

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE DE LA PRODUCTION AUTOMATISÉE
M.Eng.

PAR
Alexandre JUMELINE

UTILISATION DE SIGNAUX ÉLECTROMYOGRAPHIQUES POUR LE CONTRÔLE
D'UN BRAS ROBOTIQUE POUR LES USAGERS DE FAUTEUILS ROULANTS
MOTORISÉS.

MONTRÉAL, LE 01 OCTOBRE 2013



Alexandre Jumeline, 2013



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Rachid Aissaoui, directeur de mémoire
Département de Génie de la Production Automatisée

M. Pascal Bigras, président du jury
Département de Génie de la Production Automatisée

M. Maxime Raison, examinateur externe
Professeur à l'école Polytechnique de Montréal

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 23 SEPTEMBRE 2013

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

AVANT-PROPOS

Ce projet a été réalisé en partenariat avec la société Canadienne Kinova dans le but d'ajouter une interface de contrôle à leur bras robotique Jaco contrôlable par le biais d'un Joystick.

Le but principal était de prendre la suite du projet de maîtrise de Véronique Maheu qui consistait en l'étude de la performance de différents classificateurs dans le cadre de la classification de mouvements à partir de huit canaux électromyographiques. Le projet devait réaliser un prototype permettant le contrôle temps réel d'un bras robotique grâce à quatre canaux électromyographiques.

REMERCIEMENTS

Dans un premier temps, je tiens à remercier tout particulièrement le professeur Rachid Aissaoui de m'avoir donné la chance de réaliser ce projet avec lui. C'est un directeur de recherche particulièrement disponible, à l'écoute et avec des connaissances infinies qu'il n'hésite jamais à partager. Ce fut un réel plaisir d'avoir pu être encadré par lui.

Merci à toutes les personnes travaillant au LIO, pour leur bonne humeur quotidienne, et le cadre de travail agréable que tous participent à créer. Félix, Martine, Fernando, Julien, Laurence, Caroline, et tous les autres...

Ce projet n'aurait pu aboutir si je n'avais pas eu le soutien de ma famille lors de mon expatriation au Canada, aussi je les remercie du fond du cœur de m'avoir soutenu dans tous mes projets de vie quels qu'ils soient.

Merci à ma Perrine d'être là chaque jour de ma vie.

UTILISATION DE SIGNAUX ÉLECTROMYOGRAPHIQUES POUR LE CONTRÔLE D'UN BRAS ROBOTIQUE POUR LES USAGERS DE FAUTEUILS ROULANTS MOTORISÉS.

Alexandre JUMELINE

RÉSUMÉ

Les blessés médullaires de haut niveau lésionnel n'ont plus l'usage de leurs bras, mais conservent des capacités de mouvements résiduels variant selon la hauteur de leur lésion. La société canadienne Kinova a conçu un bras robotique, JACO, adaptable à un fauteuil roulant motorisé afin d'aider les personnes à mobilité réduite à effectuer leurs actions de la vie quotidienne plus facilement. En effet, celui-ci permet la saisie d'objets ou d'ouvrir des portes de manière intuitive en n'utilisant qu'un joystick. Cependant les personnes n'ayant plus suffisamment d'activité musculaire dans les doigts ne peuvent donc pas profiter de ce progrès. Le but de ce projet est d'adapter l'utilisation du bras JACO aux personnes blessées médullaires en utilisant les signaux électromyographiques (EMG) de 4 muscles résiduels au niveau du cou et de l'épaule.

L'acquisition des signaux EMG est faite avec des électrodes Delsys filtrées et pré-amplifiées en temps-réel à 2 kHz. Les électrodes sont placées sur quatre muscles présents chez les blessés médullaires de haut niveau : les deux trapèzes ainsi que les deux sterno-cléïdo mastoïdiens. L'énergie de Teager-Keiser est utilisée afin de déterminer si une activité musculaire est présente ou non, et chaque mouvement détecté est relié à une action du bras robot. Une calibration est effectuée pour chaque muscle afin de définir des seuils de détection personnalisés. Enfin un système de filtrage prédictif en temps réel a été intégré afin d'augmenter la sensibilité du système.

Ce projet a permis de mettre au point un prototype complet du système jusqu'au contrôle réel du bras robotique grâce à l'EMG. La validité du système a été testée dans une étude sur 10 sujets sains. Cette étude a permis de montrer la cohérence du prototype créé.

Ce projet constitue une avancée supplémentaire dans l'aide technique à la manipulation apportée aux blessés médullaire dans l'accomplissement de leurs tâches de la vie quotidienne.

Mot-clés : Kinova, JACO, contrôleur EMG, temps-réel, blessé médullaire

UTILISATION DE SIGNAUX ÉLECTROMYOGRAPHIQUES POUR LE CONTRÔLE D'UN BRAS ROBOTIQUE POUR USAGERS DE FAUTEUILS ROULANTS MOTORISÉS.

Alexandre JUMELINE

ABSTRACT

High level spinal cord injury individuals do not have the capacity to use their arms, but they still have a control of several muscles depending on the level of injury. The Canadian corporation Kinova has developed a 7 degrees of freedom robotic arm, JACO, meant to be mounted on a motorised wheelchair in order to help spinal cord injury victims in their everyday's life. JACO can be used to grab objects or open doors only using a simple joystick. However, some people don't have enough muscular capacities to use this joystick, and therefore, can not use the robotics arm. The goal of this project is to adapt the use of JACO to spinal cord injury individuals using electromyographic (EMG) signals from four muscles of the neck and shoulders.

EMG signal acquisition is performed using preamplified and filtered Delsys electrodes in real time at a sampling frequency of 2KHz. These electrodes are placed on 4 muscles : trapezius and sternocleidomastoid. These muscles were chosen because of their relation to elementary movements such as shoulder elevation for trapeziuses and head rotation for sternocleidomastoids. The Teager-Keiser energy (TKE) operator is used in order to determine if a muscular onset is performed, and each motion of the person can be linked to an action of the arm. A calibration was done for each muscle in order to set a personalised detection threshold for onset detection. Finally, a real time predictive filter has been added in order to increase the sensitivity of the system.

The result of the project was the complete achievement of a prototype controlling a robotic arm using EMG signals of the user. The validity of the system was then tested in a study including 10 healthy individuals.

This project represents a step towards technical assistance brought to spinal cord injury persons and the improvement of their living conditions.

Keywords: Kinova, JACO, EMG control, real time, spinal cord injury

TABLE DES MATIÈRES

	Page
INTRODUCTION.....	1
CHAPITRE 1 MISE EN CONTEXTE	3
1.1 Les blessés médullaires	3
1.2 Les aides techniques aux actions de la vie quotidienne (AVQ)	6
1.2.1 Les aides techniques en général.....	6
1.2.2 Le bras manipulateur Jaco.....	7
1.3 Le signal électromyographique (EMG)	9
CHAPITRE 2 REVUE DE LITTÉRATURE.....	11
2.1 Classification EMG	11
2.1.1 Fenêtrage du signal	12
2.1.2 Algorithmes de détection d'activité.....	12
2.1.3 Caractéristiques du signal EMG utilisées	14
2.1.4 Classificateurs utilisés dans la détection de mouvements.....	15
2.2 Systèmes embarqués temps réels et EMG	15
2.2.1 Le traitement temps réel.....	16
2.2.2 Méthodes d'acquisition du signal	17
2.2.3 Matériel informatique utilisé dans la littérature	17
2.3 Synthèse	18
CHAPITRE 3 PROBLÉMATIQUE ET OBJECTIFS DU PROJET	21
CHAPITRE 4 MÉTHODOLOGIE	23
4.1 Choix des muscles	23
4.2 Architecture Matérielle	25
4.2.1 Électrodes	25
4.2.2 Microcontrôleur.....	27
4.2.3 Adaptation des tensions	31
4.2.4 Liaison à l'ordinateur.....	33
4.2.5 Liaison à Jaco	36
4.2.6 Schéma matériel global.....	36
4.3 Choix des caractéristiques du signal et des outils algorithmiques et mathématiques	39
4.3.1 Découpage des fenêtres d'échantillons	39
4.3.2 Détection d'activité musculaire	41
4.3.2.1 Énergie de Teager-Keiser.....	41
4.3.2.2 Calibrage du seuil de détection	43
4.3.2.3 Vote à la majorité	44
4.3.3 Filtrage par décision dirigée.....	47
4.3.3.1 Concept.....	48

	4.3.3.2	Calibrage du filtre	49
	4.3.3.3	Description du filtre	51
4.4	Architecture Logicielle		54
	4.4.1	Firmware	54
	4.4.1.1	Fonctionnement et configuration globale du PIC32.....	54
	4.4.1.2	Alogorithme global	56
	4.4.2	Application de monitoring des signaux	58
	4.4.2.1	Choix des librairies et des langages.....	60
	4.4.2.2	Lien avec Jaco et Mono	61
	4.4.2.3	Fonctionnement global et schéma UML du projet	63
4.5	Validation du Système		64
	4.5.1	Protocole de validation	65
	4.5.1.1	Mode de recrutement	65
	4.5.1.2	Principe de fonctionnement	66
	4.5.1.3	Protocole expérimental	67
	4.5.1.4	Déroulement des acquisitions	68
	4.5.1.5	Notes importantes.....	69
	4.5.1.6	Données récoltées.....	69
	4.5.1.7	Interprétation des données.....	69
	4.5.1.8	Comparaison des données	73
CHAPITRE 5 RÉSULTATS.....			75
5.1	Performances globales		75
5.2	Temps de calcul et chronogrammes		77
5.3	Validation du système		78
CHAPITRE 6 DISCUSSION			91
6.1	Performances globales du système		91
6.2	Chronogrammes et temps de calculs		92
6.3	Validation des performances du système		95
CONCLUSION.....			99
CONTRIBUTIONS			101
ANNEXE I CONCEPTION DU MONTAGE POUR LA VALIDATION DU SYSTÈME			103
BIBLIOGRAPHIE			105

LISTE DES TABLEAUX

	Page
Tableau 1.1	Conséquence motrices en fonction de la hauteur de la lésion médullaire. 5
Tableau 1.2	Types de lésions de la moelle épinière et conséquences. 6
Tableau 4.1	Gammes de microcontrôleurs/DSP TMS320 de Texas Instrument, le prix représente le coût du composant seul. 29
Tableau 5.1	Répartition des sujets ayant participé aux expérimentations. M = masculin, F = féminin. 80
Tableau 5.2	Efficacités moyennes par angles obtenues par sujet avec filtrage. 80
Tableau 5.3	Efficacités moyennes par angles obtenues par sujet sans filtrage filtrage. 81
Tableau 5.4	Efficacités moyennes par intervalle d'angles obtenues par sujet avec filtrage. 85
Tableau 5.5	Efficacités moyennes par intervalles angles obtenues par sujet sans filtrage. 86
Tableau 5.6	Efficacités moyennes par angles obtenues par sujet avec filtrage avec application des coefficients d'angle et de vitesse. 87
Tableau 5.7	Efficacités moyennes par angles obtenues par sujet sans filtrage avec application des coefficients d'angle et de vitesse. 88
Tableau 5.8	Analyse de variance sur les angles en comparant les efficacités du système avec filtrage et du système sans filtrage. Ici p représente le pouvoir statistique. 88

LISTE DES FIGURES

	Page
Figure 1.1 Régions de la colonne vertébrale humaine. Tiré de Vsion (2005) sous la licence Creative Commons.	4
Figure 1.2 Bras manipulateur Jaco, tiré de Kinova (2012)	8
Figure 1.3 Signal EMG. Les abscisses correspondent au temps en échantillons, les ordonnées correspondent à la tension mesurée en bits (unité de mesure de l'ADC brute).	10
Figure 4.1 Muscle sterno-cléïdo mastoïdien et muscles du cou. Adapté de Berichard (2008) sous la licence Creative Commons. le muscle indiqué en 1 est le trapèze et le muscle indiqué en 2 est le sterno-cléïdo mastoïdien.....	23
Figure 4.2 Schéma bloc présentant le système global	25
Figure 4.3 Électrode Delsys DE-2.3.....	26
Figure 4.4 Électrode Delsys neutre.	27
Figure 4.5 Microcontrôleur PIC32 utilisé dans le projet. Le microcontrôleur est le gros composant au milieu de la carte. Les composants nécessaire au bon fonctionnement du PIC32 sont également présents sur la carte ainsi qu'une interface de programmation et de debug via usb.....	30
Figure 4.6 Circuit électronique d'adaptation de la tension de sortie des électrodes.	33
Figure 4.7 Circuit de câblage type du composant max3232. Extrait de la documentation officielle du composant produit par Maxim Integrated Products Inc (2007).	34
Figure 4.8 Format de la trame utilisée dans le cadre de la liaison série.	35
Figure 4.9 Schéma bloc présentant le câblage de la partie électronique du projet	37
Figure 4.10 Montage complet du prototype	38
Figure 4.11 TKE calculé sur un signal EMG. La courbe A représente le signal EMG, la courbe B présente le TKE instantané et la courbe C	

	représente le TKE moyen. Les valeurs sont exprimées en unités ADC.	42
Figure 4.12	Définition du seuil de détection d'activité musculaire pour un canal EMG. Les abscisses représentent le temps en échantillons, et les ordonnées représentent la valeur du TKE moyen en unités ADC.	45
Figure 4.13	Détection d'activité musculaire brute, sans vote à la majorité. Les abscisses représentent le temps en échantillons, et les ordonnées montre l'activité musculaire unitaire (1 pour une activité, 0 si pas d'activité).	45
Figure 4.14	Détection d'activité musculaire avec vote à la majorité. La courbe A présente le TKE moyen du signal, la courbe B la détection d'activité sans vote à la majorité, et la courbe C l'activité musculaire avec vote à la majorité. Les valeurs sont exprimées en unités ADC.	46
Figure 4.15	Fonctionnement algorithmique du filtre	50
Figure 4.16	Boucle infinie du programme principal.	58
Figure 4.17	Routine d'interruption du firmware.	59
Figure 4.18	Interfaçage de l'API Kinova à travers Mono.....	62
Figure 4.19	Diagramme UML du programme de monitoring.	63
Figure 4.20	Capture d'écran de l'interface graphique du programme de monitoring des données EMG.....	65
Figure 4.21	Capture d'écran de l'interface graphique du programme de monitoring des données EMG.....	66
Figure 4.22	Capture d'écran de l'interface graphique du programme de monitoring des données EMG.....	67
Figure 4.23	Description de l'angle de rotation de la tête.	68
Figure 4.24	Exemple de données d'angles de rotation de la tête. Ordonnées : angles en degrés, abscisses : temps en secondes.....	70
Figure 4.25	Exemple de données de vitesses angulaires de rotation de la tête.Ordonnées : vitesses en degrés par seconde, abscisses : temps en secondes.	71

Figure 4.26	Exemple de données d'EMG de rotation de la tête. Ordonnées : EMG en unités ADC, abscisses : temps en échantillons.	72
Figure 4.27	Exemple de données de détection d'activité correspondant aux signaux EMG de la figure 4.26. Ordonnées : détection d'activité, abscisses : temps en échantillons.	73
Figure 5.1	Résultats de la détection d'activité musculaire sur un canal EMG	76
Figure 5.2	Contrôle de Jaco par canaux EMG	76
Figure 5.3	Chronogrammes de mesure du temps de calcul. Le chronogramme du haut représente le temps de calcul de détection d'activité musculaire des quatre canaux EMG. Le chronogramme du bas présente le temps de calcul de filtrage d'une fenêtre d'un seul canal EMG.	77
Figure 5.4	Etude du temps de calcul utilisé par le système.	78
Figure 5.5	Vue du dessus du montage réalisé pour la validation du système. h représente la longueur entre la tête du sujet et la cible du centre, l représente la longueur entre la cible du centre et une autre cible et α représente l'angle de rotation de la tête du sujet. Ici $h = 1.14m$ et $l \simeq 20cm$	79
Figure 5.6	Efficacité du système moyenne pour des intervalles d'angles différents, en noir avec filtrage, en blanc sans filtrage. L'efficacité est une valeur normalisée (pas d'unité).	82
Figure 5.7	Vitesses de rotation atteintes par les sujets masculins et féminins. Les sujets hommes sont encadrés. Les valeurs en ordonnées sont exprimées en degrés par secondes	84
Figure 5.8	Angles maximums atteints par les sujets exprimés en degrés.	85
Figure 5.9	Carte de détection du sujet S005 pour le système avec filtrage. En abscisses sont représentées les vitesses angulaires exprimées en degrés par seconde et en ordonnées les angles exprimés en degrés. La zone blanche constitue la zone où une activité musculaire a été détectée et la zone noire une zone où aucune activité musculaire n'a été détectée.	89

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

EMG	Électromyographique
BM	Blessé médullaire
AVQ	Activité de la vie quotidienne
TKE	Énergie de Teager-Keiser
ETS	École de Technologie Supérieure
CRC	Contrôle par redondance cyclique
SOF	Start of frame
EOF	End of frame
ALU	Unité arithmétique et logique

LISTE DES SYMBOLES ET UNITÉS DE MESURE

$Efficacite$	Valeur d'efficacité du système pour un sujet donné
$Coef f_{angle}$	Coefficient de normalisation de l'efficacité en fonction de l'angle maximum atteint par le sujet
$Coef f_{vitesse}$	Coefficient de normalisation de l'efficacité en fonction de la vitesse maximum atteinte par le sujet
$Ef f_{globale}$	Efficacité globale pour un sujet
ms	milliseconde
Hz	Hertz

INTRODUCTION

Avec l'avancée des technologies de l'information, le domaine du génie des technologies de la santé est aujourd'hui en mesure de venir en aide aux personnes présentant un handicap naturel, ou résultant d'une pathologie ou d'un accident, en procurant à celles-ci des dispositifs permettant l'amélioration de leurs conditions et leur qualité de vie. Ces dispositifs sont appelés des aides techniques.

Ce projet s'intéresse principalement aux aides techniques de type robotique développées pour aider les personnes paralysées et en fauteuils roulants motorisés à effectuer leurs activités de la vie quotidienne (AVQ). L'une d'entre elles, le bras robotique Jaco a été développé par la société canadienne Kinova, et est contrôlé par un joystick de manière à ce que son utilisateur puisse facilement effectuer des actions simples, comme l'ouverture d'une porte ou la préhension d'un objet.

Les blessés médullaires (BM) de haut niveau lésionnel n'ont généralement plus l'usage de leurs bras, mais conservent des capacités de mouvements résiduels variant selon la hauteur de leur lésion. Ces capacités résiduelles concernent des mouvements restreints des doigts et des épaules principalement. Suivant le degré d'atteinte et la hauteur de la lésion de la moelle épinière, certaines personnes ne conservent pas suffisamment de liberté de mouvement dans leurs doigts pour utiliser ne serait-ce qu'un simple joystick. Pour ces personnes, les AVQ deviennent des tâches très compliquées.

Le but du projet est donc de faire en sorte de diversifier les interfaces de contrôle du bras Jaco en ajoutant de nouveau moyen de contrôle, pour permettre à des personnes incapables de le contrôler exclusivement au joystick de pouvoir avoir accès à cette avancée technologique majeure. Pour ce faire, l'utilisation du signal électromyographique (EMG) a été choisie par la société Kinova.

Ce projet de maîtrise étant en grande partie de la recherche appliquée, une grande partie présentera la conception du système.

Une première partie de ce mémoire sera consacrée au fait de situer le projet dans l'univers scientifique actuel. Une mise en contexte sera premièrement effectuée dans le chapitre 1, suivie d'une revue de littérature et enfin de la problématique centrale du projet.

La partie principale du mémoire sera alors présentée dans le chapitre 4 : la méthodologie. Ce chapitre expliquera les choix matériels, logiciels et mathématiques effectués tout au long du projet. Toutes les étapes de conceptions seront détaillées ainsi que les interactions entre les différentes parties du système.

Enfin, la dernière partie comprendra l'étape de validation du système conçu afin de déterminer l'efficacité du système et de valider la faisabilité d'un tel projet. Cette validation sera effectuée par l'application d'un protocole de recherche et à travers une étude effectuée sur dix sujets sains afin de collecter des données quantitatives permettant d'effectuer une étude statistique des performances du système.

En résumé cette étude présente la conception d'un système de contrôle d'une aide technique robotisée grâce aux signaux EMG d'une personne BM et à la validation d'une preuve de concept dans le but d'une commercialisation potentielle future par la société Kinova.

CHAPITRE 1

MISE EN CONTEXTE

Les personnes BM sont aujourd'hui en mesure de pouvoir continuer à garder une autonomie partielle dans leur vie quotidienne grâce à la combinaison de plusieurs progrès technologiques. Les fauteuils roulants motorisés sont, par exemple, une part de ce progrès qui contribue à permettre au BM de garder une certaine mobilité, et sont le point d'ancrage d'autres aides techniques dont fait partie le bras robotique Jaco.

Ce chapitre a pour but de présenter les conditions de vie des personnes BM d'aujourd'hui ainsi que les aides techniques fournies par l'industrie qui permettent d'améliorer leurs conditions quotidiennes de vie. Une brève introduction au signal EMG sera également réalisée. Cette mise en contexte est nécessaire afin de discerner dans quelle mesure ce projet de maîtrise participe au processus d'aide aux AVQ des personnes BM.

1.1 Les blessés médullaires

Les blessés médullaires (BM) sont des personnes victimes d'une lésion de la colonne vertébrale et plus précisément de la moelle épinière. Cette lésion peut avoir des conséquences graves sur la vie de la personne. Elle peut entraîner des paralysies ou des pertes sensorielles complètes ou partielles de différentes parties du corps. Cette lésion peut être de plusieurs natures différentes, en fonction de la hauteur de celle-ci, et de son importance.

La hauteur de la lésion détermine quelles parties du corps sera touchée. Cette hauteur est décrite par le nom de la vertèbre correspondante. La figure 1.1 présente les quatre régions principales du rachis qui servent à nommer les vertèbres. Le cou est composé de 7 vertèbres cervicales. Ces vertèbres sont nommées à partir du crâne en descendant de C1 à C7. Par exemple, C1 correspond à la première vertèbre cervicale.

Viennent ensuite les vertèbres thoraciques ou dorsales à qui on fait correspondre les lettres T ou D. Le rachis comprend 12 vertèbres thoraciques numérotées de T1 à T12 de haut en bas du tronc.

On retrouve ensuite au niveau des reins les vertèbres lombaires désignées par la lettre L et qui sont au nombre de 5, numérotées de L1 à L5.

Enfin, les vertèbres pelviennes ou sacrées terminent la colonne vertébrales. Ces vertèbres sont au nombre de 5, et sont soudées à l'âge adulte pour ne former qu'une seule partie osseuse : le sacrum. Elles sont désignées par la lettre S, allant de S1 à S5. Le coccyx, situé à l'extrémité du sacrum termine la colonne vertébrale.

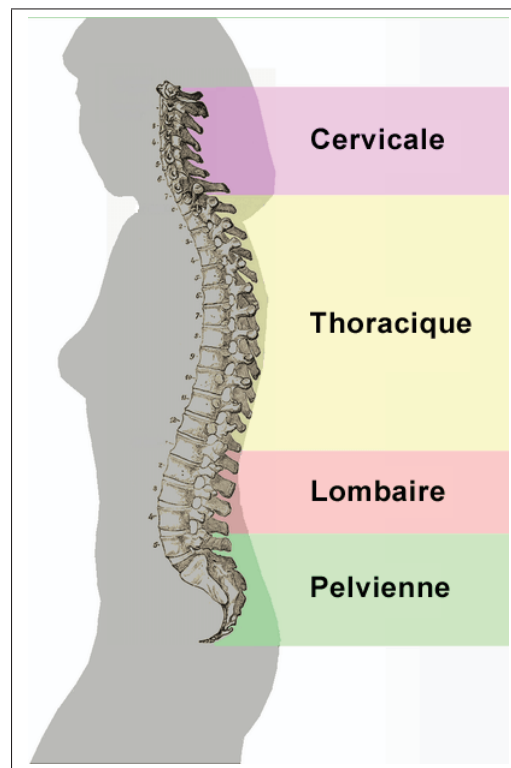


Figure 1.1 Régions de la colonne vertébrale humaine.
Tiré de Vsion (2005) sous la licence Creative Commons.

La hauteur de la lésion s'exprime donc en fonction de la vertèbre correspondante. La hauteur de la lésion influe sur les conséquences de celle-ci. En effet, plusieurs nerfs sortent de la moelle

épineière à chaque vertèbre. Ainsi une coupure de la moelle épinière entraînera une perte de connexion nerveuse pour tous les nerfs situés en dessous de la lésion. Il en résulte que plus la lésion est haute, plus l'incapacité qui en résulte est grande.

Le projet portant principalement sur les personnes BM de haut niveau, les conséquences d'une lésion entre C1 et C8 seront présentées plus en détails. Le tableau 1.1 présente les conséquences motrices et sensorielles d'une lésion médullaire au niveau des vertèbre cervicale en fonction de la vertèbre touchée (Apparelyzed, 2008).

Tableau 1.1 Conséquence motrices en fonction de la hauteur de la lésion médullaire.

C7-C8	Mouvements complets du cou et de la tête Bons mouvements des épaules Système nerveux parasympathique compromis, hyperréflexie autonome possible Paralysie complète du tronc et des jambes Mouvement partiel des doigts, bons mouvement du reste du bras
C6	Aucun mouvement des doigts, pas d'extension du coude ni de flexion du poignet
C5	Aucun mouvement du poignet
C4	Mouvements du cou dépendamment de la force musculaire résiduelle Mouvements limités des épaules Aucun mouvement du coude
C1-C3	Mouvements de la tête limités en fonction du tonus musculaire résiduel Paralysie complète des bras utilisation des muscles du thorax et diaphragme pour respirer impossible

La deuxième caractéristique de la lésion est son type. Celle-ci peut être soit complète, soit incomplète. Une lésion complète correspond à une coupure nette de toute conduction nerveuse située en dessous de la lésion, tandis qu'une lésion incomplète peut, suivant la lésion conserver les conductions nerveuses passant dans la partie de la moelle épinière non lésée.

Le tableau 1.2 présente les quatre grands types de lésions médullaires incomplète et leurs conséquence sur les pertes sensorielles et motrices qui en résultent (Apparelyzed, 2008).

Au regard de ces informations, on peut déduire que chaque personne aura besoin d'une assistance spécialisée dépendant du type de sa lésion et de sa hauteur. Le BM de niveau C1-C3 seront complètement dépendant, et auront même besoin d'une aide à la respiration.

Tableau 1.2 Types de lésions de la moelle épinière et conséquences.

Syndrome cordonal antérieur	Partie frontale de la moelle épinière endommagée Perte de sensation de douleur, toucher et chaleur Possibilité de retrouver un mouvement plus tard
Syndrome cordonal postérieur	Partie dorsale de la moelle épinière endommagée Conservation du tonus musculaire et des sensations Perte de coordination des mouvements des membres
Syndrome cordonal central	Centre de la moelle épinière endommagé Perte de l'usage des bras Conservation possible de certaines fonctions des jambes Contrôle des intestins et de la vessie possible Guérison progressive possible
Syndrome de Brown-Séquard	Domages concentrés sur un côté seulement de la moelle Perte motrice du côté de la lésion, conservation des sensations Perte sensorielle du côté opposé, conservation des mouvements

De même que pour se déplacer et réaliser les actions de la vie quotidienne, ces personnes nécessiteront des aides. Ces aides sont présentées dans la partie suivante.

1.2 Les aides techniques aux actions de la vie quotidienne (AVQ)

Chaque BM a un profil singulier en fonction de la hauteur de sa lésion et de la nature de celle-ci. Ceci implique que chaque personne aura besoin d'une aide particulière pour l'aider à vivre au quotidien. Pour ce faire, l'avancée de la technologie dans le domaine de la santé a permis de concevoir des appareils améliorant grandement les conditions de vie de ces personnes. Ces appareils sont appelés les aides techniques.

1.2.1 Les aides techniques en général

Les aides techniques se présentent sous beaucoup de forme, et peuvent être très simples comme un simple crochet pour attraper des objets, comme très complexes comme par exemple un bras robotique manipulateur. On retrouve une diversité d'objets ayant tous pour but de venir en aide aux personnes dans leurs AVQ, que ce soit, pour s'habiller, se déplacer, se nourrir, se laver, se lever ou communiquer. Les matelas anti-escarres, les orthèses et les bas de contention sont également considérés comme des aides techniques.

Notre étude se concentrera sur les aides techniques en rapport avec le projet, à savoir les fauteuil roulants motorisés (FRM) et le bras manipulateur Jaco.

Les FRM sont une des aides techniques les plus importantes pour les personnes paralysées, car elle leur permet de pouvoir se déplacer de manière autonome ou presque (le transfert dans le fauteuil devant être assisté). Toute fois, suivant le niveau de lésion d'un BM, celui-ci pourra ne pas être en mesure de s'en servir efficacement. En effet, un BM de niveau supérieur à C6 ne pourra pas se servir d'un *joystick*, du fait qu'il ne possède plus de mouvement résiduel dans les doigts.

Pour palier ce problème, les concepteurs d'aides techniques ont du mettre au point des méthodes de contrôle de ces aides adaptées aux personnes n'ayant plus l'usage de leur mains. Ainsi, des manettes de contrôles ont été rajoutées sur les côtés de la tête, activées lorsqu'elles sont poussées par l'action du cou. Un contrôle par le souffle a également été inventé, grâce à un tube placé devant la tête de la personne, dans lequel celle-ci pourra souffler ou aspirer.

Ces faits dénotent une volonté de diversifier les interfaces de contrôle des aides techniques afin qu'un maximum de personnes puisse les utiliser efficacement.

1.2.2 Le bras manipulateur Jaco

Jaco (figure 1.2) est un bras robotique manipulateur conçu par la société canadienne Kinova depuis 2006, et a pour but d'aider les personnes en FRM dans leurs AVQ. Il est conçu pour être monté directement sur le FRM d'une personne et est contrôlable à l'aide d'un *joystick*.

Le bras pèse au total 5.7 Kg, a un rayon d'action de 90 cm et peut porter une charge allant jusqu'à 1.5 Kg lorsque le bras est en demi-extension et 1 Kg lorsqu'il est en extension complète. Il est composé de 7 degrés de liberté (DDL) lui permettant toutes les combinaisons de translations et rotations dans l'espace et possède une main articulée de 3 doigts permettant la saisie de petits objets.



Figure 1.2 Bras manipulateur Jaco, tiré de Kinova (2012)

L'utilisation principale de ce robot réside donc dans la préhension d'objets, l'ouvertures de portes, permettant ainsi à une personne en fauteuil roulant et sans usage de ses bras de se déplacer chez lui, de boire et de manger.

Le *joystick* du bras possède trois axes et plusieurs bouton afin de contrôler tous les mouvements possible. De ce fait, une bonne dextérité est nécessaire pour l'utiliser efficacement, ce qui pose problème aux personnes BM de haut niveau.

Jaco est également livré avec une suite logicielle intégrée permettant de configurer le bras grâce à une liaison USB. Une bibliothèque de programmation est également fournie. Celle-ci permet de contrôler Jaco et de récupérer les informations générées par celui-ci dans un programme externe, conçu par l'utilisateur.

Cette solution permet une grande liberté et ouvre la possibilité à tout un chacun de pouvoir développer une nouvelle méthode de contrôle pour le bras.

1.3 Le signal électromyographique (EMG)

L'électromyographie consiste en l'acquisition de l'activité électrique présente sur un muscle lors de sa contraction. Le signal peut être acquis en surface grâce à une électrode collée directement sur la peau au dessus du muscle, ou directement à l'intérieur du muscle de manière invasive à l'aide d'une aiguille.

Le signal généré par un muscle s'exprime en millivolts (mV) et permet l'analyse de l'activité musculaire d'une personne durant la contraction du muscle.

Le mécanisme de contraction musculaire part du système nerveux central qui envoie un signal nerveux de contraction volontaire à un muscle. Ce signal est transporté jusqu'aux muscles en passant par la moelle épinière puis par un nerf moteur. C'est ce nerf moteur qui transmet le potentiel électrique à chaque fibre musculaire présente dans le muscle.

Chaque fibre musculaire est alors parcourue par un potentiel d'action et la somme de l'action de toutes ces fibres compose la contraction globale du muscle.

Le signal EMG généré est constitué de la somme de tous les potentiels d'action musculaires des fibres recruté pour le mouvement et se présente comme sur la figure 1.3.

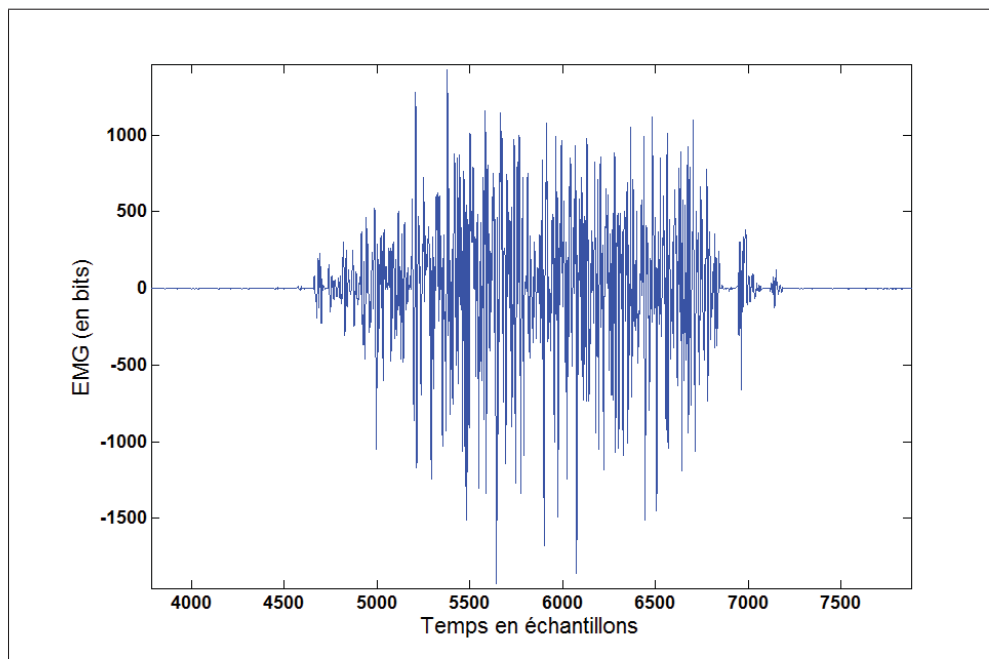


Figure 1.3 Signal EMG. Les abscisses correspondent au temps en échantillons, les ordonnées correspondent à la tension mesurée en bits (unité de mesure de l'ADC brute).

CHAPITRE 2

REVUE DE LITTÉRATURE

La classification de mouvements grâce au signal EMG a été énormément étudiée dans la littérature. Toute fois, une partie seulement de ces études l'a traitée en temps réel. Enfin un type d'application vient encore se démarquer dans ce sous ensemble : les applications embarquées.

Par application embarquée, nous entendons un système fonctionnant de manière autonome sur une carte électronique pouvant être déplacée facilement. Il faut donc enlever à ce sous ensemble toutes les applications réalisées sur ordinateurs.

La première partie de cette revue de littérature s'intéressera à la classification EMG en temps réel, tandis que la deuxième partie sera centrée sur les applications embarquées dont fait état la littérature dans son état actuel.

2.1 Classification EMG

L'analyse des signaux EMG et la détermination du mouvement effectuée à partir de ceux-ci se divise en plusieurs grandes sous-parties incluant chacune un algorithme bien précis. Chacune de ces parties peut être réalisée avec plusieurs algorithmes différents impliquant chacun une précision, et une facilité d'implémentation différente. Des compromis doivent donc être faits entre la précision engendrée par un algorithme et sa complexité d'implémentation et d'exécution qui est souvent inversement proportionnelle.

La première étape de traitement des signaux EMG consiste à déterminer le début et la fin d'une activité musculaire. Celle-ci est appelé détection d'activité, ou détection d'onset, et c'est la première partie qui sera analysée. L'étape suivante est l'extraction des caractéristiques du signal qui pourront permettre ensuite au classificateur de déterminer quel mouvement a été effectué. Enfin, une analyse de différents classificateurs sera effectuée, se basant sur les mêmes critères que les autres algorithmes, à savoir la précision, la facilité d'implémentation et le temps d'exécution.

Un point important est à souligner, et ce point constituera un premier point de cette étude : tous ces algorithmes mis bouts à bouts doivent, une fois le choix d'architecture effectué permettre de réaliser le traitement de quatre canaux EMG en temps réel.

2.1.1 Fenêtrage du signal

Le signal électromyographique est découpé lors de l'acquisition à des fins d'analyses et pour extraire les caractéristiques de celui-ci. La longueur de ces fenêtres est exprimée en temps ou en nombre d'échantillons. Un pas d'incrément est également défini, afin de choisir si les fenêtres se recouvrent ou non.

La littérature étudiée fait état de beaucoup de tailles de fenêtres et de pas d'incrément différents. La plupart ont été fixés de manière heuristique. Cependant, Englehart et Hudgins (2003) évoquent le fait que cette fenêtre ne doit pas être trop grande de manière à ce que le temps de cette fenêtre et de son temps de traitement n'excède pas les 300 ms de latence maximale au-delà desquels celle-ci est ressentie par l'utilisateur. Ainsi, Englehart et Hudgins (2003) ont fixé leur fenêtre à 250 ms avec un pas d'incrément à 125 ms de manière à garder une marge suffisante pour le temps de calcul.

Une fenêtre trop petite est sujette au biais de variance généré par les variations du signal et de ces caractéristiques qui peuvent jouer sur la précision du classificateur utilisé.

Un compromis doit donc être fait sur la longueur des fenêtres utilisées entre la précision du classificateur et son temps de réponse. Le choix du fenêtrage pour le système sera détaillé au chapitre 4.3.1.

2.1.2 Algorithmes de détection d'activité

Une fois une fenêtre du signal acquise, la première étape est la détection d'activité musculaire, ou détection d'"onset".

La détection d'onset est le principe qui consiste à détecter le début d'une activité musculaire sur le signal EMG de manière automatique. Pour ce faire, plusieurs algorithmes semblent bien

fonctionner en combinant à la fois, une bonne précision et un nombre d'opérations restreint. Cette détection d'onset permet d'économiser les ressources du processeur afin de n'extraire les caractéristiques du signal que lorsque c'est utile et surtout de délimiter les zones d'activité musculaire pour la détection de mouvement.

Il existe principalement trois caractéristiques du signal qui peuvent être extraites afin de détecter un onset. Une première utilisée par Chang *et al.* (1996) est la méthode du Zero Crossing Rate (ZCR). Elle calcule le nombre de fois où la courbe du signal EMG repasse par zéro.

Un élément important à préciser est que Chang *et al.* (1996) ont développé cette méthode sur un DSP et semble fournir des résultats satisfaisants. Cette étude présente également la mise en place d'une zone morte sur l'algorithme de zero crossing, qui permet de réduire, voire de supprimer l'influence du bruit sur le ZCR. Cependant cette zone morte diminue la sensibilité de la détection. Le ZCR est une méthode simple à implémenter et très peu coûteuse en ressources processeur pour la raison qu'elle ne nécessite qu'un nombre d'additions égal à $N-1$, N étant le nombre d'échantillons dans la fenêtre analysée.

Peleg *et al.* (2002) optent pour une deuxième méthode consistant à appliquer un seuil directement sur l'enveloppe du signal. Le signal est d'abord passé dans un filtre passe-haut à 30Hz, puis la valeur absolue du signal est réalisée, enfin, le signal est passé dans un filtre passe-bas à 2.5Hz. Cette méthode est une analogie de la méthode Mean Absolute Value (MAV), le filtre passe bas ayant pour but d'effectuer une moyenne des valeurs absolues du signal. Suite à cela, le seuil appliqué sur l'enveloppe est fixé comme étant 10 pourcents plus haut que la valeur de l'enveloppe de l'EMG au repos.

Une troisième méthode simple à implémenter et peu coûteuse en temps de calculs est la méthode de la Teager-Kaiser Energy (TKE) qui correspond à un taux de variation du signal. Elle a été utilisée par Maheu (2011) comme une caractéristique du signal utilisée dans la classification, mais peut aussi être utilisée dans la détection d'onset comme l'ont montré Li *et al.* (2007). Elle ne constitue qu'une soustraction et deux multiplications par échantillon dans la fenêtre d'analyse. Le fonctionnement du TKE est expliqué plus en détail au chapitre 4.3.2.1.

Il existe également des projet n'effectuant pas de détection d'activité musculaire et réalisant une classification des caractéristiques du signal en continu. C'est le cas de Englehart et Hudgins (2003) et Tenore *et al.* (2007). Il est toutefois difficile d'opérer une comparaison entre ces méthodes, car ces articles traitent principalement de classification. Ils présentent leur procédé de détection d'activité sans en donner leurs performances.

2.1.3 Caractéristiques du signal EMG utilisées

Une fois l'activité musculaire du signal EMG détectée, on cherche à déterminer quel mouvement a été effectué. Ce travail est effectué par un classificateur. Ce classificateur doit prendre des éléments de comparaison en compte pour déterminer l'action faite. Ces éléments doivent être significatifs du signal capté, et sont appelés des "features", ou des caractéristiques. Les caractéristiques extraites des signaux EMG sont nombreuses, aussi nous nous concentrerons sur celles les plus utilisées dans la littérature.

Une première caractéristique du signal très utilisée dans la littérature (Chang *et al.*, 1996; Peleg *et al.*, 2002; Maheu, 2011), est l'autoregression du signal EMG.

L'autorégression consiste à trouver les $n+1$ coefficients d'un polynôme d'ordre n qui correspond à une représentation du signal EMG brut. L'ordre de ce polynôme permet, plus il est grand d'augmenter la précision de la représentation du signal d'origine.

Peleg *et al.* (2002) utilisent une auto-régression d'ordre 11. Cet ordre a été déterminé de manière heuristique (Peleg *et al.*, 2002).

Chang *et al.* (1996) ont choisi l'utilisation des coefficients AR mais également les coefficients cepstraux obtenus à partir des AR. Ils préconisent également de fixer l'ordre de l'AR à 4 en justifiant par un compromis entre une bonne représentation du signal et un temps de calcul court. L'obtention des coefficients cepstraux à partir des coefficients AR se fait grâce à une relation mathématique récursive simple dépendant de l'ordre de l'auto-régression réalisée.

Les caractéristiques utilisés dans la détection d'activité musculaire sont également utilisées dans la classification. On retrouve donc dans les caractéristiques le MAV, le TKE et le ZCR.

On retrouve également dans les caractéristiques utilisées les décompositions du signal en on-delettes, ou la longueur d'onde.

2.1.4 Classificateurs utilisés dans la détection de mouvements

Une fois les caractéristiques du signal calculées, celles-ci sont entrées dans un classificateur qui, grâce à un apprentissage, permet de déterminer quel mouvement a été effectué par le sujet. Encore une fois, beaucoup d'algorithmes existent pour réaliser cette opération. Toutefois, beaucoup de ceux-ci sont très consommateurs en temps de calculs, comme les réseaux de neurones (RN).

Toute fois Englehart *et al.* (1999) ont montré que l'importance du choix du classificateur pour la reconnaissance de mouvements à partir de signaux EMG est limitée et que les gains en termes de performances de classificateurs complexes tels que les RN sont négligeables par rapport aux performances d'un classificateur plus simple comme l'analyse linéaire discriminante (LDA).

Ceci étant dit nous nous intéresserons donc plus particulièrement aux deux études principales mettant en œuvre des dispositifs temps réels embarqués.

Chang *et al.* (1996) utilisent dans leur application embarquée un algorithme utilisant les distances des maximums de vraisemblances (MMLD), tandis que Tenore *et al.* (2007) utilisent quant à eux un classificateur de type LDA.

La classification EMG est basée sur la création d'une commande à partir de l'interprétation du résultat de la classification. Cette classification est nécessaire lorsque l'on monitore plusieurs muscles participant au même mouvement.

2.2 Systèmes embarqués temps réels et EMG

Dans un premier temps, il est nécessaire de définir l'appellation "système embarqué" afin de bien comprendre les enjeux que celle-ci représente pour le projet. Un système embarqué est un système électronique et informatique, souvent temps réel spécialisé dans une fonction précise. Ce système, ayant pour contraintes critiques l'espace et la consommation d'énergie. De ces cri-

tiques découlent d'autres contraintes qui sont directement liées aux deux citées précédemment : contrainte de mémoire, de puissance de calcul.

Par conséquent, ces systèmes sont en général constitués d'une carte électronique comprenant un contrôleur principal (microcontrôleur ou DSP) et de matériel électronique spécialisé dans une tâche. Le programme informatique exécuté sur le calculateur est partie intégrante du système.

La littérature comprend très peu d'applications embarquées de traitement temps réel de signaux EMG. Ainsi un premier article de référence a été publié par Chang *et al.* (1996) et rapporte la réalisation du traitement de 4 électrodes EMG afin de contrôler un pointeur informatique dans le but de remplacer une souris. Cette étude fait référence dans notre projet car elle a été développée dans le but de venir en aide aux blessés médullaires de haut niveau.

Deux études de la même époque traitent ensuite également de traitement embarqué des EMG. Martinez *et al.* (1999) utilisent un DSP, sans toutefois préciser lequel, pour traiter un canal EMG tandis que Prasad *et al.* (1996) ont conçu une carte à base de DSP implantée dans un ordinateur pour acquérir et traiter cinq canaux EMG.

Enfin l'étude la plus récente est celle de Tenore *et al.* (2007) qui présentent dans leur article avoir réalisé l'une des premières applications embarquées classifiant le signal EMG.

Cette partie sera découpée en plusieurs grandes problématiques liées au monde de l'embarqué. Dans un premier temps, la composante temps réelle sera présentée, suivie des méthodes d'acquisition du signal EMG et du matériel utilisé dans la littérature.

2.2.1 Le traitement temps réel

Le temps réel au sens strict du terme en informatique, signifie que l'on doit savoir à tout instant combien de temps prendront les différentes actions réalisées par un programme. Ce temps peut s'exprimer de plusieurs manières, les deux principales étant le temps en secondes et le nombre d'instructions élémentaires d'une unité de calcul (microprocesseur, microcontrôleur).

Dans notre cas, et comme fixé dans la littérature du contrôle par EMG, la contrainte principale est le temps de réaction et de calcul maximal à laquelle le système doit satisfaire. La littérature traitant de l'implémentation temps réelle de tels systèmes parle ainsi d'un délai de calcul maximal de 300 millisecondes afin que l'utilisateur humain ne ressente pas la latence du système (Chu *et al.*, 2007; Englehart et Hudgins, 2003).

2.2.2 Méthodes d'acquisition du signal

La plupart des études utilisées pour cette revue utilisent des électrodes de surface pour faire leur reconnaissance de mouvements. Cependant, les manières de filtrer et d'échantillonner sont différentes à travers les études. Cette partie a pour but de les regrouper et de les comparer.

La plupart des études observent une fréquence d'échantillonnage de 1KHz permettant d'obtenir une bande utile théorique (respectant le théorème de Shannon) pour les signaux EMG allant jusqu'à 500Hz (Chang *et al.*, 1996; Chu *et al.*, 2007; Englehart et Hudgins, 2003; Tenore *et al.*, 2007). La plupart des signaux EMG bruts sont d'abord amplifiés avec des gains de l'ordre de 2000, puis passés dans des filtres passe bande entre 10 et 500 Hz.

Peleg *et al.* (2002) ont toutefois opté pour une fréquence d'échantillonnage à 500Hz, avec un filtre anti repliement à 250Hz, alors que Crawford *et al.* (2005) ont opté pour une fréquence d'échantillonnage à 2048 Hz. Lorsqu'elles sont mentionnées, la précision des ADC utilisés varie suivant les études, allant de 16 bits (Englehart et Hudgins, 2003) à 12 bits (Tenore *et al.*, 2007; Fukuda *et al.*, 2003; Prasad *et al.*, 1996). Toutefois, aucun article ne fait état d'une précision minimale dans le cas de l'acquisition de signal EMG.

2.2.3 Matériel informatique utilisé dans la littérature

La plupart des applications trouvées dans la littérature n'ont pas une définition toujours précise de la notion de temps réel. La plupart des articles présentent un système exécuté sur des machines puissantes, sur des systèmes d'exploitation souvent non temps réels.

Englehart et Hudgins (2003) ont réalisé une implémentation temps réelle d'algorithmes de reconnaissance de mouvement à partir de signaux EMG. Cette réalisation a été réalisée sous matlab et son extension de programmation temps réel sur un PC distant.

Si l'on s'intéresse plus particulièrement aux applications embarquées, Chang *et al.* (1996) ont réalisé un montage électronique basé sur un DSP TMS320C31 de chez Texas Instruments. C'est un processeur 32bits comprenant un cœur à virgule flottante tournant à une fréquence de 20MHz, avec quatre niveaux de pipe-line permettant d'exécuter une instruction à chaque cycle d'horloge. Cette architecture leur a permis d'obtenir un temps total de traitement du signal entre 140 et 160 ms ce qui satisfait à la condition évoquée au début de l'étude concernant le respect d'un temps de réponse maximum de 300 ms. Toutefois, cette étude ayant été réalisée en 1996, il existe aujourd'hui quantité de DSP plus rapide et plus efficaces.

Toute fois l'étude de Chang *et al.* (1996), ainsi que celles de Martinez *et al.* (1999) et de Prasad *et al.* (1996) ont aujourd'hui autour d'une quinzaine d'années et le matériel informatique utilisé alors a aujourd'hui beaucoup évolué.

Tenore *et al.* (2007) ont utilisé également un DSP de chez Texas Instruments : un TMS320VC5509A couplé à un ADC dédié de chez National Instruments. Ce DSP est un processor comprenant un cœur à virgule fixe exécutant des actions relativement simples, comme le calcul du ZCR du signal, le calcul de la longueur d'onde du signal et le MAV. Ce DSP leur a permis d'implémenter un classificateur LDA.

2.3 Synthèse

L'étude de la littérature présente différentes applications réalisées dans le contrôle d'appareils externes grâce aux signaux EMG. Toute fois, peu d'articles mentionnent avoir une chaîne complète qui fonctionne, c'est à dire jusqu'à l'action sur l'appareil réel contrôlé par EMG.

Tenore *et al.* (2007) présentent leur application comme une des premières applications temps réelle embarquée permettant de contrôler un bras robotique prothétique. Toutefois, l'article présente l'efficacité du système pour la reconnaissance de mouvement, mais aucune mention

n'est faite du résultat sur le contrôle du bras robotique. Aucune photo du montage complet n'est montrée, et aucune stratégie de communication entre leur système et le robot n'est présentée. Ceci laisse à penser que le projet n'est pas encore arrivé au stade de contrôle du bras réel, et n'est encore qu'au stade de développement.

De la même manière, Chang *et al.* (1996) présente les résultats de reconnaissance de son système sans jamais parler de stratégie de communication avec le périphérique sensé être contrôlé par le système.

De plus, aucun système de la littérature sur le traitement embarqué de l'EMG ne traite la question de la sensibilité. Les types de mouvement effectués sont toujours "une rotation de la tête" ou bien "une élévation des épaules". Il n'est nulle part précisé si le mouvement effectué est de grande ou de faible amplitude. Nous verrons pourtant dans notre système que ce paramètre est d'une importance cruciale, étant donné que l'amplitude du signal EMG, dépendant de l'amplitude du mouvement effectué, influe énormément sur les performances du système.

En résumé la littérature sur la classification de mouvements à partir de signaux EMG présente en grande partie des résultats présentant des pourcentages de reconnaissance.

CHAPITRE 3

PROBLÉMATIQUE ET OBJECTIFS DU PROJET

Aujourd'hui, l'avancée des technologies dans le domaine de la santé a permis de gros progrès médicaux, y compris dans le domaine de la réadaptation et de l'aide à la personne. Cette avancée a permis la conception d'aides techniques de plus en plus évoluées et capable d'améliorer sensiblement les conditions de vie des personnes dépendantes, et plus particulièrement des blessés médullaires.

Cependant, les personnes les plus sévèrement touchées par des blessures de la moelle épinière ne peuvent profiter de certaines de ces avancées, car elles n'ont plus la capacité de se servir de ces aides efficacement.

Connaissant ces paramètres, la société Kinova produisant le bras robotique manipulateur Jaco, a décidé de développer une nouvelle interface de contrôle de Jaco afin que des personnes ne possédant plus l'usage de leurs doigts soient capables d'exploiter le potentiel du robot.

Pour cela, l'utilisation du signal EMG a été préconisée à la suite du projet de maîtrise de Maheu (2011). Ce projet étudie les stratégies de contrôle du bras robotique Jaco en utilisant 8 canaux EMG et un classificateur. Cependant ce projet effectue le traitement de l'EMG en temps différé. Le but étant ensuite de réaliser ce traitement en temps réel. L'enjeu de ce projet est donc en grande partie économique pour la société Kinova, qui sera alors en mesure d'élargir son panel de clients potentiels.

L'étude de la littérature a également permis de constater que les applications de contrôle par EMG en temps réel étaient rares et encore plus lorsqu'on en vient à parler d'application embarquée. Les résultats que ces études présentent montrent des systèmes ayant pour but le contrôle d'un dispositif souvent robotique. Cette finalité de contrôle n'est souvent pas abordée du tout par les articles qui concentrent leur contenu sur les performances des classificateurs.

Ces résultats présentent des systèmes très différents par leur conceptions et leurs applications. Ainsi les résultats ne se basant pas sur les mêmes standards sont difficiles à comparer de manière pertinente.

Ceci permet ainsi d'ajouter un enjeu scientifique au projet en participant à l'avancée scientifique du monde des technologies de la santé.

Les objectifs du projet sont les suivants :

- Réaliser un prototype embarqué capable de reconnaître le mouvement effectué par l'utilisateur en temps réel et en utilisant les signaux EMG de la personne.
- Respecter la contrainte temps-réelle fixée par la littérature de 300ms
- Effectuer un traitement simultané de quatre canaux EMG
- Appliquer une commande sur le bras robotique Jaco en fonction du mouvement détecté

Le but principal n'est pas la réalisation d'un produit fini, mais l'établissement d'une preuve de concept pour cette application. La réalisation doit donc permettre de dire si la réalisation d'un tel produit est faisable ou non, et quels choix sont préconisés pour ce faire.

Le projet est réalisé en partenariat avec le CRSNG et la société Kinova sous la forme d'un projet d'engagement partenarial financé par le CRSNG.

CHAPITRE 4

MÉTHODOLOGIE

4.1 Choix des muscles

Le premier choix important est de choisir quels muscles seront utilisés pour réaliser la commande. Dans un premier temps, le critère principal est que les muscles choisis doivent toujours être fonctionnels chez les blessés médullaires. Ainsi, les muscles choisis par Chang *et al.* (1996), les trapèzes et les sterno-cléïdo mastoïdiens sont considérés comme un choix judicieux dans un premier temps. En effet, ces deux muscles sont innervés en partie par des nerfs passant par la seconde vertèbre cervicale, ainsi, la plupart des blessés médullaires conserveront au moins une activité résiduelle dans ces muscles.

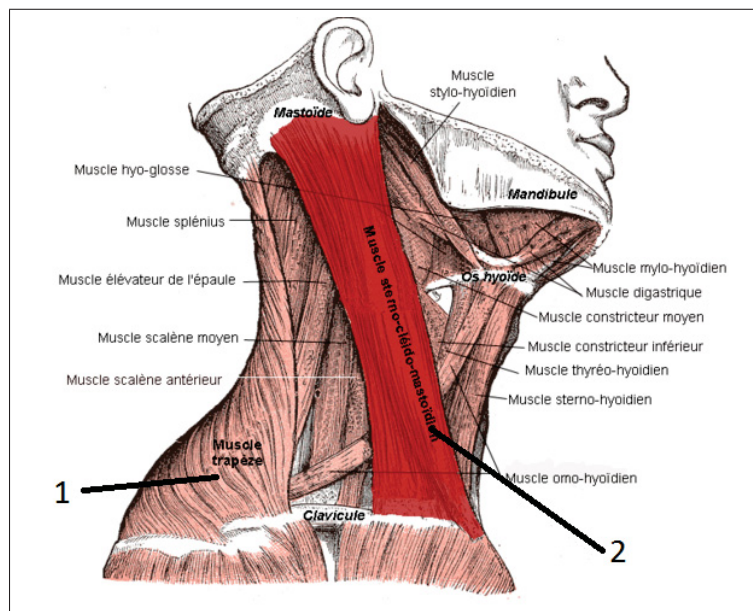


Figure 4.1 Muscle sterno-cléïdo mastoïdien et muscles du cou.
Adapté de Berichard (2008) sous la licence Creative Commons. le muscle indiqué en 1 est le trapèze et le muscle indiqué en 2 est le sterno-cléïdo mastoïdien.

Un autre muscle à considérer serait également le muscle occipito-frontal qui opère l'élévation des sourcils. Toute fois, il est peu probable que les personnes blessées médullaires apprécient

le fait d'avoir des électrodes collées sur le front en permanence. Ce muscle ne représente donc pas une alternative viable pour le projet, de même que pour tous les autres muscles faciaux.

Une autre solution serait d'utiliser les muscles fléchisseurs et extenseurs du poignet. Toutefois, ces muscles sont innervés respectivement par les nerfs médian et radial dont les faisceaux passent par les dernières vertèbres cervicales ainsi que la première vertèbre thoracique. Ainsi, les blessés médullaire de haut niveau n'auront plus suffisamment, voir plus du tout d'activité résiduelle dans ces muscles.

Un point vient également appuyer la sélection des muscles trapèzes et sterno-cléido mastoïdiens et qui va influencer la manière de développer le projet : l'acquisition de signaux EMG sur ces muscles n'implique pas de couplage c'est à dire qu'on ne voit pas l'activité d'un muscle sur l'enregistrement des autres. Cette observation a été réalisée au début de la réalisation du projet, après avoir effectué des tests préliminaires. Ainsi, en faisant des mouvements simples de rotation de la tête et d'élévation des épaules, on ne voit qu'un seul muscle s'activer. Cette considération permet de faire un choix drastique sur la complexité du projet : on choisit de ne monitorer que des mouvements unitaires simples n'activant qu'un seul muscle.

Ainsi, on est capable d'établir un lien direct entre l'activité musculaire d'un muscle particulier et le mouvement réalisé. Lorsqu'une activité est détectée sur un muscle sterno-cléido mastoïdien, c'est qu'une rotation de la tête a été effectuée, et si une activité est détectée sur un trapèze, c'est une élévation de l'épaule qui a été réalisée. Détecter une élévation des deux épaules et une rotation de la tête d'un côté en même temps est également possible.

Ce postulat permet de réduire le temps de calcul nécessaire au traitement des signaux en enlevant complètement l'étape de classification. Ce choix comporte des points négatifs et des points positifs. Un atout majeur est de retirer toute la charge de calcul nécessaire pour le classificateur, et de concentrer ces calculs sur la détection d'activité musculaire ainsi que sur le filtrage numérique du signal. En contrepartie, le nombre de mouvements reconnus par le système est réduit à quatre mouvements unitaires : un pour chaque muscle.

Cependant cette perte en diversité de mouvements pourra être en partie comblée par des stratégies de communication différentes pour ces mouvements. En effet, on pourra effectuer des mouvements prolongés ou bien des mouvements composés de plusieurs activations courtes répétées. Ces stratégies permettront d'étendre la diversité de commande tout en ne reconnaissant que quatre mouvements.

4.2 Architecture Matérielle

Une fois le choix des muscles arrêté, les premiers choix à effectuer seront les choix architecturaux, et dans un premier temps, ceux concernant l'architecture matérielle. Ce chapitre présente les choix matériels effectués pour la conception du système tout en argumentant pourquoi chaque choix a été effectué et dans quel cadre. Le schéma matériel global est présenté par la figure 4.2.

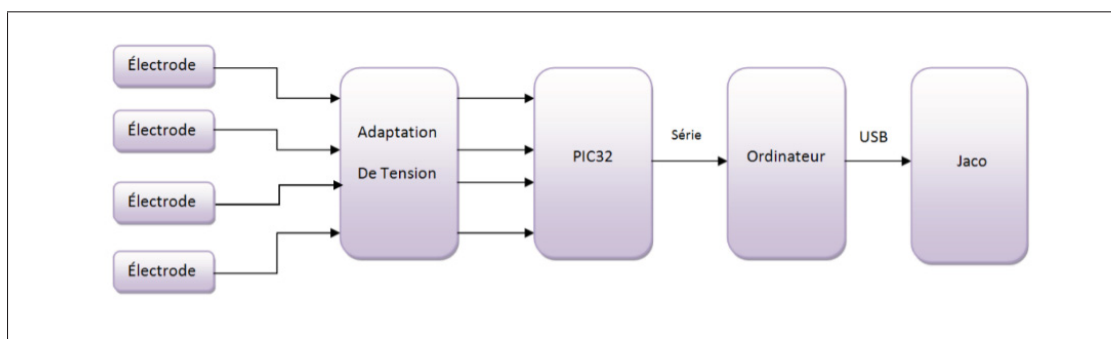


Figure 4.2 Schéma bloc présentant le système global

Les composantes de ce schéma vont être détaillées séparément tout au long de ce chapitre.

4.2.1 Électrodes

Le choix des électrodes est une phase importante du projet car leur choix détermine la charge de travail qui va en résulter. En effet, des électrodes donnent un signal brut très faible (la tension d'un signal EMG étant de l'ordre du millivolt). Ce signal est impossible à exploiter tel quel et nécessite des étapes de filtrage et d'amplification avant de pouvoir être échantillonné. Ainsi

choisir ce type d'électrode ajoute une étape de conception électronique supplémentaire afin que le signal soit exploitable.

Le LIO travaille depuis maintenant plusieurs années avec des électrodes construites par la compagnie Delsys : les électrodes DE-2.3 et se présentent comme sur la figure 4.3.



Figure 4.3 Électrode Delsys DE-2.3.

Ces électrodes, contrairement au modèle moins cher DE-2.1 sont pré-amplifiées avec un gain de $1000V/V \pm 1\%$ et pré-filtrées sur une bande passante de $20 - 450Hz \pm 10\%$. Ceci nous permet de limiter les étapes de filtrage analogique avant de faire entrer le signal de l'électrode dans le microcontrôleur. En effet, le filtrage passe-bande déjà présent permet de garder la bande utile du signal EMG tout en limitant le bruit d'échantillonnage.

Les électrodes sont câblées avec du fil blindé, protégeant des interférences électromagnétiques, le blindage étant relié à la masse du circuit. Pour que ce blindage soit efficace, le connecteur utilisé doit être également de bonne qualité. Pour ce faire, des connecteurs de marque LEMO sont utilisés. Ces connecteurs sont utilisés dans beaucoup de domaines critiques demandant des connecteurs résistants au temps et aux interférences.

Ainsi les connecteurs LEMO femelles correspondant à ceux présents sur les câbles des électrodes ont été acquis afin de relier les électrodes au système.

Plusieurs électrodes neutres, comme présentées par la figure 4.4, ont également été acquises. Ces électrodes sont nécessaires car c'est la référence utilisée pour mesurer la tension acquise

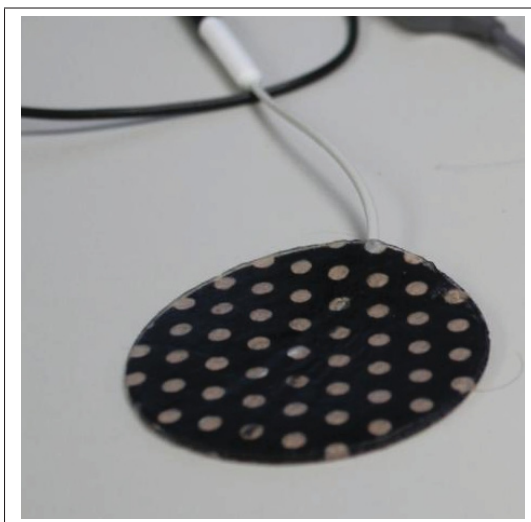


Figure 4.4 Électrode Delsys neutre.

sur les muscles. Elles seront positionnées sur une partie la moins charnue possible, comme par exemple la pointe du coude de l'utilisateur. Elles seront connectées directement à la référence du convertisseur analogique-numérique du microcontrôleur.

4.2.2 Microcontrôleur

Le deuxième choix matériel important concerne celui du microcontrôleur ou du contrôleur de signaux numériques (DSP). En effet, ce choix est déterminant car c'est lui qui opérera tous les calculs nécessaires au bon fonctionnement du système. Son choix est donc à faire de manière réfléchie.

Avant de présenter le microcontrôleur choisi pour le projet, les différents critères de choix seront décrits dans un premier temps.

Un premier critère est le nombre d'opérations élémentaires par seconde effectuées par le calculateur. En effet, le calculateur doit être capable d'effectuer les calculs et les algorithmes choisis pour la stratégie de reconnaissance de mouvement. Ce critère peut se mesurer de deux manières. La première est la fréquence de l'horloge principale du calculateur. Toute fois, même si celle-ci est un bon indicateur, suivant l'architecture du microcontrôleur, un processeur avec une horloge plus rapide qu'un autre peut parfois effectuer moins d'opérations par seconde que

le plus lent. Ceci vient du fait que certains calculateurs ont besoin de deux cycles d'horloge pour effectuer une opération, l'opération devant d'abord être chargée, puis exécutée. Certains microcontrôleurs parallélisent ces actions à l'aide d'un "pipeline" qui pendant qu'il exécute une opération, "précharge" l'opération suivante. Une autre unité permettant d'exprimer la rapidité de calcul d'un processeur est le nombre de millions d'instructions par seconde (MIPS). Cette valeur fait abstraction de la manière dont fonctionne le processeur, et donne directement le nombre de calculs par unité de temps que peut soutenir le composant.

Une autre caractéristique concerne l'unité arithmétique et logique (ALU) du calculateur. Celle-ci peut être dite soit "à virgule fixe", soit "à virgule flottante". Une variable à virgule flottante est une variable permettant de stocker un nombre réel dans lequel on peut déplacer la place de la virgule, tout en ne changeant pas le nombre grâce à un exposant. C'est un système similaire à l'écriture scientifique mais appliqué à l'informatique. Ceci permet pour une taille de variable donnée en octets d'optimiser la précision d'un nombre flottant.

La mémoire vive disponible dans le microcontrôleur est également un critère de sélection. En effet, dans le cas d'un fenêtrage de signal de $300ms$ échantillonné à $2KHz$, les données d'une fenêtre représentent 600 échantillons, qui, stockés dans des entiers de 16 bits représentent une quantité de mémoire de 1200 octets, soit environ $1.2Ko$. Sachant qu'il faut stocker une fenêtre pour la traiter et en acquérir une autre en même temps, le double de cet espace est nécessaire pour le traitement des fenêtres.

$2.4Ko$, mis à l'échelle des processeurs d'aujourd'hui ne représentent que très peu de données. Toute fois, pour un microcontrôleur, cet espace représente une contrainte forte. Contrairement à un microprocesseur qui utilise de la mémoire vive externe, la mémoire vive utilisée par un microcontrôleur est interne, et la taille de la mémoire disponible est donc très réduite.

Un élément supplémentaire de choix du contrôleur est le nombre et la diversité des entrées et sorties qu'il propose. En effet, le microcontrôleur doit avoir au moins quatre entrées analogiques auxquelles se connecteront les électrodes. Le système final devant également être intégrable au sein même de Jaco, et celui-ci utilisant un bus CAN (control area network)

dans ses communications interne, le microcontrôleur doit avoir une interface CAN. D'autres périphériques de communications comme des ports série seront également nécessaires.

Comme dit précédemment le système a pour but d'être intégré au bras Jaco, de ce fait, l'encombrement et la consommation énergétique sont également un critère à prendre en compte.

Enfin le dernier élément à prendre en compte est la dépense nécessaire pour mettre en place la solution choisie. Dans un premier temps, l'existence d'une carte de développement préconçue permet de commencer directement le développement du programme sans avoir à faire de carte électronique. Les logiciels et outils de compilation entrent également en ligne de compte, car ceux-ci sont souvent très coûteux. Le fait que ces outils soient également disponibles sur plusieurs systèmes d'exploitation est également un argument de choix, bien que moins important que tous ceux cités précédemment.

On étudie donc la classe de microcontrôleurs souvent utilisée dans la littérature : la classe TMS320 de Texas Instruments. Ces microcontrôleurs sont divisés en plusieurs gammes présentées dans le tableau 4.1.

Tableau 4.1 Gammes de microcontrôleurs/DSP TMS320 de Texas Instrument, le prix représente le coût du composant seul.

	C5000	C6000	Keystone Multicore	PIC32
Horloge	jusqu'à 300MHz	jusqu'à 1GHz	jusqu'à 16GHz	80MHz
Mémoire vive	320Ko	450Ko	DDR3 externe	128Ko
Prix	\$2 - \$10	\$5 - \$25	\$30 - \$160	\$1.5-\$6.5

Les calculateurs de type TMS320 sont des composants extrêmement gros et coûteux qui permettent d'effectuer des calculs très poussés à grande vitesse. Toute fois, les prix des cartes d'évaluation varient entre 200 et 500 dollars, et les suites de développement logiciel ont un prix allant d'environ 800 à 8000 dollars en fonction des options.

Il en va de même pour les calculateurs de chez Texas Instruments moins performants, les composants ont de bonnes caractéristiques mais nécessitent l'utilisation d'une suite logicielle coûteuse.

Cependant, par expérience, même chez Texas Instruments, certains contrôleurs sont parfois difficile à faire fonctionner. C'est le cas par exemple du microcontrôleur msp430 utilisé sur la carte de développement ez430-rf2500. Une quantité de personnes importante ne parvient pas à faire marcher ce contrôleur correctement sans que Texas n'y amène aucun élément correctif.

Le LIO possédait depuis quelques mois que les outils hardware de programmation et de debug de Microchip. Le microcontrôleur développé par Microchip répondant le mieux aux critères de choix pour le projet est le PIC32 : il possède un ADC 10 bits qui sera suffisant pour le projet du fait de la présence d'un pré-amplificateur intégré dans chaque électrode, et pouvant aller jusqu'à 16 canaux. Il tourne à une cadence de 80 MIPS pour une fréquence d'horloge de 80 MHz et dispose de beaucoup de périphériques de communication dont plusieurs liaisons séries ainsi que qu'un bus CAN.

Ce microcontrôleur possède un cœur à virgule fixe, et intègre également dans son jeu d'instructions des instructions DSP flottantes incluant divers outils dont les FFT qui seront utilisé pour le filtrage du signal. La version utilisée possède également 128Ko de mémoire vive afin de pouvoir stocker une grande quantité de données.

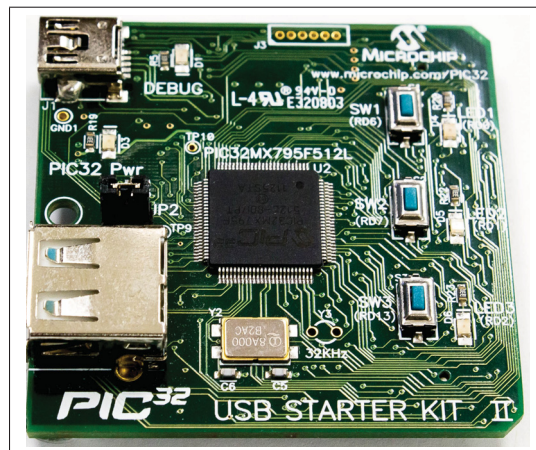


Figure 4.5 Microcontrôleur PIC32 utilisé dans le projet. Le microcontrôleur est le gros composant au milieu de la carte. Les composants nécessaires au bon fonctionnement du PIC32 sont également présents sur la carte ainsi qu'une interface de programmation et de debug via usb.

Un autre avantage du PIC32 est que la suite logicielle permettant son développement peut être utilisée dans un premier temps en version gratuite. Cette version n'inclut pas les options d'optimisation de code pendant la compilation, mais ces options ne sont pas indispensables à la réalisation du projet.

Par expérience, les microcontrôleurs de la société Microchip sont souvent plus fiables que les microcontrôleurs Atmel, qui sont le concurrent principal de Microchip.

Pour des raisons économiques, d'expérience et pratiques, le PIC32, comme présenté sur la figure 4.5 a donc été choisi pour réaliser le système.

Le modèle choisi pour le projet est le PIC32MX795F512L. C'est le modèle le plus rapide et le plus gros de la gamme. C'est également celui possédant le plus d'entrées et sorties différentes, ainsi que la quantité de mémoire vive la plus grande des microcontrôleurs produits par Microchip.

Ce modèle a été choisi également car il est commercialisé par Microchip sur une carte de développement peu coûteuse. De plus cette carte de développement se présente sous la forme d'une petite carte pouvant être montée sur une carte fille permettant de connecter simplement d'autres composants aux entrées/sorties du PIC32.

4.2.3 Adaptation des tensions

Les électrodes étant alimentées en $+5V/ - 5V$, et le PIC32 fonctionnant à $3.3V$, il est nécessaire d'adapter les signaux des électrodes de $+5V/ - 5V$ à $0V/3.3V$ afin de ne pas créer de sur-tension dans le microcontrôleur. Pour ce faire, un montage composé de deux étages d'amplificateurs opérationnels est nécessaire. Les amplificateurs utilisés pour cette réalisation sont des composants intégrés de type LM741CN. Ces composants, bien que peu récents ont l'avantage d'avoir un comportement aujourd'hui bien maîtrisé et connu.

Le premier étage est un montage inverseur ayant un gain de $1/3$. Ce montage est présenté dans la partie gauche de la figure 4.6.

La fonction de transfert de cet étage d'amplification peut être calculée par l'équation 4.1. Ainsi, on fixe respectivement les valeurs des résistances $R1$ et $R2$ à $11Kohm$ et $3.3Kohm$.

$$V_{out} = -V_{in} \times \left(\frac{R2}{R1}\right) \quad (4.1)$$

Cet étage permet théoriquement d'amener la tension entre $-5V$ et $+5V$ à une tension comprise entre $-1.5V$ et $+1.5V$. En pratique, à cause des variations de valeurs des composants, et des caractéristiques de l'amplificateur opérationnel utilisé (imperfections statiques, tension d'offset), la tension est ramenée entre $-1.75V$ et $1.5V$.

Le deuxième étage est un offset réalisé par un montage sommateur inverseur permettant de repasser la plage de tension dans un intervalle positif. Le montage est décrit dans la partie droite de la figure 4.6.

La fonction de transfert de cet étage d'amplification peut être calculée par l'équation 4.2.

$$V_{out} = -R5 \times \left(\frac{V1}{R3} + \frac{V2}{R4}\right) \quad (4.2)$$

Soit, si $R3 = R4 = R5$:

$$V_{out} = -(V1 + V2) \quad (4.3)$$

Ainsi, $V1$ représente la sortie du premier étage d'amplification et $V2$ la tension d'offset à ajouter à $V1$ pour faire passer toute la plage de tension dans les tension négatives (en théorie donc $-1.5V$). Enfin le facteur -1 généré par la partie inverseuse du montage permet de repasser l'intervalle de tension dans les tensions positives uniquement.

Le plus important pour le signal final est de ne pas dépasser $3.3V$ pour les crêtes hautes et de ne pas passer sous les $0V$ pour les crêtes basses. Le signal doit également être centré par rapport à cette plage de tension, et donc être centré aux alentours de $1.75V$.

La figure 4.6 présente le schéma électronique utilisé sur chaque canal EMG avant de le connecter au PIC32.

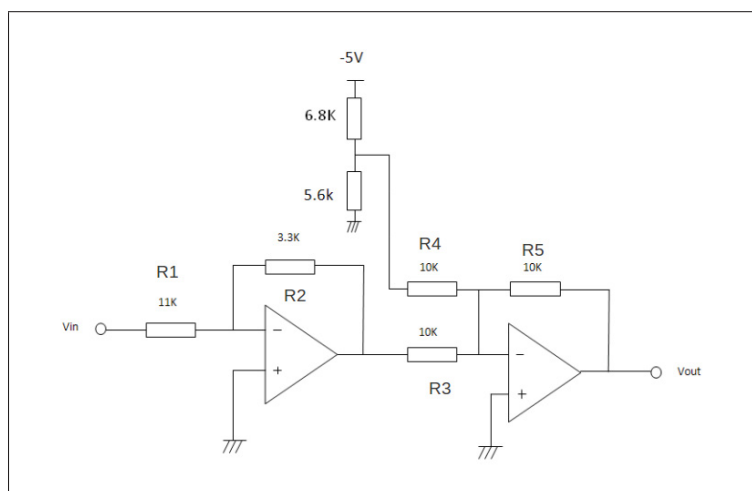


Figure 4.6 Circuit électronique d'adaptation de la tension de sortie des électrodes.

Cependant il est à souligner que la valeur exacte des composants électroniques varie suivant leur tolérance (5% pour les composants utilisés ici). Ainsi le voltage correspondant au 0 peut ainsi varier (1.72V, 1.68V, ...) et un offset sera donc présent sur les valeurs enregistrées par le convertisseur du PIC.

Le schéma 4.6 présente les valeurs de composants pratiques pour la réalisation du circuit. Celles-ci sont définies en tenant compte des imperfections statiques des amplificateurs et des variations des composants.

4.2.4 Liaison à l'ordinateur

Une fois les électrodes connectée au microcontrôleur, celui-ci doit maintenant être connecté à l'ordinateur. Cette partie a pour but de présenter la liaison série réalisée entre la carte électronique du PIC32 et l'ordinateur utilisé pour visualiser les signaux en temps réel.

Dans un premier temps, les niveaux de tension du signal série délivré par le PIC32 ne sont pas compatibles avec ceux utilisés et transmis par l'ordinateur. En effet, le PIC fournit un signal

entre 0V et +3.3V alors que les signaux du port série d'un ordinateur sont compris entre +12V et -12V.

Afin de faire la transition entre ces deux niveaux de tension, un composant électronique intégré est utilisé : un MAX3232, présenté sur la figure 4.7.

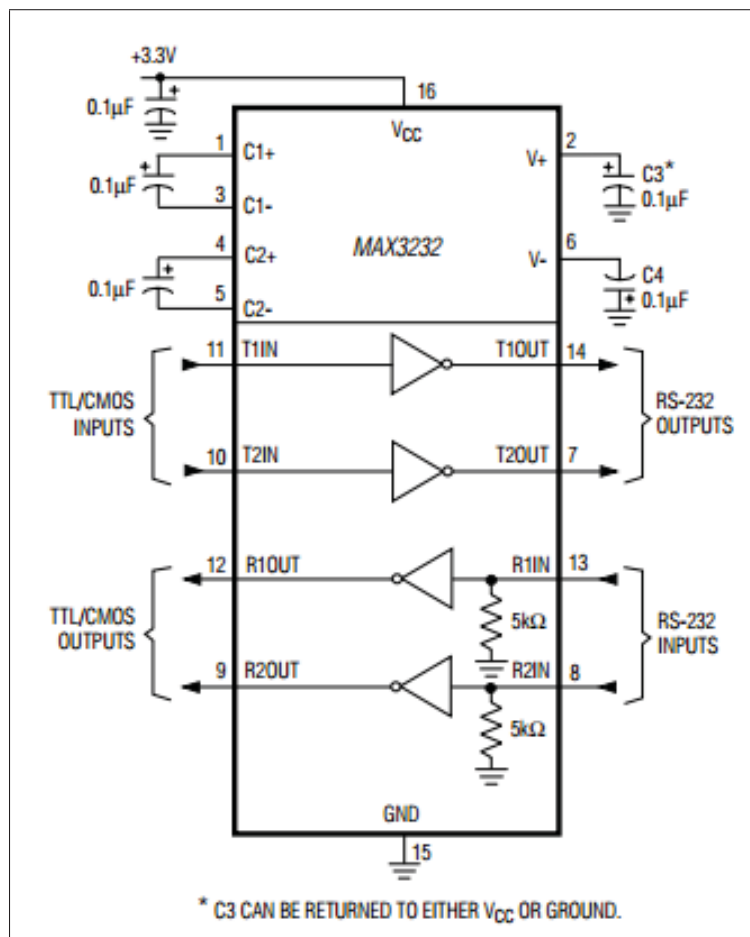


Figure 4.7 Circuit de câblage type du composant max3232.

Extrait de la documentation officielle du composant produit par Maxim Integrated Products Inc (2007).

Une fois la liaison physique réalisée, un protocole de transmission de données a été mis en place. En effet, les données arrivant sur le port série en RS-232 étant traitées octet par octet, il a fallu mettre en place une trame de données afin que celles-ci soit réceptionnées correctement, et permettant de vérifier que toutes les données reçues sont bien intègres. Le début de trame (SOF,

Start of Frame) est constitué de 4 octets fixes, ne changeant jamais de valeur. Ceci constitue une protection contre une mauvaise détection de début de trame. En effet, la même séquence de quatre octets est très improbable de se présenter en tant que données, et de ce fait on aura peu de chance de détecter un mauvais début de trame. Viennent ensuite les données. La longueur en octets des données à transmettre sur la liaison série est facilement configurable et modifiable à la fois dans le programme du PIC32 et dans le programme de l'application de monitoring sur l'ordinateur. Ensuite, les deux octets suivants concernent le contrôle de redondance cyclique (CRC) qui permet de vérifier que la trame reçue n'a pas perdu d'information, ou n'a pas vu un de ses octets corrompu lors du transfert. Le CRC est une valeur sur 2 octets obtenue grâce à un calcul récursif simple dont l'opération mathématique principale est une division modulo 2 dont le reste est le CRC.

Le CRC est calculé en prenant pour entrée uniquement la partie données de la trame. A la réception d'une nouvelle trame, le CRC des données est calculé, puis comparé au CRC transmis dans la trame. Si les deux CRC sont identiques, c'est que la trame est intègre, sinon, la trame est considérée comme non valide, et les données ne sont pas enregistrées, car corrompues.

Enfin, la fin de trame (EOF, End of Frame), est composé de 2 octets fixes qui ne changent pas non plus de valeur afin de détecter quand une fin de trame est réceptionnée. La figure 4.8 présente le schéma global de la trame utilisée dans le cadre de la liaison série.

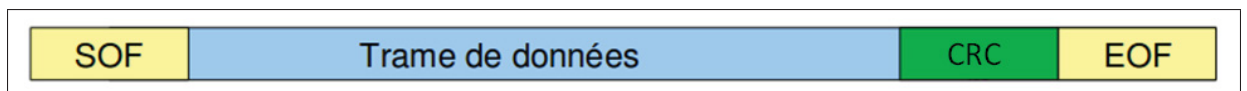


Figure 4.8 Format de la trame utilisée dans le cadre de la liaison série.

Cette trame est utilisée autant dans les données allant du PIC32 vers l'ordinateur que pour les commandes de calibration envoyée au PIC32 par l'ordinateur.

4.2.5 Liaison à Jaco

Le fonctionnement interne de Jaco utilise un bus de type CAN comme mentionné dans le paragraphe 4.2.2. Le bus CAN est un protocole de communication très répandu dans l'industrie et popularisé à l'origine par le secteur automobile.

C'est un bus série opérant en mode différentiel, c'est à dire que deux fils transportent le même signal de manière complémentaire, permettant un contrôle de l'intégrité des données. Ce bus comprend également une couche protocolaire définissant des types de messages à envoyer à une adresse sur le bus.

Dans un premier temps, il était impossible d'implanter directement le prototype du système directement dans le bras Jaco, ou même d'y faire rentrer des câbles pour s'y connecter.

Kinova livre avec le bras robotique des bibliothèques de programmation permettant à un programme fait par un utilisateur de se connecter à Jaco et de lui envoyer des commandes ou des fichiers de configuration.

Le système au stade de prototype étant connecté un ordinateur par une liaison série, il est aisé de se servir de l'ordinateur comme intermédiaire entre le système de détection de mouvement et le bras Jaco lui-même.

De ce fait, le programme de monitoring des données du système doit également s'interfacer avec Jaco de manière à lui transférer des commandes en fonction des mouvements détectés. Cette connexion est effectuée grâce à la liaison USB disponible sur Jaco et utilise les bibliothèques de programmation fournies par Kinova.

4.2.6 Schéma matériel global

La partie 4.2 présente les différentes composantes nécessaires à la réalisation matérielle du projet. Cette partie a pour but de synthétiser tous les choix effectués de manière à donner du recul et un point de vue global du prototype entier.

Détaillons dans un premier temps les entrées et sorties du PIC32.

Contrairement à beaucoup d'autres microcontrôleurs plus petits, le PIC32 ne possède pas d'entrées/sorties qui soit re-localisables. C'est à dire que l'on n'est pas en mesure de choisir quel périphérique prend ses entrées et sorties sur des pattes spécifiques du composants. Ceci implique donc des contraintes de câblage pour relier les électrodes et la liaison série sur celui-ci.

Ainsi on voit sur la figure 4.9 que les quatre électrodes sont reliées via l'adaptation de tension au entrées analogiques AN1, AN2, AN3 et AN4 du PIC32.

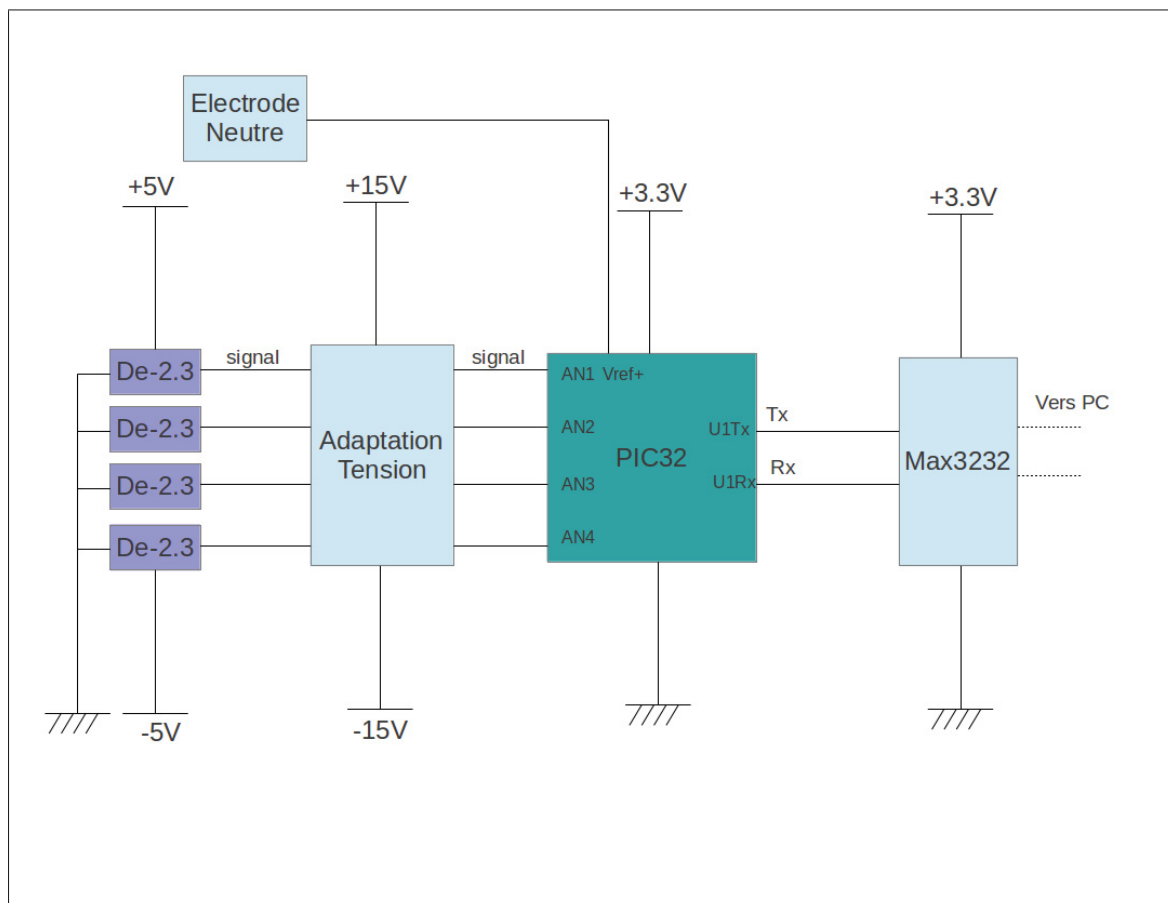


Figure 4.9 Schéma bloc présentant le câblage de la partie électronique du projet

L'électrode neutre est reliée au PIC par l'entrée VRef+ qui sert de référence de mesure pour le convertisseur analogique/numérique du microcontrôleur.

Une UART (Universal asynchronous receiver transmitter) du PIC32 est utilisée pour réaliser la liaison série à l'ordinateur. Les deux lignes de transmission et réception Tx et Rx sont reliées au Max3232 qui adapte les tensions entre le PIC et l'ordinateur par les entrées sorties U1Tx et U1Rx (UART 1 Tx et UART 1 Rx).

Si l'on prend maintenant un point de vue plus haut, nous sommes en mesure d'étudier le système dans son ensemble comme le présente la figure 4.2.

Il est nécessaire de préciser que dans cette architecture, l'ordinateur ne sert que d'intermédiaire entre le microcontrôleur et Jaco. En effet, le microcontrôleur effectue la totalité des calculs. Le mouvement effectué par l'utilisateur est envoyé sur la liaison série à l'ordinateur qui transmet la commande choisie à Jaco via la liaison USB.

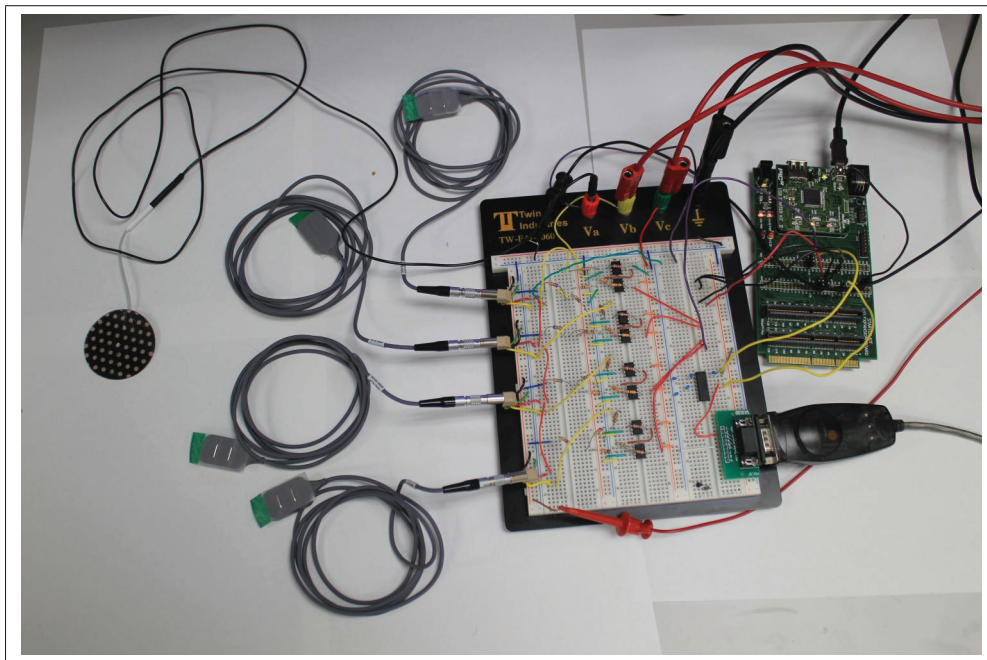


Figure 4.10 Montage complet du prototype

Il est également à noter que cette architecture va induire un délai de traitement impossible à quantifier précisément. En effet, l'ordinateur n'étant pas temps réel au sens strict, le temps entre l'information d'un mouvement effectué et le mouvement de Jaco (ou toute autre action)

commandé par USB est imprévisible précisément et sera beaucoup plus long que si le PIC32 était relié à Jaco directement par un bus CAN.

Ainsi, les temps de traitement mesurés caractériseront uniquement la partie calculatoire effectuée sur le PIC32, et non le temps entre le signal perçu par l'électrode et la commande appliquée sur le bras robotique.

4.3 Choix des caractéristiques du signal et des outils algorithmiques et mathématiques

Une fois les choix matériel effectués et le calculateur choisi, l'étape de réalisation suivante est le choix des calculs à effectuer pour le bon fonctionnement de la détection d'activité musculaire.

Cette partie se décomposera en plusieurs sous partie à des fin de compréhension. La première partie concernera la manière de découper les fenêtre d'échantillons du signal EMG.

La seconde partie concernera le coeur du projet : la détection d'activité musculaire et les outils mathématiques utilisés pour la réaliser. Dans un premier temps, l'utilisation de l'énergie de Teager-Keiser sera détaillée puis l'utilisation d'un vote à la majorité pour le lissage de la détection sera expliquée et enfin la calibration de la détection.

Enfin une dernière partie détaillera le choix du filtre numérique appliqué sur le signal et la manière dont celui-ci est calibré pour chaque sujet et chaque canal EMG.

4.3.1 Découpage des fenêtres d'échantillons

Comme vu dans la revue de littérature, l'exploitation du signal EMG ne se fait pas échantillon par échantillon, le signal est découpé en fenêtre puis chaque fenêtre est utilisée pour calculer des caractéristiques. Ces caractéristiques sont utilisées dans un classificateur pour déterminer le mouvement effectué par l'utilisateur.

La taille de la fenêtre influe principalement sur les performances du classificateur, car une petite fenêtre induit un biais en raison de la variance du signal qui peut entraîner une classification fausse.

Le projet s'étant affranchi du classificateur, s'affranchit également du problème de fenêtrage du signal, dans le sens où son choix ne joue que sur le temps de traitement de celui-ci, en effet, plus une fenêtre est longue plus le temps de calcul pour la traiter est long.

Toute fois, plus une fenêtre est courte, plus le délai avant de recevoir la prochaine fenêtre est également court.

Le choix du découpage en fenêtre du signal est donc ouvert et peut être fait en prenant tous les critères algorithmiques restant, à savoir le fonctionnement interne du PIC32, et des librairies qu'il met à disposition.

En effet, comme la partie 4.3.3 le présentera, le PIC32 met à disposition une librairie flottante incluant une fonction effectuant le calcul d'une transformée de Fourier rapide (Fast Fourier Transform - FFT). Ces fonctions sont nécessaires au filtrage appliqué sur le signal. Les fonctions de cette librairie étant spécialement prévues pour l'architecture du PIC32, celles-ci ont une vitesse d'exécution optimale. La fonction FFT fournie par la librairie PIC32 est disponible sous 3 formes :

- `fft8`
- `fft16`
- `fft32`

Le nombre mentionné dans chaque fonction correspond au nombre d'échantillons dans le signal sur lequel la fft sera calculée. Ainsi `fft8` prend en paramètre une fenêtre de signal de 8 échantillons et renvoie une FFT sur 8 points.

En conclusion, trois tailles de fenêtrage sont ici possibles de manière à faciliter le traitement du signal : 8, 16 et 32 échantillons. D'autres fenêtres seraient possible, mais il faudrait alors les découper en sous-fenêtres afin de pouvoir les filtrer. Le choix effectué est l'utilisation de fenêtres de 16 échantillons. Ce choix a été fait de manière expérimentale par rapport au temps de calcul nécessaire au filtrage d'une fenêtre.

4.3.2 Détection d'activité musculaire

La détection d'activité musculaire est le coeur du projet de contrôle de Jaco par EMG. En effet, le fait de s'être affranchi d'un classificateur implique que cette étape se déroule de manière optimale et toujours cohérente.

Une première partie présentera la caractéristique principale utilisée pour détecter les activités musculaires : l'énergie de Teager-Keiser (TKE).

Puis dans un second temps, le système de vote à la majorité sera détaillé. En effet, fixer un simple seuil sur le TKE du signal entraîne une détection d'activité en changement d'état permanent, parfois à chaque nouvelle fenêtre. Le vote à la majorité permet de lisser cette détection de manière à obtenir une détection d'activité continue sous la forme d'un échelon.

Enfin, le système de calibration pour chaque muscle et chaque sujet sera présenté.

4.3.2.1 Énergie de Teager-Keiser

L'étude de Li *et al.* (2007) a montré qu'une caractéristique du signal EMG est très efficace pour la détection d'activité musculaire automatique : l'énergie de Teager-Keiser (TKE). La raison de ce choix est la simplicité de calcul du TKE, en effet, deux multiplications et une addition sont nécessaires au calcul de cette caractéristique. La contrainte temps réel étant majeure dans le projet, le choix d'une caractéristique simple est donc judicieux. Le TKE s'apparente à un taux de variation simple du signal en comparant un échantillon à son précédent et à son suivant. Le TKE se calcule comme présenté par l'équation 4.4.

$$TKE(k) = x^2(k) - x(k-1) * x(k+1) \quad (4.4)$$

Ainsi, une fois une fenêtre acquise complètement, le TKE instantané (sur chaque échantillon) est calculé.

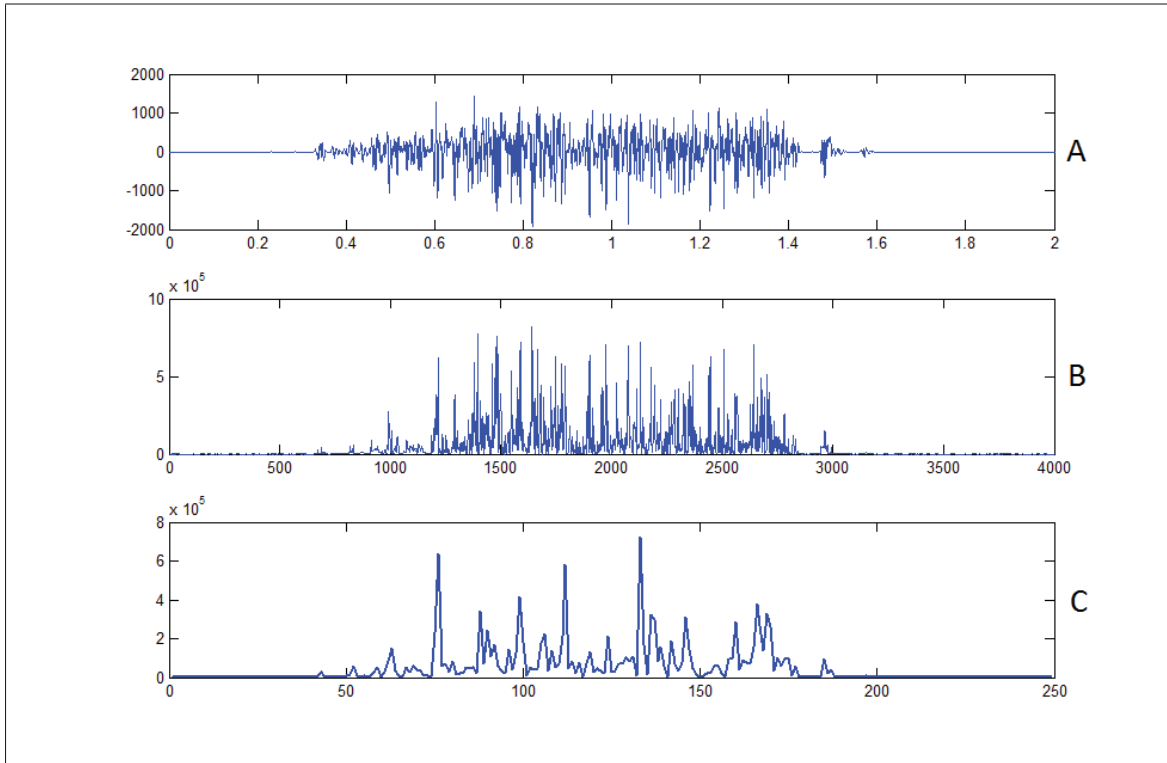


Figure 4.11 TKE calculé sur un signal EMG. La courbe A représente le signal EMG, la courbe B présente le TKE instantané et la courbe C représente le TKE moyen. Les valeurs sont exprimées en unités ADC.

Cependant, ce TKE par échantillon est difficile à utiliser directement, car il est très variable et fixer un seuil efficace pour détecter les activités serait très compliqué. Pour faciliter son exploitation, on effectue la moyenne du TKE sur toute la fenêtre, afin d'en obtenir son enveloppe. De cette manière, un seuil de détection est applicable plus facilement. La figure 4.11 présente l'obtention du TKE instantané à partir de l'EMG, et le passage du TKE instantané au TKE moyen.

Une fois le TKE d'une fenêtre calculé, le but est de savoir si le muscle est activé ou non. Pour ce faire, un seuil doit être défini. Si le TKE de la fenêtre dépasse ce seuil, alors le muscle est considéré comme étant en activité.

4.3.2.2 Calibrage du seuil de détection

Une fois le système de détection d'activité musculaire défini, celui-ci doit ensuite être adapté à chaque utilisateur, et même à chaque muscle de cet utilisateur. En effet, chaque personne développe une force différente, de la même manière que chaque personne ne présente pas le même tonus musculaire lorsqu'il est au repos.

Par exemple une personne stressée sera beaucoup plus contractée au niveau des trapèzes qu'une personne complètement détendue, et ce même sans avoir l'impression d'être contractée.

De manière à calibrer efficacement le seuil de détection d'activité musculaire, il nous faut récolter deux informations pour chaque muscle :

- le tonus musculaire au repos
- le tonus musculaire en contraction isométrique maximum

On acquiert ces données en se basant sur le TKE d'un muscle lorsqu'il est au repos, ainsi que lorsqu'il est en contraction isométrique maximum, c'est à dire que le muscle se contracte au maximum, tout en maintenant l'articulation fixe. Les équations 4.5 et 4.6 présentent les calculs permettant d'obtenir les données.

$$Tonus_{repos} = \frac{\sum_{n=1}^{20} TKE(x_{repos}(n))}{20} \quad (4.5)$$

$$Tonus_{max} = \frac{\sum_{n=1}^{20} TKE_{max}(x_{max}(n))}{20} \quad (4.6)$$

où $x(n)$ représente une fenêtre de 16 échantillons du signal brut. On calcule donc la moyenne des TKE sur 20 fenêtres, ce qui correspond à une durée de 160ms.

Ainsi, l'utilisateur reste au repos, ou bien rentre en contraction maximum, on actionne alors la calibration grâce à l'interface graphique sur l'ordinateur pour enregistrer les tonus de chaque muscle.

Une fois les tonus musculaires recueillis, on sait maintenant qu'à chaque mouvement, l'utilisateur produira un signal EMG dont le TKE moyen sera entre son tonus de repos et son tonus maximum.

Toutefois, il est peu probable que le tonus maximum soit atteint pendant un mouvement. En effet, une contraction isométrique produit un signal beaucoup plus gros qu'une contraction dans un mouvement lent. Le seuil de détection devra donc se trouver entre les deux tonus, le plus proche possible du tonus de repos sans qu'un bruit occasionnellement plus élevé sur le canal ne puisse passer au dessus du seuil.

Le seuil a donc été fixé de manière expérimentale à 25% de l'intervalle entre le tonus de repos et le tonus maximum. La figure 4.12 met en image la manière dont le seuil est fixé pour chaque muscle.

4.3.2.3 Vote à la majorité

Le seuil fixé sur le TKE pour la détection d'activité permet donc de déterminer lorsqu'un muscle est en activité ou non, en observant lorsque ce seuil est dépassé. Cependant, le TKE variant au cours d'un mouvement, il se peut que celui-ci repasse par des moments proches de 0, même pendant un mouvement.

Ceci entraîne alors une détection d'activité qui change alors souvent d'état comme le montre la figure 4.13.

Une conséquence supplémentaire est que si le seuil se veut d'être le plus bas possible afin d'augmenter la sensibilité du système, plus il est bas, plus le système est susceptible de détecter des faux positifs.

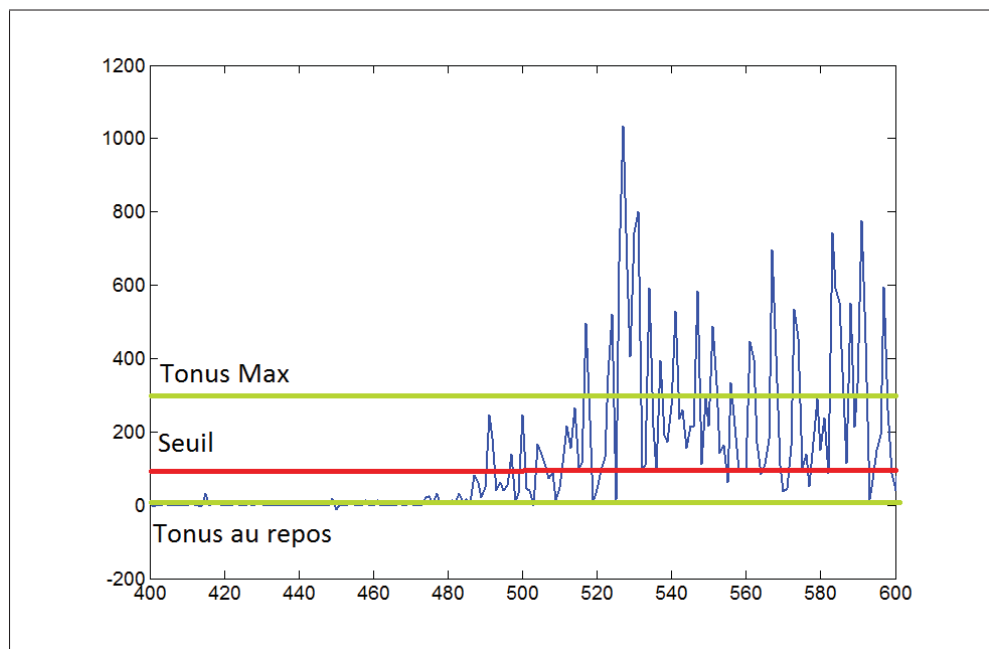


Figure 4.12 Définition du seuil de détection d'activité musculaire pour un canal EMG. Les abscisses représentent le temps en échantillons, et les ordonnées représentent la valeur du TKE moyen en unités ADC.

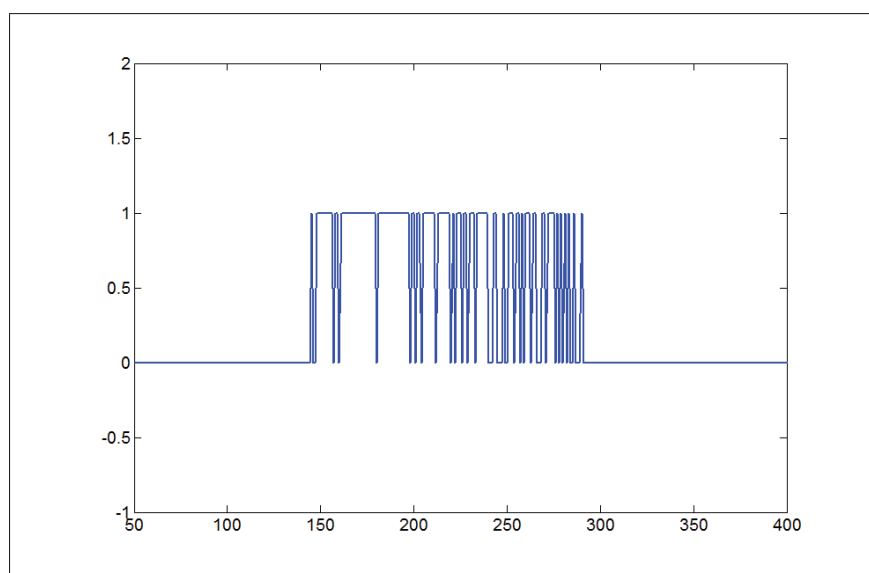


Figure 4.13 Détection d'activité musculaire brute, sans vote à la majorité. Les abscisses représentent le temps en échantillons, et les ordonnées montre l'activité musculaire unitaire (1 pour une activité, 0 si pas d'activité).

Le choix du seuil de détection est donc difficile, voir impossible à choisir idéalement. En effet, un seuil trop haut ne sera plus assez sensible pour détecter des mouvements faibles et en ratera une partie, tandis qu'un seuil trop bas sera sensible mais détectera des faux positifs si le bruit sur le TKE le dépasse.

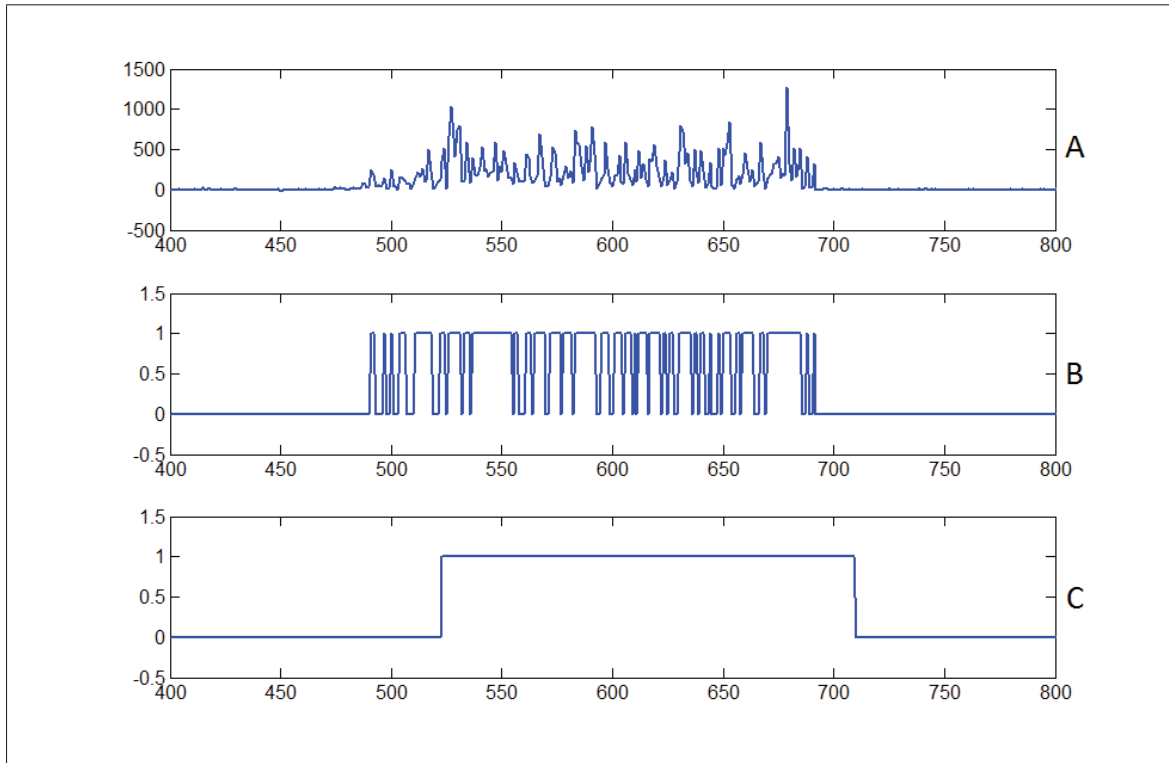


Figure 4.14 Détection d'activité musculaire avec vote à la majorité. La courbe A présente le TKE moyen du signal, la courbe B la détection d'activité sans vote à la majorité, et la courbe C l'activité musculaire avec vote à la majorité. Les valeurs sont exprimées en unités ADC.

Afin de remédier à ces deux problèmes, une méthode inspirée du vote à la majorité de Chang *et al.* (1996) a été mise en place. La méthode mise en place par Chang *et al.* (1996) consiste à comparer la caractéristique d'une fenêtre au seuil. Puis, pour déterminer la détection d'activité, on effectue un vote à la majorité sur les cinq dernières fenêtres. Si la majorité des fenêtres est au dessus du seuil, l'activité musculaire est mise à 1, sinon elle est mise à 0.

Ce système permet d'éliminer la plupart des faux positifs. Toute fois, un inconvénient persiste. En effet, une fenêtre d'activation suffit à faire la différence entre une activité ou un repos. Ainsi, la détection d'activité peut toujours osciller entre 0 et 1, si la comparaison au seuil est différente à chaque nouvelle fenêtre, résultant en une activité en dents de scie comme sur la figure 4.13.

La méthode mise au point pour le système consiste à mettre deux seuils de majorité et de minorité séparés. Ainsi, sur 32 fenêtres d'échantillons, si 16 ou plus sont au dessus du seuil de TKE, l'activité musculaire est mise à 1, puis, pour que celle-ci retombe à 0, il faut que le nombre de fenêtres au dessus du seuil retombe en dessous de 10. Cela permet de ne pas faire des allers-retours entre 15 et 16 fenêtres activées, qui entraînent des discontinuités de détection.

Le résultat de ce système de vote à la majorité est présenté par la figure 4.14.

On observe qu'il n'y a plus de retour à 0 de la détection d'activité, et les faux positifs occasionnels ont également été retirés. Ce vote à la majorité provoque toute fois une légère latence de détection, en effet, il faut attendre que 16 fenêtres sur 32 soient au dessus du seuil de détection pour que l'activité soit détectée. Ce délai temporel de détection entraîne donc une latence du système additionnelle maximum de 16 fenêtre d'échantillons, ce qui correspond à une durée de 128 ms.

4.3.3 Filtrage par décision dirigée

Dans un premier temps, le système a été réalisé sans filtrage numérique. Ainsi le système fonctionnait uniquement en se basant sur la détection d'activité musculaire décrite au chapitre 4.3.2.

Les tests ont été concluants concernant la détection d'activité seule. Cependant le système a présenté une faiblesse : pour être détecté, un mouvement doit être très ample et comporter une composante isométrique forte. Les mouvements simples et lents ne demandant pas beaucoup de force ne sont pas détectés par le système.

En résumé, le système manque de précision, et son utilisation chez des personnes blessées médullaires ayant potentiellement moins de tonus musculaire qu'une personne en bonne santé est alors compromise.

Une solution est de filtrer le signal EMG brut de manière à amplifier les phases d'activité musculaire et diminuer au maximum le bruit de mesure.

Le fait que le bruit de base soit très différent d'un utilisateur rend l'utilisation d'un filtre prédictif préconisée pour effectuer ce travail.

Cette partie présente le filtre prédictif mis en place dans le système dans le but d'augmenter sa sensibilité tout en diminuant le bruit de mesure. Une brève description du filtre sera d'abord effectuée, puis le calibrage de celui-ci sera décrit dans une seconde partie, pour enfin terminer par la description détaillée du filtre et de son algorithme.

4.3.3.1 Concept

Le but du filtrage prédictif est de pouvoir prédire quelles parties du signal le filtre doit amplifier, et quelles parties de ce même signal il doit réduire.

Dans notre cas, la partie du signal à écraser est le bruit analogique mesuré par les ADC du PIC32 ajouté au signal de tonus musculaire au repos, tandis que la partie du signal à amplifier est le signal EMG généré par le muscle sur lequel est collée l'électrode.

Le filtre ayant deux éléments à reconnaître (le bruit et le signal), il faut en caractériser un des deux, et définir que lorsque l'on ne le détecte pas, c'est que l'on est en présence de l'autre. Ainsi, on choisit de caractériser le bruit. Lorsque le système détecte la présence du bruit, alors le signal est écrasé, lorsqu'il ne le détecte pas, c'est qu'un EMG est présent dans le signal, il est alors amplifié.

Pour caractériser le bruit, on utilise les caractéristiques fréquentielles du signal. Le but est donc d'isoler le spectre d'amplitude du bruit afin d'en faire un modèle de comparaison. On enregistre

le signal d'un muscle au repos et on calcule le spectre d'amplitude moyen de ce signal. C'est la bande de base.

Puis à chaque nouvelle fenêtre, on compare la variance du spectre d'amplitude de cette fenêtre à celui de la bande de base pour calculer une probabilité de présence du bruit. Cette probabilité permet de fixer un gain qui amplifiera ou écrasera le spectre d'amplitude et donc par extension, le signal.

Il est également à noter que l'on ne veut pas créer de déphasage du signal. La phase du signal sera donc isolée de la FFT pour être réutilisée sans modification pour reconstruire le signal après que le spectre d'amplitude a été transformé. Le schéma 4.15 présente le fonctionnement algorithmique global du filtre. Cette méthode découle de celle mise au point par Plapous *et al.* (2006) dans le domaine du traitement de signal acoustique.

En résumé, le filtrage opère de la manière suivante : un modèle de bruit est enregistré, puis à chaque fenêtre, le filtre compare la fenêtre au bruit de référence. Si la fenêtre est considérée comme étant principalement du bruit, elle est écrasée, et si elle est considérée comme étant principalement composée d'un signal, alors elle est amplifiée.

4.3.3.2 Calibrage du filtre

Une fois le concept du filtre posé, le mode de fonctionnement précis du filtre va maintenant être détaillé. La première étape de la réalisation du filtre est son calibrage, ou l'enregistrement de la bande de base.

Le calibrage du filtre s'effectue sur une partie de signal où l'on est sûr qu'aucune activité musculaire ne sera présente. On demande donc à l'utilisateur de ne pas bouger pendant l'acquisition de la bande de base. Un tonus musculaire au repos est enregistré.

Le signal est alors acquis par fenêtre. Lorsqu'une fenêtre est acquise, on la normalise, c'est à dire qu'on retire sa moyenne à chaque échantillon (eq :normalCalib où $NFFT = 16$), puis on

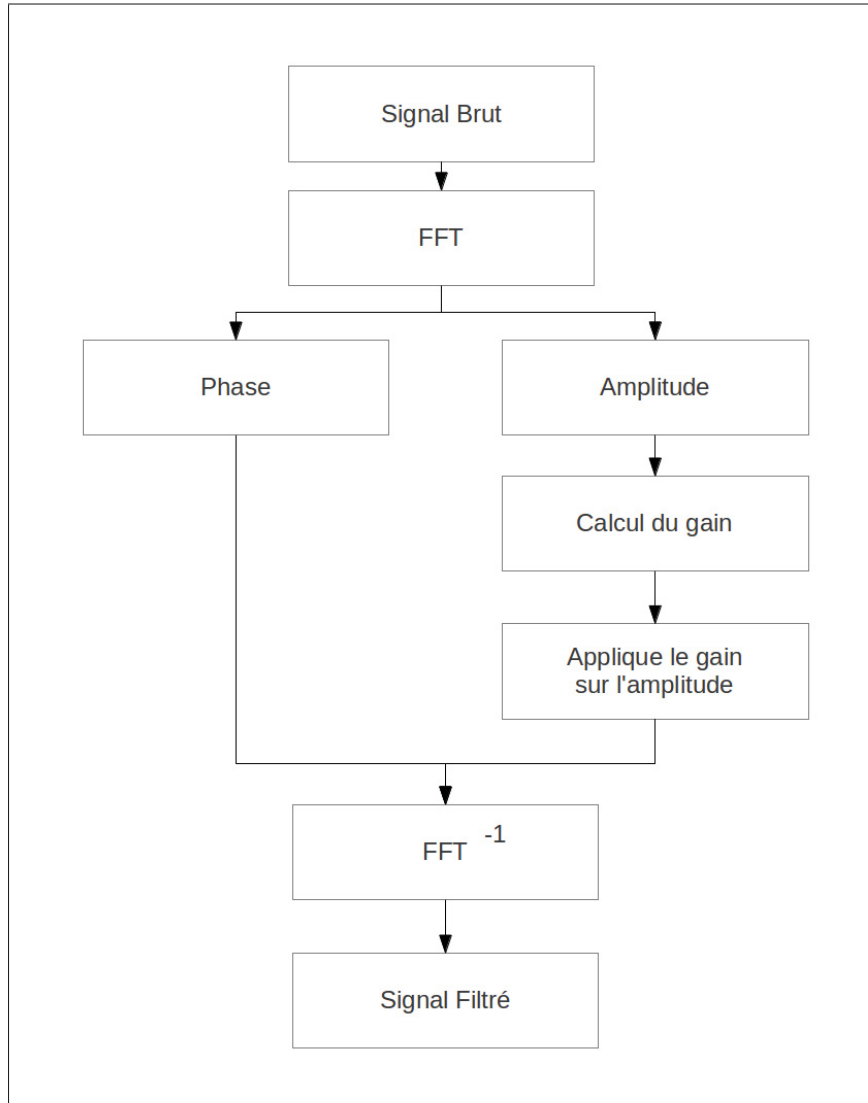


Figure 4.15 Fonctionnement algorithmique du filtre

calcule sa FFT (équation 4.8). Ici k représente le nombre d'échantillons dans la fenêtre étudiée.

$$x_{k=1}^{k=NFFT}(k) = x_{brut}(k) - \frac{\sum_{n=1}^{NFFT} x(n)}{NFFT} \quad (4.7)$$

$$fft = FFT_{16}(x(k)) \quad (4.8)$$

La FFT calculée par cette opération est donc un tableau de 16 nombres complexes représentant la transformée de Fourier de la fenêtre. En effet, lorsqu'un signal est réel, sa FFT est toujours complexe.

Une fois cette FFT calculée, on sauvegarde la phase du signal pour ne garder que le spectre d'amplitude M de la fenêtre (équation 4.9). Ici k devient la fréquence du spectre étudié.

$$M_{k=1}^{k=NF\!FT}(k) = \|fft(k)\|^2 \quad (4.9)$$

Ici, k représente le nombre d'échantillons d'une fenêtre de signal. On garde ensuite en mémoire ce spectre d'amplitude, puis on répète l'opération sur 20 fenêtres. Une fois les 20 spectres d'amplitude acquis, on calcule le spectre d'amplitude moyen de ces 20 fenêtres (équation 4.10).

$$BandeBase = \frac{\sum_{n=1}^{20} amplitude(n)}{20} \quad (4.10)$$

Ici, n représente le nombre de FFT gardées en mémoire pour le calcul de la bande de base. Ce spectre d'amplitude moyen constitue la bande de base qui sera comparée à chaque fenêtre du signal pour déterminer à chaque fenêtre la probabilité de présence/absence du bruit dans le signal.

4.3.3.3 Description du filtre

Une fois la bande de base du filtre calibrée, le filtre peut être mis en fonctionnement à chaque fenêtre de signal acquise. Cette partie détaille le fonctionnement algorithmique du filtre.

Une nouvelle fenêtre est acquise. Dans un premier temps, elle est sujette au même traitement que pour la calibration : elle est normalisée (équation 4.7) puis sa FFT est calculée (équation 4.8).

Une fois la FFT obtenue, on sépare le spectre d'amplitude (équation 4.9) de la phase ϕ qui cette fois-ci est aussi calculée (équation 4.11). En effet, celle-ci sera nécessaire pour reconstruire le

signal une fois que l'amplitude aura été modifiée.

$$\phi(k) = \text{atan2}(\Im(f \text{ft}(k)), \Re(f \text{ft}(k))) \quad (4.11)$$

On calcule ensuite la probabilité a posteriori postsnr de présence du filtre en faisant le rapport entre le spectre d'amplitude de la fenêtre et le spectre d'amplitude moyen (équation 4.12).

$$\text{postsnr}(k) = \frac{M(k)}{M_{\text{Bandedebase}}} - 1 \quad (4.12)$$

Cette probabilité est ensuite limitée pour ne pas créer de distorsion provoquée par une valeur trop petite (équation 4.13). La valeur 0.01 a été trouvée expérimentalement.

$$\text{postsnr}(k) = \max(\text{postsnr}(k), 0.01) \quad (4.13)$$

On compare ensuite la probabilité obtenue avec celle obtenue avec le spectre d'amplitude de la fenêtre précédente. Cette méthode est appelée méthode de la "décision dirigée" (équation 4.14). Une nouvelle amplitude est alors calculée en appliquant cette probabilité sur l'amplitude d'origine (équation 4.15).

$$\text{eta}(k) = \alpha * \frac{M(k-1)}{M_{\text{Bandedebase}}} + (1 - \alpha) * \text{postsnr}(k) \quad (4.14)$$

eta est l'estimation de présence d'un signal en comparaison de l'échantillon précédent. α est un coefficient fixé dans l'algorithme permettant de définir quand on est en présence de bruit ou d'un signal. Il est en général fixé à une valeur proche de 1, comme 0.999 dans le cas du système.

$$M_{\text{new}}(k) = \frac{\text{eta}}{\text{eta} + 1} * M(k) \quad (4.15)$$

M_{new} correspond à la nouvelle amplitude sur laquelle on a déterminé la probabilité de présence de bruit. Ainsi, un gain différent est appliqué sur chaque fréquence du spectre grâce à un gain de Wiener. On calcule ensuite le gain à appliquer sur le spectre d'amplitude en comparant

l'amplitude sur laquelle on a appliqué les probabilités a priori avec le spectre d'amplitude moyen (équation 4.16, 4.17, 4.18 et 4.19).

$$tsnr(k) = \frac{M_{new}(k)}{M_{Bandedbase}} \quad (4.16)$$

Ici $tsnr$ représente le rapport signal/bruit de la nouvelle amplitude comparée la bande de base. Elle est utilisée pour calculer $Gtsnr$ qui correspond au gain à appliquer sur l'amplitude pour trouver l'amplitude finale.

$$Gtsnr(k) = \frac{tsnr(k)}{tsnr(k) + 1} \quad (4.17)$$

$$Gtsnr(k) = \max(Gtsnr(k), 0.01) \quad (4.18)$$

$$M_{final}(k) = Gtsnr * M(k) \quad (4.19)$$

Une fois la nouvelle amplitude calculée, il ne reste plus qu'à reconstruire la fft du signal, en réunissant l'amplitude et la phase (équation 4.20 et 4.21). Ici k correspond toujours à une fréquence sur le spectre.

$$\Re(fft(k)) = M_{final}(k) * \cos(\phi(k)) \quad (4.20)$$

$$\Im(fft(k)) = -M_{final}(k) * \sin(\phi(k)) \quad (4.21)$$

Une fois fait, on effectue la transformée de fourrier inverse pour retrouver le signal EMG filtré. Comme le PIC32 ne possède pas de fonction FFT inverse, on effectue alors la FFT du conjugué du complexe obtenu après filtrage, ce qui fournit le même résultat que la FFT^{-1} .

4.4 Architecture Logicielle

Les parties précédentes ont présenté l'architecture matérielle ainsi que les outils mathématiques utilisés dans le projet. La dernière grande partie technique à présenter restante est donc l'architecture logicielle.

En effet, tout en prenant en compte les choix effectués dans la partie précédente, la manière de les réaliser et de les connecter ensemble doit être présentée.

Ainsi, ce chapitre présentera d'abord le fonctionnement du firmware, c'est à dire le programme exécuté sur le PIC32 puis présentera le fonctionnement de l'application de monitoring des signaux, qui permet aussi la connexion du système au bras Jaco et qui est exécutée sur un ordinateur.

4.4.1 Firmware

Le firmware est le nom donné à un programme tournant sur un microcontrôleur, ou sur une carte électronique. Ce firmware est un programme développé dans un langage dépendant du microcontrôleur choisi pour le projet.

Le PIC32 est programmable en langage C, ainsi qu'en assembleur bien qu'il soit peu utilisé, excepté pour les parties de code nécessitant une grande optimisation.

Cette partie décrira le fonctionnement du PIC32 ainsi que l'algorithme global du firmware.

4.4.1.1 Fonctionnement et configuration globale du PIC32

Pour présenter le firmware, il est important de présenter la manière de fonctionner du PIC32. La différence principale entre un microcontrôleur et un ordinateur moderne est qu'il est impossible d'exécuter plusieurs programmes en parallèle sur un microcontrôleur. En effet, c'est le système d'exploitation qui permet dans un ordinateur de paralléliser les opérations effectuées.

Ainsi, un microcontrôleur n'effectue qu'une seule série d'opérations : celle de son programme. Un programme est une suite d'instructions élémentaires compilées dans un fichier binaire.

Cette suite d'instruction est inscrite dans la mémoire de programme du microcontrôleur et chaque instruction possède une adresse dans cette mémoire. Le compteur de programme (PC - program counter) et un nombre contenant l'adresse mémoire de l'instruction en cours. Une fois que cette instruction est finie, le PC est incrémenté pour passer à l'instruction suivante et jusqu'à ce que le programme se termine.

Cependant le microcontrôleur possède des périphériques dédiés capables d'effectuer les opérations pour lesquelles ils sont prévus sans influencer le déroulement du programme principal. Par exemple, le convertisseur analogique-numérique (ADC) enregistre les échantillons du signal sans que cela ait à figurer dans le programme.

Ces périphériques possèdent des interruptions, qui permettent de signifier au programme lorsque des données sont accessibles. Ces interruptions sont déclenchées quand une condition est remplie et sont une fonction spécifique qui n'est exécutée que lorsque la condition est remplie. Par exemple, il est possible de configurer l'ADC du PIC32 pour qu'il déclenche une interruption à chaque fois qu'il a fini d'acquérir 16 samples.

Lorsqu'une interruption est déclenchée par un périphérique, le déroulement du programme est arrêté. L'adresse de l'instruction où il s'est arrêté est mise en mémoire, puis le code de la fonction d'interruption est exécuté. Une fois que cette fonction est terminée, le programme reprend son cours normal à l'adresse qui avait préalablement été mise en mémoire.

Ce système d'interruption permet d'effectuer automatiquement des opérations comme l'acquisition d'un signal, ou l'envoi et la réception de messages sur le port série sans influencer le programme principal, et donc gagner du temps de calcul.

Cependant, les interruptions bloquent le cours normal du programme principal. Les interruptions doivent donc être les plus courtes possibles pour ne pas perdre trop de temps dedans.

Il est donc important de bien choisir quels calculs sont effectués dans les interruptions, et quels calculs sont faits dans le programme principal.

Une interruption peut également être interrompue par une autre interruption. Un système de priorité détermine l'ordre dans lequel ces interruptions ont lieu. Une interruption ne pourra jamais interrompre une autre interruption ayant une priorité plus haute. Ainsi l'interruption en cours se terminera avant que l'autre puisse s'effectuer.

Ainsi, les deux périphériques utilisés dans le système fonctionneront à base d'interruptions, à savoir l'ADC permettant l'acquisition des quatre canaux, et l'UART permettant la communication série.

L'ADC du PIC32 permet de scanner plusieurs entrées les unes après les autres. Ainsi il faut donc multiplier la fréquence d'échantillonnage par le nombre de canaux afin de pouvoir acquérir tous les canaux à la bonne fréquence. On fixe donc la fréquence d'échantillonnage à 8KHz pour que chaque canal soit échantillonné à 2KHz.

L'interruption de l'ADC est configurée pour être déclenchée après avoir acquis le plus d'échantillons possible, soit 8 échantillons. Ainsi, tous les 8 échantillons, correspondant à 2 échantillons par canal, l'interruption de l'ADC est déclenchée.

L'UART du PIC32 quant à elle possède deux interruptions, une pour l'envoi de données, et l'autre pour la réception. Ces interruptions sont déclenchées respectivement à l'envoi et à la réception d'un octet complet. Ainsi, à chaque octet reçu, celui-ci est mis en mémoire dans l'interruption, et à chaque octet envoyé, on met l'octet suivant dans le "buffer" d'envoi.

4.4.1.2 Algorithme global

Une fois le fonctionnement global du PIC32 et de ses interruptions décrit, l'algorithme du programme principal ainsi que ceux des interruptions peuvent maintenant être présentés en détail.

Nous nous intéresserons dans un premier temps au fonctionnement du programme principal, puis les algorithmes des deux interruptions principales seront détaillés.

Le programme principal du firmware commence par initialiser tous les périphériques du PIC32 qui seront utilisés, ainsi que l'horloge principale et les configurations des interruptions. C'est à ce moment que l'ADC et le port série sont configurés.

Une fois les initialisations effectuées, le programme entre dans une boucle infinie qui constitue la colonne vertébrale du programme. Les actions réalisées par cette boucle sont présentées par la figure 4.16.

La boucle commence ainsi par vérifier si une commande a été reçue sur le port série. Les commandes reçues sur le port série sont les commandes de calibration du système. Ainsi chaque commande est constituée du type de calibration à effectuer, c'est à dire une calibration du tonus maximum, une calibration du tonus au repos ou bien la calibration de la bande de base du filtre, et le numéro du canal sur lequel effectuer cette calibration.

On vérifie ensuite si une fenêtre de signal a été acquise en entier. Si c'est le cas, cette fenêtre est filtrée, puis passée dans la fonction de détection d'activités. Enfin les données calculées sont envoyées sur le port série pour être monitorées.

Il est à noter que le filtrage n'est effectué que si la calibration de la bande de base a bien eu lieu précédemment, mais ce test n'est pas affiché sur le diagramme à des fins de clarté.

Cette partie montre comment le signal est traité, présentons maintenant comment celui-ci est acquis. La figure 4.17 présente la routine d'interruption de l'ADC du PIC32. Celle-ci est déclenchée à chaque fois que 8 échantillons ont été acquis en tout, soit 2 échantillons par canal.

Il est à noter que deux tableaux de données sont utilisés pour chaque canal. L'un des tableaux est rempli au fur et à mesure par l'ADC pendant que l'autre est traité, c'est à dire filtré puis passé à la détection d'activité. Une fois qu'un tableau est rempli par l'ADC, celui-ci est passé en phase de traitement, et l'autre est rempli à nouveau par l'ADC. Il est donc nécessaire que le tableau à traiter le soit avant qu'une nouvelle fenêtre ne soit acquise, sinon des données seront perdues.

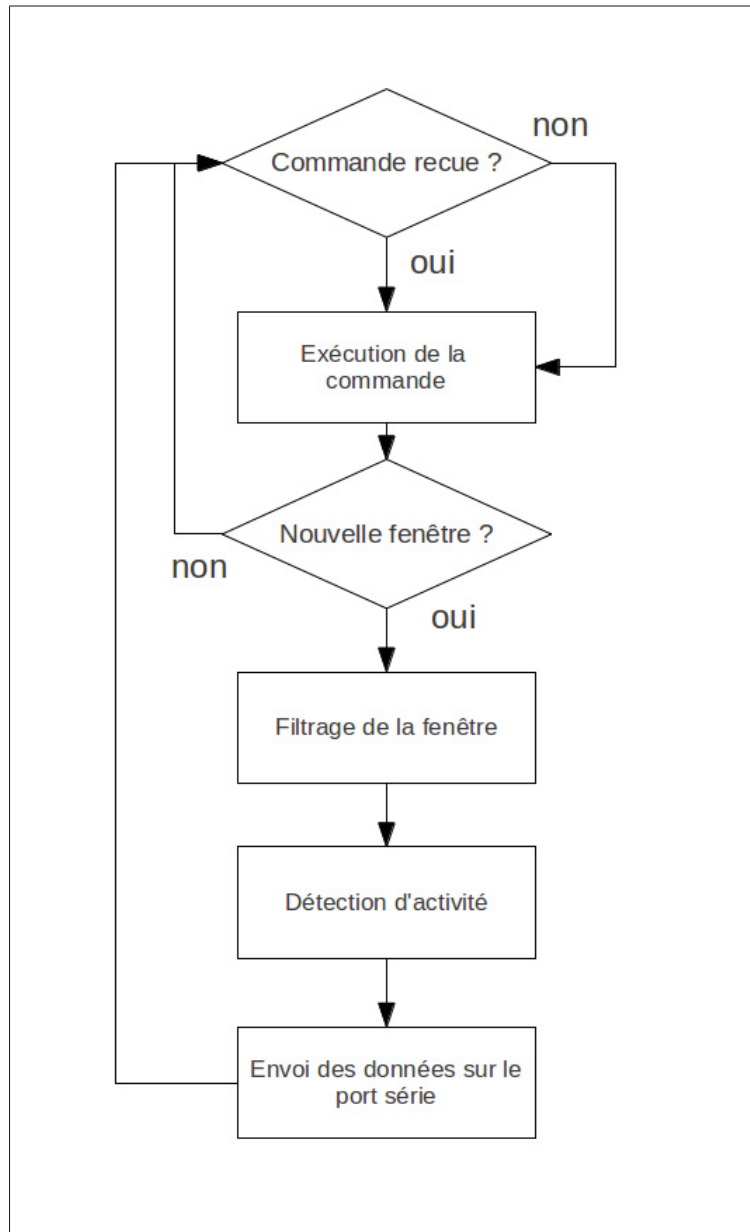


Figure 4.16 Boucle infinie du programme principal.

4.4.2 Application de monitoring des signaux

La partie firmware ayant été présentée, le dernier élément technique du projet est la réalisation du programme de monitoring des signaux. Cette application est un programme tournant sur un ordinateur. L'ordinateur, grâce aux données reçues à travers la liaison série, est en mesure d'afficher les données acquises ou calculées par le PIC32.

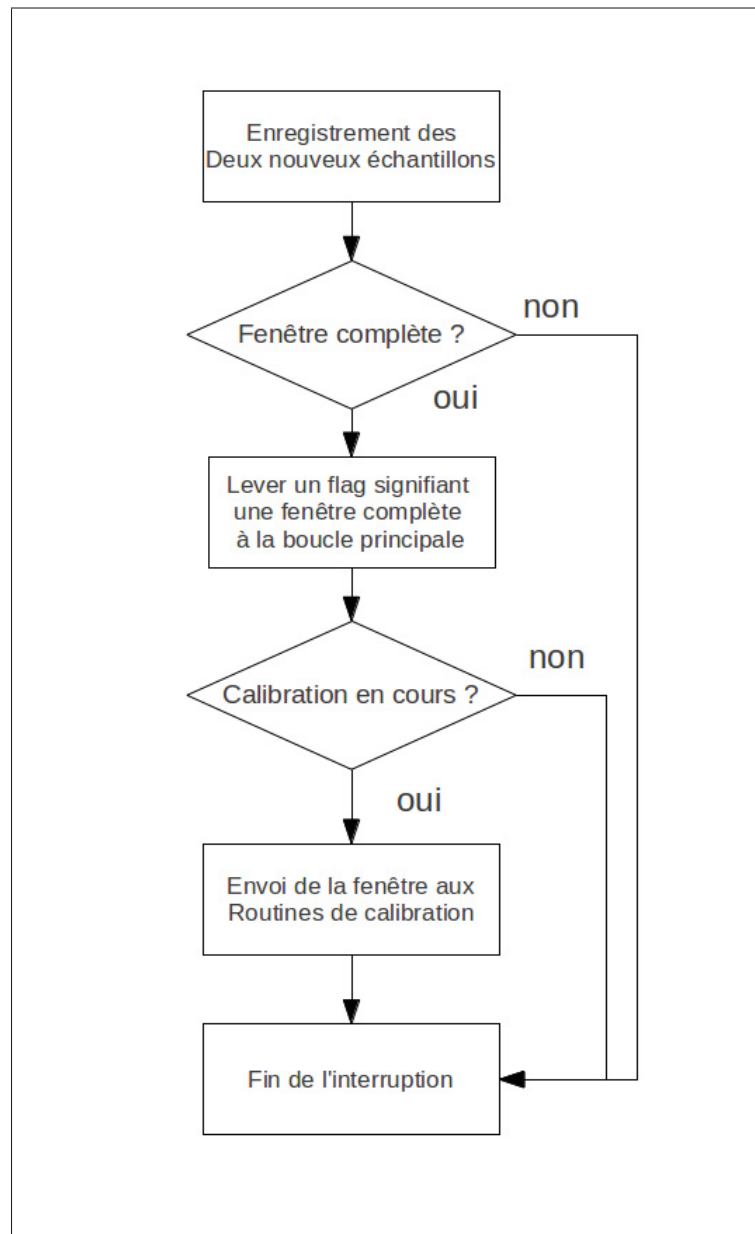


Figure 4.17 Routine d'interruption du firmware.

Ceci a été utile lors de la réalisation dans un premier temps à des fins de débogage, puis après pour l'utilisation même du système. En effet, l'interface graphique permet de lancer les phases de calibration du système, pour calibrer les seuils de détection d'activité musculaire, et pour calibrer la bande de base du filtre.

Ce programme permet également de faire le lien entre le PIC32 et le bras robotique JACO. Cette partie a nécessité l'ajout d'une sur-couche logicielle supplémentaire pour cause d'incompatibilité de langages comme le présentera la partie 4.4.2.2.

Enfin la conception complète, à savoir le diagramme UML du programme ainsi que son algorithme principal seront détaillés, ainsi que toutes les fonctionnalités finales du programme.

4.4.2.1 Choix des bibliothèques et des langages

Le choix des langages de programmation ont été effectués au début du projet. En lisant la documentation fournie par Kinova, il s'est avéré possible d'interfacer Jaco avec un programme C++. De plus, l'état de l'art de la robotique moderne utilise largement le langage C++, qui en est devenu un des standards.

En effet, le framework open source ROS (Robot Operating System) est intégralement développé en C++ et Python, deux langages facilement interfaçables.

La bibliothèque Qt est une bibliothèque graphique libre qui permet la création facile et multi plateforme d'interfaces graphiques robustes. Cette bibliothèque est l'une des plus complètes existant aujourd'hui et possède des modules supplémentaires multi plateforme en grand nombre, comme un module réseau, des bases de données, la gestion des fichiers ou encore l'utilisation d'architectures modèle-vue déjà implémentées. Ceci permet d'ouvrir beaucoup d'options pour la suite du développement du projet, dans le cas où de nouveaux éléments viendraient à y être ajoutés.

La combinaison de ces arguments a donc logiquement tourné le choix du langage informatique principal du programme vers le langage C++.

Cependant, après plusieurs mois de développement, il s'est avéré que l'utilisation des bibliothèques de Kinova doit se faire en passant par une couche logicielle supplémentaire : Mono. Cette partie est détaillée dans la partie suivante.

En effet, les bibliothèques de Kinova ont été nativement réalisées en langage C#. Le C# est un langage développé par Microsoft et qui est de ce fait, conçu pour fonctionner de manière optimale

sous Windows. Afin d'augmenter les possibilités de développement en C#, une librairie a été conçue par la communauté open source afin de rendre le C# utilisable sur d'autres systèmes d'exploitation comme linux ou MacOS.

4.4.2.2 Lien avec Jaco et Mono

Le but du projet étant de contrôler le bras JACO, il est nécessaire d'utiliser l'interface de programmation (API) de Kinova pour le contrôler. Cette librairie permet la connexion du bras JACO au programme via USB, et implémente toutes les méthodes permettant d'envoyer des ordres précis à JACO. Ces ordres peuvent être des ordres de configuration du bras, ou bien des commandes pour lire des valeurs d'état du bras (courant dans les articulation, température des moteurs, angles des articulations, ...), des commandes de déplacements des articulations, ou mêmes des trajectoires de mouvement.

L'API Kinova a été réalisée intégralement en C#, et est donc inutilisable directement par un programme codé en C++ pur. Dans sa documentation, Kinova préconise l'utilisation d'un framework appelé Mono pour l'utiliser depuis un programme C++.

Mono est une librairie permettant de mettre à disposition des structures de données (classes, fonctions...) du monde C++ accessibles au monde .NET (C#) et inversement. De ce fait, un programme C++ peut déclarer des objets définis en C# et utiliser ses méthodes.

Le monde .NET permet de créer, à la compilation un type de fichiers appelés des Assembly. Ce sont des fichiers de type .exe ou .dll directement générés par du C#. Dans le cas du système, le but est d'avoir accès au code contenu dans les fichiers dll de Kinova.

L'API kinova est constituée de sept DLL regroupant diverses classes et fonctions utiles à l'utilisation de JACO. Plutôt que d'importer les 7 dll de kinova dans mono, et de devoir aller chercher chaque classe ou fonction dans le bon Assembly, il a été conseillé par un ingénieur Kinova d'encapsuler toutes les classes et fonctions requises et de générer une nouvelle DLL unique regroupant toutes ces fonctions. Cela permet de n'appeler qu'un seul fichier .dll ne contenant

uniquement que les fonctions et classes dont on a besoin pour la réalisation du système. Cela permet un code beaucoup plus propre et beaucoup moins lourd à réaliser tant qu'à assimiler.

On crée ensuite une classe JACO en C++. Cette classe propose toutes les méthodes de la classe C# de l'API Kinova. Toutefois, les méthodes de cette classe contiennent les appels aux fonctions C# grâce à Mono. De cette manière, JACO est interfacé comme une classe normale, et aucun code concernant Mono ne sort de cette classe.

Tous ces étages réalisés permettent de faire abstraction de toute la partie gérée par Mono et d'utiliser l'API de Kinova comme si celle-ci était accessible directement en C++. Le schéma 4.18 résume l'utilisation des librairie Kinova dans le système.

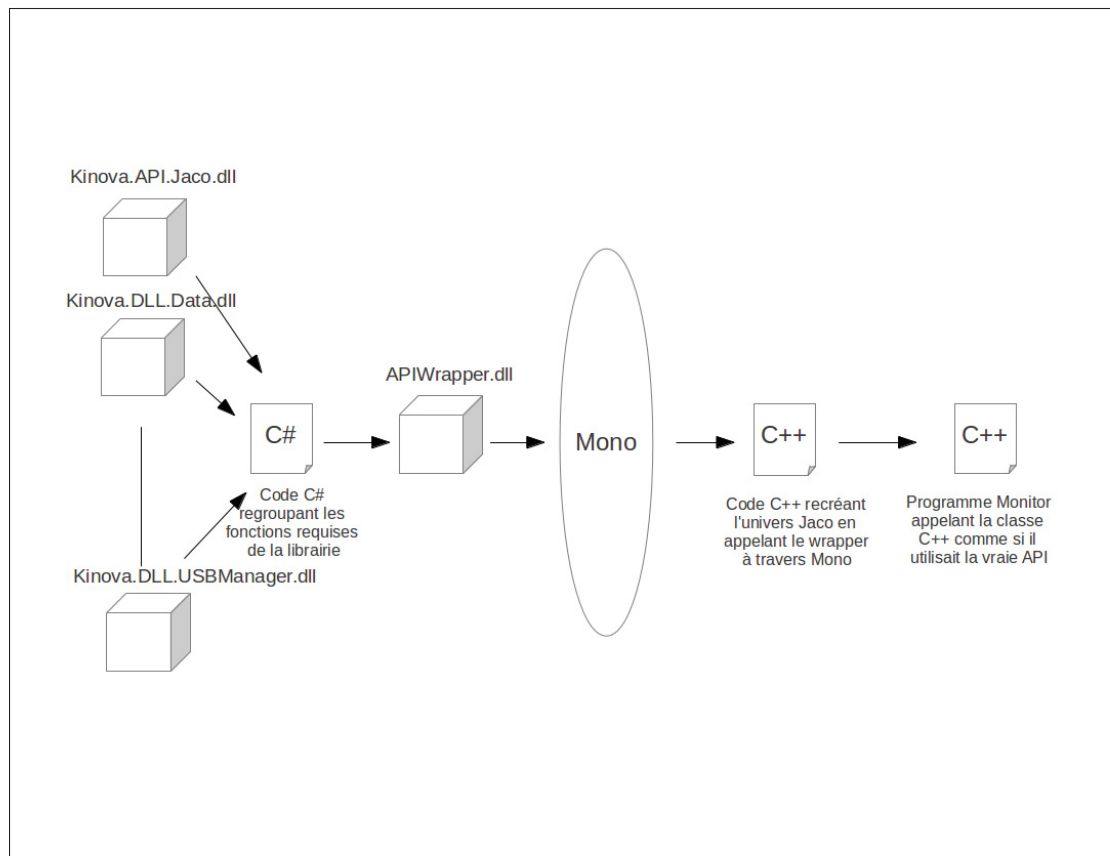


Figure 4.18 Interfaçage de l'API Kinova à travers Mono.

En synthétisant :

- On écrit un programme C# regroupant toutes les fonctions de la librairie Kinova dont ont a besoin. Ce programme fait appel directement aux fichiers .dll fournis par Kinova.
- On construit un nouvel assembly : APIWrapper.dll qui sera appelé dans le programme de monitoring
- Grâce à Mono, on charge les classes et méthode de APIWrapper dans le programme C++ grâce à une classe Jaco
- Le programme se connecte à Jaco en appelant les méthodes de la classe Jaco

4.4.2.3 Fonctionnement global et schéma UML du projet

Le logiciel de monitoring des signaux est un programme basé autour de l'interface graphique réalisée grâce à la librairie Qt. Le diagramme UML (Unified Modeling Language) du programme est présenté par la figure 4.19.

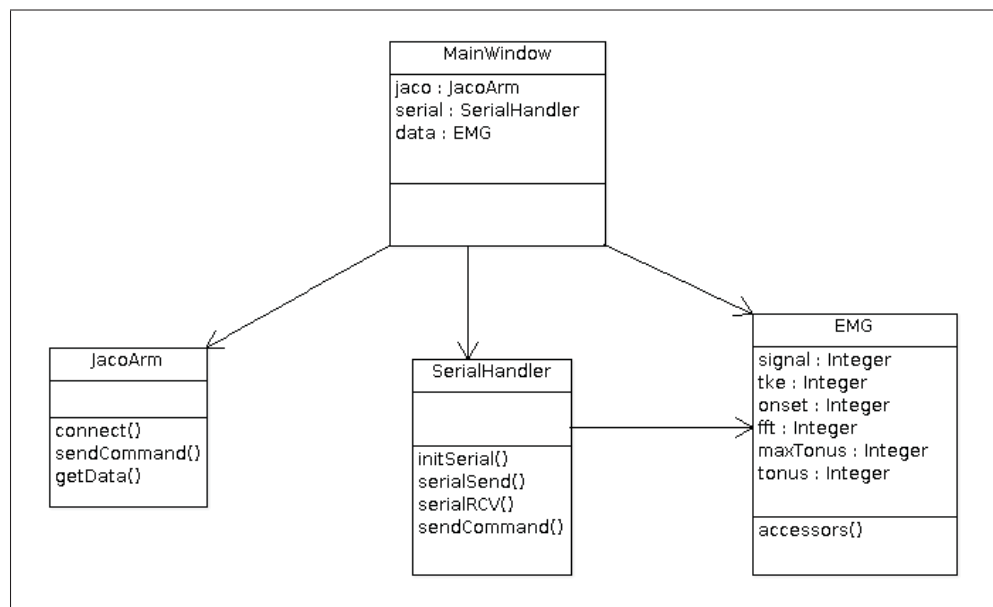


Figure 4.19 Diagramme UML du programme de monitoring.

MainWindow est la classe principale de l'interface graphique, c'est elle qui contient tous les éléments graphiques de l'interface. Elle dialogue avec les autres classes afin d'en afficher les données entrantes et sortantes.

La classe `SerialHandler` est l'objet qui gère tout ce qui touche à la communication série. Elle possède une méthode de réception des données qui tourne dans un processus séparé du reste du programme.

La classe `EMG` est la classe contenant toutes les données EMG reçues du PIC32. Lorsqu'une donnée est reçue, elle est réceptionnée par l'objet `SerialHandler` qui met à jour les données dans la classe `EMG`. `MainWindow` va ensuite chercher les données dans la classe `EMG` afin de les afficher dans l'interface.

La classe `JacoArm` est la classe décrite dans la partie précédente qui fait le lien avec les méthodes des bibliothèques de Kinova à travers `Mono`. C'est dans cette classe que `Mono` est utilisé.

L'interface graphique réalisée possède plusieurs onglet permettant chacun de visualiser un signal pour les quatre canaux. La figure 4.20 montre la fenêtre de visualisation des signaux EMG pour les quatre canaux. Un autre onglet permet de visualiser le TKE, un autre la détection d'activité musculaire et un dernier permet de monitorer une fft.

Deux onglet supplémentaires viennent s'ajouter aux quatre onglets de visualisation des signaux. Le premier, présenté par la figure 4.21 permet le contrôle de `Jaco`. En effet, `Jaco` étant normalement contrôlé par son joystick, il faut donc activer ou désactiver son contrôle par l'API. Cette fenêtre permet également de visualiser le statut de cette connexion et de visualiser les informations renvoyées par `Jaco`.

Enfin le dernier onglet présenté sur la figure 4.22 permet d'envoyer les commandes de calibration au PIC32. Ces calibrations sont la calibration du tonus au repos, du tonus maximum et de la bande de base pour le filtre, et ce pour chaque canal EMG.

4.5 Validation du Système

Le système ayant maintenant été décrit dans son ensemble, il est maintenant important de présenter la manière utilisée pour valider ses performance sur un utilisateur. La partie suivante présente le protocole de validation de performances du système tel qu'il a été présenté en comité d'éthique en mai 2013, et présente les expérimentation prévues.

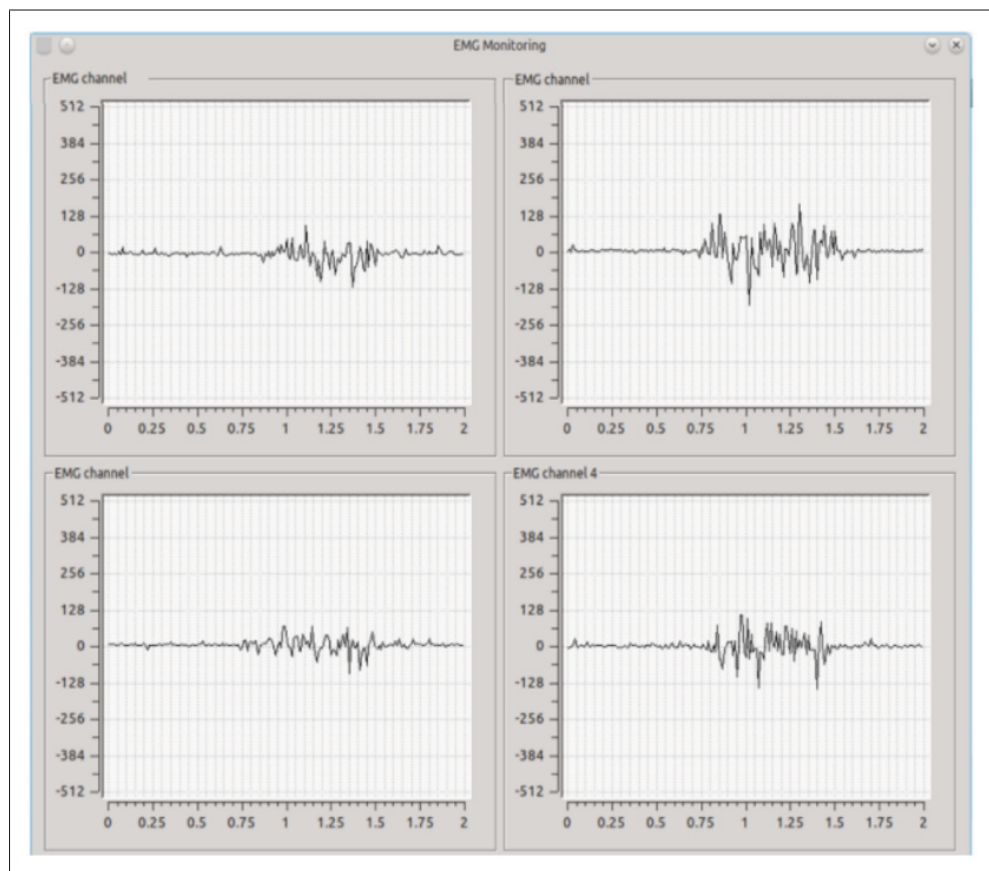


Figure 4.20 Capture d'écran de l'interface graphique du programme de monitoring des données EMG.

4.5.1 Protocole de validation

4.5.1.1 Mode de recrutement

Les participants seront recrutés par affichage au sein du laboratoire de recherche de recherche en imagerie et orthopédie ainsi que dans le département de génie de la production automatisée. Ils devront répondre aux critères suivants :

- Critères d'inclusion
 - Les sujets seront des adultes volontaires de plus de 18 ans.
- Critères d'exclusion, le sujet devra pas être sujet à :

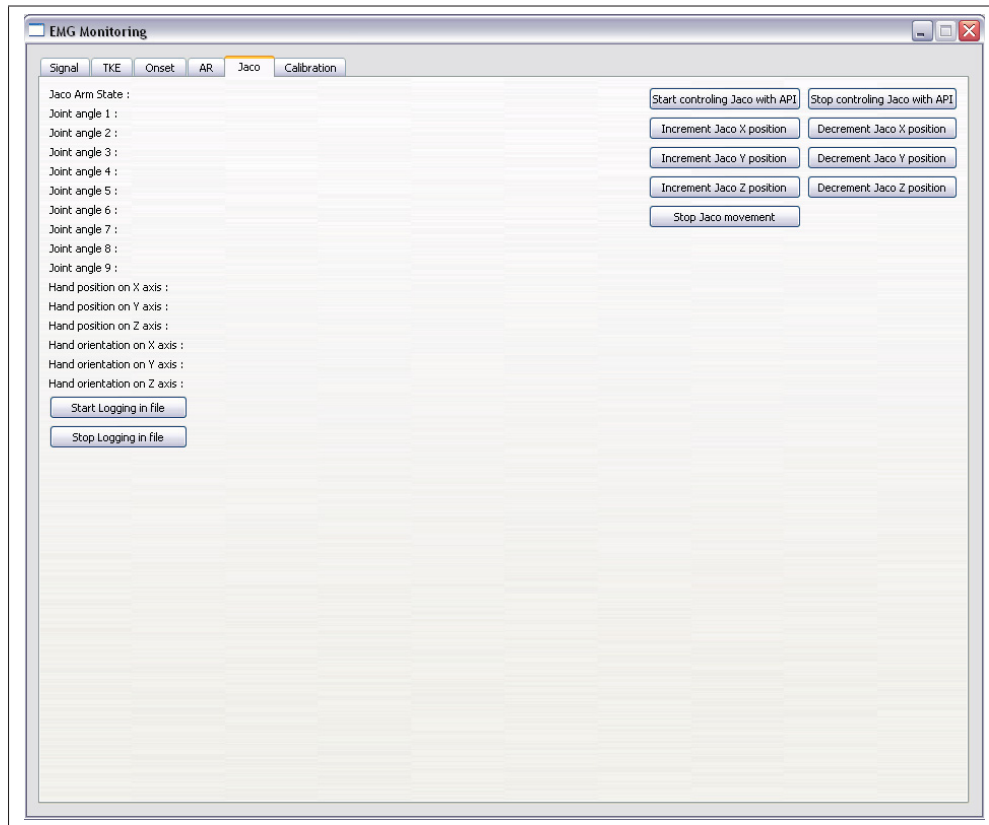


Figure 4.21 Capture d'écran de l'interface graphique du programme de monitoring des données EMG.

- des douleurs cervicales
- des torticolis fréquents
- tout problème orthopédique de la ceinture scapulaire et cervicale
- des maux de tête réguliers

4.5.1.2 Principe de fonctionnement

La détection de l'activité musculaire se fera par un seuil appliqué sur l'énergie du signal (TKE) du signal EMG échantillonné à 2KHz. Le but de ce protocole est de déterminer la sensibilité du système en le testant sur 10 sujets. L'activité musculaire générée lors d'un mouvement dépend de l'amplitude du mouvement effectué, ainsi que de la vitesse à laquelle le mouvement est effectué. Ainsi la validation s'effectuera en faisant effectuer au sujet plusieurs mouvements d'amplitudes différentes à des vitesses différentes.

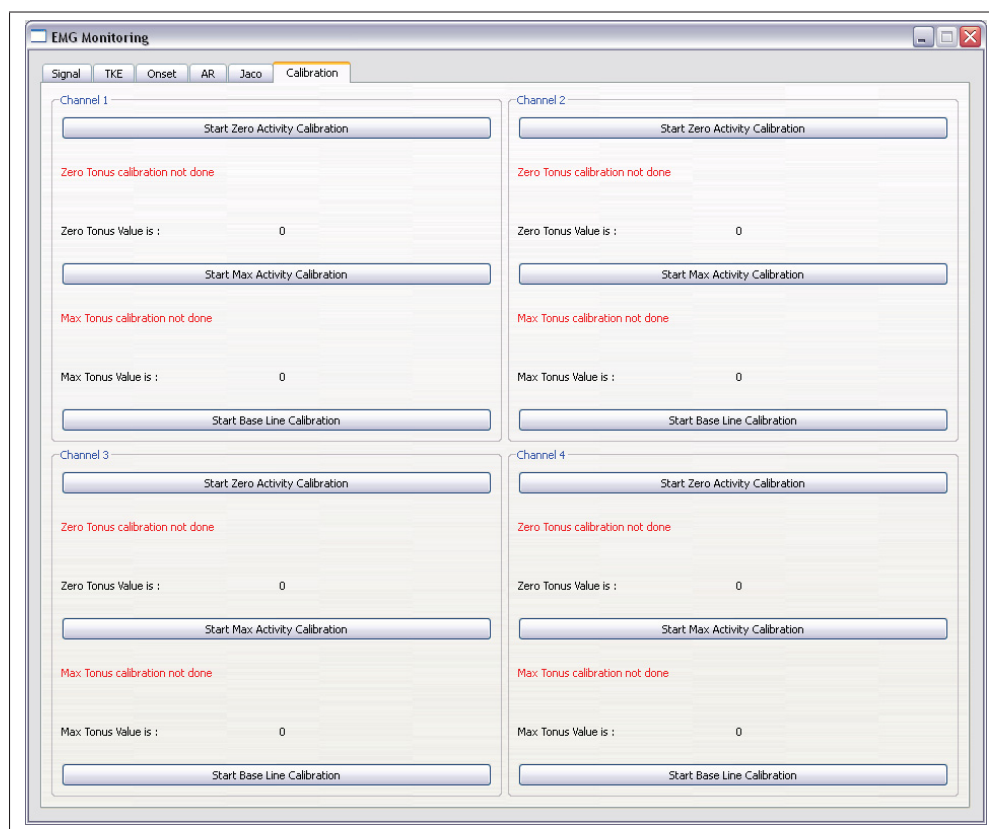


Figure 4.22 Capture d'écran de l'interface graphique du programme de monitoring des données EMG.

4.5.1.3 Protocole expérimental

Un capteur inertiel (Interlink, Microstrain) sera utilisé afin d'enregistrer l'angle de rotation de la tête ainsi que la vitesse angulaire du mouvement. Celui-ci sera disposé sur un casque de type casque d'écoute audio avec les oreilles comme point de repère de placement.

Un pointeur laser (HTCL) sera également disposé sur la tête du sujet afin que celui-ci puisse indiquer au sujet l'orientation précise de sa tête.

L'angle visé sera indiqué au sujet par des marqueurs disposés devant lui et vers lesquels il devra tourner la tête en se repérant à l'aide du pointeur laser. La vitesse visée sera indiquée au sujet par le biais d'un métronome électronique. Le sujet devra réaliser le mouvement vers le marqueur indiqué en un temps de métronome.

La figure 4.23 décrit l'angle de rotation de la tête à effectuer par le sujet.

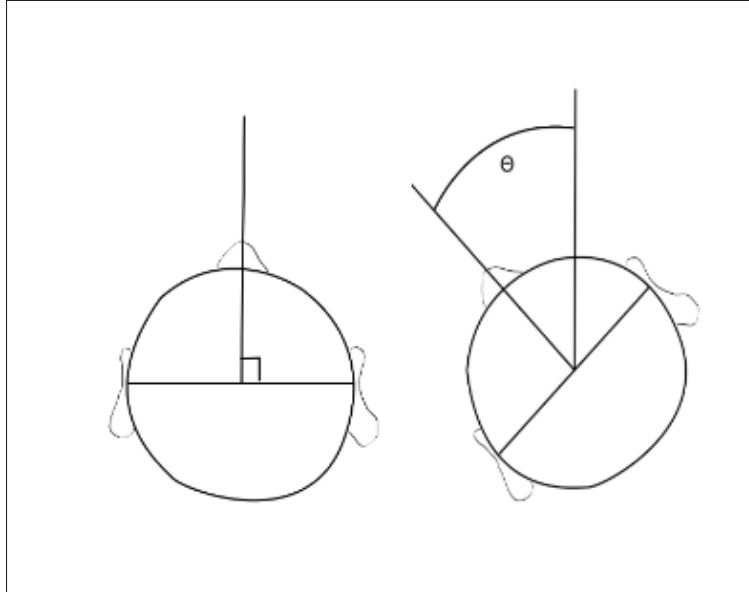


Figure 4.23 Description de l'angle de rotation de la tête.

4.5.1.4 Déroulement des acquisitions

Le sujet sera assis devant une surface comportant les cibles qu'il devra viser avec le pointeur laser. Le pointeur laser ainsi que le capteur angulaire seront disposés sur un casque d'écoute audio qui sera placé sur les oreilles du sujet. Une étape de familiarisation au métronome et d'entraînement à la visée permettra au sujet de s'entraîner à effectuer le mouvement aux angles prévus à la bonne vitesse.

Voici une liste des mesures qui seront prises pendant l'expérimentation :

- Angle de rotation de la tête
- Vitesse angulaire de rotation de la tête
- Signaux électromyographique du muscle sterno-cléïdo mastoïdien gauche
- Signal de détection d'activité musculaire

Et ce pour chaque combinaison d'angle et de vitesse de métronome ciblés.

Le sujet devra réaliser des mouvements de rotation de la tête à des angles de 10 degrés, 20 degrés, 30 degrés, 40 degrés, 50 degrés, 60 degrés, 70 degrés et 80 degrés. Chaque déplacement angulaire sera effectué à des vitesses de métronomes de *60bpm*, *90bpm*, *120bpm*, *160bpm* et *200bpm* (battement par minute) .

L'ordre des vitesses et des angles sera tiré au hasard avant l'expérimentation pour chaque sujet.

4.5.1.5 Notes importantes

Après chaque mouvement un fichier devra être sauvé sous la forme *aXXvYYY.csv* où *XX* correspond à la valeur de l'angle visé lors de l'enregistrement et *YYY* correspond à la valeur du métronome utilisé pour l'enregistrement.

A chaque acquisition, un monitoring sera effectué sur le programme Monitor afin de vérifier si le système a détecté une activité musculaire ou non. Le fichier de sauvegarde devra être sauvegardé dans le même dossier que les données du capteur angulaire sous la forme *emgaXX-vYYY.txt*

4.5.1.6 Données récoltées

Les figures 4.24, 4.26, 4.25 et 4.27 montrent un exemple de la forme que prendront les données récoltées lors d'une acquisition sur une personne : (A20, A30, A40, A50, A60 correspondent à l'angle de rotation effectué, et V60, V90, V120, V160 et V200 correspondent aux vitesses du métronome pour l'acquisition)

Ces données représentent les valeurs brutes enregistrées grâce aux capteurs inertiel et EMG.

4.5.1.7 Interprétation des données

Une fois les données acquises, 4 paramètres seront estimés pour évaluer la sensibilité du système correspondant au sujet testé : l'efficacité, le coefficient de l'angle maximum, le coefficient de la vitesse angulaire maximale ainsi que l'efficacité globale.

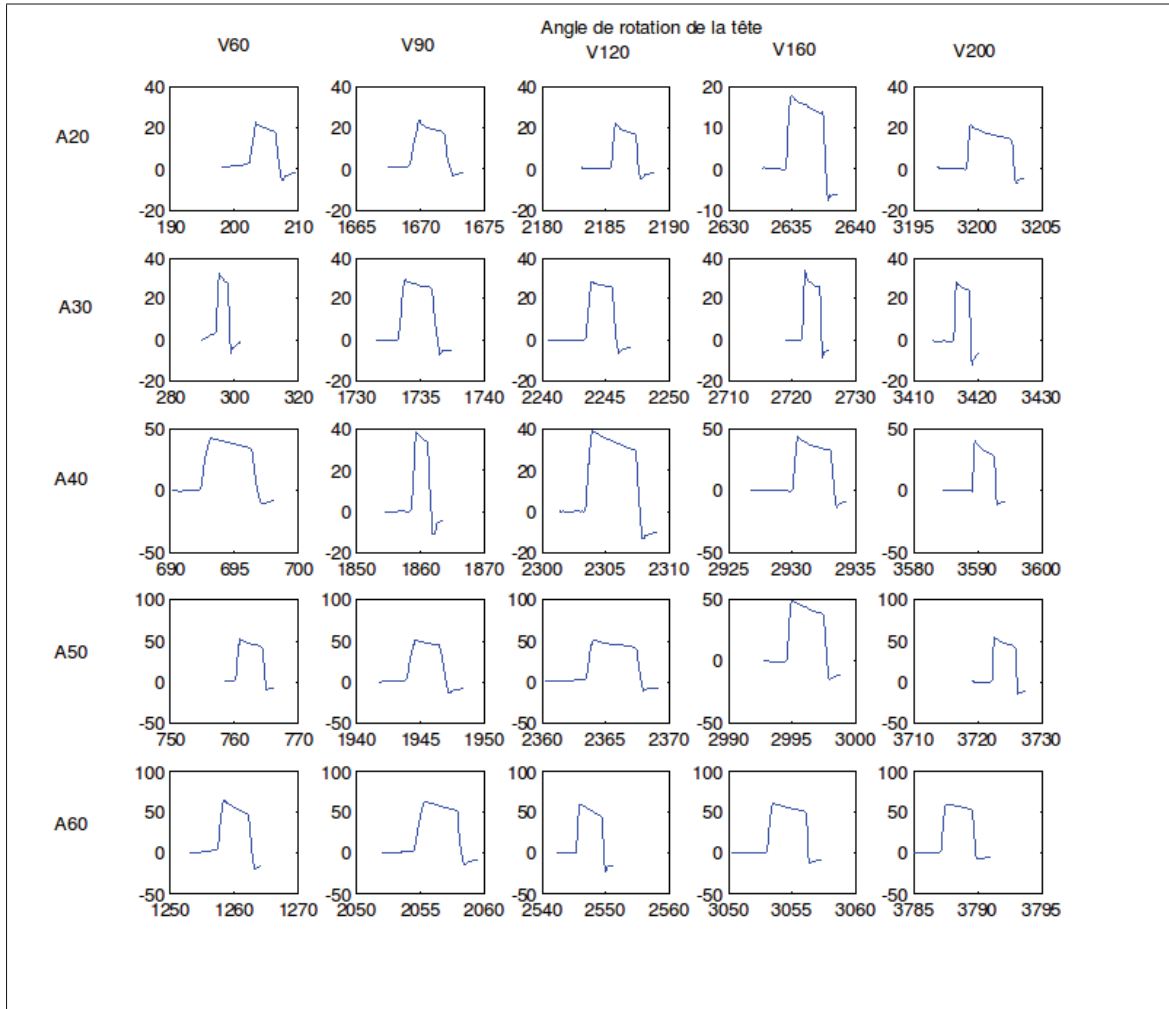


Figure 4.24 Exemple de données d'angles de rotation de la tête. Ordonnées : angles en degrés, abscisses : temps en secondes.

Le paramètre de l'efficacité est défini comme le rapport entre le nombre de détections sur le nombre d'essais total comme montré par l'équation 4.22.

$$Efficacite = \frac{Nb\ de\ dtectons}{Nb\ d'essais\ total} \quad (4.22)$$

On est en mesure de donner une valeur entre 0 et 1 à l'efficacité du système pour une personne donnée. La valeur 1 indique que le système a détecté toutes les activités réalisées par la personne et la valeur 0 indique au contraire que le système n'en a détecté aucune.

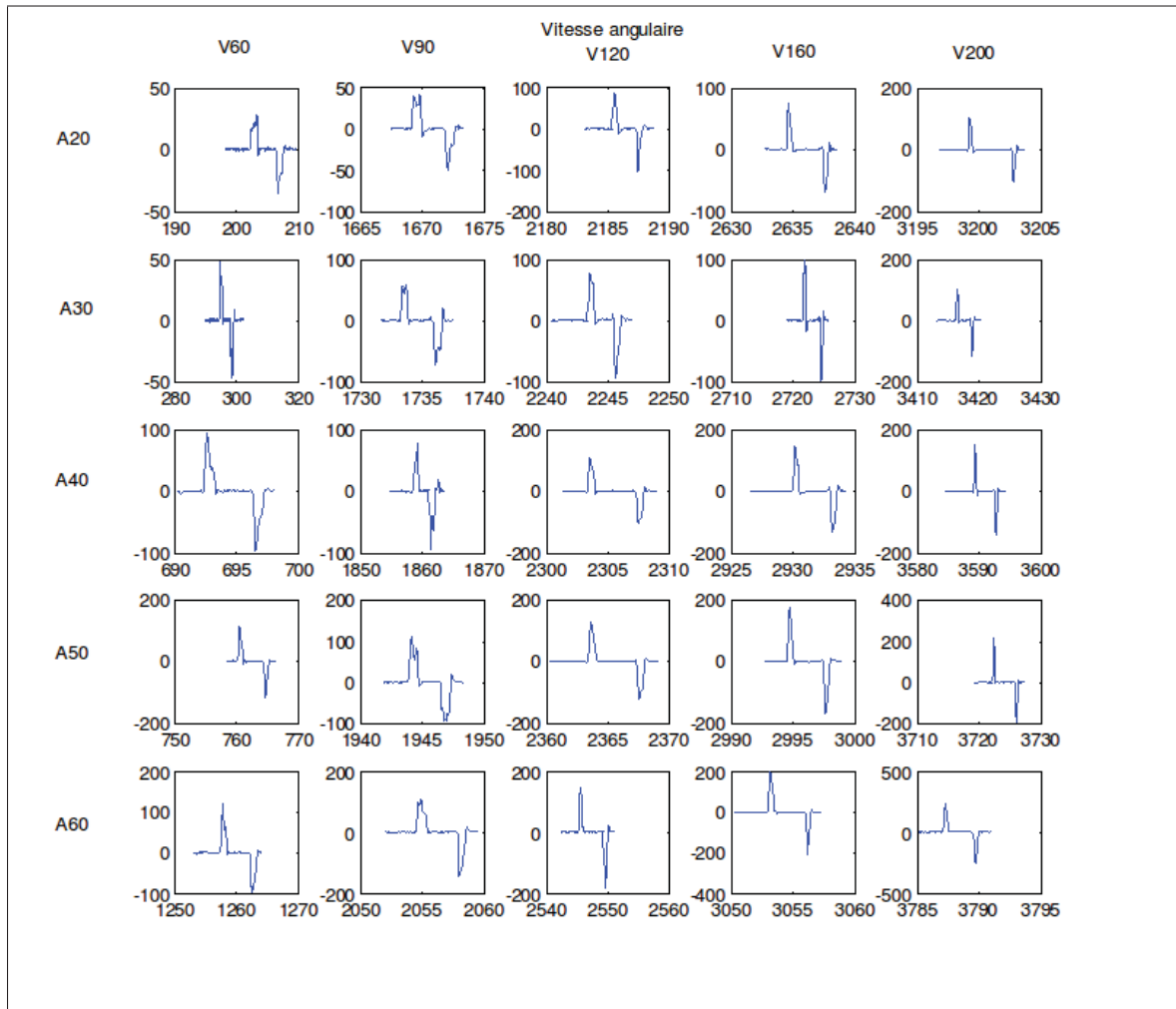


Figure 4.25 Exemple de données de vitesse angulaires de rotation de la tête. Ordonnées : vitesses en degrés par seconde, abscisses : temps en secondes.

Cependant ce paramètre ne prend pas en compte les angles et vitesses angulaires atteintes lors des expérimentations. On détermine donc un coefficient d'angle ainsi qu'un coefficient de vitesse angulaire, qui viendront pondérer le coefficient d'efficacité de chaque sujet. Ces coefficients seront basés sur l'angle maximum atteint par un sujet et la vitesse angulaire atteinte par un sujet. Ainsi les résultats seront pondérés relativement au sujet ayant atteint l'angle et la vitesse angulaire les plus grands.

Ainsi on fixe un angle maximum égal au plus grand atteint par un sujet, ce sujet aura un coefficient d'angle de 1. Les autres auront un coefficient défini comme le définit l'équation

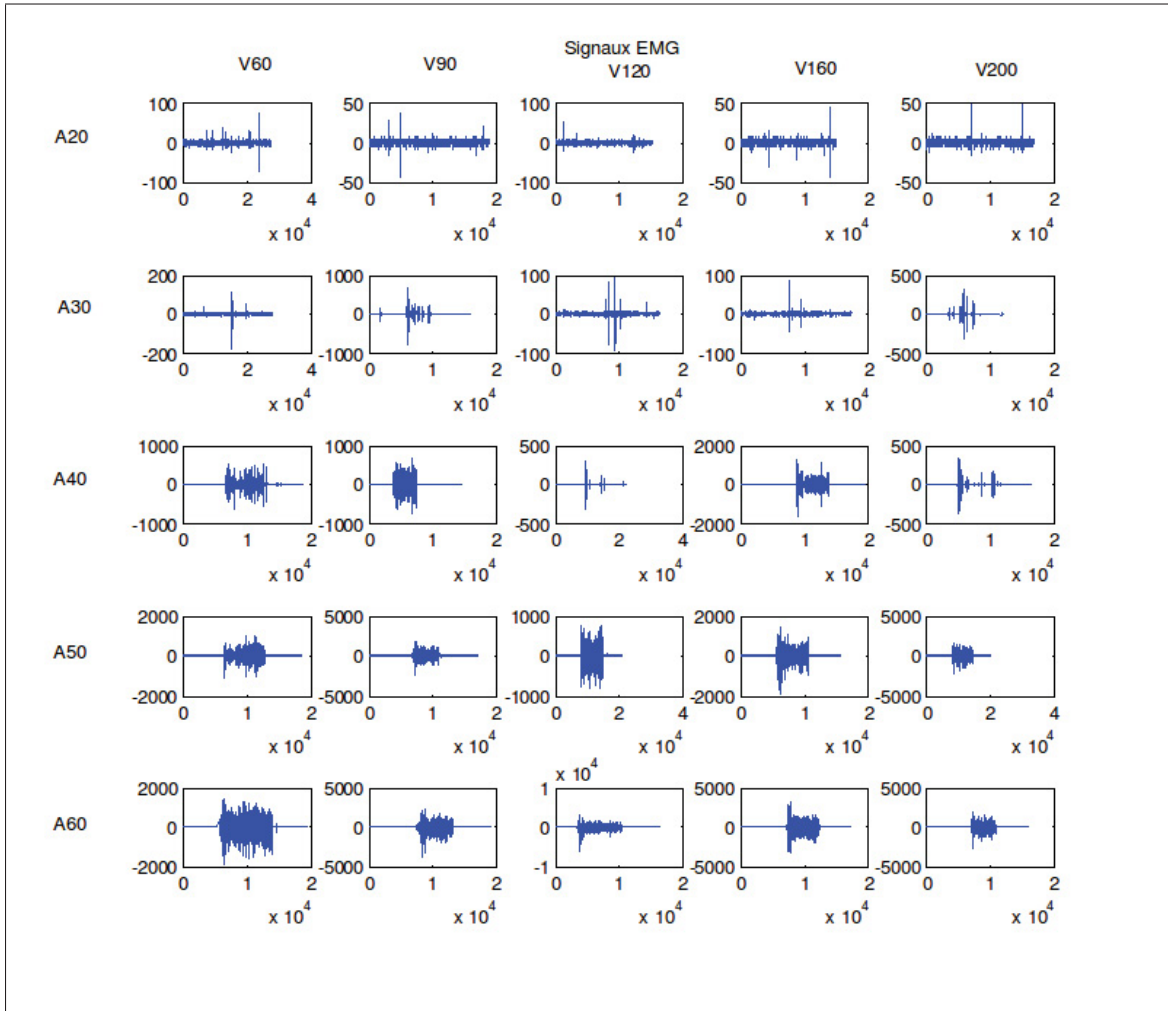


Figure 4.26 Exemple de données d'EMG de rotation de la tête. Ordonnées : EMG en unités ADC, abscisses : temps en échantillons.

4.23.

$$Coeff_{angle} = \frac{Angle\ max\ du\ sujet}{Angle\ max\ atteint\ par\ un\ sujet} \quad (4.23)$$

De la même manière un coefficient de vitesse angulaire sera appliqué. On fixe une vitesse angulaire maximum atteinte par un sujet, celui-ci aura un coefficient de vitesse angulaire de 1. Les autres auront un coefficient défini comme le définit l'équation 4.24.

$$Coeff_{vitesse} = \frac{Vitesse\ angulaire\ max\ du\ sujet}{Vitesse\ angulaire\ max\ atteint\ par\ un\ sujet} \quad (4.24)$$

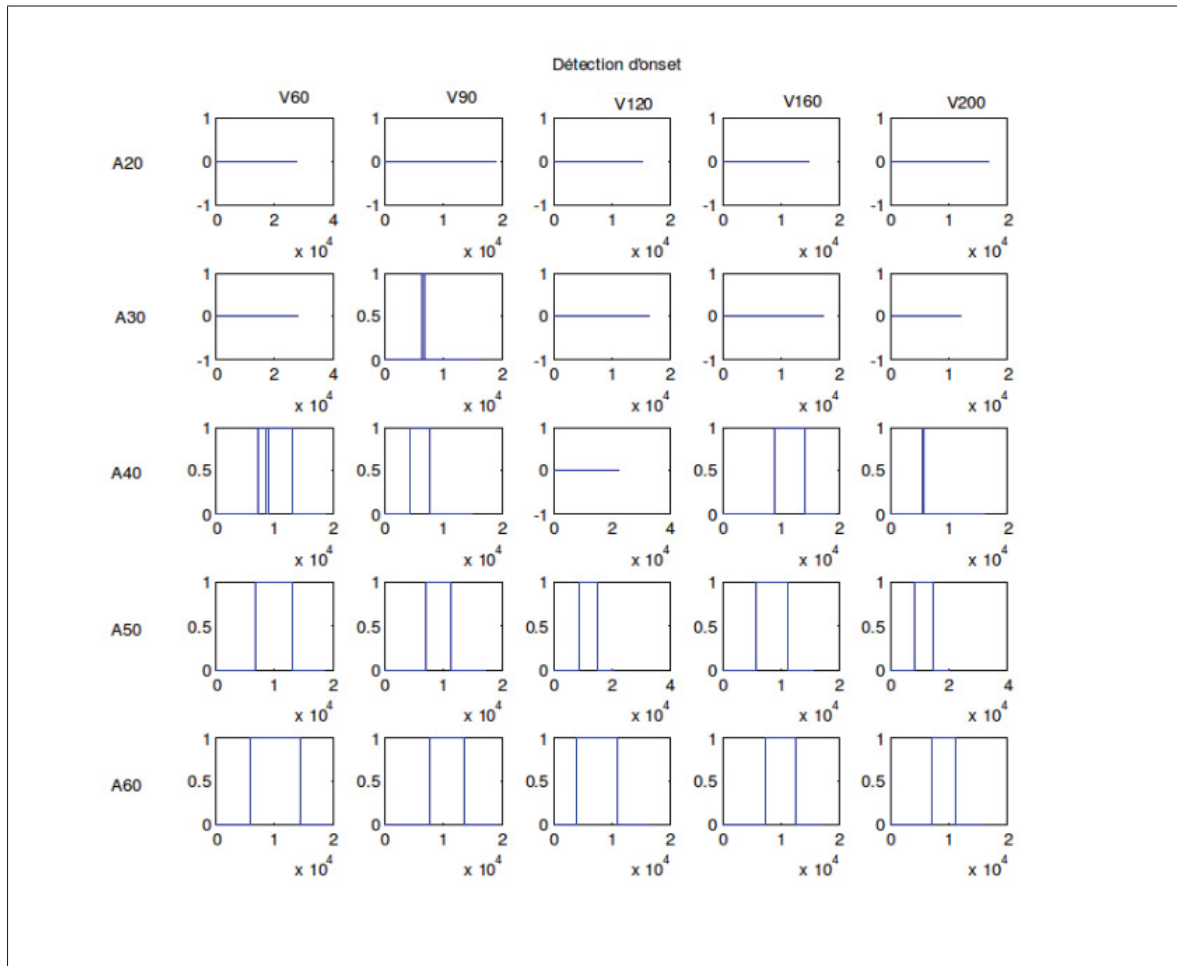


Figure 4.27 Exemple de données de détection d'activité correspondant aux signaux EMG de la figure 4.26. Ordonnées : détection d'activité, abscisses : temps en échantillons.

Cela nous permet donc de fixer une efficacité globale pour chaque sujet qui sera calculée à partir de l'équation suivante :

$$Eff_{globale} = Efficacite * Coeff_{angle} * Coeff_{vitesse} \quad (4.25)$$

4.5.1.8 Comparaison des données

Les valeurs d'efficacité globale de chaque sujet seront mises en commun et comparées les unes aux autres afin de savoir si le système fonctionne de manière similaire sur plusieurs personnes ou si ses performances sont variables d'une personne à l'autre. Une personne ayant une détec-

tion d'activité de 100% aura une efficacité de 1, une personne ayant une détection d'activité de 0% aura une efficacité de 0. Ainsi, plus un score d'efficacité est bon, plus il est proche de 1. Une efficacité moyenne, ainsi qu'une efficacité globale moyenne pourront être déterminées ainsi que leurs écarts types. En comparant les performances des sujets, si un sujet obtient un score plus faible que les autres, les données nous permettront d'en étudier les causes, et permettront potentiellement de trouver des pistes d'amélioration du système.

CHAPITRE 5

RÉSULTATS

le projet et sa conception étant maintenant détaillé, cette partie va maintenant présenter les résultats obtenus. Dans un premier temps, le comportement global du projet sera évoqué. Les temps de calculs du système seront présentés. Enfin les résultats obtenus sur 10 sujets de tests humains seront présentés.

Ces résultats seront analysés dans la discussion.

5.1 Performances globales

Dans un premier temps, avant de quantifier les performances du système, il est nécessaire de résumer ce qu'il est possible d'exécuter avec celui-ci.

La figure 5.1 montre le comportement du firmware pour un canal EMG. L'activité musculaire est détectée efficacement, et aucune retombée de l'activité ou de faux positif n'est enregistré. Cette détection d'activité a été réalisée en utilisant préalablement la calibration du tonus musculaire au repos et du tonus musculaire maximum.

Le système complet a été testé : utilisation des quatre électrodes pour contrôler Jaco. La figure 5.2 présente le système complet et en utilisation. Chaque canal EMG a été calibré séparément, puis des commandes ont été envoyées au bras robotique en fonction des activités musculaires détectées.

Pour les tests, la commande à envoyer à Jaco choisie consiste à déplacer la main du robot sur l'axe x et l'axe y . Les mouvement de rotation de la tête de l'utilisateur déplaçaient ainsi le bras à droite et à gauche tandis qu'une élévation des épaules le faisaient se déplacer en avant et en arrière.

Il est à noter que grâce à l'utilisation de la librairie de programmation fournie par Kinova, n'importe quelle action du bras peut être reliée à chacun des mouvements de l'utilisateur.

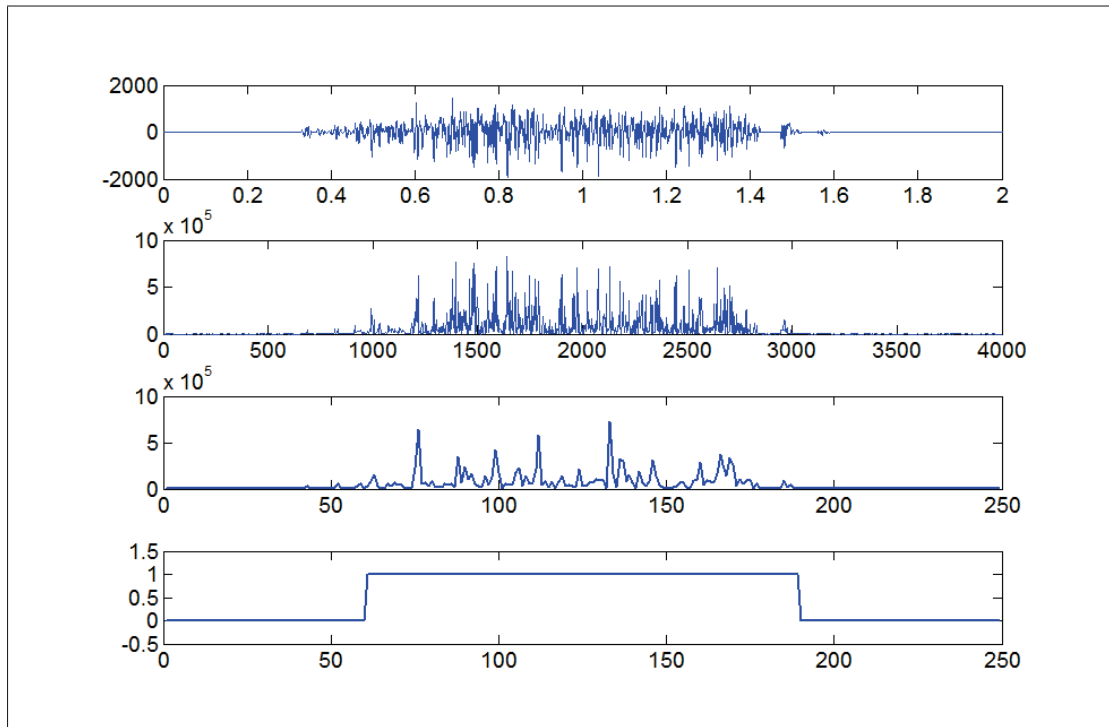


Figure 5.1 Résultats de la détection d'activité musculaire sur un canal EMG

Ces tests de fonctionnalité simples ont été réalisés sans utilisation du filtrage du signal EMG.



Figure 5.2 Contrôle de Jaco par canaux EMG

Une vidéo a été réalisée pour le projet afin de constater le fonctionnement du système de manière dynamique.

5.2 Temps de calcul et chronogrammes

Une donnée importante à collecter dans la spécification du projet est le temps de calcul nécessaire pour effectuer les calculs critiques du système, à savoir la détection d'activité musculaire et le filtrage du signal.

La filtrage ayant été mis en place seulement dans un second temps, le temps de calcul nécessaire à la détection d'activité musculaire sur les quatre canaux EMG a d'abord été enregistré.

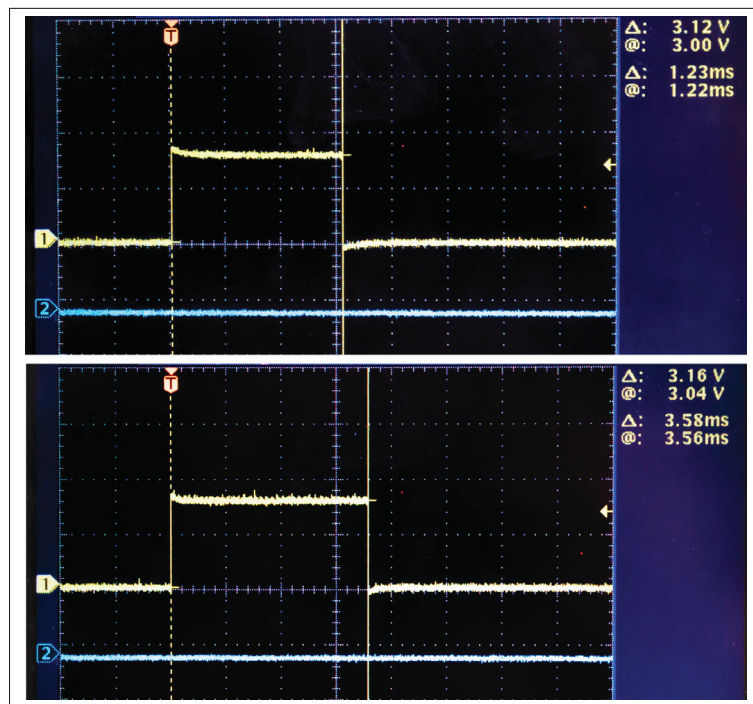


Figure 5.3 Chronogrammes de mesure du temps de calcul. Le chronogramme du haut représente le temps de calcul de détection d'activité musculaire des quatre canaux EMG. Le chronogramme du bas présente le temps de calcul de filtrage d'une fenêtre d'un seul canal EMG.

Pour ce faire, l'opération réalisée consiste à mettre une sortie du PIC32 à l'état haut au moment où le calcul commence, puis à la remettre à l'état bas lorsque le calcul est effectué. Le signal

est ensuite enregistré à l'aide d'un oscilloscope. Le temps passé à l'état haut est ensuite mesuré et utilisé comme temps de référence.

La figure 5.3 montre les chronogrammes enregistrés pour la détection d'activité des quatre canaux EMG (en haut), et pour le filtrage d'un canal EMG. Le temps de calcul nécessaire à la détection d'activité sur les quatre canaux est de 1.23 ms et celui nécessaire au filtrage d'un seul canal EMG est de 3.58 ms.

Ainsi, si on reporte ces temps de calculs sur un même chronogramme, on est en mesure de réaliser plus facilement le temps pris par ces opérations et le temps de calcul disponible avant la fenêtre d'échantillons suivante. Le chronogramme illustré dans la figure 5.4 montre la répartition du temps de calcul de la routine principale du firmware.

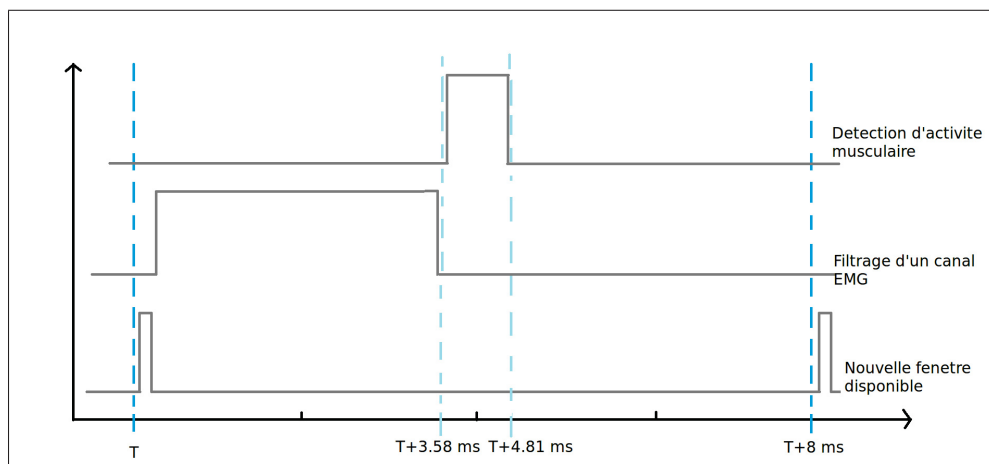


Figure 5.4 Etude du temps de calcul utilisé par le système.

Il reste donc 3.19 ms de temps de calcul disponible avant la prochaine disponibilité d'une fenêtre d'échantillons de signal EMG. Ce temps est insuffisant pour filtrer un autre canal EMG.

5.3 Validation du système

Afin de caractériser le système et d'évaluer ses performance et sa cohérence, des tests sur sujets humains sains ont été réalisés après approbation du protocole de recherche par un comité d'éthique.

Chaque sujet a été équipé d'un casque sur lequel ont été montés un laser et un capteur inertiel, ainsi que d'une électrode EMG sur le muscle sterno-cléido mastoïdien. La réalisation du montage est expliquée en détails en annexe I.

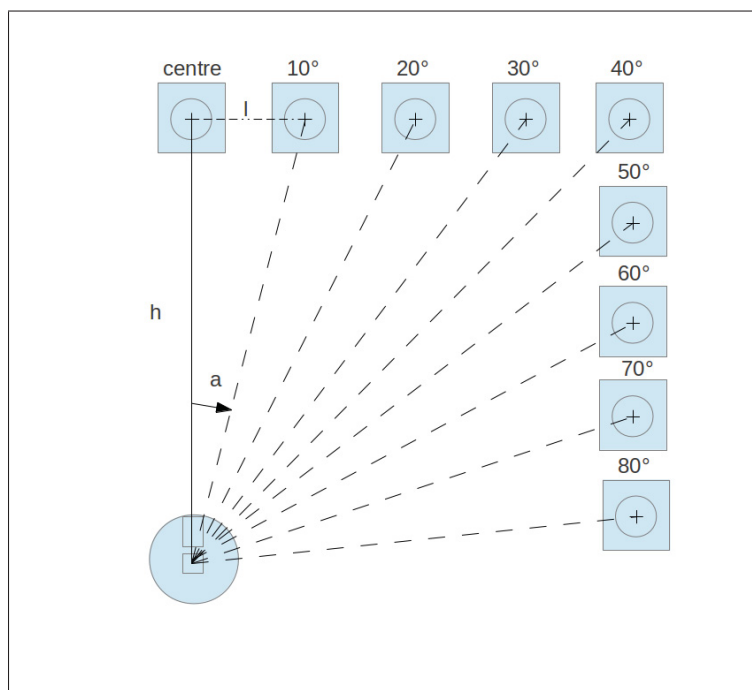


Figure 5.5 Vue du dessus du montage réalisé pour la validation du système. h représente la longueur entre la tête du sujet et la cible du centre, l représente la longueur entre la cible du centre et une autre cible et a représente l'angle de rotation de la tête du sujet. Ici $h = 1.14m$ et $l \simeq 20cm$.

Ces expérimentations ont permis d'obtenir les signaux EMG, les activités musculaires, les angles de rotations et vitesses angulaires de dix personnes. Ces données ont été compilées et mises en relations. Cette partie va présenter les données obtenues lors de ces expérimentations.

La répartition des sujets a été comme présentée par le tableau 5.1.

Le laser a permis de maintenir le capteur inertiel à l'horizontal en montant ou descendant le pointeur, obligeant le sujet à pencher la tête vers l'avant ou l'arrière pour viser les cibles. De cette manière, on est sûr que la rotation de la tête ne s'exprime que sur l'axe vertical.

Tableau 5.1 Répartition des sujets ayant participé aux expérimentations. M = masculin, F = féminin.

Sujet	Sexe	Age(ans)	Taille(cm)	Poids(kg)
S001	F	30	158	50
S002	M	26	176	63
S003	M	24	180	90
S004	F	28	162	67
S005	F	31	163	63
S006	M	29	198	112
S007	M	23	174	65
S008	M	26	179	67
S009	F	26	167	66
S010	F	23	160	50

Chaque test a également été réalisé une fois avec le filtrage du signal EMG activé, et une autre fois avec ce filtrage désactivé. Ceci permet de comparer l'efficacité du système avec et sans filtre et de quantifier le gain de performances apporté par celui-ci.

Pour chaque sujet, on calcule l'efficacité globale du système, c'est à dire le nombre de fois qu'une activité musculaire a été détectée sur le nombre de mouvements total effectué par le sujet. Le tableau 5.2 retrace les résultats bruts obtenus par les 10 sujets avec filtrage :

Tableau 5.2 Efficacités moyennes par angles obtenues par sujet avec filtrage.

Angle	10°	20°	30°	40°	50°	60°	70°	80°
S001	0	0	0.4	1	1	1	1	1
S002	0	0	0	1	1	1	1	1
S003	0	0	0	0	0.4	1	1	1
S004	0	0	0	0	0	0.2	1	1
S005	0	0	0.6	1	1	1	1	1
S006	0	0	0	0.6	1	1	1	1
S007	0	0	0	0.2	1	1	1	1
S008	0	0	0.4	1	1	1	1	1
S009	0	0	0	0.8	1	1	1	1
S010	0	0	0	0.2	1	1	1	1
Moyenne	0	0	0.14	0.58	0.84	0.92	1	1
Ecart Type	0	0	0.2319	0.4367	0.3502	0.253	0	0

Voici les mêmes résultats mais pour le système sans filtrage :

Tableau 5.3 Efficacités moyennes par angles obtenues par sujet sans filtrage filtrage.

Angle	10°	20°	30°	40°	50°	60°	70°	80°
S001	0	0	0	0	0	0.4	1	1
S002	-	-	-	-	-	-	-	-
S003	0	0	0	0	0	0.8	1	1
S004	0	0	0	0	0.8	1	1	1
S005	0	0	0	0	1	1	1	1
S006	0	0	0	0	0.8	1	1	1
S007	0	0	0	0	0.6	1	1	1
S008	0	0	0	0	1	1	1	1
S009	0	0	0	0	1	1	1	1
S010	0	0	0	0	0.6	1	1	1
Moyenne	0	0	0	0	0.6444	0.9111	1	1
Ecart Type	0	0	0	0	0.3972	0.2028	0	0

L'efficacité du système ayant été calculée par sujet, il est intéressant de regarder l'efficacité globale de celui-ci. Il est également utile d'observer cette efficacité en fonction des angles atteints par les sujets. La figure 5.6 montre l'efficacité moyenne du système, encore une fois avec et sans filtrage du signal.

Le premier couple de barres montre l'efficacité moyenne sur tous les angles du système allant de 10 degrés à 80 degrés. Celle-ci atteint une valeur de 56% pour le système avec filtrage et de 49.3% pour le système sans filtrage.

Le deuxième couple de barres montre l'efficacité moyenne du système sur les angles allant de 40 degrés à 80 degrés. Cette efficacité est de 86.8% pour le système avec filtrage et de 74% pour le système sans filtrage.

De la même manière les trois couples de barres montrent respectivement les efficacités pour les angles allant de 50 à 80 degrés, où l'efficacité avec filtrage est de 94% et sans filtrage de 90%, de 60 à 80 degrés, où les deux efficacités sont de 97.3% et de 70 et 80 degrés où les deux efficacités moyennes sont de 100%.

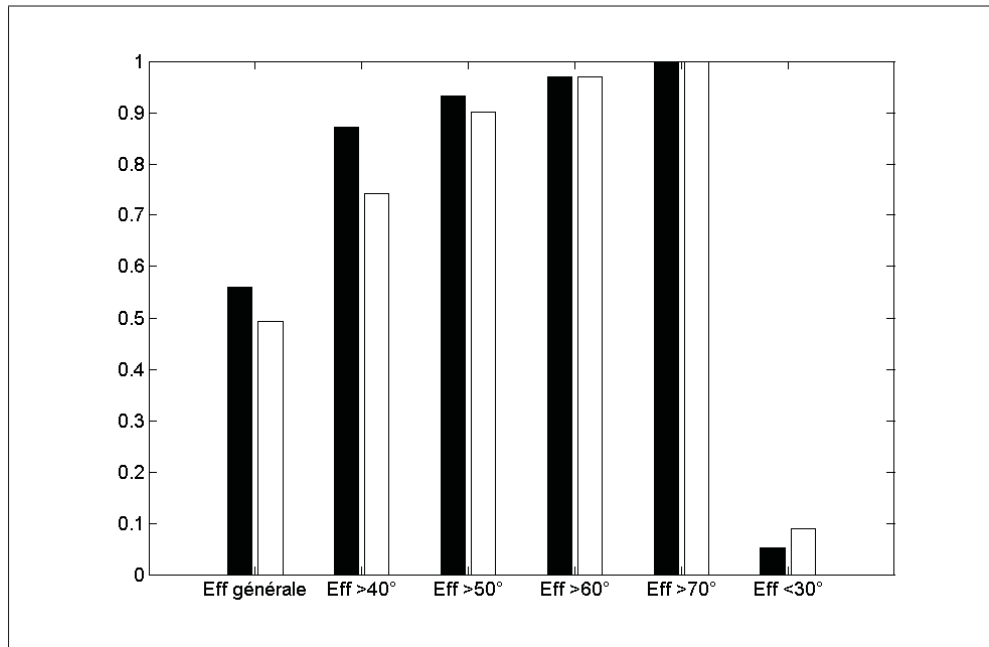


Figure 5.6 Efficacité du système moyenne pour des intervalles d'angles différents, en noir avec filtrage, en blanc sans filtrage. L'efficacité est une valeur normalisée (pas d'unité).

Enfin le dernier couple de barres montre l'efficacité moyenne du système pour les angles de 10 à 30 degrés. Cette valeur est de 4.67% pour le système avec filtrage et de 8% pour le système sans filtrage.

Les résultats ont montré dans un premier temps deux données particulières : celles du sujet S004 premièrement, puis celle du sujet S002.

Concernant le sujet S004, on constate sur la figure ?? que celui-ci est le seul sujet présentant une détection d'activité beaucoup plus faible que tous les autres sujets, et également le seul sujet présentant une performance plus élevée du système sans filtrage que celui avec filtrage (excepté le sujet S002 qui sera présenté juste après). Le sujet S004 est également le seul n'ayant pas pu physiquement atteindre l'angle de 80 degrés, comme vu sur la figure 5.8. Tout indique que l'algorithme de filtrage du signal n'a pas fonctionné correctement pour ce sujet. Ce mauvais fonctionnement n'est pas expliqué et devra être exploré dans la suite du projet.

Un autre sujet voit ses données se démarquer, le sujet S002. Cependant, seules ses données acquises lors des tests sans filtrage sont différentes des autres. Sur la figure ??, on remarque clairement que le système sans filtrage obtient de bien meilleurs résultats que pour le système avec filtrage : 92.5% contre 62.5%.

Ces résultats s'expliquent par un mauvais calibrage du système lors des expérimentation sans filtrage. En effet, lors du calibrage du système pour l'expérimentation sans filtrage, on demande dans un premier temps au sujet de mettre son muscle au repos, puis de le mettre en contraction maximum lors d'un mouvement d'amplitude maximum.

Cette calibration semble avoir été inefficace et les détections enregistrées correspondent à des faux positifs.

Ceci provient du fait que le TKE moyen enregistré pour le tonus musculaire maximum était trop proche de celui enregistré au repos.

Ce problème peut avoir plusieurs cause :

- Le sujet n'est pas allé à son angle maximum
- L'enregistrement a été effectué trop tôt, ou trop tard lorsque le sujet a relâché son mouvement

Dans tous les cas, le fait que cette calibration ait été erronée a été constaté trop tard, à la fin de l'expérimentation, et il n'était pas possible de faire recommencer les tests au sujet.

Par conséquent, les données du système sans filtrage du sujet S002 seront considérées comme aberrantes. Dans la suite de l'étude, on étudiera également le comportement du système en excluant également le sujet S004.

Intéressons nous maintenant aux données obtenues. Dans un premier temps observons les angles et vitesses angulaires maximums atteints par les participants. Sur la figure 5.7, les couples de barres encadrés sont les vitesses mesurées sur les hommes. On constate que celles-ci sont pour quatre sujets sur cinq supérieures à toutes les vitesses atteintes par les femmes.

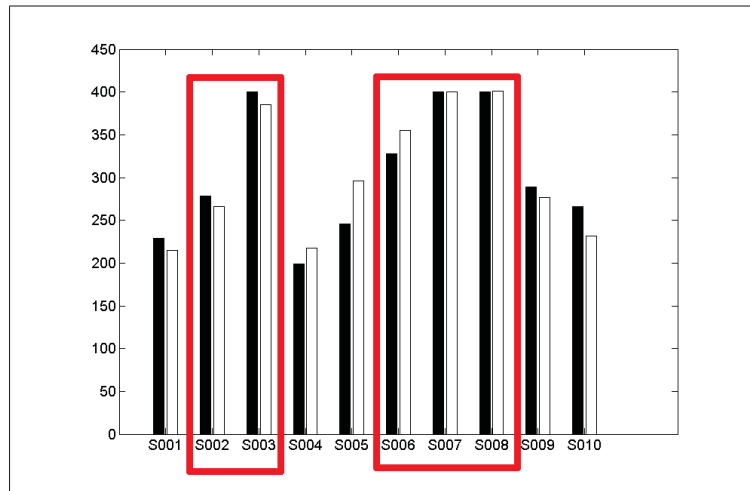


Figure 5.7 Vitesses de rotation atteintes par les sujets masculins et féminins. Les sujets hommes sont encadrés. Les valeurs en ordonnées sont exprimées en degrés par secondes

Cependant, le système ne donne pas du tout une meilleure efficacité aux hommes qu'aux femmes. En effet, les deux meilleures efficacités avec filtrages du signal sont celles des sujets S001 et S005 qui sont des femmes.

À l'inverse, sur les sujets S001, S005 et S008, on remarque que le système ne détecte jamais l'angle de 30 degrés excepté pour les deux ou trois plus grandes vitesses de rotation.

Si l'on s'intéresse aux angles maximums atteints présentés par la figure 5.8, on remarque que tout le monde, excepté le sujet S004 atteint des angles comparables d'environ 80 degrés. Le système de visée des cibles n'étant pas précis, la pondération des efficacités par l'angle maximum de chaque sujet n'apporte pas d'information supplémentaire, sauf pour le sujet S004.

En effet, ce sujet, malgré ses données aberrantes, montre que tout le monde n'est pas en mesure d'atteindre un angle de rotation fonctionnel de 80 degrés. Si l'on part du principe que le système a un fonctionnement acceptable à partir de 40 degrés, une telle personne verra son intervalle de fonctionnement diminuer de 40 à 80 degrés, à 40 à plus ou moins 70 degrés. Le système aura de ce fait une perte d'efficacité pour ces personnes là.

Intéressons nous maintenant au comportement du système en lui-même. Les deux tableaux 5.4 et 5.5 montrent l'efficacité du système sur des intervalles d'angles donnés avec et sans filtrage :

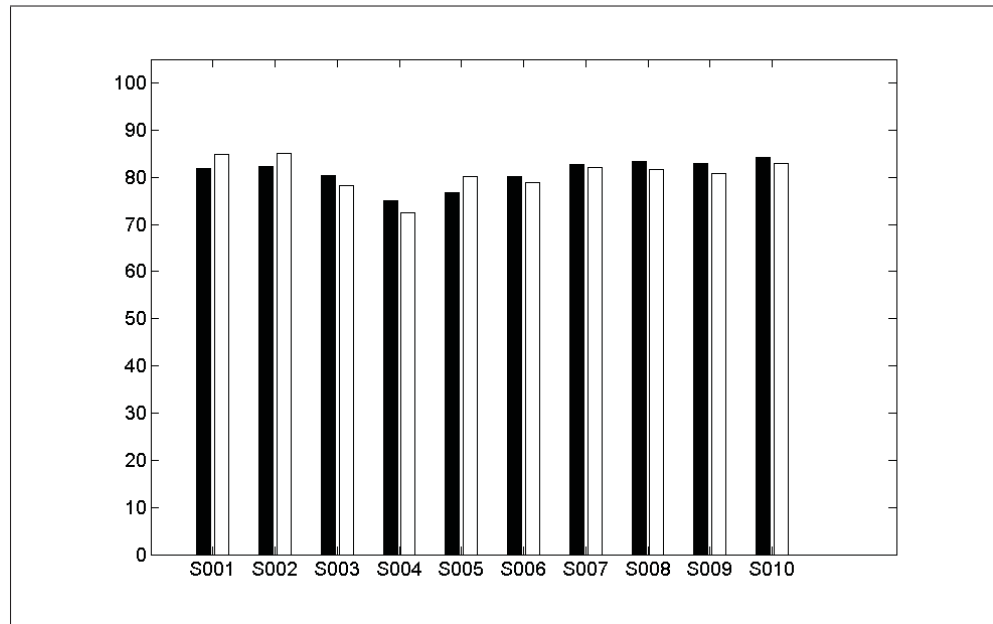


Figure 5.8 Angles maximums atteints par les sujets exprimés en degrés.

Tableau 5.4 Efficacités moyennes par intervalle d'angles obtenues par sujet avec filtrage.

Angle	10° – 80°	20° – 80°	30° – 80°	40° – 80°	50° – 80°	60° – 80°	70° – 80°
S001	0.675	0.7714	0.9	1	1	1	1
S003	0.425	0.4857	0.5667	0.68	0.85	1	1
S005	0.7	0.8	0.9333	1	1	1	1
S006	0.575	0.6571	0.7667	0.92	1	1	1
S007	0.525	0.6	0.7	0.84	1	1	1
S008	0.675	0.7714	0.9	1	1	1	1
S009	0.6	0.6857	0.8	0.96	1	1	1
S010	0.525	0.6	0.7	0.84	1	1	1
Moyenne	0.56	0.64	0.7467	0.868	0.94	0.9733	1
Ecart Type	0.1308	0.1495	0.1744	0.1829	0.1449	0.0843	0

En tenant compte des valeurs aberrantes, le système avec filtrage obtient une efficacité moyenne de 56.2%, contre 49.3% pour le système sans filtrage du signal. Si l'on tient compte des valeurs aberrantes, et que l'on enlève celles-ci du calcul, on obtient alors une efficacité moyenne pour le système de 59.16% pour le système avec filtrage contre 44.44% pour le système non filtré. Ainsi, le filtre apporte un gain d'efficacité de 33.12% aux performances du système global sur tous les angles.

Tableau 5.5 Efficacités moyennes par intervalles angles obtenues par sujet sans filtrage.

Angle	10° – 80°	20° – 80°	30° – 80°	40° – 80°	50° – 80°	60° – 80°	70° – 80°
S001	0.3	0.3429	0.4	0.48	0.6	0.8	1
S003	0.35	0.4	0.4667	0.56	0.7	0.9333	1
S005	0.5	0.5714	0.6667	0.8	1	1	1
S006	0.475	0.5429	0.6333	0.76	0.95	1	1
S007	0.45	0.5143	0.6	0.72	0.9	1	1
S008	0.5	0.5714	0.6667	0.8	1	1	1
S009	0.5	0.5714	0.6667	0.8	1	1	1
S010	0.45	0.5143	0.6	0.72	0.9	1	1
Moyenne	0.4925	0.5571	0.6333	0.74	0.9	0.9733	1
Ecart Type	0.1663	0.1737	0.1571	0.1414	0.1394	0.0644	0

En ne prenant en compte que les angles allant de 30 à 80 degrés, l'efficacité mesurée du système filtré est de 74.67% contre 63.33% pour le système non filtré en tenant compte des valeurs aberrantes. Si on retire les données aberrantes, on obtient une efficacité de 78.89% pour le système filtré et de 58.75%, soit une amélioration de 34.28% grâce au filtrage.

Si l'on ne prend en compte que les angles allant de 40 à 80 degrés, l'efficacité du système passe à 86.8% pour le système filtré contre 74% pour le système non filtré en tenant compte des données aberrantes. Si l'on retire les données aberrantes, on obtient une efficacité de 91.55% pour le système avec filtrage et de 62.22% pour le système sans filtrage. Le filtre apporte ainsi 47.14% d'efficacité supplémentaire au système sur l'intervalle de 40 à 80 degrés.

Ainsi, on effectue les mêmes calculs pour les intervalles 30-80 degrés, 50-80 degrés, 60-80 degrés et 70-80 degrés. Ces données sont réunies dans le tableau 5-1 qui permet de visualiser facilement les performances globales du système et les performances apportées par le filtrage du signal.

L'influence des angles maximums et des vitesses maximum doit également être prise en compte afin de venir pondérer les efficacités des sujets. Cette pondération est assez mince en ce qui concerne les angles, mais elle est très influente en ce qui concerne les vitesses angulaires. En effet, certains participants ont atteint des vitesses presque deux fois plus grandes que d'autres,

Tableau-A 5-1 Tableau récapitulatif des performances du système et du filtrage du signal (N=8)

Intervalle (degrés)	10-80	30-80	40-80	50-80	60-80	70-80
Eff. avec filtre	59.16%	78.89%	91.55%	98.33%	100%	100%
Eff. sans filtre	44.44%	58.75%	62.22%	88.89%	100%	100%
Gain grâce au filtrage	33.12%	34.28%	47.14%	10.62%	0%	0%

sans pour autant avoir de meilleures efficacités. Les tableaux 5.6 et 5.7 montrent les efficacités pondérées obtenues avec et sans filtrage.

Tableau 5.6 Efficacités moyennes par angles obtenues par sujet avec filtrage avec application des coefficients d'angle et de vitesse.

Angle	10°	20°	30°	40°	50°	60°	70°	80°
S001	0	0	0.2225	0.5562	0.5562	0.5562	0.5562	0.5562
S003	0	0	0	0	0.3817	0.9543	0.9543	0.9543
S005	0	0	0.3359	0.5598	0.5598	0.5598	0.5598	0.5598
S006	0	0	0	0.4674	0.7790	0.7790	0.7790	0.7790
S007	0	0	0	0.1965	0.9824	0.9824	0.9824	0.9824
S008	0	0	0.3963	0.9909	0.9909	0.9909	0.9909	0.9909
S009	0	0	0	0.5685	0.7107	0.7107	0.7107	0.7107
S010	0	0	0	0.1328	0.6640	0.6640	0.6640	0.6640
Moyenne	0	0	0.1193	0.434	0.7031	0.7747	0.7747	0.7747
Ecart Type	0	0	0.1713	0.3152	0.2116	0.1822	0.1822	0.1822

On exploite également ces données pour déduire l'efficacité du filtrage. En supposant que la répartition des sujets était normale, on est capable d'effectuer une analyse de variance pour chaque angle entre l'efficacité globale avec filtrage et celle sans filtrage. Les résultats obtenus sont présentés dans le tableau 5.8.

On constate que malgré un nombre d'échantillons très petit (N=8), on obtient un pouvoir statistique de 0.0155 pour la détection de l'angle de 40 degrés. Ceci pouvant être facilement expliqué que cet angle est très bien détecté avec le filtrage du signal, et presque jamais détecté sans le filtrage. Cet élément vient lui aussi renforcer l'importance du filtrage du signal.

Tableau 5.7 Efficacités moyennes par angles obtenues par sujet sans filtrage avec application des coefficients d'angle et de vitesse.

Angle	10°	20°	30°	40°	50°	60°	70°	80°
S001	0	0	0.2258	0.5646	0.5646	0.5646	0.5646	0.5646
S003	0	0	0	0	0.3875	0.9687	0.9687	0.9687
S005	0	0	0	0	0.7148	0.7148	0.7148	0.7148
S006	0	0	0	0	0.6749	0.8437	0.8437	0.8437
S007	0	0	0	0	0.5929	0.9882	0.9882	0.9882
S008	0	0	0	0	0.9848	0.9848	0.9848	0.9848
S009	0	0	0	0	0.6735	0.6735	0.6735	0.6735
S010	0	0	0	0	0.3468	0.5780	0.5780	0.5780
Moyenne	0	0	0.0282	0.0706	0.6175	0.7895	0.7895	0.7895
Ecart Type	0	0	0.0798	0.1996	0.2002	0.1801	0.1801	0.1801

Tableau 5.8 Analyse de variance sur les angles en comparant les efficacités du système avec filtrage et du système sans filtrage. Ici p représente le pouvoir statistique.

Angle	10°	20°	30°	40°	50°	60°	70°	80°
p	-	-	0.1943	0.0155	0.4198	0.8720	0.8720	0.8720

Enfin un dernier élément d'interprétation des données consiste en une carte de détection personnalisée par sujet et par test (avec ou sans filtre par exemple). Cette carte se présente comme sur la figure 5.9.

Grâce à ce type de visualisation, il est très facile d'observer rapidement si un sujet a eu un fort taux de détection ou non, et si il a atteint de grandes vitesses de rotation de la tête en fonction de la largeur de cette surface.

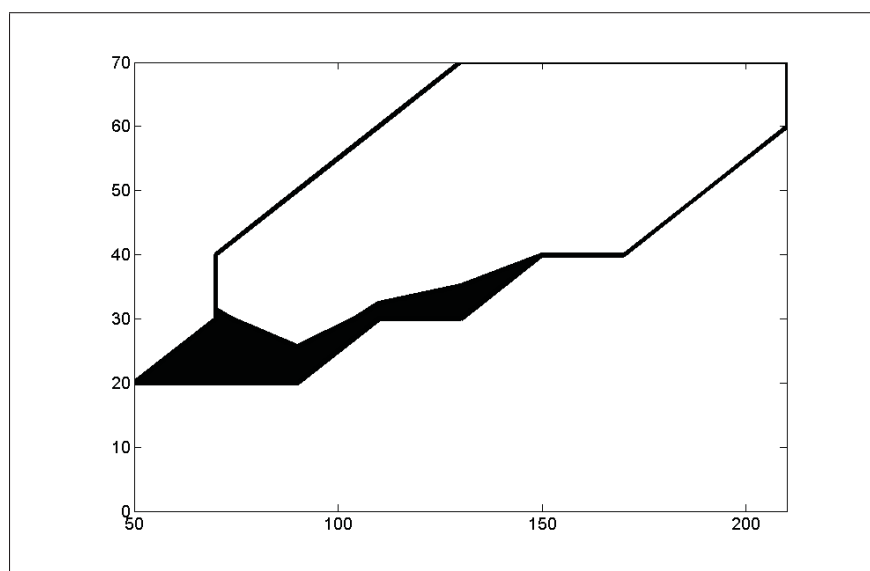


Figure 5.9 Carte de détection du sujet S005 pour le système avec filtrage. En abscisses sont représentées les vitesses angulaires exprimées en degrés par seconde et en ordonnées les angles exprimés en degrés. La zone blanche constitue la zone où une activité musculaire a été détectée et la zone noire une zone où aucune activité musculaire n'a été détectée.

CHAPITRE 6

DISCUSSION

Les résultats ayant été présentés et analysés, ceux-ci vont être étudiés dans le but de justifier le comportement du système dans le but de fixer quelles parties sont vraiment efficaces et quelles parties nécessiteront une amélioration future. Une comparaison à la littérature sera également effectuée afin de cibler les progrès apportés par le projet.

6.1 Performances globales du système

Dans un premier temps, et sans quantifier les performances du système, celui-ci s'est montré capable, lorsque l'utilisateur exécute de grands mouvements impliquant une part de contraction isométrique de commander le bras Jaco à partir de mouvements de l'utilisateur. Ce fait a été validé de manière expérimentale et une trace vidéo du contrôle du bras avec les quatre électrodes a été conservée. Toutefois, il a également été constaté expérimentalement que la sensibilité du système sans filtrage ne permet pas de détecter les mouvements de petite amplitude.

Ainsi, la preuve de concept est validée quand à son architecture générique. Il est possible d'effectuer une commande sur un bras robotique grâce à plusieurs signaux EMG acquis et traités en temps réel.

L'ajout du filtrage adaptatif du signal a permis d'augmenter en moyenne de 33% la sensibilité globale du système comme le montre le tableau 5-1. Cependant, celui-ci est extrêmement demandeur en temps de calculs, et ne peut être implémenté sur les quatre canaux en simultané comme va le présenter la partie suivante.

Ce sujet sera traité dans la partie suivante de la discussion. Quant aux performances du système elles seront abordées dans la dernière partie de ce mémoire.

Une autre question apparaît au regard du fonctionnement global du système. Lors des tests sans filtrage réalisés sur quatre électrodes en simultané, chaque mouvement de l'utilisateur a été relié à une action du bras Jaco. Les rotations de la tête étaient reliées à un déplacement sur

l'axe X : négatif lorsque la tête est tournée à gauche et positif lorsqu'elle est tournée à droite. Les élévations des épaules ont été reliées à une action sur l'axe Y : positive pour une élévation de l'épaule droite et négative lors d'une élévation de l'épaule gauche.

Lors de l'utilisation du système, on voit clairement que lorsque l'utilisateur tourne la tête à gauche ou à droite pour déplacer le bras, celui-ci sort du champ de vision de l'utilisateur qui ne peut alors plus le contrôler efficacement dans le but d'une utilisation dans la vie courante, pour attraper un objet par exemple.

Une réflexion est alors à faire sur le type de contrôle qui sera réalisé par le système sur le bras robotique. Cette action peut être un changement de mode de déplacement, allumer et éteindre le bras ou encore ouvrir ou fermer les doigts.

Si l'on compare ces résultats avec la littérature, ce qui apparaît est l'application réelle et concrète du concept de base : un bras robotique a vraiment été contrôlé par quatre signaux EMG. En effet, la littérature présente des projets concrets de contrôle de prothèses ou d'interfaces homme-machine, mais les articles présentent comme finalité des résultats de taux de classification de mouvement sans préciser si ils ont servi ou non à effectuer un contrôle réel sur le périphérique prévu.

Dans le cas de ce projet, les images vidéos collectées attestent du contrôle réel ayant eu lieu sur le bras grâce au système.

6.2 Chronogrammes et temps de calculs

L'analyse des temps de calcul du système permet de tirer plusieurs conclusions de la réalisation, et du matériel choisi.

Les temps de calculs relevés y sont présentés à l'intérieur de la boucle principale. La figure 5.4 montre la fréquence d'arrivée d'une nouvelle fenêtre de signaux EMG pour les quatre canaux. Elle permet de visualiser le temps de calcul disponible avant qu'une nouvelle fenêtre n'arrive.

Afin de rester en temps réel, le système doit être capable d'effectuer tous ses calculs avant qu'une nouvelle fenêtre ne soit acquise.

Le temps nécessaire pour détecter l'activité musculaire sur les quatre canaux est également observé. On observe que ce temps de calcul est largement inférieur au délai de traitement, et peut donc être utilisé en temps réel sans parer au bon fonctionnement du système.

Le troisième chronogramme illustre le temps nécessaire au filtrage d'un seul canal EMG. En considérant qu'il faut multiplier ce temps par 4 pour traiter les quatre canaux EMG, on remarque facilement que le temps nécessaire au traitement est trop long par rapport au délai entre deux fenêtres.

En effet, le délai étant de 8 ms auxquelles il faut retirer 1.23 ms de détection d'activité, pour un délai disponible de 6.77 ms. Il faut 14.32 ms pour filtrer les quatre canaux EMG. Ce temps est indisponible, et rend l'utilisation du filtre sur les quatre électrodes impossible.

Cette constatation permet d'établir le fait que le PIC32 est suffisamment performant pour effectuer une détection d'activité performante, mais n'est pas assez rapide pour effectuer un filtrage prédictif des quatre canaux EMG si on conserve les spécifications du projet telles qu'elles. Il est nécessaire de souligner toute fois que l'algorithme de filtrage est très complexe et nécessite l'utilisation des transformées de Fourier.

Deux solutions se présentent pour résoudre un tel problème. La première consiste à changer de microcontrôleur pour s'orienter vers une architecture pour grosse, mais plus coûteuse. La seconde est de diminuer les spécifications du projet, de plusieurs manières possibles, en baissant par exemple la fréquence d'échantillonnage du système. Si la fréquence d'échantillonnage est plus basse, plus de temps est disponible avant la prochaine fenêtre afin de faire le traitement du signal. Le signal EMG ayant une bande utile allant jusqu'à 450Hz, la fréquence d'échantillonnage théorique la plus basse est donc de 900Hz. Une fréquence d'échantillonnage de 900Hz laisserait au microcontrôleur 17.8 ms avant l'acquisition de la prochaine fenêtre. Cette possibilité n'a pu être testée expérimentalement par manque de temps dans le projet.

Cependant, une baisse de la fréquence d'échantillonnage entraînerait une perte d'information et un bruit de recouvrement qui pourrait poser un problème dans l'hypothèse où d'autres caractéristiques du signal EMG seraient utilisées dans la suite du projet.

La résolution du problème consiste donc à faire un choix important pour la suite du projet. Changer le microcontrôleur permettra d'augmenter les performances du système drastiquement tout en gardant les spécifications de base du système. Baisser les spécifications du projet permettra de faire fonctionner le système intégralement, mais ne permettra pas de le faire évoluer par la suite.

Étudions maintenant la latence générale du système. La latence induite par le calcul temps réel exécuté sur le PIC32 est quantifiable précisément. En effet, celle-ci correspond au délai de traitement des quatre canaux : 4.81 ms, auquel on ajoute le délai induit par le vote à la majorité vu à la partie 4.3.2.3 qui est de 128 ms. Ceci donne une latence totale de 133 ms environ. Elle reste donc dans le critère fixé par la littérature de 300 ms (Englehart et Hudgins, 2003). Cependant il est nécessaire de préciser que plus cette latence est faible, plus sa perceptibilité sera diminuée. Ainsi, un des objectifs futur du projet sera de diminuer la latence au maximum. La latence minimale sera fixée par l'architecture et les spécifications choisies du projet.

Aujourd'hui, le système étant au stade de prototype, il inclut un ordinateur qui fait le lien pour transmettre les commandes au bras robotique Jaco. Les étapes de transitions logicielles que sont la liaison série, l'appel des fonctions Kinova via Mono et la transmission de la commande à Jaco par USB induisent une latence variable dépendant de l'ordonnement de l'ordinateur qui fait la liaison et qui n'est pas temps réel. La latence totale peut alors monter aux alentours d'une seconde.

Cependant, ce projet a pour but d'être directement intégré à Jaco. Par conséquent, toutes ces étapes de transitions disparaîtront pour être remplacées par un bus CAN communiquant directement avec Jaco et étant très rapide.

L'objectif de rester en dessous de 300 ms de latence est donc atteint.

6.3 Validation des performances du système

Cette partie va maintenant interpréter les données acquises lors des expérimentations sur sujets humains effectuer pour valider les performances du système.

La réalisation de tests sur sujets humains a permis de mettre en évidence les facteurs clés augmentant la sensibilité du système, à savoir l'amplitude du mouvement et le placement des électrodes. Toute fois, une étude de 10 sujets est insuffisante pour déterminer précisément l'importance de la vitesse du mouvement sur la détection d'activité. En effet, on a pu voir que des sujets allant beaucoup plus vite que d'autres ont obtenus des ratios de détection bien plus faibles.

On ne peut cependant pas en tirer la conclusion que la vitesse du mouvement n'a aucune influence sur l'efficacité du système. En effet, on a vu sur certains sujets, surtout au niveau des angles les plus faibles (30 et 40 degrés) qu'ils ne sont pas détectés à faible vitesse mais le sont à grande vitesse.

On a également pu constater lors des expérimentations que mêmes des sujets saints ont une physiologie différentes qui ne leur permet pas de développer le même tonus musculaire. Il est alors normal de se demander quels seront les résultats du système sur des personnes n'ayant qu'une activité résiduelle de leurs muscles. Ces réponses devront être explorées par des tests supplémentaires.

Les tests sur sujets humains ont également permis de prouver l'efficacité du filtrage prédictif sur le signal EMG. En effet, le taux de détection moyen a été amélioré d'environ 33% entre les tests avec filtrage et les tests sans. Ceci a également été confirmé par une brève analyse de variance ayant déduit une puissance statistique de 0.0155 pour un nombre d'échantillons très réduit de $N=8$.

L'utilisation de ce type de filtrage sur le signal EMG est une première dans la littérature actuelle. En effet, ce type de filtrage a été développé pour une application sur un signal sonore dans le domaine de la reconnaissance vocale mais jamais pour un signal EMG.

Enfin, un dernier point de comparaison à aborder est la performance du système en elle-même comparée aux autres systèmes présentés dans la littérature.

Cette comparaison est très difficile pour une raison principale : tous les systèmes de la littérature présentent des résultats de classification de mouvements à partir de signaux EMG. Tous ces systèmes présentent donc des pourcentages de classifications effectués sur plusieurs sujets humains. Ainsi, pour donner un exemple générique : Tenore *et al.* (2007) montrent qu'ils ont un taux de détection de mouvements général de 79.2% sur leur plateforme embarquée qui doit détecter 9 états différents : 8 mouvements et le repos. Toute fois, aucune études sur plusieurs sujets ne semble avoir été menée.

Chang *et al.* (1996) obtiennent un taux de reconnaissance de mouvement de 95% sur une études de 6 sujets sains.

Ces deux articles et la plupart de la littérature concernant la classification EMG s'intéressent à des performances de classificateurs discriminant plusieurs types de mouvements. Cependant, aucun de ces articles ne caractérise la manière d'effectuer ce mouvement. En effet, notre projet a permis de mettre clairement en évidence l'importance du mouvement effectué. Le signal EMG d'un muscle lors d'un mouvement de petite amplitude comme réalisé dans la vie courante n'a rien à voir avec celui d'un mouvement ample, effectué spécialement pour l'occasion, et ayant une composante de contraction isométrique importante.

Ainsi, il est impossible de savoir si les systèmes de classification présentés conserveraient leur pourcentage de reconnaissance de mouvement lors de petits mouvements effectués à faible vitesse.

En quelques sorte, l'étude réalisée au cours de ce projet, est une étude de sensibilité de la détection de mouvement grâce à l'EMG du muscle sterno-cléïdo mastoïdien en fonction de l'amplitude et de la vitesse du mouvement effectué, alors que la littérature présente une étude de la classification de mouvement sans préciser le type de mouvement précis réalisé pour la classification.

En cela, cette étude contribue à l'augmentation de la connaissance dans la détection de mouvement à partir d'EMG et la reconnaissance de mouvements à partir de celle-ci.

CONCLUSION

La réalisation du projet a permis d'établir qu'un système de contrôle hybride d'une aide technique complexe est envisageable en utilisant le signal EMG en respectant une contrainte temps réelle symbolisée par une latence maximum de 300ms, et sur un dispositif embarqué. La preuve de concept faisant l'objet initial du projet avec Kinova a donc été validée.

La réalisation du système a également permis de fixer les outils mathématiques permettant de réaliser ce contrôle. L'utilisation du TKE s'est avérée fructueuse, et le filtrage adaptatif a montré qu'on pouvait améliorer drastiquement la qualité d'un signal EMG.

Le système a également pu être testé sur une dizaine de personnes afin de valider la cohérence de fonctionnement, ainsi que son intervalle de son fonctionnement et sa sensibilité.

Les mesures de temps de calculs ont permis de montrer que le PIC32 est trop lent pour effectuer les calculs les plus complexes relatifs au filtrage tout en conservant les spécifications requises.

Toutefois, les tests sur sujets réels ont permis de montrer l'efficacité du filtrage du signal en quantifiant son gain de performance. La preuve de concept qui faisait l'objectif du projet a donc été atteinte. Toutefois, le projet en lui même n'est qu'en partie atteint. En effet, l'architecture choisie basée sur le PIC32 ne permet pas d'effectuer un filtrage simultané sur les quatre canaux EMG. Toute fois, le système est viable si développé sur une architecture plus performante, et pourra donc être réalisé dans la suite du projet.

Le projet a également permis d'aborder la classification de mouvement à partir de signaux EMG sous un angle qui n'a pas été mentionné dans la littérature, à savoir la qualité de la reconnaissance en fonction du type de mouvement effectué : petite ou grande vitesse, et petite ou grande amplitude.

CONTRIBUTIONS

Contributions apportées par le projet :

- Réalisation d'un prototype d'architecture matérielle électronique et informatique pour l'interfaçage de quatre électrodes EMG à un microcontrôleur PIC32 de la société Microchip.
- Réalisation du programme du microcontrôleur permettant la détection de mouvements effectués par l'utilisateur à partir de ses signaux EMG en temps réel.
- Mise en place du filtrage adaptatif du signal dans le programme du microcontrôleur.
- Réalisation d'un protocole de communication série entre la carte électronique et l'ordinateur distant permettant la visualisation des signaux.
- Réalisation de l'interface graphique en C++/Qt permettant la visualisation des signaux et le calibrage de la carte électronique.
- Réalisation grâce à Mono de l'utilisation de la librairie de programmation C# fournie par Kinova en C++.
- Réalisation d'une stratégie de commande permettant de faire bouger le bras robotique Jaco à partir d'une détection d'activité musculaire.
- Réalisation d'un protocole expérimental permettant la validation sur plusieurs sujets humains du système global.
- Réalisation des outils Matlab permettant l'analyse des données extraites des expérimentations.
- Participation au quinzième congrès annuel des étudiants et stagiaires du centre de recherche du CHUM.
- Participation et troisième prix des présentations de projets de recherches des étudiants de l'ETS lors des journées de la découvertes 2013.

ANNEXE I

CONCEPTION DU MONTAGE POUR LA VALIDATION DU SYSTÈME

Le montage général est présenté ici par la figure I-1.

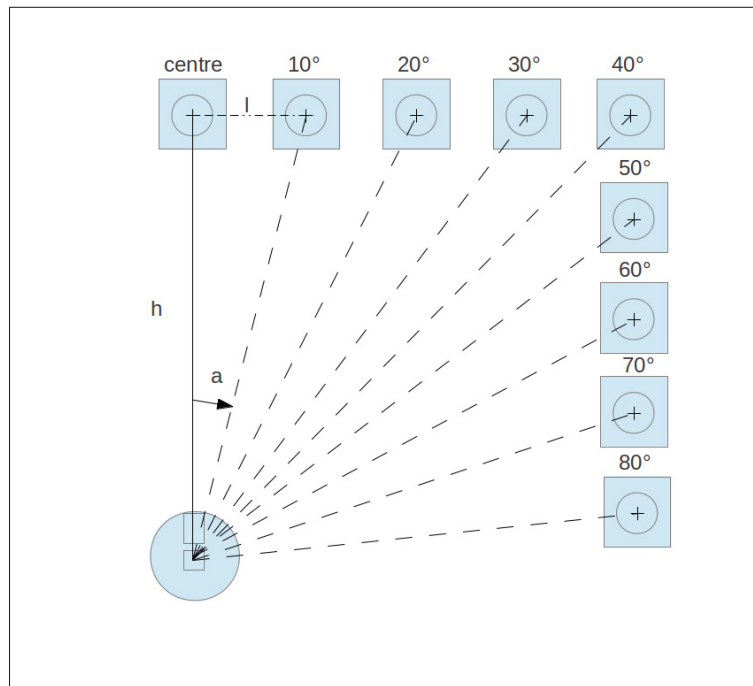


Figure-A I-1 Montage réalisé pour la validation du système. h représente la longueur entre la tête du sujet et la cible du centre, l représente la longueur entre la cible du centre et une autre cible et a représente l'angle de rotation de la tête du sujet.

Cette annexe a pour but de décrire la procédure de construction de l'expérience dans un but de reproductibilité.

La première étape est de fixer la position du sujet. Pour cela, on place des marqueurs au sol pour fixer la position de la chaise au sol. La personne doit être assise au fond de la chaise, contre le dossier, pour que la tête soit environ toujours à la même distance de la cible centrale.

Afin de placer correctement les cibles désignant les angles, on mesure la distance entre la tête du sujet et la cible centrale. Puis connaissant l'angle cible, on trouve la distance l grâce à la

relation décrite par l'équation A I-1.

$$l = h * \tan(a * \frac{\pi}{180}) \quad (\text{A I-1})$$

Ainsi on est en mesure de placer correctement chaque cible sur le mur grâce à une relation trigonométrique simple.

BIBLIOGRAPHIE

- Apparelyzed. 2008. « Types of Paralysis - Quadriplegia (Tetraplegia) and Paraplegia ». <<http://www.apparelyzed.com/paralysis.html>>.
- Berichard. 2008. « Muscle sterno-cleido mastoïdien. Anatomie humaine. Travail personnel d'après Gray's Anatomy ». <http://commons.wikimedia.org/wiki/File:Muscle_sterno-cléido_mastoïdien.png>.
- Chang, Gwo-Ching, Wen-Juh Kang, Jer-Junn Luh, Cheng-Kung Cheng, Jin-Shin Lai, Jia-Jin J Chen, et Te-Son Kuo. 1996. « Real-time implementation of electromyogram pattern recognition as a control command of man-machine interface ». *Medical engineering & physics*, vol. 18, n° 7, p. 529–537.
- Chu, Jun-Uk, Inhyuk Moon, Yun-Jung Lee, Shin-Ki Kim, et Mu-Seong Mun. 2007. « A supervised feature-projection-based real-time EMG pattern recognition for multifunction myoelectric hand control ». *IEEE/ASME Transactions on Mechatronics*, vol. 12, n° 3, p. 282–290.
- Crawford, Beau, Kai Miller, Pradeep Shenoy, et Rajesh Rao. 2005. « Real-time classification of electromyographic signals for robotic control ». In *Proceedings of the National Conference on Artificial Intelligence*. p. 523. Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ;.
- Englehart, Kevin et Bernard Hudgins. 2003. « A robust, real-time control scheme for multifunction myoelectric control ». *IEEE Transactions on Biomedical Engineering*, vol. 50, n° 7, p. 848–854.
- Englehart, Kevin, Bernard Hudgins, Philip A Parker, et Maryhelen Stevenson. 1999. « Classification of the myoelectric signal using time-frequency based representations ». *Medical engineering & physics*, vol. 21, n° 6, p. 431–438.
- Fukuda, Osamu, Toshio Tsuji, Makoto Kaneko, et Akira Otsuka. 2003. « A human-assisting manipulator teleoperated by EMG signals and arm motions ». *IEEE Transactions on Robotics and Automation*, vol. 19, n° 2, p. 210–222.
- Kinova. 2012. « Kinova corporate internet website ». <<http://kinovarobotics.com/>>.
- Li, Xiaoyan, Ping Zhou, et Alexander S Aruin. 2007. « Teager–Kaiser energy operation of surface EMG improves muscle activity onset detection ». *Annals of biomedical engineering*, vol. 35, n° 9, p. 1532–1538.
- Maheu, V. 2011. « Développement des critères d'apprentissage pour le contrôle d'un bras robot manipulateur à 7 DDL par le traitement des signaux EMG chez les blessés médullaires. ». Maîtrise en génie des technologies de la santé, Montréal, École de Technologie Supérieure, 133 p.

- Martinez, R, R Munoz, L Leija, PR Hernandez, et Ja Alvarez. 1999. « A control unit for myoelectrically controlled prosthesis ». In *[Engineering in Medicine and Biology. 21st Annual Conf. and the 1999 Annual Fall Meeting of the Biomedical Engineering Soc.] BMES/EMBS Conference. Proceedings of the First Joint.* p. 641–vol. IEEE.
- Maxim Integrated Products Inc, a. 2007. « Documentation des composants Max2222/Max3232/Max3237/Max3241 ». <<http://datasheets.maximintegrated.com/en/ds/MAX3222-MAX3241.pdf>>.
- Peleg, Dori, Eyal Braiman, Elad Yom-Tov, et Gideon F Inbar. 2002. « Classification of finger activation for use in a robotic prosthesis arm ». *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, n° 4, p. 290–293.
- Plapous, Cyril, Claude Marro, et Pascal Scalart. 2006. « Improved signal-to-noise ratio estimation for speech enhancement ». *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, n° 6, p. 2098–2108.
- Prasad, G, S Srinivasan, et KM Patil. 1996. « A new DSP-based multichannel EMG acquisition and analysis system ». *Computers and biomedical research*, vol. 29, n° 5, p. 395–406.
- Tenore, Francesco, Ander Ramos, Amir Fahmy, Soumyadipta Acharya, Ralph Etienne-Cummings, et Nitish V Thakor. 2007. « Towards the control of individual fingers of a prosthetic hand using surface EMG signals ». In *Engineering in Medicine and Biology Society. EMBS. 29th Annual International Conference of the IEEE.* p. 6145–6148. IEEE.
- Vsion, translation by Medium69. 2005. « Les différentes régions de la colonne vertébrale. ». <[http://commons.wikimedia.org/wiki/File:Spinal_column_curvature_\(french\).png](http://commons.wikimedia.org/wiki/File:Spinal_column_curvature_(french).png)>.