intensité lumineuse en plaçant temporairement un photodétecteur à la place de la préparation, mais cela n'est pas pratique. La meilleure

solution demeure la duplication de la dernière partie commune des trajets lumineux qui permet de placer un photodétecteur qui voit presque la même chose que la préparation. La seule différence entre le trajet original et le nouveau trajet demeure la présence d'une membrane optique supplémentaire dans le trajet original. Cela ne change en rien la comparaison entre les faisceaux lumineux et connaissant le facteur d'atténuation de la membrane, il est quand même possible d'obtenir une très bonne approximation de l'intensité lumineuse arrivant sur la préparation.

Finalement, la surface de la photodiode doit être assez grande pour que les faisceaux lumineux demeurent toujours sur la photodiode, peu importe leur position et leur diamètre. La section 3.13 de la page 68 donne plus d'information sur le fonctionnement du photodétecteur.

3.3.4 La base

Puisque les faisceaux lumineux couvrent une région limitée de l'espace, il est nécessaire de positionner précisément la préparation au centre de la région couverte par ces faisceaux tout en maintenant leur orientation perpendiculaire à la surface de la préparation. La base permet donc de déplacer et d'orienter la préparation. Elle permet aussi de déplacer les microélectrodes nécessaires à l'enregistrement des réponses des cellules nerveuses. Nous pouvons aussi y installer un dispositif permettant de mesurer et de contrôler la température de la préparation. Dans le dernier cas, la construction d'un habitacle autour de la préparation permettrait de mieux contrôler la température.

La base comprend une première série de déplacements permettant de changer l'emplacement de la préparation, des microélectrodes et du contrôle de la température simultanément. Ces déplacements sont complètement contrôlés par l'ordinateur et permettent des mouvements selon les axes X, Y et Z de l'espace ainsi que la rotation de la préparation sur elle-même autour des axes X et Y qui sont respectivement appelés phi et thêta.

Pour que la préparation tourne sur elle-même, il faut qu'elle soit placée au centre de rotation de phi et thêta. Pour ce faire, il y a trois autres déplacements manuels qui permettent de

déplacer la préparation par rapport à la base. De plus, la préparation est montée sur un petit support permettant un ajustement grossier de l'angle de la préparation.

Afin de déplacer les microélectrodes pour les faire pénétrer dans la préparation, il faut deux autres séries de trois déplacements. La première série de déplacements permet la manipulation de la microélectrode de référence et la seconde série sert à la manipulation de l'électrode de mesure. Le déplacement permettant de faire pénétrer la microélectrode de mesure est automatisé et permet de connaître une estimation de la profondeur de la microélectrode (voir section 4.11.5, page 131). Dans le cas où le montage nécessite deux microélectrodes de mesure, celles-ci sont collées ensemble et déplacées simultanément.

3.3.5 Le microscope

Ce n'est pas tout de pouvoir placer la préparation dans la région de l'espace couverte par les faisceaux à l'aide de la base, il faut aussi voir l'emplacement de la préparation dans l'espace. De plus, il est indispensable d'avoir une image de la préparation pour positionner les faisceaux lumineux sur la préparation lors d'une expérience.

Puisque le diamètre de la région couverte par les faisceaux est d'environ 200 μm et que la précision de l'emplacement des faisceaux est d'environ 1 μm , il est essentiel de munir le stimulateur optique d'un microscope. Le microscope permet à l'utilisateur de voir la surface de la préparation en regardant directement dans l'oculaire ou par l'intermédiaire d'une caméra. L'ajustement de la position de la préparation selon l'axe des X et l'axe des Y se fait directement en regardant l'image. La mise au point de l'image permet d'ajuster la position de la préparation selon l'axe des Z. Cependant, l'orientation de la préparation ne peut être faite directement en regardant dans le microscope. Différentes techniques utilisées pour l'alignement de la préparation sont abordées dans la section 5.5 de la page 149.

Le microscope est aussi muni de deux iris permettant d'améliorer la qualité de l'image. L'ajustement des iris est nécessaire pour réussir à bien voir la pseudopupille cornéenne réduite (voir section 2.4.2, page 13).

3.3.6 Discussion

Plusieurs mécanismes que l'on retrouve dans le stimulateur optique sont très spécifiques et introuvables dans le milieu commercial. Il a donc été nécessaire de construire plusieurs de ces mécanismes. Certains d'entre eux sont relativement simples, mais d'autres sont beaucoup plus complexes et ont nécessité la réalisation de plusieurs prototypes pour atteindre les performances voulues. La réalisation des différents mécanismes a été rendue possible grâce à la collaboration des membres de l'atelier de mécanique du département de génie électrique. L'ingéniosité des personnes présentes dans ce service et leur bonne collaboration ont permis d'envisager plusieurs solutions plausibles. Par contre, la construction des pièces a été entièrement faite par les membres de l'atelier et la validation des prototypes, par nous. La quantité de travail est énorme. À titre d'exemple, la réalisation d'une dizaine de prototypes a nécessité le design de 107 pièces différentes et la construction d'environ 234 pièces au total.

Parmi les réalisations effectuées, nous retrouvons la base, les déplacements 2D, les iris automatiques, les contrôles de l'intensité lumineuse, le contrôle du polariseur, le contrôle de la longueur d'onde, les obturateurs (voir section 3.7, page 54), les supports pour les séparateurs de faisceaux, les supports pour les membranes et beaucoup d'autres pièces.

3.4 Ordinateur

Dans la section 3.2 de la page 18, on peut imaginer toutes les possibilités que peut apporter un stimulateur optique si sophistiqué. Le problème avec un tel stimulateur est qu'il devient très difficile pour un utilisateur d'en contrôler chacun des différents aspects manuellement. Donc, pour diminuer au minimum le temps nécessaire à la réalisation d'une expérience ou pour rendre possibles certaines expériences, un ordinateur est greffé au montage et chacun des mécanismes du montage est automatisé pour que l'ordinateur puisse le contrôler.

Étant donné que l'ordinateur utilise un système d'exploitation qui n'est pas prévu pour fonctionner en temps réel, c'est-à-dire Windows NT, il est nécessaire qu'il y ait des circuits auxiliaires qui complètent certaines tâches. Ces circuits seront directement contrôlés par l'ordinateur. Parmi ces circuits, nous retrouvons le contrôleur de moteurs et le générateur de séquences.

3.5 Contrôleur de moteurs

Le stimulateur lumineux comprend plusieurs mécanismes permettant de modifier les caractéristiques des stimuli lumineux. La majorité de ces mécanismes utilisent des pièces mécaniques qui sont rendues mobiles grâce à l'utilisation de moteurs électriques. L'ordinateur principal doit donc contrôler les différents moteurs électriques pour ajuster les diverses caractéristiques du stimulateur. Or, vu le grand nombre de moteurs et le temps nécessaire pour contrôler un moteur avec le système d'exploitation utilisé, il n'est pas possible que l'ordinateur effectue lui-même le contrôle de tous ces moteurs. La réalisation d'un circuit auxiliaire, c'est-à-dire le contrôleur de moteurs, est donc nécessaire pour effectuer le contrôle de chacun des moteurs.

Dans le présent montage, il faut bien faire la différence entre le rôle du contrôleur de moteurs et le rôle de l'ordinateur. Le rôle de l'ordinateur est de décider, avec l'aide de l'utilisateur, quels sont les déplacements que les différents moteurs effectueront. Le rôle du contrôleur de moteurs est d'effectuer les déplacements proprement dits des moteurs selon les consignes de l'ordinateur et d'aviser l'ordinateur lorsque les différents déplacements sont complétés. Il doit aussi aviser l'ordinateur de la position courante de chacun des moteurs. De plus, le contrôleur peut effectuer quelques opérations de plus haut niveau. La description des différentes possibilités du contrôleur de moteurs est présentée à la section 4.4 de la page 85.

Pour avoir un bon contrôle sur chacun des moteurs, le circuit utilise des HCTL1100 qui sont des circuits intégrés spécialisés dans l'asservissement et le déplacement des moteurs. Le circuit comprend aussi un petit ordinateur permettant une communication entre l'ordinateur principal et les HCTL1100. L'ordinateur du contrôleur de moteurs permet aussi d'effectuer des tâches qui n'ont pas besoin d'être faites par l'ordinateur principal. Le circuit sera aussi muni d'une mémoire avec pile pour conserver la position des moteurs lors de la mise hors tension du circuit. La communication entre l'ordinateur principal et le contrôleur de moteurs se fait par l'intermédiaire d'un port de communication sériel utilisant le protocole RS-232.

3.5.1 Le circuit.

L'organisation interne du contrôleur est sensiblement la même que l'organisation interne d'un ordinateur conventionnel. Il y a la carte maîtresse qui contient un microcontrôleur et les périphériques nécessaires à son fonctionnement. Le contrôleur contient un bus permettant d'accéder à d'autres périphériques plus spécialisés et les périphériques eux-mêmes qui sont des cartes contenant les HCTL1100. La figure 3-4 présente l'organigramme général du contrôleur.

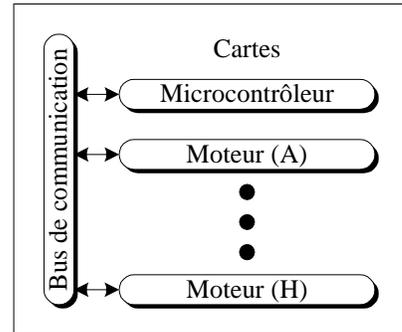


Figure 3-4 : Organigramme du contrôleur de moteurs.

La description de la carte maîtresse est faite à la section 3.5.1.1 et la description des cartes périphériques est faite à la section 3.5.1.2 de la page 33. Puisque le contrôleur de moteurs utilise un microcontrôleur, il doit aussi y avoir un logiciel pour faire fonctionner le tout. Ce logiciel et ses possibilités sont présentés à la section 4.2 de la page 76.

3.5.1.1 La carte maîtresse

Pour que la carte maîtresse puisse se comporter comme un ordinateur, elle doit posséder plus qu'un microcontrôleur. Elle requiert donc de la mémoire pour emmagasiner le logiciel et les données. Elle nécessite aussi des périphériques lui permettant de communiquer

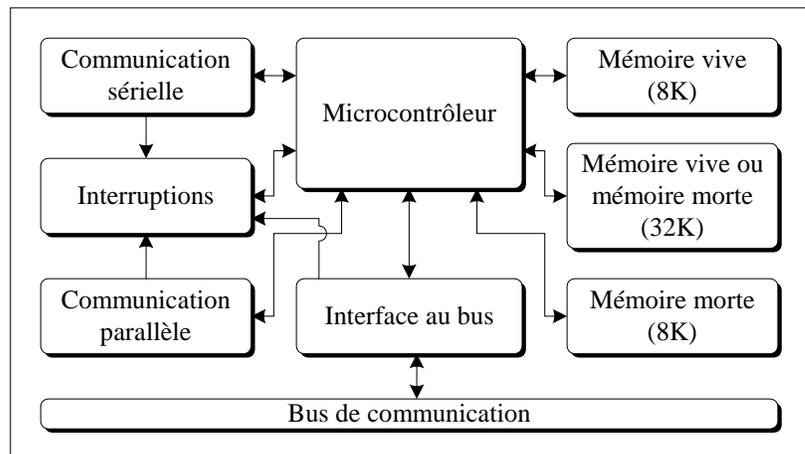


Figure 3-5 : Organigramme de la carte maîtresse.

avec le monde extérieur, c'est-à-dire l'ordinateur central. Enfin, elle a besoin d'un bus de communication lui permettant d'avoir accès à d'autres périphériques que ceux déjà présents

sur la carte. L'organigramme de la figure 3-5 montre bien les interrelations entre les différents modules qui sont décrites plus en détail dans les quatre sections qui suivent.

3.5.1.1.1 Le microcontrôleur

Le microcontrôleur est essentiellement un circuit intégré, soit le 68HC11, qui contient plus qu'un microprocesseur. Parmi les items utiles que nous retrouvons dans le 68HC11, il y a des ports de communication, des compteurs programmables, des interruptions, un chien de garde et une mémoire interne. Le 68HC11 est aussi muni d'une interface lui permettant d'accéder à une mémoire externe au circuit intégré. Cela permet d'augmenter considérablement la quantité de mémoire disponible pour le logiciel et aussi d'ajouter des périphériques externes qui sont associés à des plages de mémoire très précises.

3.5.1.1.2 Les mémoires

Étant donné que le microcontrôleur ne possède pas beaucoup de mémoire interne, il est essentiel de lui en donner davantage. Puisqu'il s'agit d'un contrôleur autonome ne possédant aucune unité de stockage externe, le logiciel doit être inscrit dans une mémoire non volatile. L'utilisation de la mémoire morte est donc indispensable pour emmagasiner le logiciel. Cependant, le logiciel doit avoir accès à des variables qu'il peut modifier lors de son exécution. Donc, il faut aussi une zone de mémoire où le logiciel peut emmagasiner les données. L'utilisation d'une mémoire vive est donc aussi indispensable pour le bon fonctionnement du logiciel. Comme il est montré dans l'organigramme de la figure 3-5 de la page 30, il y a une mémoire morte de 8 kilo-octets pour emmagasiner le logiciel et une mémoire vive de 8 kilo-octets pour emmagasiner les données. En plus de ces deux mémoires, la carte maîtresse possède une autre mémoire de 32 kilo-octets que l'utilisateur peut changer selon les besoins du logiciel. Si un logiciel est petit et a beaucoup de données, l'utilisateur doit installer une mémoire vive de 32 kilo-octets. Par contre, si le programme est plus grand et qu'il utilise peu de données, l'utilisateur doit installer une mémoire morte de 32 kilo-octets.

Il est aussi possible d'utiliser la mémoire vive pour emmagasiner le logiciel, mais cela nécessite l'utilisation d'un périphérique de communication pour transférer le logiciel en mémoire vive à partir d'un ordinateur hôte. La mémoire morte de 8 kilo-octets doit contenir

un logiciel de transfert permettant de transférer le logiciel de l'ordinateur hôte à la mémoire vive. Par la suite, le logiciel est exécuté en mémoire vive.

3.5.1.1.3 Les périphériques de communication

Les périphériques de communication permettent au microcontrôleur d'échanger de l'information avec d'autres ordinateurs ou d'autres types de composants. Une des communications les plus utilisées pour communiquer en informatique et supportées par tous les ordinateurs est la communication sérielle. La carte maîtresse est donc munie d'un 68B50, qui est un circuit intégré faisant la conversion de signaux respectant le protocole RS-232 en nombres binaires. La carte peut changer la configuration du 68B50 pour supporter une grande plage de vitesses de communication et les différentes séquences de codage, c'est-à-dire le nombre de bits d'arrêt, le nombre de bits de données et la parité. Afin que la présente carte soit la plus versatile possible, elle possède aussi un port de communication parallèle, soit le 146823P, qui peut se comporter comme un port parallèle conventionnel ou simplement être utilisé comme ligne d'entrée ou de sortie. Les circuits intégrés 68B50 et 146823P peuvent signaler au microcontrôleur l'arrivée de données en utilisant des lignes d'interruption.

3.5.1.1.4 Le bus de communication

Puisque la présente carte maîtresse est conçue pour commander des moteurs ou des séquenceurs (voir section 3.6, page 38), il est nécessaire qu'elle puisse communiquer avec d'autres cartes spécialisées. L'accès aux autres cartes se fait à travers des accès à différentes plages de mémoire réservée pour chacune des cartes. Le logiciel devra donc connaître l'emplacement des plages de mémoire ainsi que la signification de chacun des octets des différentes plages de mémoire. La communication avec les périphériques se fera en écrivant ou en lisant dans les différentes plages de mémoire. Il y a aussi des lignes d'interruption que les périphériques peuvent utiliser pour aviser le microcontrôleur d'un événement quelconque. En plus, la carte maîtresse met à la disposition des autres cartes d'autres signaux qui ne seront pas présentés dans la présente thèse.

3.5.1.1.5 Discussion

La carte maîtresse est munie d'un petit microcontrôleur assez rapide pour que les demandes de l'ordinateur hôte puissent être effectuées sans délai apparent pour une soixantaine de moteurs. Cependant, dans le cas où l'on demande au microcontrôleur d'effectuer une tâche de longue durée, les communications entre l'ordinateur hôte et le microcontrôleur sont interrompues pendant la réalisation de la tâche. Cela implique que les tâches spécialisées de bas niveau interrompent la communication entre l'ordinateur hôte et le microcontrôleur. Cependant, les commandes envoyées au microcontrôleur pendant l'exécution de l'algorithme sont conservées et seront exécutées quand celui-ci sera terminé. Par contre, le microcontrôleur peut conserver un nombre limité de commandes dans une pile de réception. Si la tâche est trop longue, le circuit se doit de couper la communication en indiquant à l'ordinateur hôte, par l'intermédiaire du contrôle de flux matériel, qu'il ne peut plus recevoir de données. Cela implique que les tâches doivent être assez courtes pour ne pas provoquer un débordement de la pile de réception.

En conclusion, il faut conserver à l'esprit que le microcontrôleur est un petit CPU avec une horloge de 2MHz et que les opérations mathématiques se font avec des registres de 8 bits. Donc, il faut que les tâches soient les plus simples possible et qu'elles ne comportent pas beaucoup de fonctions mathématiques utilisant des nombres en point flottant ou des entiers de 32 bits.

3.5.1.2 La carte moteur

La carte moteur est le périphérique que la carte maîtresse utilise pour contrôler les moteurs. Puisque la superficie de la carte est insuffisante pour que tous les contrôles des moteurs y soient, il doit y avoir plusieurs cartes moteurs. Cela permet aussi de varier le nombre de moteurs dans un contrôleur. Une carte comprend donc six moteurs et le contrôleur peut avoir jusqu'à huit cartes, chacune utilisant une plage de mémoire différente. Chaque carte est munie de trois cavaliers qui lui permettent de sélectionner l'une des huit plages de mémoire disponible. Il est important que les plages de mémoire de chacune des cartes ne se chevauchent pas pour un bon fonctionnement du système. Cependant, vu le nombre de

connecteurs disponible sur le bus de communication, il n'est pas possible d'utiliser plus de six cartes moteurs simultanément, donc trente-six moteurs.

À première vue, contrôler un moteur semble quelque chose de simple; on applique une tension et, quand le moteur est rendu à la bonne position, on arrête d'alimenter le moteur. Cependant, ce n'est pas aussi facile. Bien contrôler un moteur nécessite des algorithmes spécialisés qui sont plus ou moins complexes selon la charge du moteur et les types de déplacement à effectuer. Dans le présent montage, les déplacements à effectuer sont très simples. Les conditions suivantes doivent être respectées: le moteur ne doit pas vibrer pour déstabiliser la préparation, il ne doit en aucun cas dépasser le point d'arrivée et l'erreur entre la position d'arrêt du moteur et la position demandée doit être la plus petite possible.

Pour empêcher le moteur de vibrer ou de tourner trop vite, l'algorithme doit contrôler la vitesse maximale du moteur et éviter les arrêts ou les départs brusques. De plus, la variation des commandes envoyées au moteur ne doit pas être trop saccadée. Pour éviter que le moteur ne dépasse la position d'arrêt, l'algorithme doit diminuer graduellement la vitesse lorsque le moteur approche de la position d'arrivée. Afin de diminuer au minimum l'erreur entre la position d'arrêt et la position visée, l'algorithme filtre la commande envoyée au moteur. L'algorithme contrôlera donc l'accélération, la vitesse maximale et la décélération du moteur. En plus, la commande envoyée au moteur sera filtrée avec un filtre à avance de phase qui possède un pôle, un zéro et un gain. Puisqu'il s'agit d'un système discret, le circuit échantillonnera la position du moteur et changera la commande du moteur selon une fréquence donnée. L'ajustement des paramètres mentionnés si haut est très difficile, car il n'y a pas d'association directe entre un paramètre et un phénomène donnés. Donc, chaque paramètre affecte tous les aspects du contrôle avec plus ou moins d'importance.

Puisqu'il ne faut pas surcharger le microcontrôleur, avec l'asservissement possible de trente-six moteurs, il est nécessaire d'utiliser des circuits intégrés spécialisés qui remplissent assez bien les conditions nécessaires au déplacement d'un moteur. L'utilisation du circuit intégré HCTL1100 est donc un bon choix pour le type d'application visé par le présent contrôleur.

La carte moteur est un périphérique comprenant le nécessaire pour faire le contrôle de six moteurs qui sont numérotés de zéro à cinq. La figure 3-6 présente les interrelations entre les différents modules de la carte. Chacun de ces modules est décrit dans les six sections qui suivent.

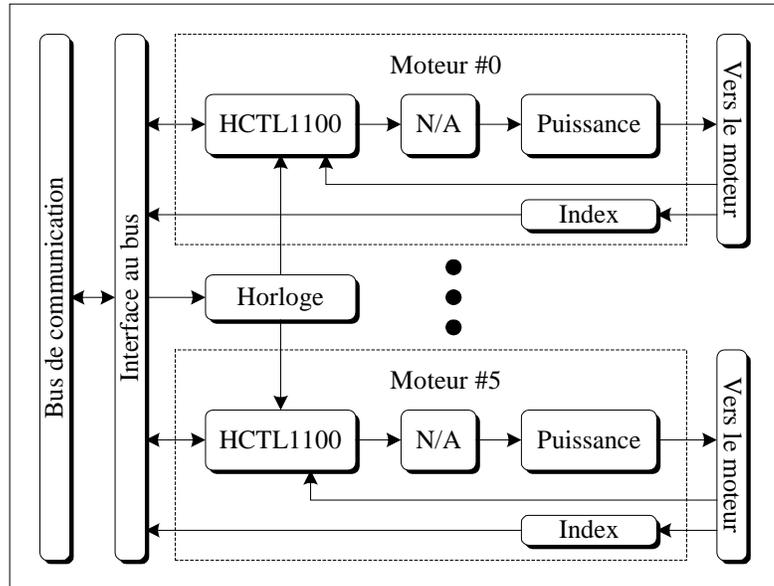


Figure 3-6 : Organigramme de la carte moteur.

3.5.1.2.1 L'interface au bus

Pour permettre au microcontrôleur de la carte maîtresse d'accéder aux différents modules de la présente carte, il doit y avoir un circuit qui permet d'assigner à chacun des modules un espace mémoire dans la plage de mémoire réservée pour la carte. L'interface au bus permet donc au microcontrôleur d'utiliser les HCTL1100, de lire les index et de changer l'horloge de la carte.

3.5.1.2.2 L'horloge

Originellement, le HCTL1100 est muni d'une horloge interne qu'il est possible de modifier pour changer la fréquence d'échantillonnage d'un moteur. Or, certains moteurs fonctionnent mieux lorsque la fréquence d'échantillonnage est plus petite que la valeur minimale qu'il est possible d'obtenir avec le HCTL1100. L'ajout de l'horloge externe permet donc d'obtenir des fréquences d'échantillonnage plus faibles. Par contre, l'ajustement de l'horloge externe affecte tous les HCTL1100 de la carte. De plus, il ne faut pas trop diminuer l'horloge, car cela diminue aussi la fréquence à laquelle le microcontrôleur peut accéder au HCTL1100.

3.5.1.2.3 Le HCTL1100

Comme mentionné précédemment, le HCTL1100 permet de contrôler le positionnement d'un moteur. Il fait lui-même la lecture des encodeurs du moteur et contrôle la tension appliquée sur le moteur. Il contrôle aussi l'accélération, la vitesse maximale et la décélération d'un moteur lors d'un déplacement et filtre la commande envoyée au moteur à l'aide d'un filtre à avance de phase. Le circuit permet aussi de maintenir la position des moteurs. Si un moteur est déplacé manuellement, le HCTL1100 le remet à sa place sans aucune intervention du microcontrôleur. Le rôle du microcontrôleur se limite donc à programmer les différents paramètres de l'accélération, de la vitesse maximale, du filtre à avance de phase et à demander le positionnement des moteurs. Le microcontrôleur peut aussi interroger le HCTL1100 pour connaître la position actuelle d'un moteur. Les différentes possibilités du HCTL1100 sont vues plus en détail dans la section 4.4.2 de la page 88.

3.5.1.2.4 Le convertisseur numérique analogique

Comme le HCTL1100 est un circuit numérique, il retourne la commande du moteur sous forme de nombre binaire de 8 bits où le bit le plus significatif est la polarité de la tension. L'utilisation d'un convertisseur numérique analogique et d'un amplificateur opérationnel est donc nécessaire pour convertir le nombre binaire en tension couvrant une plage de tension allant de -12 V à +12 V.

3.5.1.2.5 L'amplificateur de puissance

Puisque les moteurs nécessitent une certaine quantité de courant que le convertisseur numérique analogique ne peut pas fournir, le circuit possède un amplificateur de puissance permettant d'utiliser des moteurs qui consomment jusqu'à trois ampères. Si un utilisateur désire utiliser des moteurs qui consomment une plus grande quantité de courant, il devra ajouter un amplificateur de puissance extérieur et utiliser la tension à la sortie du convertisseur numérique analogique.

3.5.1.2.6 L'index

Certains moteurs, en plus des encodeurs conventionnels, utilisent un index qui n'est présent qu'une seule fois par tour de moteur. Le circuit permet à la carte maîtresse de faire la lecture de cet index. Son utilisation n'est pas limitée aux index présents sur les encodeurs optiques des moteurs. Il pourrait très bien y avoir un interrupteur qui servirait d'index.

3.5.1.2.7 Discussion

Comme montré précédemment, le HCTL1100 est un circuit permettant de diminuer énormément la tâche du microcontrôleur. Cependant, chaque solution a ses inconvénients. Dans le cas du HCTL1100, il y a certaines limites à l'utilisation des algorithmes implantés et ces derniers ne sont pas toujours appropriés au rôle du moteur. En outre, l'ajout du module horloge est nécessaire pour étendre la plage de fréquence d'échantillonnage possible. En plus, les registres du HCTL1100 sont souvent trop limitatifs. Par exemple, le HCTL1100 utilise des registres de 8 bits pour la configuration des différents déplacements. Des registres de 16 bits auraient permis un meilleur contrôle des moteurs. La solution optimale pour un tel circuit aurait été d'utiliser des microcontrôleurs au lieu des HCTL1100. La réalisation du circuit aurait été différente, la programmation un peu plus complexe, mais une maîtrise totale de l'algorithme d'asservissement aurait été possible et l'implantation de divers algorithmes, selon le rôle joué par le moteur, aurait permis un meilleur contrôle.

3.5.2 Discussion

Tel que vu dans la présente section, le contrôleur de moteur a été conçu dans un but très précis. Les déplacements des moteurs doivent respecter certains critères et chacun d'eux est déplacé indépendamment des autres. Cela n'empêche cependant pas l'ordinateur principal d'associer des moteurs pour réaliser des tâches nécessitant plus d'un moteur. Toutefois, ceux-ci ne pourront jamais être déplacés en synchronisation tout en utilisant les algorithmes du HCTL1100. Les moteurs sont déplacés à vitesse constante et ne peuvent avoir de variation de vitesse lors d'un déplacement.

3.6 Générateur de séquences

Afin de bien synchroniser les différents événements présents lors d'une expérience, l'ordinateur principal doit générer des signaux de contrôle synchronisés entre eux. Ces signaux sont utilisés pour contrôler les obturateurs, les injections de courant, le début des acquisitions et la synchronisation de l'oscilloscope. Cependant, l'ordinateur ne peut pas générer les différents signaux, car leur résolution temporelle est trop grande. La réalisation d'un circuit auxiliaire, c'est-à-dire le générateur de séquences, s'avère donc nécessaire pour effectuer le travail.

Dans le présent montage, il faut bien faire la distinction entre le rôle du générateur de séquences et le rôle de l'ordinateur. Le rôle de l'ordinateur est de décider, avec l'aide de l'utilisateur, quels sont les signaux de contrôle qui doivent être générés. Le rôle du générateur de séquences est de générer les différents signaux de contrôle proprement dits. Il doit aussi aviser l'ordinateur du changement d'état des signaux et de la fin de la génération d'une séquence de signaux.

Pour ce faire, le circuit possède huit lignes de sortie et chacune génère un signal. Chaque ligne génère une séquence de niveau logique haut et bas limité. Par contre, chaque séquence peut être répétée indéfiniment ou un nombre donné de fois. Afin de générer des formes d'ondes complexes, chaque séquence est composée de un à huit événements et chaque événement est composé de deux étapes successives, soit un état logique haut et un état logique bas. L'ordre des étapes ainsi que la durée sont programmables pour chacun des événements. Chacune des étapes peut avoir des durées variant entre 25 μ s et 29,83 heures avec une résolution de 25 μ s.

Chacune des lignes peut être en mode arrêt ou en mode compte. En mode arrêt, la ligne peut être mise à un niveau logique haut ou à niveau logique bas. Quand la ligne passe dans le mode compte, elle attend un signal de départ pour commencer la génération de la séquence. Le signal de départ peut être donné par l'utilisateur ou par une transition montante ou descendante d'une autre ligne du circuit.

La figure 3-7 présente un exemple de configuration pour les cinq premières lignes du circuit. Les chiffres sous les courbes représentent le numéro des événements et les flèches, leur durée.

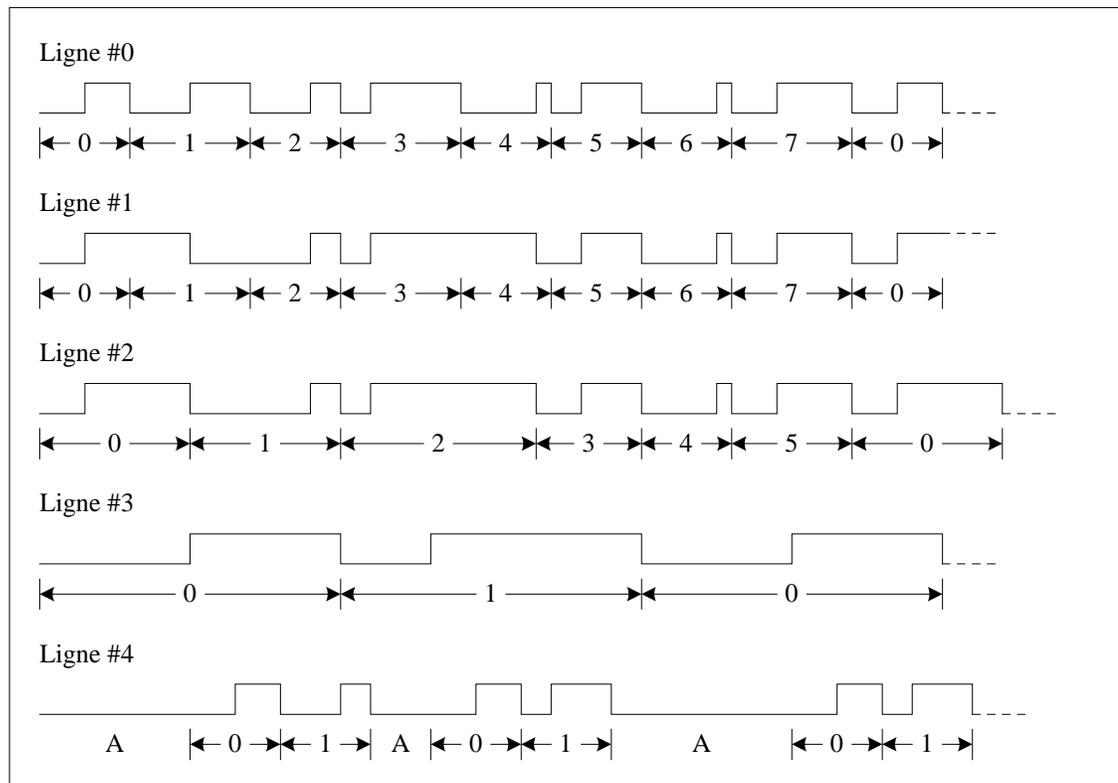


Figure 3-7 : Diagramme temporel des séquences du générateur de séquences.

La lettre A indique que la ligne est en attente d'un signal de départ. Voici donc une description du comportement de chacune des lignes.

La première ligne (ligne #0) présente un cas où les huit événements sont utilisés. Chacun des événements peut avoir une durée différente et possède deux étapes de durée quelconque. Quand les huit événements ont été présentés, le circuit repart avec l'événement zéro. Le tout s'arrête quand le nombre de répétitions de la séquence est atteint ou à la suite d'une demande de l'utilisateur. L'arrêt se fait à la fin d'une séquence, quand le nombre de répétitions est atteint, ou de façon immédiate, s'il s'agit d'une demande de l'utilisateur. Il va de soi que dans le cas où le nombre de répétitions est infini, seule une demande d'arrêt de l'utilisateur peut interrompre le circuit.

La deuxième ligne (ligne #1) montre l'effet produit par l'inversion de la polarité des événements un et quatre de la première ligne alors que la troisième ligne (ligne #2) présente une façon différente de programmer la deuxième ligne. Dans ce dernier cas, il est possible de n'utiliser que six événements. Les événements zéro, un et deux de la deuxième ligne sont remplacés par les événements zéro et un de la troisième ligne et les événements trois, quatre et cinq de la deuxième ligne sont remplacés par les événements deux et trois de la troisième ligne.

Enfin, les deux dernières lignes montrent l'interrelation entre deux lignes quand l'une des deux utilise l'autre comme signal de départ. Ici, la cinquième ligne (ligne #4) a été programmée pour démarrer sur le front montant de la quatrième ligne (ligne #3) et son nombre de répétition est de un. Donc, la cinquième ligne attend de voir une transition positive sur la quatrième ligne avant de démarrer la séquence. Quand la séquence de la cinquième ligne est terminée, la ligne attend une nouvelle transition positive sur l'autre ligne pour redémarrer la séquence. Pour qu'une transition sur la quatrième ligne démarre la séquence de la cinquième ligne, il faut que la cinquième ligne soit en attente. Donc, les transitions présentes sur la quatrième ligne ne sont pas prises en considération si la séquence de la cinquième ligne n'est pas terminée.

De plus, le circuit est muni d'une autre ligne de sortie permettant de synchroniser l'enregistrement des signaux par la carte d'acquisition et la visualisation des signaux par l'oscilloscope avec les différents stimuli. Contrairement aux autres lignes, cette ligne ne génère que des impulsions de courte durée.

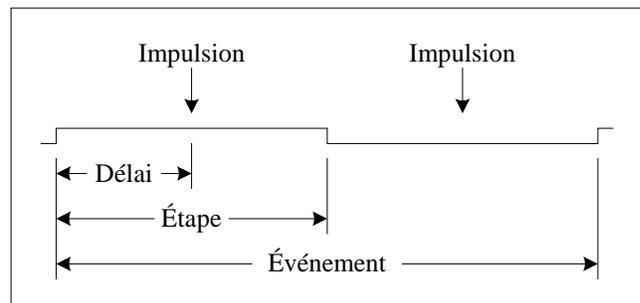


Figure 3-8 : Diagramme temporel d'un événement du générateur de séquences.

L'emplacement de ces impulsions dans un événement est présenté à la figure 3-8. Pour chaque événement, il est possible d'avoir deux impulsions : une première à la première étape et une seconde à la seconde étape. Puisqu'il peut y avoir huit événements dans une séquence,

il peut y avoir seize impulsions par séquence. L'emplacement de l'impulsion dans l'étape ainsi que sa présence sont programmables. Dans une expérience typique, il y a une seule impulsion par séquence et son emplacement est tel qu'elle permet de démarrer la carte d'acquisition 10 ms avant le début des stimuli. Puisqu'il y a des systèmes qui nécessitent des impulsions à logique négative, le circuit est muni de deux sorties. La première est normalement à 0 V et génère des impulsions allant à 5 V. La seconde sortie est normalement à 5 V et génère des impulsions allant à 0 V. La durée de l'impulsion est de 125 ns.

3.6.1 Le circuit du générateur de séquences

Le circuit du générateur de séquences a la même structure générale que le contrôleur de moteurs. Il est muni d'une carte maîtresse et peut avoir jusqu'à deux cartes périphériques de type séquenceur simultanément (voir figure 3-9). La carte maîtresse permet à l'ordinateur principal de communiquer avec chacune des cartes périphériques qui contiennent les circuits électroniques permettant la génération des signaux de commande.

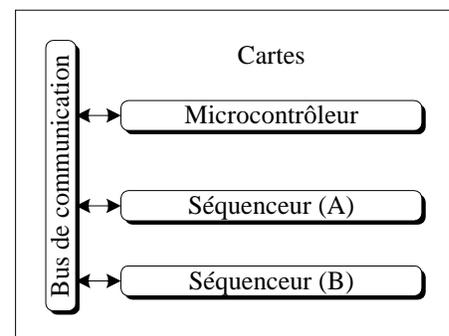


Figure 3-9 : Organigramme du générateur de séquences.

3.6.1.1 La carte maîtresse

La carte maîtresse du générateur de séquences est exactement la même que celle du contrôleur de moteurs présentée à la section 3.5.1.1, page 30. Cependant, le logiciel utilisé n'est pas le même et il est expliqué à la section 4.5 de la page 91.

3.6.1.2 La carte séquenceur

La carte séquenceur est de loin la plus complexe de tout le montage. Une première étude a vite montré qu'il n'était pas approprié de construire un circuit de génération de séquences par ligne car il faudrait un compteur de 32 bits et un comparateur de 32 bits par ligne. Étant donné que les compteurs et les comparateurs de 32 bits ne sont pas facilement disponibles commercialement, il faut construire les compteurs de 32 bits à partir de compteurs de 16 bits et les comparateurs de 32 bits à partir de comparateurs de 8 bits. Puisqu'il y a huit lignes, cela

donne 48 circuits intégrés sans compter les circuits intégrés nécessaires pour contrôler les compteurs et les comparateurs.

Pour éviter de tomber dans un design comprenant beaucoup de circuits intégrés, il a été décidé de construire une machine à état qui utilise deux mémoires de 8 kilo-octets et un compteur de 8 bits. Donc, la machine à état génère successivement les séquences des huit lignes. Pour éviter qu'il y ait un décalage entre les changements des états sur les lignes, le circuit met à jour toutes les lignes simultanément. De cette façon, les séquences de chacune des lignes sont parfaitement synchronisées même si le circuit travaille de façon séquentielle.

Le circuit est en réalité constitué de deux machines à état. La première est utilisée pour générer la longueur de chacune des étapes. La seconde est utilisée pour dicter l'étape suivante quand la première machine à état a fini de générer la longueur de l'étape en cours. La figure 3-10 donne l'organigramme de la carte séquenceur. On retrouve l'explication du rôle de chacun des blocs dans les sections qui suivent.

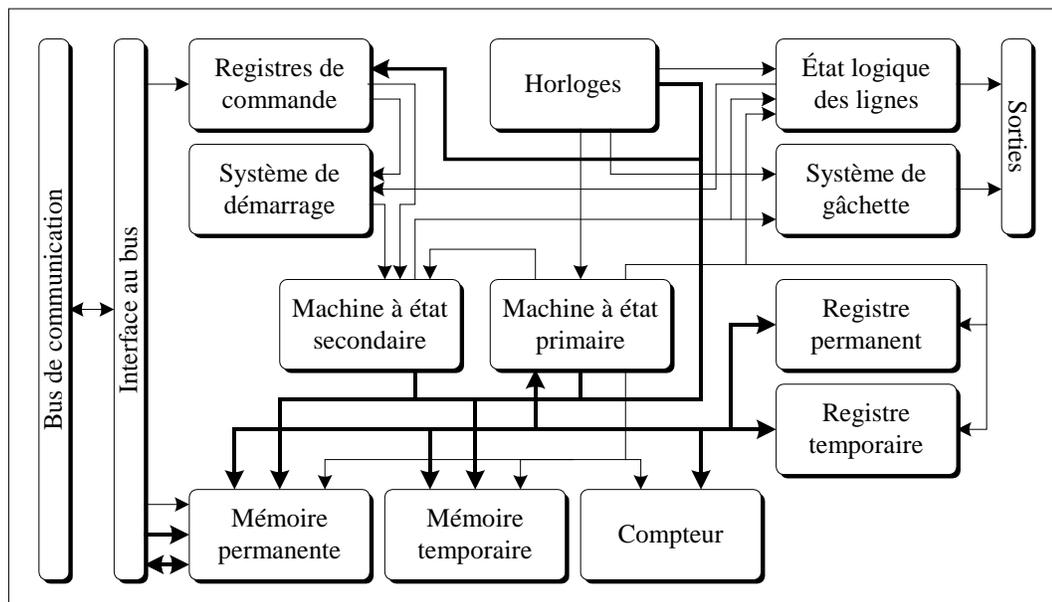


Figure 3-10 : Organigramme de la carte séquenceur.

3.6.1.2.1 L'interface au bus de communication

Pour assurer au microcontrôleur de la carte maîtresse l'accès aux différents modules de la présente carte, il doit y avoir un circuit qui permet de donner à chacun des modules un espace mémoire dans la plage de mémoire réservée pour la carte. L'interface au bus permet donc au microcontrôleur d'accéder à la mémoire permanente et au registre de commande.

3.6.1.2.2 Les horloges

Pour bien synchroniser les différents modules, le circuit requiert deux horloges et un compteur de trois bits. L'horloge la plus rapide est à 8 MHz et donne la durée d'un état de la machine à état primaire. L'horloge la plus lente est à 320 KHz et donne le temps dont la machine à état primaire dispose pour traiter une ligne. Le compteur de trois bits est alimenté par l'horloge de 320 KHz et sélectionne la ligne sous traitement.

Donc, la machine à état primaire dispose de $3,125 \mu\text{s}$ ($1 / 320 \text{ KHz}$) pour faire le décompte d'une étape et du nombre de répétitions d'une ligne, c'est-à-dire le traitement d'une ligne. Puisque l'horloge de la machine à état primaire est de 8 MHz, cela donne un maximum de 25 états ($3,125 \mu\text{s} \times 8 \text{ MHz}$) pour effectuer le travail. Étant donné que le circuit possède huit lignes, la machine à état primaire nécessite $25 \mu\text{s}$ pour effectuer une itération, c'est-à-dire traiter les huit lignes et par la suite faire la mise à jour des sorties. De plus, la machine à état primaire doit se synchroniser avec l'horloge de 320 KHz pour les raisons suivantes :

- Dans le circuit, c'est le compteur de trois bits qui sélectionne la ligne sous traitement et non la machine à état. La machine à état doit donc attendre que le compteur change de ligne et cela se produit sur le front montant de l'horloge de 320 KHz.
- Puisque le nombre d'états nécessaire au traitement d'une ligne n'est pas toujours le même, la synchronisation de la machine à état avec l'horloge de 320 KHz est essentielle pour que le temps alloué à chaque itération soit le même. Dans le cas contraire, la durée d'une itération dépendrait de l'état actuel du décompte et générerait des itérations de durée variable. Cela empêcherait d'avoir une relation linéaire entre les nombres indiquant la durée d'une étape et la durée de l'étape proprement dite. Dans le présent cas, le facteur de

conversion entre le nombre et la durée est de 25 μ s, soit le temps nécessaire au circuit pour procéder au traitement des huit lignes.

Enfin, il faut bien comprendre que le compteur de trois bits fait automatiquement la sélection du registre de commande, de la région de la mémoire permanente et de la région de la mémoire temporaire correspondant à la ligne sous traitement. Les autres modules du circuit refont le même traitement qu'à l'itération précédente sans même savoir le numéro de la ligne sous traitement. Toutefois, le module *système de démarrage* doit tenir compte de la ligne sous traitement, et le module *état logique des lignes* enregistre le résultat de chacune des lignes et transfère le tout dans un registre de sortie à la fin de chaque itération.

3.6.1.2.3 Les registres de commande

Les registres de commande sont en réalité une mémoire à deux ports ayant huit valeurs de quatre bits. Chacune des valeurs de la mémoire est assignée à une ligne du circuit et chacun des bits commande le déroulement de la génération d'une séquence. Cela correspond à avoir huit registres de quatre bits chacun. Le premier bit sert à mettre la ligne en mode arrêt ou en mode compte. Le second bit indique si la ligne est à un niveau logique bas ou à un niveau logique haut quand le mode arrêt est actif. Le troisième bit permet de démarrer la séquence manuellement quand le mode compte est actif. Le dernier bit signale à la machine à état primaire que l'utilisateur a changé le contenu de la mémoire permanente et qu'il doit mettre à jour la mémoire temporaire. Les deux derniers bits doivent être actifs au moins 30 μ s pour que le circuit ait le temps de les prendre en considération.

L'utilisation d'une mémoire à deux ports réduit la complexité du circuit et évite les conflits d'accès à la mémoire entre l'ordinateur et le circuit. L'ordinateur peut aller écrire dans l'un des huit registres en même temps que le circuit lit dans un autre registre. Il peut cependant y avoir un léger conflit si l'ordinateur écrit dans un registre que le circuit a sélectionné, mais le problème se réglera de lui-même à la prochaine itération du séquenceur primaire.

Comme mentionné dans la section 3.6.1.2.2 de la page 43, la sélection du registre est faite automatiquement par le compteur de trois bits pour que le registre sélectionné soit celui de la ligne sous traitement.

3.6.1.2.4 La mémoire permanente

La mémoire permanente contient toutes les configurations du circuit: le nombre de répétitions d'une séquence, le nombre d'événements, la largeur de toutes les étapes, etc. Lorsque l'ordinateur veut changer une configuration du circuit, il doit l'écrire dans cette mémoire. C'est aussi par l'intermédiaire de cette mémoire que la machine à état primaire prend connaissance des séquences à générer. Tout comme le registre de commande, l'utilisation d'une mémoire à deux ports simplifie la réalisation du circuit et évite les conflits d'accès à la mémoire. L'emplacement des différentes valeurs dans la mémoire est présenté au tableau 3-1.

À première vue, la mémoire semble drôlement organisée. Cependant, l'organisation est faite de façon à simplifier au minimum le circuit et, pour ce faire, les adresses de la mémoire sont contrôlées par divers modules. Les adresses A10, A9 et A8

Tableau 3-1 : Organisation des mémoires permanente et temporaire.

A7	A6 à A4	A3 et A2	A1 et A0	Adresse
R	Registre temporaire			01
e	Nombre de répétitions			02 à 03
g	Registre permanent de l'événement #0			00, 04, 08, 0C
i	Registre permanent de l'événement #1			10, 14, 18, 1C
s	Registre permanent de l'événement #2			20, 24, 28, 2C
t	Registre permanent de l'événement #3			30, 34, 38, 3C
r	Registre permanent de l'événement #4			40, 44, 48, 4C
e	Registre permanent de l'événement #5			50, 54, 58, 5C
s	Registre permanent de l'événement #6			60, 64, 68, 6C
	Registre permanent de l'événement #7			70, 74, 78, 7C
		EAV1	R4, R3, R2, R1	80 à 83
	Événement	EAV2	R4, R3, R2, R1	84 à 87
	#0	EAP1	R4, R3, R2, R1	88 à 8B
		EAP2	R4, R3, R2, R1	8C à 8F
		EAV1	R4, R3, R2, R1	90 à 93
	Événement	EAV2	R4, R3, R2, R1	94 à 97
	#1	EAP1	R4, R3, R2, R1	98 à 9B
		EAP2	R4, R3, R2, R1	9C à 9F
		EAV1	R4, R3, R2, R1	A0 à A3
	Événement	EAV2	R4, R3, R2, R1	A4 à A7
	#2	EAP1	R4, R3, R2, R1	A8 à AB
		EAP2	R4, R3, R2, R1	AC à AF
		EAV1	R4, R3, R2, R1	B0 à B3
D	Événement	EAV2	R4, R3, R2, R1	B4 à B7
o	#3	EAP1	R4, R3, R2, R1	B8 à BB
n		EAP2	R4, R3, R2, R1	BC à BF
n		EAV1	R4, R3, R2, R1	C0 à C3
é	Événement	EAV2	R4, R3, R2, R1	C4 à C7
e	#4	EAP1	R4, R3, R2, R1	C8 à CB
s		EAP2	R4, R3, R2, R1	CC à CF
		EAV1	R4, R3, R2, R1	D0 à D3
	Événement	EAV2	R4, R3, R2, R1	D4 à D7
	#5	EAP1	R4, R3, R2, R1	D8 à DB
		EAP2	R4, R3, R2, R1	DC à DF
		EAV1	R4, R3, R2, R1	E0 à E3
	Événement	EAV2	R4, R3, R2, R1	E4 à E7
	#6	EAP1	R4, R3, R2, R1	E8 à EB
		EAP2	R4, R3, R2, R1	EC à EF
		EAV1	R4, R3, R2, R1	F0 à F3
	Événement	EAV2	R4, R3, R2, R1	F4 à F7
	#7	EAP1	R4, R3, R2, R1	F8 à FB
		EAP2	R4, R3, R2, R1	FC à FF

(non présentées dans le tableau) sont contrôlées par le compteur de trois bits et elles permettent de faire la distinction entre les huit plages de mémoire assignées à chacune des lignes. La ligne A7 est contrôlée par la machine à état primaire et sépare les registres et le nombre de répétitions de la longueur des différentes étapes. Les lignes A6, A5 et A4 sont contrôlées par la machine à état primaire quand on désire accéder au registre temporaire ou au nombre de répétitions, mais sont contrôlées par la machine à état secondaire dans les autres cas. Ces dernières font la distinction entre les différents événements. Les lignes A3 et A2 sont contrôlées de la même façon que les lignes A6, A5 et A4 sauf qu'elles font la distinction entre les différentes étapes d'un événement. Finalement, les lignes A1 et A0 sont entièrement contrôlées par la machine à état primaire et elles permettent de faire la distinction entre les quatre octets formant un nombre de 32 bits représentant la longueur d'une étape. On peut aussi remarquer que les registres permanents se retrouvent à quatre endroits différents pour chaque événement. Cela permet d'avoir un registre permanent différent pour chacune des étapes d'un événement.

3.6.1.2.5 La mémoire temporaire

Lorsque la machine à état primaire génère une étape, elle doit utiliser des valeurs inscrites dans la mémoire permanente et les diminuer de un à chaque itération. Puisqu'elle traite successivement les différentes lignes, elle doit conserver l'état actuel des différentes valeurs pour chacune des lignes. Cependant, elle ne peut modifier la mémoire permanente, car elle a besoin de ses valeurs lorsqu'elle recommence la génération d'une séquence. L'utilisation d'une seconde mémoire permet donc de conserver l'état actuel de chacune des valeurs. Pour simplifier au minimum la réalisation du circuit, cette nouvelle mémoire a été divisée exactement de la même façon que la mémoire permanente. La seule différence réside dans le fait que la mémoire permanente contient les valeurs initiales et que la mémoire temporaire contient les valeurs actuelles.

Quand la machine à état primaire a terminé le décompte d'une valeur, elle transfère la valeur suivante à traiter de la mémoire permanente à la mémoire temporaire. Lorsque la machine à état reviendra sur cette valeur, elle n'aura qu'à en effectuer le décompte. Cela implique que la mémoire temporaire doit contenir les valeurs initiales des prochaines valeurs à traiter. Il est

donc nécessaire d'avoir une ligne de commande qui permet de transférer une partie du contenu de la mémoire permanente à la mémoire temporaire lorsque l'ordinateur modifie le contenu de la mémoire permanente. Les valeurs transférées sont celles du registre temporaire, le nombre de répétitions et la longueur de la première étape à être traitée.

3.6.1.2.6 Le registre permanent et le registre temporaire

Lors du traitement d'une ligne, le circuit doit connaître en tout temps l'état de certaines valeurs qui sont dans la mémoire permanente ou dans la mémoire temporaire. Puisque qu'il n'y a que deux nombres de huit bits à connaître, deux registres ont été ajoutés et

Tableau 3-2 : Organisation des registres permanent et temporaire.

Registre	Bits	Contenu
	0 et 1	Étape courante
Temporaire	2 à 4	Événement courant
	5	État de la ligne
	6	Arrêt ou compte
	0	Polarité de l'étape
	1	Impulsion dans la première étape
Permanent	2	Impulsion dans la seconde étape
	3 à 5	Ligne de démarrage
	6	Polarité de la ligne de démarrage
	7	Dernier événement de la séquence

et sont mis à jour chaque fois que la machine à état primaire commence le traitement d'une nouvelle ligne. Le registre permanent est utilisé pour conserver les informations qui ne changeront jamais lors de la génération de la séquence. De son côté, le registre temporaire contient des valeurs qui changeront selon la progression des machines à état. Le contenu exact des deux registres est donné au tableau 3-2.

3.6.1.2.7 Le compteur

Le compteur est utilisé par la machine à état primaire pour diminuer un nombre de huit bits d'une valeur unitaire. Il possède un port d'entrée de huit bits, un port de sortie de huit bits, une ligne d'écriture et une ligne pour activer le port de sortie. Pour l'utiliser, il faut placer un nombre sur le port d'entrée et activer la ligne d'écriture pour que le nombre soit enregistré dans le compteur. Par la suite, il ne reste qu'à activer la sortie pour lire la nouvelle valeur. Les ports d'entrée et de sortie sont connectés ensemble pour que le compteur fonctionne comme une mémoire. Le compteur est aussi muni d'une ligne indiquant si le nombre est égal à 255. Cette ligne est utilisée par la machine à état primaire pour savoir si le décompte du présent nombre est terminé.

3.6.1.2.8 La machine à état primaire

Lorsque le circuit génère la longueur d'une étape, il fait le décompte de cette longueur qui est inscrite dans la mémoire permanente. Pour ce faire, le circuit utilise une machine à état qui lit les différentes longueurs des étapes et qui en diminue la valeur de une unité à chaque itération.

L'organigramme du tableau 3-3 de la page suivante montre bien ce que fait la machine à état primaire. Au tout début, la machine à état attend le début d'une nouvelle ligne. Le cas échéant, elle vérifie si elle doit initialiser la mémoire temporaire. Dans le présent cas, elle transfère l'information minimale requise pour commencer une séquence de la mémoire permanente à la mémoire temporaire. Il s'agit du registre variable, du nombre de répétitions et de la longueur de la première étape du premier événement. La longueur des autres étapes sera transférée au fur et à mesure que la machine à état progresse dans la génération de la séquence.

Par la suite, la machine à état initialise le contenu du registre permanent et du registre temporaire avec les valeurs contenues dans les mémoires permanente et temporaire respectivement. Elle poursuit en vérifiant si la ligne est stable ou en progression. Si la ligne est stable, la machine à état retourne à l'état initial et attend le début d'une nouvelle ligne. Dans le cas contraire, la machine à état procède au décompte de la longueur de l'étape sélectionnée par l'état de la machine à état secondaire. Il ne faut pas confondre le signal de départ du registre de commande avec le signal utilisé ci-dessus. Ce signal est activé si le mode compte du registre de commande est actif et s'il y a un signal de départ émanant du registre de commande ou d'une autre ligne. Il est remis à zéro par la fin du nombre de répétitions ou par la retour au mode arrêt du registre de commande.

Tableau 3-3 : Organigramme de la machine à état primaire.

État présent	Condition à vérifier pour effectuer le branchement		État suivant s'il y a branchement	Adresse courante des mémoires	Action du séquenceur
00	Nouvelle ligne	Non	00		Attend le début d'une nouvelle ligne
	Mise à zéro	Oui	60		
00a	>>>		00		
00b	>>>		00		
01				RT	MT(RT) >> RT
02				RP	MP(RP) >> RP
03					
04	Compte	Non	00a	R1	MT(R1) >> C ; C = C - 1
	Fin décompte	Non	00a	R1	C >> MT(R1)
05				R2	MT(R2) >> C ; C = C - 1
06	Fin décompte	Non	00a	R2	C >> MT(R2)
07				R3	MT(R3) >> C ; C = C - 1
08	Fin décompte	Non	00a	R3	C >> MT(R3)
09				R4	MT(R4) >> C ; C = C - 1
10	Fin décompte	Non	00a	R4	C >> MT(R4)
11					Passé au prochain état du séquenceur secondaire
12				R2	MP(R2) >> MT(R2)
13				R4	MP(R4) >> MT(R4)
14				R1	MP(R1) >> MT(R1)
15				R3	MP(R3) >> MT(R3)
16	Fin séquence	Non	50c		
16	Répétition infinie	Oui	50c	C1	MT(C1) >> C ; C = C - 1
17	Fin décompte	Non	50b	C1	C >> MT(C1)
18				C2	MT(C2) >> C ; C = C - 1
19	Fin décompte	Non	50b	C2	C >> MT(C2)
20				C1	MP(C1) >> MT(C1) ; RT >> MT(RT)
21				C2	MP(C2) >> MT(C2)
22	>>>		00b	RT	RT >> MT(RT)
50a	>>>		50c	RT	RT >> MT(RT)
50b	>>>		50c	RT	RT >> MT(RT)
50c	>>>		00	RT	RT >> MT(RT)
60				RT	MP(RT) >> MT(RT)
61				C2	MP(C2) >> MT(C2)
62				C1	MP(C1) >> MT(C1)
63				R1	MP(R1) >> MT(R1)
64				R2	MP(R2) >> MT(R2)
65				R3	MP(R3) >> MT(R3)
66	>>>		00	R4	MP(R4) >> MT(R4)

Tableau 3-3 : Organigramme de la machine à état primaire (suite).

Légende	
MP	Mémoire permanente
MT	Mémoire temporaire
RP	Registre permanent
RT	Registre temporaire
R1 à R4	R1 à R4 sont respectivement les octets de plus en plus significatifs d'un nombre de 32 bits contenant la longueur de l'état présent.
C1 et C2	C1 est l'octet le moins significatif et C2 l'octet le plus significatif d'un nombre de 16 bits contenant le nombre de répétitions de la séquence.
C	Compteur
>>	Copie un nombre de 8 bits
>>>	Branchement sans condition

Le décompte de la longueur d'une étape se fait en diminuant le nombre le moins significatif (R1) de une unité à l'aide du compteur. Si ce nombre tombe à 255, c'est que la machine à état doit aussi diminuer de une unité la partie la plus significative du nombre qui, dans le présent cas, est R2. Quand le nombre de 32 bits composé des registres R1 à R4 a été diminué d'une unité, la machine à état retourne à l'état de départ.

Lorsque le nombre complet de 32 bits passe en bas de zéro, la machine à état considère que le décompte de l'étape en cours est complété et avise la machine à état secondaire qu'elle peut passer à la prochaine étape et à l'événement suivant, si c'est le cas. Lorsque la machine à état secondaire change d'état, la machine à état primaire voit la nouvelle valeur qu'elle devra traiter à la prochaine itération. Alors, elle en profite pour transférer cette valeur de la mémoire permanente à la mémoire temporaire.

Après, la machine à état primaire vérifie si la génération d'une séquence est terminée. Si c'est le cas, elle diminue d'une unité le nombre de répétitions. Si le nombre de répétitions est infini, la machine à état retourne à l'état de départ. Dans le cas contraire, la machine à état fait le décompte de la valeur en procédant de la même façon que pour le décompte de la longueur d'une étape sauf que dans ce cas, il s'agit d'un nombre de 16 bits au lieu d'un nombre de 32 bits. Lorsque le nombre de 16 bits passe sous zéro, c'est que le nombre de répétitions est atteint, alors la machine à état met la ligne en mode arrêt.

La machine à état termine son travail en effectuant le transfert du nombre de répétitions de la mémoire permanente à la mémoire temporaire, si ce dernier est passé sous zéro.

3.6.1.2.9 La machine à état secondaire

Lorsque la machine à état primaire a terminé le décompte d'une étape, le circuit doit posséder une autre machine à état pour connaître l'étape suivante et le prochain événement, s'il s'agit de la dernière étape d'un événement. La machine à état secondaire est composée de deux petites machines à état. La première indique quelle sera l'étape suivante à l'intérieur d'un événement et la seconde indique le prochain événement à l'intérieur d'une séquence.

L'organigramme de la première des deux machines à état est donné à la figure 3-11 et doit générer l'événement présenté à la figure 3-8 de la page 40. Dans l'événement à générer, il y a deux étapes et un délai. Puisque la comparaison du délai avec le décompte actuel de l'étape ne facilite pas la réalisation de la machine à état primaire, il est plus approprié de séparer chaque étape en deux nouvelles étapes. Lorsque la machine à état termine l'étape EAV1 (étape avant gâchette 1), elle vérifie si elle doit générer une impulsion de synchronisation. Si c'est le cas, elle passe à l'étape EAP1 (étape après gâchette 1) sans que la polarité de la ligne change. Par contre, si elle ne doit pas générer une impulsion de synchronisation, elle passera directement à l'étape EVA2 (étape avant gâchette 2) et la polarité de la ligne changera. Les deux signaux de contrôle G1 (gâchette 1) et G2 (gâchette 2) indiquent la génération d'une impulsion et sont accessibles à l'aide du registre permanent.

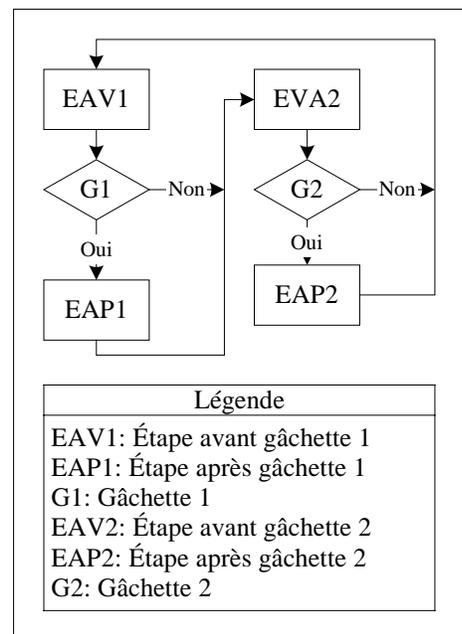


Figure 3-11 : Organigramme de la génération des étapes de la machine à état secondaire.

La seconde machine à état, présentée à la figure 3-12 de la page suivante, est aussi très simple et possède seulement deux signaux de contrôle. Le premier signal indique si la machine à état doit passer dans le prochain événement (PE). Ce signal est actif quand la machine à état

responsable des étapes retourne à la première étape. Le second signal indique s'il s'agit du dernier événement (DE) de la séquence. L'état de ce signal est codé dans le registre permanent.

L'horloge de la machine à état secondaire est asynchrone et n'a lieu que lorsque la machine à état primaire termine le décompte d'une étape. Puisque l'horloge des deux sous-machines à état est la même, la machine à état responsable des événements possède une ligne PE qui lui indique quand elle peut changer d'événement (EVx).

3.6.1.2.10 Le système de démarrage

Comme il a déjà été mentionné à la section 3.6.1.2 de la page 41, il est possible de démarrer une séquence manuellement, par l'intermédiaire du registre de commande, ou automatiquement, par l'intermédiaire d'une autre ligne. Le présent module permet donc de générer le signal de départ nécessaire à la machine à état primaire. Lorsque le signal de départ est manuel, le signal est simplement transmis à la machine à état. Par contre, dans le cas d'un signal de départ automatique, le circuit observe les différentes lignes et génère un signal de départ, si la bonne transition a lieu sur la bonne ligne.

Pour savoir s'il y a eu une transition sur une ligne, le circuit conserve l'état actuel et l'état précédent de chacune des lignes dans deux registres de huit bits, où chaque bit correspond à une ligne. Les

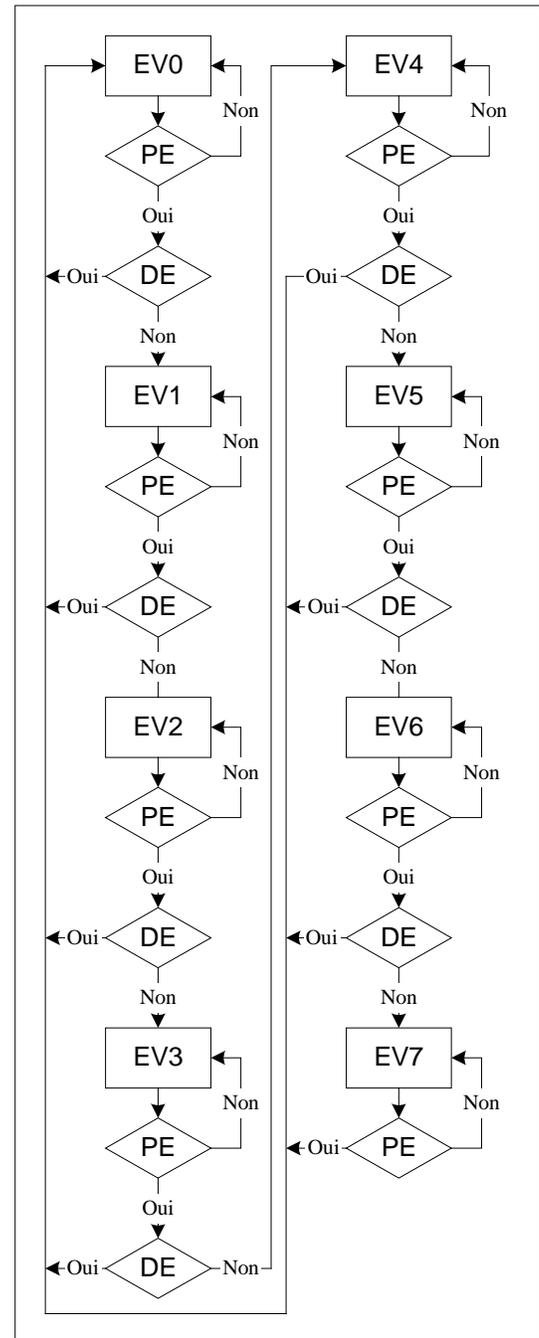


Figure 3-12 : Organigramme de la génération des événements de la machine à état secondaire.

deux registres sont mis à jour chaque fois que la machine à état primaire finit de traiter les huit lignes et qu'elle recommence avec la première ligne. Le numéro de la ligne et le sens de la transition que doit respecter le circuit sont accessibles à partir du registre permanent. Le registre permanent sélectionnera un des huit bits pour chacun des deux registres. Si les deux bits de chacun des registres représentent une transition comme indiquée dans le registre permanent, le signal de départ est donné à la machine à état primaire. Notez que la machine à état n'utilise pas directement ce signal mais qu'elle vérifie son état au moment opportun (voir section 3.6.1.2.8, page 48.).

3.6.1.2.11 Le système de gâchette

Le système de gâchette est un très petit module qui génère les impulsions sur les deux lignes de synchronisation du circuit. Le module regarde donc l'étape de la machine à état secondaire et la transition de la machine à état passant d'une étape EAVX à une étape EAPX (voir figure 3-11, page 51). Le circuit génère alors une impulsion de 125 ns. Tout comme pour les autres modules, l'impulsion est générée quand la machine à état primaire a fini de traiter la dernière ligne.

3.6.1.2.12 L'état logique des lignes

Le présent module génère les signaux de chacune des lignes en fonction de l'étape présente dans la machine à état secondaire et de la polarité de cette étape. Il utilise un registre à décalage de huit bits pour enregistrer l'état de chacune des lignes au fur et à mesure que la machine à état primaire traite une ligne. Quand les huit lignes ont été traitées, le contenu du registre à décalage est transféré dans le registre de l'état actuel abordé à la section 3.6.1.2.10 de la page 52. La sortie du présent registre est la sortie du générateur de séquences.

3.6.2 Discussion

La réalisation du circuit du générateur de séquences nécessite une trentaine de circuits intégrés et une dizaine de circuits intégrés programmables (PAL). Peu après la réalisation du circuit, le laboratoire s'est muni d'une nouvelle technologie de circuits qui peuvent contenir beaucoup plus de fonctions, c'est-à-dire les FPGA. Cette technologie aurait permis de

diminuer le nombre de circuits intégrés, mais l'utilisation des mémoires et d'un circuit qui travaille de façon séquentielle n'aurait pu être évitée, car les FPGA contiennent un nombre insuffisant de compteurs et de comparateurs de 32 bits. Cependant, l'approche utilisée aurait pu être sensiblement différente et la programmation des séquences plus simple.

Même si le circuit fonctionne très bien, il comporte un inconvénient qui vient du fait que le circuit change l'état d'une ligne à la fin d'une étape et non au début d'une étape. Cela implique que si une personne donne une polarité positive à la première étape du premier événement, la ligne restera à un niveau logique haut à la fin de la séquence et avant le début de la génération d'une séquence. Pour avoir un niveau logique bas après et avant la génération d'une séquence, il faut absolument que la polarité de la première étape du premier événement soit négative.

3.7 Obturateur

Comme mentionné précédemment, il est nécessaire de contrôler la durée des stimuli lumineux. Selon le type de lampe utilisé, il n'est pas toujours possible d'avoir des temps d'ouverture et de fermeture très courts. Dans le cas où l'on utilise une lampe halogène, le temps de fermeture de la lampe est très long. De plus, après son ouverture, il faut attendre environ une minute avant que l'intensité lumineuse se stabilise. Donc, il n'est pas possible d'utiliser l'alimentation de la lampe pour changer l'état du stimulus lumineux. L'utilisation d'un obturateur mécanique est alors nécessaire pour empêcher la lumière de passer. De plus, un obturateur mécanique, contrairement à d'autres types d'obturateurs, permet de ne pas diminuer l'intensité lumineuse et de ne pas modifier les caractéristiques telles que la longueur d'onde ou l'angle de polarisation.

Afin de diminuer le temps de réalisation du montage, nous avons sérieusement regardé la possibilité d'utiliser des obturateurs commerciaux comme ceux de Uniblitz Vincent Associates qui sont de haute performance et ont des vitesses semblables à ce que nous recherchions. Ils ont même un nouveau modèle qui est deux fois plus rapide que notre design. Le principal problème réside dans les dimensions physiques des obturateurs. Le diamètre des plus petits obturateurs est au minimum de 7.5 cm au lieu des 4 cm requis dans notre montage.

Comme il y a un jeu de 1 cm autour des trajets, cela limite grandement la grosseur des obturateurs. De plus, la façon de fixer l'obturateur est incompatible avec le système de tiges utilisées dans le montage. Ces tiges, qui passent dans les quatre coins des différents composants optiques, permettent d'ajuster et de positionner facilement les pièces. Il n'est pas possible de percer les obturateurs commerciaux pour laisser passer les tiges qui sont essentielles dans notre montage. Finalement, les obturateurs doivent être positionnés entre un objectif et une fibre optique ce qui donne un jeu mécanique d'environ 1 mm. Puisque les obturateurs commerciaux ont une épaisseur d'environ 2.5 cm, il est alors impossible de satisfaire à cette dernière exigence.

Dans les sections qui suivent, on fait la description d'un obturateur mécanique, de l'électronique nécessaire à son bon fonctionnement et on donne une mesure des performances obtenues.

3.7.1 Le design mécanique

L'obturateur a été conçu pour avoir un temps d'ouverture et de fermeture assez rapide dans le but de générer des stimuli lumineux d'une durée de 2 ms. Dans les constructions standard, l'utilisation d'un mécanisme avec un électroaimant permet de fermer l'obturateur et un ressort est utilisé pour ouvrir à nouveau l'obturateur lorsque l'alimentation de l'électroaimant est coupée. L'inconvénient d'une telle méthode est que lors de la fermeture de l'obturateur, l'électroaimant doit combattre le ressort pour fermer l'obturateur, ce qui diminue grandement le temps de fermeture de l'obturateur. De plus, un tel système a des temps d'ouverture et de fermeture différents.

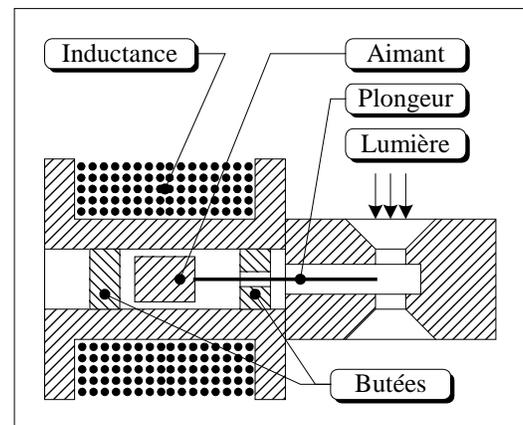


Figure 3-13 : Plan de l'obturateur.

L'approche utilisée ici consiste à placer un aimant de très bonne qualité (R28E, 0.235" DIA., 0.157" THK. de la compagnie Dexter Sylvania, Ohio.) au centre d'une inductance et de fixer le plongeur, qui sert à bloquer le faisceau de lumière, après l'aimant. Lorsque l'inductance est

alimentée par un courant dans une direction donnée, l'aimant se place au centre de l'inductance et laisse ainsi passer la lumière. Quand on alimente l'inductance avec un courant inverse, l'aimant sort de l'inductance, ce qui amène le plongeur à bloquer le passage de la lumière. En plaçant des butées aux bons endroits, on réussit à contrôler la course du plongeur. La figure 3-13 de la page précédente donne un schéma représentant les principales structures d'un tel obturateur.

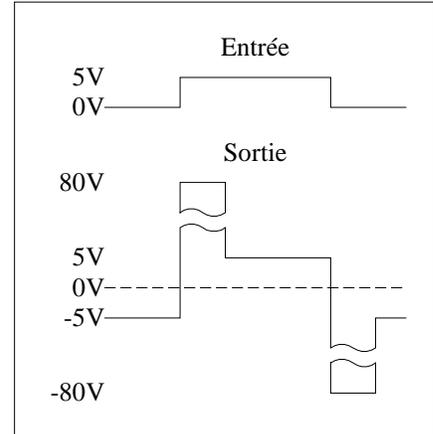


Figure 3-14 : Diagramme temporel du signal d'entrée et du signal de sortie du circuit de contrôle de l'obturateur.

3.7.2 Le design électrique

Afin d'avoir des temps de réponse rapides, il faut alimenter l'obturateur de telle sorte que le plongeur bouge rapidement. Pour ce faire, l'injection d'un fort courant dans l'inductance est nécessaire. Cependant, l'utilisation d'un fort courant a tendance à faire chauffer le filage de l'inductance. Pour éviter que le filage ne devienne trop chaud, il faut limiter la durée d'injection du fort courant au minimum et ne produire ce courant que durant les ouvertures et les fermetures de l'obturateur. Donc, l'injection d'un fort courant positif est utilisée pour ouvrir l'obturateur et un fort courant négatif, pour le fermer. Pour éviter que l'obturateur change de position entre les ouvertures et les fermetures, on produit aussi un faible courant positif pour le maintenir ouvert et un faible courant négatif pour le maintenir fermé.

La figure 3-14 donne un exemple de la tension générée par le circuit en fonction de la tension de commande. Le fort courant est produit lorsque la tension est de ± 80 V et le faible courant quand la tension est de ± 5 V.

Un schéma bloc du circuit réalisé pour contrôler l'obturateur est présenté à la figure 3-15. Le circuit est divisé en deux

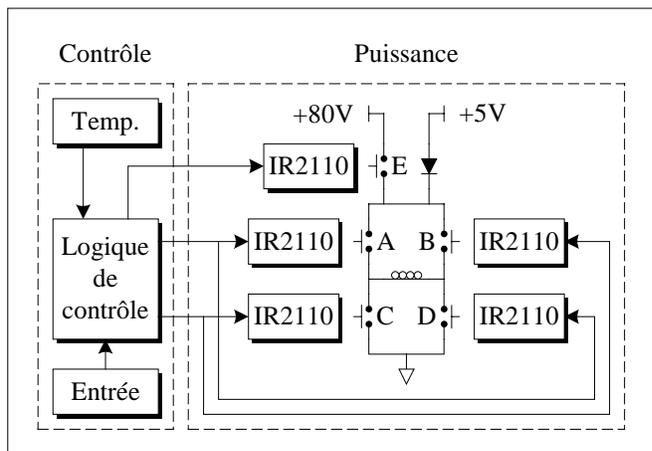


Figure 3-15 : Organigramme du circuit de contrôle de l'obturateur.

sous-circuits délimités par les cadres pointillés. Le circuit de contrôle est consacré à la génération des signaux de commande permettant de changer la direction du courant et le niveau de tension à appliquer à l'inductance en fonction du signal d'entrée. Le circuit de puissance génère les tensions requises et fixe la direction du courant sur l'inductance en accord avec les signaux de commande.

3.7.2.1 Le circuit de contrôle

Afin de bien contrôler l'ouverture et la fermeture de l'obturateur, le circuit se doit de générer la forme d'onde présentée à la figure 3-14 de la page précédente. Pour ce faire, le circuit génère une série de trois signaux. Un de ces signaux est utilisé pour changer la valeur de la tension et les deux autres signaux permettent de changer la direction du courant, c'est-à-dire la polarité de la tension.

Le tableau 3-4 montre la relation entre la combinaison des deux signaux contrôlant la direction du courant et l'incidence de cette combinaison sur le courant circulant dans l'inductance. Il est donc possible de sélectionner le sens du courant ou de ne pas imposer

Tableau 3-4 : Interprétation des signaux de commande du courant dans le circuit de contrôle de l'obturateur.

Signal A	Signal B	Effet
Inactif	Inactif	Aucun courant
Inactif	Actif	Courant direct
Actif	Inactif	Courant inverse
Actif	Actif	Destruction du circuit de puissance

de courant sur l'inductance. Par contre, il y a une combinaison très dangereuse qui détruit le circuit de puissance. Le circuit de contrôle est fait de telle sorte que cette dernière combinaison ne peut pas être générée. Lors de la génération de la forme d'onde, il est nécessaire de laisser un temps mort d'environ 1 μ s entre chaque inversion de courant. Cela évite que les transistors de puissance conduisent simultanément lors de leur transition. Enfin, la direction du courant est directement contrôlée par l'entrée, à l'exception du court délai entre chaque transition.

Le troisième signal est utilisé pour fixer le niveau de la tension au borne de l'inductance. Il est donc possible d'avoir une tension de 80 V et une tension de 5 V. La tension de 80 V est appliquée pendant une durée de 5 ms après chaque transition. Le reste du temps, c'est la

tension de 5 V qui est appliquée. Cela correspond aussi à contrôler la quantité de courant circulant dans l'inductance.

Afin de protéger l'obturateur, le circuit est muni d'un système de limitation en température. Cela empêche l'obturateur de surchauffer quand la fréquence d'activation de l'obturateur est trop grande. Lorsque la température maximale est atteinte, l'entrée du circuit est déconnectée et le demeure jusqu'au moment où la température redescend à un niveau acceptable.

Finalement, le circuit utilise un optocoupleur pour découpler le signal d'entrée. Cela permet de protéger les instruments de commande contre le bruit présent dans le circuit de puissance. Le bruit est causé par les changements rapides de tension et de courant du circuit de puissance. À titre d'exemple, le circuit de puissance passe d'une tension de 80 V à 0 V avec un courant qui passe de 8 A à 0 A dans un laps de temps de 250 ns.

3.7.2.2 Le circuit de puissance

Le rôle du circuit de puissance est d'utiliser les trois signaux générés par le circuit de contrôle pour alimenter correctement l'inductance. Pour ce faire, le circuit est muni de cinq interrupteurs et d'une diode de puissance rapide. Les interrupteurs sont des transistors de puissance à effet de champ. Le circuit est aussi muni de circuits intégrés spécialisés, soit les IR2110, pour que les signaux de commande soient convertis en niveaux de tension et de courant nécessaires à l'opération des transistors (voir figure 3-15, page 56).

Pour changer la direction du courant, le circuit doit utiliser quatre interrupteurs branchés deux à deux et formant un pont. Lorsque les interrupteurs A et D sont fermés et que les interrupteurs B et C sont ouverts, le courant circule de droite à gauche dans l'inductance. Si les interrupteurs B et C sont fermés et que les interrupteurs A et D sont ouverts, le courant circule de gauche à droite dans l'inductance. Si les quatre interrupteurs sont ouverts, il n'y a aucun courant qui circule dans l'inductance. Par contre, si les quatre interrupteurs sont fermés, il y a destruction d'un ou de plusieurs interrupteurs.

Le contrôle du niveau de tension est assuré par l'interrupteur E. Lorsque l'interrupteur est fermé, la tension sur le pont est de 80 V et la diode ne conduit pas, car la tension de la cathode, qui est de 80 V, est supérieure à la tension de l'anode, qui est de 5 V. Dans le cas où l'interrupteur est ouvert, la tension chute jusqu'au moment où elle devient inférieure à 5 V et que la diode commence à conduire, empêchant ainsi la tension de descendre en bas de 4,3 V, qui est égale à la tension de la source moins la tension de chute de la diode. Évidemment, les interrupteurs A, B, C et D devront être dans un état permettant au courant de circuler dans l'inductance.

3.7.3 Discussion

Afin de bien visualiser le comportement de l'obturateur, la figure 3-16 de la page suivante présente trois séries de résultats. La première série (A) permet de voir l'ensemble des signaux utilisés pour générer le déplacement du plongeur. La deuxième série (B) présente une vue agrandie de la première transition de la série (A), et la dernière série (C) montre la durée minimale que l'obturateur peut atteindre sans diminuer l'intensité lumineuse.

Dans la première série, il y a d'abord le signal de commande qui est présenté à l'entrée du circuit de contrôle. Par la suite, il y a les deux signaux de commande du courant puis le signal de commande du niveau de tension. L'avant-dernier signal donne la tension appliquée aux bornes de l'inductance, et le dernier signal représente le profil lumineux vu par une photodiode.

La seconde série montre bien l'utilité du délai qui a été ajouté pour éviter que les quatre transistors conduisent simultanément et qu'il y ait destruction du circuit. Par contre, dans le cas où tous les transistors sont ouverts, on voit que la tension de sortie monte quand même et converge vers 80 volts avant que la commande de tension n'ait été activée. Nous pouvons aussi remarquer qu'il y a beaucoup de bruit qui est généré durant la transition.

La troisième série possède seulement deux signaux, soit l'entrée du circuit de commande et le profil lumineux vu par une photodiode. Cette dernière série montre bien le délai de 2 ms entre

la commande et la réponse lumineuse et la durée minimale de 1,5 ms qu'il est possible d'atteindre.

Les résultats de la figure 3-16 montrent bien que le circuit rencontre les objectifs fixés. Cependant, il faut bien prendre conscience de ses limites. La première limite vient du fait que l'obturateur ne peut laisser passer que des faisceaux lumineux plus petits que 2200 μm de diamètre. Il est à noter que, dans les résultats présentés ci-dessus, le faisceau lumineux avait un diamètre de 1000 μm . L'augmentation du diamètre du faisceau lumineux aura comme

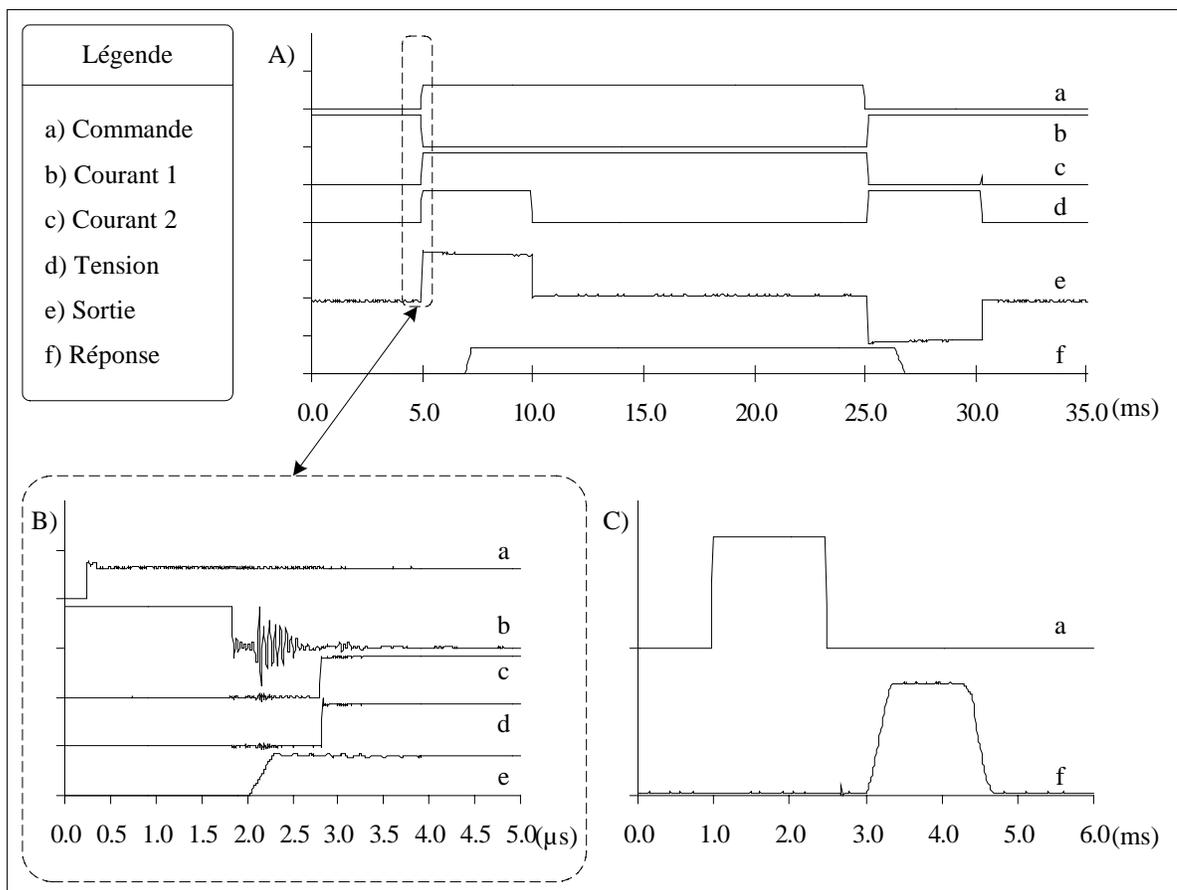


Figure 3-16 : Diagramme temporel des signaux nécessaires au contrôle de l'obturateur.

conséquence d'augmenter la durée du temps de montée et du temps de descente. Cela indique donc une augmentation de la largeur minimale de l'impulsion de lumière. La seconde limite vient du fait qu'une forte injection de courant dans l'inductance fait monter la température. Puisque le fort courant n'est présent que lors des transitions, il est important de limiter le nombre de transitions par seconde. Habituellement, le nombre de transitions maximal par

seconde est de l'ordre de 10 transitions. Si une application nécessitait un système plus rapide, l'utilisateur devrait ajouter un système de refroidissement sur les obturateurs pour augmenter le nombre de transitions par seconde. Toutefois, d'autres limites doivent être considérées comme le refroidissement des transistors qui impose une limite de l'ordre de 500 Hz et la vitesse des optocoupleurs qui limite la vitesse à 10 KHz. Par contre, la limite absolue est donnée par la réponse mécanique de l'obturateur qui est d'environ 5 ms, donc 200 Hz. Cette limite est plus grande que la durée minimale d'une impulsion lumineuse, car l'obturateur commence à bouger avant que la lumière commence à augmenter et finit de bouger après que la lumière ait fini de diminuer.

3.8 Caméra

Pour permettre à l'ordinateur et à l'utilisateur d'avoir un bon contrôle sur l'expérience, il est essentiel d'avoir une image de la préparation avec le moins de bruit possible. La faible luminosité des sources lumineuses amène l'obligation d'utiliser une caméra très sensible et d'intégrer l'image pendant un temps donné. Nous verrons plus loin qu'il est important que l'intensité lumineuse envoyée sur la préparation ne soit pas trop importante. Les chapitres IV et V présentent successivement des explications sur les logiciels utilisant la caméra et le rôle de la caméra lors d'une expérience.

Il s'agit donc d'utiliser une caméra commerciale minimisant le bruit et permettant d'intégrer l'image sur une période de temps qu'il est possible de modifier. De plus, la caméra devra avoir une interface permettant de transférer les images à l'ordinateur. La caméra STAR I de Photometrics Ltd. et la carte AT-GPIB/TNT de National Instrument Corp. nous a permis de rencontrer les présents objectifs.

L'utilisation de d'autres types de caméras est aussi possible. Il suffit de modifier les logiciels de bas niveau pour que la communication entre les logiciels de haut niveau et la caméra soit assurée, voir section 4.9 de la page 113.

3.9 Acquisition et oscilloscope

Dans le type d'expérience visé par le présent montage, il faut mesurer l'activité électrique des cellules de la préparation et informer l'ordinateur de l'état du montage. Pour ce faire, l'utilisation d'une carte d'acquisition est indispensable. La carte d'acquisition doit être munie d'un certain nombre de canaux permettant d'enregistrer des signaux électriques et de les stocker dans la mémoire de l'ordinateur.

L'ordinateur peut, grâce à la carte d'acquisition, enregistrer l'activité électrique des cellules tout en connaissant l'état des stimuli lumineux et des injections de courant. Il peut aussi mesurer l'intensité lumineuse de chacun des canaux grâce au radiomètre et connaître la température de la préparation grâce au contrôleur de température.

Une ligne de synchronisation externe doit être présente sur la carte d'acquisition pour permettre une bonne synchronisation entre les enregistrements, les stimuli lumineux et les injections de courant.

Puisqu'il y a beaucoup de cartes d'acquisition commerciales, il est inutile d'en construire une. L'utilisation de la carte d'acquisition DT2801-A de Data Translation Inc. permet d'effectuer toutes les opérations requises par le présent montage. La carte possède aussi des ports numériques et des sorties analogiques qui pourraient être utilisés dans des versions futures des logiciels.

Tout comme pour la caméra, il est possible d'utiliser une autre carte d'acquisition respectant les contraintes mentionnées ci-haut. Dans un tel cas, une version différente des logiciels de bas niveau est nécessaire. Plus concrètement, un deuxième stimulateur est monté avec la carte DAS-16 de Keithley Metrabyte Corp. La section 4.7.3 de la page 101 montre les techniques utilisées pour interchanger les cartes d'acquisition.

3.10 Amplificateur

L'enregistrement dans des cellules d'un animal nécessite l'utilisation de microélectrodes et d'un amplificateur très spécialisé. L'amplificateur doit aussi permettre l'injection d'un

courant dans une cellule. L'injection de courant est nécessaire et permet plusieurs opérations indispensables au bon déroulement d'une expérience telle, entre autres, la mesure de l'impédance d'entrée d'une cellule. Puisqu'il existe de tels amplificateurs sur le marché, il n'est pas approprié d'en construire un. L'amplificateur utilisé dans le présent montage est l'amplificateur Axoclamp-2A de Axon Instrument Inc.

Il est toutefois possible d'utiliser un amplificateur d'une autre compagnie, mais l'amplificateur doit être muni d'entrées spéciales pour que l'ordinateur, par l'intermédiaire du générateur de séquences, puisse contrôler les injections de courant. L'ajustement de la valeur du courant se fait par l'amplificateur.

3.11 Compensation des microélectrodes

Lors de certaines expériences, il est nécessaire de prendre des mesures avec deux microélectrodes simultanément. Cela permet de bien vérifier l'interaction électrique entre deux cellules différentes. Toutefois, un problème majeur apparaît et empêche de bien isoler les deux signaux. Ce problème est dû au couplage électrique entre les deux microélectrodes. Une variation de tension sur une microélectrode a tendance à faire changer la tension de la seconde microélectrode. Pour résoudre ce problème, il est nécessaire d'ajouter un circuit de compensation permettant de minimiser l'effet du couplage capacitif entre les deux microélectrodes.

Les détails du circuit conçu par le D^r Simon Gagné (Gagné et al. 1997a) ne sont pas présentés ici mais un aperçu de son fonctionnement est illustré à la figure 3-17. Les deux premières courbes donnent la forme d'onde obtenue

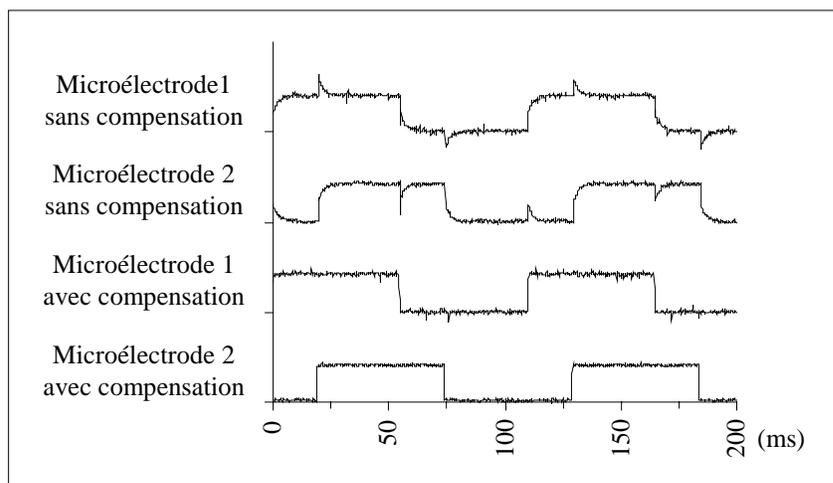


Figure 3-17 : Diagramme temporel des signaux électriques du circuit de compensation.

quand il n'y a pas de circuit de compensation. La montée des signaux est arrondie et, lors de la transition sur un des signaux, le couplage capacitif présent entre les deux microélectrodes engendre une pointe de tension sur le second signal. On voit que l'ajout du circuit de compensation élimine les pointes de tension et l'arrondissement des signaux. Cependant, il existe plusieurs circuits de compensation qui peuvent éliminer les arrondissements, mais aucun ne peut éliminer les pointes de tension.

3.12 Synchronisation

Comme vu dans la section 3.3.1 de la page 22, il est possible d'ouvrir et de fermer les lumières des différents trajets lumineux. Les séquences lumineuses sont générées par le générateur de séquences et dictées par l'ordinateur. Afin d'établir la

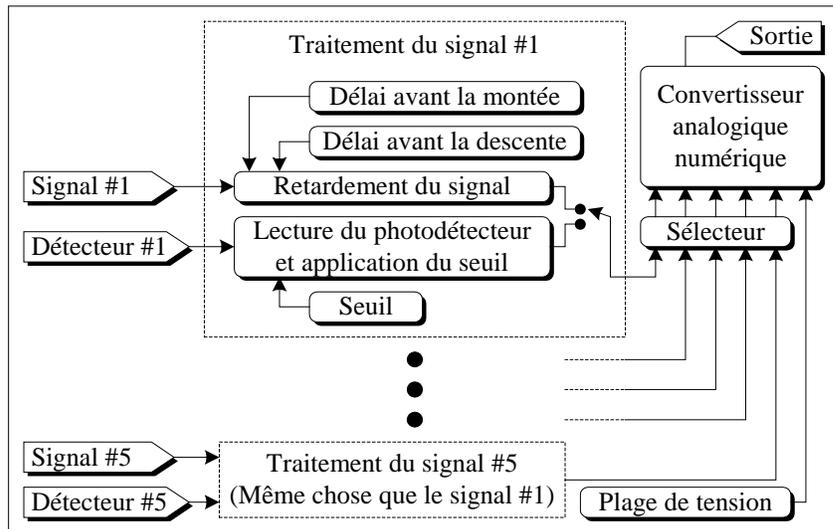


Figure 3-18 : Organigramme du circuit de synchronisation.

différentes réponses et les stimuli lumineux, il est nécessaire de faire simultanément l'acquisition des différentes réponses et de l'état des stimuli lumineux. Puisque les obturateurs engendrent un délai entre la commande et les stimuli lumineux (voir figure 3-16-C, page 60), il n'est pas possible d'utiliser les commandes pour représenter l'état des stimuli lumineux. On a donc mis au point un circuit qui utilise deux différentes techniques pour corriger le problème tout en diminuant le nombre de canaux utilisés. L'organigramme du circuit est présenté à la figure 3-18.

3.12.1 Synchronisation avec l'ajout d'un délai

La solution la plus simple consiste à retarder le signal de commande pour que ce dernier corresponde aux stimuli lumineux. Cela est possible, car le délai engendré par les obturateurs est toujours le même et il peut être ajusté. L'ajustement du délai des obturateurs se fait grâce à

deux vis permettant de régler la course du plongeur de l'obturateur. La première vis permet d'ajuster le délai avant la montée et la seconde vis, le délai avant la descente (voir section 3.7, page 54). Évidemment, il est impossible de n'avoir aucun délai, et le délai minimum est de 2 ms. Le circuit génère donc un second signal de commande pour lequel il est possible d'ajuster le délai avant la montée et le délai avant la descente. En ajustant le délai avant la montée du circuit, comme celui de l'obturateur, et en faisant de même pour le délai avant la descente, le circuit génère une nouvelle commande qui représente fidèlement l'état du stimulus lumineux.

Cette solution n'est pas parfaite, car deux petits problèmes persistent toujours.

- Lorsque l'obturateur est en opération, il se déplace rapidement et cause beaucoup de vibrations. Après une longue durée d'utilisation, environ un mois, il arrive que les vis d'ajustement se déplacent et qu'il faille ajuster à nouveau les délais.
- Pour compter le temps nécessaire à la génération des deux délais, le circuit utilise deux monostables dont l'un se déclenche sur le front montant et l'autre, sur le front descendant de la commande. Parfois, quand les signaux sont trop rapides ou qu'il y a du bruit, le circuit détecte une seule des deux transitions. Cela a comme conséquence que la commande générée est incorrecte jusqu'à la prochaine commande. Il s'agit heureusement d'un problème mineur car, dans le déroulement normal d'une expérience, cela ne se produit pas.

3.12.2 Synchronisation avec un photodétecteur

La seconde solution consiste à insérer un photodétecteur dans chacun des trajets lumineux pour surveiller l'état des stimuli lumineux. Si l'intensité lumineuse est supérieure à un seuil donné, le circuit génère un niveau logique haut et, dans le cas contraire, il génère un niveau logique bas. Le circuit est aussi muni d'une hystérésis pour empêcher la sortie d'osciller quand l'intensité lumineuse est proche du seuil. Il est possible d'ajuster le seuil manuellement par l'intermédiaire d'un potentiomètre.

Toutefois, pour optimiser la quantité de lumière générée par chacune des sources lumineuses, les filtres d'intensité doivent être placés devant les obturateurs. Puisque les photodétecteurs sont placés derrière les obturateurs (voir la figure 3-3, page 21), il y a variation de l'intensité lumineuse sur les photodétecteurs. Étant donné que le signal fourni par le circuit est basé sur un seuil, le délai et la largeur du signal généré changent selon l'intensité. Dans le cas où l'intensité est trop faible, les photodétecteurs ne détectent pas la lumière. La figure 3-19 montre bien le comportement du signal fourni par le circuit en fonction du seuil et de l'intensité lumineuse.

Pour qu'un seuil soit bien ajusté, il faut que la durée de la réponse soit la même que celle à demi amplitude du signal. Le seuil est donc ajusté à environ 50% de l'amplitude maximale de la réponse du photodétecteur. Dans ce dernier cas, la largeur du signal généré est la même que la largeur du signal de commande. Sur la figure, cela correspond au cas *Seuil 2, intensité 1*. Lorsque le seuil est ajusté à *Seuil 1*, la largeur de la réponse est trop petite et le signal est en retard et, lorsque le seuil est ajusté à *Seuil 3*, la largeur de la réponse est trop longue et en avance. Par contre, quand l'intensité lumineuse est plus faible et que le seuil reste le même, le signal généré est modifié et se comporte comme le cas *Seuil 1*. La comparaison des courbes

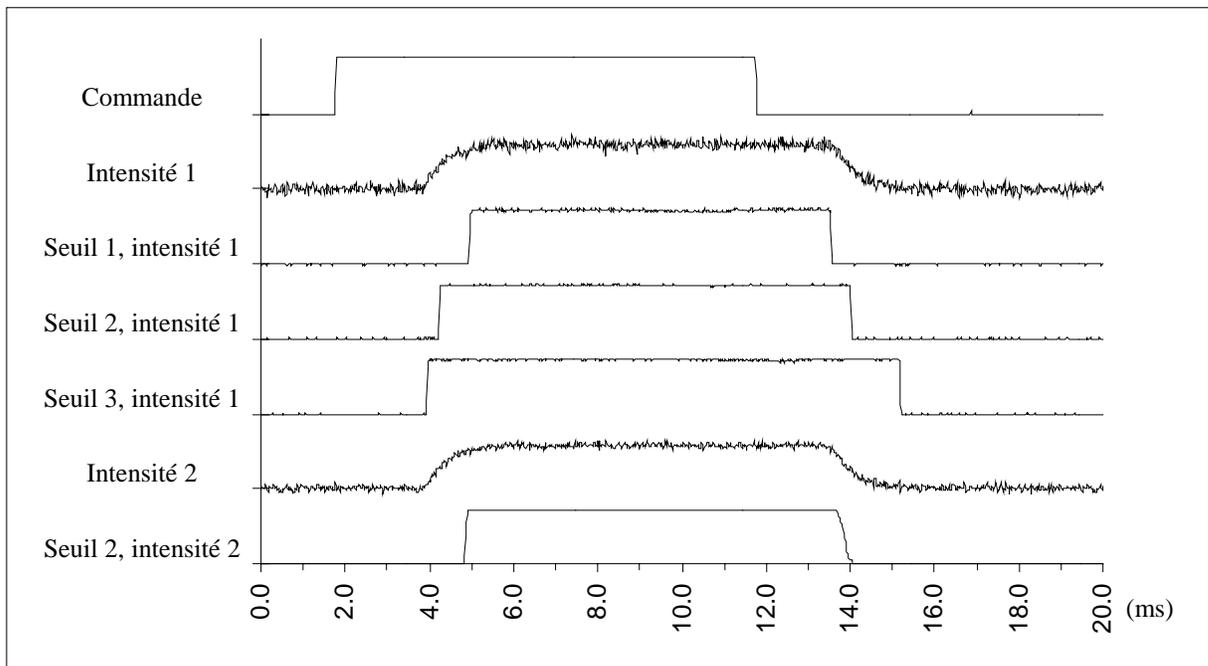


Figure 3-19 : Diagramme temporel des signaux du circuit de synchronisation.

Seuil 2, intensité 1 et *Seuil 2, intensité 2* montre bien la différence entre les deux cas. Dans le présent cas, une variation de 1% de l'intensité lumineuse change considérablement le délai et la largeur de la réponse.

3.12.3 Multiplexage des signaux

Sur une carte d'acquisition, la fréquence d'échantillonnage multipliée par le nombre de canaux est égale à une constante, c'est-à-dire que la fréquence maximale permise diminue avec l'augmentation du nombre de canaux. Il en est de même pour le nombre maximal de points qu'un canal peut enregistrer. Dans un montage typique, il y a deux canaux pour la mesure des réponses des microélectrodes, un canal pour la mesure de l'injection de courant et quatre canaux pour la mesure de l'état des stimuli lumineux. Puisque les signaux représentant l'état des stimuli lumineux sont binaires, il est possible d'utiliser un multiplexeur pour regrouper cette information dans un seul canal. Dans l'exemple précédent, cela réduit l'utilisation du nombre de canaux à quatre au lieu de sept.

Le multiplexage des signaux binaires est fait à l'aide d'un convertisseur numérique analogique. Chaque signal correspond à un bit du convertisseur. L'ordinateur fait le démultiplexage du signal reçu en effectuant une conversion analogique numérique pour retrouver les signaux d'origine. Évidemment, l'ordinateur et le circuit

Signaux numériques # 3 2 1		Signal analogique Tension (volts)
<u>1 1 1</u>	4	<u>3,75</u>
<u>1 1 0</u>	3,5	<u>3,25</u>
<u>1 0 1</u>	3	<u>2,75</u>
<u>1 0 0</u>	2,5	<u>2,25</u>
<u>0 1 1</u>	2	<u>1,75</u>
<u>0 1 0</u>	1,5	<u>1,25</u>
<u>0 0 1</u>	1	<u>0,75</u>
<u>0 0 0</u>	0,5	<u>0,25</u>
	0	

devront être programmés selon le nombre de canaux et la plage de tension à utiliser. La figure 3-20 montre la correspondance entre la tension

Figure 3-20 : Conversion des signaux binaires en un signal analogique utilisé par le circuit de synchronisation.

analogique et les niveaux logiques de chacun des canaux pour un circuit avec trois canaux et une plage de tension de quatre volts. La colonne de gauche donne l'état des signaux numériques et la colonne de droite, les tensions correspondantes. La colonne du centre donne les plages de tension utilisées lors du démultiplexage pour retrouver les signaux d'origine.

3.13 Radiomètre

Lors d'une expérience impliquant plusieurs trajets lumineux, il est important que l'intensité lumineuse de chacun des trajets soit la même. Cela permet la comparaison des réponses obtenues sans tenir compte du trajet utilisé. Pour ce faire, une photodiode a été placée de façon à mesurer l'intensité lumineuse générée par tous les trajets. Afin d'obtenir une mesure qui représente le plus possible la valeur de ce que la préparation reçoit, l'emplacement de la photodiode dans le montage doit être tel que la quantité et la forme de la lumière reçue par la photodiode soient les mêmes que celles reçues par la préparation, voir figure 3-3 à la page 21.

Pour convertir la tension mesurée par la photodiode en mesure d'illuminance, soit des W/cm^2 , l'utilisation d'un radiomètre, qui a préalablement été calibré pour la photodiode utilisée, est nécessaire. Le radiomètre donne directement une lecture en W/cm^2 . Par contre, on ne peut considérer cette lecture comme étant correcte pour les deux raisons suivantes:

- La lecture en W/cm^2 n'est valable que pour une longueur d'onde donnée, c'est-à-dire la longueur d'onde utilisée lors de la calibration du radiomètre et de la photodiode. Or, lors des expériences, la majorité du temps, la lumière est blanche ou a une longueur d'onde bien précise qui n'est pas nécessairement la longueur d'onde utilisée pour la calibration du radiomètre.
- La surface utilisée par le faisceau lumineux ne couvre pas toute la surface de la photodiode. Cependant, une correction est possible. La valeur réelle est obtenue en multipliant la valeur lue par le rapport de la surface de la photodiode sur la surface du faisceau lumineux. Pour une photodiode avec une surface de 1 cm^2 et un faisceau avec un diamètre d , la valeur réelle est donnée par :

$$\text{Valeur Réelle} = \text{Valeur Lue} \times 10^{-4} / ((d/2)^2 \times \pi)$$

Même si la valeur lue ne reflète pas la quantité réelle de lumière que reçoit la préparation, cela ne fausse en rien la comparaison entre les trajets. Deux trajets donnant la même illuminance

sur le radiomètre ont forcément la même illuminance sur la préparation, si ces deux trajets ont le même diamètre.

Présentement, la comparaison entre les trajets est faite manuellement par l'utilisateur. Cependant, rien n'empêcherait l'ordinateur d'établir cette comparaison. Cela est possible, car l'intensité lumineuse de chacun des trajets est contrôlée par l'ordinateur et elle peut être lue par l'ordinateur à l'aide de la carte d'acquisition.

Cependant, un problème persiste dans l'utilisation du radiomètre. Lors de certaines expériences, il faut travailler avec de faibles intensités lumineuses. Or, la photodiode n'est pas assez sensible pour ces faibles intensités. Pour remédier à ce problème, l'utilisation d'un compteur de photons ou d'une photodiode plus sensible serait nécessaire.

3.14 Lecture et contrôle de la température

Dans certaines expériences, il est intéressant de contrôler la température de la préparation. Le contrôle s'effectue par le contrôleur de température TC-202 de Medical Systems Corp. qui permet de fixer et de maintenir la température de la préparation. En apportant des modifications au contrôleur de température, il serait possible de contrôler la température de la préparation par l'intermédiaire de l'ordinateur.

Dans le présent montage, tout a été prévu pour installer le bain de contrôle de température sur le montage pour que la lecture de la température soit faite par l'ordinateur. Lors de l'enregistrement des données par l'application *Acquisition et traitement des courbes*, il est possible de faire aussi la lecture de la température. Ainsi, les données sont conservées avec la température de la préparation.

Avec la modification du contrôleur de température et quelques ajouts à l'application *Contrôle des expériences*, il serait possible d'effectuer des expériences où l'on ferait varier la température pour vérifier l'influence de la température sur le phénomène étudié.

3.15 Conclusion

Au tout début de ce chapitre, nous avons présenté le montage comme étant un montage qui devait être polyvalent et qui devait permettre l'automatisation des expériences. D'une part, la polyvalence a été atteinte en divisant au maximum le montage en modules différents. Le découpage en modules permet de faire des modifications à un module sans en affecter les autres. Cette approche s'est avérée très efficace car, en ce moment, il y a trois montages possédant chacun des points communs et des points très différents. Cependant, il ne faut pas oublier que le logiciel (voir chapitre IV) qui permet de contrôler le tout doit lui aussi être polyvalent.

D'autre part, l'automatisation a été atteinte en construisant des modules qui peuvent être contrôlés par un ordinateur. Cette approche déplace la responsabilité de l'automatisation vers les logiciels, et la partie matérielle du montage ne forme que les outils nécessaires au logiciel pour effectuer son automatisation. Par contre, la partie matérielle du stimulateur effectue des tâches complexes et des tâches qui demanderaient beaucoup de temps CPU à l'ordinateur. Cela libère l'ordinateur pour ainsi lui laisser les prises de décision.

Finalement, le montage réalisé n'est pas parfait. Il suffit de se référer aux discussions de chacune des sections du présent chapitre pour connaître les avantages, inconvénients et problèmes de chacun des modules. Cependant, les problèmes qui demeurent sont évitables facilement et de seconde importance. Il est certain que régler ces problèmes rendrait l'utilisation du montage plus agréable et offrirait d'autres possibilités d'expérimentation à l'utilisateur. Malgré tous ses petits problèmes, le montage fonctionne très bien et permet d'effectuer des expériences très complexes qui seraient autrement pratiquement impossibles à réaliser. Les figure 3-21, figure 3-22 et figure 3-23 de la page suivante présentent le montage construit dans le cadre des présentes études doctorales.

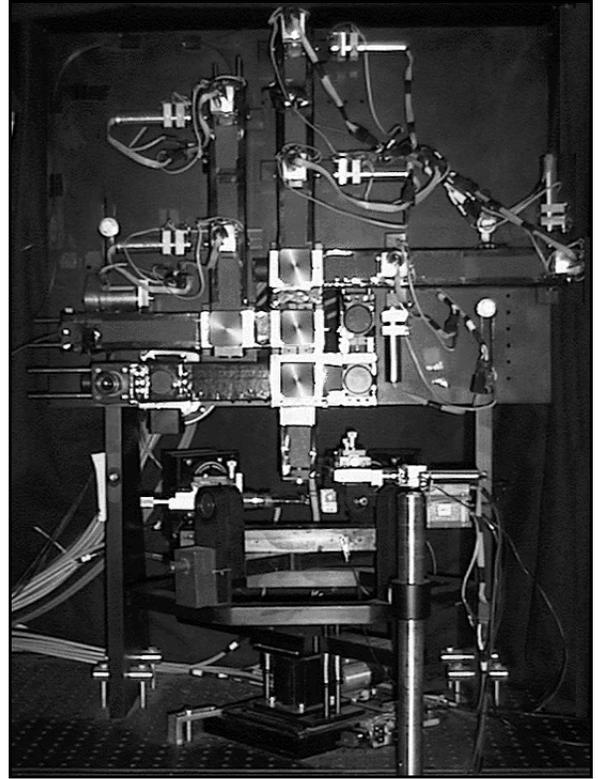


Figure 3-21 : Photographie du montage (I).

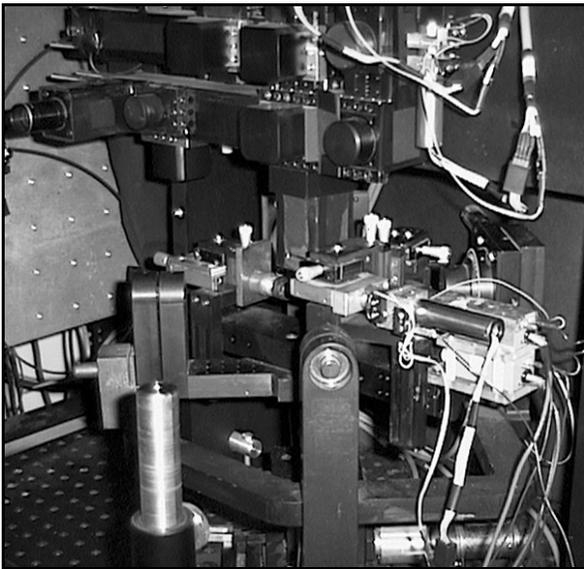


Figure 3-22 : Photographie du montage (II).

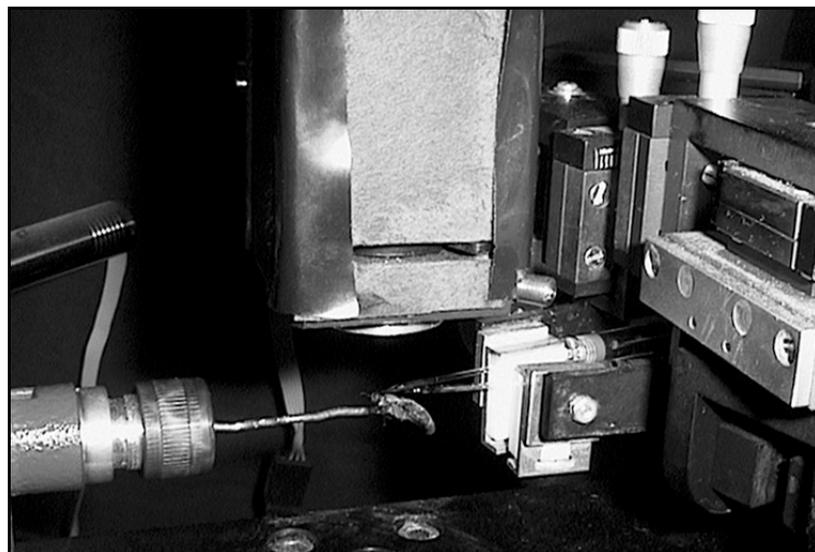


Figure 3-23 : Photographie du montage (III).

CHAPITRE IV

LE MONTAGE (PARTIE LOGICIELLE)

4.1 Introduction

Lors de la réalisation de la partie matérielle du montage, la responsabilité de gérer les expériences a été confiée à l'ordinateur principal. Cependant, la réalisation d'une seule application monstre pour contrôler tous les modules du montage n'est pas appropriée. Il est beaucoup plus intéressant de construire plusieurs applications spécialisées où chacune contrôle un module du montage et une application centrale communiquant avec toutes les autres applications pour diriger le déroulement de l'expérience. Cette structure poursuit la philosophie visée au chapitre III qui consiste à construire un montage modulaire autant pour la partie logicielle que pour la partie matérielle.

La réalisation de cette structure devient possible grâce au système d'exploitation utilisé, soit Windows NT. Ce système d'exploitation permet à diverses applications de communiquer entre elles à l'aide d'un protocole de communication que l'on nomme *Dynamique Data Exchange (DDE)*. Ce protocole est d'une simplicité remarquable et très rapide. Il fonctionne selon le principe d'une relation client-serveur entre deux applications, où le serveur est celui qui satisfait les requêtes du ou des clients. La puissance de ce protocole est telle qu'une application peut être un serveur de plusieurs autres applications de type client et en même

temps le client de plusieurs autres serveurs. Le type d'information circulant entre les applications peut être aussi bien sous forme binaire que sous forme texte. Un client peut aussi bien transmettre un bloc de données au serveur qu'en recevoir un. Il y a aussi un mode très intéressant qui permet au serveur d'aviser le client chaque fois qu'une nouvelle information est disponible sur le serveur. L'annexe A fait un survol des notions de la communication DDE.

Pour que la structure soit vraiment modulaire, il faut que les applications soient les plus indépendantes possible du matériel utilisé, c'est-à-dire de la carte d'acquisition, de la carte vidéo, etc. Pour ce faire, on génère des « librairies » pour chaque type de carte où les points d'entrée des « librairies » sont identiques pour un type de carte donné. Par exemple, dans le cas des cartes d'acquisition, il y a deux « librairies », une pour la carte d'acquisition DT2801-A et une autre pour la carte d'acquisition DAS 16. Cependant, les fonctions des « librairies » visibles par l'application sont toutes les mêmes, et l'application ne voit pas la différence entre les deux « librairies ». Alors, le choix de la carte d'acquisition se fait en changeant de « librairie ». Mais, ce qui est le plus intéressant, c'est qu'il est possible de générer des « librairies » qui sont liées à l'exécutable lors de l'exécution et non à la compilation, ce que l'on nomme les *Dynamic Link Library (DLL)*. Il suffit donc de placer la « librairie » de la carte d'acquisition que l'on désire utiliser sur le disque dur de l'ordinateur. Il est aussi possible de mettre les deux « librairies » sur le disque dur et de faire la sélection de l'une ou l'autre par l'intermédiaire de l'application les utilisant. Cependant, cela complique un peu l'appel des fonctions à l'intérieur de l'application.

Puisqu'il s'agit d'un système d'exploitation préventif qui protège très bien les applications en leur évitant de sortir de la plage de mémoire qu'elles ont préalablement réservée, il n'est pas possible d'accéder directement au matériel de l'ordinateur à partir d'une application standard. Il faut donc construire des pilotes de bas niveau pour chacune des cartes disponibles. Ces pilotes sont accessibles par l'intermédiaire des DLL ou des applications en utilisant les fonctions systèmes prévues à cet effet. Ces pilotes sont construits de telle façon qu'il ne peut pas y avoir deux pilotes utilisant les mêmes interruptions, les mêmes DMA ou le même espace mémoire évitant ainsi toute forme de conflit entre les pilotes. La réalisation de ces

pilotes doit être méticuleusement faite pour éviter à l'ordinateur de perdre le contrôle. Pour comprendre pourquoi un pilote peut faire perdre le contrôle à l'ordinateur, il faut savoir que sous Windows NT, l'espace mémoire théorique est segmenté en deux parties. Pour un système possédant des pointeurs de 32 bits, il y a une première partie de 2 giga-octets réservée par le système d'exploitation (espace système) et une seconde partie de 2 giga-octets réservée pour les applications (espace utilisateur). Cette division est faite dans le but d'optimiser la vitesse du système d'exploitation tout en diminuant le nombre de vérifications faites dans l'espace système. Puisque les pilotes sont considérés comme faisant partie intégrante du système d'exploitation et qu'ils sont exécutés dans l'espace système, un pointeur fou peut causer la perte de contrôle de l'ordinateur. Par contre, les applications et les DLL sont exécutés dans l'espace utilisateur et sont sévèrement surveillés, ils ne peuvent donc pas causer de dommages au système d'exploitation et aux autres applications.

La structure modulaire est donc subdivisée en six applications, trois ou quatre DLL et trois pilotes (voir figure 4-1, page suivante). Il y a une application principale (Contrôle des expériences) qui contrôle le déroulement des expériences en communiquant à l'aide de DDE avec les quatre applications de contrôle. La première de ces quatre applications (Contrôleur de moteurs) permet de contrôler tous les moteurs du montage. La seconde application (Générateur de séquences) sert à synchroniser les divers événements du montage dans le temps. La troisième application (Acquisition et traitement des courbes) est responsable de l'acquisition, du traitement et du stockage des courbes acquises par l'intermédiaire de la carte d'acquisition. La dernière application de contrôle (Acquisition et traitement des images) est responsable de l'acquisition, du traitement et du stockage des images obtenues par la caméra. La dernière des six applications est utilisée par l'application de contrôle (Contrôleur de moteurs) pour communiquer, à l'aide du port série COM3, avec le contrôleur de moteurs proprement dit. Cette dernière application est aussi utilisée par l'application de contrôle (Générateur de séquences) pour communiquer, à l'aide du port série COM4, avec le générateur de séquences proprement dit. Puisque les deux contrôleurs sont de petits ordinateurs, ils nécessitent l'utilisation de logiciels pour communiquer et faire leur travail.

De leur côté, les DLL et les pilotes sont utilisés comme intermédiaires pour communiquer avec le matériel. Cependant, ce ne sont pas tous les DLL et pilotes qui sont utilisés. Les deux interrupteurs de la figure 4-1 représentent les deux choix possibles pour deux des applications de contrôle.

La figure 4-1 permet de voir rapidement les interrelations entre les différents modules de l'ordinateur. La partie supérieure (Logiciel (applications)) comprend toutes les applications réalisées, et la communication entre chacune des applications est faite en utilisant le DDE. La

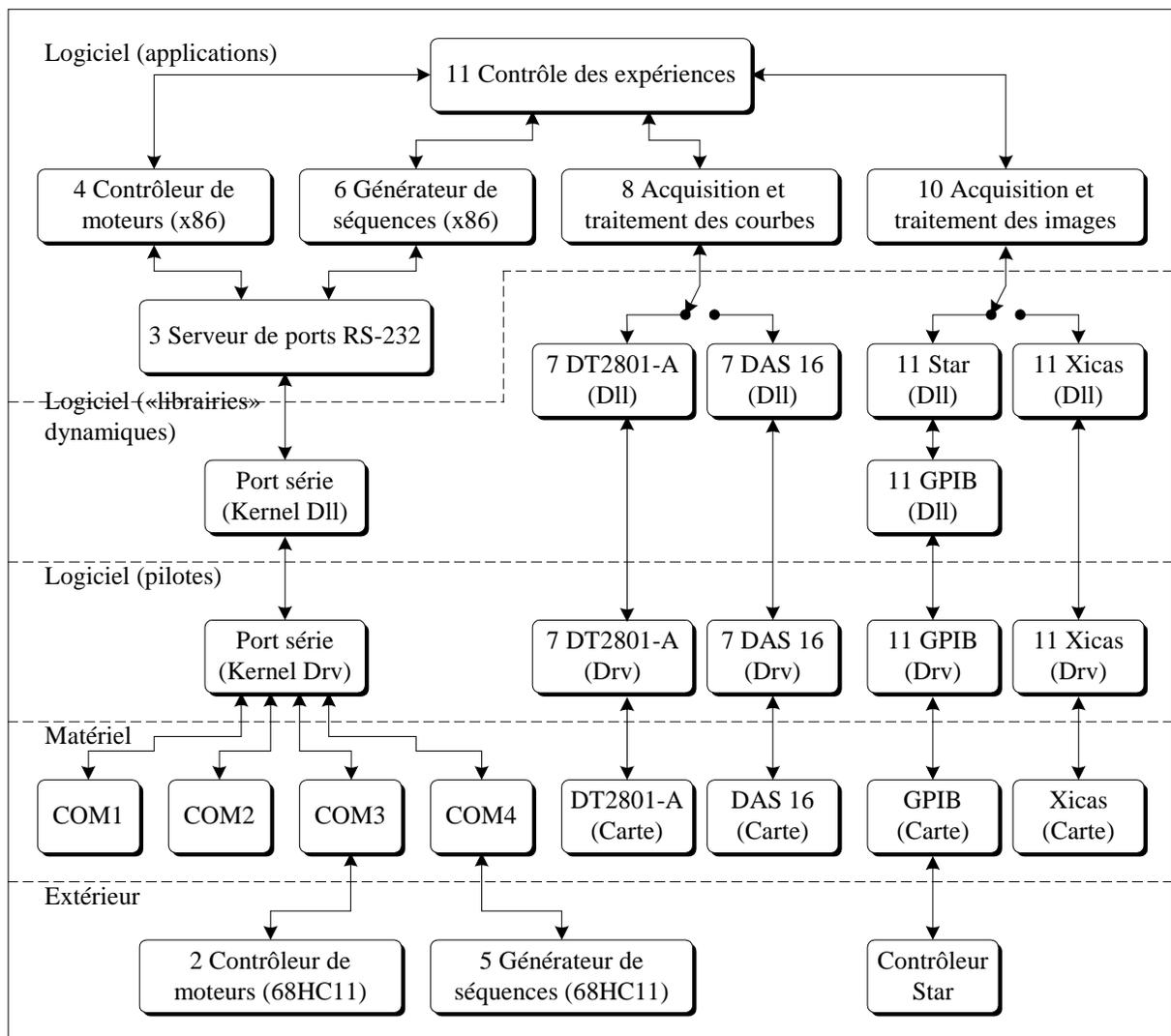


Figure 4-1 : Organigramme de l'ordinateur principal.

seconde partie (Logiciel (« bibliothèques » dynamiques)) contient les « bibliothèques » utilisées par le système. La communication entre les applications et les DLL est faite par un simple appel de

fonctions standard. La troisième partie (Logiciel (pilotes)) regroupe les différents pilotes. La communication entre les DLL et les pilotes est faite par l'intermédiaire de fonctions systèmes spécialement réservées à cette fin. La quatrième partie (Matériel) présente les différents périphériques nécessaires au montage. La communication entre les périphériques et les pilotes est faite à l'aide des DMA, les interruptions ou les accès aux registres des cartes. La dernière partie (Extérieur) montre la connexion série avec le *Contrôleur de moteurs* et le *Générateur de séquences* et la communication GPIB avec le contrôleur Star.

Certains des logiciels de la figure 4-1 de la page précédente ont été réalisés, d'autres ont été achetés (GPIB) et d'autres font partie intégrante du système d'exploitation (Kernel). Le numéro présent à côté des noms des logiciels fait le lien avec la section du chapitre expliquant le fonctionnement du logiciel.

4.2 L'application *Contrôleur de moteurs (68HC11)*

Lors de la réalisation du montage, un contrôleur a été construit pour gérer efficacement tous les moteurs du montage (voir section 3.5, page 29). Puisque le contrôleur est muni d'un microcontrôleur, il a fallu développer un logiciel pour qu'il fonctionne. Cette application a comme rôle principal d'établir une communication entre l'ordinateur principal et les circuits intégrés spécialisés (HCTL1100) responsables de l'asservissement et du déplacement des moteurs.

4.2.1 Les configurations de départ

La première tâche à être exécutée après la mise sous tension du contrôleur est la configuration du microcontrôleur et des HCTL1100. Pour ce faire, la tâche est subdivisée en quatre étapes différentes :

- La première étape consiste à effectuer la configuration des registres du microcontrôleur et des variables globales pour rendre opérationnels la communication série, l'accès aux HCTL1100 et toutes les interruptions utilisées par l'application.

- La seconde étape permet de vérifier si le contenu de la mémoire morte (réalisé avec une mémoire vive contenant une pile) a été perdu. Le cas échéant, une configuration par défaut est utilisée et dans le cas contraire, la configuration présente dans la mémoire avec pile est utilisée. Lorsque l'ordinateur maître se connectera, il devra vérifier si l'erreur s'est produite et si c'est le cas, il devra mettre à jour la configuration du contrôleur.
- Puisqu'il est possible de mettre d'une à six cartes dans le contrôleur et un nombre différent de HCTL1100 (1 à 6) sur chacune des cartes, la troisième étape fait la détection des HCTL1100 présents dans le contrôleur. Par la suite, on déduit les cartes présentes à partir des HCTL1100 présents. S'il n'y a aucun HCTL1100 sur une carte, la carte est considérée comme absente même si elle est présente. Cette vérification est très simple. Il suffit d'écrire successivement deux nombres aléatoires dans un registre correspondant à chacun des HCTL1100. Si, après leur écriture, on relit les mêmes nombres, c'est que le HCTL1100 est présent. Après la détection d'un moteur, on initialise les HCTL1100 avec les valeurs obtenues à l'étape précédente.
- La dernière étape vérifie s'il y avait des moteurs en mouvement lors de la mise hors tension du contrôleur. Cette vérification est possible car, lors du déplacement d'un moteur, il y a un fanion dans la mémoire avec pile qui indique que le moteur est en mouvement. Il est alors facile de lire l'état du fanion après la mise sous tension du circuit pour en déduire l'état des moteurs lors de la mise hors tension du circuit. Cette vérification est très importante, car le contrôleur perdrait la position d'un moteur qui serait en mouvement lors de la mise hors tension du circuit. Dans une telle situation, il faut considérer que les calibrations du montage qui dépendent du moteur sont incorrectes et qu'il faut refaire ces calibrations. L'ordinateur maître devra donc vérifier pour chacun des moteurs si une erreur s'est produite.

Après ces différentes vérifications, il ne reste qu'à activer la communication série et amorcer la boucle principale de traitement de l'application.

4.2.2 Les queues de communication

Avant de passer à la boucle principale de traitement, nous présentons le fonctionnement du système de communication entre l'ordinateur principal et le contrôleur. Pour éviter que la boucle principale attende après la communication sérielle, il y a deux queues de communication qui sont utilisées. La première sert à la réception des caractères et la seconde, à la transmission des caractères.

Une queue est en réalité un tableau d'une longueur fixe ; la manipulation du tableau est faite à l'aide de deux index. Le premier indique le début des données et le second, la fin des données. Lorsque l'on ajoute des données dans la queue, on avance l'index de la fin. Quand l'index de la fin est à une position avant l'index du début, c'est que la queue est pleine. Lorsque l'on vide la queue, on avance l'index du début. Quand les deux index sont identiques, la queue est alors vide. La queue est considérée comme circulaire. Si un index est à la fin du tableau et qu'il faut l'augmenter, alors on le positionne au début du tableau. Cette technique est très pratique et évite de déplacer les données dans le tableau.

La manipulation des deux queues est indépendante et différente. Pour la queue de réception, le remplissage est fait par une routine d'interruption qui est appelée lorsqu'il y a un caractère de reçu par le UART. La routine est très simple. Elle fait la lecture du caractère reçu par le UART et le place dans la queue de la réception. Si la queue est pleine, la ligne RTS du UART est désactivée et la communication sérielle est interrompue jusqu'au moment où la queue sera vidée. La lecture de la queue est faite par la boucle principale. À chaque itération de la boucle principale, il y a un caractère de la queue qui est lu.

Dans le cas de la queue de transmission, le remplissage est fait par l'intermédiaire d'une fonction qui est appelée par une autre fonction de l'application qui désire transmettre de l'information à l'ordinateur maître. La fonction possède deux paramètres qui consistent en un pointeur au bloc de données à transmettre et la longueur du bloc de données. Par contre, la liste de transmission est vidée de la même façon que la liste de réception. C'est donc la boucle principale qui retire un caractère de la liste à chaque itération.

4.2.3 La boucle principale

La boucle principale de l'application est une boucle sans fin qui s'exécute jusqu'à la mise hors tension du contrôleur. C'est elle qui dirige toutes les opérations. L'organigramme de la figure 4-2 présente les opérations réalisées. Le tout commence par la configuration du

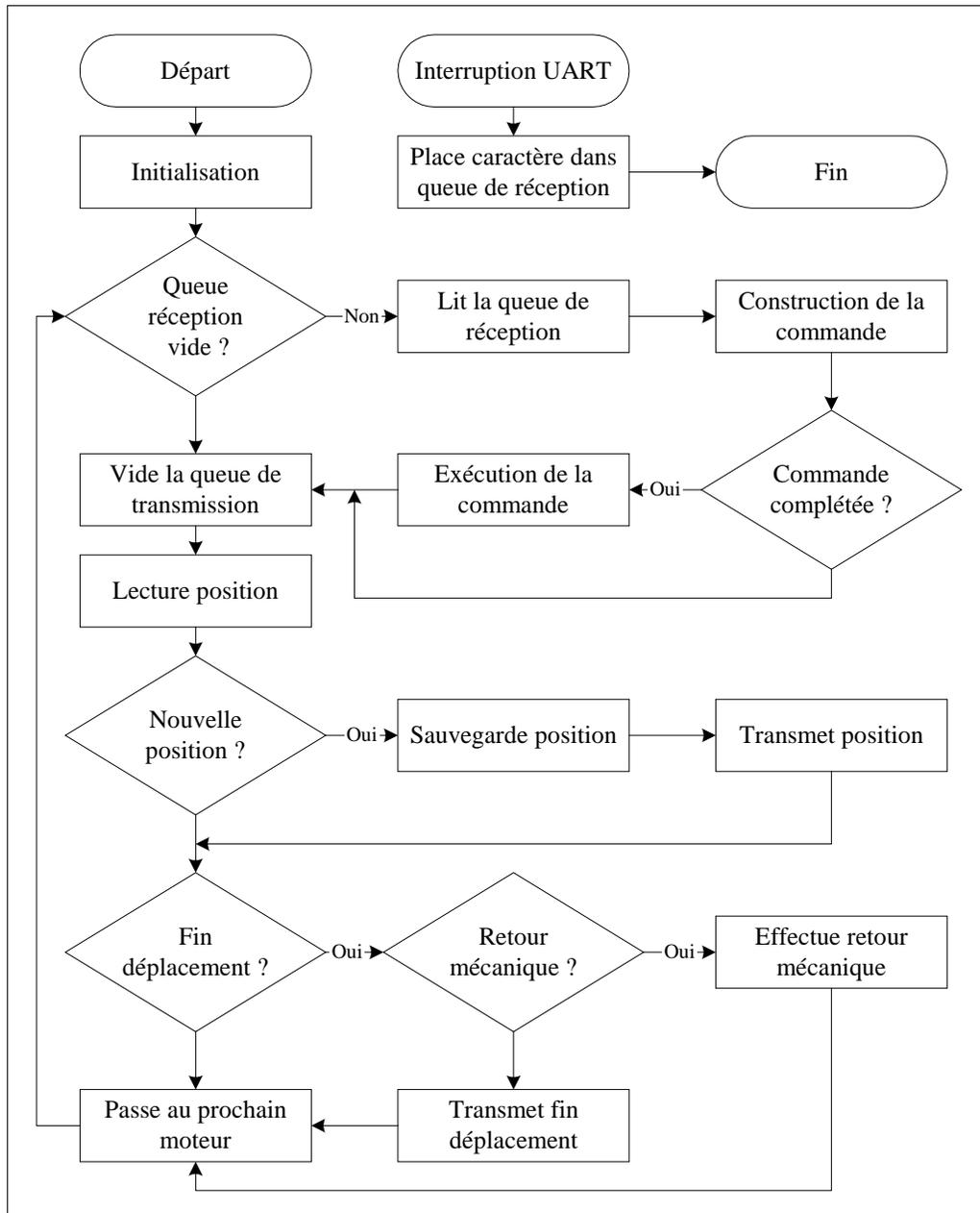


Figure 4-2 : Algorithme de la boucle principale de l'application *Contrôleur de moteurs (68HC11)*.

contrôleur qui a été expliquée à la section 4.2.1 de la page 76. Par la suite, il y a lecture d'un

caractère sur la queue de réception à condition qu'elle ne soit pas vide. Le caractère est ensuite transmis à une fonction qui décode la commande reçue. Lorsque tous les caractères de la commande sont reçus, la commande est exécutée.

Les commandes sont construites à partir d'un seul octet. Donc, quand la fonction reçoit le premier caractère, elle identifie la commande et anticipe le nombre de caractères nécessaires pour représenter les paramètres de la commande. Avec une telle structure, il est évident que l'ordinateur principal doit envoyer la commande et ses paramètres dans un ordre prédéterminé.

La prochaine opération consiste à retirer un caractère de la queue de transmission et à l'envoyer à l'ordinateur principal. Si l'ordinateur principal n'est pas prêt à recevoir le caractère, la boucle continue sans transmettre aucun caractère.

Par la suite, l'algorithme fait la lecture de la position du moteur. S'il s'agit d'une nouvelle position, la position est sauvegardée en mémoire et elle est transmise à l'ordinateur principal. Pour éviter qu'il y ait engorgement sur la ligne de transmission, la position du moteur est envoyée à l'ordinateur principal une seule fois par seconde. Pour que la position soit envoyée à l'ordinateur principal, il faut qu'il ait préalablement avisé le contrôleur qu'il désirait se voir informé de la position des moteurs.

L'opération suivante consiste à vérifier si le moteur a terminé son déplacement. La fin du déplacement d'un moteur est signalée par le fait que la position du moteur n'a pas changé pour une vingtaine d'itérations. Puisqu'il peut arriver qu'un moteur oscille indéfiniment entre une ou deux positions d'encodeur (à la suite d'une mauvaise caractérisation du moteur), l'algorithme considère un moteur comme arrêté quand sa position reste à l'intérieur d'une plage de positions. Cette largeur de plage est définie comme la tolérance de l'arrêt. Lorsque la fin du déplacement est identifiée, le contrôleur effectue la correction du jeu mécanique quand c'est nécessaire. Pour que la correction du jeu mécanique soit effectuée, il faut que l'option ait été activée pour ce moteur et qu'il s'agisse de la fin d'un déplacement de type absolu ou relatif (voir section 4.4.2, page 88). S'il n'y a pas de correction du jeu mécanique, l'ordinateur

principal est avisé de la fin du déplacement du moteur s'il a préalablement avisé le contrôleur qu'il désirait être informé de la fin des déplacements des moteurs.

L'algorithme ferme alors la boucle en passant au prochain moteur à contrôler. Quand tous les moteurs ont été contrôlés, l'algorithme recommence avec le premier moteur.

4.2.4 Les commandes

Les commandes sont essentiellement des fonctions que l'ordinateur principal peut appeler pour utiliser les fonctionnalités du contrôleur. Le codage des commandes est très simple : il comprend un premier octet représentant le numéro de la commande suivi d'une série d'octets représentant ses paramètres. Le nombre d'octets utilisé par les paramètres dépend du type de commande. Certaines commandes ne nécessitent qu'un seul octet et d'autres, une centaine d'octets. Cependant, les commandes les plus souvent utilisées requièrent environ six octets.

Afin d'informer l'ordinateur principal de l'acceptation d'une commande, le contrôleur retourne un code indiquant à l'ordinateur principal que la commande a été acceptée. Le refus d'une commande peut être occasionné par un paramètre qui n'est pas valide ou, tout simplement, par le fait que la commande n'existe pas.

Le contrôleur possède plusieurs types de commandes qui sont subdivisées en six catégories :

- Commandes systèmes : Les commandes de cette catégorie permettent de gérer la communication entre l'ordinateur principal et le contrôleur. Puisque ces commandes sont indépendantes des cartes de périphérie, elles sont implantées de la même façon sur les deux contrôleurs du montage. La même implantation sur tous les contrôleurs permet à l'ordinateur principal d'utiliser une de ces commandes pour connaître le nom du contrôleur.
- Commandes de configuration : Comme le nom l'indique, ces commandes sont utilisées pour modifier la configuration du contrôleur aussi bien que la configuration des HCTL1100.

- Commandes de déplacement : Cette catégorie regroupe les commandes les plus utilisées par l'ordinateur principal. Il s'agit des commandes permettant de déplacer les moteurs.
- Commandes de calibration : Les commandes de calibration sont utilisées lors de la construction du contrôleur pour ajuster les plages de tension des convertisseurs numériques analogiques présents sur les cartes de périphérie.
- Commandes d'informations : Cette catégorie contient les commandes que l'ordinateur principal utilise pour connaître l'état du contrôleur et des moteurs. Il y a des fonctions pour connaître la position des moteurs, les cartes et moteurs présents sur le contrôleur, les erreurs générées lors de la configuration du contrôleur, etc.
- Commandes spécialisées : La dernière catégorie regroupe les commandes permettant l'exécution d'algorithmes de haut niveau. À titre d'exemple, il y a les commandes utilisées pour effectuer la recherche de l'index d'un moteur.

4.2.5 Discussion

L'utilisation d'un petit microcontrôleur impose certaines limites de rapidité. Par exemple, le microcontrôleur a une fréquence d'horloge de 2 MHz et peut seulement effectuer des opérations mathématiques sur des entiers de 16 bits. Il est donc approprié que l'information envoyée au microcontrôleur soit déjà traitée et prête à être inscrite dans les différents registres des HCTL1100 ou du microcontrôleur.

De plus, l'application est construite de façon à diminuer à l'essentiel le nombre de données circulant entre l'ordinateur principal et le contrôleur. Pour ce faire, les commandes sont codées par un octet, et les paramètres sont tous envoyés en nombres binaires. La seule information de type texte est le nom de l'application que l'ordinateur peut se procurer par l'intermédiaire d'une commande système. De plus, le codage du nom des moteurs se limite à un numéro entre 0 et 63 et le codage du nom des cartes, à un numéro entre 0 et 5. Les positions des moteurs sont transmises en position d'encodeur et les vitesses et accélérations, en nombre de pas d'encodeur par période d'échantillonnage.

4.3 L'application *Serveur de port RS-232*

Cette application peut paraître superflue à première vue sauf qu'elle rend les applications *Contrôleur de moteurs (x86)* et *Générateur de séquences (x86)* indépendantes du support de communication utilisé. Elle peut supporter quatre lignes de communication indépendantes où chaque ligne est composée d'une communication DDE et d'un port série (voir figure 4-3). L'application est aussi munie d'une petite interface usager permettant de voir l'information circulant sur les lignes de communication.

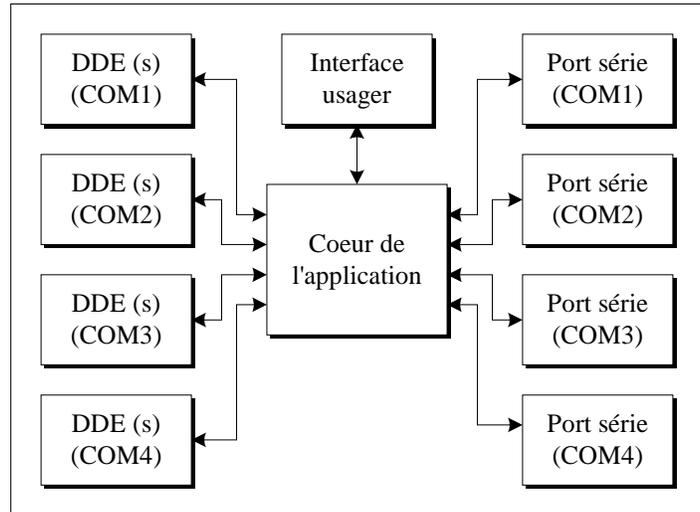


Figure 4-3 : Organigramme de l'application *Serveur de port RS-232*.

La communication débute par une demande de connexion de la part de l'application de type client en spécifiant l'un des quatre topiques disponibles, c'est-à-dire COM1, COM2, COM3 ou COM4. La communication est acceptée quand le topique et le port série équivalent sont disponibles. La gestion des ports séries est très bien faite par Windows NT et lorsqu'une application réserve un port série, le port devient inaccessible aux autres applications. Il est donc important de bien libérer les ports séries quand ils ne sont plus utilisés. Puisque le port COM1 est souvent utilisé par la souris, il est rarement accessible.

Lorsque la communication est acceptée, l'application crée un « thread ». Le « thread » permet d'avoir une fonction qui tourne indépendamment et en parallèle avec le reste de l'application. Cela permet d'utiliser une fonction système qui arrête l'exécution de la fonction « thread » pour attendre l'arrivée de caractères sur le port série sans interrompre le fonctionnement du reste de l'application. De plus, cette fonction système est construite de telle façon qu'elle minimise l'utilisation du temps du microprocesseur. En réalité, elle indique au système qu'elle veut être avisée lors de la prochaine interruption générée par le port série.

Le fonctionnement complet d'une ligne de communication est présenté à la figure 4-4. Il y a premièrement la fonction « thread ». Cette fonction est exécutée lors de la connexion du client et utilise une fonction système qui attend la réception de caractères sur le port série. Lorsque

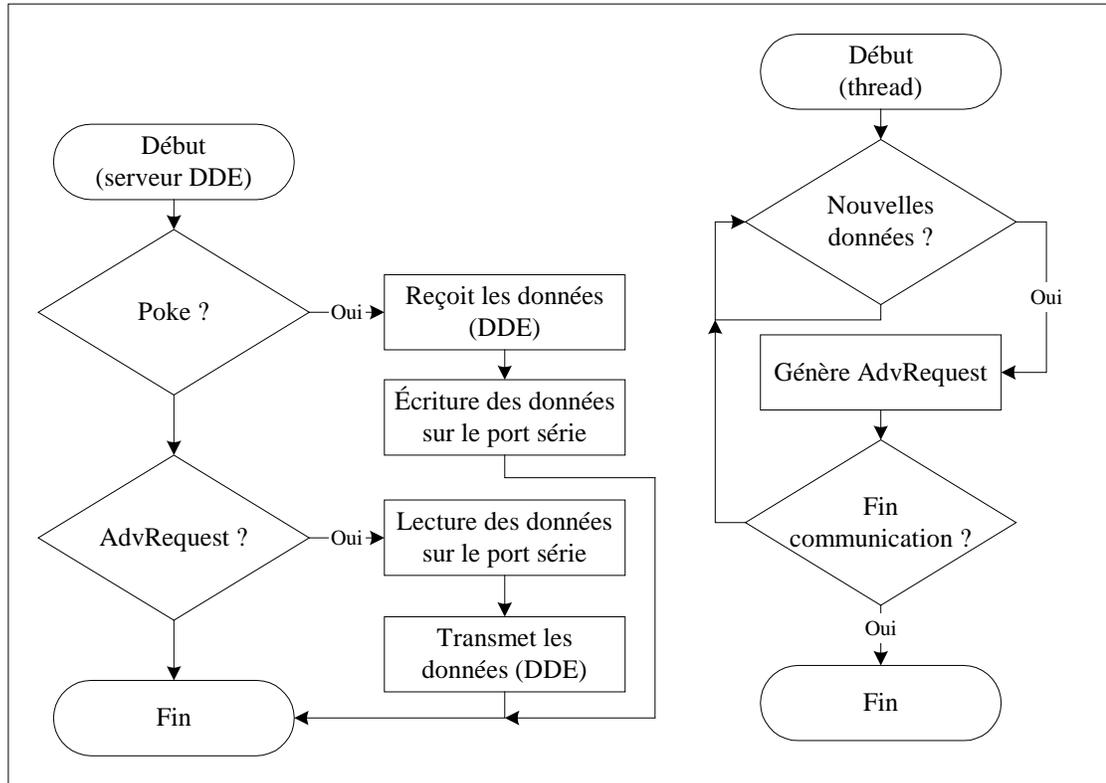


Figure 4-4 : Algorithme d'une ligne de communication de l'application *Serveur de port RS-232*.

c'est le cas, la fonction avise le serveur DDE par l'intermédiaire d'une commande *AdviceRequest*. Quand la connexion est terminée par le client, la fonction est aussi terminée et, du même coup, le « thread » est détruit. La seconde fonction est la fonction *Callback* du serveur DDE. Cette fonction est exécutée par le système d'exploitation chaque fois qu'il y a une transaction DDE à effectuer. Les deux transactions les plus importantes sont la réception de données du client et la transmission de données au client. La réception de données se fait par l'intermédiaire de la commande *Poke* envoyée par le client. Le serveur prend alors les données attachées à cette commande et les envoie sur le port série. De plus, la transmission de données au client se fait par l'intermédiaire de la commande *AdviceRequest* qui a été envoyée par la fonction « thread ». À ce moment, l'algorithme fait la lecture des données présentes sur le port série et les envoie au client.

La manipulation du port série est très simple et consiste à utiliser la fonction *CreateFile*, en utilisant comme nom de fichier COMx, pour l'ouverture du port série, la fonction *WriteFile* pour écrire sur le port série et la fonction *ReadFile* pour lire le contenu du port série. Il existe aussi une autre fonction pour modifier la configuration des différents paramètres du port série. Puisque les ports série sont munis de queues de réception et de transmission, il arrive souvent que, lors de la lecture du port série, il y ait plusieurs caractères qui soient lus simultanément. Cela évite d'avoir une transaction DDE pour chaque caractère reçu. De plus, la queue de transmission empêche que la fonction *WriteFile* attende que le bloc de données soit complètement transmis avant de retourner la main à l'application.

Ici, nous avons seulement présenté les grandes lignes de la présente application. Cependant, l'application en soi est un peu plus complexe. C'est une application qui fonctionne très bien et qui est complètement asynchrone. Nous entendons par asynchrone qu'il n'y a aucune horloge qui est utilisée pour vérifier s'il y a des données présentes sur le port série et que l'application de type client n'a pas besoin d'interroger périodiquement le serveur pour savoir s'il y a eu réception de données. Cette application ne consomme donc aucun temps du microprocesseur quand il n'y a pas de données à recevoir ou à transmettre.

4.4 L'application *Contrôleur de moteurs (x86)*

Afin de permettre un bon contrôle des moteurs, l'application *Contrôleur de moteurs (x86)* doit accomplir quatre tâches essentielles. Dans un premier temps, l'application doit, de toute évidence, posséder une interface permettant à l'utilisateur de commander le déplacement des moteurs et d'en connaître la position. Afin de mieux refléter le rôle de chacun des moteurs, la seconde tâche permet d'assigner un nom à chacun des moteurs et d'exprimer les déplacements et les positions des moteurs à l'aide des unités appropriées. À titre d'exemple, le moteur responsable du déplacement de la base selon l'axe des Z est appelé *Base Z* et la position est exprimée en μm , la vitesse, en $\mu\text{m/s}$ et l'accélération, en $\mu\text{m/s}^2$. En outre, pour parvenir à bien contrôler chacun des moteurs, la troisième tâche permet d'ajuster la configuration des moteurs par l'intermédiaire de fenêtres de configuration ou d'algorithmes spécialisés. Enfin, la dernière tâche consiste à rendre accessibles à une autre application le contrôle et la position des moteurs en utilisant le protocole de communication DDE.

4.4.1 Le fonctionnement général de l'application

Pour parvenir à construire une application avec deux interfaces, il faut subdiviser l'application en trois grands blocs (voir figure 4-5). Le premier correspond au cœur de l'application qui contient les configurations, les positions et toutes les commandes possibles. C'est aussi ce

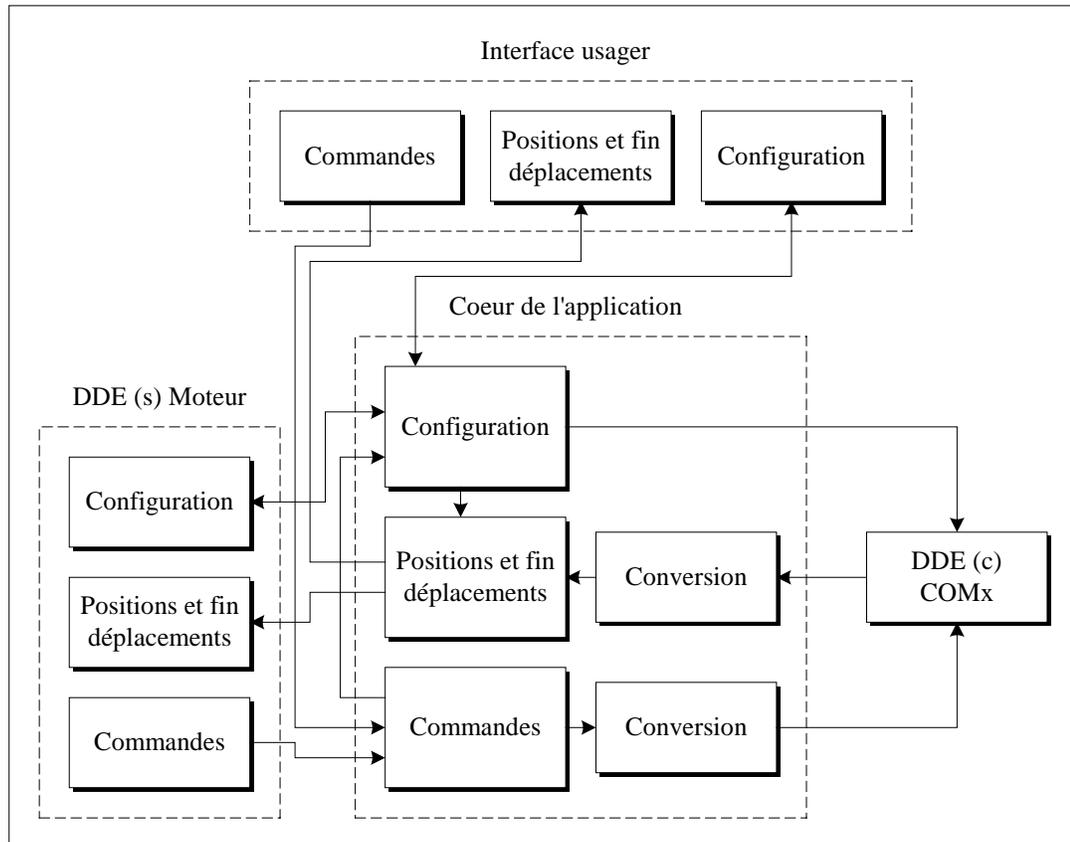


Figure 4-5 : Organigramme de l'application *Contrôleur de moteurs (x86)*.

bloc qui est responsable de la communication, de type DDE, avec le contrôleur. Donc, toutes les informations et commandes relatives aux moteurs passeront par le cœur de l'application. L'accès au cœur de l'application peut être fait à travers deux interfaces qui correspondent aux deux autres blocs. La première interface correspond à l'interface usager de l'application et donne accès à toutes les commandes et les configurations. La seconde interface est un serveur DDE qui permet à une autre application d'accéder à un nombre limité de commandes et de configurations.

L'exécution d'une commande commence par un appel au module de commandes du cœur de l'application par une des deux interfaces. À la suite de cet appel, le module de commandes

fait la validation des paramètres et vérifie si le contrôleur peut exécuter la commande. Il arrive que le contrôleur rejette une commande, car le moteur est déjà en mouvement. Avant de transmettre la commande au contrôleur, une conversion est effectuée pour transformer les unités réelles de l'application en unités supportées par le contrôleur. Cette conversion réalise la seconde tâche de l'application et permet aussi d'accélérer la transmission en diminuant le nombre de caractères à transmettre au contrôleur.

Lorsque la position d'un moteur change, le contrôleur envoie la nouvelle position du moteur au cœur de l'application. Les positions sont alors converties et transmises à l'interface usager et à l'application de type client à l'aide d'un lien chaud de la communication DDE. Le cœur de l'application et les interfaces sont aussi avisés de la fin du déplacement d'un moteur.

La modification de la configuration peut être faite à partir de l'interface usager de l'application de type client ou par le module de commandes qui possède deux fonctions permettant de sauvegarder la configuration dans un fichier et de charger une configuration préalablement sauvegardée dans un fichier. L'utilisation des fichiers de configuration n'est pas nécessaire, car l'application, lors de sa fermeture, fait une sauvegarde automatique de la configuration dans le *Registry* de Windows NT. Évidemment, l'application utilise la configuration présente dans le *Registry*, lors de son ouverture, pour remettre l'application dans le même état que lors de la fermeture. L'utilisation du *Registry* est très pratique, car elle permet de subdiviser les configurations en deux groupes. Le premier groupe permet de sauvegarder la configuration commune à tous les usagers et l'autre groupe de sauvegarder la configuration spécifique à chaque usager. L'accès au groupe spécifique à l'utilisateur permet d'avoir une configuration différente pour chacun des usagers sans aucune gestion spéciale de l'application.

Afin que les deux interfaces soient bien synchronisées entre elles, la modification d'une configuration par l'une des deux interfaces est automatiquement signalée à la seconde. Cependant, l'avis est seulement envoyé pour les configurations supportées par les deux interfaces. Il arrive aussi que le changement d'une configuration nécessite la mise à jour de la position d'un ou de plusieurs moteurs.

4.4.2 Les différentes façons de déplacer un moteur

Afin de faciliter le déplacement des moteurs selon les différentes situations, il est apparu qu'il fallait déplacer les moteurs de trois façons différentes. La première façon (déplacement absolu) consiste à inscrire la position de la destination dans la liste déroulante appropriée et à



Figure 4-6 : Fenêtre principale de l'application *Contrôleur de moteurs* (x86).

activer la commande de déplacement pour que le moteur se rende à la position de destination. La seconde façon (déplacement relatif) consiste à inscrire la distance à parcourir dans la liste déroulante appropriée et à activer la commande de déplacement avec le bouton gauche de la souris pour un déplacement négatif et avec le bouton droit de la souris pour un déplacement positif. Pour un tel déplacement, la position de destination est donnée par la position de départ du moteur avec en plus ou en moins la distance à parcourir. La dernière façon (déplacement

continu) est très différente des deux premières. Elle consiste à inscrire la vitesse de déplacement du moteur dans la liste déroulante appropriée. Le déplacement débute lorsque la commande de déplacement est activée, en appuyant sur le bouton, et s'arrête lorsque la commande de déplacement est désactivée, en relâchant le bouton. Tout comme pour le déplacement relatif, le choix du bouton de la souris indique le sens du déplacement. Afin d'éviter les démarrages et les arrêts brusques du moteur, l'accélération et la décélération du moteur sont contrôlées. La sélection de la façon de déplacer un moteur est faite à l'aide de la série de trois boutons de radio (*Absolu*, *Relatif* et *Continu*), et le déplacement est activé en appuyant sur le grand bouton (voir figure 4-6, page précédente).

Pour les deux premières façons de déplacer un moteur, il est possible d'utiliser deux algorithmes de déplacement. Le premier type génère un déplacement où seulement le filtre à avance de phase est utilisé. Dans le cas présent, la tension appliquée au moteur est proportionnelle à la distance qu'il reste à parcourir. Avec cet algorithme, le moteur démarre brusquement pour atteindre sa vitesse maximale et, lors de la fin du déplacement, il décélère rapidement pour finalement atteindre, tout en oscillant un peu autour d'elle, la position de destination. L'autre algorithme génère un déplacement de forme trapézoïdale où l'accélération, la vitesse maximale et la décélération du moteur sont limitées. Cela évite au moteur de dépasser la position de destination et, généralement, l'erreur entre la position de destination voulue et celle obtenue est plus faible qu'avec le premier algorithme. L'utilisation de l'un ou l'autre des algorithmes dépend de la fonction du moteur. Si on désire déplacer un moteur rapidement et brusquement, le premier algorithme est de rigueur; par contre, si un moteur nécessite une très grande précision, le deuxième algorithme est préférable. À titre d'exemple, l'utilisation du premier algorithme se prête bien au déplacement de la microélectrode, car il oblige des déplacements brusques. Par contre, les déplacements générés par les moteurs responsables de l'emplacement des faisceaux lumineux doivent être très précis, donc l'utilisation du second algorithme est plus appropriée. La sélection de l'algorithme est faite à l'aide du bouton de sélection *Trapézoïde* de l'application.

Lors de la réalisation de pièces mécaniques, il est pratiquement impossible qu'il n'y ait aucun jeu mécanique, ce qui a comme effet de fausser la position réelle des objets. Cela signifie que la position indiquée par l'encodeur optique des moteurs est différente de la position réelle de l'objet. Par exemple, la figure 4-7-A présente le cas où la position finale d'un déplacement linéaire, représentée par une barre verticale, est différente selon la direction d'approche pour une même lecture de l'encodeur optique. Pour contourner ce problème, il faut toujours terminer le déplacement d'un moteur en

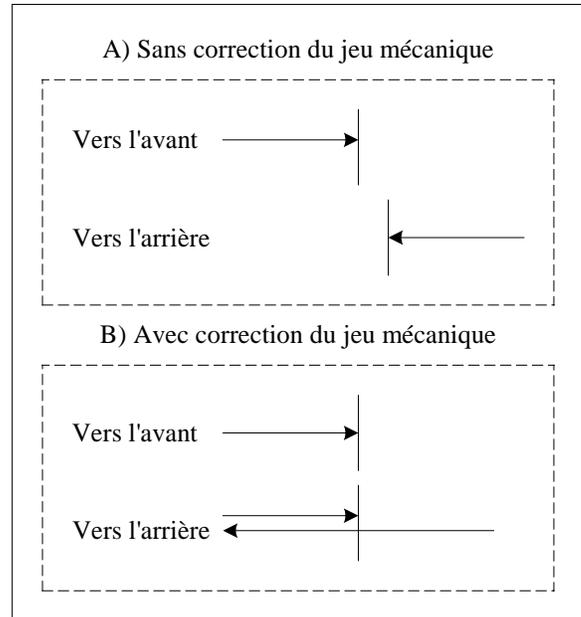


Figure 4-7 : Exemple de l'effet du jeu mécanique sur un déplacement linéaire.

direction. Pour ce faire, l'application dépasse le point d'arrivée pour les déplacements à reculons et retourne au point d'arrivée par la suite (voir figure 4-7-B). Pour que le tout fonctionne correctement, la distance de dépassement du point d'arrivée doit être assez grande pour entièrement enlever le jeu mécanique. Il est à noter que l'utilisation de la correction du jeu mécanique n'est pas efficace avec le premier algorithme de déplacement d'un moteur, car l'algorithme génère une oscillation autour de la position de destination.

La correction du jeu mécanique est automatiquement effectuée pour les commandes de déplacement absolu et relatif d'un moteur, si l'utilisateur le désire. Il est aussi possible d'effectuer la correction du jeu mécanique sans déplacer un moteur. Dans ce dernier cas, le moteur est simplement reculé puis avancé de la distance requise pour enlever le jeu mécanique.

4.4.3 Discussion

Lors de la réalisation de l'application *Contrôleur de moteurs (x86)*, la difficulté majeure a été de produire un logiciel asynchrone permettant de travailler efficacement avec deux interfaces

en parallèle. Le reste de l'application consistait simplement à donner un accès convivial aux commandes du contrôleur.

Le fonctionnement de la présente application est tel que les moteurs sont tous indépendants les uns des autres et qu'ils sont tous contrôlés de la même façon à l'exception des paramètres de l'asservissement qui peuvent être ajustés pour tenir compte de la force du moteur et de sa charge. Le lien entre les moteurs et le stimulateur optique est donc assuré au niveau de l'application de type client *Contrôle des expériences*.

4.5 L'application *Générateur de séquences (68HC11)*

Tout comme pour le *Contrôleur de moteurs*, un logiciel permettant de communiquer avec le contrôleur *Générateur de séquences* (voir section 3.6, page 38) et gérant efficacement les synchronisations temporelles du montage est nécessaire. Puisque la seule différence entre les deux contrôleurs réside dans les périphériques utilisés, seules les commandes nécessaires au contrôle des périphériques ainsi que les notifications du contrôleur vers l'ordinateur principal sont différentes.

4.5.1 Les configurations de départ

La première tâche à être exécutée après la mise sous tension du contrôleur est la configuration du microcontrôleur et des séquenceurs. Pour ce faire, la configuration est subdivisée en deux étapes différentes :

- La première étape consiste à effectuer la configuration des registres du microcontrôleur et des variables globales pour rendre opérationnels la communication sérielle, l'accès aux séquenceurs et toutes les interruptions utilisées par l'application.
- Puisqu'il est possible de mettre deux cartes de périphériques dans le contrôleur, la seconde étape consiste à détecter la présence des deux cartes. La détection est faite en écrivant successivement deux nombres aléatoires dans les mémoires des deux cartes. Si, après l'écriture d'un nombre, le nombre relu est le même, c'est que la carte est présente. Après la détection d'une carte, on initialise les séquenceurs avec les configurations par défaut.

La configuration prend fin en activant la communication sérielle et en amorçant la boucle principale de traitement de l'application.

4.5.2 Les queues de communication

Pour communiquer avec l'ordinateur principal, l'application utilise une queue de réception et une queue de transmission. Le fonctionnement des queues est identique au fonctionnement des queues de l'application *Contrôleur de moteurs (68HC11)* (voir section 4.2.2, page 78).

4.5.3 La boucle principale

Le fonctionnement de la boucle principale de cette application est semblable au fonctionnement de la boucle principale de l'application *Contrôleur des moteurs (68HC11)* (voir section 4.2.3, page 79), seules trois différences se présentent.

Les deux premières différences résident dans la lecture de l'état des séquenceurs au lieu de la lecture de la position des moteurs et la signalisation de la fin d'une séquence d'un séquenceur au lieu de la fin du positionnement d'un moteur. La dernière différence réside dans le fait que l'algorithme fait la lecture de tous les séquenceurs à chaque itération. Cela est possible, car la lecture de l'état d'un séquenceur est environ mille fois plus rapide que la lecture de la position d'un moteur et qu'il y a seulement huit ou seize séquenceurs au lieu de soixante-quatre moteurs. La figure 4-8 de la page suivante présente le nouvel algorithme de l'application.

4.5.4 Les commandes

La nomenclature des commandes est exactement la même que pour le *Contrôleur de moteurs* (voir section 4.2.4, page 81). La seule différence réside dans les tâches que les commandes doivent accomplir. Le contrôleur possède plusieurs commandes qui sont répertoriées dans les quatre catégories suivantes :

- Commandes systèmes : Les commandes de cette catégorie permettent de gérer la communication entre l'ordinateur principal et le contrôleur. Ces commandes sont

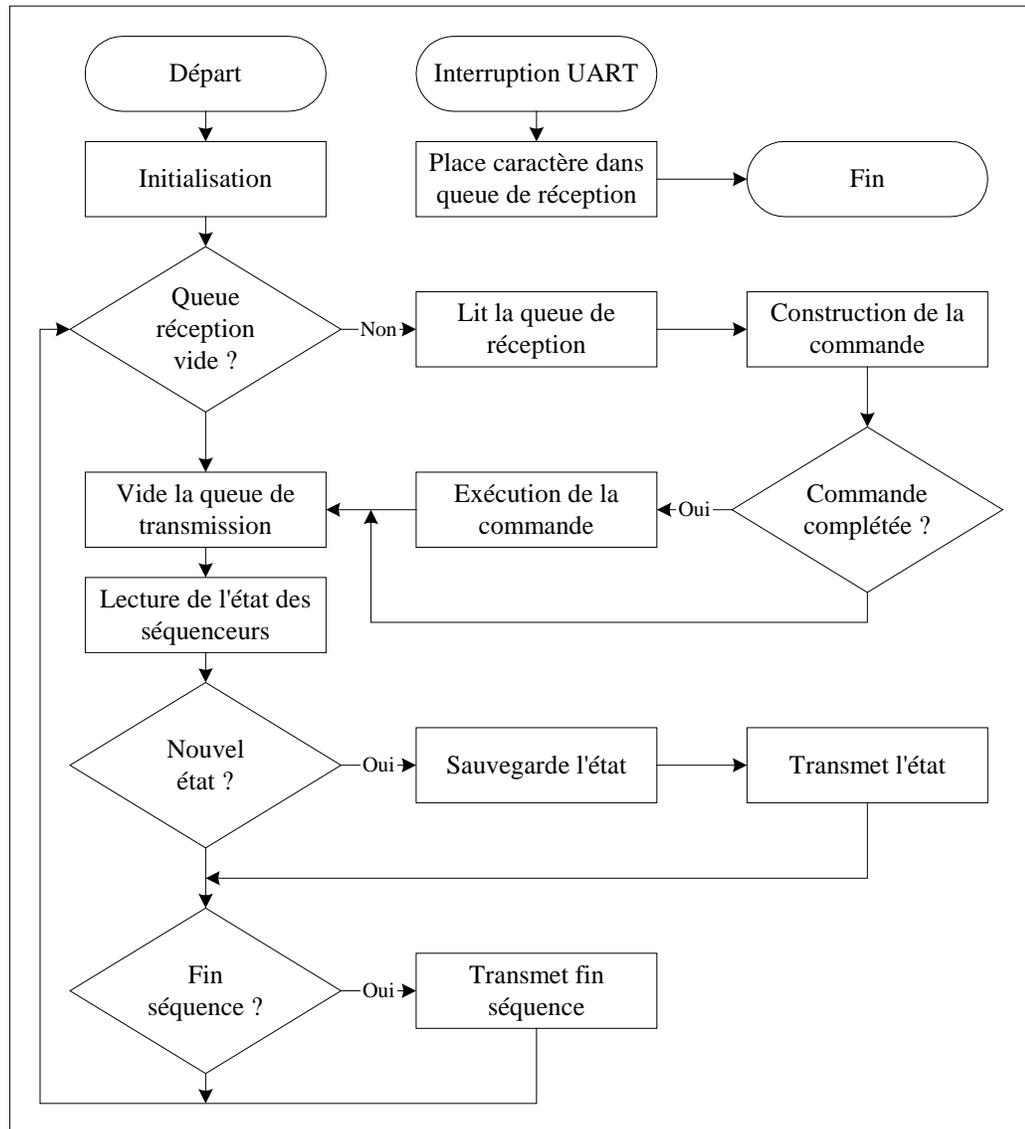


Figure 4-8 : Algorithme de l'application *Générateur de séquences (68HC11)*.

exactement les mêmes que les commandes du *Contrôleur de moteurs*, à l'exception du nom retourné par la commande responsable d'identifier le contrôleur.

- Commandes de configuration : Comme le nom l'indique, ces commandes sont utilisées pour modifier la configuration du contrôleur aussi bien que celle des séquenceurs.
- Commandes d'action : Cette catégorie regroupe les commandes les plus utilisées par l'ordinateur principal. Il s'agit des commandes utilisées pour contrôler le départ, l'arrêt et l'état des séquenceurs.

- Commandes d'informations : Ce type de commande contient les informations que l'ordinateur principal peut utiliser pour connaître certains états du contrôleur. Il y a des fonctions pour connaître l'état des séquenceurs, les cartes présentes sur le contrôleur, etc.

4.5.5 Discussion

Tout comme pour le *Contrôleur de moteurs*, il faut diminuer au minimum le nombre de transactions entre l'ordinateur principal et le contrôleur. Pour ce faire, seulement des commandes binaires sont utilisées, et la numérotation des séquenceurs est représentée par des nombres allant de 0 à 7, les cartes sont identifiées par les nombres 0 et 1 et les temps sont exprimés en facteur de 25 μ s.

De plus, il n'y a aucune synchronisation possible entre les deux cartes présentes sur le contrôleur. Donc, si l'on désire avoir certains événements synchronisés entre eux, ils devront tous être sur la même carte.

4.6 L'application Générateur de séquences (x86)

Afin de faciliter l'utilisation des séquenceurs, l'application *Générateur de séquences (x86)*, tout comme l'application *Contrôleur de moteurs (x86)*, doit réaliser quatre tâches bien précises. Dans un premier temps, l'application doit posséder une interface permettant à l'utilisateur de démarrer et d'arrêter les séquenceurs ainsi que de connaître l'état des séquences. Afin de mieux refléter l'utilisation des séquenceurs, la seconde tâche permet d'assigner un nom et d'exprimer les durées des séquences en ms. La troisième tâche permet d'ajuster la configuration des séquenceurs par l'intermédiaire d'une fenêtre de configuration. La dernière tâche de l'application consiste à permettre à une autre application de contrôler les séquenceurs en utilisant une communication DDE.

4.6.1 Le fonctionnement général de l'application

La réalisation d'une application possédant deux interfaces nécessite une subdivision de sa structure en trois blocs (voir figure 4-9, page suivante). Le premier bloc correspond au cœur

de l'application et contient la configuration, l'état des séquences et toutes les commandes. C'est aussi ce bloc qui est responsable de la communication avec le contrôleur à l'aide d'une communication DDE. Cela implique que toutes les informations et commandes relatives aux séquenceurs passent par le cœur de l'application, et l'accès au cœur de l'application peut être fait par l'intermédiaire de l'une des deux interfaces qui correspondent aux deux autres blocs.

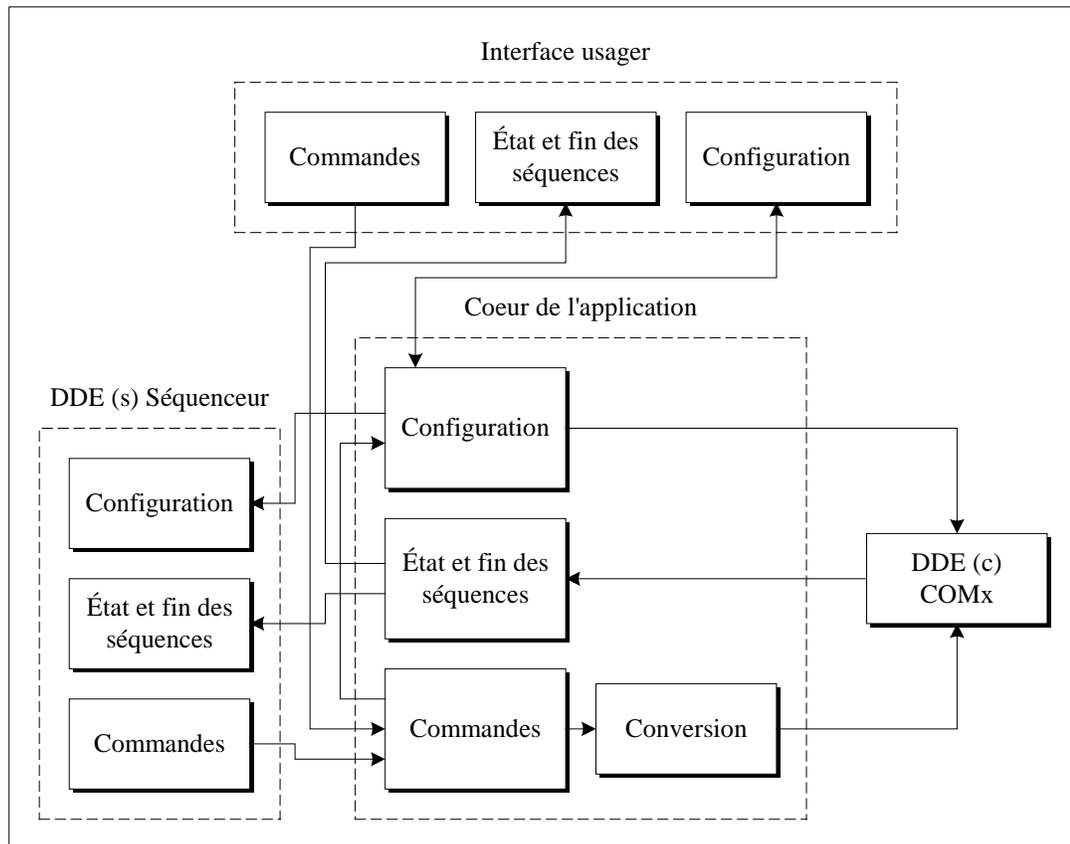


Figure 4-9 : Organigramme de l'application *Générateur de séquences (x86)*.

La première est l'interface usager de l'application qui donne accès à toutes les commandes et configurations, et la seconde correspond au serveur DDE qui permet à une autre application d'avoir accès à certaines commandes et configurations.

L'exécution d'une commande commence par un appel au module de commandes du cœur de l'application à l'aide d'une des deux interfaces. À la suite de cet appel, le module de commandes fait une validation des paramètres et les conversions nécessaires avant de transférer la commande au contrôleur. Puisque l'unité temporelle de base du contrôleur est de

25 μ s, la conversion consiste à multiplier les durées de l'application, exprimées en ms, par 40 lors de la transmission d'une durée vers le contrôleur.

Lorsque l'état d'une séquence change, ce nouvel état est transmis au cœur de l'application. Alors, le cœur de l'application avise l'interface usager et l'application de type client du nouvel état. Le contrôleur avise aussi l'application quand la séquence d'un séquenceur est terminée.

La modification de la configuration est accessible par l'interface usager ou à l'aide du module de commandes qui possède plusieurs fonctions permettant de sauvegarder la configuration dans un fichier et de charger une configuration préalablement sauvegardée. Cependant, tout comme pour l'application *Contrôleur de moteurs (x86)*, le *Registry* est utilisé pour sauvegarder les configurations de l'application. Il est toutefois possible de sauvegarder la configuration d'un seul séquenceur ou de tous les séquenceurs simultanément.

Afin que les deux interfaces affichent correctement le mode des séquenceurs, le changement de mode par une interface est toujours signalé à la seconde interface.

4.6.2 Le contrôle des séquenceurs

L'application *Contrôle des séquenceurs* est relativement simple. Chaque séquenceur comprend quatre boutons pour son contrôle. Il y a premièrement une série de trois boutons de radio permettant de changer le mode d'un séquenceur. Il est alors

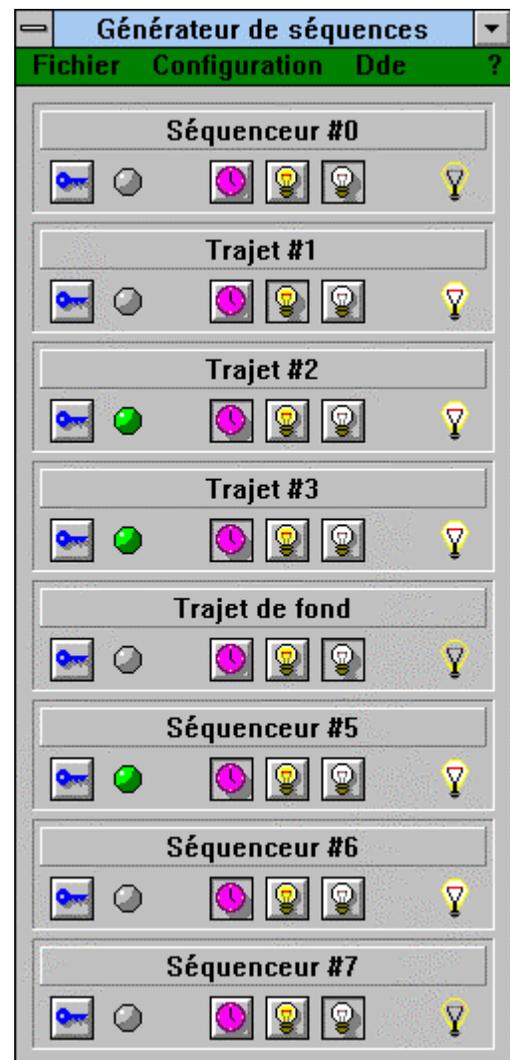


Figure 4-10 : Fenêtre principale de l'application *Générateur de séquences (x86)*.

possible d'activer le séquenceur () , de désactiver le séquenceur en maintenant l'état de la séquence dans un niveau logique haut () ou de désactiver le séquenceur en maintenant l'état de la séquence dans un état logique bas () . Un dernier bouton permet de démarrer et d'activer un séquenceur () . La figure 4-10 de la page précédente présente la fenêtre principale de l'application utilisée pour contrôler chacun des séquenceurs. En plus des contrôles des séquenceurs, il y a deux indicateurs permettant de connaître l'état d'une séquence () ainsi que la fin d'une séquence () . Puisque la fréquence de la notification de l'état d'une séquence du contrôleur vers l'ordinateur principal est de 1 Hz, l'indicateur de l'état n'est pas significatif pour une séquence possédant des signaux de durée inférieure à une seconde.

4.6.3 La configuration des séquenceurs

La configuration des séquenceurs a été abordée à la section 3.6 de la page 38 et est réalisée à l'aide de la fenêtre de configuration de la figure 4-11 de la page suivante. Cette fenêtre possède un nouveau champ permettant d'assigner un nom à chacun des séquenceurs.

4.6.4 Discussion

La réalisation de l'application *Générateur de séquences (x86)* a été relativement simple, car elle a été produite en clonant l'application *Contrôleur de moteurs (x86)* tout en modifiant les interfaces et les commandes.

4.7 Les « librairies » dynamiques et les pilotes de la carte d'acquisition

Comme il est mentionné dans l'introduction, l'utilisation d'une « librairie » dynamique favorise l'aspect modulaire du montage. Cependant, la réalisation des « librairies » doit être faite en conséquence. Ainsi, les « librairies » doivent toutes avoir le même ensemble de fonctions. Une « librairie » peut toutefois posséder d'autres fonctions, mais ces dernières ne seront accessibles qu'à l'intérieur de celle-ci.

La communication avec la carte d'acquisition se fait à l'aide d'un pilote, car Windows NT interdit aux applications et « librairies » standard de communiquer avec les parties matérielles de l'ordinateur.

4.7.1 La « librairie » dynamique

La « librairie » dynamique possède plusieurs fonctions subdivisées en deux catégories. La première regroupe les fonctions permettant de modifier la configuration de la carte d'acquisition, et la seconde contient les fonctions utilisées pour générer les opérations de la

The screenshot shows a dialog box titled "Configuration des séquenceurs". It contains the following fields and options:

- Séquenceur #**: A dropdown menu showing "1".
- Nom**: A text field containing "Trajet #1".
- Nombre de répétitions**: A text field containing "0".
- Nombre d'événements**: A dropdown menu showing "1".
- Numéro du séquenceur de démarrage**: A dropdown menu showing "1".
- Démarre sur une transition vers**: Radio buttons for "le haut" (selected) and "le bas".
- Événement #**: A dropdown menu showing "0".
- Polarité**: Radio buttons for "Positive" and "Négative" (selected).
- Durées**: Two text fields: "État 0 : 798.000 ms" and "État 1 : 200.000 ms".
- Délai avant démarreur externe**: Two text fields: "État 0 : 788.000 ms" (checked) and "État 1 : 0.025 ms" (unchecked).
- Buttons**: "Accepte" (with a green checkmark) and "Annule" (with a red X).

Figure 4-11 : Fenêtre de configuration des séquenceurs de l'application *Générateur de séquences* (x86).

carte d'acquisition.

4.7.1.1 La configuration de la carte d'acquisition

Le système de configuration de la « librairie » doit tenir compte du fait que chaque carte d'acquisition fonctionne différemment et que la configuration de chacune des cartes est différente. Cependant, il doit y avoir une partie de la configuration qui est la même pour chacune des cartes. Ces configurations seront respectivement nommées la configuration privée et la configuration commune.

La « librairie » possède donc un premier ensemble de fonctions permettant de modifier la configuration commune. Parmi ces fonctions, il y en a certaines qui sont utilisées pour changer la configuration de la prochaine demande d'acquisition et d'autres qui servent à connaître les facteurs de conversion appliqués aux données acquises pour que les valeurs des données soient exprimées en valeurs réelles. Ces facteurs de conversion permettent de conserver les données dans des entiers, ce qui nécessite moins d'espace mémoire, tout en connaissant les valeurs réelles des données. Les facteurs de conversion sont essentiellement une origine et un facteur d'échelle. Les deux équations suivantes permettent de convertir une valeur entière en une valeur réelle et vice versa :

$$\text{ValeurRéelle} = (\text{ValeurEntière} - \text{Origine}) * \text{FacteurÉchelle}$$

$$\text{ValeurEntière} = \text{ValeurRéelle} / \text{FacteurÉchelle} + \text{Origine}$$

La modification des configurations privées se fait par l'appel d'une fonction affichant une boîte de configuration. Par la suite, l'application utilise une autre fonction lui permettant de récupérer la nouvelle configuration sous forme d'un bloc de données. Connaissant la longueur du bloc de données, l'application peut le sauvegarder ou le charger plus tard pour remettre dans le même état la configuration de la carte d'acquisition.

Puisque le bloc de données diffère d'une carte d'acquisition à l'autre, l'application a accès à une autre fonction lui permettant de connaître le nom de la carte d'acquisition. Si jamais

l'utilisateur change de carte d'acquisition, l'application reconnaîtra que la carte d'acquisition a été changée. Le cas échéant, l'application doit considérer que le bloc de données qu'elle possède est invalide et qu'elle doit demander à l'utilisateur de refaire la configuration pour la nouvelle carte d'acquisition. Il n'est cependant pas possible de refilet un faux bloc de données à la « librairie », car chaque bloc de données contient un en-tête l'identifiant et cet en-tête est vérifié chaque fois que le bloc de données est envoyé à la « librairie ».

Finalement, le bloc de données contient aussi les configurations communes. Cela permet à une application de modifier toutes les configurations de la carte d'acquisition en effectuant une seule commande.

4.7.1.2 Les opérations de la carte d'acquisition

Les opérations sont des fonctions permettant d'utiliser la carte d'acquisition proprement dite. Il y a les opérations permettant d'utiliser les ports de la carte d'acquisition, les opérations assurant l'acquisition d'une ou de plusieurs courbes à l'aide des convertisseurs analogiques numériques et les opérations permettant de générer une ou plusieurs courbes sur les convertisseurs numériques analogiques de la carte.

4.7.2 Le pilote de bas niveau

Contrairement aux anciens systèmes d'exploitation comme DOS et Windows 3.1, il n'est pas possible d'accéder directement aux ports, à la mémoire, aux DMA et aux interruptions en utilisant les fonctions standard comme inport et outport. Cette restriction permet d'éviter qu'il y ait des conflits entre les périphériques ou des accès par erreur aux périphériques de l'ordinateur. Pour gérer et protéger tous les périphériques de l'ordinateur, Windows NT impose la construction d'un pilote très spécial qui possède un ensemble de fonctions permettant d'accéder aux ports, à la mémoire, aux DMA et aux interruptions de l'ordinateur. De plus, le pilote doit être construit selon des règles très précises. L'accès au pilote se fera par l'intermédiaire d'un ensemble de fonctions spécialement réservées à cet effet.

Pour éviter qu'il y ait un conflit entre différentes cartes d'acquisition, le pilote doit réserver les ressources qu'il désire utiliser comme l'espace mémoire, les ports, les DMA ou les

interruptions. Lors de la réservation, il y a une vérification qui est faite pour savoir si les ressources convoitées sont déjà utilisées. Si c'est le cas, le pilote ne peut pas utiliser ces ressources et dans le cas contraire, les ressources sont réservées, et aucun autre pilote ne peut les utiliser. C'est donc le premier pilote à être exécuté qui a accès aux ressources demandées. La réservation est inscrite dans le *Registry* de Windows NT et est accessible grâce à l'application *Windows NT diagnostics*. Il est donc facile de connaître les ports, DMA et interruptions utilisés par le système.

Puisqu'il s'agit d'un pilote système, la communication avec le pilote se fait à l'aide des fonctions standard de création, d'écriture et de lecture de fichier. Lors de la création, on utilisera la fonction *CreateFile* en utilisant comme nom de fichier le nom du pilote. La lecture d'une courbe sur un convertisseur analogique numérique se fait en utilisant la fonction *ReadFile*, et l'écriture d'une courbe sur un convertisseur numérique analogique est assurée par la fonction *WriteFile*. Il y a aussi la fonction *DeviceIoControl* qui est utilisée pour changer la configuration de la carte d'acquisition et accéder aux ports de la carte. Puisqu'il y a une seule fonction disponible pour accéder aux registres de la carte d'acquisition, il faut construire un langage de communication entre le pilote et la « librairie » en utilisant la fonction *DeviceIoControl* comme canal de communication. La fermeture de la communication se fait à l'aide de la fonction *CloseHandle*.

L'utilisation des pilotes a comme inconvénient de compliquer légèrement l'accès aux registres de la carte d'acquisition. Cependant, elle diminue grandement la difficulté d'utilisation des DMA et interruptions. L'accès aux DMA réside dans l'appel d'une fonction système très simple d'utilisation contrairement à la programmation des registres du contrôleur DMA qu'il fallait faire pour les autres systèmes d'exploitation.

4.7.3 Discussion

Enfin, il y a deux ensembles de « libraires » et pilotes qui ont été réalisés. Le premier ensemble permet l'accès à la carte d'acquisition DT2801-A, et le second ensemble à la carte d'acquisition DAS16. Il est possible de mettre les deux cartes d'acquisition simultanément sur le même ordinateur et d'en faire fonctionner les pilotes si chacune des cartes d'acquisition

utilise des ressources différentes de l'autre carte d'acquisition. Cependant, il n'est pas possible de faire fonctionner les deux « librairies » dynamiques simultanément. Cette impossibilité vient du fait que les « librairies » ont exactement le même nom et qu'il n'est pas possible d'avoir deux « librairies » dynamiques différentes ayant le même nom et fonctionnant simultanément.

L'utilisation du même nom pour les deux « librairies » est importante car, lors de l'exécution de l'application *Acquisition et traitement des courbes*, il y a une recherche qui est faite sur le disque dur de l'ordinateur pour trouver la « librairie » dynamique ayant comme nom *CarteAcquisition*. Cependant, il serait possible de générer deux « librairies » ayant des noms différents et de modifier l'application *Acquisition et traitement des courbes* pour qu'elle fonctionne avec les deux noms. Par contre, cela compliquerait énormément l'application, et l'emploi simultané de deux cartes d'acquisition n'est pas nécessaire pour réaliser les objectifs du montage.

Les « librairies » et pilotes ont été construits dans le but de supporter une seule carte d'acquisition de même type à la fois. Il n'est donc pas possible de faire fonctionner deux cartes identiques sur le même ordinateur.

4.8 L'application *Acquisition et traitement des courbes*

Le but ultime lors de la réalisation d'une expérience est d'obtenir les réponses électriques de la préparation soumise à différents stimuli électriques ou lumineux. Par la suite, l'analyse de ces résultats permet de valider ou non la théorie reliée à l'expérience. L'acquisition des réponses électriques représente donc la tâche principale de la troisième application de contrôle. Cependant, le rôle de cette application dépasse largement la simple acquisition d'une courbe et comprend toutes les fonctionnalités suivantes :

- un système d'acquisition permettant une synchronisation parfaite entre les stimuli électriques et lumineux. Les courbes acquises sont automatiquement nommées et sauvegardées après leur acquisition sans l'intervention de l'utilisateur. Le nom des courbes est formé de la date et de l'heure de l'acquisition, et elles peuvent être

sauvegardées en mémoire ou directement sur le disque dur de l'ordinateur. Un calepin est associé à chacune des courbes et permet à l'utilisateur de sauvegarder un commentaire avec la courbe. Il est possible de faire l'acquisition de l'état des stimuli électriques et lumineux en même temps que la réponse convoitée.

- une interface graphique permettant de visualiser les courbes acquises. Cette interface permet de superposer de différentes façons plusieurs courbes pour les comparer facilement.
- un système de traitement des courbes permettant à l'utilisateur d'aligner ou de modifier les courbes acquises. Ce système permet aussi d'obtenir des informations comme le maximum, le minimum, etc. de la courbe. Dans le cas où un traitement n'est pas supporté par l'application, il est possible de transférer la courbe à une autre application à l'aide du presse-papiers de Windows NT.
- un serveur DDE permettant à une application de type client d'effectuer des acquisitions et des traitements. Le client peut aussi récupérer les résultats des acquisitions et des traitements.

La figure 4-12 de la page suivante présente l'organigramme de l'application *Acquisition et traitement des courbes*. Les modules principaux de l'application y sont présentés et sont décrits dans les sections 4.8.1 à 4.8.5.

4.8.1 Les projets

Afin de permettre à l'utilisateur d'avoir plusieurs configurations d'acquisition et de passer d'une à l'autre rapidement, les courbes sont regroupées sous forme de projets. Le projet est une structure contenant la configuration de l'acquisition ainsi que les courbes acquises. Lors de l'acquisition, l'utilisateur doit choisir un projet, et ce choix dicte la configuration à utiliser pour l'acquisition ainsi que l'endroit où sauvegarder les courbes acquises. Cependant, par la suite il est possible de transférer, en respectant certaines contraintes, les courbes d'un projet à l'autre.

Afin d'avoir un accès rapide et facile aux courbes, une interface de type *Multiple Document Interface (MDI)* a été construite. Le MDI est géré par Windows NT et permet d'avoir plusieurs fenêtres regroupées dans un même espace et les fenêtres de même type partageant le même code tout en possédant des données différentes. La figure 4-15 de la page 110 présente la fenêtre principale de l'application qui est de type MDI. À l'intérieur de la fenêtre principale, il y a deux fenêtres de type projet, une fenêtre de type graphique et une fenêtre de type propriété. Le nombre de fenêtres est illimité. Par contre, le nombre de fenêtres de projets est limité par le nombre de projets que l'utilisateur créera. Pour des raisons pratiques, il n'est pas possible d'avoir plus d'une fenêtre de traitement. Cependant, les fenêtres de type graphique et de type propriété peuvent exister en nombre illimité.

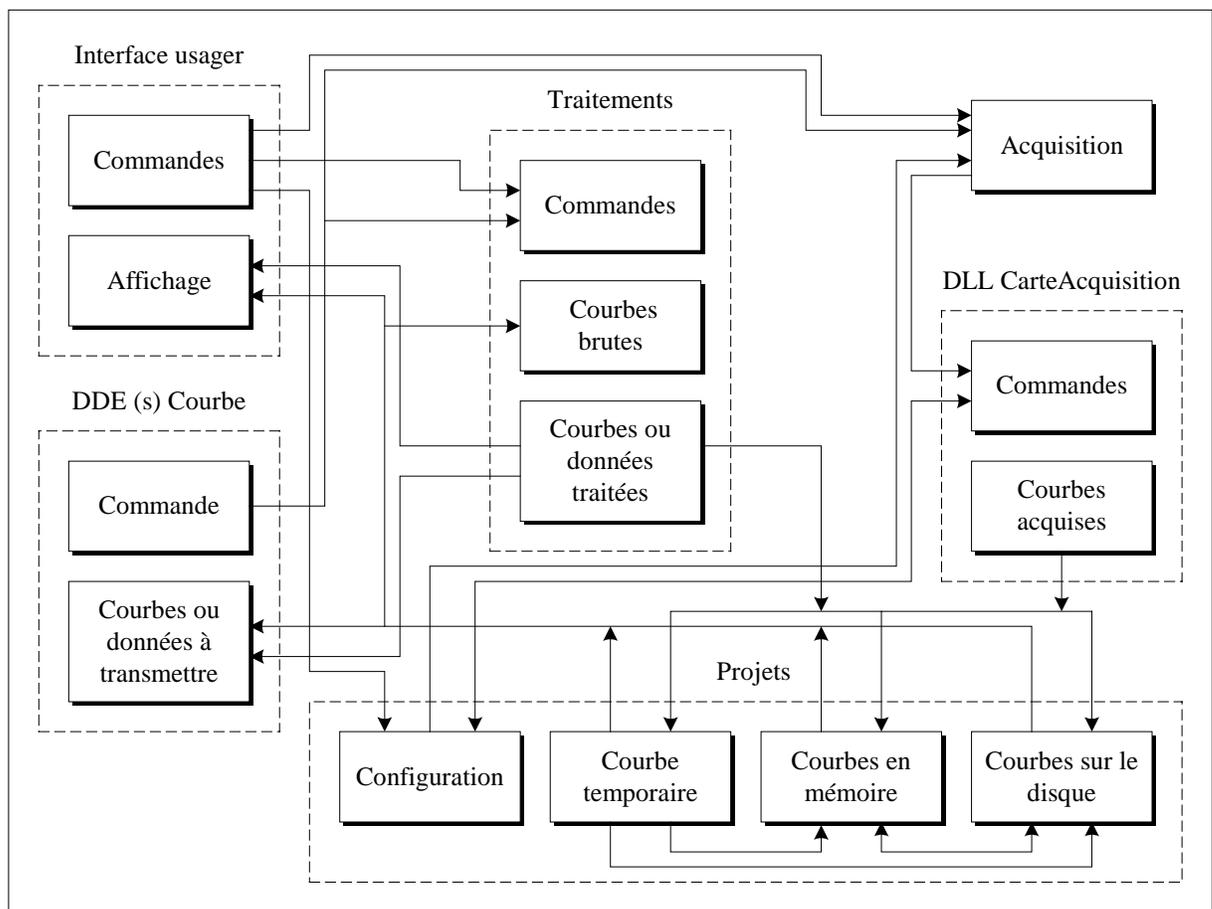


Figure 4-12 : Organigramme de l'application *Acquisition et traitement des courbes*.

Pour mieux comprendre l'organisation des courbes d'un projet, regardons la figure 4-13 qui présente une fenêtre de projet qui contient l'arborescence donnant accès à toutes les courbes d'un projet. Le premier niveau de l'arborescence est simplement le nom du projet, que l'on retrouve aussi dans le titre de la fenêtre.

Le second niveau représente les différentes façons dont les courbes sont sauvegardées dans le projet. Il y a essentiellement trois façons de sauvegarder les courbes. L'item *temporaire* permet de conserver une courbe jusqu'à l'acquisition suivante ou au prochain traitement. Pour conserver une courbe plus longtemps, l'item *mémoire* est utilisé et conserve les courbes jusqu'à la fermeture de l'application. Si l'on désire conserver les courbes de façon indéfinie, il suffit d'utiliser l'item *disque* et les courbes seront sauvegardées automatiquement sur le disque dur de l'ordinateur.

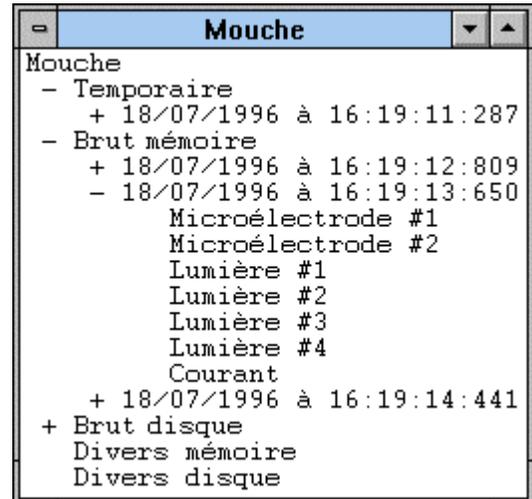


Figure 4-13 : Fenêtre de projet de l'application *Acquisition et traitements des courbes*.

Afin d'établir la différence entre les courbes originales et les courbes émanant d'un traitement mathématique, le projet possède deux classes d'items. Les items ayant comme préfixe *brut* sont utilisés pour sauvegarder les courbes générées lors de l'acquisition, et les items ayant comme préfixe *divers*, les courbes provenant d'un traitement mathématique. De son côté, l'item *temporaire* reçoit les courbes provenant des deux sources.

Le troisième niveau représente chacune des acquisitions qui ont été faites, et le dernier niveau les courbes acquises simultanément lors d'une acquisition. Le nom des items du troisième niveau est construit à partir de la date et de l'heure de l'acquisition, et le nom des items du dernier niveau indique le nom de chacune des courbes. Le choix du nom des courbes est fait dans la fenêtre de configuration du projet (voir section 4.8.2, page 106).

Les signes + et - indiquent qu'un item de l'arborescence peut être respectivement déployé ou refermé en appuyant sur le signe avec la souris. Les items n'ayant pas de signe n'ont pas de sous-items.

Finalement, la configuration d'acquisition présente dans le projet est modifiable par l'utilisateur. Si l'on désire travailler avec deux configurations simultanément, il suffit de créer deux projets qui utilisent chacun une des deux configurations. Lors de l'acquisition, le choix du projet dicte le choix de la configuration.

4.8.2 L'acquisition

Le fonctionnement de l'acquisition est très simple (voir figure 4-12, page 104). L'acquisition s'active à l'aide d'une commande provenant de l'utilisateur ou de l'application de type client. Lors de la demande d'acquisition, le nom du projet et la destination de la courbe acquise doivent être inscrits. Connaissant le projet, le module d'acquisition fixe la configuration de la carte d'acquisition selon la configuration du projet. Lorsque l'acquisition est terminée, la courbe est envoyée dans l'item *temporaire*, *brut mémoire* ou *brut disque* selon la destination choisie.

Si la courbe est envoyée dans l'item *temporaire*, la courbe déjà présente dans l'item est détruite pour faire place à la nouvelle. Pour ce qui est des deux autres items, la courbe est sauvegardée à la suite des autres courbes et immédiatement sauvegardée sur le disque dur pour l'item *brut disque*.

La dernière étape de l'acquisition consiste à avertir l'utilisateur et l'application de type client de la fin de l'acquisition. Cet avertissement est important, car il permet à l'application de tourner librement durant l'acquisition qui peut durer plusieurs minutes. En réalité, la durée de l'acquisition dépend de la fréquence d'échantillonnage, du nombre d'échantillons et du départ réel de l'acquisition. Le départ réel de l'acquisition est initié par la gâchette de la carte d'acquisition qui est activée par le contrôleur *Générateur de séquences*.

Configuration du projet

Canal de départ : <input type="text" value="0"/>	Canal : <input type="text" value="2"/>
Canal de fin : <input type="text" value="3"/>	Nom : <input type="text" value="Lumière"/>
<input checked="" type="checkbox"/> Gâchette	Unité : <input type="text"/>
<input type="checkbox"/> Température	Gain : <input type="text" value="1.000"/>
Canal : <input type="text"/>	Graphique : <input type="text" value="1G"/>
Gain : <input type="text"/>	Type canal : <input type="text" value="Multiplexé"/>
<input type="checkbox"/> Effectuer la moyenne	Plage de tension : <input type="text" value="5.000"/> V
Nombre d'itérations : <input type="text"/>	Nombre de signaux : <input type="text" value="4"/>
Fréquence : <input type="text" value="24000"/> Hz	
Nombre d'échantillons : <input type="text" value="12000"/>	
<p style="text-align: center;">Pour chaque courbe</p> Fréquence : <input type="text" value="6000"/> Hz	
Nombre d'échantillons : <input type="text" value="3000"/>	
<input type="button" value="✓ Accepte"/>	<input type="button" value="✗ Annule"/>

Figure 4-14 : Fenêtre de configuration d'un projet de l'application *Acquisition et traitement des courbes*.

La configuration d'un projet permet de choisir les signaux à acquérir et leurs caractéristiques. La figure 4-14 présente la fenêtre utilisée pour changer la configuration d'un projet qui comprend les options suivantes :

- La première option permet de choisir les canaux utilisés lors de l'acquisition. Il s'agit d'indiquer le canal de départ et le canal de fin. Lors de l'acquisition, les canaux sont balayés successivement pour générer le même nombre de courbes que de canaux.

- La seconde option permet à la carte d'acquisition d'utiliser une ligne de synchronisation externe (gâchette) pour démarrer l'acquisition. Dans un tel cas, l'application amorcera l'acquisition, mais la véritable acquisition débute seulement quand la gâchette de la carte d'acquisition est activée.
- La troisième option est utilisée pour lire la température de la préparation et la sauvegarder avec la courbe acquise. Il est alors nécessaire de spécifier le canal à utiliser pour la lecture de la température et le facteur de conversion qu'il faut appliquer à la valeur lue pour exprimer la température en degré Celsius.
- La quatrième option offre la possibilité de faire la moyenne d'une séquence de courbes dès leur acquisition. Si c'est le cas, on doit indiquer le nombre de courbes utilisées pour calculer la moyenne. Les courbes sont acquises et sommées les unes avec les autres. Lorsque la dernière courbe est acquise, les courbes sommées sont divisées par le nombre de courbes acquises et seulement la courbe résultante est sauvegardée dans le projet.
- Les cinquième et sixième options permettent de connaître la fréquence d'échantillonnage et le nombre de points utilisés pour l'acquisition. Le changement de ces paramètres peut être fait en utilisant la configuration de la carte d'acquisition, un sujet qui n'est pas traité dans la présente thèse.
- La dernière option, illustrée dans la partie droite de la fenêtre, permet d'adapter chacun des canaux au signal qu'il représente. En premier lieu, il y a le nom qui assure l'identification du canal dans la fenêtre du projet. Par la suite, il y a l'unité utilisée lors de l'affichage de la courbe et un paramètre de gain. Ce gain est nécessaire, car certains signaux sont amplifiés avant d'être présentés à la carte d'acquisition. Si c'est le cas, on indique le gain de l'amplificateur, et les tensions acquises sont traitées pour que les tensions avant amplification soient considérées par l'application. Le paramètre suivant permet d'assigner la courbe à une section d'un graphique. À titre d'exemple, il est possible de choisir entre la section du haut et la section du bas du graphique *Grappe #1* (voir figure 4-15, page 110). Les trois derniers paramètres sont utilisés pour indiquer le

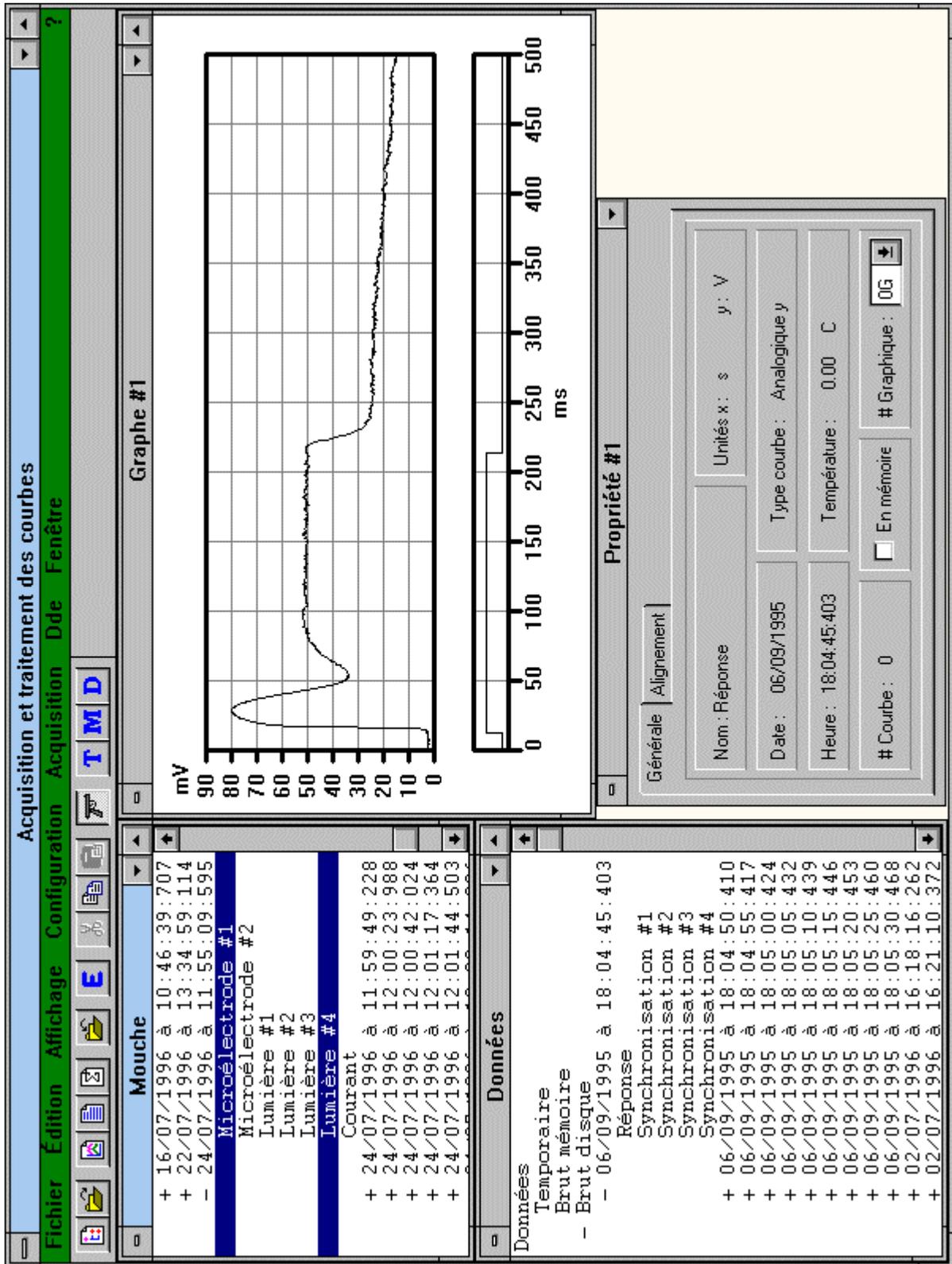
type de signal présenté au canal de la carte d'acquisition. Il y a les signaux de type analogique et ceux de type numérique. Un signal de type numérique est utilisé lorsque des signaux numériques ont été préalablement encodés à l'aide du circuit de synchronisation (voir section 3.12, page 64). Le cas échéant, il faut indiquer le nombre de signaux encodés ainsi que la tension maximale utilisée par l'encodeur. Le signal de type analogique est utilisé pour tous les autres signaux analogiques.

4.8.3 L'interface usager

L'interface usager possède plusieurs contrôles utilisés à différentes fins. Cependant, seule la description de la technique utilisée par l'application pour sélectionner les courbes à afficher ou à traiter est abordée ici.

Puisqu'il s'agit d'une interface MDI qui peut posséder plusieurs projets, fenêtres graphiques, fenêtres de propriétés et une fenêtre de traitement, la sélection des courbes est faite en cliquant sur les courbes que l'on désire dans les projets. La sélection est faite pour la dernière fenêtre à être activée sans tenir compte des fenêtres de projets. Les items en inverse vidéo dans les fenêtres des projets représentent la sélection de la dernière fenêtre à être activée. De plus, la sélection des projets est mise à jour chaque fois que l'on active une autre fenêtre pour refléter la sélection associée à cette fenêtre.

À titre d'exemple, la figure 4-15 de la page 110 présente un graphique affichant deux courbes provenant du projet *Mouche* et une fenêtre de propriétés affichant les propriétés d'une courbe du projet *Données*. Dans cet exemple, la sélection des projets représente la sélection de la fenêtre graphique, car c'est la dernière fenêtre à avoir été activée.

Figure 4-15 : Fenêtre principale de l'application *Acquisition et traitement des courbes*.

4.8.4 Les traitements

Les traitements supportés par l'application sont, pour l'instant, très primaires. Cependant, la structure utilisée permet à un programmeur expérimenté sous Windows NT d'ajouter de nouveaux traitements lorsque le besoin s'en fera sentir. Les traitements déjà implantés sont utilisés pour aligner les courbes sur un graphique afin de faciliter la

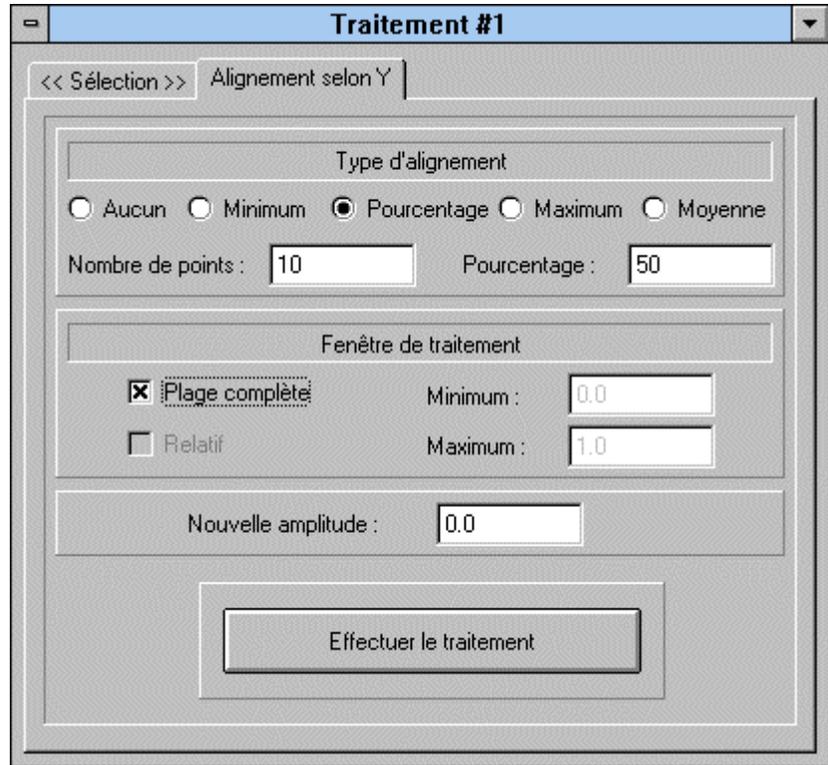


Figure 4-16 : Fenêtre de traitement de l'application *Acquisition et traitement des courbes*.

Il y a aussi les traitements

nécessaires à l'application de type client pour prendre des décisions sur la progression de l'expérience. Il n'est pas nécessaire d'implanter des traitements qui ne sont pas utilisés lors d'une expérience, car il est toujours possible de transférer les courbes dans une autre application comme Excel et Matlab en utilisant le presse-papiers de Windows NT.

Le traitement d'une courbe est fait à l'aide de la fenêtre de traitement (voir figure 4-16) qui permet de choisir les courbes que l'on désire traiter, le traitement à effectuer ainsi que le projet de destination du résultat. Si le résultat n'est pas une courbe, il sera directement affiché à l'écran au moyen d'une boîte de dialogue. La sélection des courbes à traiter est faite de la même façon que la sélection des courbes à afficher (voir section 4.8.3, page 109).

L'application est aussi munie d'un système d'historiques permettant de connaître tous les traitements qui ont été effectués sur une courbe depuis son acquisition. Cependant, cette fenêtre ne conserve que les noms des courbes antérieures ainsi que les valeurs des paramètres

des traitements qui ont été effectués. Il n'est donc pas possible de retrouver les courbes originales si elles ont été perdues et qu'il s'agissait d'un traitement non linéaire.

L'application possède aussi une fenêtre de propriétés permettant de connaître les différentes propriétés des projets et des courbes. La figure 4-15 de la page 110 présente cette fenêtre de propriétés qui affiche les propriétés de la courbe *réponse* du groupe de courbes *06/09/1995 à 18:04:45:403*.

4.8.5 La communication DDE

Tout comme les autres applications de contrôle, la présente application possède plusieurs commandes accessibles à une application de type client. Ces commandes permettent de faire des acquisitions et, par la suite, de traiter les courbes acquises dans le but de diriger la progression d'une expérience.

Tout comme l'interface usager, le client est avisé de la fin d'une acquisition. Le message de fin d'acquisition est accompagné du nom de la courbe qui peut être utilisé ultérieurement par le client pour effectuer un traitement sur la courbe ou récupérer la courbe.

Lorsque le client effectue un traitement générant une nouvelle courbe, cette dernière est ajoutée dans un projet, comme si le traitement avait été fait par l'utilisateur.

4.8.6 Discussion

La réalisation de la présente application ne peut nous être attribuée en pleine partie. Une première version de cette application a été réalisée dans le cadre d'un projet de maîtrise (Moszkowicz, 1994). Nous collaborions étroitement au travail, qui consistait à réaliser la présente application pour Windows 3.1, le système d'exploitation de l'époque. Par la suite, quand l'ensemble du projet progressait, la nécessité d'un nouveau système d'exploitation et de nouvelles options, ainsi que de nouveaux traitements mathématiques, a demandé une modification majeure de l'application. Cette reconstruction a permis de profiter des erreurs du passé pour construire une meilleure interface et utiliser des techniques de programmation plus récentes comme le langage C++ et la « librairie » *Microsoft Fondation Classe (MFC)*.

Même s'il y a eu plusieurs versions de la présente application, il y a de nombreuses modifications qui restent à apporter pour augmenter la puissance de l'application. On peut penser à de nouveaux traitements plus complets ainsi qu'à des algorithmes spécialisés qui permettraient de comparer directement un modèle théorique avec les données expérimentales.

4.9 Les « librairies » dynamiques et les pilotes de la carte vidéo

La structure des « librairies » et pilotes de la carte vidéo est sensiblement la même que celle des « librairies » et pilotes de la carte d'acquisition. La seule différence réside dans les fonctions utilisées. Le montage est conçu pour supporter deux types de cartes vidéo. Il y a la carte vidéo XICAS et le contrôleur STAR I. La carte vidéo XICAS est directement installée dans l'ordinateur, et la communication avec la carte se fait comme pour les cartes d'acquisition. Cependant, le contrôleur STAR I utilise un contrôleur GPIB pour communiquer avec l'ordinateur. Le contrôleur GPIB est accompagné d'une « librairie » dynamique et d'un pilote permettant de gérer le contrôleur STAR I avec des commandes GPIB. Le travail s'est donc limité à la création d'une « librairie » dynamique ayant les mêmes points d'entrée que celle de la carte vidéo XICAS, sauf qu'elle générerait les commandes nécessaires à la « librairie » dynamique fournie avec la carte GPIB.

L'utilisation des « librairies » GPIB montre les possibilités de notre système modulaire. Il est donc possible d'utiliser différents modèles de cartes vidéo tout en étant certain qu'il est toujours possible de construire une « librairie » dynamique et les pilotes nécessaires pour faire un lien direct avec la carte utilisée ou de construire une « librairie » dynamique qui utilise les « librairies » et pilotes fournis avec la carte utilisée.

4.10 L'application *Acquisition et traitement des images*

Cette dernière application de contrôle vient compléter l'ensemble des outils nécessaires au déroulement d'une expérience. Elle se consacre à l'acquisition, à la sauvegarde et au traitement des images captées par la caméra du montage. Sa structure est sensiblement la même que celle de l'application *Acquisition et traitement des courbes* (voir section 4.8, page 102). La seule différence réside dans la manipulation d'images au lieu de la manipulation de

courbes. Puisque l'application ne se limite pas à la simple acquisition d'une image, elle possède toutes les fonctionnalités suivantes :

- Un système d'acquisition permet de faire l'acquisition d'une image en contrôlant le temps d'intégration de la caméra. Les images acquises sont automatiquement nommées et sauvegardées après leur acquisition sans l'intervention de l'utilisateur. Le nom utilisé pour identifier les images acquises est composé à partir de la date et de l'heure de l'acquisition. Les images sont sauvegardées en mémoire ou directement sur le disque dur de l'ordinateur. Un calepin est greffé à chacune des images et sauvegardé avec l'image. Ce calepin permet à l'utilisateur de joindre un commentaire à l'image.
- Une interface graphique permet de visualiser les images. Cette interface possède différents traitements pour améliorer la qualité de la visualisation. L'interface peut aussi présenter

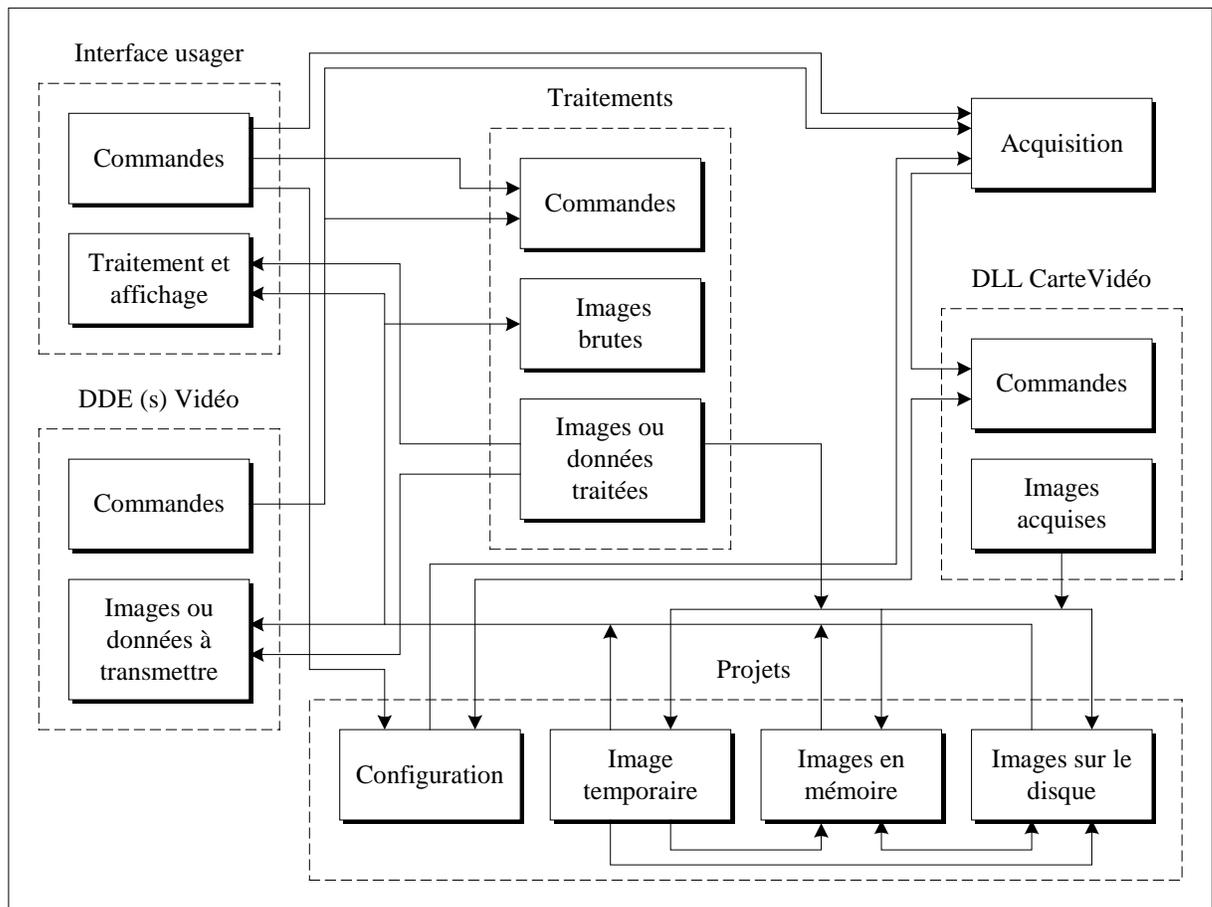


Figure 4-17 : Organigramme de l'application *Acquisition et traitement des images*.

l'image en perspective en trois dimensions, tracer son histogramme et afficher le profil de l'intensité lumineuse d'une ligne passant sur l'image.

- Un système de traitement des images permet d'effectuer divers traitements sur les images. Dans le cas où un traitement n'est pas supporté par l'application, il est possible de transférer l'image à une autre application au moyen du presse-papiers de Windows NT.
- Un serveur DDE permet à un client d'effectuer des acquisitions et des traitements. Le client peut aussi obtenir les résultats des acquisitions et des traitements.

La figure 4-17 de la page précédente présente l'organigramme de l'application *Acquisition et traitement des images*. Puisque l'application est structurée comme celle de *Acquisition et traitement des courbes*, l'organigramme est pratiquement le même que celui de la figure 4-12 de la page 104. L'illustration présente donc les modules principaux de l'application qui sont décrits dans les sections 4.10.1 à 4.10.5.

4.10.1 Le projet

Le projet est semblable au projet de l'application *Acquisition et traitement des courbes* (voir section 4.8.1, page 103). La seule différence réside dans le nombre de niveaux de l'arborescence. Il y a un niveau de moins, car le concept d'acquisition de plusieurs courbes simultanément n'existe pas pour l'acquisition des images. L'arborescence se limite donc à trois niveaux où le premier niveau contient le nom du projet, le second niveau différencie la sauvegarde des images et le dernier niveau contient les images proprement dites (voir figure 4-18).

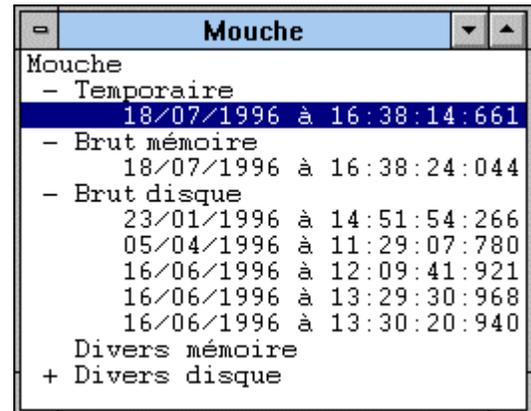


Figure 4-18 : Fenêtre de projet de l'application *Acquisition et traitement des images*.

4.10.2 L'acquisition

Pour bien comprendre comment fonctionne l'acquisition d'une image, il faut savoir que les cartes vidéo possèdent une mémoire tampon et une sortie vidéo permettant d'afficher, sur un moniteur, l'image présente dans la mémoire tampon.

La séquence d'acquisition consiste à demander à la carte vidéo de faire la sauvegarde de l'image dans sa mémoire tampon. Par la suite, on demande de transférer l'image à l'ordinateur en la plaçant dans un des trois items possibles du projet, c'est-à-dire les items *temporaire*, *brut mémoire* ou *brut disque*.

Le système d'acquisition est donc composé de cinq commandes différentes. Une première commande *Exposition* permet de demander à la carte vidéo de transférer l'image de la caméra à la mémoire tampon de la carte d'acquisition. Une série de trois autres commandes est utilisée pour transférer le contenu de la mémoire tampon dans un des trois items du projet. Une dernière série de trois commandes permet de faire une exposition et de transférer immédiatement l'image dans un des trois items du projet. Il serait possible de diminuer le nombre de commandes d'acquisition en n'utilisant que les trois dernières commandes. Cependant, il est souvent utile de visionner l'image du moniteur sans qu'il ne soit nécessaire de transférer l'image dans l'ordinateur.

L'acquisition se termine par une notification de la fin de l'acquisition à l'interface usager et à l'application de type client. L'utilisation de la notification évite d'arrêter l'application durant la durée de l'acquisition qui est fixée par le temps d'intégration. À titre d'exemple, le temps d'intégration de la caméra STAR I peut aller de 0,1 s à 10 000 s. Il est donc possible, durant l'acquisition, d'utiliser toutes les fonctionnalités de l'application, sauf une nouvelle demande d'acquisition.

4.10.3 L'interface usager

Tout comme pour le fonctionnement du reste de l'application, l'interface usager fonctionne exactement de la même façon que celle de l'application *Acquisition et traitement des courbes*

(voir section 4.8.3, page 109). La seule différence réside dans l'affiche d'images au lieu de l'affichage de courbes (voir figure 4-23, page 121).

Puisque les images acquises sont souvent de très faible luminosité et qu'elles n'utilisent pas toute la plage dynamique disponible, elles doivent être traitées avant d'être affichées. Parmi les traitements disponibles, il y a l'algorithme d'égalisation d'histogramme, très connu, qui est utilisé pour certains types d'images. Cependant, l'implantation de ces traitements ne modifie pas les données originales de l'image, car la structure interne du logiciel comprend deux images, soit l'image originale et l'image traitée, qui est seulement utilisée pour la visualisation. L'utilisation d'une structure à deux images évite surtout de refaire le traitement chaque fois que le système demande une mise à jour de la fenêtre de l'image.

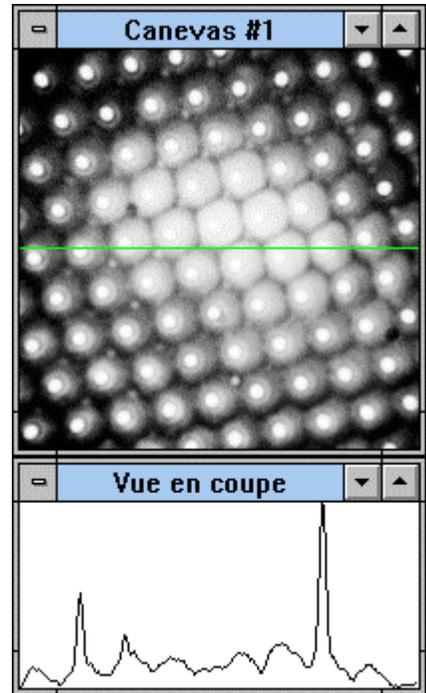


Figure 4-19 : Fenêtre de vue en coupe de l'application *Acquisition et traitement des images*.

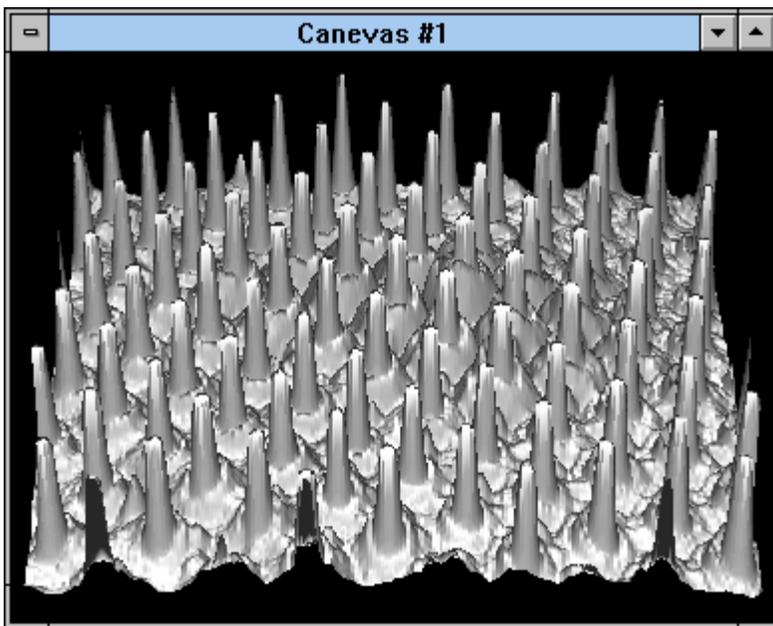


Figure 4-20 : Fenêtre de perspective de l'application *Acquisition et traitement des images*.

En plus de l'affichage normal d'une image, il est possible d'avoir le profil d'intensité d'une ligne passant sur l'image, une vue en perspective de l'image et l'histogramme de l'image.

Le profil d'intensité est présenté à l'aide d'une nouvelle fenêtre, appelée *vue en coupe*, qui est placée sous la fenêtre normale d'affichage de

l'image (voir figure 4-19, page précédente). Lorsque la vue en coupe est activée, une ligne horizontale est dessinée sur la fenêtre normale, et la fenêtre *vue en coupe* présente le profil d'intensité des pixels sous la ligne. L'utilisateur peut par la suite déplacer verticalement la ligne pour voir le profil d'intensité des autres régions de l'image.

La vue en perspective utilise la même fenêtre que la vue normale et présente l'image dans un espace à trois dimensions pour mieux voir son profil lumineux. L'utilisateur peut par la suite choisir le point de vue de l'observateur et de la source lumineuse pour bien visualiser l'image (voir figure 4-20, page précédente).

L'histogramme est aussi présenté dans la même fenêtre que l'image normale (voir figure 4-21). Par la suite, l'utilisateur peut choisir le nombre de catégories de l'histogramme ou voir l'histogramme de l'image originale ou de l'image traitée pour l'affichage.

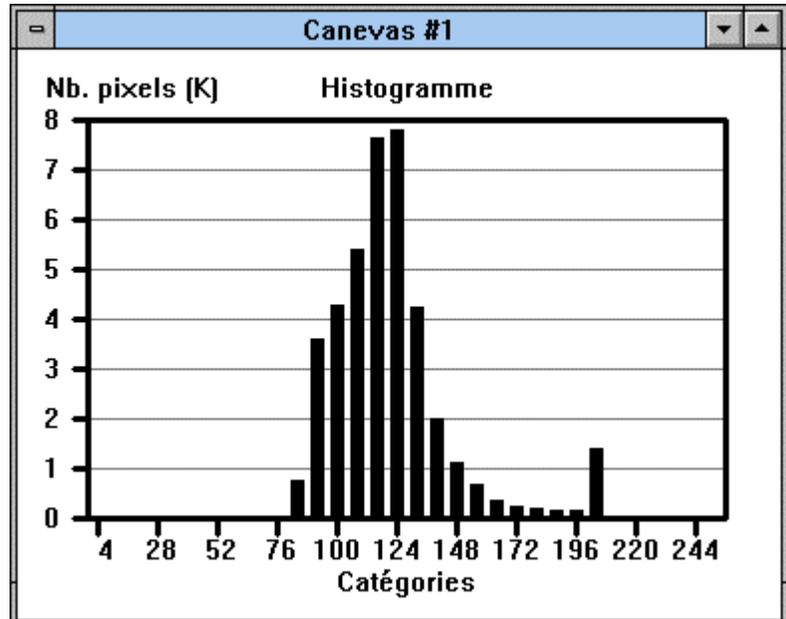


Figure 4-21 : Fenêtre d'histogramme de l'application *Acquisition et traitement des images*.

4.10.4 Les traitements

Les traitements implantés sur la présente application sont, pour l'instant, très primaires. Il s'agit simplement d'une suite d'opérateurs mathématiques permettant d'additionner, de soustraire, de multiplier et de diviser une image avec une constante ou une autre image (voir figure 4-22, page suivante). Cependant, tout comme pour l'application *Acquisition et traitement des courbes*, il est possible, pour un programmeur expérimenté sous Windows NT, d'ajouter de nouveaux traitements. Il est aussi possible de copier l'image dans le presse-papiers. L'image est donc accessible par toutes les applications supportant le presse-papiers et les images de format

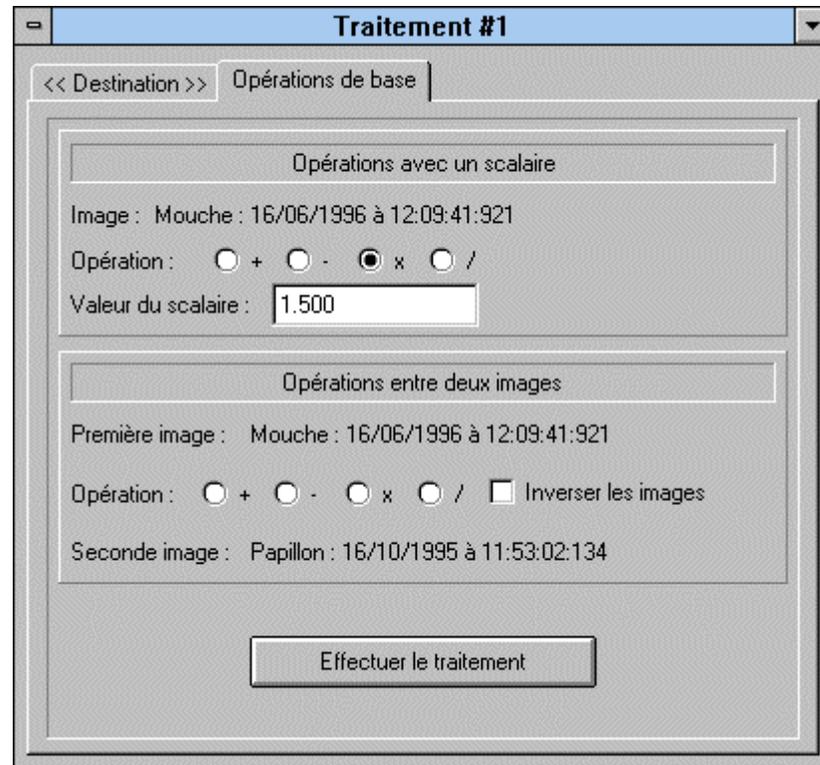


Figure 4-22 : Fenêtre de traitement de l'application *Acquisition et traitement des images*.

BMP. Cela comprend pratiquement toutes les applications travaillant avec des images, car le format BMP est le format standard supporté par le système d'exploitation.

La façon de traiter une image, la génération de l'historique et l'utilisation de la fenêtre de propriétés sont faites de la même manière que pour l'application *Acquisition et traitement des courbes* (voir section 4.8.4, page 111).

4.10.5 La communication DDE

La communication DDE est encore une fois utilisée dans la présente application. Il devient donc possible pour une application de type client de demander l'acquisition d'une image ou d'effectuer un traitement sur une image. Tout comme l'interface usager, l'application de type client est avisée de la fin d'une acquisition d'une image. Elle reçoit du même coup le nom de l'image qu'elle peut utiliser pour effectuer des traitements sur l'image.

4.10.6 Discussion

Tout comme pour l'application *Acquisition et traitement des courbes*, la présente application fait suite à une première version qui avait été réalisée dans le cadre d'un projet de maîtrise (Moszkowicz, 1994). Les raisons de la mise à jour sont surtout dues au changement de système d'exploitation. Cependant, cela a quand même permis d'ajouter plusieurs fonctionnalités à l'application comme les traitements, la vue en perspective, l'histogramme, la vue en coupe, les fenêtres de propriétés, la sélection de différents traitements avant l'affichage de l'image, etc.

La présente application est assez générale et pourrait être utilisée pour simplement faire l'acquisition d'images de toutes sortes. Cependant, il faudrait ajouter les options de sauvegarde pour sauvegarder les images dans des formats connus. Il serait aussi intéressant d'ajouter des traitements de haut niveau pour accélérer le déroulement d'une expérience. Par exemple, il pourrait y avoir un algorithme identifiant automatiquement la position d'une pseudopupille sur l'image ainsi que le centre de chacune des facettes.

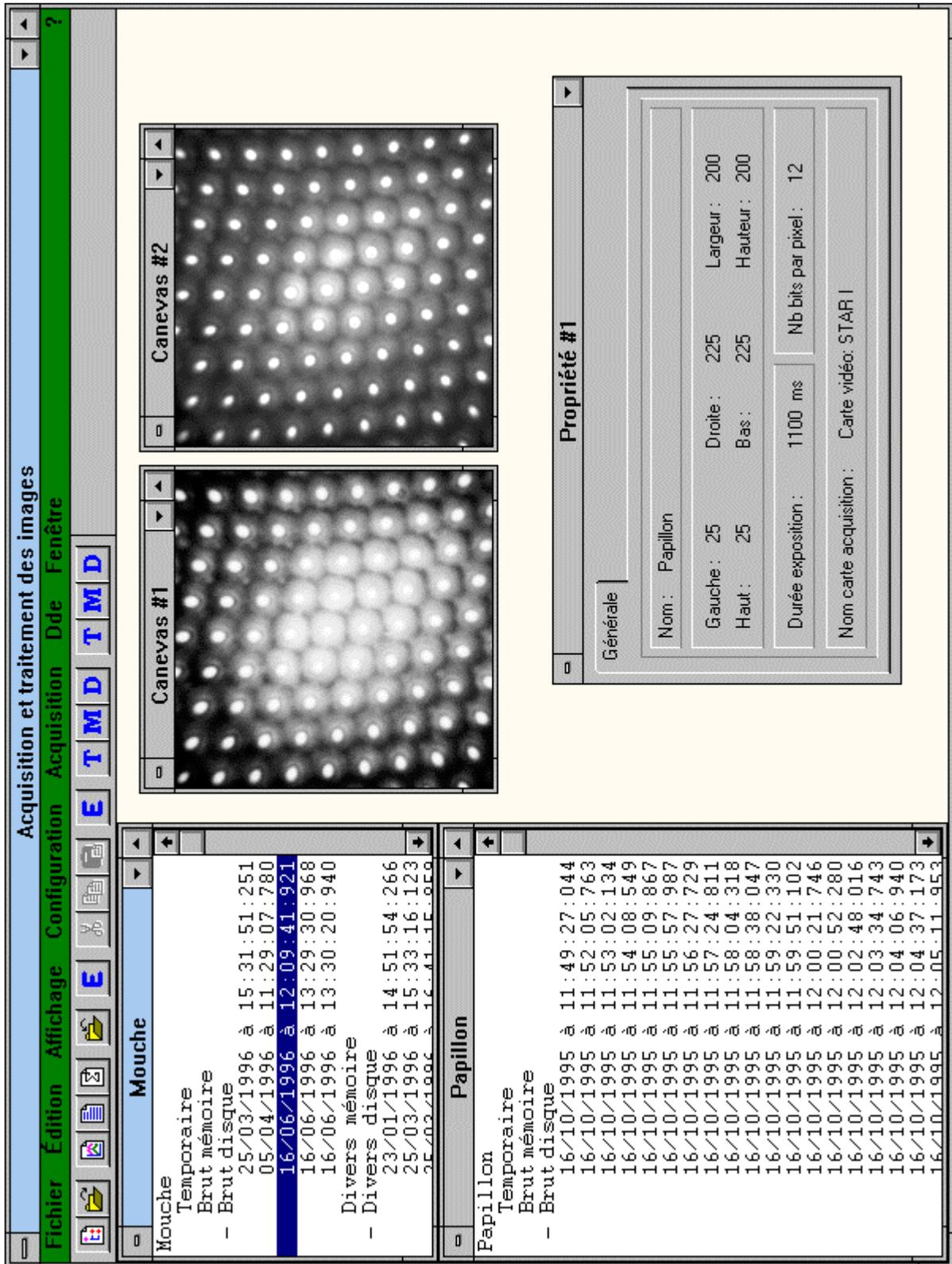


Figure 4-23 : Fenêtre principale de l'application *Acquisition et traitement des images*.

4.11 L'application *Contrôle des expériences*

Afin d'atteindre les objectifs d'automatisation d'une expérience, les applications de contrôle sont insuffisantes, car il n'y a aucune communication entre elles pour générer des tâches complexes. La responsabilité de gérer les expériences revient à une autre application qui communiquera avec les quatre applications de contrôle pour générer des suites de commandes permettant de réaliser une expérience.

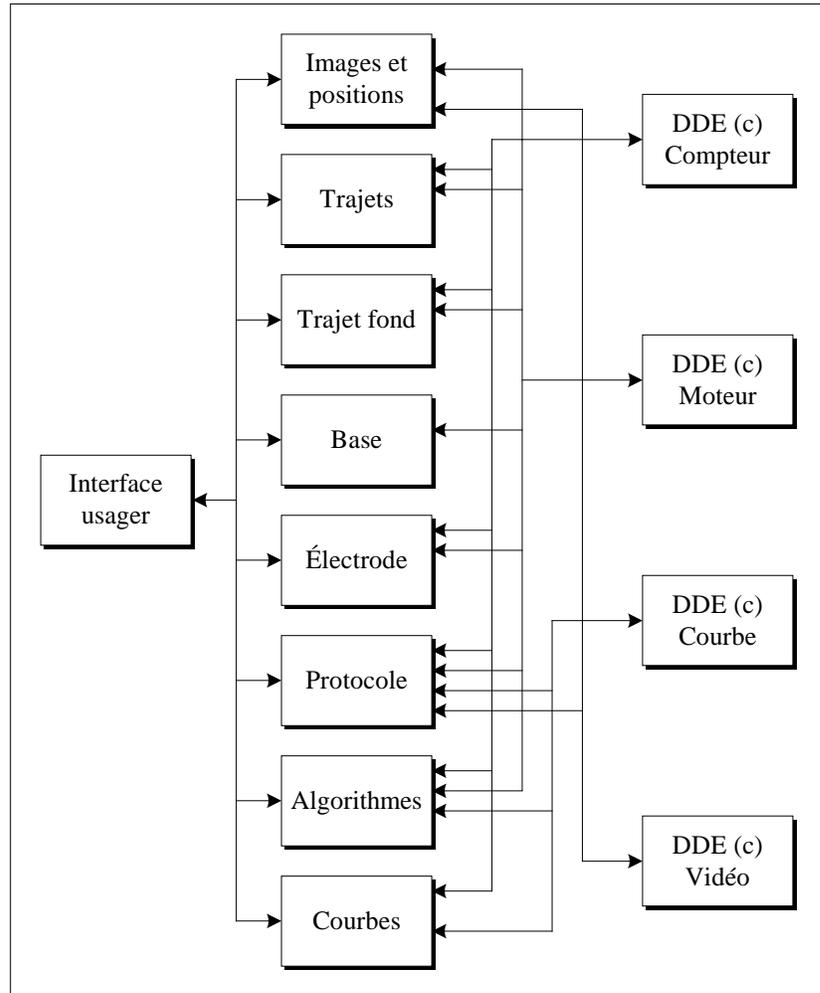


Figure 4-24 : Organigramme de l'application *Contrôle des expériences*.

L'application *Contrôle des expériences* est construite

de façon à refléter les différentes parties du stimulateur optique (voir section 3.3, page 21). Donc, pour chaque partie du stimulateur optique, il y a un module correspondant dans l'application. À titre d'exemple, l'application possède un module responsable de la commande de la base qui contrôle la base du stimulateur lumineux. Chacun de ces modules communique avec les applications de contrôle d'intérêt pour générer les effets voulus, et l'accès au module par l'utilisateur se fait par l'intermédiaire de l'interface usager (voir figure 4-24). Cependant, l'application est munie d'autres modules, comme le module protocole, le module algorithmes et le module courbes, qui utilisent plusieurs parties du stimulateur lumineux simultanément pour accomplir des tâches plus complexes.

4.11.1 L'interface usager

L'interface usager est essentiellement une grande boîte de dialogue subdivisée en plusieurs régions où chacune d'elles représente une partie du stimulateur optique (voir figure 4-25, page 124). Puisque la surface de la boîte de dialogue est limitée, il y a une région de cette boîte qui est interchangeable et qui permet de représenter successivement plusieurs parties du stimulateur lumineux. Cependant, un soin particulier a été accordé pour ne pas représenter des parties du stimulateur optique, dans cette région, qu'il serait intéressant de manipuler simultanément. La région interchangeable permet de manipuler la base (voir figure 4-28, page 128), la microélectrode (voir figure 4-30, page 131) et le protocole expérimental (voir figure 4-33, page 136).

L'interface usager donne aussi accès à une multitude de fenêtres de configuration et de calibration permettant d'ajuster les paramètres des modules pour bien refléter le fonctionnement du stimulateur lumineux. Cependant, puisque la configuration et la calibration ne changent en rien la fonctionnalité de l'application, elles ne seront pas décrites dans la présente thèse.

4.11.2 La communication DDE

Afin de communiquer avec les applications de contrôle, l'application possède quatre clients DDE. Chacun de ces clients permet de communiquer avec les applications *Acquisition et traitement des courbes*, *Contrôleur de moteurs (x86)*, *Générateur de séquences (x86)* et *Acquisition et traitement des images*. Les quatre clients sont complètement indépendants et peuvent fonctionner simultanément. Par contre, pour un client donné, il est impossible d'amorcer une transaction tant que la transaction précédente n'est pas complétée.

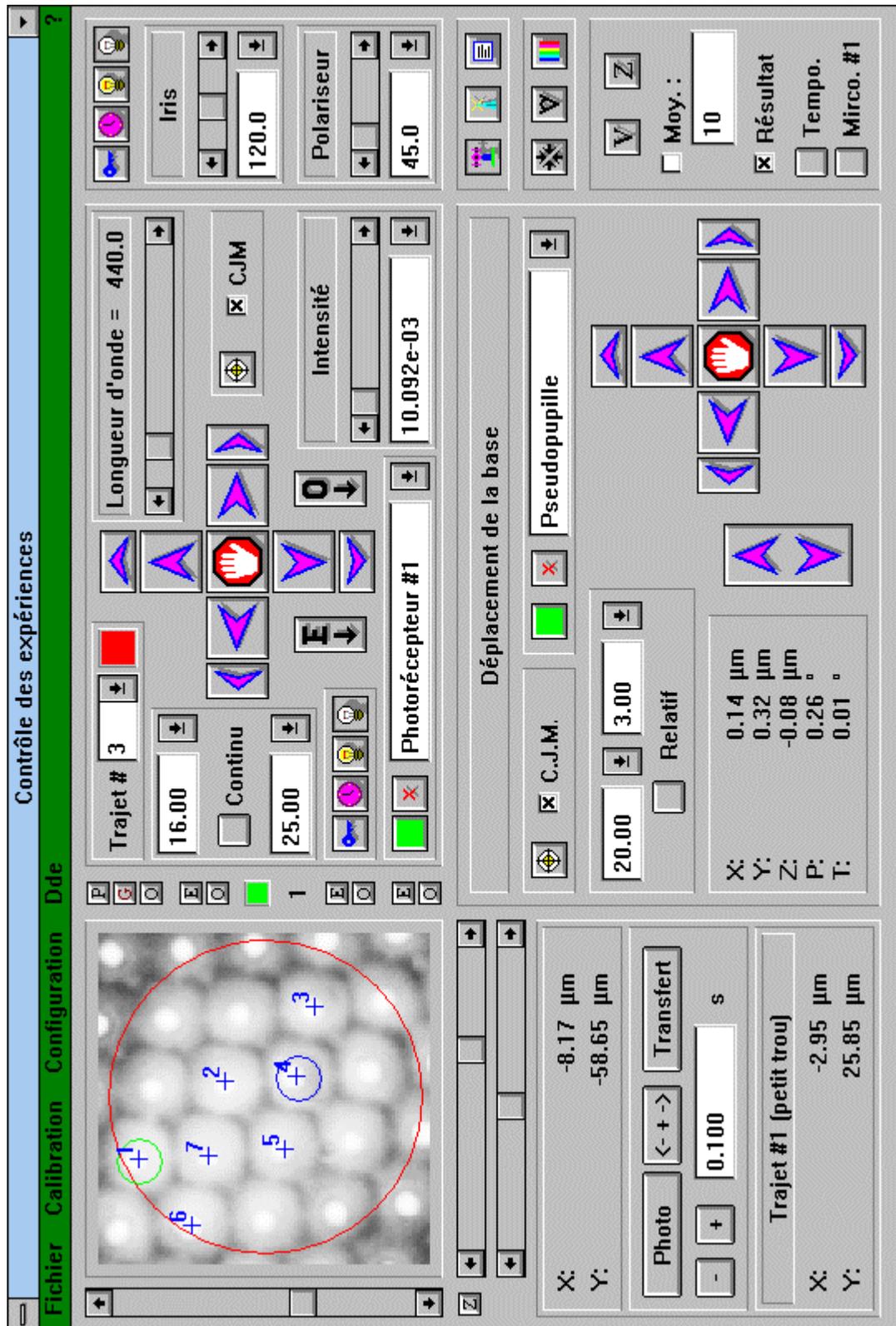


Figure 4-25 : Fenêtre principale de l'application *Contrôle des expériences*.

Pour faciliter l'utilisation des commandes DDE, l'application est munie de quatre ensembles de fonctions, chacun correspondant à un client DDE, qui permettent d'effectuer toutes les commandes DDE possibles. Les différents modules de l'application utiliseront ces fonctions pour communiquer avec les applications de contrôle. La réalisation de ces fonctions permet une validation de tous les paramètres et elle s'assure que la transaction précédente est complétée avant d'en amorcer une seconde. Cela évite d'effectuer des transactions erronées avec les applications de contrôle.

4.11.3 L'image

Afin de bien connaître l'emplacement de la préparation par rapport au centre de l'axe optique du montage, il faut se fier à l'image de la préparation qui est obtenue à l'aide de la caméra du montage (voir section 3.8, page 61). La figure 4-26 présente l'image et l'ensemble des contrôles l'accompagnant. Il y a tout d'abord une série de cinq boutons et une boîte d'édition permettant de commander la prise d'une image et le temps d'intégration de la caméra.

Il y a également un ensemble de trois ascenseurs permettant d'agrandir l'image et de changer la région de l'image à visualiser. Il est

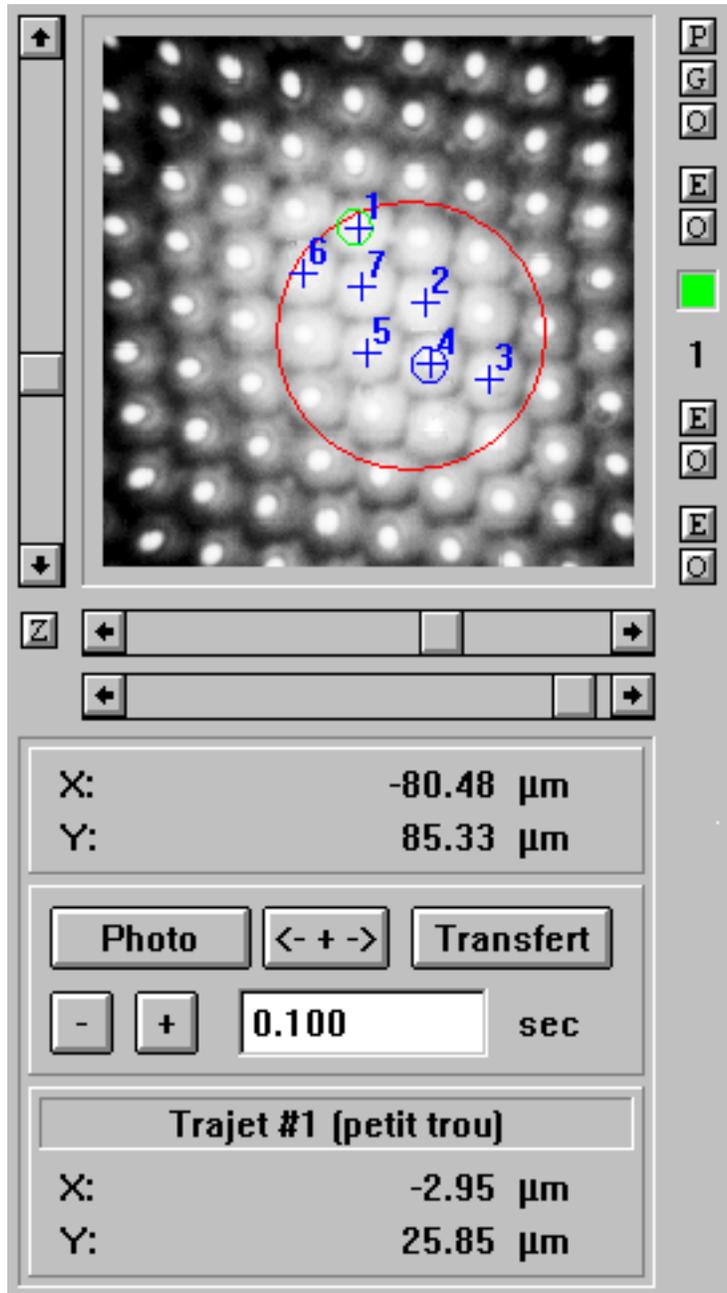


Figure 4-26 : Représentation des contrôles relatifs à la manipulation de l'image de l'application *Contrôle des expériences*.

possible d'agrandir l'image jusqu'à ce qu'un pixel de l'image occupe une région de vingt par vingt pixels sur l'écran.

La fenêtre est aussi accompagnée de deux indicateurs permettant de connaître la position de la souris sur l'image pour différents systèmes de coordonnées (voir figure 4-29, page 130). Les unités des systèmes de coordonnées sont préalablement calibrées à l'aide d'un algorithme de calibration permettant de trouver les unités réelles de l'image. La calibration est faite en plaçant une échelle graduée, précise au μm près, et en pointant sur deux points de l'échelle avec la souris. Par la suite, l'utilisateur indique à l'application la distance entre les deux points obtenue à l'aide de l'échelle graduée pour qu'elle puisse calculer la correspondance entre une distance exprimée en pixels et une distance exprimée en μm . Le succès de l'algorithme de calibration est possible seulement si l'échelle de l'axe des X est la même que celle de l'axe des Y.

L'avant-dernière série de petits boutons située à droite de l'image permet de choisir le type de déplacement effectué par une action de la souris (voir section 4.11.4, page 127 et section 4.11.7, page 133). Lors du changement du type de déplacement, le curseur, lorsqu'il est dans la fenêtre de l'image, est aussi changé pour indiquer le type de déplacement utilisé. Le curseur change aussi de forme lorsqu'il s'agit de l'exécution d'un algorithme de calibration pour indiquer chacune des étapes de la calibration. Le changement du type de déplacement peut aussi être fait à l'aide d'un menu contextuel commandé par le bouton droit de la souris lorsque cette dernière est dans l'image.

La dernière série de contrôles permet simplement de connaître l'emplacement ou l'orientation des faisceaux lumineux (voir 4.11.7, page 133). Puisqu'il n'est pas possible d'afficher les valeurs de tous les faisceaux lumineux, seules les valeurs du faisceau lumineux sélectionné sont affichées. Par contre, l'emplacement et le diamètre de tous les faisceaux lumineux sont indiqués par des cercles dessinés sur l'image où le diamètre du cercle est le diamètre du faisceau lumineux. De plus, il est possible de différencier les faisceaux lumineux en regardant la couleur des cercles.

Il est aussi possible de placer des marqueurs, c'est-à-dire de petites croix bleues, pour marquer la position des facettes présentes sur l'image. Ces marqueurs sont par la suite utilisés par le protocole expérimental pour positionner les faisceaux lumineux (voir section 4.11.8, page 134).

4.11.4 Le contrôle de la base

Lors de l'installation de la base sur le montage, il est pratiquement impossible d'avoir les axes X et Y de la base parfaitement orientés avec les axes X et Y de l'image de la caméra. Puisqu'il est plus pratique de spécifier les déplacements selon l'image, l'application doit les convertir en déplacements selon la base. La conversion consiste à effectuer un changement de référentiel en considérant l'angle et le facteur d'échelle entre les deux référentiels (voir figure 4-27). L'angle sert à orienter les deux référentiels, et le facteur d'échelle à convertir les unités de l'image (μm) en unités de moteur (pas d'encodeur). Cependant, les origines des deux référentiels sont identiques.

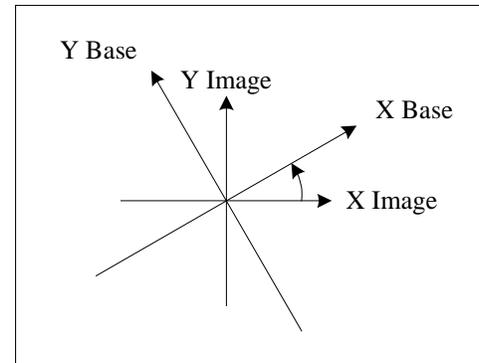


Figure 4-27 : Exemple d'une relation entre le référentiel de l'image et le référentiel de la base où l'orientation et l'échelle sont différentes.

Donc, tous les déplacements demandés par l'utilisateur sont dans le référentiel de l'image et convertis dans le référentiel de la base avant d'être envoyés à l'application *Contrôleur des moteurs* (*x86*). Dans le même ordre d'idées, la réception de la position des moteurs est dans le référentiel de la base et elle est convertie dans le référentiel de l'image avant d'être présentée à l'utilisateur.

La calibration de l'angle et du facteur d'échelle est faite à l'aide d'un algorithme de calibration. L'algorithme demande à l'utilisateur de placer un point fixe de la base dans trois régions différentes de l'image en déplaçant les moteurs X et Y de la base. Pour chacune des trois régions, l'utilisateur doit pointer avec la souris, sur l'image, l'emplacement exact du point. Après avoir indiqué les trois points, l'algorithme fait le calcul de l'angle et du facteur

d'échelle. Par la suite, il faut spécifier l'emplacement du zéro de la base à l'aide d'un autre algorithme de calibration qui consiste simplement à pointer le centre de la base sur l'image.

Le contrôle de la base est donc composé du contrôle de l'axe des X, de l'axe des Y, de l'axe des Z, de l'orientation phi et de l'orientation thêta (voir section 3.3.4, page 26). La conversion des référentiels est effectuée pour les axes X et Y et aussi pour les orientations phi et thêta. L'angle utilisé pour la conversion des référentiels des orientations phi et thêta est le même que pour la conversion des référentiels des axes X et Y. D'autre part, le facteur d'échelle utilisé pour la conversion des référentiels des orientations phi et thêta permet de convertir les unités de rotation degrés en unités de moteur (pas d'encodeur). De son côté, l'axe des Z utilise seulement le facteur d'échelle pour convertir les unités de hauteur (μm) en unités de moteur (pas d'encodeur).

Le contrôle de la base et l'affichage de sa position sont assurés par les contrôles de la figure 4-28. En premier lieu, on y retrouve une série de boutons disposés en forme du symbole '+' permettant de déplacer et d'orienter la base selon la direction des flèches dessinées sur les

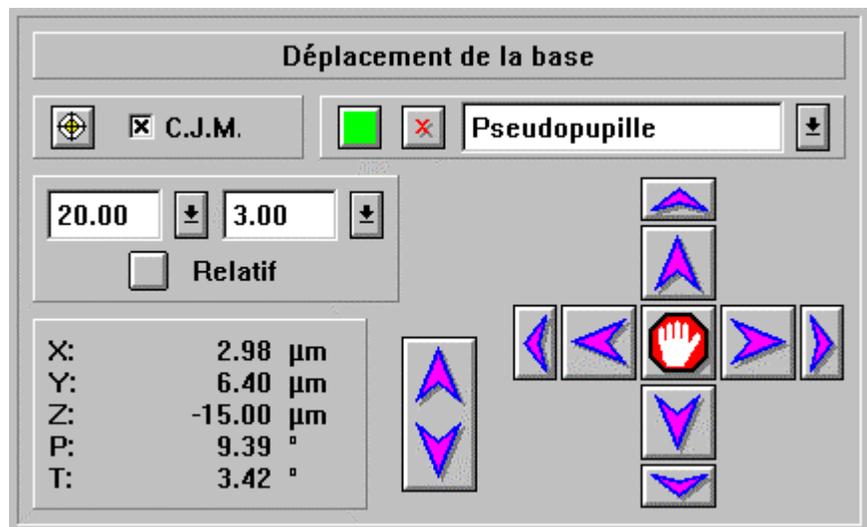


Figure 4-28 : Représentation des contrôles relatifs au déplacement de la base de l'application *Contrôle des expériences*.

boutons. Les boutons intérieurs sont utilisés pour le déplacement de la base (x, y), et les boutons extérieurs pour l'orientation de la base (phi, thêta). Le bouton du centre est utilisé pour annuler un déplacement en cours. Toutefois, le contrôle de la hauteur de la base est fait par le bouton avec deux flèches. Il faut activer ce bouton avec le bouton gauche de la souris pour faire descendre la base et avec le bouton droit de la souris pour la faire monter.

Tout comme dans l'application *Contrôleur de moteurs (x86)*, il est possible de déplacer la base de différentes façons. Il est possible de la déplacer selon des déplacements relatifs ou continus. Les distances ou les vitesses utilisées doivent être inscrites dans l'une des deux listes déroulantes situées à gauche de la fenêtre. La liste déroulante de gauche est utilisée pour les axes X, Y et Z, et la liste déroulante de droite pour les orientations phi et thêta. La sélection du mode de déplacement se fait à l'aide du bouton situé sous les deux listes déroulantes. De plus, les listes déroulantes mémorisent les dix dernières valeurs pour chaque type de déplacement.

Puisqu'il est souvent utile de pouvoir corriger les jeux mécaniques des moteurs, on retrouve aussi les deux contrôles permettant d'activer ou de générer les corrections du jeu mécanique pour tous les moteurs de la base simultanément.

Afin de connaître l'emplacement et l'orientation de la base, une série de cinq indicateurs est présente dans la fenêtre. Toutefois, pour l'emplacement et l'orientation de la base, il est possible d'afficher la position ou l'orientation en utilisant un système de coordonnées cartésiennes ou polaires. Le choix du système de coordonnées se fait à l'aide d'un menu contextuel accessible au moyen du bouton droit de la souris quand la souris est au-dessus des indicateurs.

La dernière série de trois contrôles permet de mémoriser la position et l'orientation de la base. La mémorisation est effectuée en inscrivant simplement un nom dans la liste déroulante. Par la suite, on choisit la destination préalablement sauvegardée dans la liste et on utilise le bouton vert pour effectuer le déplacement. Le bouton avec la croix rouge permet de retirer un élément de la liste. L'utilisation de ce système impose cependant la mémorisation et le déplacement des trois axes et des deux orientations simultanément.

Il est aussi possible de déplacer et d'orienter la base de façon absolue et relative à l'aide de l'image de la préparation et de la souris. Pour déplacer la base, il faut préalablement sélectionner le type de déplacement avec l'un des quatre boutons inférieurs des contrôles relatifs à l'image (voir figure 4-26, page 125). Les deux boutons les plus hauts permettent de

sélectionner un type de déplacement absolu, et les deux boutons du bas de choisir un déplacement de type relatif. Les boutons avec la lettre E sont utilisés pour changer l'emplacement des axes X et Y et les boutons avec la lettre O permettent de changer les orientations phi et theta. Pour mieux refléter le déplacement en cours, le curseur de la souris change aussi de forme pour indiquer le type de déplacement sélectionné. Donc, le déplacement de la base est effectué en pointant dans la région de l'image voulue. L'emplacement de la base selon les axes X et Y est directement donné par la position de la souris sur l'image. Par contre, la relation entre la position de la souris sur l'image et l'orientation de la base résultante n'est pas liée aux coordonnées de l'image. Il s'agit d'un système en degrés où la bordure de l'image correspond à 15° et où il est possible de varier les orientations phi et theta de $\pm 15^\circ$ (voir figure 4-29).

Pour donner un meilleur contrôle à l'utilisateur, il est possible d'exprimer et de contrôler l'emplacement et l'orientation de la base en utilisant un système de coordonnées cartésiennes ou un système de coordonnées polaires.

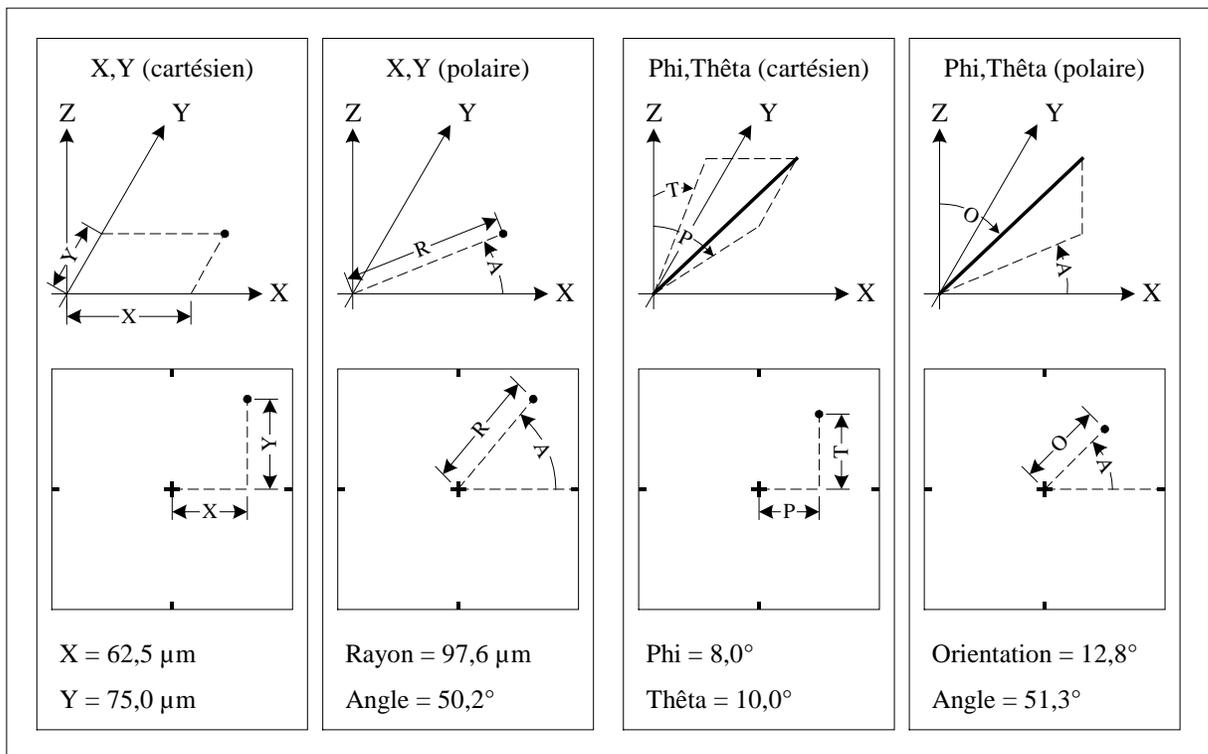


Figure 4-29 : Correspondance entre l'emplacement de la souris et le déplacement effectué par la base dans l'application *Contrôle des expériences*.

Pour bien comprendre le fonctionnement des déplacements avec la souris, la figure 4-29 de la page précédente présente les quatre cas possibles. Premièrement, il y a dans le haut de la figure les graphiques de l'emplacement de la base, représenté par le point noir, ou de l'orientation de la base, représentée par une ligne noire correspondant à la normale à la préparation. Deuxièmement, il y a les graphiques du bas représentant l'endroit où doit pointer la souris, représenté par le point noir, pour générer le déplacement ou l'orientation, présentés dans les graphiques du haut, par rapport au zéro du référentiel, représenté par la croix. Finalement, le texte du bas et les lignes pointillées présentent la position ou l'orientation et la façon de mesurer la position ou l'orientation de la base.

Pour ce qui est des deux premiers cas, il s'agit des systèmes de coordonnées cartésiennes et polaires standard où la souris peut modifier l'emplacement selon les axes X et Y de la base seulement. Pour ce qui est des deux derniers cas, il s'agit d'une extension au système de coordonnées standard où les positions en μm ont été changées pour des angles en degré. L'utilisation de la présente technique pour orienter la base facilite énormément l'alignement de la préparation sans avoir besoin de faire des calculs (voir section 5.5, page 149).

4.11.5 Le contrôle de la microélectrode

L'automatisation de l'avancement de la microélectrode de mesure permet de mieux connaître l'emplacement de cette dernière dans la préparation. Elle est réalisée par l'ensemble des contrôles de la figure 4-30. Dans un premier temps, il y a les trois boutons du bas de la



Figure 4-30 : Représentation des contrôles du déplacement de la microélectrode de l'application *Contrôle des expériences*.

fenêtre qui permettent de reculer la microélectrode, d'annuler un déplacement ou d'avancer la

microélectrode. Le type de déplacement de la microélectrode est contrôlé par un bouton et une liste déroulante. Le bouton permet de choisir entre un déplacement de type relatif ou un déplacement de type continu. La distance à parcourir ou la vitesse de déplacement est entrée dans la liste déroulante qui conserve les dix dernières valeurs utilisées pour chaque type de déplacement. La position de la microélectrode est affichée en μm , et il est possible d'ajuster le zéro de l'indicateur.

Parmi les autres contrôles, il y a un bouton pour faire sortir la microélectrode quand l'expérience est terminée et un bouton permettant de désactiver les autres contrôles. Il est utile de désactiver les autres contrôles quand la microélectrode est correctement insérée dans la préparation et quand on veut éviter qu'un déplacement de la microélectrode soit fait par mégarde.

Enfin, il y a un bouton permettant d'activer la mesure d'impédance d'une cellule. La mesure est faite à l'aide de l'amplificateur AxoClamp et le générateur de séquences. L'amplificateur est responsable de fixer l'amplitude du courant à injecter et de la mesure de l'impédance, et le générateur de séquence est responsable de générer une onde carrée utilisée par l'amplificateur pour connaître le moment où il doit injecter le courant.

4.11.6 Le contrôle du trajet de fond et du polariseur

Lors de la réalisation d'une expérience, il est nécessaire d'avoir un éclairage global utilisé lors de la recherche d'un photorécepteur au début d'une expérience ou pour contrôler la lumière d'ambiance d'une expérience. Cet éclairage global est réalisé par le quatrième trajet lumineux que l'on nomme *trajet de fond*. Le faisceau de lumière généré par ce trajet est positionné sur le centre de l'axe optique du montage, et il n'est pas possible de le déplacer. Cependant, il est possible de changer le diamètre du faisceau lumineux.

Les contrôles relatifs au trajet de fond permettent donc de contrôler l'éclairement et le diamètre du faisceau lumineux. Le contrôle de l'éclairement est fait par une série de quatre boutons permettant d'allumer () , d'éteindre () ou de faire clignoter le faisceau lumineux ( et ). Ces boutons sont une réplique des boutons de l'application *Générateur*

de séquences (x86) (voir figure 4-31). En outre, le contrôle du diamètre est fait à l'aide d'un ascenseur et d'une liste déroulante qui contient les dix dernières valeurs à être entrées. Le contrôle du diamètre du faisceau lumineux est directement donné en μm .

Dans le bas de la figure 4-31, on montre un autre ascenseur et une autre liste déroulante permettant de contrôler l'angle du polariseur du stimulateur optique. Le contrôle de l'angle du polariseur est directement donné en degré, et la liste déroulante conserve les dix dernières valeurs à être entrées.

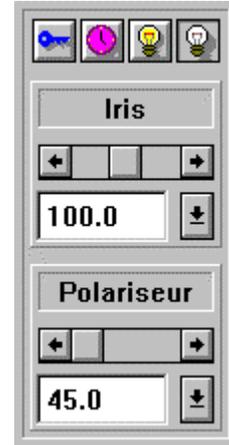


Figure 4-31 : Représentation des contrôles du trajet de fond et du polariseur de l'application *Contrôle des expériences*.

4.11.7 Le contrôle des trajets lumineux

Afin de stimuler individuellement les cellules, le montage est muni d'une série de trois faisceaux lumineux dont le diamètre peut être assez petit pour éclairer une seule facette.

Ces faisceaux lumineux sont prévus pour être déplacés et orientés individuellement, et chaque faisceau peut avoir deux diamètres différents (voir section 3.3.2, page 24). Il est aussi possible, pour chacun de ces faisceaux lumineux, d'en contrôler l'intensité lumineuse. La

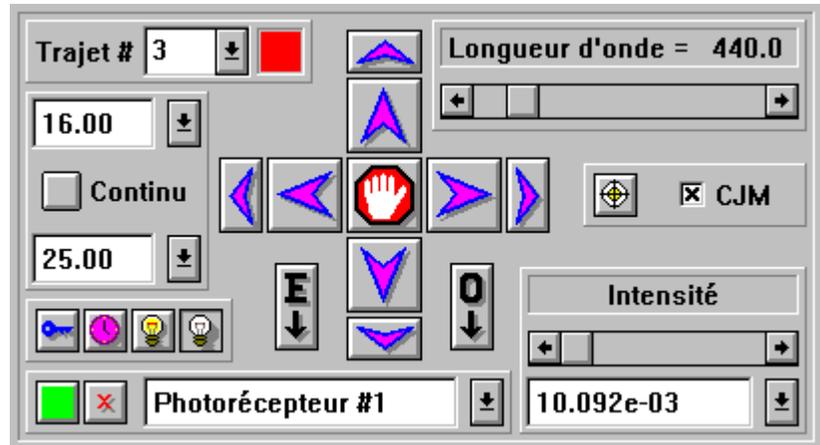


Figure 4-32 : Représentation des contrôles responsables des trajets lumineux de l'application *Contrôle des expériences*.

longueur d'onde peut être contrôlée seulement pour le troisième faisceau lumineux.

L'ensemble des contrôles des trajets lumineux est présenté à la figure 4-32. La série de contrôles comprend une liste déroulante qui permet de choisir quel trajet lumineux sera commandé par les autres contrôles. On constate que les contrôles sont les mêmes que ceux de

la base (voir section 4.11.4, page 127). On pourra donc déplacer et orienter la base, activer et effectuer le retour mécanique, sauvegarder la position du trajet et choisir le type de déplacement. Il est aussi possible de déplacer le trajet de façon absolue en utilisant la fenêtre de l'image (voir figure 4-26, page 125). Il faut cependant utiliser les trois boutons supérieurs ,  et  pour changer respectivement l'emplacement du petit trou, l'emplacement du grand trou et l'orientation du faisceau lumineux de façon absolue. Les deux boutons  et  permettent de changer l'emplacement et l'orientation des trajets de façon relative. La sélection du trajet est faite à l'aide d'un menu contextuel accessible en pointant, avec le bouton droit de la souris, le carré donnant la couleur et le numéro du trajet sélectionné. La position ou l'orientation du faisceau est affichée dans les indicateurs tout en bas de l'image en utilisant un des deux systèmes de coordonnées choisis, encore une fois, à l'aide d'un menu contextuel.

La fenêtre est aussi munie d'un ensemble de quatre boutons permettant d'allumer () , d'éteindre () et de faire clignoter ( et ) le faisceau lumineux. Ces boutons sont une réplique des contrôles utilisés dans l'application *Générateur de séquences (x86)*.

Finalement, il y a deux séries de contrôles permettant de modifier la longueur d'onde et l'intensité du faisceau lumineux. Il y a donc un ascenseur et un indicateur permettant de changer et de voir la longueur d'onde du faisceau de lumière et un autre ascenseur et une liste déroulante permettant de changer et de voir l'intensité lumineuse du faisceau lumineux.

4.11.8 Le protocole expérimental

La réalisation d'une expérience consiste en une séquence d'opérations qu'il faut effectuer dans un ordre et un temps précis. Cette séquence est habituellement réalisée par l'utilisateur qui doit chronométrer les événements et activer les bons contrôles au bon moment. Cette étape de l'expérience est très importante et très sujette à des erreurs, car la majorité des expériences se font à la noirceur. La présente section consiste donc en une liste de directives. Les directives présentes permettent d'effectuer pratiquement toutes les opérations nécessaires à la réalisation d'une expérience. Il est donc possible de placer les différentes directives les unes à la suite des autres pour générer le protocole expérimental. Pour ce faire, il y a sept groupes de directives.

- Boucles : Ce groupe de directives permet d'insérer des boucles dans le protocole pour répéter une suite de directives.
- Chronomètres : Ces commandes permettent d'ajouter des délais dans le protocole et de synchroniser les directives. Les délais sont surtout utilisés pour générer des temps d'adaptation, et la synchronisation pour synchroniser le début des boucles.
- Séquenceurs : Cet ensemble de directives permet d'ouvrir, de fermer, de démarrer, d'activer, de changer la configuration et de changer le nombre de répétitions des séquenceurs de l'application *Générateur de séquences (x86)*.
- Courbes : Cette section permet de démarrer des acquisitions pour enregistrer les réponses électriques de la préparation. Les courbes acquises sont manipulées et sauvegardées par l'application *Acquisition et traitement des courbes*.
- Faisceaux lumineux : Ces directives sont nécessaires pour déplacer, changer le diamètre, changer la longueur d'onde et changer l'intensité des faisceaux lumineux. Le déplacement des faisceaux se fait en indiquant lequel des faisceaux doit se positionner sur un marqueur quelconque. Les marqueurs sont placés sur l'image à l'aide du bouton droit de la souris (voir figure 4-26, page 125).
- Commentaires : Cette commande permet simplement d'ajouter des commentaires dans le protocole.
- Vidéo : Cette série de directives permet de commander l'acquisition d'images et de contrôler le temps d'intégration. Les images sont manipulées et sauvegardées par l'application *Acquisition et traitement des images*.

La réalisation et l'exécution du protocole expérimental sont faites à l'aide des contrôles de la figure 4-33. Il y a deux listes : celle de gauche qui contient le protocole proprement dit et celle de droite qui contient les directives qu'on peut

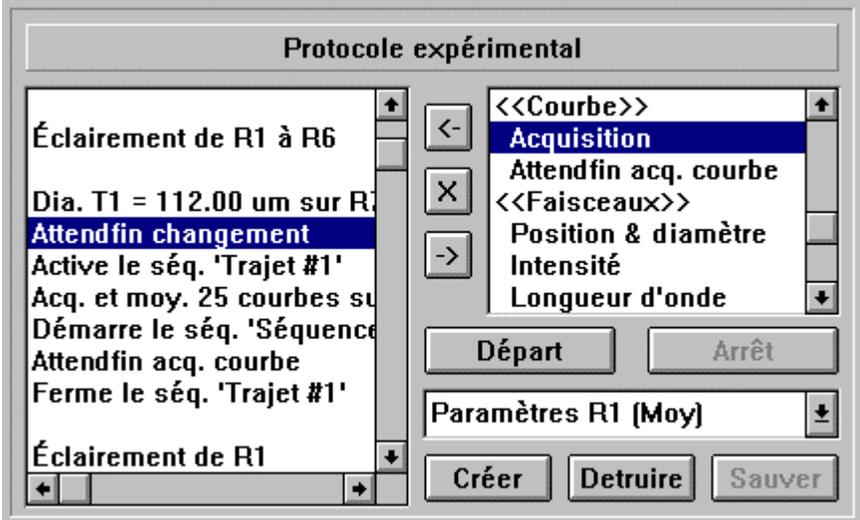


Figure 4-33 : Représentation des contrôles responsables du protocole expérimental de l'application *Contrôle des expériences*.

insérer dans le protocole.

Le bouton avec la flèche pointant vers la gauche est utilisé pour insérer une directive dans le protocole avant la directive sélectionnée, et le bouton avec une flèche pointant vers la droite permet de retirer la directive sélectionnée. Il y a aussi le bouton avec un X qui permet d'effacer complètement le protocole.

Le départ et l'arrêt du protocole se font à l'aide des deux boutons portant les noms *départ* et *arrêt*. Il y a aussi une liste déroulante qui permet de choisir parmi les protocoles présents sur le disque et une série de trois boutons qu'on peut utiliser pour créer, détruire et sauvegarder un protocole sur le disque dur de l'ordinateur. L'annexe B de la page 193 présente un protocole expérimental qui pourrait être utilisé pour obtenir les informations nécessaires à la résolution du modèle du chapitre VI.

4.11.9 Les algorithmes spécialisés

Toujours dans le but de diminuer le temps nécessaire à la réalisation d'une expérience, il est utile d'implanter des algorithmes de haut niveau permettant d'assister l'utilisateur dans son travail. L'application possède trois algorithmes spécialisés.

Le premier de ces trois algorithmes est utilisé pour centrer la cellule sous enregistrement. Avant de procéder, l'utilisateur doit préalablement trouver une cellule et la placer dans la région éclairée par un faisceau lumineux en déplaçant la base. Après avoir trouvé la cellule, l'utilisateur démarre l'algorithme qui procède comme montré à la figure 4-34, où la position du faisceau lumineux est représentée par le grand cercle, et la cellule sous enregistrement par la croix noire. La première étape consiste à déplacer la base vers la droite pour trouver la limite droite du faisceau lumineux. La position de la limite du faisceau lumineux est obtenue en faisant clignoter le faisceau

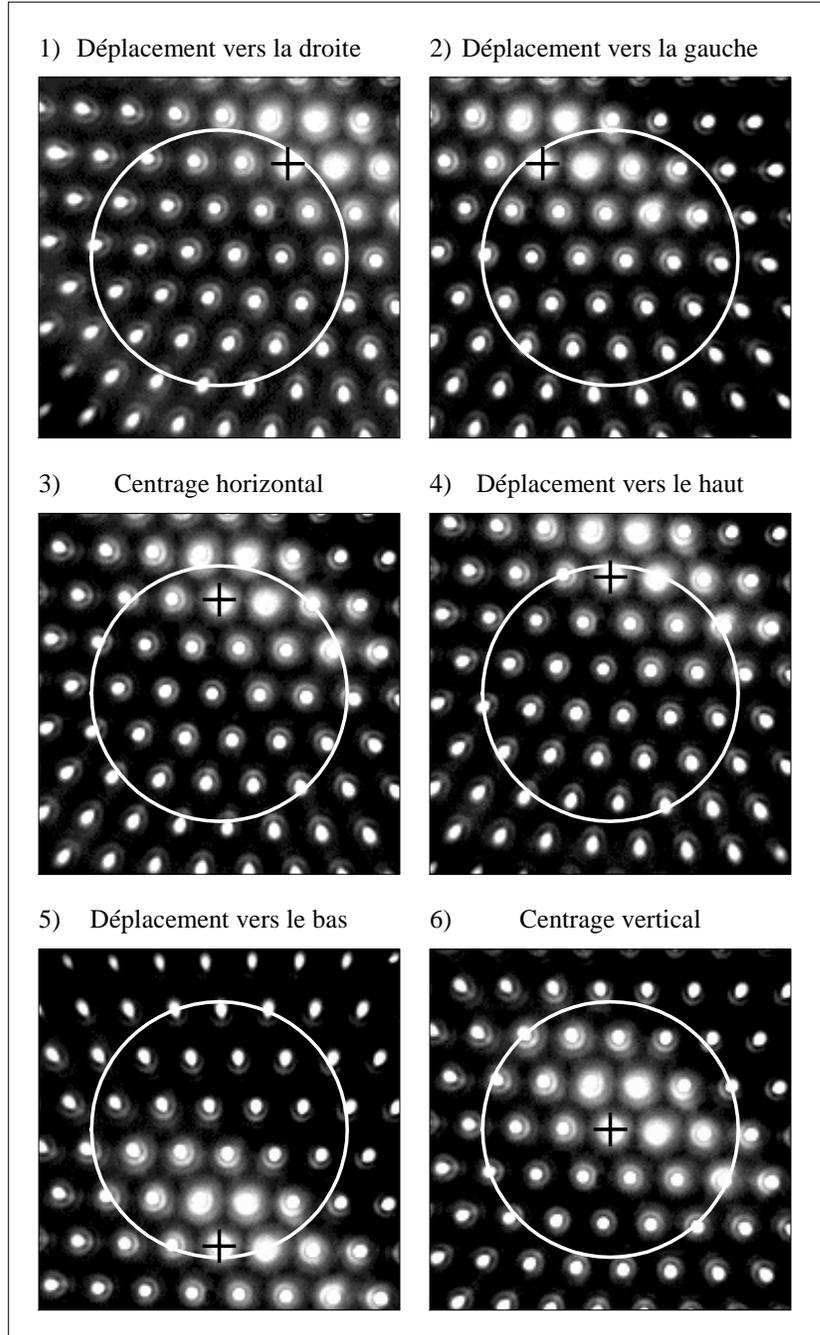


Figure 4-34 : Représentation de la recherche des facettes de l'application *Contrôle des expériences*.

lumineux et en mesurant la réponse de la cellule. Lorsque la réponse est très faible, le photorécepteur étant à l'extérieur du faisceau lumineux, c'est que la limite du faisceau lumineux est atteinte. A ce moment précis, l'algorithme prend en note la position de la base. L'algorithme poursuit en déplaçant la base vers la gauche pour trouver la limite gauche en

procédant comme à la première étape. Après avoir obtenu ces deux limites, la troisième étape de l'algorithme consiste à calculer la position et à positionner la base pour que la cellule se retrouve au centre du faisceau lumineux. Après avoir fait l'alignement horizontal, l'algorithme refait les mêmes étapes pour faire l'alignement vertical de la base.

Le second algorithme est utilisé pour aligner l'orientation de la cellule sous enregistrement dans l'axe optique du montage. Pour ce faire, l'algorithme fait tourner la base et mesure la réponse de la cellule pour différentes orientations. Après, l'algorithme tente de faire correspondre un polynôme du deuxième ordre à la série de mesures. Le maximum du polynôme donnera l'orientation qui génère la réponse maximale de la cellule (voir figure 4-35). Finalement, cette orientation est utilisée pour orienter la base. L'algorithme est répété pour les deux orientations de la base, c'est-à-dire ϕ et θ .

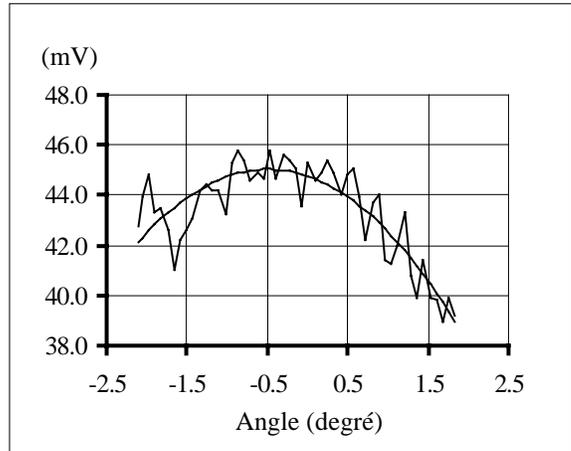


Figure 4-35 : Représentation de l'alignement d'une cellule. La courbe dentelée représente la réponse de la cellule, et la courbe continue le polynôme. Dans le présent cas, la réponse maximale se situe à -0.5° .

Le dernier algorithme permet de mesurer la réponse spectrale d'un photorécepteur sous enregistrement. L'algorithme consiste à trouver l'intensité lumineuse du faisceau lumineux pour que la réponse du photorécepteur soit toujours la même pour une série de filtres d'interférence. La position du filtre d'intensité pour chacun des filtres d'interférence donnera l'atténuation nécessaire pour chaque filtre d'interférence. Les atténuations et les longueurs d'onde des filtres d'interférence seront par la suite utilisées pour générer la courbe de sensibilité spectrale du photorécepteur. En dernier lieu, la courbe est transmise à l'application *Acquisition et traitement des courbes* pour que l'utilisateur puisse la conserver ou la visualiser.

La construction des trois algorithmes présentés ci-dessus permet à l'utilisateur de gagner énormément de temps lors de la réalisation d'une expérience. Le temps est un facteur important, car il est difficile de maintenir un photorécepteur sous enregistrement pendant plus de deux heures.

4.11.10 L'acquisition de courbes

La série de contrôles de la figure 4-36 est utilisée pour permettre à l'utilisateur de connaître rapidement l'amplitude ou l'impédance d'une cellule. Il est aussi possible de faire la moyenne d'une série d'acquisitions lorsque les réponses convoitées sont de très faible amplitude. Le résultat obtenu est directement affiché à l'écran ou sauvegardé dans les items *temporaire*, *brut mémoire* ou *brut disque* de l'application *Acquisition et traitement des courbes* selon l'état des contrôles du bas. Il est aussi possible de choisir une des deux microélectrodes disponibles pour faire la mesure. Lors de la mesure de l'impédance d'une cellule, il faut que l'amplificateur soit préalablement bien ajusté avant de procéder (voir section 5.4, page 146).



Figure 4-36 : Représentation des contrôles responsables de l'acquisition des courbes de l'application *Contrôle des expériences*.

4.11.11 Discussion

La réalisation de l'application *Contrôle des expériences* s'est avérée l'aboutissement de la réalisation du montage. L'application a subi et subira une multitude de mutations pour améliorer le contrôle de l'expérience tout en contrôlant de mieux en mieux les parties du stimulateur lumineux. Les fonctionnalités de la présente application doivent refléter la réalité. Il faut faciliter le contrôle de l'expérience tout en n'oubliant jamais qu'elle est faite à la noirceur et qu'il faut diminuer au minimum l'utilisation du clavier de l'ordinateur car il est dans l'obscurité et peu convivial.

La difficulté de la réalisation de la présente application ne réside pas dans les algorithmes mais plutôt dans la construction d'une interface efficace. Le but ultime de ce travail a été de

faciliter le travail requis des expérimentateurs qui utiliseront le montage et qui effectueront des expériences.

Enfin, une modification au montage pourra se limiter à une modification d'une partie du stimulateur optique et à la modification de la présente application. Dans la majorité des cas, il ne sera pas nécessaire de modifier les applications de contrôle.

4.12 Conclusion

La réalisation des différentes applications présentées dans ce chapitre permet d'avoir un montage qui fonctionne correctement et qui rencontre les objectifs fixés. L'aboutissement de ce travail a nécessité en moyenne trois versions majeures de logiciels pour chacune des applications : une première version qui permettait d'avoir les fonctionnalités voulues mais très difficiles à utiliser, une deuxième version améliorant grandement l'interface usager et une dernière version permettant de faire les ajustements nécessaires après l'utilisation de l'application dans le cadre d'une vraie expérience. De plus, le passage de la deuxième à la troisième version a été accompagné d'un changement de système d'exploitation.

Le plus intéressant est que la structure utilisée pour gérer le contrôle d'une expérience s'est avérée à la hauteur. Compte tenu de la lenteur des expérimentations à réaliser, on pourrait même qualifier l'ensemble des applications comme fonctionnant en temps réel. L'utilisation de la communication DDE et des DLL a permis d'avoir un système parfaitement modulaire dont chacun des modules peut être modifié sans avoir besoin de modifier les autres modules. Le tout résidait dans la définition d'une bonne convention de communication entre les modules.

Cependant, il serait illusoire de croire que les applications sont parfaites. Il demeure encore quelques erreurs mineures dans les différentes applications qui ne nuisent cependant pas au bon fonctionnement d'une expérience. Par contre, elles rendent les applications fragiles pour des utilisateurs inexpérimentés qui ne comprennent pas bien chacune des options. À titre d'information, il y a un fichier associé à chacune des applications indiquant les erreurs encore présentes dans chacune d'entre elles et les suggestions d'amélioration à apporter. Parmi les

suggestions, il y a surtout des idées qu'il serait avantageux d'implanter pour faciliter la réalisation d'expériences.

Finalement, voici le tableau 4-1 donnant un aperçu de l'ampleur du projet.

Tableau 4-1 : Statistiques concernant la réalisation des logiciels du projet.

Applications, « librairies » ou pilotes	Nombre de lignes de codes	Nombre de boîtes de dialogue	Nombre d'images
Contrôleur de moteurs (68HC11)	3594	0	0
Générateur de séquences (68HC11)	1527	0	0
Contrôleur des moteurs (x86)	10365	0	15
Générateur de séquences (x86)	5265	4	5
Acquisition et traitement des courbes	23860	21	5
Acquisition et traitement des images	22194	19	6
Contrôle des expériences	29063	40	76
Serveur de ports RS-232	1773	1	0
« Librairie » et pilote DT2801-A	5996	5	0
« Librairie » et pilote DAS16	5805	5	0
« Librairie » STAR I	2883	1	0
« Librairie » et pilote XICAS	5234	1	0
« Librairie » diverses	3766	1	32
Total	121325	98	139

CHAPITRE V

ÉTAPES PRÉLIMINAIRES D'UNE EXPÉRIENCE

5.1 Introduction

Malgré la multitude d'expériences qu'il est possible de réaliser avec le montage, les cinq premières étapes sont pratiquement toujours identiques pour toutes les expériences. Les deux premières étapes consistent à installer successivement la préparation et les microélectrodes sur le stimulateur optique. La recherche et l'alignement d'une cellule sont réalisés lors des troisième et quatrième étapes. La dernière étape est par contre facultative. Elle consiste à identifier les facettes participant à la superposition neuronale et formant une pseudopupille cornéenne réduite.

5.2 Installation de la préparation sur le stimulateur optique

L'installation de la préparation débute par la capture d'une mouche domestique femelle âgée de 3 à 9 jours après l'éclosion de la puppe. L'utilisation de la femelle nous assure une uniformité des interconnexions neuronales peu importe la région de l'œil sous enregistrement. Cette uniformité n'est pas respectée chez le mâle car celui-ci possède une région frontale spécialisée dans la poursuite de la femelle en vue des accouplements sexuels. Le délai de 3 à 9 jours nous permet d'avoir une mouche avec des tissus ayant une certaine fermeté (après 3

jours) sans être trop rigides (après 9 jours). L'insecte est par la suite collé sur un support à l'aide d'une cire d'abeille colorée de noir pour diminuer les reflets lumineux. Afin de permettre aux microélectrodes d'atteindre les centres nerveux d'intérêt de l'insecte, un trou est percé, à l'aide d'un scalpel, dans la paroi arrière de la tête de l'insecte. Avec cette technique, la mouche demeure en vie durant toute l'expérience qui peut s'étendre sur plusieurs heures.

Par la suite, l'insecte est orienté à l'aide du support pour choisir la région de l'œil à étudier. Le choix de la région de l'œil se fait en orientant l'insecte de telle sorte que la pseudopupille cornéenne de l'insecte corresponde à la région de l'œil d'intérêt. L'orientation de l'insecte doit aussi permettre à la microélectrode, qui entre par le trou situé à l'arrière de l'œil, d'atteindre la région correspondant à la pseudopupille cornéenne. Puisque les degrés de liberté du support sont limités, l'orientation de l'insecte doit surtout être faite à l'étape du collage.

Pour réussir à bien orienter l'insecte, l'ajustement est fait sous un télescope binoculaire où le support est placé comme sur le montage. En connaissant la position et l'orientation des microélectrodes, il est alors facile de savoir comment aligner l'insecte (voir figure 5-1). Il est aussi nécessaire de laver la surface de l'œil de l'insecte avec de l'eau pour enlever les saletés présentes sur l'œil. Ce lavage permet de mieux voir la pseudopupille cornéenne et évite de fausser l'alignement des faisceaux lumineux lors de l'expérience.

L'ajustement de l'insecte et l'emplacement exact du trou dans la paroi arrière de l'œil dépendront aussi du type de cellule que l'on désire étudier. Pour l'étude dans le segment interne des photorécepteurs, l'ajustement est tel qu'il permet aux microélectrodes de passer dans la région de la rétine de l'œil. Pour l'étude dans les axones des

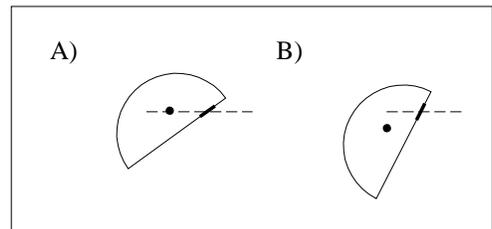


Figure 5-1 : Représentation de l'alignement de l'œil de l'insecte sur le support en tenant compte de l'orientation des microélectrodes (ligne pointillée), du trou (trait gras) et de la pseudopupille (point noir). A) Cas où l'insecte est correctement aligné. B) Cas où l'insecte est incorrectement aligné et où les microélectrodes ne peuvent passer par la pseudopupille cornéenne.

photorécepteurs, dans les grosses cellules monopolaires et dans les cartouches, l'ajustement permet aux microélectrodes de passer dans la région lamina de l'œil (voir figure 2-2, page 5).

Après que l'insecte ait été fixé sur le support, celui-ci est placé sur le montage en prenant soin de l'aligner correctement. Un bon alignement est assuré lorsque la pseudopupille cornéenne est placée au centre de l'axe optique du microscope et que la mise au point du microscope est ajustée pour bien voir la surface de l'œil dans l'oculaire et la caméra du stimulateur optique. Il faut aussi que le zéro des axes X et Y de la base corresponde à l'axe optique du microscope et que le zéro de l'axe des Z corresponde à la pseudopupille profonde de l'insecte.

L'ajustement de la pseudopupille cornéenne au centre de l'axe optique du microscope (voir, figure 3-3, page 21) augmente les chances que la position future de la microélectrode soit aussi au centre de l'axe optique du microscope. De plus, l'ajustement en hauteur diminue le désalignement de la pseudopupille cornéenne avec l'axe optique du microscope et la mise au point du microscope lors d'une rotation de la base. La figure 5-2-C illustre comment la mise au point du microscope et l'alignement de la pseudopupille ne seront plus assurés à la suite d'une rotation lorsque le centre de la base ne correspond pas au centre de l'œil.

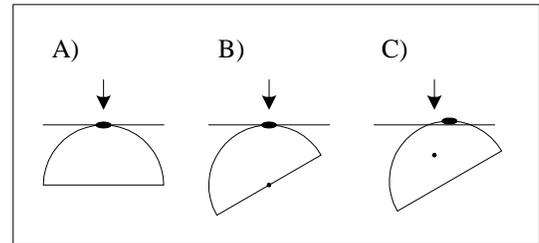


Figure 5-2 : Représentation de la modification de l'alignement de la pseudopupille cornéenne (ellipse) avec le centre de l'axe optique du microscope (flèche) et de la mise au point du microscope (ligne horizontale) à la suite d'une rotation de la base où le point noir indique le centre de rotation de la base. A) Cas avant rotation. B) Cas d'une rotation de 30° où le zéro de la base correspond au centre de l'œil. C) Cas d'une rotation de 30° où le zéro de la base diffère du centre de l'œil.

L'alignement de la pseudopupille cornéenne avec l'axe optique du microscope est fait à l'aide de l'oculaire ou de la caméra du stimulateur optique. Puisque le centre de l'image, obtenu par la caméra, correspond à l'axe optique du microscope, il suffit d'aligner la pseudopupille cornéenne au centre de l'image. Il est aussi possible d'utiliser l'oculaire, mais l'alignement est plus difficile à effectuer, car il n'y a pas de point de référence comme dans le cas de l'image.

Afin d'obtenir la bonne hauteur de la préparation, on ajuste la mise au point du microscope sur la pseudopupille profonde de l'insecte pour une hauteur de la base égale à zéro. Après, la hauteur de la base est ajustée pour que la mise au point du microscope corresponde à la surface de l'œil de l'insecte. Le zéro de la base est préalablement calibré pour que son centre de rotation corresponde au plan focal et à l'axe optique du microscope.

Dans le cas où il n'est pas possible de voir la pseudopupille profonde, il faut estimer la distance du centre de l'œil et faire la mise au point du microscope sur la surface de l'œil de l'insecte en ayant préalablement ajusté la hauteur de la base selon l'estimation de la distance du centre de l'œil.

5.3 Installation des microélectrodes sur le stimulateur optique

Afin d'enregistrer les réponses des cellules, il est nécessaire de fabriquer deux ou trois microélectrodes selon les besoins. La première est faite de borosilicate et possède une grosse pointe d'environ 10 μm . Cette microélectrode sert de point de masse et est appelée microélectrode indifférente. La deuxième est faite de quartz et possède une pointe d'environ 30 nm, ce qui permet d'enregistrer la tension interne des cellules. Dans le cas où l'on travaille avec deux microélectrodes de mesure, on en fabrique une troisième de borosilicate avec une pointe d'environ 100 nm, qui est collée à celle de quartz. Puisque les microélectrodes s'élargissent rapidement à partir de 8 mm du bout de la pointe, il est nécessaire de plier la pointe de la microélectrode de borosilicate pour que les deux pointes restent très proches l'une de l'autre. Il n'est pas possible d'utiliser le quartz pour fabriquer la troisième microélectrode, car l'équipement actuel du laboratoire est inadéquat pour plier des microélectrodes de quartz.

Par la suite, les microélectrodes sont remplies avec une solution d'acétate de potassium. Le branchement avec l'amplificateur est fait à l'aide de jonctions Ag-AgCl pour contrôler la stabilité dans les connexions et obtenir des jonctions liquide-métal dites symétriques.

L'installation sur le stimulateur optique commence par l'insertion de la microélectrode indifférente dans le trou de la partie arrière de l'œil de l'insecte. Par la suite, la microélectrode de mesure est alignée en plaçant sa pointe au-dessus de l'œil de l'insecte et au centre de la

pseudopupille. À ce moment, le zéro de la position de la microélectrode est fixé dans l'application *Contrôle des expériences* (voir section 4.11.5, page 131). Après que le zéro soit fixé, la microélectrode est reculée à l'arrière de la tête de l'animal et descendue vis-à-vis le trou sans modification de sa position latérale. Ensuite, on fait avancer la microélectrode dans le trou tout en prenant soin de ne pas toucher à la paroi du trou pour ne pas la briser.

La dernière étape consiste à vérifier, à l'aide de l'amplificateur, si la ou les microélectrodes de mesure sont en bon état. Seules les microélectrodes ayant une impédance supérieure à 180 M Ω sont considérées comme étant satisfaisantes.

5.4 Recherche d'une cellule

Après avoir inséré les microélectrodes dans l'œil de l'insecte, on commence la recherche de la cellule convoitée. Pour ce faire, il suffit d'avancer, par sauts de 2 μm , la microélectrode de mesure tout en faisant clignoter un faisceau lumineux. Une cellule est détectée lorsque la réponse de l'oscilloscope affiche une dépolarisation ou une hyperpolarisation synchronisée avec le clignotement du faisceau lumineux.

Pour que la recherche de la cellule se réalise adéquatement, il faut utiliser le faisceau lumineux du trajet de fond, car c'est le seul qui peut éclairer la totalité de l'œil de l'insecte. De plus, le faisceau lumineux du trajet de fond a un grand angle de convergence et il a été spécialement conçu avec cette caractéristique. L'angle de convergence est important, car les photorécepteurs de la rétine regardent dans une direction de l'espace bien précise et leur champ de vision forme un cône ayant un angle de vision d'environ 2°. La sensibilité du photorécepteur est supérieure au centre du cône, diminue lorsque l'on s'éloigne du centre du cône et devient complètement absente à la périphérie du cône (van Hateren, 1984). Donc, si l'angle du faisceau lumineux diffère de plus de 1° avec l'axe optique de la cellule, celle-ci ne génère aucune réponse même si le faisceau lumineux est projeté sur la facette au-dessus de la cellule.

Puisque l'angle de convergence du faisceau lumineux du trajet de fond est d'environ 15°, il est possible de stimuler un grand nombre de photorécepteurs de l'œil de l'insecte. Par contre,

les autres faisceaux lumineux possèdent un angle de convergence d'environ $0,6^\circ$ et ils peuvent stimuler seulement les huit photorécepteurs d'une même pseudopupille cornéenne réduite. C'est ce faible angle de convergence et l'angle du faisceau lumineux par rapport à l'orientation de la facette qui permettent d'éclairer un seul des photorécepteurs présents sous la facette. Par contre, il n'est pas possible d'éclairer R7 et R8 individuellement, car ils ont tous deux la même orientation et sont superposés.

La figure 5-3 présente les photorécepteurs, les LMC et les cartouches stimulés pour les deux types de faisceaux lumineux lorsque le diamètre du faisceau lumineux du trajet de fond est le même que pour les autres faisceaux lumineux. On voit très bien que seules les cellules de la pseudopupille cornéenne réduite participant à la superposition neuronale sont stimulées pour les faisceaux lumineux avec un angle de convergence de $0,6^\circ$, car ce sont les seules dont l'angle est inférieur à l'angle de convergence du faisceau lumineux. Par contre, pour le faisceau lumineux du trajet de fond, toutes les cellules présentes sous les facettes éclairées sont stimulées.

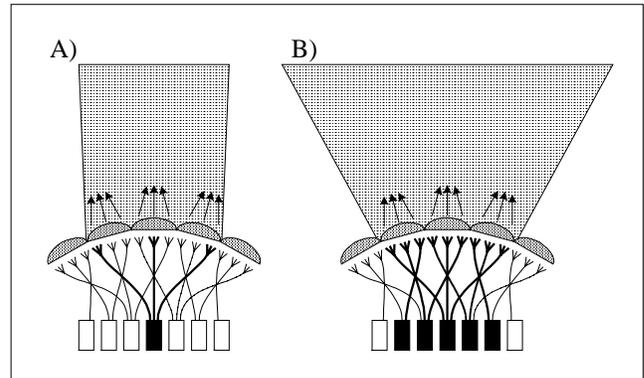


Figure 5-3 : Représentation des cellules stimulées selon l'angle de convergence du faisceau lumineux. Les flèches représentent l'orientation des photorécepteurs, les lignes foncées, les photorécepteurs stimulés et les boîtes noires, les LMC et cartouches qui sont stimulés. A) Faisceaux lumineux générés par les trajets autres que le trajet de fond. B) Faisceau lumineux généré par le trajet de fond.

Après avoir trouvé une cellule, il faut premièrement l'identifier pour savoir s'il s'agit de la cellule convoitée. Une autre distinction rapide consiste à vérifier s'il s'agit d'une dépolarisation ou d'une hyperpolarisation. Dans le cas de la dépolarisation, il peut s'agir d'un photorécepteur ou d'une cartouche. Le cas échéant, il faut passer à l'étape subséquente pour trancher la question. Par contre, une réponse de type hyperpolarisation indique qu'il s'agit d'une des trois LMC. La distinction entre les trois LMC se fait selon la technique de Uusitalo et al. (1995).

Après avoir accompli l'étape subséquente, il est très facile de distinguer le photorécepteur de la cartouche. Il s'agit simplement d'éclairer successivement les différentes facettes de la pseudopupille cornéenne. Si on obtient une réponse pour une seule des facettes, c'est qu'il s'agit d'un photorécepteur.

Si c'est un photorécepteur, il faut aussi estimer l'emplacement de l'électrode dans la cellule pour savoir si nous sommes dans le segment interne ou l'axone de la cellule. Si la réponse du photorécepteur est précédée d'une pointe de tension rapide, il s'agit d'un enregistrement dans l'axone du photorécepteur (Weckström et al., 1992). Il est aussi possible de prédire l'emplacement de la pointe de la microélectrode en remarquant si des LMC ont été obtenues lors du déplacement de la microélectrode. Si c'est le cas, la cellule sous enregistrement est fort probablement une cartouche ou l'axone du photorécepteur.

Il faut aussi mentionner qu'un bon expérimentateur sait comment placer l'insecte et les microélectrodes pour atteindre le centre nerveux qu'il désire. Pour un tel expérimentateur, il est facile, avec son expérience, de prévoir s'il atteindra l'axone ou le segment interne d'un photorécepteur.

L'étape de la recherche de la cellule prend fin par une calibration de l'amplificateur pour compenser l'impédance des microélectrodes qui peuvent générer des erreurs lors de l'injection d'un courant dans la cellule. Cette étape est surtout importante si l'utilisateur désire prendre des mesures d'impédance de la cellule lors de

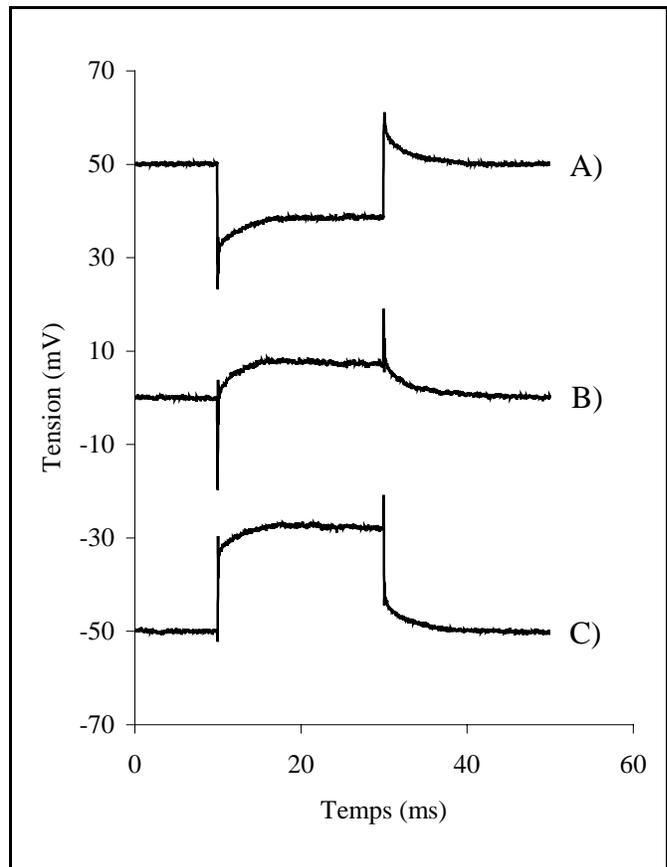


Figure 5-4 : Courbes obtenues lors de la compensation de la résistance de la microélectrode. A) Compensation insuffisante de la microélectrode. B) Bonne compensation de la microélectrode. C) Compensation exagérée de la microélectrode.

l'expérience. La calibration commence par la compensation de la capacité de la microélectrode. La microélectrode n'est pas suffisamment compensée quand le bruit du signal est anormalement faible et trop compensée quand le signal oscille. Une bonne calibration est obtenue lorsque le signal est sur le point d'osciller. La seconde étape de la calibration consiste à compenser la chute de tension qui se produira aux bornes de la microélectrode lors de l'injection d'un courant dans la cellule. Pour réussir à bien faire cet ajustement, il faut prendre soin de ne soustraire que la tension générée par la microélectrode lors de l'injection d'un courant dans la cellule. L'ajustement est possible, car la constante de temps de la microélectrode est beaucoup plus rapide que celle de la cellule (Finkel et Redman, 1985). La figure 5-4-B de la page précédente illustre les trois résultats qu'il est possible d'obtenir lors de l'injection d'un courant dans la cellule. Les cas A et C montrent respectivement une sous-compensation et une hypercompensation alors que le cas B est celui où la chute de tension dans la microélectrode est parfaitement compensée.

Enfin, la position de la microélectrode, lors de la recherche d'une cellule, est limitée à une plage allant de $-100\ \mu\text{m}$ à $100\ \mu\text{m}$ du centre de la pseudopupille cornéenne. Si la cellule trouvée est à l'extérieur de cette plage, la limitation physique de l'orientation de la base peut être atteinte et ainsi empêcher de bien aligner la cellule lors de la prochaine étape.

5.5 Alignement de la cellule sous enregistrement

Lorsque l'on trouve une cellule, il est très peu probable qu'elle fasse partie de la pseudopupille cornéenne d'origine. Il est donc nécessaire de déplacer et d'orienter la cellule pour qu'elle fasse partie de la pseudopupille cornéenne vue par le microscope. L'alignement est fait en deux temps. En premier lieu, la cellule sous enregistrement est centrée et orientée grossièrement et par la suite, un alignement fin est effectué.

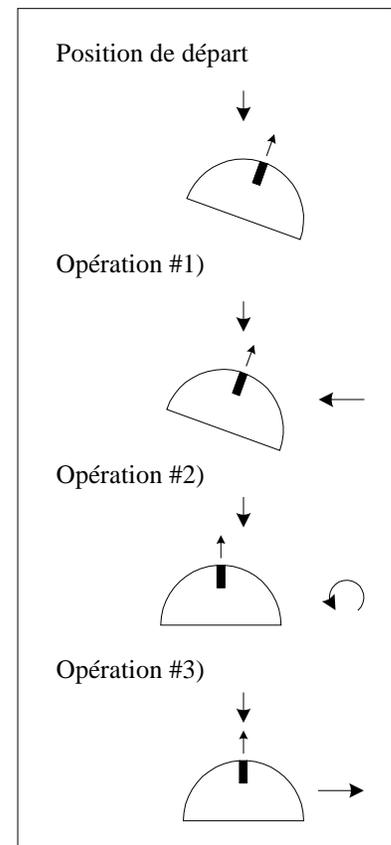


Figure 5-5 : Représentation des opérations de l'alignement grossier d'une cellule.

5.5.1 Alignement grossier de la cellule sous enregistrement

L'alignement grossier de la cellule est rendu possible grâce à la séquence d'opérations de la figure 5-5 de la page précédente. La première opération consiste simplement à exécuter l'algorithme de centrage d'une cellule avec le faisceau lumineux du trajet de fond pour centrer la cellule sous enregistrement sur l'axe optique du microscope. Avant d'exécuter l'algorithme, l'utilisateur doit prendre soin de diminuer au minimum l'iris du trajet de fond pour accélérer l'exécution de l'algorithme. Toutefois, le diamètre de l'iris doit être assez grand pour que la cellule soit éclairée. De plus, il n'est pas possible d'utiliser un autre faisceau que celui du trajet de fond, car l'angle initial entre ce faisceau lumineux et celui de la cellule est pratiquement toujours supérieur à 1° .

La deuxième opération consiste à faire tourner l'œil de l'insecte pour que l'angle entre la cellule et le faisceau lumineux soit le plus faible possible. Pour ce faire, l'utilisateur regarde la position de la pseudopupille cornéenne sur une image récemment prise et compte le nombre de facettes entre le centre de l'image, c'est-à-dire la cellule sous enregistrement, et la pseudopupille cornéenne visible. Ce nombre de facettes permet de déduire l'angle approximatif présent entre l'orientation de la cellule et le faisceau lumineux. Puisqu'il y a environ $1,5^\circ$ entre l'orientation de chacune des facettes (Hardie, 1985), la rotation de la base est donnée par le nombre de facettes multiplié par 1,5. L'orientation de l'angle qu'il faut appliquer est donnée par la droite passant par le centre de la pseudopupille cornéenne et le centre de l'image.

L'ajustement de l'orientation des angles ϕ et θ de la base est facilement fait à l'aide du système de coordonnées polaires de la figure 4-29 de la page 130, et l'opération à effectuer est présentée à la figure 5-6.

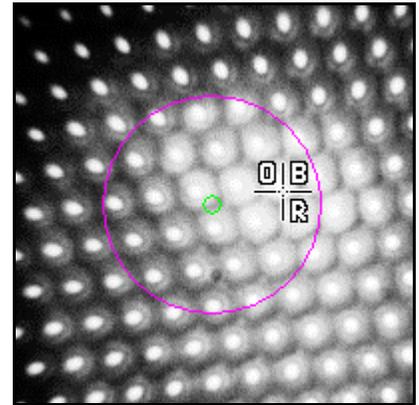


Figure 5-6 : Représentation de la technique utilisée pour changer l'orientation de la base. Il ne s'agit que de pointer le centre de la pseudopupille cornéenne réduite avec la souris en ayant préalablement sélectionné un changement d'orientation de la base en relatif.

La troisième opération consiste simplement à exécuter de nouveau l'algorithme de centrage d'une cellule pour la replacer au centre de l'axe optique du microscope, car elle a été déplacée lors de la rotation de la base. Si, après la présente opération, la pseudopupille cornéenne visible ne se retrouve pas au centre de l'image, c'est que l'angle évalué dans la deuxième étape n'était pas correct. Il ne reste alors qu'à raffiner l'alignement en recommençant les deuxième et troisième opérations jusqu'à ce que la pseudopupille cornéenne soit au centre de l'image.

Malheureusement, il n'est pas très utile d'utiliser l'algorithme d'alignement d'une cellule. Cette caractéristique vient du fait que le grand angle de convergence du faisceau lumineux du trajet de fond élimine les variations d'amplitude de la cellule lors de la rotation de la base.

5.5.2 Alignement fin de la cellule sous enregistrement

L'alignement fin de la cellule se fait simplement en exécutant l'algorithme d'alignement d'une cellule en utilisant un trajet lumineux avec un diamètre de 120 μm pour éclairer l'ensemble de la pseudopupille cornéenne et avec un angle de divergence de $0,6^\circ$. Il est alors possible d'exécuter l'algorithme d'alignement d'une cellule en demandant de tourner la base ou de changer l'angle du faisceau.

Cependant, il est préférable de changer l'alignement du faisceau lumineux, car la rotation de la base a comme conséquence de produire un déplacement latéral de la cellule qui peut la faire sortir du champ du faisceau lumineux. Par la suite, il faut transférer l'angle obtenu avec le faisceau lumineux à la base. Cela consiste simplement à prendre en note l'angle du faisceau lumineux et de l'ajouter à l'angle actuel de la base, puis de remettre l'angle du faisceau lumineux à zéro. Une commande permet de faire cette opération automatiquement dans l'application *Contrôle des expériences*.

L'alignement de la cellule prend fin en exécutant l'algorithme de centrage d'une cellule avec un faisceau d'environ 120 μm de diamètre pour aligner la cellule au centre de l'axe optique du microscope.

5.6 Identification des facettes de la pseudopupille cornéenne réduite

La dernière étape consiste à identifier les facettes appartenant à la pseudopupille cornéenne réduite à l'aide des marqueurs prévus à cet effet (voir figure 5-8, page 153). L'identification des facettes permet par la suite de stimuler individuellement un photorécepteur, les LMC d'une même cartouche et une cartouche (voir figure 5-7).

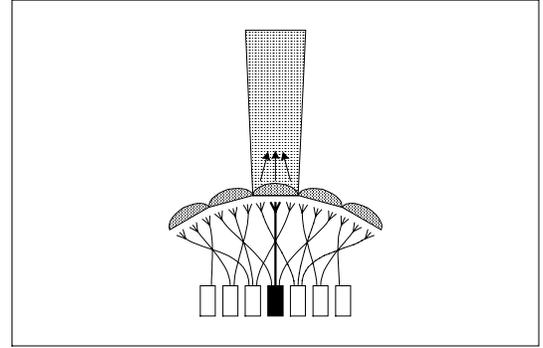


Figure 5-7 : Représentation de la stimulation d'un photorécepteur, des LMC d'une même cartouche et d'une cartouche avec un faisceau lumineux de 20 μm de diamètre et un angle de convergence de $0,6^\circ$.

Puisqu'il est possible d'enregistrer les réponses électriques de différents types de cellules il faut donc utiliser l'une des trois techniques présentées dans les sections qui suivent. Cependant, pour chacun des trois cas, il est important de ne pas utiliser une trop grande intensité lumineuse pour éviter qu'une diffusion de lumière excite les cellules voisines.

Dans le cas où il est possible de voir la pseudopupille cornéenne réduite, il est facile d'identifier directement les facettes. Les trois techniques qui suivent pourront cependant servir à vérifier l'exactitude de la pseudopupille cornéenne réduite.

5.6.1 Identification des facettes avec la réponse d'un photorécepteur

La technique utilisée pour identifier les facettes consiste à mesurer la réponse du photorécepteur lorsqu'on éclaire individuellement les différentes facettes de la pseudopupille cornéenne. Dans le présent cas, l'éclairement doit être fait avec un petit faisceau lumineux ayant un diamètre d'environ 20 μm et un angle de convergence de $0,6^\circ$.

L'identification des facettes est rendue possible grâce aux synapses électriques présentes entre les axones des photorécepteurs (voir section 6.2.5, page 161). Ces synapses ont comme effet de générer une faible dépolarisation dans le photorécepteur sous enregistrement quand on éclaire un photorécepteur voisin faisant partie de la même pseudopupille cornéenne réduite. Cependant, l'amplitude de la réponse varie selon la distance des voisins. Pour les deux voisins

immédiats, la réponse est un peu plus grande que pour les autres voisins, et la réponse générée par le troisième voisin est presque négligeable.

La première facette à être marquée est celle contenant le photorécepteur sous enregistrement qui est fortement dépolarisé lorsque la facette est éclairée. Dans le présent cas, la réponse est presque aussi grande que lors des étapes précédentes. Par la suite, la technique consiste à éclairer les deux premiers cercles de facettes entourant la facette contenant le photorécepteur sous enregistrement pour trouver les deux voisins immédiats. Déjà à l'aide de l'emplacement des deux voisins immédiats, il est possible de prédire l'emplacement des trois autres voisins. Cela est possible si l'on connaît l'orientation de la pseudopupille cornéenne réduite.

Puisque l'expérimentateur travaille toujours dans la même région de l'œil et qu'il place toujours l'insecte de la même façon sur le montage, il n'a qu'à déterminer l'orientation de la pseudopupille cornéenne réduite à l'aide de la pseudopupille profonde.

Après avoir identifié toutes les facettes faisant partie de la pseudopupille cornéenne réduite, l'utilisateur les numérote correctement (voir figure 5-8) et vérifie l'exactitude des facettes en les éclairant individuellement et en vérifiant s'il y a une réponse dans le photorécepteur sous enregistrement. Puisque l'amplitude générée par l'éclairage des photorécepteurs voisins est d'amplitude semblable au bruit électrique du montage, il est préférable de faire la vérification en effectuant la moyenne d'une douzaine de mesures.

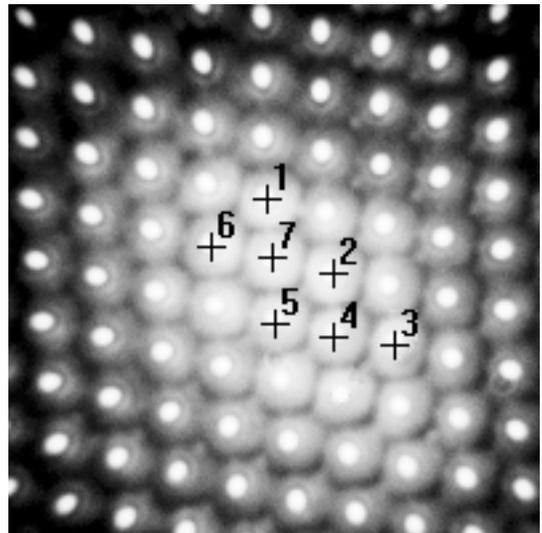


Figure 5-8 : Représentation du marquage des facettes participant à la superposition neuronale et faisant partie de la même pseudopupille cornéenne réduite.

Finalement, lorsque l'enregistrement est effectué dans les axones des photorécepteurs, l'amplitude

générée par les photorécepteurs voisins est plus grande que lors de l'enregistrement dans le segment interne. Donc, dans un tel cas, il n'est pas nécessaire de faire la moyenne de plusieurs mesures lors de la vérification.

Si le patron obtenu est différent de celui de la figure 5-8 de la page précédente, c'est que la structure de l'œil a été endommagée. Il faut alors recommencer la recherche d'une nouvelle cellule.

5.6.2 Identification des facettes avec la réponse d'une LMC

L'identification des facettes pour un enregistrement dans une LMC est beaucoup plus simple que pour le photorécepteur. Cette facilité vient du fait que tous les photorécepteurs de la pseudopupille cornéenne réduite font connexion avec les LMC de la cartouche et que les LMC génèrent toutes des réponses de grande amplitude.

Donc, la technique consiste à éclairer toutes les facettes de la pseudopupille cornéenne et à trouver les six facettes générant une réponse dans la LMC. Contrairement au cas des photorécepteurs, le patron est directement obtenu et doit être le même que pour les photorécepteurs (voir figure 5-8, page 153).

De plus, il n'est pas nécessaire de faire la moyenne de plusieurs mesures, car la réponse des LMC est suffisamment grande pour une intensité lumineuse assez forte. Si l'intensité lumineuse est plus faible, l'amplitude de la réponse sera plus faible que pour les étapes précédentes, mais quand même beaucoup plus grande que le bruit électrique du montage.

5.6.3 Identification des facettes avec la réponse d'une cartouche

L'identification des facettes de la pseudopupille cornéenne réduite se fait de façon pratiquement identique à la technique utilisée pour les LMC. La seule différence réside dans une réponse positive de la cartouche au lieu d'une réponse négative de la LMC. Encore une fois, un patron obtenu différent de celui de la figure 5-8 de la page 153 indique une anomalie dans la structure de l'œil de l'insecte.

5.7 Conclusion

Les étapes présentées dans le présent chapitre sont toutes nécessaires à la réalisation d'une expérience, à l'exception de la dernière étape qui est surtout utile pour l'expérience présentée au chapitre VI.

La réalisation des étapes semble simple et facile d'exécution, mais elle comporte plusieurs difficultés. À titre d'exemple, il arrive souvent que la pseudopupille cornéenne est pratiquement invisible, que l'algorithme d'alignement de la cellule ne fonctionne pas, car le faisceau sort du trajet lumineux, que l'angle couvert par le trajet lumineux est insuffisant, car le raffinement de l'alignement grossier de la cellule n'a pas été suffisant, ou qu'il y a perte de la réponse de la cellule sous enregistrement durant l'une des étapes. La perte de la cellule est provoquée par des mouvements du montage ou de l'animal qui font sortir la microélectrode de la cellule sous enregistrement. La perte peut aussi être occasionnée par le fait que la cellule sous enregistrement a été endommagée lors de l'expérimentation.

Cependant, une fois toutes les étapes réalisées, le travail de l'expérimentateur est terminé et l'ordinateur prend la relève pour compléter l'expérience. Pour ce faire, l'utilisateur n'a qu'à exécuter le protocole expérimental qu'il a préalablement établi (voir section 4.11.8, page 134). Par contre, l'utilisateur doit demeurer vigilant pour surveiller la perte éventuelle de la cellule sous enregistrement durant l'exécution du protocole.

CHAPITRE VI

ESTIMATION DES PARAMÈTRES D'UN MODÈLE REPRÉSENTANT L'INTERCONNEXION ENTRE SIX PHOTORÉCEPTEURS PARTICIPANT À LA MÊME SUPERPOSITION NEURONALE

6.1 Introduction

On peut trouver dans la littérature une multitude d'expériences qui proposent des solutions pour déterminer les différents aspects du fonctionnement du système visuel de la mouche. Celle de van Hateren (1986) a joué un rôle tout spécial dans le cadre de cette thèse. Son travail comprend une étude par microdissection optique des photorécepteurs participant à la même superposition neuronale.

Toutefois, le modèle de van Hateren fait appel à des structures distribuées, ce qui rend impossible l'identification de chacun des paramètres à l'aide de résultats expérimentaux. Dans une série de communications (Gagné et Comtois, 1993 à 1997), nous avons montré qu'il est possible d'obtenir un modèle plus simple qui permet de déterminer les paramètres à l'aide de mesures expérimentales. Ces communications présentent aussi une première série d'évaluations des paramètres obtenus à partir des résultats expérimentaux de van Hateren (voir section 6.5, page 174).

Nous mentionnons ici comment nous avons mis au point la technique permettant de déterminer les paramètres du modèle de Gagné et Comtois (1993 à 1997). Le chapitre est subdivisé en cinq sections distinctes. La première section présente le modèle électrique. La seconde section montre les différents types de stimulations qu'il est intéressant d'effectuer et les équations qui en résultent. La troisième section explique comment obtenir les équations mathématiques du modèle électrique et la façon de résoudre le système d'équations. Cette procédure va du même coup permettre de déterminer les mesures expérimentales nécessaires pour résoudre le modèle électrique. La quatrième section donne les valeurs des sept paramètres permettant d'expliquer les résultats expérimentaux de van Hateren (1986). La cinquième section présente une comparaison entre une première série de mesures obtenues à l'aide du présent montage et celles tirées du travail de van Hateren.

6.2 Le modèle électrique

De prime abord, le modèle électrique (voir figure 6-1) paraît complexe, car il est composé de plusieurs sources de tension et d'une multitude d'impédances qui correspondent aux paramètres inconnus que l'on veut identifier. Le modèle est en trois dimensions et de forme hexagonale où chaque coin supérieur de l'hexagone représente un des six photorécepteurs participant à la même superposition neuronale. Le centre de l'hexagone, dans le haut de la figure, correspond au milieu extracellulaire de la rétine et celui du bas de la figure, au milieu

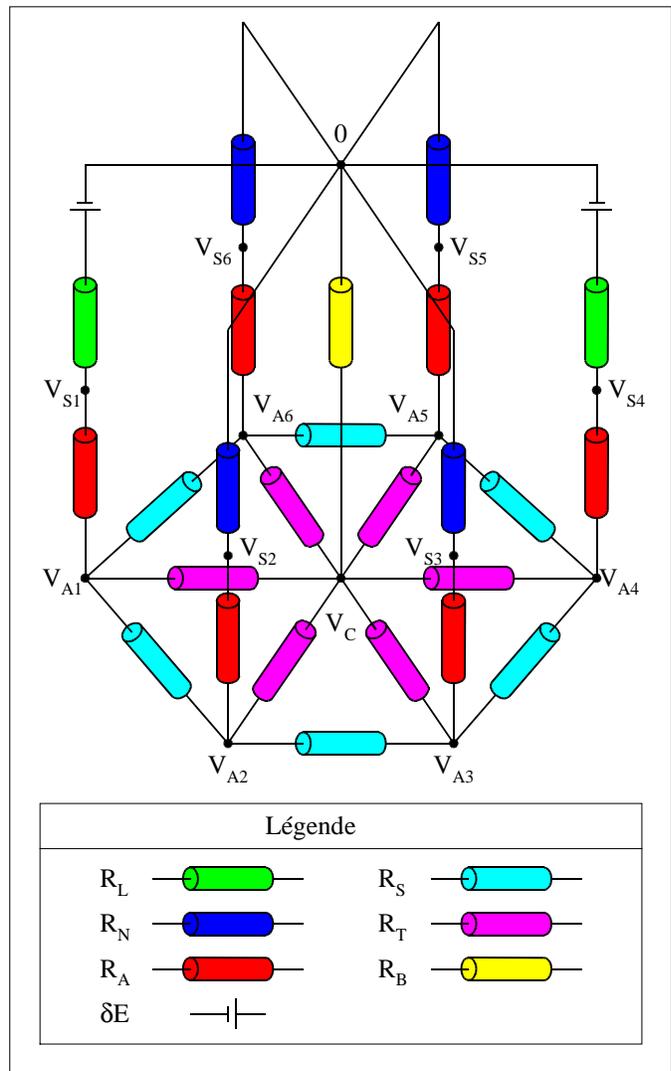


Figure 6-1 : Modèle électrique de l'organisation des photorécepteurs pour la stimulation lumineuse de deux d'entre eux.

intérieur de la cartouche. Les six coins inférieurs représentent chacun l'endroit où la tension dans les terminaisons des six photorécepteurs peut être mesurée.

La figure 6-2 permet de mieux visualiser la nature de chacun des paramètres du modèle électrique de la figure 6-1 de la page précédente. Nous voyons qu'une coupe perpendiculaire à l'œil permet d'identifier les sept paramètres présentés à la figure 6-1 de la page précédente. Notons qu'un des paramètres correspondant à R_N lorsque le photorécepteur est à la noirceur ou à R_L lorsqu'il est illuminé.

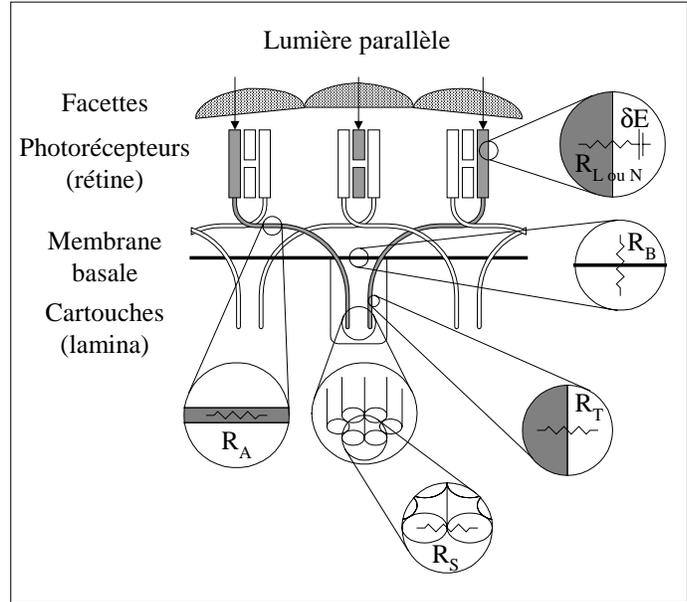


Figure 6-2 : Correspondance entre les paramètres du modèle électrique et le système visuel de la mouche.

6.2.1 Les paramètres R_L , δE et R_N

Le segment interne d'un photorécepteur comprend plusieurs conductances et sources de tension que l'on représente par un équivalent électrique. Cet équivalent comprend soit la résistance R_L et la source de

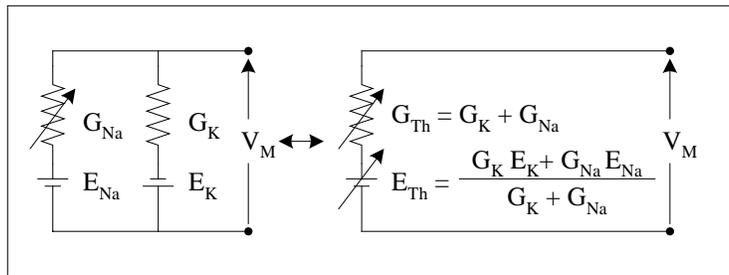


Figure 6-3 : Équivalent Thévenin du photorécepteur.

tension δE pour un photorécepteur stimulé par une lumière, ou soit la résistance R_N pour un photorécepteur non illuminé. L'équivalent électrique utilisé est celui de Thévenin (voir figure 6-3). Puisqu'il est très difficile de mesurer la tension absolue des cellules, nous choisissons une tension nulle pour tout récepteur non illuminé. Il s'ensuit qu'un photorécepteur non illuminé ne comprend qu'une résistance R_N sans source de tension. La source de tension δE ne représente donc pas la tension absolue du photorécepteur sous illumination, mais la

différence de tension entre un photorécepteur sous illumination et un photorécepteur non illuminé. On peut donc écrire :

$$\begin{aligned} R_N &= 1/G_{Th \text{ (NOIRCEUR)}}, \\ R_L &= 1/G_{Th \text{ (LUMIÈRE)}} \text{ et} \\ \delta E &= E_{Th \text{ (LUMIÈRE)}} - E_{Th \text{ (NOIRCEUR)}}. \end{aligned}$$

La figure 6-1 de la page 157 possède donc deux photorécepteurs sous illumination et quatre photorécepteurs à la noirceur. Pour que les valeurs de R_L et δE utilisées soient les mêmes, il faut que l'intensité lumineuse soit la même pour tous les photorécepteurs sous illumination.

6.2.2 Le paramètre R_A

Lors de la réalisation de son modèle, van Hateren (1986) a utilisé un modèle distribué pour représenter l'axone, ce qui n'est pas faux en soi. Cependant, l'utilisation d'un modèle distribué augmente le nombre de paramètres dans le modèle et le rend plus difficile, voire pratiquement impossible, à résoudre. Nous avons pu démontrer (Gagné et Comtois, 1993 à 1997) que la distribution des paramètres R_A n'était pas nécessaire et que le modèle distribué peut être remplacé par une seule impédance que l'on nomme R_A . Cette modification est très importante, car elle permet d'obtenir un modèle qui contient seulement sept inconnues.

La démonstration est faite à partir des équations de lignes de transmission présentée dans le livre « Electric current flow in excitable cells » (Jack et al., 1983). Deux preuves montrant qu'il n'est pas nécessaire d'utiliser le modèle distribué peuvent être données.

La première vient de la forme de la variation de la tension qui se développe dans la cellule suite à une injection d'un échelon de courant. Si après une durée d'une constante de temps, la tension a atteint 84 % de sa valeur en régime permanent, et que la forme de la réponse correspond à une erf (fonction d'erreur), une modélisation avec une ligne distribuée est alors adéquate. Par contre, si après une durée d'une constante de temps, la tension a atteint 63 % de sa valeur en régime permanent, et que la forme de la réponse correspond à une exponentielle, l'utilisation du modèle distribué n'est pas nécessaire. Or, van Hateren (1986) a montré (voir la

figure 10 de son article) que la réponse de la cellule suite à une injection de courant est bien représentée par une exponentielle.

La deuxième preuve consiste à prédire l'atténuation du signal dans l'axone en tenant compte de sa longueur, de sa constante de longueur et du type de terminaison que l'axone possède. La constante de longueur donne la distance à laquelle un signal transmis passivement sera atténué de 63 %. Gagné et Comtois (1993 à 1997) ont montré que l'atténuation du signal se situe entre 0,04 % et 5,4 % pour la longueur totale de l'axone. Même dans le pire cas, un modèle distribué n'est pas justifié car l'atténuation de 5,4 % est beaucoup plus faible que celle de 63 % et donc que la constante de longueur de l'axone est beaucoup plus grande que la longueur de l'axone.

L'impédance R_A représente donc la résistance qu'il y a entre le segment interne du photorécepteur et la terminaison du photorécepteur. Puisqu'il y a six photorécepteurs, le modèle possède six impédances R_A .

6.2.3 Le paramètre R_T

Le paramètre R_T représente l'impédance de la terminaison de l'axone du photorécepteur. La chute de tension générée aux bornes de cette impédance est importante, car il s'agit de la tension vue, à travers une synapse chimique, par les grosses cellules monopolaires. Il est important de réaliser que la tension mesurée dans le segment interne ne correspond pas la tension vue par les grosses cellules monopolaires.

Les impédances R_T sont au nombre de six et ce sont elles qui représentent les liens entre l'intérieur de chacun des axones des six photorécepteurs et l'intérieur de la cartouche.

6.2.4 Le paramètre R_B

L'avant-dernière impédance R_B fait simplement le lien entre l'intérieur de la cartouche et le milieu extracellulaire de la rétine. Contrairement aux autres impédances, elle est unique et commune à tous les photorécepteurs.

6.2.5 Le paramètre R_S

Les terminaisons des photorécepteurs sont aussi munies de synapses électriques, représentées par R_S , qui permettent de lier un photorécepteur à ses deux voisins. Ce sont ces synapses électriques qui nous amènent à vouloir comprendre le fonctionnement d'un tel circuit et sur son rôle dans les traitements effectués par le système visuel de la mouche.

Il y a donc six impédances R_S qui font le lien entre les terminaisons des photorécepteurs voisins.

6.3 Extraction des équations selon différentes stimulations

Afin de réussir à identifier les sept inconnues du modèle, il faut trouver un ensemble de sept équations indépendantes qu'il serait possible de solutionner à partir d'enregistrements obtenus dans une même cellule, car les valeurs des enregistrements varient d'une cellule à l'autre.

Il est donc impossible de mesurer toutes les tensions des segments internes ou des axones de tous les photorécepteurs simultanément. Toutefois, il est possible de déplacer le stimulus lumineux au lieu de la microélectrode de même qu'il est possible de stimuler plusieurs photorécepteurs simultanément pour générer de nouvelles équations. Par contre, le choix des photorécepteurs stimulés est fait de façon à simplifier le modèle résultant.

La structure hexagonale des photorécepteurs nous permet d'identifier cinq structures de base selon l'illumination. Nous retrouvons celle pour l'illumination d'un photorécepteur, de deux photorécepteurs, de trois photorécepteurs et de six photorécepteurs. La dernière structure ne subit aucune illumination.

Pour bien représenter les tensions des nœuds des modèles et les différentes équations, nous utiliserons la notation présentée au tableau 6-1 de la page suivante. Notons que le système d'indices est basé sur un enregistrement fait dans le premier photorécepteur. Pour un enregistrement dans un autre photorécepteur, il faudra voir les indices comme des valeurs relatives au photorécepteur sous enregistrement.

Tableau 6-1 : Notation des tensions du modèle, des équations de tension et des équations d'impédance.

V_{Sa}	Tension du segment interne pour le photorécepteur a.
Z_{Sa}	Impédance du segment interne pour le photorécepteur a.
V_{Aa}	Tension de l'axone pour le photorécepteur a.
Z_{Aa}	Impédance de l'axone pour le photorécepteur a.
V_C	Tension de la cartouche.
$V_{Ssabc\dots}$	Tension du segment interne du premier photorécepteur (V_{S1}) pour une stimulation des photorécepteurs a, b, c, etc.
$Z_{Ssabc\dots}$	Impédance du segment interne du premier photorécepteur (V_{S1}) pour une stimulation des photorécepteurs a, b, c, etc.
$V_{Asabc\dots}$	Tension de l'axone du premier photorécepteur (V_{A1}) pour une stimulation des photorécepteurs a, b, c, etc.
$Z_{Asabc\dots}$	Impédance de l'axone du premier photorécepteur (V_{A1}) pour une stimulation des photorécepteurs a, b, c, etc.
$V_{Csabc\dots}$	Tension de la cartouche (V_C) pour une stimulation des photorécepteurs a, b, c, etc.

Les cinq sous-sections qui suivent donnent les équations qu'il est possible d'obtenir pour les différentes stimulations lumineuses et injections de courant. La sélection finale des équations sera faite dans la section 6.4.2 de la page 173, où seules les équations nécessaires à la résolution du modèle seront conservées.

6.3.1 Stimulation lumineuse d'un seul photorécepteur avec ou sans injection de courant dans ce même photorécepteur

Lors de la stimulation d'un seul photorécepteur, le courant dans le photorécepteur voisin de gauche est le même que dans le photorécepteur voisin de droite, et le courant dans le deuxième voisin de gauche est le même que dans le deuxième voisin de droite. Cela nous donne un axe de symétrie qui est exploité pour réduire le circuit original. Le modèle simplifié est donné dans la partie droite de la figure 6-4 de la page suivante. Le circuit résultant a les mêmes tensions que le circuit de départ, mais a des impédances différentes.

La simplification du circuit résulte en une division par un facteur deux de la valeur de certaines impédances. Nous avons inscrit à l'intérieur de ces impédances de la figure 6-4 le facteur multiplicatif obtenu.

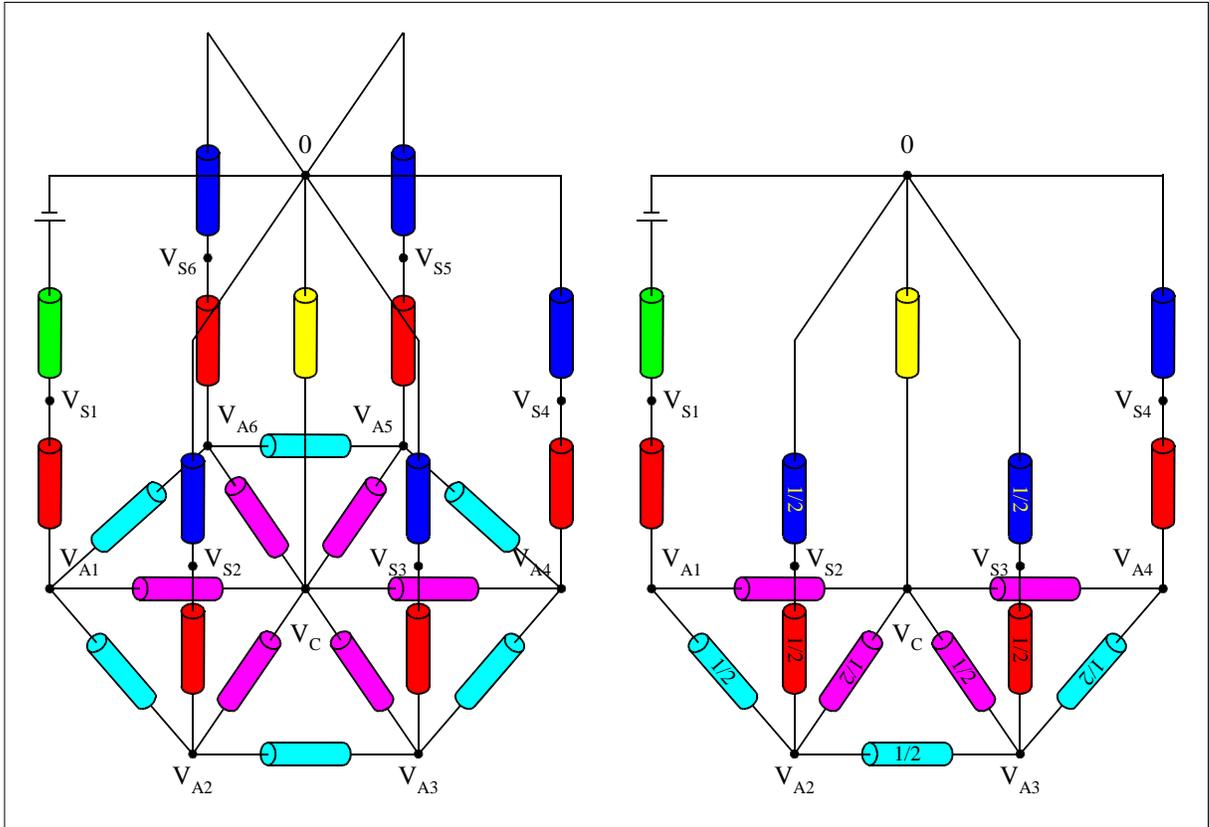


Figure 6-4 : Modèle simplifié pour la stimulation lumineuse d'un seul photorécepteur avec ou sans injection de courant dans ce même photorécepteur.

Le circuit simplifié nous donne une série de neuf équations. Les quatre premières équations proviennent des enregistrements dans les quatre segments internes du circuit simplifié (V_{S1} , V_{S2} , V_{S3} et V_{S4}). Les quatre équations suivantes résultent des enregistrements dans les quatre axones du circuit simplifié (V_{A1} , V_{A2} , V_{A3} et V_{A4}). La dernière équation est obtenue pour un enregistrement dans la cartouche (V_C).

Notons que les enregistrements à un même niveau, segment interne ou axone, peuvent être obtenus en faisant un déplacement virtuel de la microélectrode tout en la laissant dans le photorécepteur d'enregistrement. Puisque le circuit est symétrique, l'action d'enregistrer dans un photorécepteur et de stimuler le second photorécepteur revient exactement à enregistrer

dans le voisin et à stimuler ce propre photorécepteur. Donc, les équations donnant les tensions V_{S1} , V_{S2} , V_{S3} , V_{S4} , V_{A1} , V_{A2} , V_{A3} , V_{A4} et V_C deviendront respectivement les équations V_{SS1} , V_{SS2} , V_{SS3} , V_{SS4} , V_{AS1} , V_{AS2} , V_{AS3} , V_{AS4} et V_{CS1} .

Retenons que les équations V_{SS2} , V_{SS3} , V_{AS2} et V_{AS3} sont respectivement les mêmes que les équations V_{SS6} , V_{SS5} , V_{AS6} et V_{AS5} et que l'équation V_{CS1} est la même que les équations V_{CS2} , V_{CS3} , V_{CS4} , V_{CS5} et V_{CS6} .

Il est aussi possible d'effectuer des mesures d'impédance, en plus des mesures de tension, dans le segment interne ou dans l'axone. Toutefois, ces mesures ne peuvent être faites que dans un photorécepteur situé sur l'axe de symétrie du circuit. Nous obtenons donc les mesures d'impédance Z_{S1} et Z_{A1} qui généreront les équations Z_{SS1} et Z_{AS1} .

6.3.2 Stimulation lumineuse de deux photorécepteurs

Afin d'obtenir de nouvelles équations, il est aussi possible de stimuler deux photorécepteurs simultanément. Cependant, puisque nous désirons simplifier au minimum le circuit, nous devons stimuler deux photorécepteurs opposés. Nous utiliserons donc les photorécepteurs 1 et 4, les photorécepteurs 2 et 5 ou les photorécepteurs 3 et 6. De telles stimulations réduisent le circuit original d'un facteur quatre (voir figure 6-5, page suivante).

Il est possible de réaliser cinq mesures dans ce circuit. Il y a deux mesures dans le segment externe (V_{S1} et V_{S2}), deux autres dans l'axone (V_{A1} , V_{A2}) et une autre dans la cartouche (V_C). Tout comme pour la stimulation d'un seul photorécepteur, nous pouvons intervertir les équations pour que la microélectrode demeure toujours dans le même photorécepteur lors d'une expérience. Les équations des tensions des nœuds V_{S1} , V_{S2} , V_{A1} , V_{A2} et V_C deviennent respectivement les équations V_{SS14} , V_{SS25} , V_{AS14} , V_{AS25} et V_{C14} .

Notons que les équations V_{SS25} et V_{AS25} sont respectivement les mêmes que les équations V_{SS36} et V_{AS36} et que l'équation V_{C14} est la même que les équations V_{C25} et V_{C26} .

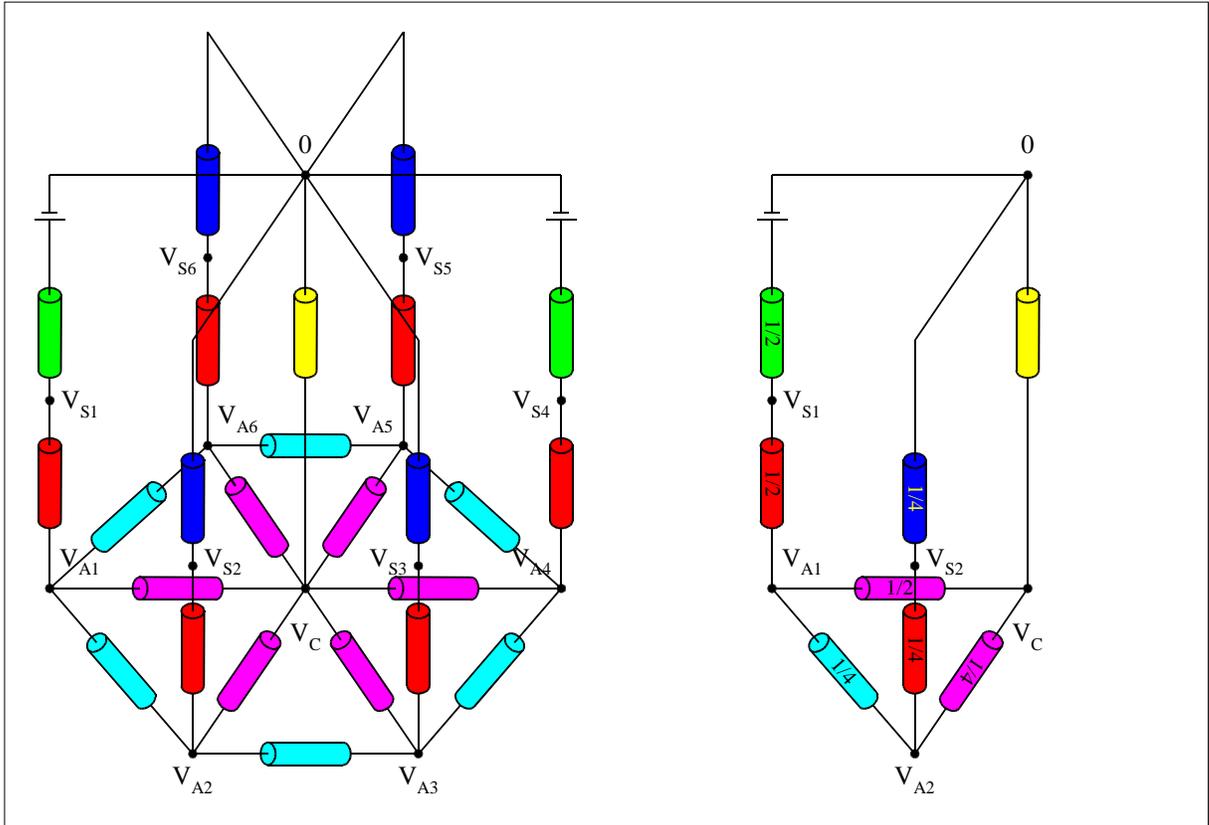


Figure 6-5 : Modèle simplifié pour la stimulation lumineuse de deux photorécepteurs.

6.3.3 Stimulation lumineuse de trois photorécepteurs

Pour augmenter davantage le nombre d'équations, il est aussi possible de stimuler trois photorécepteurs simultanément. Afin de simplifier au minimum le circuit, nous devons stimuler trois photorécepteurs isolés les uns des autres par un photorécepteur non stimulé. Nous obtenons ainsi deux combinaisons : les photorécepteurs 1, 3 et 5 ou les photorécepteurs 2, 4 et 6. De telles stimulations nous donnent le circuit simplifié de la partie droite de la figure 6-6 de la page suivante.

Il nous est possible d'effectuer cinq mesures dans ce circuit. Il y a deux mesures dans le segment interne (V_{S1} et V_{S2}), deux mesures dans l'axone (V_{A1} , V_{A2}) et une mesure dans la cartouche (V_C). Tout comme pour la stimulation d'un seul photorécepteur, nous pouvons intervertir les équations pour que la microélectrode demeure toujours dans le même photorécepteur lors d'une expérience. Donc, les équations des tensions des nœuds V_{S1} , V_{S2} ,

V_{A1} , V_{A2} et V_C deviennent respectivement les équations V_{SS135} , V_{SS246} , V_{AS135} , V_{AS246} et V_{C135} .

Notons que l'équation V_{C135} est la même que l'équation V_{C246} .

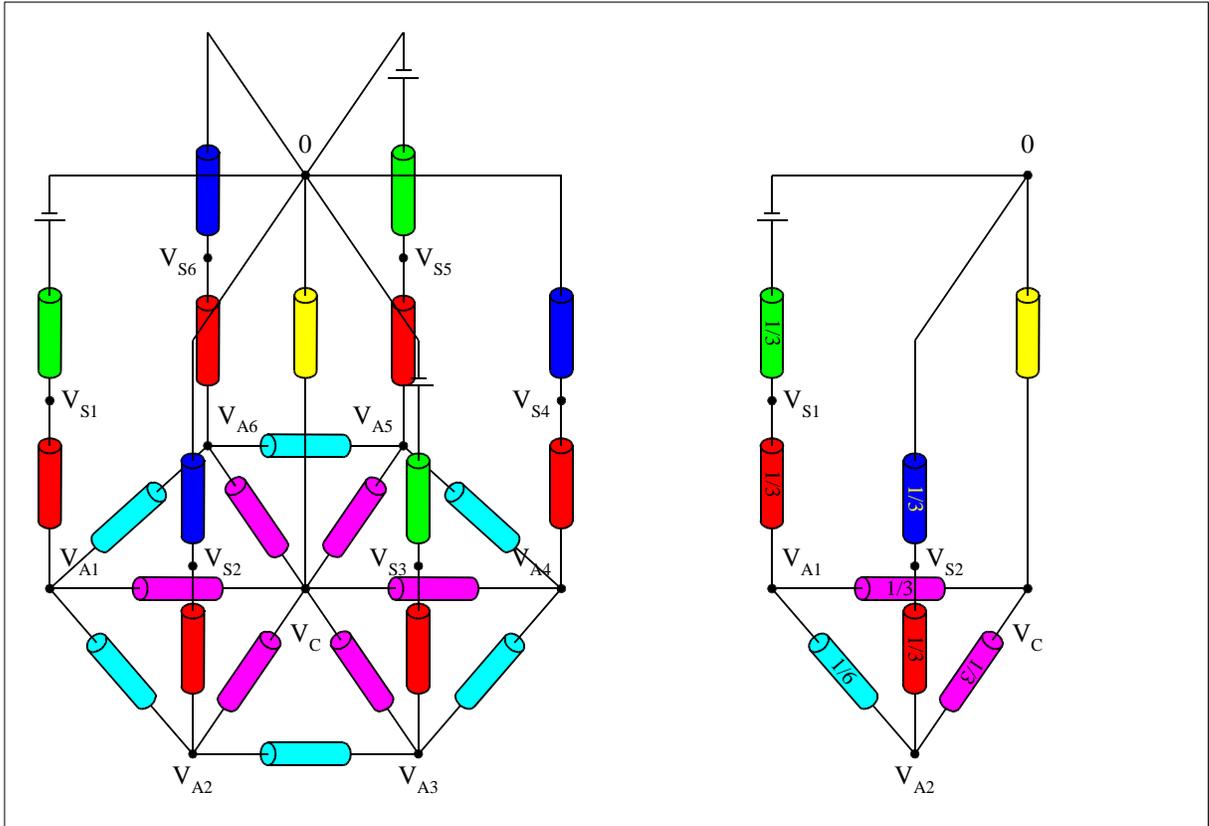


Figure 6-6 : Modèle simplifié pour la stimulation lumineuse de trois photorécepteurs.

6.3.4 Stimulation lumineuse des six photorécepteurs avec ou sans injection de courant dans un seul photorécepteur

La simplification maximale que nous pouvons faire pour augmenter le nombre d'équations consiste à stimuler six photorécepteurs simultanément. Avec une telle stimulation, nous retrouvons le circuit de la figure 6-7 de la page suivante.

Nous pouvons obtenir trois mesures de ce circuit : une dans le segment interne (V_{S1}), une deuxième dans l'axone (V_{A1}) et une autre dans la cartouche (V_C). Donc, les équations des

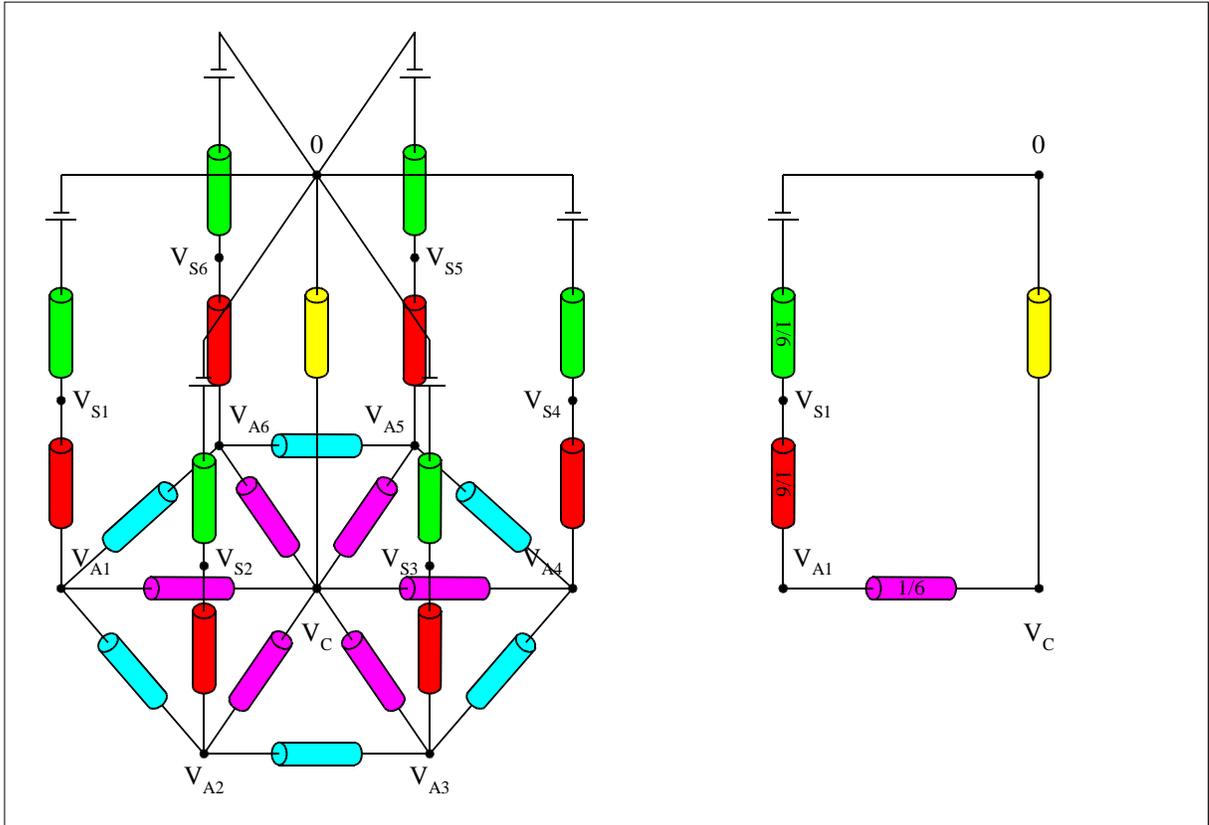


Figure 6-7 : Modèle simplifié pour la stimulation lumineuse des six photorécepteurs.

tensions des nœuds V_{S1} , V_{A1} et V_C deviennent respectivement les équations $V_{SS123456}$, $V_{AS123456}$ et $V_{C123456}$.

Tout comme pour la stimulation d'un seul photorécepteur, il est possible d'effectuer des mesures d'impédance lors de la stimulation de six photorécepteurs. Nous pouvons avoir une mesure d'impédance dans le segment interne (Z_{S1}) et une autre dans l'axone (Z_{A1}). Ces deux mesures d'impédance donnent respectivement les équations $Z_{SS123456}$ et $Z_{AS123456}$. Cependant, à cause de l'effet de brisure de symétrie causé par l'injection d'un courant, le circuit des impédances obtenu dans ce cas aura le même niveau de complexité que lors de la stimulation d'un seul photorécepteur (voir figure 6-8, page suivante).

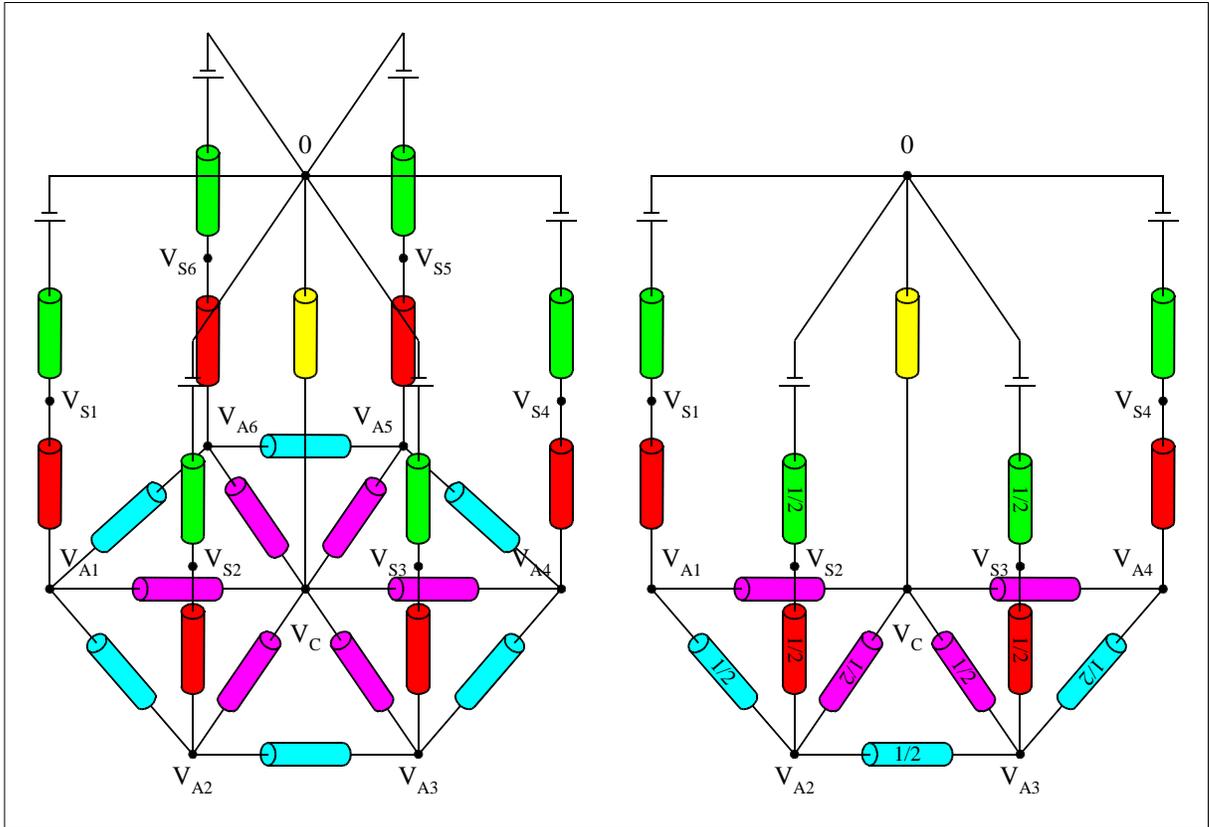


Figure 6-8 : Modèle simplifié pour la stimulation lumineuse des six photorécepteurs avec l'injection de courant dans un seul photorécepteur.

6.3.5 Injection de courant dans un seul photorécepteur sans stimulation lumineuse

La seule injection de courant dans un photorécepteur nous permet d'obtenir deux autres équations. Puisqu'il n'y a pas de stimulation lumineuse, il n'est pas possible d'enregistrer de tension valable dans le circuit, mais rien ne nous empêche d'effectuer une mesure d'impédance (voir figure 6-9, page suivante).

La mesure de l'impédance dans le segment interne (Z_{S1}) nous permet d'obtenir l'équation Z_{SS0} et la mesure de l'impédance dans l'axone (Z_{A1}) nous permet d'obtenir l'équation Z_{AS0} . Dans les expressions des équations, le zéro est utilisé pour indiquer qu'il n'y a aucun photorécepteur stimulé.

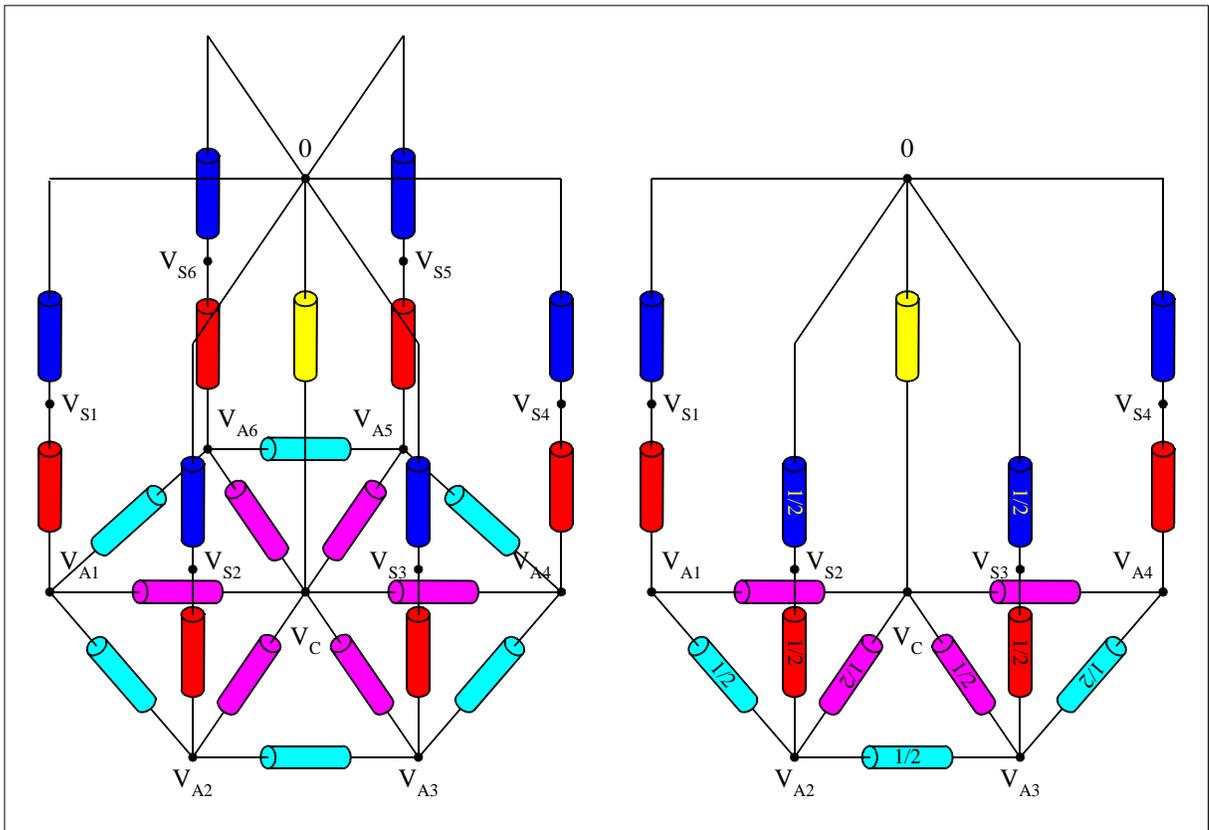


Figure 6-9 : Modèle simplifié pour l'injection de courant dans un photorécepteur sans stimulation lumineuse.

6.3.6 Discussion sur les équations obtenues

La sélection des équations utilisées s'est faite de façon à choisir celles qui sont les moins complexes. Il est évident qu'il y a beaucoup d'autres équations qui pourraient être tirées du modèle. Cependant, sans la simplification attribuable à la symétrie du modèle, les équations seraient beaucoup plus longues et difficiles à manipuler.

Nous avons choisi trois séries d'équations qui correspondent à trois points d'enregistrement distinct, soit le segment interne (voir tableau 6-2, page 170), l'axone (voir tableau 6-3, page 170) et la cartouche (voir tableau 6-4, page 171).

Tableau 6-2 : Paramètres du segment interne.

V_{SS1}	Tension du segment interne lors de la stimulation du premier photorécepteur.
V_{SS2}	Tension du segment interne lors de la stimulation du deuxième photorécepteur.
V_{SS3}	Tension du segment interne lors de la stimulation du troisième photorécepteur.
V_{SS4}	Tension du segment interne lors de la stimulation du quatrième photorécepteur.
Z_{SS1}	Impédance du segment interne lors de la stimulation du premier photorécepteur.
V_{SS14}	Tension du segment interne lors de la stimulation du premier et du troisième photorécepteur.
V_{SS25}	Tension du segment interne lors de la stimulation du deuxième et cinquième photorécepteur.
V_{SS135}	Tension du segment interne lors de la stimulation du premier, du troisième et du cinquième photorécepteur.
V_{SS246}	Tension du segment interne lors de la stimulation du deuxième, du quatrième et du sixième photorécepteur.
$V_{SS123456}$	Tension du segment interne lors de la stimulation des six photorécepteurs.
$Z_{SS123456}$	Impédance du segment interne lors de la stimulation des six photorécepteurs.
Z_{SS0}	Impédance du segment interne en l'absence de stimulation d'un photorécepteur.

Tableau 6-3 : Paramètres de l'axone.

V_{AS1}	Tension de l'axone lors de la stimulation du premier photorécepteur.
V_{AS2}	Tension de l'axone lors de la stimulation du deuxième photorécepteur.
V_{AS3}	Tension de l'axone lors de la stimulation du troisième photorécepteur.
V_{AS4}	Tension de l'axone lors de la stimulation du quatrième photorécepteur.
Z_{AS1}	Impédance de l'axone lors de la stimulation du premier photorécepteur.
V_{AS14}	Tension de l'axone lors de la stimulation du premier et du troisième photorécepteur.
V_{AS25}	Tension de l'axone lors de la stimulation du deuxième et cinquième photorécepteur.
V_{AS135}	Tension de l'axone lors de la stimulation du premier, du troisième et du cinquième photorécepteur.
V_{AS246}	Tension de l'axone lors de la stimulation du deuxième, du quatrième et du

	sixième photorécepteur.
$V_{AS123456}$	Tension de l'axone lors de la stimulation des six photorécepteurs.
$Z_{AS123456}$	Impédance de l'axone lors de la stimulation des six photorécepteurs.
ZAS0	Impédance de l'axone en l'absence de stimulation d'un photorécepteur.

Tableau 6-4 : Paramètres de la cartouche.

V_{CS1}	Tension de la cartouche lors de la stimulation du premier photorécepteur.
V_{CS14}	Tension de la cartouche lors de la stimulation du premier et du quatrième photorécepteur.
V_{CS135}	Tension de la cartouche lors de la stimulation du premier, du troisième et du cinquième photorécepteur.
$V_{CS123456}$	Tension de la cartouche lors de la stimulation des six photorécepteurs.

À première vue, la dernière série semble inutile, car elle comprend moins de sept équations et ne peut donner de solution sans un déplacement de la microélectrode dans une autre cellule. Cependant, nous verrons que, lors de l'inversion du modèle, nous aurons seulement six équations indépendantes pour chacune des deux premières séries. Il faudra donc utiliser une équation dans la troisième série pour obtenir sept équations indépendantes.

6.4 Résolution du modèle

La résolution du modèle est subdivisée en deux étapes. La première consiste à retrouver les équations de la section précédente, et la seconde consiste à prendre les équations et à les résoudre pour obtenir les valeurs des impédances en fonction des tensions.

Puisque nous pourrions avoir une série d'équations avec des enregistrements dans le segment interne et une seconde série avec des enregistrements dans l'axone, il serait donc possible d'obtenir les paramètres du modèle à partir de différentes mesures expérimentales. Ces deux possibilités de solutions sont complémentaires et pourraient être utilisées pour comparer les solutions obtenues et du même coup valider le modèle. Cependant, seule la résolution de la première série de résultats a été réalisée et est présentée.

6.4.1 Génération des équations

La première étape consiste à écrire les équations. Nous avons utilisé la technique des graphes de fluence, car elle permet d'écrire le circuit sous une forme qui simplifie l'analyse. Puisque le modèle est assez complexe, la réalisation d'un logiciel permettant de générer les équations a été nécessaire.

Nous explicitons la technique utilisée pour l'obtention de l'équation V_{SS1} . La première opération consiste à utiliser le modèle (voir figure 6-10) et d'y apposer les tensions et les courants nécessaires à l'écriture du graphe de fluence.

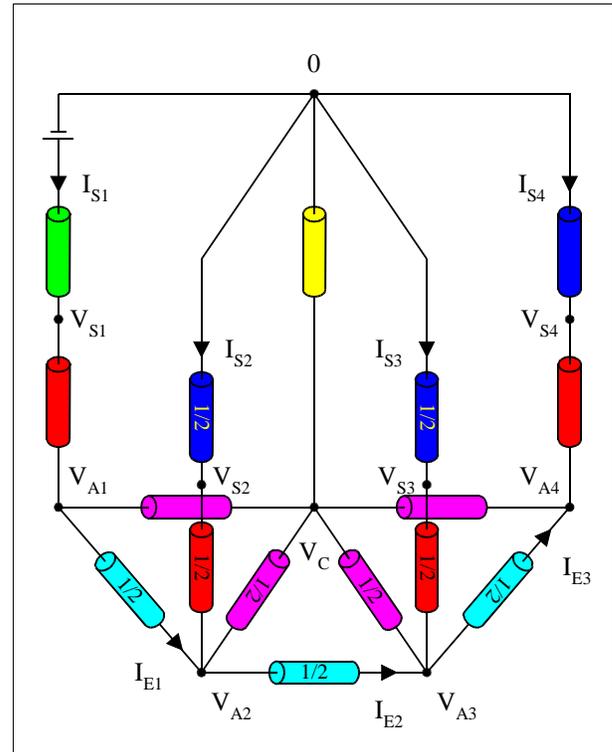


Figure 6-10 : Modèle représentant la stimulation lumineuse d'un seul photorécepteur et comprenant les tensions et courants nécessaires à la réalisation du graphe de fluence.

Le graphe de fluence (voir figure 6-11, page

suivante) est par la suite construit en prenant soin de représenter chacune des tensions et chacun des courants du circuit par des nœuds. Les flèches, quant à elles, représentent les relations entre les courants et les tensions.

Par la suite, un fichier contenant la description du graphe de fluence ainsi que les équations voulues est créé. Le fichier de description du graphe est très simple et comprend seulement la liste des nœuds où chaque nœud est suivi de la liste des flèches émanant de ce dernier. Le fichier des équations contient la liste des nœuds dont on désire obtenir l'équation. Cependant, il faut aussi donner pour chaque nœud une source de tension ou de courant de laquelle sera dérivée l'équation. Dans le cas de l'équation V_{SS1} , il faut demander l'équation du nœud V_{S1} par rapport au nœud δE .

Le logiciel génère donc un fichier contenant l'équation dans un format difficile à interpréter pour un humain, mais qui peut être directement lu par le logiciel Mathematica de la compagnie Wolfram Research Inc. À l'aide de la commande FullSimplify du logiciel Mathematica, nous obtenons les équations présentées à l'annexe C.

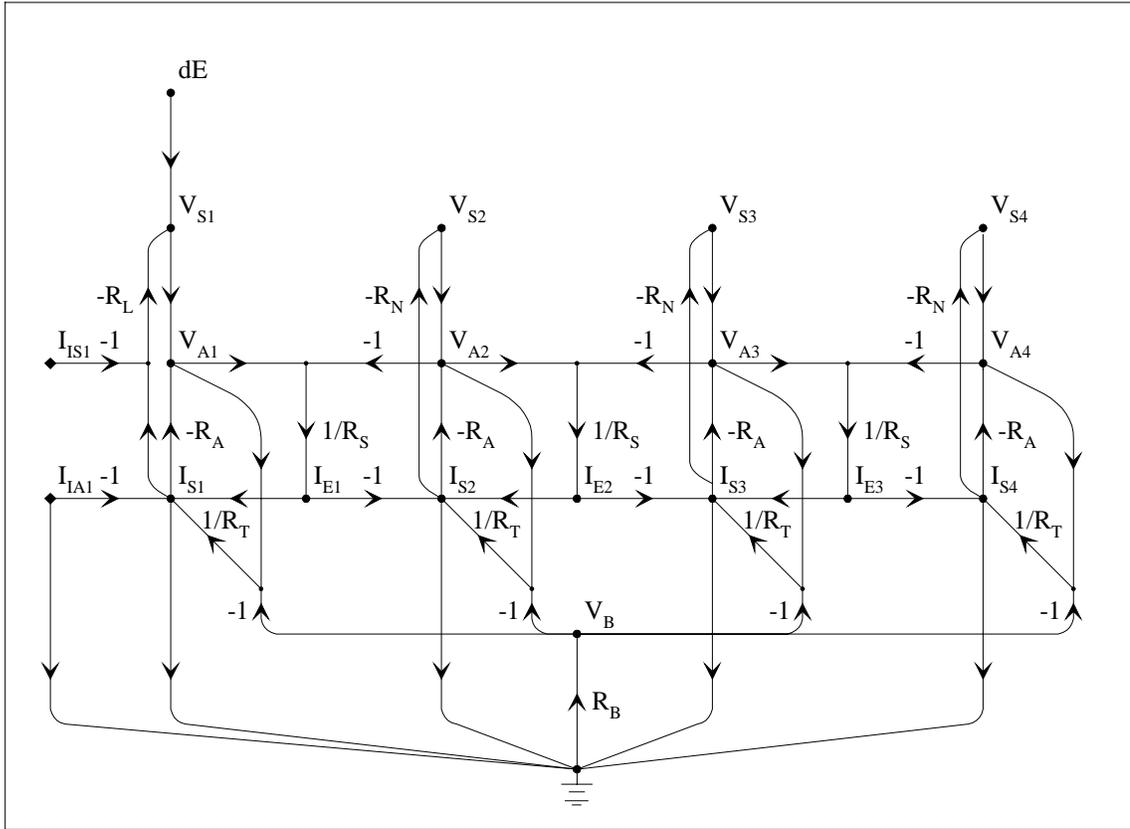


Figure 6-11 : Graphe de fluence du modèle pour la stimulation lumineuse et l'injection de courant pour un seul photorécepteur.

Nous pouvons remarquer (voir annexe C) que les équations générées sont assez longues et justifient amplement la création d'un logiciel pour analyser les graphes de fluence ainsi que l'utilisation du logiciel Mathematica pour les manipuler.

Les autres graphes de fluence sont présentés à l'annexe C.

6.4.2 Résolution du système d'équations

La résolution du système d'équations permettra d'obtenir les paramètres en fonction des mesures expérimentales. La technique est très simple et consiste à extraire successivement les

paramètres des équations et à les remplacer dans les autres équations. L'ordre d'extraction des paramètres n'est pas aléatoire et est réalisé de façon à minimiser la complexité des équations.

La résolution du système d'équations est d'abord faite de façon symbolique, mais il faut travailler avec des valeurs numériques pour obtenir le dernier paramètre. Le processus est subdivisé en sept itérations où chaque itération permet d'isoler un paramètre qui est par la suite substitué dans les autres équations. Au fur et à mesure de la progression, le nombre d'équations diminue, car les équations utilisées pour extraire un paramètre ne peuvent être utilisées plus d'une fois. La procédure complète est donnée à l'annexe C.

Une fois les équations des paramètres du modèle obtenues, il est facile d'estimer les sept paramètres de celui-ci. L'ordre d'estimation des paramètres est important : il faudra commencer par l'évaluation de R_T , puis insérer le résultat dans l'équation de R_S , ce qui permet d'évaluer R_S , et ainsi de suite jusqu'à l'évaluation de δE . La forme des sept équations se lit comme suit (les équations complètes sont données à l'annexe C) :

$$\begin{aligned}\delta E &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\ R_N &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}) \\ R_L &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\ R_B &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\ R_A &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\ R_S &= f(R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}) \\ R_T &= f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}, Z_{S123456})\end{aligned}$$

L'évaluation des paramètres du modèle est donc possible si nous avons les mesures expérimentales de V_{SS25} , V_{SS14} , $V_{SS123456}$, V_{SS246} , V_{SS1} , V_{CS1} et $Z_{S123456}$.

6.5 Estimation des paramètres à partir des résultats expérimentaux de van Hateren

Lors de l'estimation des paramètres du modèle avec les résultats expérimentaux de van Hateren (1986), nous ne pouvons pas utiliser l'approche analytique développée précédemment, car les valeurs des mesures disponibles ne correspondent pas à celles de la série d'équations indépendantes que nous avons obtenue.

Les mesures utilisées et extraites du travail de van Hateren sont données dans le tableau 6-5. Puisqu'il n'a pas été possible d'utiliser seulement des mesures faites au niveau du segment interne du photorécepteur, nous avons aussi utilisé des mesures effectuées dans l'axone du photorécepteur et dans la cartouche. Toutefois, pour minimiser les erreurs attribuables à des enregistrements provenant de différentes cellules, nous prenons les rapports de deux mesures lorsque l'on utilise des mesures de l'axone ou de la cartouche. Bien que l'article de van Hateren (1986) contienne une multitude de résultats expérimentaux, couvrant une plage d'intensité lumineuse allant de 2 à 34 (unités relatives), nous avons dû nous limiter aux résultats correspondant à une intensité de 10. Seules les mesures à cette intensité comprennent des résultats de quantités suffisants pour évaluer les paramètres de façon acceptable.

Tableau 6-5 : Liste des tensions utilisées pour l'évaluation des paramètres à partir des résultats expérimentaux de van Hateren.

Paramètres	Valeurs
V_{SS1}	2.7 mV
V_{SS2}	0.6 mV
$V_{SS123456}$	3.6 mV
$V_{SA123456} / V_{SA1}$	1.5
$V_{CS123456} / V_{CS1}$	4.2
Z_{SS0}	28 M Ω
$Z_{SS123456}$	11.5 M Ω

Lors de l'évaluation des paramètres, nous avons minimisé la fonction d'erreur relative suivante :

$$Err = \min_{i=1}^7 \left(\left| \frac{M_i - E_i}{M_i} \right|^2 \right)$$

où M_i correspond à une des sept mesures expérimentales

et E_i correspond aux valeurs estimées pour les sept équations.

L'utilisation de l'erreur relative permet d'accorder le même poids à chaque mesure. Pour minimiser l'erreur, il s'agit simplement de faire varier les différents paramètres et d'évaluer la fonction d'erreur pour les différentes combinaisons des paramètres. La combinaison qui donnera l'erreur la plus faible nous fournira les meilleures valeurs des paramètres (voir tableau 6-6, page suivante). Il va de soi que cette technique est loin d'être optimale et

demande beaucoup de temps de calcul vu la longueur des équations. Afin de diminuer ce temps de calcul, la région de recherche a été limitée à des valeurs positives de paramètres. De plus, les régions de recherche ont été délimitées pour n'utiliser que des valeurs plausibles de paramètres. Cela nous a donc permis d'avoir une première approximation des paramètres.

Tableau 6-6 : Liste des paramètres obtenus à partir des résultats expérimentaux de van Hateren.

Paramètres	Valeurs
R_A	10 M Ω
R_B	11 M Ω
R_L	17 M Ω
R_N	90 M Ω
R_S	100 M Ω
R_T	40 M Ω
δE	4.0 mV

6.6 Comparaison entre une première série de résultats et ceux tirés de l'article de van Hateren

Malgré la sophistication du montage présenté dans la présente thèse, il a été difficile d'amasser les résultats expérimentaux nécessaires à l'évaluation des paramètres du modèle. Parmi le grand nombre d'expérimentations réalisées, une seule série de données semble valable. Cette série de données a été obtenue par monsieur Christian Gourdeau (étudiant au deuxième cycle de Simon Gagné) à l'aide du protocole de l'annexe B.

Pour déterminer si une série est valable, nous utilisons les critères suivants :

- Lors d'une expérience, nous comparons les réponses de la cellule à un même stimulus lumineux, les unes au début et les autres à la fin de l'expérience. Si les deux enregistrements ont plus d'un certain pourcentage de différence, la série de données est rejetée.
- L'amplitude du photorécepteur sous enregistrement doit être supérieure à celle des deux voisins immédiats, et ces trois amplitudes doivent être supérieures à celles des autres voisins.
- Le dernier test consiste à vérifier si la réponse à un stimulus lumineux du photorécepteur sous enregistrement est plus faible que la réponse de ce même photorécepteur stimulé en

même temps que tous les autres photorécepteurs. Si nous ajoutons un autre photorécepteur à ceux précédemment stimulés, la réponse devrait croître de plus en plus.

La série de mesures répondant aux critères précédents et réalisée par monsieur Christian Gourdeau est présentée au tableau 6-7.

La mesure $V_{SS123456}$ de cette série doit être corrigée, car la surface éclairée d'une facette, lorsque toutes les facettes sont illuminées, est plus grande que si une seule facette est illuminée. Cela s'explique par le fait que le diamètre des faisceaux lumineux, lors de l'illumination d'une seule facette, est plus petit que le diamètre même de la facette. Lorsqu'on éclaire plusieurs facettes, toute la surface de la facette est inévitablement éclairée.

À l'aide de cette série de résultats, nous avons résolu le système d'équations du modèle en utilisant les équations développées précédemment. Malheureusement les résultats obtenus comprennent des valeurs d'impédance négatives, ce qui est irréaliste. De plus, certaines valeurs d'impédance sont très grandes (valeurs plus grandes qu'un million de $M\Omega$). Nous avons essayé de vérifier si des petites variations des mesures n'apporteraient pas une amélioration sensible. Les résultats n'ont pas varié de façon significative.

Afin de résoudre le système d'équations, nous avons utilisé la même technique que celle utilisée pour l'obtention des paramètres du

Tableau 6-7 : Valeurs expérimentales obtenues avec le montage.

Paramètres	Valeurs
$V_{SS123456}$ (Début)	16.5 mV
V_{SS1}	0.4 mV
V_{SS2}	0.8 mV
V_{SS3}	1.5 mV
V_{SS4}	11.5 mV
V_{SS5}	0.9 mV
V_{SS6}	0.5 mV
V_{SS14}	12.8 mV
V_{SS25}	1.8 mV
V_{SS36}	1.8 mV
V_{SS135}	6.6 mV
V_{SS246}	14.5 mV
$V_{SS123456}$ (Fin)	17.5 mV
Z_{SS0}	25.0 $M\Omega$
$Z_{SS123456}$	10.0 $M\Omega$
Z_{SS4}	12.5 $M\Omega$
V_{C1}	2.8 mV
V_{C14}	4.4 mV
V_{C135}	5.6 mV
$V_{C123456}$	12.0 mV

modèle à partir des résultats de van Hateren (voir section 6.5, page 174). Cependant, nous avons utilisé les sept équations indépendantes trouvées à la section 6.4.2 de la page 173. Nous avons obtenu certains paramètres avec des valeurs inférieures à $1 \text{ M}\Omega$ et d'autres paramètres avec des valeurs supérieures à $1 \text{ G}\Omega$. Nous n'avons pas obtenu de valeurs négatives, car nous avons limité la région de recherche aux valeurs positives.

Il nous est apparu intéressant d'effectuer une comparaison des mesures de van Hateren (1986) et les nôtres. Puisque toutes les mesures de tension et d'impédance ne sont pas données dans l'article de van Hateren (1986), nous avons estimé certaines mesures à partir des paramètres

Tableau 6-8 : Étude comparative entre une série de résultats obtenus avec notre montage et une série tirée des résultats expérimentaux de van Hateren (1986).

Paramètres	Données (montage)	Données (van Hateren)	Facteur
$V_{SS123456}$	16.5 mV	3.49 mV	4.72
V_{SS1}	0.4 mV	0.33 mV	1.21
V_{SS2}	0.8 mV	0.37 mV	2.16
V_{SS3}	1.5 mV	0.61 mV	2.46
V_{SS4}	11.5 mV	2.81 mV	4.09
V_{SS5}	0.9 mV	0.61 mV	1.48
V_{SS6}	0.5 mV	0.37 mV	1.35
V_{SS14}	12.8 mV	2.95 mV	4.34
V_{SS25}	1.8 mV	0.92 mV	1.96
V_{SS36}	1.8 mV	0.92 mV	1.96
V_{SS135}	6.6 mV	1.35 mV	4.89
V_{SS246}	14.5 mV	3.09 mV	4.69
Z_{SS0}	25.0 $\text{M}\Omega$	27.9 $\text{M}\Omega$	0.90
$Z_{SS123456}$	10.0 $\text{M}\Omega$	11.6 $\text{M}\Omega$	0.86
Z_{SS4}	10.5 $\text{M}\Omega$	12.0 $\text{M}\Omega$	0.88
V_{CS1}	2.8 mV	0.49 mV	5.71
V_{CS14}	4.4 mV	0.91 mV	4.84
V_{CS135}	5.6 mV	1.27 mV	4.41
$V_{CS123456}$	12.0 mV	1.98 mV	6.06

trouvés dans la section précédente. Le tableau 6-8 de la page précédente contient les deux séries de mesures de même qu'un facteur multiplicatif donnant le rapport entre les valeurs des deux séries.

Toute comparaison devant être faite avec attention, il faut mentionner des différences importantes entre les deux séries de résultats. Les résultats de van Hateren (1986) ont été obtenus avec la mouche caliphora alors que les nôtres ont été obtenus avec la mouche domestique. van Hateren a utilisé des échelons de lumière de 1 ms lors de ses stimulations et il a mesuré la valeur maximale de la réponse obtenue. Dans notre cas, nous avons utilisé des échelons lumineux de 200 ms et nous avons mesuré la valeur moyenne des réponses comprises entre les temps de 50 ms et 150 ms après le début du stimulus. L'utilisation de long stimulus est nécessaire pour faire une étude en courant continu qui nécessite une région non transitoire de la réponse électrique de la cellule. Finalement, notons que nos intensités lumineuses utilisées sont plus fortes que celles de van Hateren.

Tout comme pour les résultats de van Hateren, nous observons un comportement liant les amplitudes des réponses des photorécepteurs selon leur emplacement dans la pseudopupille cornéenne réduite. Pour la stimulation d'un seul photorécepteur, nous avons la réponse maximale pour le photorécepteur sous enregistrement et la réponse minimale pour le voisin le plus éloigné. Avec les premiers voisins, nous avons une réponse intermédiaire. Ce comportement est aussi conforme à celui de van Hateren pour ce qui est des stimulations de plusieurs photorécepteurs simultanément. Pour les photorécepteurs qui ne participent pas à la superposition neuronale, aucune réponse n'est perçue. Un comportement similaire entre les deux séries de résultats montre que notre préparation a beaucoup de points communs avec celle de van Hateren. Cela indique que les différences entre les séries de résultats expérimentaux pourraient davantage venir des conditions expérimentales comme le type de mouche et les caractéristiques des stimuli lumineux utilisés.

Nous notons que les mesures d'impédance sont sensiblement les mêmes pour les deux séries de données. Cela s'explique par le fait que les impédances ne sont pas très affectées par

l'intensité lumineuse. Nous remarquons aussi que les rapports pour les tensions de la cartouche sont sensiblement les mêmes pour toutes les tensions.

Pour ce qui est des tensions du photorécepteur, le rapport varie beaucoup d'une mesure à l'autre. Nous pouvons remarquer que plus les tensions en jeu sont élevées, plus les rapports sont élevés. Il y a donc une différence significative entre les deux séries de données. Cependant, notre modèle étant passif, les rapports entre les deux séries de résultats devraient être sensiblement les mêmes pour les mesures ayant le même nombre de photorécepteurs stimulés. Ainsi, nous aurions un même facteur pour les mesures V_{SS1} à V_{SS6} , un autre facteur pour les mesures V_{SS14} à V_{SS36} et un autre facteur pour les mesures V_{SS134} et V_{SS246} .

Cette différence pourrait s'expliquer par le fait que les circuits neuronaux de caliphora et de domestica sont distincts.

Quand nous comparons les valeurs des deux voisins immédiats, nous pouvons remarquer que dans le cas de van Hateren, ces deux voisins immédiats ont exactement la même amplitude (0.53 mV), ce qui n'est pas le cas pour nos résultats (0.9 mV et 1.5 mV). Cette asymétrie a aussi été observée par Shaw (1984). Toutefois, les résultats de Shaw ont été obtenus avec lucilia en utilisant un stimulus de 100 ms et une intensité lumineuse plus grande que celle que nous avons utilisée. De plus, Shaw stimulait toute l'ommatidie alors que van Hateren et nous stimulations un photorécepteur à la fois.

Deux facteurs peuvent contribuer à expliquer les différences observées :

- 1) Chez domestica, les « bumps (petites réponses spontanées)» ne sont pas perçus entre les photorécepteurs alors qu'ils le sont chez caliphora, ce qui indique des différences significatives d'opération des synapses électriques.
- 2) Les synapses électriques peuvent être non linéaires et directionnelles. Il s'ensuit que des stimuli lumineux faibles ou forts pourraient donner des différences significatives de fonctionnement des photorécepteurs.

Avant de rejeter ou de conserver le modèle, il faudra faire plusieurs autres séries d'expériences.

En dernière analyse, une étude rapide a été faite pour vérifier la sensibilité aux erreurs du modèle. Comme nous allons le voir, celui-ci est effectivement très sensible aux erreurs. L'étude a consisté à varier successivement de 1% chacune des mesures et à évaluer les écarts des sept paramètres par rapport aux valeurs originelles (voir tableau 6-9). Une erreur de 1 % pour certaines mesures peut entraîner des erreurs de 1000 % sur certains paramètres et même donner des valeurs négatives de paramètres. Il est donc crucial de prendre la moyenne de plusieurs séries d'expériences si nous voulons utiliser le modèle pour trouver la valeur des paramètres. Il serait aussi possible d'exprimer autrement les équations pour les rendre moins sensibles aux erreurs, mais cela risque d'être une tâche très complexe. Nous pouvons aussi

Tableau 6-9 : Étude de sensibilité du modèle aux erreurs de mesures expérimentales.

Mesures	Paramètres		R _A	R _B	R _L	R _N	R _S	R _T	δE
	Théorique	(+1%)	10	11	17	90	100	40	4
V _{P11}	2.81 V	2.84 V	14.7	21.3	15.8	64.9	64.5	79.4	3.7
			47%	94%	7%	28%	36%	99%	8%
V _{P21}	2.95 V	2.98 V	72.5	6.4	1.89	237	1060	-41	4.4
			625%	42%	89%	163%	960%	202%	10%
V _{P22}	0.92 V	0.93 V	14.2	12.1	16.8	85.3	67.5	41.1	3.9
			42%	10%	1%	5%	33%	3%	3%
V _{P32}	1.35 V	1.36 V	6.1	10.8	17.1	91.3	134.7	42.8	4.0
			39%	2%	1%	1%	35%	7%	0%
V _{P61}	3.49 V	3.53 V	12.5	12.9	16.9	74.1	70.3	45.3	4.0
			25%	17%	1%	18%	30%	13%	0%
V _{B1}	0.49 V	0.50 V	10.2	12.0	16.8	85.6	98.8	41.6	4.0
			2%	9%	1%	5%	1%	4%	0%
Z _{P61}	11.6 MΩ	11.72 MΩ	10.2	11.3	17.1	89.3	98.7	40.9	4.0
			2%	3%	1%	1%	1%	2%	0%

optimiser la technique itérative utilisée avec les résultats de van Hateren afin d'augmenter la vitesse du traitement des données. Il est à noter que l'analyse de nos données a nécessité un mois de calcul avec un Pentium II 266MHz.

6.7 Conclusion

Le système d'équations développé dans le présent chapitre permet d'obtenir les différents paramètres du modèle électrique en courant continu d'une ommatidie. Malheureusement, il n'est pas possible de trouver une série d'équations permettant de faire l'ensemble des mesures dans une seule cellule. Il faut donc effectuer des mesures dans le segment interne ou l'axone d'un photorécepteur puis obligatoirement une mesure dans la cartouche (milieu extracellulaire de lamina). Comme il est impératif d'avoir des enregistrements dans des cellules distinctes, il faudra effectuer une série de plusieurs expériences avec différentes mouches.

De plus, des expériences à différentes intensités lumineuses permettraient de mieux saisir le fonctionnement du circuit. Nous pourrions ainsi voir quels sont les paramètres qui varient vraiment en fonction de l'intensité lumineuse. Si la seule impédance variant à la lumière est R_L , c'est que le modèle précédent représente bien le fonctionnement du système visuel. Dans le cas contraire, cela voudrait dire qu'il y a variation d'au moins une autre impédance dans le modèle. Il serait alors intéressant de modifier le modèle pour mieux représenter le fonctionnement du circuit. Il faut aussi garder à l'esprit que le modèle est simplifié et qu'il néglige plusieurs connexions synaptiques vers d'autres cellules. Ces connexions peuvent venir modifier légèrement ou fortement le comportement du circuit. Il serait alors intéressant d'effectuer une série d'expériences dans lesquelles nous pourrions isoler les photorécepteurs soit par dissection ou en inhibant les synapses chimiques. En comparant les différentes séries de résultats il serait alors possible de voir les influences des autres cellules nerveuses.

Mentionnons que le modèle n'est valable que pour une étude en courant continu. Il est donc important que les mesures soient prises dans les régions non transitoires de la réponse des cellules. Si le modèle fonctionne correctement en courant continu, il sera alors possible de le modifier pour tenir compte de l'aspect temporel.

CHAPITRE VII

CONCLUSION GÉNÉRALE

Nous avons réalisé un stimulateur optique permettant d'étudier les différentes parties de la chaîne de traitement de l'information visuelle d'insectes possédant des yeux à facettes. Nous pouvons penser à l'étude du ERP (Early Receptor Potential) qui est un phénomène lié aux pigments visuels. À un niveau supérieur il est possible de faire des études sur les réponses des photorécepteurs à différents stimuli ou de leurs interconnexions comme au chapitre VI. Plus loin dans la chaîne de traitement de l'information visuelle, l'observation des LMC (Large Monopolar Cell) ou des cellules H1 permet d'étudier des phénomènes plus complexes comme la détection des arrêtes ou du mouvement.

La décomposition du montage en divers modules permet d'avoir un système relativement facile à modifier et qui ouvre la porte à des ajouts futurs au montage actuel. Nous retrouvons cette structure modulaire autant au niveau de l'optique, de la mécanique, de l'électronique et de l'informatique. Au niveau optique, cela permet d'interchanger facilement les types de sources lumineuses utilisées. Du côté de la mécanique, il est alors possible d'interchanger les mécanismes de déplacement des fibres optiques ou de la base. Pour l'électronique cela permet d'interchanger les cartes d'acquisition ou les contrôleurs. Finalement, pour l'informatique, l'ajout de nouveaux algorithmes de traitements ou de contrôles des expériences est possible. Il

est évident que l'ajout de nouvelles options au montage nécessite des modifications à différents niveaux, mais cela demeure simple étant donné l'aspect modulaire.

L'aspect modulaire nous a permis de réaliser trois versions du montage. Toutes ces versions permettent de réaliser différents types d'expériences et sont très différentes au niveau optique et mécanique. Heureusement, un expérimentateur connaissant l'un des montages peut facilement réaliser des expériences sur les deux autres.

Lors de la mise au point du montage, nous avons pris soin de faciliter la tâche d'un futur expérimentateur en effectuant diverses expériences. Suite à ces expériences, nous avons mieux compris les besoins des expérimentateurs et avons instauré des algorithmes diminuant le temps nécessaire à la réalisation d'une expérience.

Le système de gestion des protocoles expérimentaux permet d'atteindre un taux d'automatisation complet d'une expérience. Ce système donne accès à toutes les fonctionnalités automatiques du montage. Il est donc possible de créer à l'avance plusieurs protocoles et de les exécuter lors d'une expérience. L'automatisation d'une expérience permet d'en augmenter la vitesse, d'avoir des événements temporels très précis et d'éliminer les erreurs de manipulations. Finalement, avec un tel système de protocole, il est possible d'effectuer des expériences beaucoup plus complexes.

Malgré le bon fonctionnement du stimulateur optique, il reste encore beaucoup de travail à effectuer pour améliorer la puissance du montage. Voici quelques exemples.

- Il faudrait entre autres améliorer la fiabilité à long terme de certains des composants mécaniques du stimulateur optique. À titre d'exemple, nous pouvons penser à modifier la mécanique des obturateurs qui n'est pas assez robuste.
- Certains algorithmes fonctionnent relativement bien, mais ne tiennent pas compte de certains imprévus. Mentionnons l'algorithme de centrage d'une cellule qui se comporte de façon chaotique lors de la perte de la réponse électrique de la cellule sous enregistrement.

- Il y aurait intérêt à améliorer les algorithmes de détection des réponses électriques pour travailler avec des réponses provenant de cellules autres que les photorécepteurs et les LMC. De plus, certains algorithmes sont très sensibles au bruit et peu fiables quand les réponses électriques des cellules sont trop faibles.
- Il serait très utile d'optimiser certaines fonctionnalités du stimulateur lumineux. Par exemple, l'algorithme de mesure de la sensibilité spectrale peut prendre jusqu'à une demi-heure pour mesurer la sensibilité spectrale d'une cellule. Avec quelques modifications mineures au niveau matériel et logiciel, il serait possible d'avoir un algorithme qui peut faire le travail en deux minutes.
- Il serait aussi intéressant d'implanter des algorithmes de vision de haut niveau pour identifier la pseudopupille cornéenne réduite et numérotter automatiquement les facettes. Il serait aussi possible d'implanter un algorithme de recherche d'une cellule qui avancerait automatiquement la microélectrode pour trouver une cellule.
- Afin d'avoir un stimulateur optique des plus fiables, il faudrait aussi développer des algorithmes d'auto-vérification du stimulateur optique pour s'assurer que le montage est toujours calibré correctement. Il arrive que certaines pièces du montage se déplacent de quelques microns, ce qui rend invalides les algorithmes de déplacement des faisceaux lumineux.
- Pour mieux contrôler les expériences, il faudrait augmenter la sensibilité de certains senseurs du stimulateur optique. Il y a entre autres la photodiode qui permet de connaître l'intensité lumineuse de chacun des trajets lumineux qui n'est pas assez sensible pour les faibles intensités lumineuses.

Un montage expérimental n'est jamais réellement fini. Une multitude de particularités peuvent être améliorées. Il est intéressant de constater qu'aucune de ces améliorations ne nécessite des modifications majeures dans la structure du présent stimulateur optique. Il serait donc possible de traiter chacun des problèmes indépendamment. De plus, il y a aussi

beaucoup d'autres améliorations qui paraîtront utiles lors de la réalisation de futures expériences. Cependant, il faut toujours voir si une amélioration est vraiment utile avant de l'entreprendre pour éviter que le stimulateur possède une multitude de fonctionnalités qui ne servent pas ou qui serviront une seule fois.

En ce qui a trait à l'utilisation et l'amélioration du stimulateur optique, seule l'imagination de l'expérimentateur en limite les possibilités. Après avoir parcouru plusieurs articles décrivant des expériences sur la mouche, on réalise que certaines expériences complexes sont maintenant réalisables facilement avec ce stimulateur.

Au cours du présent doctorat, nous avons étudié un modèle électrique des interconnexions de certaines cellules d'une ommatidie. Cette étude n'a cependant pas pu être totalement complétée à cause de l'énorme travail nécessaire à la réalisation du montage. Par contre, les résultats donnent un bon point de départ pour des études subséquentes qui nécessitent un stimulateur optique comme celui présenté dans cette thèse.

La simple résolution théorique des équations du modèle de photorécepteurs est très intéressante, car elle permettra, à l'aide des résultats fournis par le stimulateur optique, de donner directement les paramètres du modèle. Cela permettra dans un premier temps de valider le modèle et dans un second temps de vérifier si seulement la variation de l'impédance du segment interne des cellules change lorsque l'intensité lumineuse varie.

Au fur et à mesure qu'un modèle valable se développera, il sera possible de faire plusieurs simulations théoriques pour découvrir le rôle de la structure des photorécepteurs R1 à R6 dans le traitement de l'information visuelle chez la mouche.

RÉFÉRENCES

Comtois S. and Gagné S. (1994) Presynaptic enhancement of signal transients in the fly photoreceptor terminals is facilitated by high-pass filtering. Proceedings of the 22nd Goettingen Neurobiology Conference, vol I, 35.

Finkel A. S. & Redman S. J. (1985) Optimal voltage clamping with a single microelectrode. In Voltage Clamping with Microelectrodes, ed. T. G. Smith et al, Williams & Wilkins, Baltimore.

Franceschini N. et Kirschfeld K. (1971) Les phénomènes de pseudopupille dans l'œil composé de *Drosophila*. Kybernetik 9, 159-182.

Gagné S., Ganguly U. and Comtois S. (1997a) Differential capacity compensation. Soumis pour publication au journal Medical & Biological Engineering & Computing.

Gagné S. and Comtois S. (1997) Oscillations in fly R1-R6 photoreceptors. Proceedings of the 25th Goettingen Neurobiology Conference, vol II, 123.

Gagné S. and Comtois S. (1996) What does the fly photoreceptors R1-6 electrical circuit tell us about information processing in the retina? Proceedings of the 24th Goettingen Neurobiology Conference, vol II, 323.

Gagné S. and Comtois S. (1995) Intriguing conditions found in the circuitry of the fly photoreceptors (R1-R6) related to polarized light. Proceedings of the 23rd Goettingen Neurobiology Conference, vol II, 397.

Gagné S. and Comtois S. (1995) Do large monopolar cells of the fly encode polarized light information coming from photoreceptors (R1-R6)? 25th Annual Meeting of the Society for neurosciences, San Diego, CA, Nov 11-16.

Gagné S. and Comtois S. (1994) The 3D electrical circuit of fly photoreceptors (R1-R6) and its parameter value measurements. 24th Annual Meeting of the Society for neurosciences, Abstract 326.11, part 1, p. 779, Miami, FL Nov 13-18.

Gagné S. and Comtois S. (1994) Electrical parameter evaluation of fly photoreceptors (R1-R6). Proceedings of the 22nd Goettingen Neurobiology Conference, vol I, 34.

Gagné S. and Comtois S. (1993) Signal attenuation is normally negligible in fly photoreceptors. 23rd Annual Meeting of the Society for neurosciences, Washington, DC.

Hardie R. C. (1985) Functional Organisation of the Fly Retina. Progress in Sensory Physiology 5, Springer-Verlag, Berlin Heidelberg New York, 1-79.

Howard J., Blakeslee B. and Laughlin S.B. (1987) The intracellular pupil mechanism and photoreceptor signal:noise ratios in the fly *Lucilia cuprina*. Proc. R. Soc. Lond B, 231, 415-435.

Jack J.J.B., Noble D. and Tsien R.W. (1983) Electric current flow in excitable cells. Oxford University Press

Mosckowich T. (1994) Logiciel de commande et de traitement de données pour un montage de vision biologique. Mémoire, Université Laval, Québec, Canada.

Rossel S. (1989) Polarization sensitivity in compound eyes Facets of Vision (D.G Stavenga, R. C. Hardie (Eds.)), Springer-Verlag, Berlin Heidelberg New York, 300-316.

Shaw S. R. (1984) Early visual processing in insects Journal of Experimental Biology, 112, 225-251.

Strausfeld N. J. (1976) Atlas of an insect brain. Springer-Verlag, Berlin Heidelberg New York.

Stausfeld N. J. (1989) Beneath the Compound Eye : Neuroanatomical Analysis and Physiological Correlates in the study of Insect Vision. Facets of Vision (D.G Stavenga, R.C. Hardie (Eds.)), Springer-Verlag, Berlin Heidelberg New York, 317-359.

Uusitalo R. O., Juusola M. and Weckström M. (1995) Graded Responses and Spiking Properties of Identified First-Order Visual Interneurons of the Fly Compound Eye. Journal of Neurophysiology Vol. 73, No.5, 1782-1792.

van Hateren J. H. (1984) Waveguide theory applied to optically measured angular sensitivities of fly photoreceptors. J. Comp. Physiol. A., 154, 761-771.

van Hateren J. H. (1986) Electrical coupling of neuro-ommatidial photoreceptor cells in the blowfly. J. Comp. Physiol. A., 158, 795-811.

van Hateren J. H. (1987) Photoreceptor optics and neural microcircuitry in the insect eye. Thesis, Univ of Groningen.

Weckström M., Juusola M. and Laughlin S.B. (1992) Presynaptic enhancement of signal transients in photoreceptor terminals in the compound eye. Proc. R. Soc. Lond B, 250, 83-89.

ANNEXE A

LA COMMUNICATION DDE SOUS WINDOWS NT

Puisque la communication DDE (Dynamique Data Exchange) est utilisée de façon intensive dans les diverses applications du montage, la présente annexe fait un bref survol de son fonctionnement. Pour de plus amples informations sur la communication DDE, vous pouvez vous référer à l'aide que l'on retrouve dans plusieurs produits de Microsoft (Visual C++, Software Development Kit (SDK), Microsoft Development Network (MSDN)).

Pour permettre la communication entre deux applications, il faut préalablement définir le rôle de chacune des applications. Une application est de type serveur si elle donne accès à certaines informations ou commandes à une autre application. De l'autre côté de la communication, l'application est de type client et c'est elle qui fait la demande d'information auprès du serveur ou exécute une commande sur le serveur.

Lorsqu'une transaction est établie entre deux applications, il y a trois niveaux de référence. Il y a premièrement le service qui représente la communication proprement dite. Habituellement, il y a un service par application, mais rien n'interdit au serveur ou au client d'en utiliser plusieurs. Par la suite, il y a le topique qui permet de subdiviser un service en différents groupes. Enfin, il y a l'item qui indique la donnée à partager ou la commande à exécuter. À titre d'exemple, on peut penser à l'application *Serveur de port RS-232* qui est

subdivisée comme suit. Premièrement, il y a le service qui s'appelle *Serveur de port RS-232*. L'application possède quatre topiques où chaque topique donne accès à un port série. Ils sont appelés COM1, COM2, COM3 et COM4. Finalement, il y a deux items qui peuvent être associés à chacun des topiques, c'est-à-dire un item pour la transmission de données vers le serveur et un item pour la réception de données du serveur, qui s'appellent respectivement *DonnesClientServeur* et *DonnesServeurClient*.

Lorsque le serveur débute, il doit enregistrer les services, les topiques et les items auprès du système d'exploitation et activer les services qu'il offre. Par la suite, il attend qu'un client demande une connexion. De son côté, le client doit aussi enregistrer les services, les topiques et les items. Cependant, il n'a pas besoin d'activer les services.

La communication commence quand le client demande une connexion auprès du système d'exploitation en spécifiant le service et le topique sous lesquels il désire opérer. La demande de connexion est alors acheminée au serveur, et ce dernier peut accepter ou refuser la connexion. Après que la connexion soit établie, le transfert de données entre les deux applications peut commencer.

Il est aussi possible, pour un client, de connaître tous les topiques et les items qu'un serveur possède. Cependant, il faut avoir un consensus sur les formats des données et des commandes.

Le transfert de données entre deux applications peut se faire de trois façons différentes. La première façon est la plus commune et permet au client d'envoyer un bloc de données au serveur avec la commande *Poke* ou de demander un bloc de données avec la commande *Request*. Le bloc de données à transférer dépend du topique et de l'item que le client utilise. On nomme ce type de communication une communication froide. Le deuxième type de communication permet au client de demander au serveur de l'aviser quand les données associées à un topique et à un item changeront. Lorsque c'est le cas, le serveur envoie le message *AdviceRequest* au client pour l'aviser que l'item a été modifié. Le cas échéant, le client peut utiliser la commande *Request* de la communication froide pour aller chercher les nouvelles données. La présente communication est dite communication tiède. Le dernier type

de communication est une dérivée de la communication tiède. La seule différence, c'est que le serveur envoie le nouveau bloc de données au client à l'aide de la commande *AdviceData* au lieu de seulement l'aviser du changement des données. On appelle ce type de communication une communication chaude. Le client peut à tout moment arrêter la communication chaude ou la communication tiède.

A titre d'exemple, l'application *Contrôle des expériences* utilise les trois types de communication. La communication froide permet d'envoyer des commandes de déplacement de moteurs à l'application *Contrôleur de moteurs (x86)*. La communication tiède permet à l'application *Contrôle des expériences* d'être avisée de la fin de l'acquisition d'une nouvelle courbe par l'application *Acquisition et traitement des courbes*. Par la suite, elle peut utiliser la commande *Request* pour récupérer la courbe ou simplement connaître l'amplitude maximale de la courbe, l'amplitude minimale de la courbe, etc. La communication chaude permet à l'application *Contrôle des expériences* de recevoir la nouvelle position des moteurs de l'application *Contrôleur de moteurs (x86)*.

La communication prend fin quand l'une des deux applications demande l'arrêt de la communication. La cas échéant, la seconde application ne peut pas refuser la demande.

ANNEXE B

PROTOCOLE EXPÉRIMENTAL UTILISÉ POUR LA RÉOLUTION DU MODÈLE DU CHAPITRE VI

Nous présentons le protocole expérimental qui est utilisé pour trouver les tensions nécessaires à la résolution du modèle du chapitre VI. Ces explications donneront du même coup donner une idée de la puissance du protocole expérimental de l'application *Contrôle des expériences*.

Nous expliquerons chacun des items du protocole expérimental. Pour éviter de répéter inutilement la description de ces items, nous les avons fait suivre d'un indice tout en convenant que ceux possédant un indice identique font le même genre de travail. Cependant, les items en italique correspondent à des commentaires qui servent à augmenter la lisibilité du protocole.

Configuration générale

Ferme le séquenceur *Synchronisation*¹

Cet item permet de fermer un séquenceur associé à un trajet lumineux. Dans la présente configuration, il y a un séquenceur virtuel et cinq séquenceurs physiques. Le séquenceur virtuel *Synchronisation* commande les autres séquenceurs et n'est associé à aucun mécanisme du stimulateur. Les séquenceurs *Trajet n° 1*, *Trajet n° 2*, *Trajet n° 3*

et *Trajet de fond* permettent de contrôler respectivement les lumières des trajets 1, 2, 3 et du trajet de fond. Le séquenceur *Impédance* permet de contrôler l'injection de courant.

Ferme le séquenceur *Trajet n° 1*¹

Ferme le séquenceur *Trajet n° 2*¹

Ferme le séquenceur *Trajet n° 3*¹

Ferme le séquenceur *Trajet de fond*¹

Ferme le séquenceur *Impédance*¹

Configuration *ParamètresMoy.cfg* pour tous les séquenceurs²

Cette commande sert à configurer les séquenceurs à partir du fichier de configuration *ParamètresMoy.cfg*. Ce fichier programme les cinq derniers séquenceurs pour qu'ils soient synchronisés avec le premier séquenceur. Les séquenceurs associés aux trajets lumineux auront une durée de 200 ms et démarreront 50 ms après la montée du séquenceur *Synchronisation*. Le séquenceur *Impédance* aura une durée de 240 ms et démarrera 30ms après le séquenceur *Synchronisation*. La période du séquenceur *Synchronisation* est fixée à cinq secondes et dicte l'intervalle entre les acquisitions. La gâchette de la carte d'acquisition est fixée à 10 ms après la montée du séquenceur de synchronisation. Donc, pour commencer une stimulation, il ne suffira que de démarrer le séquenceur *Synchronisation* après avoir activé les autres séquenceurs que l'on désire utiliser.

Nombre répétition = 20 pour le séquenceur *Synchronisation*³

Cette directive fixe le nombre de répétitions du séquenceur *Synchronisation* à 20. Cela veut dire que le séquenceur effectuera 20 périodes avant de s'arrêter. Dans le présent cas, chaque période correspondra à une acquisition.

Délai de 300000 ms⁴

Cette consigne insère un temps d'attente dans le protocole. Dans le présent cas, cela permet de s'assurer que la mouche est laissée au moins cinq minutes à la noirceur avant de poursuivre. Il faut bien sûr que les séquenceurs soient déjà fermés pour que la mouche soit à la noirceur.

Éclairement de R1 à R6 (début)

Dia. T1 = 112.00 μm sur M1 ⁵

Cet item déplace les trajets lumineux. Dans le présent cas, le trou de 112 μm du trajet n° 1 est déplacé sur le marqueur M1. Pour que le protocole fonctionne correctement, il faut que tous les marqueurs utilisés par le protocole soient placés avant de démarrer le protocole. Dans le présent protocole, les marqueurs F1 à F7 sont utilisés pour marquer les sept facettes de la pseudopupille cornéenne réduite, le marqueur M0 pour marquer le photorécepteur sous enregistrement, et le marqueur M1 pour marquer le centre de la pseudopupille cornéenne réduite.

Attend fin changement ⁶

Cette commande attend le protocole jusqu'au moment où tous les moteurs sont arrêtés avant de poursuivre. Cela permet donc, dans le présent cas, d'être certain que le faisceau lumineux du trajet n° 1 est sur le marqueur M1 avant de poursuivre.

Active le séquenceur *Trajet n° 1* ⁷

Cette directive active le séquenceur associé au premier trajet du stimulateur. Cela est nécessaire si on veut que ce dernier s'active lors du démarrage du séquenceur de synchronisation.

Acq. et moy. 20 courbes en utilisant le projet *Expérience* ⁸

Cette consigne programme la carte d'acquisition pour faire l'acquisition et la moyenne de 20 courbes. Par la suite, la courbe résultante est transférée dans le projet *Expérience* du logiciel *Acquisition et traitement des courbes*. L'acquisition des courbes sera synchronisée avec la gâchette du séquenceur *Synchronisation*.

Démarre le séquenceur *Synchronisation* ⁹

Cet item démarre le séquenceur et l'acquisition des courbes du même coup. Le séquenceur effectuera 20 itérations qui correspondront à l'acquisition de 20 courbes. Après les 20 itérations, le séquenceur arrêtera et attendra la prochaine commande.

Attend fin acq. du projet *Courbe* ¹⁰

Cette commande bloque le protocole expérimental jusqu'au moment où l'acquisition des courbes sera complétée.

Ferme le séquenceur *Trajet n° 1* ¹***Impédance noireur***

Active le séquenceur *Impédance*⁷
 Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸
 Démarre le séquenceur *Synchronisation*⁹
 Attend fin acq. du projet *Courbe*¹⁰
 Ferme le séquenceur *Impédance*¹

Impédance avec éclairage R1 à R6

Dia. T1 = 112.00 um sur M1⁵
 Attend fin changement⁶
 Active le séquenceur *Impédance*⁷
 Active le séquenceur *Trajet n° 1*⁷
 Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸
 Démarre le séquenceur *Synchronisation*⁹
 Attend fin acq. du projet *Courbe*¹⁰
 Ferme le séquenceur *Impédance*¹
 Ferme le séquenceur *Trajet n° 1*¹

Impédance avec éclairage M0

Dia. T1 = 12.00 um sur M0⁵
 Attend fin changement⁶
 Active le séquenceur *Impédance*⁷
 Active le séquenceur *Trajet n° 1*⁷
 Acq. et moy. 20 courbes en utilisant le projet '*Expérience*'⁸
 Démarre le séquenceur *Synchronisation*⁹
 Attend fin acq. du projet *Courbe*¹⁰
 Ferme le séquenceur *Impédance*¹
 Ferme le séquenceur *Trajet n° 1*¹
Éclairage de R1

Dia. T1 = 12.00 um sur F1⁵

Attend fin changement⁶

Active le séquenceur *Trajet n° 1*⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸

Démarre le séquenceur *Synchronisation*⁹

Attend fin acq. du projet *Courbe*¹⁰

Ferme le séquenceur *Trajet n° 1*¹

Éclairement de R2

Dia. T1 = 12.00 um sur F2⁵

Attend fin changement⁶

Active le séquenceur *Trajet n° 1*⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸

Démarre le séquenceur *Synchronisation*⁹

Attend fin acq. du projet *Courbe*¹⁰

Ferme le séquenceur *Trajet n° 1*¹

Éclairement de R3

Dia. T1 = 12.00 um sur F3⁵

Attend fin changement⁶

Active le séquenceur *Trajet n° 1*⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸

Démarre le séquenceur *Synchronisation*⁹

Attend fin acq. du projet *Courbe*¹⁰

Ferme le séquenceur *Trajet n° 1*¹

Éclairement de R4

Dia. T1 = 12.00 um sur F4⁵

Attend fin changement ⁶

Active le séquenceur *Trajet n° 1* ⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience* ⁸

Démarre le séquenceur *Synchronisation* ⁹

Attend fin acq. du projet *Courbe* ¹⁰

Ferme le séquenceur *Trajet n° 1* ¹

Éclairement de R5

Dia. T1 = 12.00 um sur F5 ⁵

Attend fin changement ⁶

Active le séquenceur *Trajet n° 1* ⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience* ⁸

Démarre le séquenceur *Synchronisation* ⁹

Attend fin acq. du projet *Courbe* ¹⁰

Ferme le séquenceur *Trajet n° 1* ¹

Éclairement de R6

Dia. T1 = 12.00 um sur F6 ⁵

Attend fin changement ⁶

Active le séquenceur *Trajet n° 1* ⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience* ⁸

Démarre le séquenceur *Synchronisation* ⁹

Attend fin acq. du projet *Courbe* ¹⁰

Ferme le séquenceur *Trajet n° 1* ¹

Éclairement de R7

Dia. T1 = 12.00 um sur F7 ⁵

Attend fin changement ⁶

Active le séquenceur *Trajet n° 1*⁷
Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸
Démarre le séquenceur *Synchronisation*⁹
Attend fin acq. du projet *Courbe*¹⁰
Ferme le séquenceur *Trajet n° 1*¹

Éclairement de R1 et R4

Dia. T1 = 12.00 um sur F1⁵
Dia. T2 = 12.00 um sur F4⁵
Attend fin changement⁶
Active le séquenceur *Trajet n° 1*⁷
Active le séquenceur *Trajet n° 2*⁷
Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸
Démarre le séquenceur *Synchronisation*⁹
Attend fin acq. du projet *Courbe*¹⁰
Ferme le séquenceur *Trajet n° 1*¹
Ferme le séquenceur *Trajet n° 2*¹

Éclairement de R2 et R5

Dia. T1 = 12.00 um sur F2⁵
Dia. T2 = 12.00 um sur F5⁵
Attend fin changement⁶
Active le séquenceur *Trajet n° 1*⁷
Active le séquenceur *Trajet n° 2*⁷
Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸
Démarre le séquenceur *Synchronisation*⁹
Attend fin acq. du projet *Courbe*¹⁰
Ferme le séquenceur *Trajet n° 1*¹
Ferme le séquenceur *Trajet n° 2*¹

Éclairement de R3 et R6**Dia. T1 = 12.00 um sur F1**⁵**Dia. T2 = 12.00 um sur F4**⁵**Attend fin changement**⁶**Active le séquenceur *Trajet n° 1***⁷**Active le séquenceur *Trajet n° 2***⁷**Acq. et moy. 20 courbes en utilisant le projet *Expérience***⁸**Démarre le séquenceur *Synchronisation***⁹**Attend fin acq. du projet *Courbe***¹⁰**Ferme le séquenceur *Trajet n° 1***¹**Ferme le séquenceur *Trajet n° 2***¹***Éclairement de R1, R3 et R5*****Dia. T1 = 12.00 um sur F1**⁵**Dia. T2 = 12.00 um sur F3**⁵**Dia. T3 = 12.00 um sur F5**⁵**Attend fin changement**⁶**Active le séquenceur *Trajet n° 1***⁷**Active le séquenceur *Trajet n° 2***⁷**Active le séquenceur *Trajet n° 3***⁷**Acq. et moy. 20 courbes en utilisant le projet *Expérience***⁸**Démarre le séquenceur *Synchronisation***⁹**Attend fin acq. du projet *Courbe***¹⁰**Ferme le séquenceur *Trajet n° 1***¹**Ferme le séquenceur *Trajet n° 2***¹**Ferme le séquenceur *Trajet n° 3***¹***Éclairement de R2, R4 et R6***

Dia. T1 = 12.00 um sur F2⁵

Dia. T2 = 12.00 um sur F4⁵

Dia. T3 = 12.00 um sur F6⁵

Attend fin changement⁶

Active le séquenceur *Trajet n° 1*⁷

Active le séquenceur *Trajet n° 2*⁷

Active le séquenceur *Trajet n° 3*⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸

Démarre le séquenceur *Synchronisation*⁹

Attend fin acq. du projet *Courbe*¹⁰

Ferme le séquenceur *Trajet n° 1*¹

Ferme le séquenceur *Trajet n° 2*¹

Ferme le séquenceur *Trajet n° 3*¹

Éclairement de R1 à R6 (fin)

Dia. T1 = 112.00 um sur M1⁵

Attend fin changement⁶

Active le séquenceur *Trajet n° 1*⁷

Acq. et moy. 20 courbes en utilisant le projet *Expérience*⁸

Démarre le séquenceur *Synchronisation*⁹

Attend fin acq. du projet *Courbe*¹⁰

Ferme le séquenceur *Trajet n° 1*¹

Fin du protocole

ANNEXE C

RÉSOLUTION, ÉQUATIONS ET GRAPHES DE FLUENCE DU MODÈLE

La présente annexe contient la procédure de résolution des équations, les graphes de fluence, les équations du modèle et les équations de six des sept paramètres. L'équation du paramètre R_T n'est pas présente, car il faut travailler directement avec les valeurs numériques pour réussir à l'obtenir.

C.1 Résolution des équations

Les lignes qui suivent présentent la séquence d'opérations qu'il faut effectuer pour résoudre le système de sept équations à sept inconnues. Afin d'alléger le texte, nous ne donnerons pas les équations complètes, mais seulement une forme abrégée pour représenter l'évolution des équations.

0) Initialement, nous avons les équations suivantes :

$$V_{SS1} = f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E)$$

$$V_{SS2} = f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E)$$

$$V_{SS3} = f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E)$$

$$V_{SS4} = f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E)$$

$$\begin{aligned}
V_{SS14} &= f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E) \\
V_{SS25} &= f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E) \\
V_{SS135} &= f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E) \\
V_{SS246} &= f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E) \\
V_{SS123456} &= f(R_A, R_B, R_L, R_T, \delta E) \\
Z_{SS0} &= f(R_A, R_B, R_N, R_S, R_T) \\
Z_{SS1} &= f(R_A, R_B, R_L, R_N, R_S, R_T) \\
Z_{SS123456} &= f(R_A, R_B, R_L, R_S, R_T) \\
V_{CS1} &= f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E) \\
V_{CS14} &= f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E) \\
V_{CS135} &= f(R_A, R_B, R_L, R_N, R_S, R_T, \delta E) \\
V_{CS123456} &= f(R_A, R_B, R_L, R_T, \delta E)
\end{aligned}$$

1) En isolant δE dans l'équation V_{SS25} et en le remplaçant dans les autres équations, nous obtenons les équations suivantes :

$$\begin{aligned}
\delta E &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{SS1} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{SS2} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{SS3} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{SS4} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{SS14} &= f(R_A, R_B, R_N, R_S, R_T, V_{SS25}) \\
V_{SS25} &= f() \\
V_{SS135} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{SS246} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{SS123456} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
Z_{SS0} &= f(R_A, R_B, R_N, R_S, R_T) \\
Z_{SS1} &= f(R_A, R_B, R_L, R_N, R_S, R_T) \\
Z_{SS123456} &= f(R_A, R_B, R_L, R_S, R_T) \\
V_{CS1} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{CS14} &= f(R_A, R_B, R_N, R_S, R_T, V_{SS25}) \\
V_{CS135} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25}) \\
V_{CS123456} &= f(R_A, R_B, R_L, R_N, R_S, R_T, V_{SS25})
\end{aligned}$$

Après le remplacement de δE dans les différentes équations, il est normal que l'équation utilisée pour extraire δE n'existe plus.

2) En isolant R_N dans l'équation V_{SS14} et en le remplaçant dans les autres équations, nous obtenons les équations suivantes :

$$\begin{aligned}
 R_N &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{SS1} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{SS2} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{SS3} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{SS4} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{SS14} &= f() \\
 V_{SS135} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{SS246} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{SS123456} &= f(R_A, R_B, R_L, R_S, R_T) \\
 Z_{SS0} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}) \\
 Z_{SS1} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 Z_{SS123456} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{CS1} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{CS14} &= f(R_A, R_B, R_T, V_{SS25}, V_{SS14}) \\
 V_{CS135} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14}) \\
 V_{CS123456} &= f(R_A, R_B, R_L, R_S, R_T, V_{SS25}, V_{SS14})
 \end{aligned}$$

3) En isolant R_L dans l'équation $V_{SS123456}$ et en le remplaçant dans les autres équations, nous obtenons les équations suivantes :

$$\begin{aligned}
 R_L &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
 V_{SS1} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
 V_{SS2} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
 V_{SS3} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
 V_{SS4} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
 V_{SS135} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
 V_{SS246} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
 V_{SS123456} &= f()
 \end{aligned}$$

$$\begin{aligned}
Z_{SS0} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}) \\
Z_{SS1} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
Z_{SS123456} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
V_{CS1} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
V_{CS14} &= f(R_A, R_B, R_T, V_{SS25}, V_{SS14}) \\
V_{CS135} &= f(R_A, R_B, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}) \\
V_{CS123456} &= f(R_A, R_B, R_T, V_{SS123456})
\end{aligned}$$

4) En isolant R_B dans l'équation V_{SS246} et en le remplaçant dans les autres équations, nous obtenons les équations suivantes :

$$\begin{aligned}
R_B &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS1} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS2} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS3} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS4} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS135} &= f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS246} &= f() \\
Z_{SS0} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
Z_{SS1} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
Z_{SS123456} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{CS1} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{CS14} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{CS135} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{CS123456} &= f(R_A, R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246})
\end{aligned}$$

À la suite de cette itération, nous voyons que l'équation V_{SS135} ne dépend plus des impédances. On peut dire de cette équation qu'elle est dépendante des quatre premières équations utilisées, donc elle ne peut pas être utilisée pour l'extraction des autres paramètres. Cette équation est donc rejetée pour les itérations suivantes.

5) En isolant R_A dans l'équation V_{SS1} et en le remplaçant dans les autres équations, nous obtenons les équations suivantes :

$$\begin{aligned}
R_A &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\
V_{SS1} &= f() \\
V_{SS2} &= f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS3} &= f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}) \\
V_{SS4} &= f(V_{SS25}, V_{SS14}, V_{SS123456}) \\
Z_{SS0} &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\
Z_{SS1} &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\
Z_{SS123456} &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\
V_{CS1} &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\
V_{CS14} &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\
V_{CS135} &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}) \\
V_{CS123456} &= f(R_S, R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1})
\end{aligned}$$

Le phénomène de dépendance est ici présent pour les équations V_{SS2} , V_{SS3} et V_{SS4} .

6) En isolant R_S dans l'équation V_{CS1} et en le remplaçant dans les autres équations, nous obtenons les équations suivantes :

$$\begin{aligned}
R_S &= f(R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{CS1}, Z_{SS0}) \\
Z_{SS0} &= f(R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}) \\
Z_{SS1} &= f(R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}) \\
Z_{SS123456} &= f(R_T, V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}) \\
V_{CS1} &= f() \\
V_{CS14} &= f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS1}, V_{CS1}) \\
V_{CS135} &= f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}) \\
V_{CS123456} &= f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS1}, V_{CS1})
\end{aligned}$$

Tout comme pour les cas précédents, il y a encore des équations indépendantes, soit V_{CS14} , V_{CS135} et $V_{CS123456}$.

7) En isolant R_T dans l'équation $Z_{SS123456}$ et en le remplaçant dans les autres équations, nous obtenons les équations suivantes :

$$R_T = f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}, Z_{SS123456})$$

$$Z_{SS0} = f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}, Z_{SS123456})$$

$$Z_{SS1} = f(V_{SS25}, V_{SS14}, V_{SS123456}, V_{SS246}, V_{SS1}, V_{CS1}, Z_{SS123456})$$

$$Z_{SS123456} = f()$$

Vu la complexité de l'équation $Z_{SS123456}$, il n'est pas possible d'extraire l'équation symbolique de R_T . Pour compléter le traitement, il faut donc remplacer les tensions par des valeurs numériques (voir tableau C-1), ce qui simplifie énormément les équations et permet de trouver la valeur de R_T .

Tableau C-1 : Données numériques utilisées pour compléter la résolution du modèle.

Paramètres	Mesures expérimentales
$R_A = 18 \text{ M}\Omega$	$V_{SS1} = 2,70 \text{ mV}$
$R_B = 17 \text{ M}\Omega$	$V_{SS14} = 2,81 \text{ mV}$
$R_L = 18 \text{ M}\Omega$	$V_{SS25} = 0,81 \text{ mV}$
$R_N = 91 \text{ M}\Omega$	$V_{SS246} = 1,19 \text{ mV}$
$R_S = 100 \text{ M}\Omega$	$V_{CS123456} = 1,95 \text{ mV}$
$R_T = 50 \text{ M}\Omega$	$V_{CS1} = 0,47 \text{ mV}$
$\delta E = 3,6 \text{ mV}$	$Z_{SS123456} = 13,20 \text{ M}\Omega$

Puisqu'il n'y avait pas de données

expérimentales disponibles au moment de la rédaction, les valeurs numériques utilisées ont été générées par les équations de départ et en utilisant les valeurs d'impédance et de source trouvées dans les articles de Gagné et Comtois (1993 à 1997).

C.2 Graphes de fluences

Voici le graphe de fluence pour la stimulation lumineuse avec ou sans injection de courant d'un photorécepteur (figure C-1), pour la stimulation lumineuse de deux photorécepteurs (figure C-2, page 209), pour la stimulation lumineuse de trois photorécepteurs (figure C-3, page 209), pour la stimulation lumineuse de six photorécepteurs et l'injection de courant dans un photorécepteur (figure C-4, page 210), et pour l'injection de courant dans un photorécepteur (figure C-5, page 211).

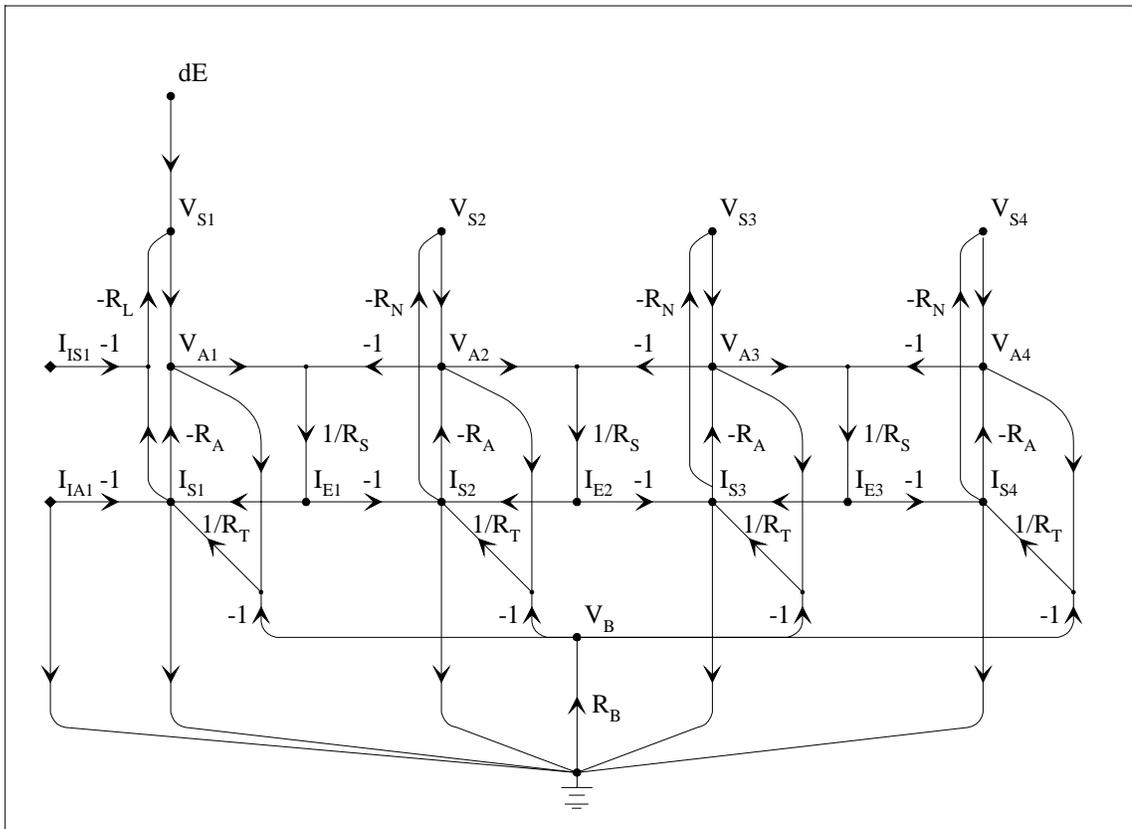


Figure C-1 : Graphe de fluence pour la stimulation lumineuse d'un photorécepteur avec ou sans injection de courant dans ce photorécepteur.

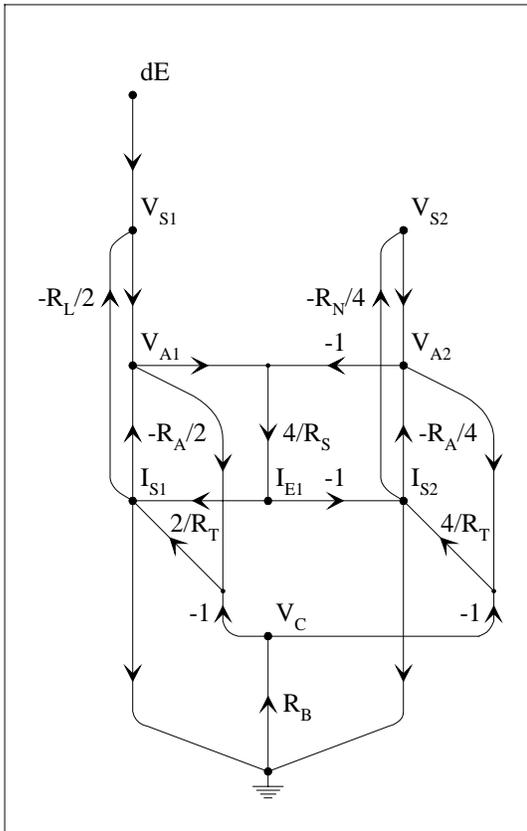


Figure C-2 : Graphe de fluence pour la stimulation lumineuse de deux photorécepteurs.

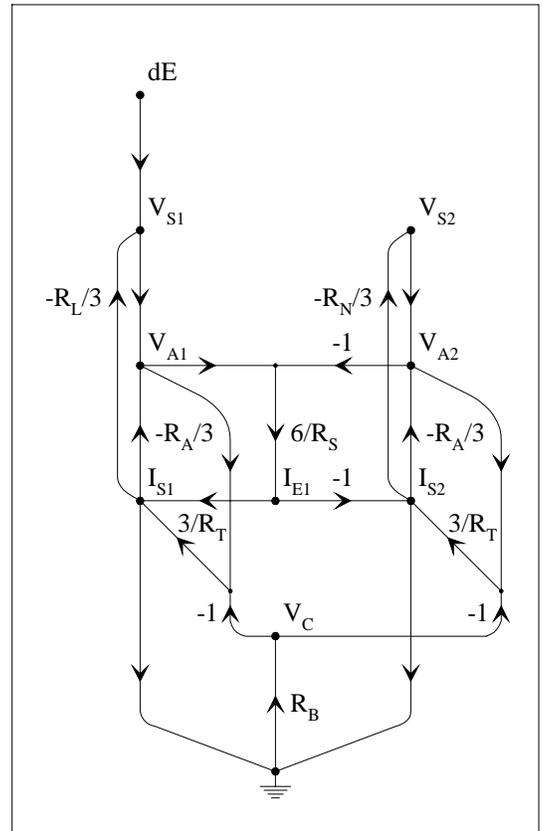


Figure C-3 : Graphe de fluence pour la stimulation lumineuse de trois photorécepteurs.

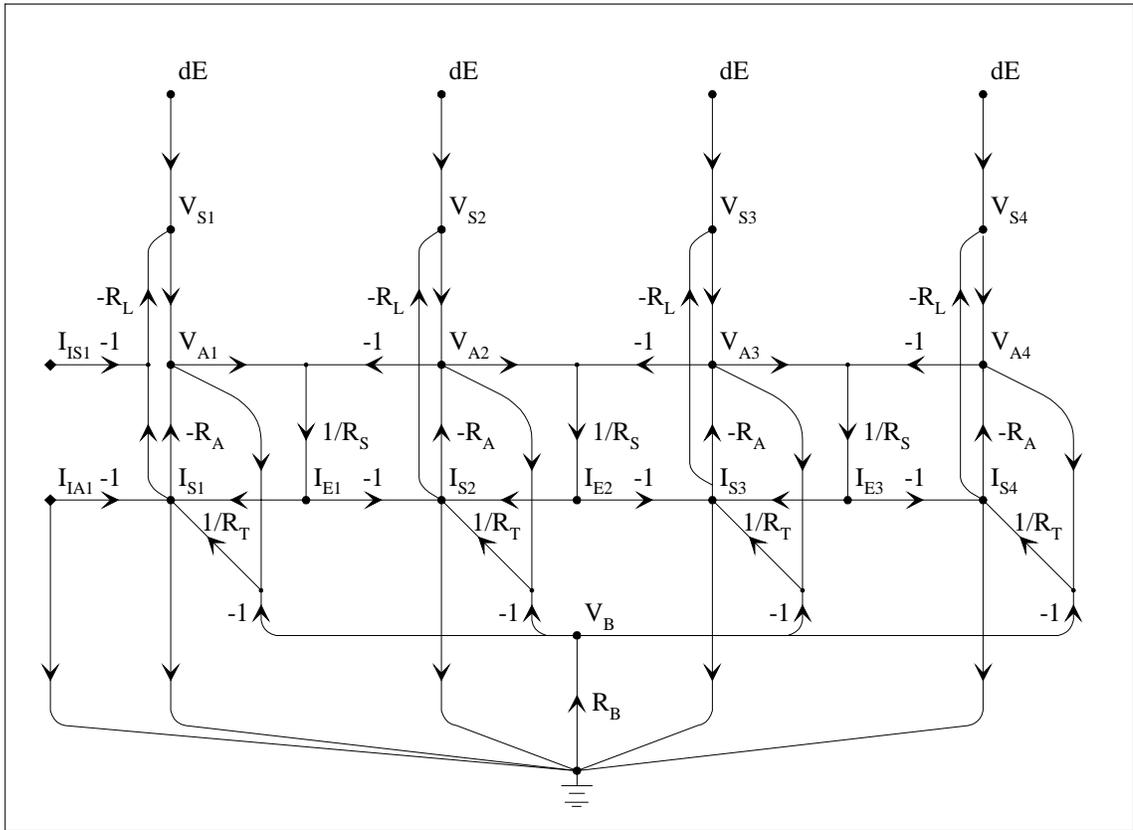


Figure C-4 : Graphe de fluence pour la stimulation lumineuse de six photorécepteurs avec l'injection de courant dans un photorécepteur.

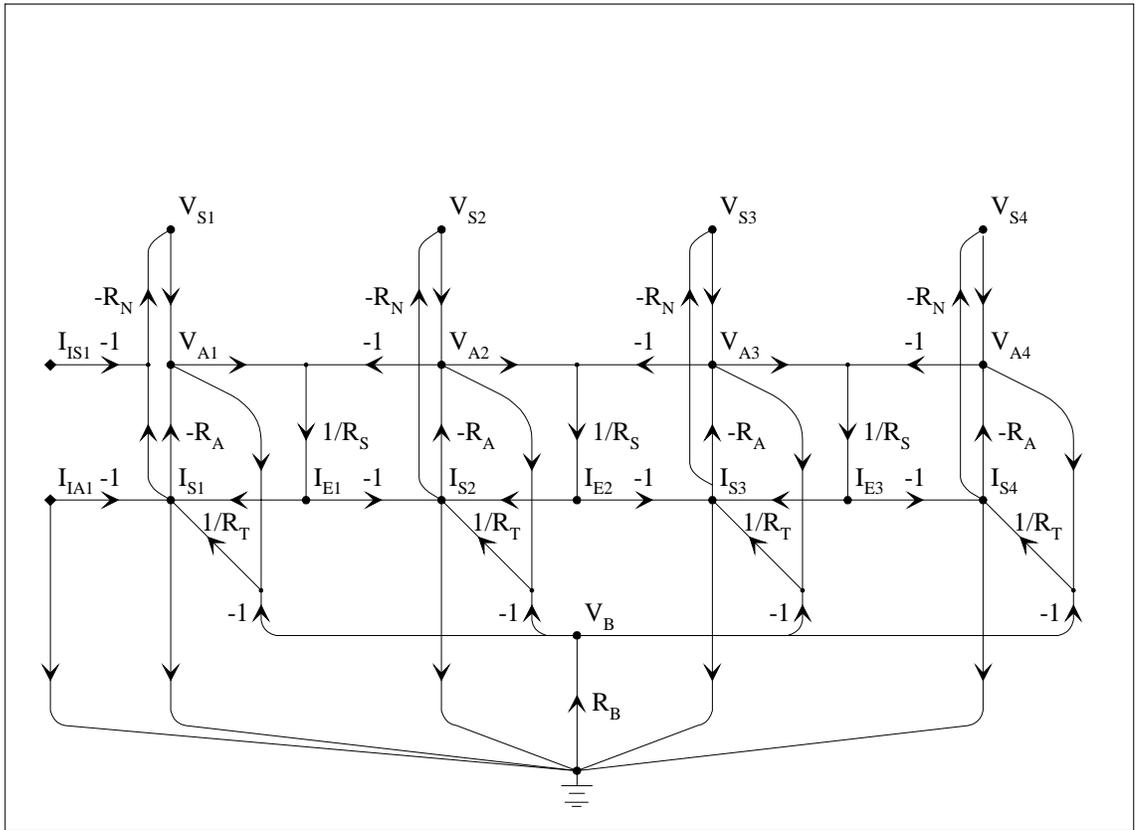


Figure C-5 : Graphe de fluence pour l'injection de courant dans un photorécepteur sans aucune stimulation lumineuse.

C.3 Équations du modèle

Cette section contient les équations des nœuds V_{SS1} , V_{SS2} , V_{SS3} , V_{SS4} , V_{SS14} , V_{SS25} , V_{SS135} , V_{SS246} , $V_{SS123456}$, Z_{SS0} , Z_{SS1} , $Z_{SS123456}$, V_{CS1} , V_{CS14} , V_{CS135} et $V_{CS123456}$.

$$\begin{aligned}
V_{SS1} = & \delta E (R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 R_B R_N R_S^3 + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 \\
& + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B R_S^2 R_T + 24 R_A^3 R_N R_S^2 R_T + 104 R_A^2 R_B \\
& R_N R_S^2 R_T + 24 R_A^2 R_N^2 R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T + 4 \\
& R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 R_A R_B R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T + 8 R_B \\
& R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B \\
& R_N R_S R_T^2 + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 + \\
& 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 R_T^2 + 54 R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 \\
& R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 + 9 R_A R_N R_S^3 R_T^2 + \\
& 13 R_B R_N R_S^3 R_T^2 + 3 R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 36 R_A^3 R_N R_T^3 + 156 R_A^2 R_B R_N \\
& R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 38 R_A^3 R_S R_T^3 + 114 \\
& R_A^2 R_B R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 + 168 R_A R_B R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 R_S R_T^3 \\
& + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 \\
& R_N^2 R_S^2 R_T^3 + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 26 R_A^2 R_N R_T^4 + 16 R_A \\
& R_N^2 R_T^4 + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 6 R_N R_S^2 \\
& R_T^4 + R_S^3 R_T^4)/(R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + R_A^3 R_L R_S^3 + 5 R_A^2 R_B R_L R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 \\
& R_B R_N R_S^3 + 3 R_A^2 R_L R_N R_S^3 + 10 R_A R_B R_L R_N R_S^3 + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 + 3 R_A R_L \\
& R_N^2 R_S^3 + 5 R_B R_L R_N^2 R_S^3 + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 + R_L R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B \\
& R_S^2 R_T + 8 R_A^3 R_L R_S^2 R_T + 40 R_A^2 R_B R_L R_S^2 R_T + 24 R_A^3 R_N R_S^2 R_T + 104 R_A^2 R_B R_N R_S^2 R_T + 24 \\
& R_A^2 R_L R_N R_S^2 R_T + 80 R_A R_B R_L R_N R_S^2 R_T + 24 R_A^2 R_N^2 R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 24 R_A \\
& R_L R_N^2 R_S^2 R_T + 40 R_B R_L R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T + 8 R_L R_N^3 R_S^2 R_T + 4 \\
& R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 3 R_A^2 R_L R_S^3 R_T + 10 R_A R_B R_L R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 \\
& R_A R_B R_N R_S^3 R_T + 6 R_A R_L R_N R_S^3 R_T + 10 R_B R_L R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T + 8 R_B R_N^2 R_S^3 R_T \\
& + 3 R_L R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 + 19 R_A^3 R_L R_S R_T^2 + 95 R_A^2 \\
& R_B R_L R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B R_N R_S R_T^2 + 57 R_A^2 R_L R_N R_S R_T^2 + 190 R_A R_B R_L \\
& R_N R_S R_T^2 + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + 57 R_A R_L R_N^2 R_S R_T^2 + 95 R_B R_L R_N^2 R_S \\
& R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 + 19 R_L R_N^3 R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 \\
& R_T^2 + 18 R_A^2 R_L R_S^2 R_T^2 + 52 R_A R_B R_L R_S^2 R_T^2 + 54 R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 \\
& R_A R_L R_N R_S^2 R_T^2 + 52 R_B R_L R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 18 R_L R_N^2 R_S^2 \\
& R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 + 3 R_A R_L R_S^3 R_T^2 + 5 R_B R_L R_S^3 R_T^2 + 9 \\
& R_A R_N R_S^3 R_T^2 + 13 R_B R_N R_S^3 R_T^2 + 3 R_L R_N R_S^3 R_T^2 + 3 R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B \\
& R_T^3 + 12 R_A^3 R_L R_T^3 + 60 R_A^2 R_B R_L R_T^3 + 36 R_A^3 R_N R_T^3 + 156 R_A^2 R_B R_N R_T^3 + 36 R_A^2 R_L R_N R_T^3
\end{aligned}$$

$$\begin{aligned}
& + 120 R_A R_B R_L R_N R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 R_T^3 + 36 R_A R_L R_N^2 R_T^3 + 60 R_B R_L R_N^2 R_T^3 \\
& + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 12 R_L R_N^3 R_T^3 + 38 R_A^3 R_S R_T^3 + 114 R_A^2 R_B R_S R_T^3 + 29 R_A^2 R_L R_S R_T^3 \\
& + 60 R_A R_B R_L R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 + 168 R_A R_B R_N R_S R_T^3 + 58 R_A R_L R_N R_S R_T^3 + 60 R_B R_L R_N R_S R_T^3 \\
& + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 R_S R_T^3 + 29 R_L R_N^2 R_S R_T^3 + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 \\
& + 48 R_A R_B R_S^2 R_T^3 + 12 R_A R_L R_S^2 R_T^3 + 12 R_B R_L R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 \\
& + 12 R_L R_N R_S^2 R_T^3 + 12 R_N^2 R_S^2 R_T^3 + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + R_L R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 \\
& + 12 R_A^3 R_T^4 + 10 R_A^2 R_L R_T^4 + 26 R_A^2 R_N R_T^4 + 20 R_A R_L R_N R_T^4 + 16 R_A R_N^2 R_T^4 + 10 R_L R_N^2 R_T^4 \\
& + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 10 R_A R_L R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 10 R_L R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 \\
& + 8 R_A R_S^2 R_T^4 + 2 R_L R_S^2 R_T^4 + 6 R_N R_S^2 R_T^4 + R_S^3 R_T^4)
\end{aligned}$$

$$\begin{aligned}
V_{SS2} = & \delta E R_N (R_A^2 R_B R_S^3 + 2 R_A R_B R_N R_S^3 + R_B R_N^2 R_S^3 + 8 R_A^2 R_B R_S^2 R_T + 16 R_A R_B R_N R_S^2 R_T + 8 R_B R_N^2 R_S^2 R_T \\
& + 2 R_A R_B R_S^3 R_T + 2 R_B R_N R_S^3 R_T + 19 R_A^2 R_B R_S R_T^2 + 38 R_A R_B R_N R_S R_T^2 + 19 R_B R_N^2 R_S R_T^2 \\
& + R_A^2 R_S^2 R_T^2 + 14 R_A R_B R_S^2 R_T^2 + 2 R_A R_N R_S^2 R_T^2 + 14 R_B R_N R_S^2 R_T^2 + R_N^2 R_S^2 R_T^2 \\
& + R_B R_S^3 R_T^2 + 12 R_A^2 R_B R_T^3 + 24 R_A R_B R_N R_T^3 + 12 R_B R_N^2 R_T^3 + 4 R_A^2 R_S R_T^3 + 24 R_A R_B R_S R_T^3 \\
& + 8 R_A R_N R_S R_T^3 + 24 R_B R_N R_S R_T^3 + 4 R_N^2 R_S R_T^3 + 2 R_A R_S^2 R_T^3 + 6 R_B R_S^2 R_T^3 + 2 R_N R_S^2 R_T^3 \\
& + 2 R_A^2 R_T^4 + 4 R_A R_N R_T^4 + 2 R_N^2 R_T^4 + 4 R_A R_S R_T^4 + 4 R_N R_S R_T^4 + R_S^2 R_T^4)/(R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 \\
& + R_A^3 R_L R_S^3 + 5 R_A^2 R_B R_L R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 R_B R_N R_S^3 + 3 R_A^2 R_L R_N R_S^3 + 10 R_A R_B R_L R_N R_S^3 \\
& + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 + 3 R_A R_L R_N^2 R_S^3 + 5 R_B R_L R_N^2 R_S^3 + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 \\
& + R_L R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B R_S^2 R_T + 8 R_A^3 R_L R_S^2 R_T + 40 R_A^2 R_B R_L R_S^2 R_T + 24 R_A^3 R_N R_S^2 R_T \\
& + 104 R_A^2 R_B R_N R_S^2 R_T + 24 R_A^2 R_L R_N R_S^2 R_T + 80 R_A R_B R_L R_N R_S^2 R_T + 24 R_A^2 R_N^2 R_S^2 R_T \\
& + 64 R_A R_B R_N^2 R_S^2 R_T + 24 R_A R_L R_N^2 R_S^2 R_T + 40 R_B R_L R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T \\
& + 8 R_L R_N^3 R_S^2 R_T + 4 R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 3 R_A^2 R_L R_S^3 R_T + 10 R_A R_B R_L R_S^3 R_T \\
& + 9 R_A^2 R_N R_S^3 R_T + 26 R_A R_B R_N R_S^3 R_T + 6 R_A R_L R_N R_S^3 R_T + 10 R_B R_L R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T \\
& + 8 R_B R_N^2 R_S^3 R_T + 3 R_L R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 + 19 R_A^3 R_L R_S R_T^2 \\
& + 95 R_A^2 R_B R_L R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B R_N R_S R_T^2 + 57 R_A^2 R_L R_N R_S R_T^2 + 190 R_A R_B R_L R_N R_S R_T^2 \\
& + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + 57 R_A R_L R_N^2 R_S R_T^2 + 95 R_B R_L R_N^2 R_S R_T^2 + 19 R_A R_N^3 R_S R_T^2 \\
& + 19 R_B R_N^3 R_S R_T^2 + 19 R_L R_N^3 R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 R_T^2 + 18 R_A^2 R_L R_S^2 R_T^2 + 52 R_A R_B R_L R_S^2 R_T^2 \\
& + 54 R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 R_A R_L R_N R_S^2 R_T^2 + 52 R_B R_L R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 R_T^2 \\
& + 44 R_B R_N^2 R_S^2 R_T^2 + 18 R_L R_N^2 R_S^2 R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 \\
& + 3 R_A R_L R_S^3 R_T^2 + 5 R_B R_L R_S^3 R_T^2 + 9 R_A R_N R_S^3 R_T^2 + 13 R_B R_N R_S^3 R_T^2 + 3 R_L R_N R_S^3 R_T^2 \\
& + 3 R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 12 R_A^3 R_L R_T^3 + 60 R_A^2 R_B R_L R_T^3 + 36 R_A^3 R_N R_T^3 \\
& + 156 R_A^2 R_B R_N R_T^3 + 36 R_A^2 R_L R_N R_T^3 + 120 R_A R_B R_L R_N R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 R_T^3 \\
& + 36 R_A R_L R_N^2 R_T^3 + 60 R_B R_L R_N^2 R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 12 R_L R_N^3 R_T^3 \\
& + 38 R_A^3 R_S R_T^3 + 114 R_A^2 R_B R_S R_T^3 + 29 R_A^2 R_L R_S R_T^3 + 60 R_A R_B R_L R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 \\
& + 168 R_A R_B R_N R_S R_T^3 + 58 R_A R_L R_N R_S R_T^3 + 60 R_B R_L R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 R_S R_T^3 \\
& + 29 R_L R_N^2 R_S R_T^3 + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48
\end{aligned}$$

$$\begin{aligned}
& R_A R_B R_S^2 R_T^3 + 12 R_A R_L R_S^2 R_T^3 + 12 R_B R_L R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 \\
& R_L R_N R_S^2 R_T^3 + 12 R_N^2 R_S^2 R_T^3 + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + R_L R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 \\
& R_A^3 R_T^4 + 10 R_A^2 R_L R_T^4 + 26 R_A^2 R_N R_T^4 + 20 R_A R_L R_N R_T^4 + 16 R_A R_N^2 R_T^4 + 10 R_L R_N^2 R_T^4 + 2 \\
& R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 10 R_A R_L R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 10 R_L R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + \\
& 8 R_A R_S^2 R_T^4 + 2 R_L R_S^2 R_T^4 + 6 R_N R_S^2 R_T^4 + R_S^3 R_T^4)
\end{aligned}$$

$$\begin{aligned}
V_{SS3} = & \delta E R_N (R_A^2 R_B R_S^3 + 2 R_A R_B R_N R_S^3 + R_B R_N^2 R_S^3 + 8 R_A^2 R_B R_S^2 R_T + 16 R_A R_B R_N R_S^2 R_T + 8 R_B \\
& R_N^2 R_S^2 R_T + 2 R_A R_B R_S^3 R_T + 2 R_B R_N R_S^3 R_T + 19 R_A^2 R_B R_S R_T^2 + 38 R_A R_B R_N R_S R_T^2 + 19 R_B \\
& R_N^2 R_S R_T^2 + 8 R_A R_B R_S^2 R_T^2 + 8 R_B R_N R_S^2 R_T^2 + R_B R_S^3 R_T^2 + 12 R_A^2 R_B R_T^3 + 24 R_A R_B R_N R_T^3 \\
& + 12 R_B R_N^2 R_T^3 + R_A^2 R_S R_T^3 + 6 R_A R_B R_S R_T^3 + 2 R_A R_N R_S R_T^3 + 6 R_B R_N R_S R_T^3 + R_N^2 R_S R_T^3 + \\
& 2 R_A^2 R_T^4 + 4 R_A R_N R_T^4 + 2 R_N^2 R_T^4 + R_A R_S R_T^4 + R_N R_S R_T^4)/(R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + R_A^3 R_L \\
& R_S^3 + 5 R_A^2 R_B R_L R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 R_B R_N R_S^3 + 3 R_A^2 R_L R_N R_S^3 + 10 R_A R_B R_L R_N \\
& R_S^3 + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 + 3 R_A R_L R_N^2 R_S^3 + 5 R_B R_L R_N^2 R_S^3 + R_A R_N^3 R_S^3 + R_B \\
& R_N^3 R_S^3 + R_L R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B R_S^2 R_T + 8 R_A^3 R_L R_S^2 R_T + 40 R_A^2 R_B R_L R_S^2 R_T \\
& + 24 R_A^3 R_N R_S^2 R_T + 104 R_A^2 R_B R_N R_S^2 R_T + 24 R_A^2 R_L R_N R_S^2 R_T + 80 R_A R_B R_L R_N R_S^2 R_T + 24 \\
& R_A^2 R_N^2 R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 24 R_A R_L R_N^2 R_S^2 R_T + 40 R_B R_L R_N^2 R_S^2 R_T + 8 R_A R_N^3 \\
& R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T + 8 R_L R_N^3 R_S^2 R_T + 4 R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 3 R_A^2 R_L R_S^3 R_T \\
& + 10 R_A R_B R_L R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 R_A R_B R_N R_S^3 R_T + 6 R_A R_L R_N R_S^3 R_T + 10 R_B R_L \\
& R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T + 8 R_B R_N^2 R_S^3 R_T + 3 R_L R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + \\
& 114 R_A^3 R_B R_S R_T^2 + 19 R_A^3 R_L R_S R_T^2 + 95 R_A^2 R_B R_L R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B R_N \\
& R_S R_T^2 + 57 R_A^2 R_L R_N R_S R_T^2 + 190 R_A R_B R_L R_N R_S R_T^2 + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S \\
& R_T^2 + 57 R_A R_L R_N^2 R_S R_T^2 + 95 R_B R_L R_N^2 R_S R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 + 19 R_L \\
& R_N^3 R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 R_T^2 + 18 R_A^2 R_L R_S^2 R_T^2 + 52 R_A R_B R_L R_S^2 R_T^2 + 54 \\
& R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 R_A R_L R_N R_S^2 R_T^2 + 52 R_B R_L R_N R_S^2 R_T^2 + 36 R_A R_N^2 \\
& R_S^2 R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 18 R_L R_N^2 R_S^2 R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 \\
& R_T^2 + 3 R_A R_L R_S^3 R_T^2 + 5 R_B R_L R_S^3 R_T^2 + 9 R_A R_N R_S^3 R_T^2 + 13 R_B R_N R_S^3 R_T^2 + 3 R_L R_N R_S^3 R_T^2 \\
& + 3 R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 12 R_A^3 R_L R_T^3 + 60 R_A^2 R_B R_L R_T^3 + 36 R_A^3 R_N \\
& R_T^3 + 156 R_A^2 R_B R_N R_T^3 + 36 R_A^2 R_L R_N R_T^3 + 120 R_A R_B R_L R_N R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B \\
& R_N^2 R_T^3 + 36 R_A R_L R_N^2 R_T^3 + 60 R_B R_L R_N^2 R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 12 R_L R_N^3 R_T^3 \\
& + 38 R_A^3 R_S R_T^3 + 114 R_A^2 R_B R_S R_T^3 + 29 R_A^2 R_L R_S R_T^3 + 60 R_A R_B R_L R_S R_T^3 + 85 R_A^2 R_N R_S \\
& R_T^3 + 168 R_A R_B R_N R_S R_T^3 + 58 R_A R_L R_N R_S R_T^3 + 60 R_B R_L R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 \\
& R_B R_N^2 R_S R_T^3 + 29 R_L R_N^2 R_S R_T^3 + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 12 R_A R_L \\
& R_S^2 R_T^3 + 12 R_B R_L R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 R_L R_N R_S^2 R_T^3 + 12 R_N^2 \\
& R_S^2 R_T^3 + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + R_L R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 10 R_A^2 R_L R_T^4 + \\
& 26 R_A^2 R_N R_T^4 + 20 R_A R_L R_N R_T^4 + 16 R_A R_N^2 R_T^4 + 10 R_L R_N^2 R_T^4 + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + \\
& 10 R_A R_L R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 10 R_L R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 2 R_L R_S^2 \\
& R_T^4 + 6 R_N R_S^2 R_T^4 + R_S^3 R_T^4)
\end{aligned}$$

$$\begin{aligned}
V_{SS4} = & \delta E R_N (R_A^2 R_B R_S^3 + 2 R_A R_B R_N R_S^3 + R_B R_N^2 R_S^3 + 8 R_A^2 R_B R_S^2 R_T + 16 R_A R_B R_N R_S^2 R_T + 8 R_B \\
& R_N^2 R_S^2 R_T + 2 R_A R_B R_S^3 R_T + 2 R_B R_N R_S^3 R_T + 19 R_A^2 R_B R_S R_T^2 + 38 R_A R_B R_N R_S R_T^2 + 19 R_B \\
& R_N^2 R_S R_T^2 + 8 R_A R_B R_S^2 R_T^2 + 8 R_B R_N R_S^2 R_T^2 + R_B R_S^3 R_T^2 + 12 R_A^2 R_B R_T^3 + 24 R_A R_B R_N R_T^3 \\
& + 12 R_B R_N^2 R_T^3 + 2 R_A^2 R_T^4 + 4 R_A R_N R_T^4 + 2 R_N^2 R_T^4) / (R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + R_A^3 R_L R_S^3 + 5 \\
& R_A^2 R_B R_L R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 R_B R_N R_S^3 + 3 R_A^2 R_L R_N R_S^3 + 10 R_A R_B R_L R_N R_S^3 + 3 \\
& R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 + 3 R_A R_L R_N^2 R_S^3 + 5 R_B R_L R_N^2 R_S^3 + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 + \\
& R_L R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B R_S^2 R_T + 8 R_A^3 R_L R_S^2 R_T + 40 R_A^2 R_B R_L R_S^2 R_T + 24 R_A^3 \\
& R_N R_S^2 R_T + 104 R_A^2 R_B R_N R_S^2 R_T + 24 R_A^2 R_L R_N R_S^2 R_T + 80 R_A R_B R_L R_N R_S^2 R_T + 24 R_A^2 R_N^2 \\
& R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 24 R_A R_L R_N^2 R_S^2 R_T + 40 R_B R_L R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + \\
& 8 R_B R_N^3 R_S^2 R_T + 8 R_L R_N^3 R_S^2 R_T + 4 R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 3 R_A^2 R_L R_S^3 R_T + 10 R_A \\
& R_B R_L R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 R_A R_B R_N R_S^3 R_T + 6 R_A R_L R_N R_S^3 R_T + 10 R_B R_L R_N R_S^3 R_T \\
& + 6 R_A R_N^2 R_S^3 R_T + 8 R_B R_N^2 R_S^3 R_T + 3 R_L R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 \\
& R_B R_S R_T^2 + 19 R_A^3 R_L R_S R_T^2 + 95 R_A^2 R_B R_L R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B R_N R_S R_T^2 \\
& + 57 R_A^2 R_L R_N R_S R_T^2 + 190 R_A R_B R_L R_N R_S R_T^2 + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + \\
& 57 R_A R_L R_N^2 R_S R_T^2 + 95 R_B R_L R_N^2 R_S R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 + 19 R_L R_N^3 \\
& R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 R_T^2 + 18 R_A^2 R_L R_S^2 R_T^2 + 52 R_A R_B R_L R_S^2 R_T^2 + 54 R_A^2 \\
& R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 R_A R_L R_N R_S^2 R_T^2 + 52 R_B R_L R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 \\
& R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 18 R_L R_N^2 R_S^2 R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 \\
& + 3 R_A R_L R_S^3 R_T^2 + 5 R_B R_L R_S^3 R_T^2 + 9 R_A R_N R_S^3 R_T^2 + 13 R_B R_N R_S^3 R_T^2 + 3 R_L R_N R_S^3 R_T^2 + 3 \\
& R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 12 R_A^3 R_L R_T^3 + 60 R_A^2 R_B R_L R_T^3 + 36 R_A^3 R_N R_T^3 + \\
& 156 R_A^2 R_B R_N R_T^3 + 36 R_A^2 R_L R_N R_T^3 + 120 R_A R_B R_L R_N R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 \\
& R_T^3 + 36 R_A R_L R_N^2 R_T^3 + 60 R_B R_L R_N^2 R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 12 R_L R_N^3 R_T^3 + 38 \\
& R_A^3 R_S R_T^3 + 114 R_A^2 R_B R_S R_T^3 + 29 R_A^2 R_L R_S R_T^3 + 60 R_A R_B R_L R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 + \\
& 168 R_A R_B R_N R_S R_T^3 + 58 R_A R_L R_N R_S R_T^3 + 60 R_B R_L R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 \\
& R_S R_T^3 + 29 R_L R_N^2 R_S R_T^3 + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 12 R_A R_L R_S^2 R_T^3 \\
& + 12 R_B R_L R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 R_L R_N R_S^2 R_T^3 + 12 R_N^2 R_S^2 R_T^3 + \\
& 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + R_L R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 10 R_A^2 R_L R_T^4 + 26 R_A^2 \\
& R_N R_T^4 + 20 R_A R_L R_N R_T^4 + 16 R_A R_N^2 R_T^4 + 10 R_L R_N^2 R_T^4 + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 10 R_A \\
& R_L R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 10 R_L R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 2 R_L R_S^2 R_T^4 + 6 \\
& R_N R_S^2 R_T^4 + R_S^3 R_T^4)
\end{aligned}$$

$$\begin{aligned}
V_{SS14} = & \delta E (R_A^2 R_S + 6 R_A R_B R_S + R_A R_N R_S + 2 R_B R_N R_S + 3 R_A^2 R_T + 18 R_A R_B R_T + 3 R_A R_N R_T + 6 R_B \\
& R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_N R_S R_T + 3 R_A R_T^2 + R_N R_T^2 + R_S R_T^2) / (R_A^2 R_S + 6 R_A R_B R_S \\
& + R_A R_L R_S + 4 R_B R_L R_S + R_A R_N R_S + 2 R_B R_N R_S + R_L R_N R_S + 3 R_A^2 R_T + 18 R_A R_B R_T + 3 R_A \\
& R_L R_T + 12 R_B R_L R_T + 3 R_A R_N R_T + 6 R_B R_N R_T + 3 R_L R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_L R_S \\
& R_T + R_N R_S R_T + 3 R_A R_T^2 + 2 R_L R_T^2 + R_N R_T^2 + R_S R_T^2)
\end{aligned}$$

$$V_{SS25} = \frac{\delta E R_N (2 R_B R_S + 6 R_B R_T + R_T^2)}{(R_A^2 R_S + 6 R_A R_B R_S + R_A R_L R_S + 4 R_B R_L R_S + R_A R_N R_S + 2 R_B R_N R_S + R_L R_N R_S + 3 R_A^2 R_T + 18 R_A R_B R_T + 3 R_A R_L R_T + 12 R_B R_L R_T + 3 R_A R_N R_T + 6 R_B R_N R_T + 3 R_L R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_L R_S R_T + R_N R_S R_T + 3 R_A R_T^2 + 2 R_L R_T^2 + R_N R_T^2 + R_S R_T^2)}$$

$$V_{SS135} = \frac{\delta E (R_A^2 R_S + 6 R_A R_B R_S + R_A R_N R_S + 3 R_B R_N R_S + 4 R_A^2 R_T + 24 R_A R_B R_T + 4 R_A R_N R_T + 12 R_B R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_N R_S R_T + 4 R_A R_T^2 + 2 R_N R_T^2 + R_S R_T^2)}{(R_A^2 R_S + 6 R_A R_B R_S + R_A R_L R_S + 3 R_B R_L R_S + R_A R_N R_S + 3 R_B R_N R_S + R_L R_N R_S + 4 R_A^2 R_T + 24 R_A R_B R_T + 4 R_A R_L R_T + 12 R_B R_L R_T + 4 R_A R_N R_T + 12 R_B R_N R_T + 4 R_L R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_L R_S R_T + R_N R_S R_T + 4 R_A R_T^2 + 2 R_L R_T^2 + 2 R_N R_T^2 + R_S R_T^2)}$$

$$V_{SS246} = \frac{\delta E R_N (3 R_B R_S + 12 R_B R_T + 2 R_T^2)}{(R_A^2 R_S + 6 R_A R_B R_S + R_A R_L R_S + 3 R_B R_L R_S + R_A R_N R_S + 3 R_B R_N R_S + R_L R_N R_S + 4 R_A^2 R_T + 24 R_A R_B R_T + 4 R_A R_L R_T + 12 R_B R_L R_T + 4 R_A R_N R_T + 12 R_B R_N R_T + 4 R_L R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_L R_S R_T + R_N R_S R_T + 4 R_A R_T^2 + 2 R_L R_T^2 + 2 R_N R_T^2 + R_S R_T^2)}$$

$$V_{SS123456} = \frac{\delta E (R_A + 6 R_B + R_T)}{(R_A + 6 R_B + R_L + R_T)}$$

$$Z_{SS0} = \frac{R_N (R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 R_B R_N R_S^3 + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B R_S^2 R_T + 24 R_A^3 R_N R_S^2 R_T + 104 R_A^2 R_B R_N R_S^2 R_T + 24 R_A^2 R_N^2 R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T + 4 R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 R_A R_B R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T + 8 R_B R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B R_N R_S R_T^2 + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 R_T^2 + 54 R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 + 9 R_A R_N R_S^3 R_T^2 + 13 R_B R_N R_S^3 R_T^2 + 3 R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 36 R_A^3 R_N R_T^3 + 156 R_A^2 R_B R_N R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 38 R_A^3 R_S R_T^3 + 114 R_A^2 R_B R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 + 168 R_A R_B R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 R_S R_T^3 + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 R_N^2 R_S^2 R_T^3 + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 26 R_A^2 R_N R_T^4 + 16 R_A R_N^2 R_T^4 + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 6 R_N R_S^2 R_T^4 + R_S^3 R_T^4)}{(R_A + 6 R_B + R_N + R_T) (R_A R_S + R_N R_S + R_A R_T + R_N R_T + R_S R_T) (R_A R_S + R_N R_S + 3 R_A R_T + 3 R_N R_T + R_S R_T) (R_A R_S + R_N R_S + 4 R_A R_T + 4 R_N R_T + R_S R_T)}$$

$$\begin{aligned}
Z_{SS1} = & R_L (R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 R_B R_N R_S^3 + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 \\
& + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B R_S^2 R_T + 24 R_A^3 R_N R_S^2 R_T + 104 R_A^2 R_B \\
& R_N R_S^2 R_T + 24 R_A^2 R_N^2 R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T + 4 \\
& R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 R_A R_B R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T + 8 R_B \\
& R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B \\
& R_N R_S R_T^2 + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 + \\
& 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 R_T^2 + 54 R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 \\
& R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 + 9 R_A R_N R_S^3 R_T^2 + \\
& 13 R_B R_N R_S^3 R_T^2 + 3 R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 36 R_A^3 R_N R_T^3 + 156 R_A^2 R_B R_N \\
& R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 38 R_A^3 R_S R_T^3 + 114 \\
& R_A^2 R_B R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 + 168 R_A R_B R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 R_S R_T^3 \\
& + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 \\
& R_N^2 R_S^2 R_T^3 + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 26 R_A^2 R_N R_T^4 + 16 R_A \\
& R_N^2 R_T^4 + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 6 R_N R_S^2 \\
& R_T^4 + R_S^3 R_T^4)/(R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + R_A^3 R_L R_S^3 + 5 R_A^2 R_B R_L R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 \\
& R_B R_N R_S^3 + 3 R_A^2 R_L R_N R_S^3 + 10 R_A R_B R_L R_N R_S^3 + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 + 3 R_A R_L \\
& R_N^2 R_S^3 + 5 R_B R_L R_N^2 R_S^3 + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 + R_L R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B \\
& R_S^2 R_T + 8 R_A^3 R_L R_S^2 R_T + 40 R_A^2 R_B R_L R_S^2 R_T + 24 R_A^3 R_N R_S^2 R_T + 104 R_A^2 R_B R_N R_S^2 R_T + 24 \\
& R_A^2 R_L R_N R_S^2 R_T + 80 R_A R_B R_L R_N R_S^2 R_T + 24 R_A^2 R_N^2 R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 24 R_A \\
& R_L R_N^2 R_S^2 R_T + 40 R_B R_L R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T + 8 R_L R_N^3 R_S^2 R_T + 4 \\
& R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 3 R_A^2 R_L R_S^3 R_T + 10 R_A R_B R_L R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 \\
& R_A R_B R_N R_S^3 R_T + 6 R_A R_L R_N R_S^3 R_T + 10 R_B R_L R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T + 8 R_B R_N^2 R_S^3 R_T \\
& + 3 R_L R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 + 19 R_A^3 R_L R_S R_T^2 + 95 R_A^2 \\
& R_B R_L R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B R_N R_S R_T^2 + 57 R_A^2 R_L R_N R_S R_T^2 + 190 R_A R_B R_L \\
& R_N R_S R_T^2 + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + 57 R_A R_L R_N^2 R_S R_T^2 + 95 R_B R_L R_N^2 R_S \\
& R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 + 19 R_L R_N^3 R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 \\
& R_T^2 + 18 R_A^2 R_L R_S^2 R_T^2 + 52 R_A R_B R_L R_S^2 R_T^2 + 54 R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 + 36 \\
& R_A R_L R_N R_S^2 R_T^2 + 52 R_B R_L R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 18 R_L R_N^2 R_S^2 \\
& R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 + 3 R_A R_L R_S^3 R_T^2 + 5 R_B R_L R_S^3 R_T^2 + 9 \\
& R_A R_N R_S^3 R_T^2 + 13 R_B R_N R_S^3 R_T^2 + 3 R_L R_N R_S^3 R_T^2 + 3 R_N^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B \\
& R_T^3 + 12 R_A^3 R_L R_T^3 + 60 R_A^2 R_B R_L R_T^3 + 36 R_A^3 R_N R_T^3 + 156 R_A^2 R_B R_N R_T^3 + 36 R_A^2 R_L R_N R_T^3 \\
& + 120 R_A R_B R_L R_N R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 R_T^3 + 36 R_A R_L R_N^2 R_T^3 + 60 R_B R_L R_N^2 \\
& R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 12 R_L R_N^3 R_T^3 + 38 R_A^3 R_S R_T^3 + 114 R_A^2 R_B R_S R_T^3 + 29 \\
& R_A^2 R_L R_S R_T^3 + 60 R_A R_B R_L R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 + 168 R_A R_B R_N R_S R_T^3 + 58 R_A R_L R_N R_S \\
& R_T^3 + 60 R_B R_L R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 R_S R_T^3 + 29 R_L R_N^2 R_S R_T^3 + 9 R_N^3 R_S \\
& R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 12 R_A R_L R_S^2 R_T^3 + 12 R_B R_L R_S^2 R_T^3 + 36 R_A R_N R_S^2 \\
& R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 R_L R_N R_S^2 R_T^3 + 12 R_N^2 R_S^2 R_T^3 + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + R_L
\end{aligned}$$

$$R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 10 R_A^2 R_L R_T^4 + 26 R_A^2 R_N R_T^4 + 20 R_A R_L R_N R_T^4 + 16 R_A R_N^2 R_T^4 + 10 R_L R_N^2 R_T^4 + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 10 R_A R_L R_S R_T^4 + 28 R_A R_N R_S R_T^4 + 10 R_L R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 2 R_L R_S^2 R_T^4 + 6 R_N R_S^2 R_T^4 + R_S^3 R_T^4)$$

$$\begin{aligned} Z_{SS123456} = & R_L (R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + 3 R_A^3 R_L R_S^3 + 13 R_A^2 R_B R_L R_S^3 + 3 R_A^2 R_L^2 R_S^3 + 8 R_A R_B R_L^2 R_S^3 \\ & + R_A R_L^3 R_S^3 + R_B R_L^3 R_S^3 + 8 R_A^4 R_S^2 R_T + 48 R_A^3 R_B R_S^2 R_T + 24 R_A^3 R_L R_S^2 R_T + 104 R_A^2 R_B R_L R_S^2 R_T \\ & + 24 R_A^2 R_L^2 R_S^2 R_T + 64 R_A R_B R_L^2 R_S^2 R_T + 8 R_A R_L^3 R_S^2 R_T + 8 R_B R_L^3 R_S^2 R_T + 4 R_A^3 R_S^3 R_T \\ & + 18 R_A^2 R_B R_S^3 R_T + 9 R_A^2 R_L R_S^3 R_T + 26 R_A R_B R_L R_S^3 R_T + 6 R_A R_L^2 R_S^3 R_T + 8 R_B R_L^2 R_S^3 R_T \\ & + R_L^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 + 57 R_A^3 R_L R_S R_T^2 + 247 R_A^2 R_B R_L R_S R_T^2 \\ & + 57 R_A^2 R_L^2 R_S R_T^2 + 152 R_A R_B R_L^2 R_S R_T^2 + 19 R_A R_L^3 R_S R_T^2 + 19 R_B R_L^3 R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 \\ & + 96 R_A^2 R_B R_S^2 R_T^2 + 54 R_A^2 R_L R_S^2 R_T^2 + 140 R_A R_B R_L R_S^2 R_T^2 + 36 R_A R_L^2 R_S^2 R_T^2 + 44 R_B R_L^2 R_S^2 R_T^2 \\ & + 6 R_L^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 + 9 R_A R_L R_S^3 R_T^2 + 13 R_B R_L R_S^3 R_T^2 \\ & + 3 R_L^2 R_S^3 R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 36 R_A^3 R_L R_T^3 + 156 R_A^2 R_B R_L R_T^3 + 36 R_A^2 R_L^2 R_T^3 \\ & + 96 R_A R_B R_L^2 R_T^3 + 12 R_A R_L^3 R_T^3 + 12 R_B R_L^3 R_T^3 + 38 R_A^3 R_S R_T^3 + 114 R_A^2 R_B R_S R_T^3 \\ & + 85 R_A^2 R_L R_S R_T^3 + 168 R_A R_B R_L R_S R_T^3 + 56 R_A R_L^2 R_S R_T^3 + 54 R_B R_L^2 R_S R_T^3 + 9 R_L^3 R_S R_T^3 \\ & + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 36 R_A R_L R_S^2 R_T^3 + 36 R_B R_L R_S^2 R_T^3 + 12 R_L^2 R_S^2 R_T^3 \\ & + 4 R_A R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + 3 R_L R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 26 R_A^2 R_L R_T^4 + 16 R_A R_L^2 R_T^4 + 2 R_L^3 R_T^4 \\ & + 19 R_A^2 R_S R_T^4 + 28 R_A R_L R_S R_T^4 + 9 R_L^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 6 R_L R_S^2 R_T^4 + R_S^3 R_T^4) \\ & / ((R_A + 6 R_B + R_L + R_T) (R_A R_S + R_L R_S + R_A R_T + R_L R_T + R_S R_T) (R_A R_S + R_L R_S + 3 R_A R_T + 3 R_L R_T + R_S R_T) \\ & (R_A R_S + R_L R_S + 4 R_A R_T + 4 R_L R_T + R_S R_T)) \end{aligned}$$

$$\begin{aligned} V_{CS1} = & \delta E R_B (R_A R_S + R_N R_S + R_A R_T + R_N R_T + R_S R_T) (R_A R_S + R_N R_S + 3 R_A R_T + 3 R_N R_T + R_S R_T) (R_A R_S + R_N R_S + 4 R_A R_T + 4 R_N R_T + R_S R_T) \\ & / (R_A^4 R_S^3 + 6 R_A^3 R_B R_S^3 + R_A^3 R_L R_S^3 + 5 R_A^2 R_B R_L R_S^3 + 3 R_A^3 R_N R_S^3 + 13 R_A^2 R_B R_N R_S^3 + 3 R_A^2 R_L R_N R_S^3 \\ & + 10 R_A R_B R_L R_N R_S^3 + 3 R_A^2 R_N^2 R_S^3 + 8 R_A R_B R_N^2 R_S^3 + 3 R_A R_L R_N^2 R_S^3 + 5 R_B R_L R_N^2 R_S^3 + R_A R_N^3 R_S^3 + R_B R_N^3 R_S^3 + R_L R_N^3 R_S^3 + 8 R_A^4 R_S^2 R_T \\ & + 48 R_A^3 R_B R_S^2 R_T + 8 R_A^3 R_L R_S^2 R_T + 40 R_A^2 R_B R_L R_S^2 R_T + 24 R_A^3 R_N R_S^2 R_T + 104 R_A^2 R_B R_N R_S^2 R_T + 24 R_A^2 R_L R_N R_S^2 R_T \\ & + 80 R_A R_B R_L R_N R_S^2 R_T + 24 R_A^2 R_N^2 R_S^2 R_T + 64 R_A R_B R_N^2 R_S^2 R_T + 24 R_A R_L R_N^2 R_S^2 R_T + 40 R_B R_L R_N^2 R_S^2 R_T + 8 R_A R_N^3 R_S^2 R_T + 8 R_B R_N^3 R_S^2 R_T \\ & + 8 R_L R_N^3 R_S^2 R_T + 4 R_A^3 R_S^3 R_T + 18 R_A^2 R_B R_S^3 R_T + 3 R_A^2 R_L R_S^3 R_T + 10 R_A R_B R_L R_S^3 R_T + 9 R_A^2 R_N R_S^3 R_T + 26 R_A R_B R_N R_S^3 R_T \\ & + 6 R_A R_L R_N R_S^3 R_T + 10 R_B R_L R_N R_S^3 R_T + 6 R_A R_N^2 R_S^3 R_T + 8 R_B R_N^2 R_S^3 R_T + 3 R_L R_N^2 R_S^3 R_T + R_N^3 R_S^3 R_T + 19 R_A^4 R_S R_T^2 + 114 R_A^3 R_B R_S R_T^2 \\ & + 19 R_A^3 R_L R_S R_T^2 + 95 R_A^2 R_B R_L R_S R_T^2 + 57 R_A^3 R_N R_S R_T^2 + 247 R_A^2 R_B R_N R_S R_T^2 + 57 R_A^2 R_L R_N R_S R_T^2 + 190 R_A R_B R_L R_N R_S R_T^2 \\ & + 57 R_A^2 R_N^2 R_S R_T^2 + 152 R_A R_B R_N^2 R_S R_T^2 + 57 R_A R_L R_N^2 R_S R_T^2 + 95 R_B R_L R_N^2 R_S R_T^2 + 19 R_A R_N^3 R_S R_T^2 + 19 R_B R_N^3 R_S R_T^2 \\ & + 19 R_L R_N^3 R_S R_T^2 + 24 R_A^3 R_S^2 R_T^2 + 96 R_A^2 R_B R_S^2 R_T^2 + 18 R_A^2 R_L R_S^2 R_T^2 + 52 R_A R_B R_L R_S^2 R_T^2 + 54 R_A^2 R_N R_S^2 R_T^2 + 140 R_A R_B R_N R_S^2 R_T^2 \\ & + 36 R_A R_L R_N R_S^2 R_T^2 + 52 R_B R_L R_N R_S^2 R_T^2 + 36 R_A R_N^2 R_S^2 R_T^2 + 44 R_B R_N^2 R_S^2 R_T^2 + 18 R_L R_N^2 R_S^2 R_T^2 + 6 R_N^3 R_S^2 R_T^2 + 6 R_A^2 R_S^3 R_T^2 + 18 R_A R_B R_S^3 R_T^2 + 3 R_A R_L R_S^3 R_T^2) \end{aligned}$$

$$\begin{aligned}
& R_S^3 R_T^2 + 5 R_B R_L R_S^3 R_T^2 + 9 R_A R_N R_S^3 R_T^2 + 13 R_B R_N R_S^3 R_T^2 + 3 R_L R_N R_S^3 R_T^2 + 3 R_N^2 R_S^3 \\
& R_T^2 + 12 R_A^4 R_T^3 + 72 R_A^3 R_B R_T^3 + 12 R_A^3 R_L R_T^3 + 60 R_A^2 R_B R_L R_T^3 + 36 R_A^3 R_N R_T^3 + 156 R_A^2 \\
& R_B R_N R_T^3 + 36 R_A^2 R_L R_N R_T^3 + 120 R_A R_B R_L R_N R_T^3 + 36 R_A^2 R_N^2 R_T^3 + 96 R_A R_B R_N^2 R_T^3 + 36 \\
& R_A R_L R_N^2 R_T^3 + 60 R_B R_L R_N^2 R_T^3 + 12 R_A R_N^3 R_T^3 + 12 R_B R_N^3 R_T^3 + 12 R_L R_N^3 R_T^3 + 38 R_A^3 R_S \\
& R_T^3 + 114 R_A^2 R_B R_S R_T^3 + 29 R_A^2 R_L R_S R_T^3 + 60 R_A R_B R_L R_S R_T^3 + 85 R_A^2 R_N R_S R_T^3 + 168 R_A \\
& R_B R_N R_S R_T^3 + 58 R_A R_L R_N R_S R_T^3 + 60 R_B R_L R_N R_S R_T^3 + 56 R_A R_N^2 R_S R_T^3 + 54 R_B R_N^2 R_S R_T^3 \\
& + 29 R_L R_N^2 R_S R_T^3 + 9 R_N^3 R_S R_T^3 + 24 R_A^2 R_S^2 R_T^3 + 48 R_A R_B R_S^2 R_T^3 + 12 R_A R_L R_S^2 R_T^3 + 12 \\
& R_B R_L R_S^2 R_T^3 + 36 R_A R_N R_S^2 R_T^3 + 36 R_B R_N R_S^2 R_T^3 + 12 R_L R_N R_S^2 R_T^3 + 12 R_N^2 R_S^2 R_T^3 + 4 R_A \\
& R_S^3 R_T^3 + 6 R_B R_S^3 R_T^3 + R_L R_S^3 R_T^3 + 3 R_N R_S^3 R_T^3 + 12 R_A^3 R_T^4 + 10 R_A^2 R_L R_T^4 + 26 R_A^2 R_N R_T^4 \\
& + 20 R_A R_L R_N R_T^4 + 16 R_A R_N^2 R_T^4 + 10 R_L R_N^2 R_T^4 + 2 R_N^3 R_T^4 + 19 R_A^2 R_S R_T^4 + 10 R_A R_L R_S \\
& R_T^4 + 28 R_A R_N R_S R_T^4 + 10 R_L R_N R_S R_T^4 + 9 R_N^2 R_S R_T^4 + 8 R_A R_S^2 R_T^4 + 2 R_L R_S^2 R_T^4 + 6 R_N R_S^2 \\
& R_T^4 + R_S^3 R_T^4)
\end{aligned}$$

$$\begin{aligned}
V_{CS14} = & 2 \delta E R_B (R_A R_S + R_N R_S + 3 R_A R_T + 3 R_N R_T + R_S R_T) / (R_A^2 R_S + 6 R_A R_B R_S + R_A R_L R_S + 4 R_B R_L \\
& R_S + R_A R_N R_S + 2 R_B R_N R_S + R_L R_N R_S + 3 R_A^2 R_T + 18 R_A R_B R_T + 3 R_A R_L R_T + 12 R_B R_L R_T + \\
& 3 R_A R_N R_T + 6 R_B R_N R_T + 3 R_L R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_L R_S R_T + R_N R_S R_T + 3 R_A \\
& R_T^2 + 2 R_L R_T^2 + R_N R_T^2 + R_S R_T^2)
\end{aligned}$$

$$\begin{aligned}
V_{CS135} = & 3 \delta E R_B (R_A R_S + R_N R_S + 4 R_A R_T + 4 R_N R_T + R_S R_T) / (R_A^2 R_S + 6 R_A R_B R_S + R_A R_L R_S + 3 R_B R_L \\
& R_S + R_A R_N R_S + 3 R_B R_N R_S + R_L R_N R_S + 4 R_A^2 R_T + 24 R_A R_B R_T + 4 R_A R_L R_T + 12 R_B R_L R_T + \\
& 4 R_A R_N R_T + 12 R_B R_N R_T + 4 R_L R_N R_T + 2 R_A R_S R_T + 6 R_B R_S R_T + R_L R_S R_T + R_N R_S R_T + 4 R_A \\
& R_T^2 + 2 R_L R_T^2 + 2 R_N R_T^2 + R_S R_T^2)
\end{aligned}$$

$$V_{CS123456} = 6 \delta E R_B / (R_A + 6 R_B + R_L + R_T)$$

C.4 Équations des paramètres

Cette section contient les équations des paramètres δE , R_N , R_L , R_B , R_A et R_S .

$$\delta E = \frac{((R_L R_N R_S + 3 R_L R_N R_T + R_L R_S R_T + R_N R_S R_T + 2 R_L R_T^2 + R_N R_T^2 + R_S R_T^2 + R_A^2 (R_S + 3 R_T) + 2 R_B (2 R_L R_S + R_N R_S + 6 R_L R_T + 3 R_N R_T + 3 R_S R_T) + R_A (R_L R_S + R_N R_S + 3 R_L R_T + 3 R_N R_T + 2 R_S R_T + 3 R_T^2 + 6 R_B (R_S + 3 R_T))) V_{SS25}) / (R_N (2 R_B R_S + 6 R_B R_T + R_T^2))$$

$$R_N = \frac{-(((R_A + 6 R_B + R_T) (R_S R_T + R_A (R_S + 3 R_T)) V_{SS25}) / (R_A R_S V_{SS25} + 2 R_B (R_S + 3 R_T) (-V_{SS14} + V_{SS25}) + R_T (-R_T V_{SS14} + (3 R_A + R_S + R_T) V_{SS25})))$$

$$R_L = \frac{-(((R_A + 6 R_B + R_T) (R_S R_T + R_A (R_S + 3 R_T)) (V_{SS14} - V_{SS123456})) / (4 R_B (R_S + 3 R_T) (V_{SS14} - V_{SS25}) + R_A (R_S + 3 R_T) (V_{SS14} - V_{SS123456}) + R_T (R_S V_{SS14} + 2 R_T V_{SS14} - 2 R_T V_{SS25} - R_S V_{SS123456})))$$

$$R_B = \frac{(2 R_T^2 (-((R_S R_T V_{SS25} + R_A (R_S V_{SS25} + R_T (V_{SS14} + 2 V_{SS25}))) V_{SS246}) + (R_S R_T + R_A (R_S + 3 R_T)) (2 V_{SS25} - V_{SS246}) V_{SS123456})) / ((R_A (R_S + 3 R_T) (R_S + 4 R_T) (V_{SS14} + 2 V_{SS25}) + R_S R_T (R_S V_{SS14} + 2 (R_S + 6 R_T) V_{SS25})) V_{SS246} + 3 (R_S + 4 R_T) (R_S R_T + R_A (R_S + 3 R_T)) (-2 V_{SS25} + V_{SS246}) V_{SS123456})$$

$$R_A = \frac{(R_S R_T (V_{SS1} V_{SS25} (V_{SS14} + 2 V_{SS25})^3 V_{SS246}^2 - (V_{SS14} + 2 V_{SS25})^2 V_{SS246} (2 V_{SS1} V_{SS25} (V_{SS14} + 2 V_{SS25}) + (-2 V_{SS1} V_{SS14} + V_{SS14}^2 + 3 V_{SS1} V_{SS25} - 4 V_{SS14} V_{SS25} + 2 V_{SS25}^2) V_{SS246}) V_{SS123456} + (V_{SS14} + 2 V_{SS25}) (2 V_{SS25} - V_{SS246}) (-2 (V_{SS14} - V_{SS25}) (V_{SS14} + 2 V_{SS25}) V_{SS246} + V_{SS1} (-2 V_{SS14} V_{SS25} + 2 V_{SS25}^2 + 4 V_{SS14} V_{SS246} + 5 V_{SS25} V_{SS246})) V_{SS123456}^2 + (2 V_{SS1} V_{SS14} - V_{SS14}^2 + 7 V_{SS1} V_{SS25} - 6 V_{SS14} V_{SS25} - 2 V_{SS25}^2) (-2 V_{SS25} + V_{SS246})^2 V_{SS123456}^3) / (- (V_{SS1} (V_{SS14} + 2 V_{SS25})^3 (R_S V_{SS25} + R_T (V_{SS14} + 2 V_{SS25})) V_{SS246}^2) + (V_{SS14} + 2 V_{SS25})^2 V_{SS246} (2 (R_S + 3 R_T) V_{SS1} V_{SS25} (V_{SS14} + 2 V_{SS25}) + (V_{SS14} (R_T (-3 V_{SS1} + V_{SS14}) + R_S (-2 V_{SS1} + V_{SS14})) + (R_S + 2 R_T) (3 V_{SS1} - 4 V_{SS14}) V_{SS25} + 2 (R_S + 2 R_T) V_{SS25}^2) V_{SS246}) V_{SS123456} + (V_{SS14} + 2 V_{SS25}) (2 V_{SS25} - V_{SS246}) (2 (R_S + 4 R_T) V_{SS1} (V_{SS14} - V_{SS25}) V_{SS25} - (R_S (-2 (V_{SS14} - V_{SS25}) (V_{SS14} + 2 V_{SS25}) + V_{SS1} (4 V_{SS14} + 5 V_{SS25})) + R_T (9 V_{SS1} (V_{SS14} + 2 V_{SS25}) - 2 (V_{SS14} - V_{SS25}) (V_{SS14} + 6 V_{SS25}))) V_{SS246}) V_{SS123456}^2 - (V_{SS14} ((2 R_S + 5 R_T) V_{SS1} - (R_S + R_T) V_{SS14}) + ((7 R_S + 22 R_T) V_{SS1} - 6 (R_S + 3 R_T) V_{SS14}) V_{SS25} - 2 (R_S + 4 R_T) V_{SS25}^2) (-2 V_{SS25} + V_{SS246})^2 V_{SS123456}^3)$$

$$R_S = \frac{(2 R_T V_{SS123456} (V_{SS1} (V_{SS14} + 2 V_{SS25} - V_{SS123456}) - 2 V_{SS25} V_{SS123456}) (-((V_{SS1} - V_{SS14}) (V_{SS14} + 2 V_{SS25})^2 V_{SS246}^2) + (V_{SS14} + 2 V_{SS25}) (-2 V_{SS25} + V_{SS246}) (-2 V_{SS1} V_{SS25} + V_{SS14} V_{SS246}) V_{SS123456} + (V_{SS1} - 2 (V_{SS14} + V_{SS25})) (-2 V_{SS25} + V_{SS246})^2 V_{SS123456}^2) (- (V_{SS1} (V_{SS14} + 2 V_{SS25})^4 V_{SS246}^2) + (V_{SS14} + 2 V_{SS25})^2 V_{SS246} (6 V_{SS1} V_{SS25} (V_{SS14} + 2 V_{SS25}) + (-3 V_{SS1} V_{SS14} + V_{SS14}^2 + 6 V_{SS1} V_{SS25} - 8 V_{SS14} V_{SS25} + 4 V_{SS25}^2) V_{SS246}) V_{SS123456} + (V_{SS14} + 2 V_{SS25}) (2 V_{SS25} - V_{SS246}) (8 V_{SS1} (V_{SS14} - V_{SS25}) V_{SS25}$$

$$\begin{aligned}
& - 9 V_{SS1} (V_{SS14} + 2 V_{SS25}) V_{SS246} + 2 (V_{SS14} - V_{SS25}) (V_{SS14} + 6 V_{SS25}) V_{SS246} V_{SS123456}^2 - (5 V_{SS1} \\
& V_{SS14} - V_{SS14}^2 + 22 V_{SS1} V_{SS25} - 18 V_{SS14} V_{SS25} - 8 V_{SS25}^2) (-2 V_{SS25} + V_{SS246})^2 V_{SS123456}^3)/(-2 \\
& V_{SS123456} (V_{SS1} (V_{SS14} + 2 V_{SS25} - V_{SS123456}) - 2 V_{SS25} V_{SS123456}) - ((V_{SS1} - V_{SS14}) (V_{SS14} + 2 V_{SS25})^2 \\
& V_{SS246}^2) + (V_{SS14} + 2 V_{SS25}) (-2 V_{SS25} + V_{SS246}) (-2 V_{SS1} V_{SS25} + V_{SS14} V_{SS246}) V_{SS123456} + (V_{SS1} - 2 \\
& (V_{SS14} + V_{SS25})) (-2 V_{SS25} + V_{SS246})^2 V_{SS123456}^2) - (V_{SS1} V_{SS25} (V_{SS14} + 2 V_{SS25})^3 V_{SS246}^2) + (V_{SS14} + 2 \\
& V_{SS25})^2 V_{SS246} (2 V_{SS1} V_{SS25} (V_{SS14} + 2 V_{SS25}) + (-2 V_{SS1} V_{SS14} + V_{SS14}^2 + 3 V_{SS1} V_{SS25} - 4 V_{SS14} V_{SS25} \\
& + 2 V_{SS25}^2) V_{SS246}) V_{SS123456} + (V_{SS14} + 2 V_{SS25}) (2 V_{SS25} - V_{SS246}) (2 (V_{SS14} - V_{SS25}) (V_{SS14} + 2 V_{SS25}) \\
& V_{SS246} + V_{SS1} (2 V_{SS14} (V_{SS25} - 2 V_{SS246}) - V_{SS25} (2 V_{SS25} + 5 V_{SS246}))) V_{SS123456}^2 - (2 V_{SS1} V_{SS14} - \\
& V_{SS14}^2 + 7 V_{SS1} V_{SS25} - 6 V_{SS14} V_{SS25} - 2 V_{SS25}^2) (-2 V_{SS25} + V_{SS246})^2 V_{SS123456}^3) + V_{B1} (V_{SS14} + 2 \\
& V_{SS25}) (V_{SS14} - V_{SS123456}) (V_{SS14} V_{SS246} + 2 V_{SS25} (V_{SS246} - 3 V_{SS123456}) + 3 V_{SS246} V_{SS123456}) (V_{SS1} \\
& (V_{SS14} + 2 V_{SS25})^2 V_{SS246} - (4 V_{SS1} - 3 V_{SS14} + 2 V_{SS25}) (V_{SS14} + 2 V_{SS25}) V_{SS246} V_{SS123456} + 3 (V_{SS1} - \\
& V_{SS14}) (-2 V_{SS25} + V_{SS246}) V_{SS123456}^2) (V_{SS1} (V_{SS14} + 2 V_{SS25})^2 V_{SS246} - 2 (V_{SS14} + 2 V_{SS25}) (V_{SS1} V_{SS25} \\
& + (V_{SS1} - V_{SS14} + V_{SS25}) V_{SS246}) V_{SS123456} + (V_{SS1} - 2 (V_{SS14} + V_{SS25})) (-2 V_{SS25} + V_{SS246}) V_{SS123456}^2) \\
& (V_{SS14} V_{SS246} - V_{SS246} V_{SS123456} + 2 V_{SS25} (V_{SS246} + V_{SS123456})))
\end{aligned}$$