



HAL
open science

PériSIM : un simulateur haptique d'apprentissage du geste de l'anesthésie péridurale

Thibault Senac

► **To cite this version:**

Thibault Senac. PériSIM : un simulateur haptique d'apprentissage du geste de l'anesthésie péridurale. Autre. Université de Lyon, 2019. Français. NNT : 2019LYSEC032 . tel-02487738

HAL Id: tel-02487738

<https://theses.hal.science/tel-02487738>

Submitted on 21 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



ÉCOLE
CENTRALE LYON

N°d'ordre NNT : 2019LYSEC32

**THÈSE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de l'École centrale de Lyon**

**Ecole Doctorale N° 160
Électronique Électrotechnique et Automatique**

Spécialité de doctorat : Robotique

Soutenue publiquement le 25/10/2019, par :
Thibault SÉNAC

**PériSIM : un simulateur haptique
d'apprentissage du geste de
l'anesthésie péridurale**

Devant le jury composé de :

N. Franck PLESTAN	Professeur	LS2N	ECNantes	<i>Président</i>
Mme Christine BARTHOD	Professeure	SYMME	USMB	<i>Rapportrice</i>
M. Gérard POISSON	Professeur	PRISME	Université d'Orléans	<i>Rapporteur</i>
Mme Marie-Aude VITRANI	MdC HdR	ISIR	Sorbonne Université	<i>Jury</i>
M. Christian BAUER	Praticien hospitalier	Hospices Civils de Lyon		<i>Invité</i>
M. Florent SIGWALT	Praticien hospitalier	Hospices Civils de Lyon		<i>Invité</i>
M. Laurent KRÄHENBÜHL	DR CNRS	Laboratoire Ampère	ECL	<i>Directeur de thèse</i>
M. Richard MOREAU	MdC	Laboratoire Ampère	INSA-Lyon	<i>Co-directeur de thèse</i>
M. Arnaud LELEVÉ	MdC	Laboratoire Ampère	INSA-Lyon	<i>Co-directeur de thèse</i>

*Il est absurde de chercher un puits,
au hasard, dans l'immensité du
désert. Cependant nous nous mêmes
en marche.*

*Quand nous eûmes marché, des
heures, en silence, la nuit tomba, et
les étoiles commencèrent de
s'éclairer.*

ANTOINE DE SAINT-EXUPÉRY
(Le Petit Prince)

Remerciements

Ce doctorat a été réalisé sur trois ans aux laboratoires Ampère de Lyon (CNRS UMR5005) au sein de l'école doctorale EEA. Je tiens ici à adresser quelques remerciements à celles et ceux qui ont rendu ces travaux possibles et qui les ont facilités.

Merci à mon encadrement, Laurent KRÄHENBÜHL mon directeur de thèse, Arnaud LELEVÉ et Richard MOREAU mes co-encadrants pour avoir rendu cette thèse possible. Merci pour l'accompagnement au cours de ces trois années.

Merci tout particulier à Tanneguy REDARCE pour l'aide et l'accompagnement apporté en particulier sur la période de rédaction du manuscrit et de préparation de la soutenance.

Merci à Xavier BRUN, Minh-Tu PHAM, Wilfrid MARQUIS-FAVRE et Emmanuel BOUTLEUX pour l'aide et les conseils apportés pour la rédaction d'articles et la préparation de la soutenance.

Merci à l'équipe d'automatique de Centrale Lyon et en particulier Catherine MUSY et Emmanuel Boutleux (et tous les autres que j'oublie) pour l'accueil et l'encadrement apporté concernant les TD, TP et BE que j'ai eu la chance de donner à Centrale Lyon. L'encadrement et l'implication proposée par l'ensemble de l'équipe a été une formidable occasion de participer à essayer de proposer un enseignement de qualité.

Un grand merci à Christian BAUER et Florent SIGWALT. Sans leur aide, leurs connaissances et leur implication ce projet ne serait que l'ombre de ce qu'il est aujourd'hui. Le travail réalisé et le temps consacré ont vraiment une valeur inestimable. J'espère que ces travaux pourront donner lieu à des suites.

Merci à Hugues DORGERONS, Pascal BEVILACQUA et Quentin ROUBY pour l'aide apportée dans la conception de diverses parties du simulateur.

Un merci tout particulier à ma famille, mes parents et grands parents pour leur accompagnement et à ma sœur pour m'avoir montré la voie pour travailler sur des sujets en lien avec la médecine. Cette thèse n'aurait peut-être jamais eu lieu sans cette influence.

Des remerciements teintés d'admiration pour mes Colocataires Charles BARNOUIN et Marine LECLERC durant ces trois ans. Sans cette colocation, cette période aurait été bien moins agréable. Merci de m'avoir supporté dans la bonne humeur !

Merci aussi à mes amis de longue date, Kim, Robin et Benjamin de toujours être disponibles pour toutes sortes d'aventures.

Merci à l'équipe des doctorants et stagiaires pour avoir maintenu une atmosphère de travail et de vie d'une grande qualité. Merci à Jérémie pour son talent au volleyball, à Vincent pour les mots fléchés, à David d'avoir été un si bon collègue de bureau, à Kévin (continu) d'avoir maintenu une cohésion parmi tous ces doctorants et pour s'être occupé d'organiser beaucoup trop de sorties, à Aboubacar d'avoir supporté tous les surnoms qu'on a pu lui proposer. Merci aussi à Setareh et Elliot de nous avoir montré la voie en soutenant les premiers. Merci aussi à tous les autres : Iris, Kévin (discret), Tahereh, Arnaud, Laurent, Tanguy pour qui je n'ai pas réussi à trouver de petit mot mais qui méritent bien quelques remerciements.

Enfin, merci à In Flames, Carpenter Brut, Monstercat, Aphex Twin, Vald, Nekfeu, Orelsan, Damso et j'en passe, pour la playlist à la fois hétéroclite et parfaite qu'ils ont pu me fournir tout au long de ces trois années.

Table des matières

Introduction	3
1 Description du geste de l'anesthésie péridurale et présentation des objectifs de la thèse	5
1.1 Contexte médical : l'anesthésie péridurale, un geste commun mais complexe	6
1.1.1 Bases anatomiques	6
1.1.2 Description du geste	7
1.2 Enjeux et état de l'art de la formation au geste	8
1.2.1 Enjeux et difficultés	8
1.2.2 Moyens de formation existants	9
1.3 Contexte scientifique : l'haptique	11
1.3.1 De l'intérêt de l'haptique	11
1.3.2 État de l'art de l'haptique et contraintes	13
1.4 Objectifs du doctorat	16
1.4.1 Apports concernant l'haptique	16
1.4.2 Apports concernant l'étude du geste	16
2 Conception du simulateur	19
2.1 Caractéristiques du simulateur final : cahier des charges	20
2.1.1 Caractéristiques principales	20
2.1.2 Rendus haptiques nécessaires au simulateur	21
2.1.3 Vers une évaluation automatique et objective du geste : EF ₅	23
2.1.4 Résumé du cahier des charges	23
2.2 Choix de conception	23
2.2.1 Représentation de l'insertion de l'aiguille	24
2.2.2 Représentation de la seringue et simulation de la perte de résistance	26
2.2.3 Vers une meilleure représentation du patient	28
2.2.4 Génération d'archétypes de patients	29
3 Reproduire la sensation de perte de résistance : contrôle du vérin pneumatique	33
3.1 Modélisation du vérin : la transformée A-T	34
3.2 Contrôle du vérin	36
3.2.1 Contrôle en position	37

3.2.2	Contrôle en effort	39
3.2.3	Algorithme de sélection de la loi de commande	41
3.3	Implémentation et essais	49
3.3.1	Implémentation sur banc d'essai dédié	49
3.3.2	Résultats	50
4	Reproduire l'insertion de l'aiguille à l'aide d'un bras haptique	55
4.1	Architecture de l'application et modélisation de l'espace intervertébral et de la pointe d'aiguille	56
4.1.1	Architecture logicielle et fonctionnements basiques	56
4.1.2	Représentation de l'espace intervertébral	59
4.1.3	Représentation de l'aiguille	61
4.2	Détails d'implémentation des rendus haptiques	62
4.2.1	Restriction à une trajectoire rectiligne	62
4.2.2	Résistance des couches physiques	64
4.2.3	Représentation du contact dur	65
4.2.4	Représentation du perçage de la peau	66
4.2.5	Représentation du frottement sec de type <i>stick-slip</i>	67
4.3	Fonctionnalités supplémentaires : vers un simulateur idéal	71
4.3.1	Paramétrages du modèle : représentation simplifiée de différents types de patients	71
4.3.2	Ajout d'une visualisation IRM	73
4.3.3	Acquisition des données de mesures	73
4.3.4	Design de l'interface utilisateur	74
4.3.5	Synchronisation du Virtuose et du vérin pneumatique	74
4.4	Exemple commenté	76
5	Essais, résultats et étude du geste	79
5.1	Essais : protocole et données brutes	80
5.1.1	Protocole expérimental	80
5.1.2	Données enregistrées et traitements préalables	81
5.1.3	Vers une analyse du geste	83
5.2	Analyses «naïves»	83
5.2.1	Notations	84
5.2.2	Variables intéressantes	84
5.2.3	Une première méthode de classement	86
5.3	Réductions de dimensions et classification	88
5.3.1	Algorithmes de réductions de dimension utilisés	88
5.3.2	Sélection de variables et pré-traitement des données	93
5.4	Algorithmes de classification proposés	97
5.4.1	Algorithme des k-plus proches voisins (kPPV)	97
5.4.2	Algorithme Machine à Vecteurs de Support (SVM)	98
5.4.3	Réseau de neurones	100
5.5	Méthode de classification et d'évaluation choisies	102
5.6	Discussion	103

Références	110
Annexe A Article présenté à la conférence IFAC 2017 à Toulouse	117
Annexe B Résultats d’essais supplémentaires utilisant l’algorithme de contrôle de vérin pneumatique	123
Annexe C Questionnaire soumis aux novices après l’utilisation du simulateur	129
Annexe D Questionnaire soumis aux experts après l’utilisation du simulateur	131

Table des figures

1.1	Plans de coupe du corps humain	6
1.2	Schéma en coupe sagittale de l'espace intervertébral présentant la position souhaitée de l'aiguille	7
1.3	Photographie de seringues de perte de résistance ou seringues <i>LOR</i>	8
1.4	Simulateurs utilisés pour l'apprentissage	10
1.5	Simulateurs haptiques existants	11
1.6	Interfaces haptiques Haption	13
1.7	Exemple d'un système de proxy où l'utilisateur entre en contact avec une surface virtuelle	15
1.8	Exemple de fonctionnement d'un algorithme <i>god-object</i> lorsque l'utilisateur entre en contact avec un objet virtuel dans la simulation.	15
2.1	Interfaces haptiques du commerce	25
2.2	Vérin pneumatique instrumenté	28
2.3	Aperçu du dispositif mécanique représentant le dos du patient	28
2.4	Détail de la partie représentant le point d'insertion de l'aiguille	29
2.5	Simulateur complet	30
3.1	Schéma du vérin et des variables utilisées	34
3.2	Schéma bloc de la loi de commande en position	37
3.3	Schéma bloc de la loi de commande en effort	39
3.4	Diagramme d'état simplifié de l'algorithme gouvernant le choix de la loi de commande à utiliser	42
3.5	Schéma bloc de l'architecture du contrôle du vérin pneumatique	42
3.6	Test de la condition de sécurité de l'algorithme de contrôle du vérin pneumatique	44
3.7	Influence du temps t_{del} sur le fonctionnement de l'algorithme de contrôle du vérin pneumatique	45
3.8	Simulation présentant des cas où aucune restriction n'est mise en place pour limiter le nombre de changements de loi de commande	46
3.9	Logigramme complet de l'algorithme de contrôle du vérin pneumatique	48
3.10	Banc d'essai utilisé pour tester la simulation de la seringue	49
3.11	Caractérisation entrée sortie de la servovalve	51
3.12	Résultats expérimentaux obtenus en simulant un patient «classique»	52
3.13	Résultats expérimentaux présentant un patient «difficile»	52

3.14	Résultats expérimentaux présentant un patient «intermédiaire» et une hésitation de l'utilisateur à $t \approx 35$ s	52
4.1	Architecture logicielle simplifiée de l'application et des interactions principales	56
4.2	Exemple d'un système de proxy où l'utilisateur vient toucher une surface virtuelle	58
4.3	Principe de fonctionnement schématique d'un algorithme de proxy / «god-object»	58
4.4	Repères utilisés dans la simulation	60
4.5	Modèle de l'environnement de simulation 3D utilisé pour représenter l'espace intervertébral	60
4.6	Modélisation du derme, ligaments, espace péri-dural et dure mère	62
4.7	Graphe d'héritage de la classe <code>CNeedleTip</code>	62
4.8	Positionnement du repère avatar par rapport au repère outil	63
4.9	Graphe d'héritage de la classe <code>CSoftTrajColSpace</code>	64
4.10	Graphe d'héritage de la classe <code>CResistantBoxSpace</code>	64
4.11	Graphe d'héritage de la classe <code>CShakyBox</code>	66
4.12	Trajectoire en effort aidant à la représentation du contact dur [23]	66
4.13	Graphe d'héritage de la classe <code>CSoftPlane</code>	67
4.14	Mécanisme servant à l'étude du phénomène <i>stick-slip</i>	68
4.15	Graphe d'héritage de la classe <code>CFrictionBoxSpace</code>	69
4.16	Mécanisme équivalent pour représenter le phénomène de <i>stick-slip</i> dans notre simulation	70
4.17	Algorithme de choix entre une phase de glissement et une phase d'adhérence et efforts associés	71
4.18	Images IRM prises dans le plan sagittal utilisées dans l'application	73
4.19	Fenêtres de l'application contrôlant l'interface haptique électrique	75
4.20	Architecture complète de contrôle du simulateur	76
5.1	Réponses obtenues des utilisateurs concernant la facilité d'utilisation du simulateur et son intérêt en tant qu'outil de formation d'après l'ensemble des utilisateurs	81
5.2	Variables présentant une différence notable entre novices et experts	85
5.3	Échantillon d'arbres hiérarchiques obtenus d'après les mesures	87
5.4	Taux de bons classements correspondant aux arbres de classification présentés en Figure 5.3	89
5.5	Visualisation des données après utilisation d'un algorithme <i>t-SNE</i>	93
5.6	Pourcentage d'inertie contenu dans les différentes composantes résultant de l'ACP sur le jeu de données complet	94
5.7	Contributions des variables de bases aux 5 axes majeurs de l'ACP sur les données initiales	95
5.8	Représentation des mesures après application d'un algorithme d'ACP	96
5.9	Taux de bons classement selon le type de données utilisé et la valeur de k	98
5.10	Classifications obtenues par algorithme des kPPV	99
5.11	Classifications obtenues par algorithme SVM	100

5.12	Aperçu du réseau de neurones utilisés sur les données brutes	101
5.13	Classifications obtenues par réseau de neurones	102
B.1	Vérin instrumenté utilisé pour reproduire le comportement d'une seringue	123

Acronymes

- ACP:** Analyse en Composantes Principale
EC_i: Exigence Contrainte
EF_i: Exigence fonctionnelle
HAS: Haute Autorité de la Santé
kPPV: k-plus proches voisins
LOR: perte de résistance (*Loss Of Resistance*)
ODE: *Open Dynamics Engine*
PWM: *Pulse Width Modulation*
SAMSEI: HStratégies d'Apprentissage des Métiers de Santé en Environnement Immersif
TCT: *Task Completion Time*
t-SNE: *Stochastic Neighbor Embedding*

Nomenclature

$F_{raideur}$	force ressentie lorsque la pointe de l'aiguille transperce un tissu	N
$F_{frottement}$	force due à la friction entre les tissus traversés et le corps de l'aiguille	N
F_{coupe}	force de coupe d'un tissu	N
F_{pneu}	force pneumatique générée par le vérin	N
F_d	force pneumatique souhaitée	N
F_d^{hap}	résistance haptique souhaitée	N
K_{pneu}	raideur pneumatique	N/m
p_P	pression dans la chambre P	Pa
p_N	pression dans la chambre N	Pa
V_0	volume total du vérin	m ³
V_P	volume de la chambre P	m ³
V_N	volume de la chambre N	m ³
S_P	section interne du vérin dans la chambre P	m ²
S_N	section interne du vérin dans la chambre N	m ²
y	position du vérin	m
v	vitesse du piston	m/s
v_d	vitesse souhaitée du piston	m/s
a_d	accélération souhaitée du piston	m/s ²
j_d	jerk souhaité du piston	m/s ³
q_{mP}	débit massique entrant dans la chambre P	kg/s
q_{mN}	débit massique entrant dans la chambre N	kg/s
q_{mA}	débit massique actif	kg/s

q_{mT}	débit massique de pressurisation	kg/s
T	température	K
b	coefficient de frottement visqueux	N.s/m
R	constante des gaz parfaits	J/(kg.K)
M	masse déplacée par la tige du vérin (tige comprise)	kg
k	constante polytropique	kg/s
l	longueur totale de la course du piston	m
C_i	constantes positives des lois de commandes par backstepping	-
K_{int}	constantes positives liée à l'action intégrale	-
K_{cl}	raideur en boucle fermée	N/m
B_{cl}	amortissement en boucle fermée	N/(m.s)
y_s	position pour laquelle la tige du vérin est complètement sortie	m
t_{del}	retard	s

Introduction Générale

Le geste concerné par ce travail est celui de la pose de péridurale. La pose de péridurale, ou anesthésie péridurale est généralement réalisée lors d'un accouchement et a pour objectif de réaliser une anesthésie locale du bas du corps de la patiente. S'il s'agit d'un geste courant et d'apparence simple, il demande un long entraînement afin d'être suffisamment maîtrisé. Cependant du fait du manque d'outils de formation efficaces, l'apprentissage du geste a généralement lieu sur patient. Le futur médecin reçoit au préalable une formation théorique concernant le geste puis il doit le réaliser sous la tutelle d'un anesthésiste expert. Dans ces conditions, il est difficile pour le novice d'apprendre efficacement et sans risque et il est difficile pour le médecin expert de fournir une appréciation de la maîtrise du novice. L'anesthésie péridurale, comme d'autres gestes de la médecine souffre donc d'un manque de moyens de formation efficace.

Dans ces travaux nous allons donc proposer une réponse à ce besoin à travers le développement d'un simulateur haptique dédié. La simulation est un mode d'apprentissage très utilisé lorsqu'il s'agit d'apprendre des procédures complexes. En effet, c'est un formidable moyen d'apprendre, car il permet de se former dans un environnement contrôlé et sans risque. A travers les simulation, les apprenants peuvent répéter une procédure jusqu'à en avoir une maîtrise suffisante pour passer à la pratique. Que ce soit dans la médecine, l'aviation, la course automobile ou autre, la simulation permet de s'entraîner *a priori* afin de limiter les écueils *a posteriori*. Dans ce manuscrit, nous allons détailler le fonctionnement d'un simulateur permettant d'apprendre le geste de la pose de péridurale à travers une simulation haptique.

Depuis plusieurs années, le laboratoire Ampère développe des simulateurs médicaux pour l'apprentissage de gestes dans différentes spécialités. On peut citer entre autre un simulateur d'accouchement pour les sages femmes et les obstétriciens ou un simulateur de chirurgie laparoscopique. Tous ont pour but d'éviter que les futurs médecins apprennent un geste directement sur un patient avec les problématiques que l'on imagine. Ce projet s'inscrit dans la dynamique de créations de simulateurs médicaux du Laboratoire Ampère et a pu être soutenu par la SAMSEI (Stratégies d'Apprentissage des Métiers de Santé en Environnement Immersif) afin de proposer une réponse aux exigences de la Haute Autorité de la Santé.

À l'origine de ces travaux les anesthésistes partenaires de ce projet souhaitaient un simulateur qui puisse reproduire un grand nombre de cas et de morphologies avec précision. Suite à des travaux préliminaires, il a été décidé de concevoir un

simulateur mi-électrique mi-pneumatique afin de tirer profit des divers avantages de chaque actionnement. Ce découpage spécifique a permis des rendus haptiques précis, réalistes et largement personnalisables.

Le premier chapitre pose les éléments de contexte nécessaires à la compréhension du geste de l'anesthésie péridurale et présente une étude de l'existant en matière de simulation de cette procédure. Nous présentons, dans ce chapitre les contributions attendues de ce doctorat afin de démarquer le simulateur proposé des simulateurs existants. Ce sera l'occasion de présenter les notions essentielles qui seront utilisées tout au long de ce manuscrit.

Le deuxième chapitre a pour objectif de dresser le cahier des charges de notre simulateur. L'ensemble des fonctionnalités attendues mais aussi les choix de conceptions préalables y sont détaillés.

Le troisième chapitre aborde le contrôle de la partie pneumatique du simulateur. Il s'agit là de la contribution majeure en termes d'automatique et de contrôle. Le chapitre est concentré sur la synthèse des lois de commandes utilisées ainsi que sur l'étude, de leurs performances sur un banc d'essais haptique.

Le quatrième chapitre détaille le fonctionnement de la partie électrique du simulateur. Dans ce chapitre, nous détaillons les effets haptiques mis en œuvre et adaptés de la littérature. Nous développons aussi en détail le fonctionnement de la simulation haptique mais aussi son implémentation.

Le cinquième et dernier chapitre expose une campagne de tests réalisée fin 2018. Les résultats obtenus y sont détaillés et les premiers retours d'utilisateurs mis en avant. La seconde partie de ce chapitre mène une analyse des données acquises lors de cette première campagne de tests. Ces analyses ont pour but de déboucher sur une méthode d'évaluation objective de la performance concernant l'anesthésie péridurale.

Chapitre 1

Description du geste de l'anesthésie péridurale et présentation des objectifs de la thèse

Introduction

L'anesthésie péridurale est un geste médical consistant à réaliser une anesthésie locale du bas du corps. Elle est pratiquée, la plupart du temps, pour limiter les douleurs lors d'un accouchement. L'anesthésique est administré dans une zone appelée espace péridural, situé entre deux vertèbres lombaires. L'anesthésie péridurale reste un geste médical courant pour un médecin anesthésiste qui peut cependant entraîner des complications si il est mal réalisée. Il peut s'agir de complications légères comme des céphalées (maux de tête) ou des hématomes, voire des complications plus graves pouvant aller jusqu'à la paralysie. Le geste nécessite donc une certaine expertise pour être réalisé correctement, ce qui requiert un long apprentissage.

Ce chapitre présente la procédure et décrit l'anatomie concernée. Nous évoquons aussi la formation à ce geste. La pose de péridurale souffre en effet d'un manque de moyens formation performants que ne peuvent combler les solutions classiques faisant appel à des mannequins par exemple. C'est ce besoin de formation qui motive ce travail de doctorat dont les objectifs seront avancés au cours de ce chapitre.

Une fois les notions médicales détaillées, nous introduirons les termes et notions que nous allons utiliser dans ce manuscrit, concernant notamment les technologies haptiques. L'haptique correspond à la fois au toucher et aux phénomènes kinesthésiques. Nous considérerons tout particulièrement les sensations de type retour d'efforts ainsi que la perception consciente de la position.

1.1 Contexte médical : l'anesthésie péridurale, un geste commun mais complexe

1.1.1 Bases anatomiques

Avant d'expliquer la procédure de l'anesthésie péridurale, présentons rapidement l'anatomie de la partie du dos qui nous intéresse. L'anesthésie péridurale est réalisée en passant entre deux vertèbres lombaires du patient, et plus précisément entre deux apophyses épineuses de vertèbres lombaires. La position des vertèbres lombaires dans la colonne vertébrale et la forme des épineuses sont illustrées sur la Figure 1.1.

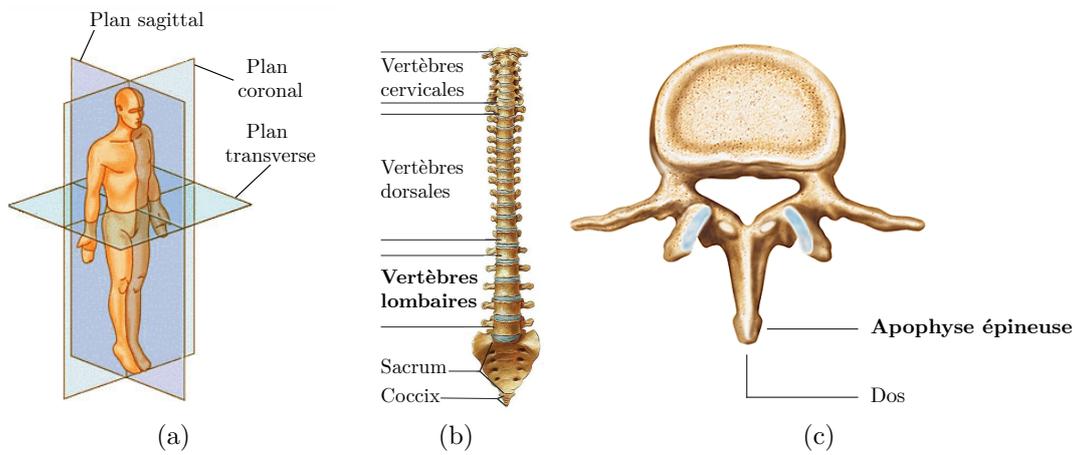


FIGURE 1.1 – (a) : plans de coupe du corps humain (b) : colonne vertébrale selon le plan coronal, (c) : vertèbre lombaire dans le plan transverse

Le fait de passer entre deux lombaires permet d'accéder à l'espace péridural. Cet espace correspond à un espace libre dans la colonne vertébrale qui ne se retrouve que dans la partie lombaire. En injectant un anesthésique dans cet espace le patient reçoit alors une anesthésie locale du bas du corps. Ainsi lorsqu'un anesthésiste souhaite poser une péridurale, l'aiguille utilisée va devoir traverser les couches et ligaments suivants :

- l'épiderme et le derme : il s'agit ici de la peau et peut aussi contenir plus ou moins de graisse selon la morphologie du patient ;
- le ligament supra-spinal est le premier ligament rencontré reliant deux vertèbres successives et situé en surface de ces vertèbres ;
- le ligament interspinal se situe entre les deux épineuses ;
- le ligament jaune est le dernier ligament rencontré avant de parvenir à l'espace péridural ;
- l'espace péridural est l'espace dans lequel on souhaite positionner l'aiguille ;
- la dure mère renferme des terminaisons nerveuses et du liquide céphalo-rachidien susceptible de s'échapper lorsque la dure mère est touchée par l'aiguille. Dans ce cas, des complications peuvent apparaître.

La Figure 1.2 représente l'ensemble de ces éléments en coupe sagittale. Ces éléments se situent sur le trajet normal de l'aiguille lors d'une pose de péridurale. Pour être plus complet, il faut ajouter la présence de muscles de part et d'autre de l'espace intervertébral. Ces muscles se retrouvent donc parallèles au plan de coupe de la Figure 1.2 et ce à partir du ligament supraspinal, jusqu'à la dure mère.

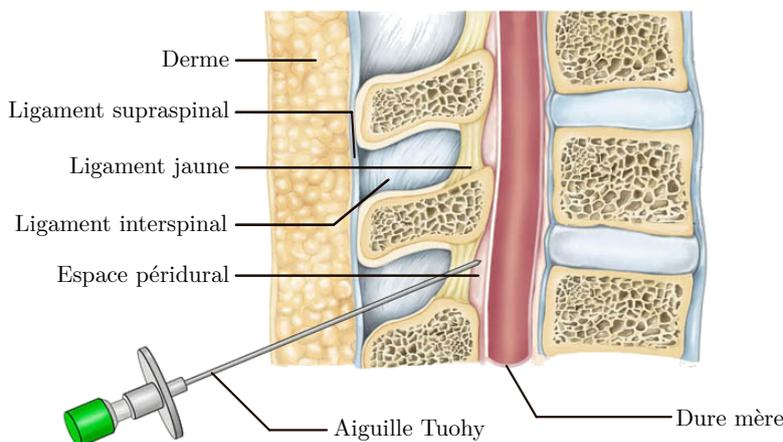


FIGURE 1.2 – Schéma en coupe sagittale de l'espace intervertébral présentant la position souhaitée de l'aiguille

1.1.2 Description du geste

Plus que l'anesthésie en elle-même, c'est le parcours de l'aiguille qui est difficile à réaliser. La procédure complète est réalisée en trois temps. Premièrement le praticien doit définir son point d'insertion par la palpation du dos du patient. Une fois le point d'insertion choisi, le médecin peut alors commencer à insérer l'aiguille en cherchant à atteindre l'espace péridural. Enfin, une fois l'espace atteint il peut alors poser un cathéter et procéder à l'anesthésie.

La difficulté de la procédure réside dans le fait que l'insertion de l'aiguille se fait entièrement à l'aveugle. C'est à dire sans avoir recours à des outils de visualisations tels que des IRM ou des échographies. Il faut donc être en mesure de différencier un contact osseux, un muscle, ou une insertion réussie sans information visuelle simplement par le sens du toucher. Afin de positionner l'aiguille à la profondeur adéquate, l'anesthésiste utilise une seringue de perte de résistance ou seringue *LOR* (*Loss Of Resistance* Figure 1.3) qu'il remplit d'un sérum physiologique. Tout au long de l'insertion de l'aiguille, le praticien injecte le liquide contenu dans la seringue dans le but de détecter une perte de résistance au niveau du piston de la seringue. Cette perte de résistance se produit lorsque la pointe de l'aiguille finit de traverser le ligament jaune et pénètre dans l'espace péridural. À ce moment précis, il n'y a plus de résistance à l'écoulement du liquide, on parle alors de perte de résistance. Il est aussi possible de ressentir une perte de résistance au niveau de l'insertion de l'aiguille. En effet, l'espace péridural étant un espace vide, l'aiguille n'a plus à couper de tissu : il ne reste donc plus que de les frottements imposés par l'ensemble des ligaments sur le corps de l'aiguille. Cependant, cette perte de résistance est bien moins marquée

que sur le piston de la seringue. L'utilisation de la perte de résistance est la méthode la plus couramment utilisée chez les praticiens car elle reste pratique et efficace.

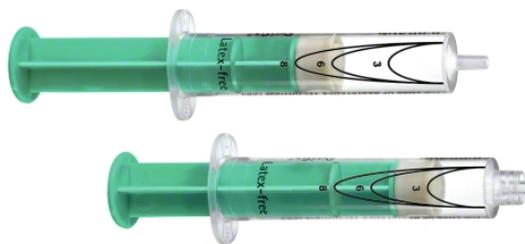


FIGURE 1.3 – Photographie de seringues de perte de résistance ou seringues *LOR*

En pratique, la résistance ressentie, que ce soit la résistance à l'avancée de l'aiguille ou la résistance à l'injection de liquide, augmente sur toute la longueur du trajet pour chuter plus ou moins brusquement lorsque l'aiguille atteint l'espace périuridural. S'il est difficile de quantifier précisément la soudaineté de la perte de résistance, nous chercherons à reproduire une sensation réaliste de l'avis de médecins expérimentés. La résistance ressentie, ainsi que les transitions d'un niveau de résistance à un autre va alors dépendre de la couche physique traversée, mais aussi de la morphologie ou encore de l'âge du patient.

Les praticiens ont recours à deux techniques principales pour poser une périuridurale et repérer la perte de résistance. La première, consiste à appliquer une pression constante sur le piston de la seringue tout au long de l'insertion de l'aiguille. En faisant cela, le médecin ressent une résistance croissante puis la perte de résistance. La seconde méthode consiste à appliquer une pression par à-coups sur le piston. Ces à-coups peuvent ensuite être plus fréquents lorsque l'anesthésiste ressent une forte résistance. Il s'assure de ne pas rater l'occurrence du phénomène de perte de résistance. L'existence de plusieurs méthodes provient principalement du mode d'apprentissage du geste qui se fait généralement par l'encadrement d'un anesthésiste expert lors des premières poses. L'élève apprend donc la méthode que lui propose l'expert sans qu'il n'y ait de standard d'apprentissage. L'ensemble de ces techniques doit cependant être pris en compte lors du développement du simulateur haptique car il est souhaitable que chacune d'entre elles puisse être appliquées par les médecins utilisant le simulateur.

1.2 Enjeux et état de l'art de la formation au geste

1.2.1 Enjeux et difficultés

Rater une pose de périuridurale peut avoir deux conséquences distinctes. La première a lieu si l'aiguille n'atteint pas l'espace périuridural, soit en s'arrêtant trop tôt, soit en terminant sa course dans les tissus musculaires. Dans ce cas l'anesthésie n'aura aucun effet sans que l'anesthésiste ne le sache. L'autre conséquence intervient lorsque l'aiguille touche la dure mère et crée une brèche dans cette dernière. Dans ce

cas, du fait du diamètre de l'aiguille utilisée, la brèche ne pourra pas être colmatée suffisamment rapidement et une fuite du liquide céphalo-rachidien contenu dans la dure mère apparaît. Cette fuite peut alors entraîner de très importantes céphalées chez le patient, qui peuvent durer plusieurs jours sans qu'il y ait de moyen efficace de calmer la douleur.

L'entraînement se révèle être un élément crucial concernant la réussite de la péridurale et en particulier en ce qui concerne la constance des résultats. Une étude réalisée par KONRAD *et al.* [1] a mis en avant le fait qu'au moins 90 tentatives sont nécessaires pour atteindre un taux de réussite de seulement 80%. Bien évidemment, un tel taux de réussite ne saurait être suffisant et le nombre de tentatives pour y parvenir est extrêmement élevé. De plus la majeure partie de l'apprentissage est actuellement réalisé directement sur patient par manque de moyens de formation satisfaisants. Cependant un rapport de la Haute Autorité de la Santé [2] (HAS) préconise de ne pas réaliser les premiers pas de la formation sur des patients. Il est donc nécessaire de mettre à disposition des moyens de formation appropriés et efficaces.

Un des plus grands enjeux du point de vue médical est donc de fournir une formation de qualité qui raccourcisse le temps d'apprentissage du geste et assure une meilleure sécurité pour les patients. Dans cette optique, une étude de VAUGHAN *et al.* [3] a mis en avant plusieurs points cruciaux en ce qui concerne les moyens de formation à ce geste médical en particulier. Cette étude rassemble les pré-requis suivants :

- proposer une portion palpable représentant le dos d'un patient couplée avec une simulation numérique ;
- permettre de modifier les caractéristiques du patient, que ce soit la taille ou la résistance ressentie ;
- permettre de modifier le ressenti de perte de résistance généré, en particulier pour reproduire des pertes de résistance moins marquées ;
- proposer un compte rendu des performances pour les élèves à la suite d'un essai ;
- diffuser des enregistrements sonores de salles d'accouchement pour reproduire un peu mieux l'environnement de stress ;
- permettre l'ajustement des paramètres de la simulation en utilisant des données IRM ou CT scan par exemple ;
- permettre l'enregistrement des insertions réalisées.

1.2.2 Moyens de formation existants

Comme mentionné dans l'étude de VAUGHAN *et al.* [3] des solutions de formation ont été développées et pour certaines commercialisées. Cependant les solutions existantes se révèlent souvent trop incomplètes pour être suffisamment efficaces.

Parmi les solutions les plus communes, on peut mentionner l'utilisation de mannequins anatomiques. De tels mannequins sont très performants en terme de

réalisme du toucher et surtout de réalisme de la représentation du dos du patient (Figure 1.4a). Toutefois, ce type de simulateur reste peu utilisé car trop limité. En effet, ils représentent souvent un seul type de patient, le plus souvent le cas *moyen*. De plus ces mannequins imposent de changer des pièces régulièrement lorsque plusieurs essais ont été réalisés à cause de l'usure. En effet lorsque la peau factice a été percée plusieurs fois, elle devient inutilisable. Un autre modèle utilisé consiste en une simple banane comme présenté dans la Figure 1.4b représentant sans doute un des simulateur les plus simple et les moins couteux du marché, aussi appelé modèle de *greengrocer* [4] (modèle de l'épicier). Cependant les performances de celui-ci restent très limitées comme on pourrait l'imaginer.

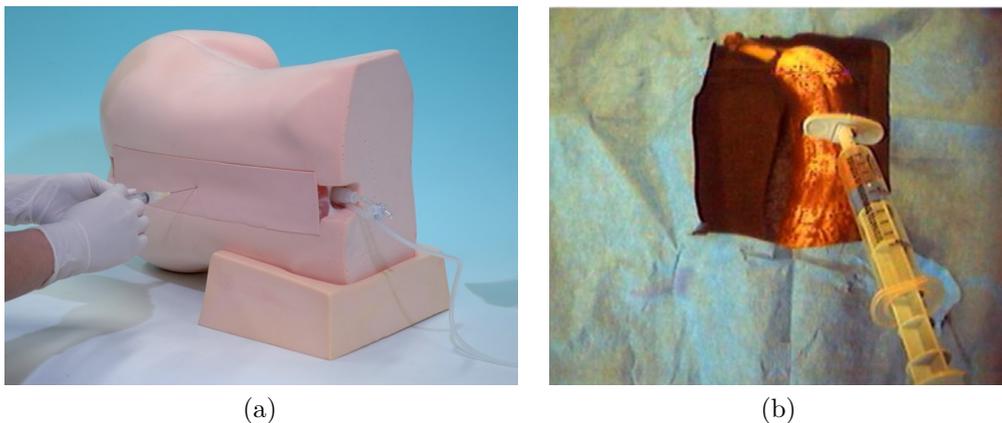


FIGURE 1.4 – Simulateurs utilisés pour l'apprentissage : (a) Exemple de mannequin de formation à la pose de péridurale de Limbs & Things, (b) Modèle de *greengrocer* [4]

Il existe aussi un certain nombre de simulateurs proposant d'utiliser une interface haptique pour représenter les retours de forces nécessaires (Figure 1.5). Certains sont mentionnés dans l'étude de VAUGHAN *et al.* [3], auxquels nous ajoutons d'autres simulateurs présentés dans la littérature : MAGILL *et al.* [5] (Figure 1.5a), MANOHARAN *et al.* [6] (Figure 1.5b), HIEMENZ *et al.* [7] ou THAO *et al.* [8]. Là encore, si ces solutions apportent quelques réponses aux recommandations formulées par la HAS [2] ou VAUGHAN [3], elles ne remplissent pas l'ensemble des exigences concernant un tel simulateur.

Ces solutions ont certains avantages : elles permettent de modifier les paramètres du patient et ne nécessitent pas de changer régulièrement des pièces du simulateur. Toutefois, en ce qui concerne la perte de résistance, ces simulateurs restent trop simplistes se limitant au tout-ou-rien. Pour reproduire la perte de résistance, une valve tout-ou-rien est montée sur une seringue classique. Ainsi, pour créer l'effet de perte de résistance, il suffit de maintenir la valve fermée pour l'ouvrir d'un seul coup au moment souhaité. Cela va à l'encontre de ce que préconise VAUGHAN *et al.* [3] qui recommande de jouer sur la soudaineté de la perte de résistance ainsi que sur les efforts à reproduire tout au long de la procédure. En effet, cette solution a deux principaux inconvénients. Premièrement elle ne permet pas de contrôler le ressenti produit avant la perte de résistance car elle bloque simplement l'avancée du piston de la seringue au lieu d'en restreindre le mouvement. En particulier, ce type de

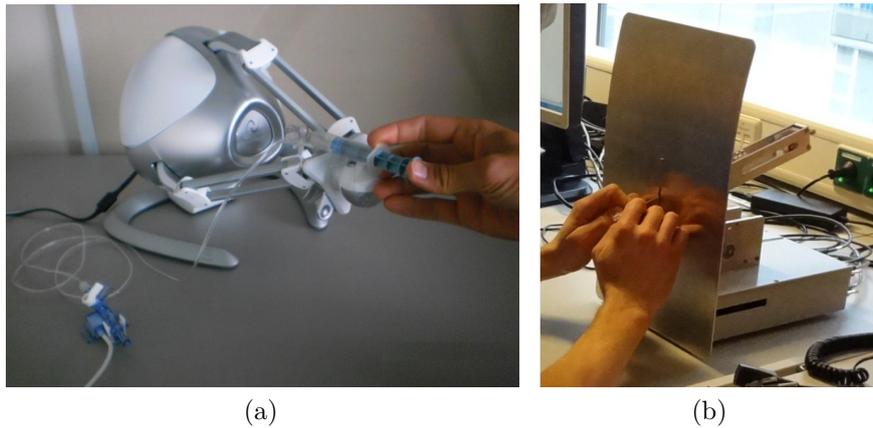


FIGURE 1.5 – (a) : simulateur haptique pour la péridurale développé par MAGILL [9, 5] proposant une simulation de la perte de résistance en utilisant une valve on/off, (b) : simulateur haptique pour la péridurale développé par MANOHARAN [6]

solution ne permet pas de faire varier la résistance à l’injection en fonction de la couche traversée. Or des mesures réalisées par TRAN *et al.* [10] sur la perte de résistance indiquent que l’effort résistant à l’injection ressenti au niveau du piston de la seringue varie tout au long de la procédure en fonction de la couche ou du ligament traversé par l’aiguille. Deuxièmement, cette méthode ne permet pas de contrôler la soudaineté de la perte de résistance. Or cette soudaineté est caractéristique de certains types de patient. Si nous souhaitons représenter la présence de parties graisseuse, caractéristiques d’un patient en sur-poids, il faut pouvoir produire une perte de résistance plus subtile. Il faut donc être en mesure de contrôler la soudaineté de la perte de résistance et ainsi offrir la représentation la plus réaliste et complète possible.

Un résumé des avantages et inconvénients des simulateurs cités précédemment est présenté dans la Table 1.1. Si chacun des simulateurs affiche des avantages, il est clair que le point noir de tous ces exemples est le manque de simulation précise de la perte de résistance. C’est particulièrement sur cet aspect que notre proposition de simulateur améliore le réalisme général.

1.3 Contexte scientifique : l’haptique

1.3.1 De l’intérêt de l’haptique

L’haptique correspond aux sciences, technologies et application visant à transmettre des informations sur un environnement ou un objet à travers le sens du toucher. Les applications de ce type de technologies sont assez vastes et diversifiées. Nous pouvons, par exemple, trouver dans la littérature des gants haptiques [11], des simulateurs d’apprentissage [5, 6, 8, 12] mais on retrouve aussi l’haptique dans des applications grand public comme par exemple dans les vibrations de manettes de jeux vidéos pour n’en citer qu’un (manette Sony Dual Shock 2 pour PS2 par exemple).

TABLE 1.1 – Comparaison des simulateurs existants

Simulateur	Avantages	Inconvénients
MANOHARAN [6]	<ul style="list-style-type: none"> • Représentation précise des efforts d'insertion d'aiguille • Enregistrement des données d'essais 	<ul style="list-style-type: none"> • Un seul degré de liberté dans le choix de l'angle d'insertion • Uniquement de la <i>LOR</i> tout-ou-rien • Pas d'utilisation de données IRM ou CT scan • Pas de système d'évaluation • Pas de partie palpable
MAGILL <i>et al.</i> [9, 5]	<ul style="list-style-type: none"> • Palpation du point d'insertion possible • Possibilité de faire varier la position du patient • Enregistrement des essais • Enregistrement des essais 	<ul style="list-style-type: none"> • Peu de variabilités dans le type de patient représenté • Uniquement de la <i>LOR</i> tout-ou-rien • Pas d'utilisation de données IRM ou CT scan • Pas d'évaluation de la performance
THAO <i>et al.</i> [8]	<ul style="list-style-type: none"> • Présence d'images par rayons X • Présence de deux degrés de liberté pour le choix de l'angle d'insertion • Choix de la position du patient • Enregistrement des essais réalisés • Différents niveaux de difficulté pour aider à l'apprentissage 	<ul style="list-style-type: none"> • Pas de réelle évaluation du geste réalisé • Manque de variabilité dans le patient représenté (âge, poids) • Uniquement de la <i>LOR</i> tout-ou-rien • Pas de partie palpable
HIEMENZ <i>et al.</i> [7]	<ul style="list-style-type: none"> • Présence d'image IRM pour localiser l'aiguille • Plusieurs angles de vue disponibles 	<ul style="list-style-type: none"> • Un seul degré de liberté pour l'insertion de l'aiguille • Manque de variabilité dans le patient représenté (âge, poids) • Pas de représentation de la <i>LOR</i> • Pas de partie palpable

Quelle que soit l'application, le recours à l'haptique permet d'ajouter un niveau de réalisme supplémentaire à une application. Il peut, par exemple, s'agir de représenter un cube virtuel et de le faire deviner à un utilisateur à travers la sensation du toucher. Des solutions utilisant de l'haptique ont été développées précédemment au sein du laboratoire comme par exemple le simulateur d'accouchement BirthSIM, issu notamment des travaux de HERZIG [13, 14] et de MOREAU [15]. Dans ce type d'application, l'haptique n'est pas un simple ajout à un produit plus vaste mais est au cœur de l'application. Dans ce contexte, il existe plusieurs fabricants d'interfaces haptiques généralistes comme Haption [16] ou encore 3D Systems [17] pour les plus connus. Certaines de leurs interfaces sont présentées en Figure 1.6. Ces fabricants proposent des interfaces complètes et un ensemble d'outil de développement pour les contrôler afin de les adapter à l'application souhaitée.



FIGURE 1.6 – (a) : l'interface haptique Touch X de 3D Systems, (b) : l'interface haptique Virtuose 6D de Haption

Dans le cadre de ce doctorat nous nous situons dans le cas des applications où l'haptique constitue l'essence de l'application développée. Il ne s'agit donc pas d'ajouter de légères touches d'haptique pour améliorer un réalisme global mais bien de reproduire la complexité d'une procédure réelle à l'aide d'une interface haptique.

1.3.2 État de l'art de l'haptique et contraintes

Les interfaces haptiques sont la plupart du temps couplées à une simulation 3D leur permettant de définir des efforts afin de produire la sensation de retour de force. Il est donc souvent nécessaire de définir cet environnement 3D de manière efficace et précise pour obtenir la meilleure simulation possible.

La contrainte temps interactif

Pour qu'un rendu haptique soit perçu comme réaliste, il est nécessaire que cela soit réalisé en «temps réel» et avec une fréquence d'échantillonnage relativement élevée (quelques kilohertz) par rapport à une simple simulation 3D visuelle (quelques dizaines de hertz). Si cette fréquence d'échantillonnage n'est pas assez élevée l'utilisateur pourra alors ressentir ce que l'on appelle «l'instabilité perçue». Celle-ci

correspond au fait que l'utilisateur se retrouve confronté à un retour haptique qui lui semble irréaliste. Un tel effet a pu être étudié dans le cas de ressenti de textures par exemple [18]. De plus, la période d'échantillonnage est liée à la capacité de l'interface à reproduire des objets très rigides [19] telles que des surfaces dures par exemple. C'est pourquoi les constructeurs d'interfaces haptiques préconisent de ne pas descendre en deçà de certaines valeurs de fréquence d'échantillonnage afin de pouvoir obtenir un rendu haptique suffisamment réaliste. De manière générale, plus la fréquence d'échantillonnage est élevée, plus il est simple de représenter des éléments rigides. La fréquence d'échantillonnage couramment utilisée est de 1 kHz [19, 20, 21].

Les systèmes à objets virtuels et proxy

La méthode la plus courante, lorsqu'il s'agit de reproduire les sensations du toucher avec un environnement virtuel, est d'avoir recours à un objet virtuel. Cet objet virtuel a pour objectif de représenter la position de l'utilisateur (ou de la poignée dans le cas d'un bras haptique) dans l'environnement de simulation afin de prendre en compte la présence de surfaces, d'une gravité particulière, d'une viscosité particulière, etc.

Si ce principe d'objet virtuel est à la base de la plupart des simulations utilisant des bras haptiques, il existe aussi plusieurs variantes permettant de créer l'effort, qui sera la consigne à transmettre à l'interface haptique. Parmi les premiers travaux sur le sujet, ZILLES *et al.* [22] présentent un fonctionnement dit de «*god-object*». Le fonctionnement général de ce genre de méthode est présenté en Figure 1.7. Le principe est de simuler la présence d'un mécanisme virtuel entre l'objet virtuel et la position réelle de la poignée constitué d'une raideur et d'un amortissement virtuels. Cette approche permet de facilement explorer des surfaces [22]. L'algorithme *god-object* a pour avantage d'être peu coûteux en terme de calcul et permet une exploration de surface simple et efficace y compris sur des surfaces complexes. La Figure 1.8 indique comment sont dirigés les efforts à reproduire connaissant la forme de l'objet et l'angle d'attaque de la surface par l'objet virtuel. Lorsque l'objet virtuel attaché à l'interface haptique entre en contact avec une surface dans l'environnement de simulation, celui-ci s'arrête sur la surface. La position réelle de l'interface, elle, n'est *a priori* pas contrainte à rester sur cette surface virtuelle et l'utilisateur peut donc continuer à avancer. À ce moment précis, la position réelle de l'interface et la position de l'objet virtuel attaché ne sont plus confondues et, grâce au mécanisme virtuel liant ces deux éléments, l'algorithme de *god-object* détermine un effort à appliquer sur l'interface. L'utilisateur peut alors ressentir un effort de contact.

Comme mentionné précédemment, cette méthode est très peu coûteuse en terme de calculs à réaliser et permet donc aisément de rester dans les contraintes imposées par le temps interactif. Classiquement, la norme du retour d'effort généré et simplement lié à la distance entre la position réelle de l'interface et la position de l'objet virtuel et la direction du retour d'effort est portée par la direction entre les deux points. Cette méthode peut donc s'appliquer à de l'exploration de surface ou à une force arbitraire que nous choisirions d'appliquer à l'objet virtuel.

Toutefois, il faut noter que la présence du mécanisme virtuel a pour effet de

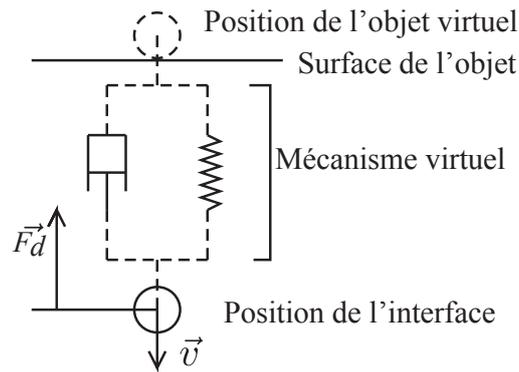


FIGURE 1.7 – Exemple d'un système de proxy où l'utilisateur entre en contact avec une surface virtuelle

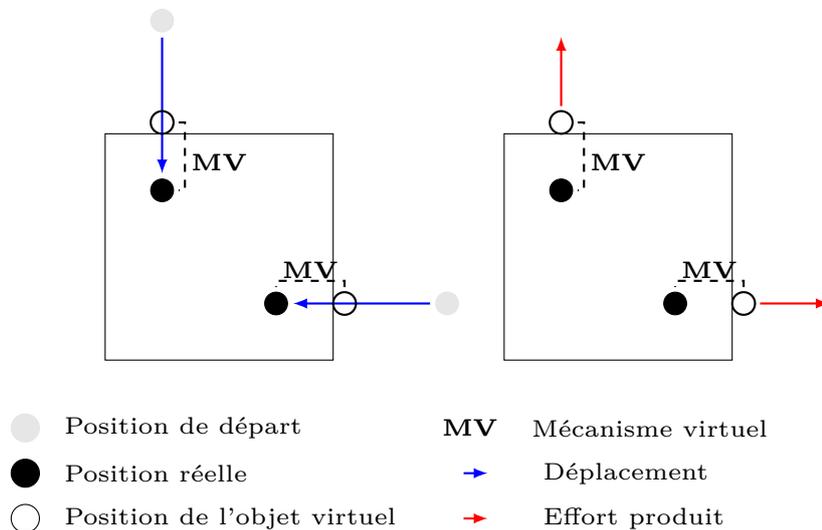


FIGURE 1.8 – Exemple de fonctionnement d'un algorithme *god-object* lorsque l'utilisateur entre en contact avec un objet virtuel dans la simulation. À gauche la direction du déplacement de la position réelle et à gauche la direction de l'effort produit.

filtrer l'effort ressenti. Cela peut être bénéfique dans certains cas en permettant de proposer un retour d'effort aussi lisse que possible. De plus, lorsqu'il s'agit d'objet «dur», ce type de fonctionnement se retrouve vite limité par la raideur utilisée dans le mécanisme virtuel. En effet, pour reproduire un objet parfaitement rigide, il faudrait être en mesure de reproduire une raideur quasi infinie, c'est à dire de bloquer mécaniquement l'interface haptique, ce qui n'est pas réalisable. En effet, la raideur et l'amortissement sont limités par les fréquences d'échantillonnage. Ainsi en utilisant uniquement ce fonctionnement, les surfaces palpées ont tendance à paraître «molles». Pour palier cela des solutions ont pu être proposées [23, 24, 25]. Certaines de ces solutions seront abordées dans le Chapitre 4 de ce manuscrit.

1.4 Objectifs du doctorat

1.4.1 Apports concernant l’haptique

La simulation de l’insertion de l’aiguille est réalisée en utilisant une logique de mécanisme virtuel et de proxy [22], nous permettant de générer une grande partie des rendus haptiques nécessaires. Pour améliorer le ressenti d’un contact avec un os, nous utiliserons les travaux de KUCHENBECKER *et al.* [23] en les adaptant à une application en 3 dimensions. Il reste cependant à reproduire les frottements et en particulier les sensations lors de la traversée du ligament jaune. Lors de cette traversée, selon le patient, il est possible de ressentir une sorte de craquement. Ce craquement est dû aux frottements secs et à un phénomène de *stick and slip* dont il faut proposer une simulation.

Concernant la reproduction de la perte de résistance, la solution proposée dans la littérature est d’utiliser une électrovanne connectée à une seringue, ouverte brusquement pour produire l’effet de perte de résistance. Dans notre cas, nous souhaitons apporter plus de nuances à la perte de résistance produite. Nous connecterons donc une servovalve à un vérin pneumatique pour reproduire le comportement réel de la seringue à l’aide d’un vérin pneumatique contrôlé à l’aide d’une servovalve. Il s’agit donc de synthétiser les lois commandes nécessaires à cette émulation.

Pour finir sur la partie haptique, le simulateur proposé utilise deux types d’actionneurs qui sont donc commandés indépendamment mais avec des paramètres communs et une synchronisation. Il faut en effet contrôler une interface haptique électrique pour représenter l’aiguille et un vérin pneumatique pour représenter la seringue. Ce type de solution hybride à deux énergies différentes est inédit en ce qui concerne les interfaces haptiques. Il a donc été nécessaire de synchroniser le fonctionnement des deux parties du simulateur à savoir l’interface haptique électrique et le contrôleur de l’ensemble servovalve-vérin.

1.4.2 Apports concernant l’étude du geste

Pour qu’un simulateur puisse être utilisé dans un contexte de formation, celui-ci doit permettre une évaluation de la performance, afin de quantifier la progression des apprenants par exemple. L’évaluation n’est bien entendu pas une fin en soi et il est aussi nécessaire de permettre à l’apprenant de s’améliorer en comprenant ses erreurs. Nos travaux se sont cependant majoritairement concentrés sur l’évaluation du geste. Dans notre cas il faut évaluer objectivement la maîtrise du geste de pose de péridurale. Il existe de nombreuses méthodes d’évaluation propres à chaque type de geste concerné. Dans le domaine médical, de nombreuses méthodes d’évaluation ont été développées pour étudier les gestes chirurgicaux. Nous pouvons par exemple citer la méthode OSATS [26] (pour la chirurgie classique) ou la méthode GOALS [27] (pour la chirurgie mini-invasive et en particulier la laparoscopie). Ces deux méthodes ont pour objectif de définir un score objectif de la performance d’un chirurgien et peuvent donc être utilisées à des fins de formations afin d’évaluer la maîtrise des futurs chirurgiens. Cependant, elles requièrent la présence d’un évaluateur observant l’apprenant et décernant la note. Dans le cadre de notre simulateur nous avons évalué

le geste à partir des données acquises lors d'essais.

Il existe de nombreuses métriques couramment utilisées pour l'évaluation de gestes médicaux, en particulier pour la chirurgie. Ces métriques ont pu être mises en avant au sein d'études de synthèse [28, 29] parmi ces études, les métriques présentées sont :

- le *TCT* [30] (*Task Completion Time*) qui correspond simplement au temps nécessaire à la réalisation d'une tâche ;
- la déviation par rapport à un chemin optimal [29] ;
- la régularité des mouvements [28] ;
- l'économie de mouvement [29] ;
- la longueur de déplacement [28] ;
- la courbure des trajectoires et la vitesse affine [31].

Le principe général de ces métriques est de définir des variables objectives permettant d'évaluer la maîtrise d'un geste en particulier. La plupart des métriques présentées ci-dessus sont appliquées à des gestes réalisés dans un environnement en trois dimensions et sont particulièrement pertinentes pour les gestes chirurgicaux. Ces gestes nécessitent dextérité et navigation dans l'espace, d'où la pertinence de métriques comme la vitesse affine, la courbure, la régularité de mouvement etc. Ce n'est pas le cas du geste de l'anesthésie péridurale. La difficulté principale se concentre sur l'interprétation des ressentis haptiques.

Dans notre travail nous acquérons des données des interfaces pendant la réalisation du geste. Avec ces données nous pouvons reconstruire certaines des métriques présentées précédemment lorsque celle-ci sont pertinentes. À cela s'ajoute des métriques spécifique au geste de la pose de péridurale. Parmi les variables clés, nous pouvons citer l'occurrence d'un éventuel échec de la procédure. Si l'on se réfère à l'étude de KONRAD *et al.*[1], une maîtrise importante est nécessaire pour réaliser une anesthésie péridurale avec un faible taux d'échec. Une autre métrique importante est le «dépassement», plus couramment appelé *overshoot*, qui correspond à la distance que parcourt la pointe de l'aiguille dans l'espace péridural. Ce dépassement a été mis en avant dans les travaux de MANOHARAN *et al.* [6] ou de VAN ADRICHEM [32]. Il est aussi l'objet d'études lorsqu'il s'agit d'étudier la qualité du geste de la péridurale [33]. Ce paramètre est donc un des paramètres à étudier pour le geste de la péridurale en particulier.

Notre approche consiste donc à définir un ensemble de paramètres mesurés à extraire de chaque mesures effectuées sur le simulateur pour établir un système d'évaluation spécifique au geste de la pose de péridurale.

Chapitre 2

Conception du simulateur

Introduction

Dans ce chapitre, nous allons aborder les choix de conception que nous avons fait. L'objectif est de répondre aux attentes formulées dans le chapitre précédent. Nous traiterons donc ici la représentation de l'insertion de l'aiguille et la représentation de la perte de résistance séparément. La solution proposée a la particularité d'utiliser deux types d'actionnements, à savoir une interface haptique électrique couplée à une vérin pneumatique. Chacune de ces parties est dédiée à la représentation d'une fraction du geste.

Nous traiterons aussi de l'ensemble des ajouts permettant de rendre l'aspect et l'utilisation du simulateur plus immersive et plus réaliste.

2.1 Caractéristiques du simulateur final : cahier des charges

Afin de réaliser un simulateur complet pour l'anesthésie péridurale, il convient de définir un ensemble de contraintes de conception préalables. Pour obtenir ces contraintes, nous nous sommes basés sur les travaux de VAUGHAN *et al.* [3] que nous avons par la suite étoffées et hiérarchisées à l'aide de deux anesthésistes des hospices civils de Lyon : Christian BAUER et Florent SIGWALT. Pour l'ensemble de ce cahier des charges nous noterons EC_i les exigences correspondant à des contraintes et EF_i les exigences fonctionnelles.

2.1.1 Caractéristiques principales

Réalisme anatomique

Pour pouvoir transposer au plus vite les compétences acquises sur le simulateur, il est nécessaire que celui-ci soit relativement représentatif de la réalité (EC_1). Il faut permettre la palpation (EF_1) de la zone du corps du patient concernée par l'anesthésie. Cette représentation peut être relativement simple. Toutefois, représenter quelques épineuses est essentiel pour permettre de bien situer l'opération ($EF_{1.1}$) et de localiser le point d'insertion de l'aiguille et ainsi donner des repères aux novices ($EF_{1.2}$).

Réalisme des outils

Toujours dans cette optique de transposition des acquis, même s'il n'est pas possible d'utiliser exactement les mêmes outils que lors d'une péridurale réelle, il est nécessaire que le simulateur propose de manipuler des outils relativement proches des instruments utilisés (EC_2). Il faut faire en sorte que la seringue utilisée soit représentée relativement fidèlement ($EC_{2.1}$) tout comme la longueur de l'aiguille utilisée ($EC_{2.2}$). En effet, pour le novice, il est important de pouvoir trouver une position d'utilisation confortable et reproductible. Si le *design* du simulateur s'éloigne trop de la réalité en se focalisant simplement sur le ressenti, il ne permettra pas aux apprenants d'acquérir une bonne technique qu'ils transposeront ensuite sur un patient.

Paramétrisation des patients et essais

Dans la réalité toute personne est différente. Cette affirmation évidente a de fortes implications concernant notre simulateur. En effet, le point noir de la majorité des simulateurs à base de mannequins est qu'ils ne proposent pas (ou très difficilement) de moyens de représenter différentes morphologies. Cependant les variations morphologies ont une importance cruciale sur les paramètres de simulation. Le poids de la personne influence fortement la proportion de graisses à traverser avant de parvenir aux ligaments, l'âge de la personne peut influencer sur la résistance à l'insertion de l'aiguille, etc. Tous ces paramètres sont essentiels à la réalisation d'une simulation complète et réaliste et devront être inclus dans notre travail (EF_2).

2.1.2 Rendus haptiques nécessaires au simulateur

Pour décider des rendus haptiques nécessaires pour notre simulateur, il convient de détailler le geste à reproduire afin de discerner les retours haptiques clés. La réalisation de ces points sera détaillée dans le Chapitre 3. Ces rendus haptiques peuvent se séparer en deux parties : l’insertion de l’aiguille et l’utilisation d’une seringue *LOR*.

L’insertion d’aiguille : EF₃

1. Cheminement de l’aiguille : EF_{3,1}

Lorsque l’insertion de l’aiguille a commencé, il devient impossible d’en diriger la pointe. Elle se retrouve contrainte à suivre une trajectoire rectiligne. Certains des tissus traversés sont en partie élastiques, comme le derme et les muscles, il est alors possible de s’éloigner momentanément de la direction initiale. Cependant, les ligaments sont eux beaucoup plus rigides et contraignent bien plus la trajectoire de l’aiguille. Notre simulateur devra être en mesure de maintenir l’utilisateur sur une trajectoire, ou l’y ramener s’il s’en éloigne, avec plus ou moins de rigidité selon le tissu traversé. De plus, lorsque l’aiguille se situe dans les ligaments, celle-ci ne doit plus être en mesure de pivoter sur elle-même.

2. Reproduire les efforts d’insertion d’une aiguille : EF_{3,2}

Pour cela, il faut s’intéresser à la manière dont est généralement modélisée l’insertion d’aiguilles dans la littérature. Selon OKAMURA *et al.* [34] les efforts en jeu lors d’une insertion peuvent s’écrire :

$$F_{ins} = F_{raideur} + F_{frottement} + F_{coupe} \quad (2.1)$$

Dans cette équation, $F_{raideur}$ correspond à la force ressentie lorsque la pointe de l’aiguille transperce un tissu, $F_{frottement}$ est la force due aux frottements entre les tissus traversés et le corps de l’aiguille et F_{coupe} correspond à la force nécessaire pour avancer dans un tissu ou force de coupe du tissu. Dans notre cas, $F_{raideur}$ est appliquée lorsque l’aiguille pénètre le derme. F_{coupe} dépend de la couche traversée et reste constante dans une couche donnée. Finalement $F_{frottement}$ dépend non seulement de la profondeur d’insertion actuelle de l’aiguille (celle-ci étant directement corrélée à la surface en contact entre aiguille et tissus), mais aussi de la couche physique traversée. Cette force peut par exemple être utilisée pour signifier l’apparition de frottements secs et de «craquements» en particulier pour le ligament jaune. Sur certains patients, le ligament jaune peut être partiellement calcifié ou même simplement extrêmement compact et il n’est alors pas rare de ressentir des craquements lorsque la pointe de l’aiguille le traverse.

3. Reproduire la sensation de toucher un os : EF_{3,3}

Concernant l’insertion de l’aiguille, le dernier point à considérer et à reproduire est la possibilité de rencontrer une vertèbre lors de l’insertion. Dans ce cas, il faudra

être en mesure de représenter un contact dur et de signifier à l'utilisateur qu'il n'a d'autre choix que de reculer. Dans un cas réel, lorsque la pointe de l'aiguille entre en contact avec un os, elle se plante légèrement et l'anesthésiste doit alors retirer l'aiguille afin de modifier son angle d'insertion. Il faut en particulier être en mesure de contrer l'impression de toucher un objet déformable mou, comme c'est souvent le cas avec une solution par simple proxy.

Pour résumer l'ensemble de ces efforts peut s'écrire comme suit :

$$\vec{F}_{ins} = \vec{F}_{raideur} + \vec{F}_{frottement} + \vec{F}_{coupe} + \vec{F}_{traj} + \vec{F}_{contact} \quad (2.2)$$

Dans cette équation, nous reprenons donc la modélisation de OKAMURA *et al.* [34]. Nous y ajoutons la force permettant de fixer la trajectoire avec (\vec{F}_{traj}) qui correspond à l'ensemble des retour d'efforts générés pour reproduire la sensation d'une trajectoire rectiligne. Enfin, la force de contact avec un os est noté ($\vec{F}_{contact}$) représente une collision éventuelle de l'aiguille avec une seringue.

Simulation de la seringue : EF₄

1. Reproduire la sensation de perte de résistance : EF_{4.1}

La méthode classique permettant de reconnaître l'entrée dans l'espace péridural est d'utiliser le phénomène de perte de résistance. Il est donc nécessaire de mettre en œuvre une simulation de seringue qui sera utilisée pour ressentir cette perte de résistance. Si la plupart des simulateurs reproduit cette perte de résistance, ils ne permettent pas de moduler cet effet. Mais d'après VAUGHAN *et al.* [3], être en mesure de reproduire différentes sortes de pertes de résistance est nécessaire afin de simuler différents types de patient et donc d'améliorer la formation proposée.

2. Simuler la diffusion de sérum dans les tissus : EF_{4.2}

Lorsque l'aiguille traverse les divers tissus présents dans l'espace intervertébral, il est possible d'injecter un peu de liquide. En effet, le sérum contenu dans la seringue peut diffuser dans ces tissus. Toutefois, cette capacité de diffusion est différente selon le tissu traversé. Cela a pour effet de produire une résistance à l'injection différente selon le tissu au niveau du piston de la seringue. Notre simulation de seringue doit permettre de simuler ces différents niveaux de résistance pour refléter ce phénomène.

3. Maintenir le piston en position si l'utilisateur n'appuie pas dessus : EF_{4.3}

Pour se rapprocher au plus des outils réels il faut parvenir à reproduire le fonctionnement basique d'une seringue. Lorsque l'on utilise une seringue, si l'utilisateur n'agit pas sur le piston, celui-ci reste en place. Si cela est extrêmement évident pour une seringue qui est intrinsèquement passive, ce n'est pas le cas *a priori* de notre simulateur. En effet, si l'on ne se focalise que sur la reproduction de la sensation de perte de résistance, il suffit de produire différents efforts qui s'opposeront à la force

appliquée par l'utilisateur. En s'arrêtant là, lorsque l'utilisateur cesse d'appuyer, le simulateur pourra lui continuer à appliquer une force et le piston de notre seringue simulée finira donc par ressortir complètement. Il faut donc être en mesure de non seulement produire des efforts distincts lorsque l'utilisateur presse sur le simulateur, mais aussi de garder le piston en position lorsqu'il n'y a plus aucune pression appliquée par l'utilisateur.

4. Bloquer le mouvement du piston si nécessaire : EF_{4.4}

Tout comme le fait de toucher un os a un effet sur la représentation haptique côté insertion de l'aiguille, cet évènement a une influence côté seringue. Dans l'ensemble des tissus mous (ligaments, peau, muscles), il est possible d'injecter ne serait-ce qu'une infime partie du contenu de la seringue car celui-ci peut diffuser au sein de ces tissus. Cependant, lorsque l'aiguille rencontre un os, le liquide contenu dans la seringue ne peut pas diffuser dans l'os, il n'est alors pas possible d'injecter et la résistance opposée par le piston devient «infinie». Cela correspond donc à une position fixe au niveau du piston. Ainsi pour compléter notre simulation nous devons être en mesure de bloquer le mouvement du piston si nécessaire.

2.1.3 Vers une évaluation automatique et objective du geste : EF₅

L'objectif final d'un tel simulateur est d'aider à la formation et de limiter le recours à l'apprentissage *in vivo*. Cependant, il faut être en mesure de produire une évaluation. C'est donc un objectif supplémentaire et le simulateur présenté ici doit permettre d'évaluer des niveaux d'expertise. Cette évaluation se doit de rester la plus objective et précise possible.

Pour cela, il est nécessaire de se pencher sur les métriques et données à utiliser afin de permettre une telle évaluation ainsi que sur les points à valider afin de déterminer une possible action de formation.

2.1.4 Résumé du cahier des charges

En compilant l'ensemble des exigences présentées ci-dessus, nous pouvons établir le tableau récapitulatif présenté dans la Table 2.1.

2.2 Choix de conception

Pour ce qui est des choix de conception, nous avons séparé le simulateur en deux parties correspondant chacune à une portion de la procédure : une partie chargée de la représentation de l'insertion d'aiguille et l'autre chargée de la représentation de la seringue. Nous avons donc choisi deux solutions haptiques distinctes qui seront par la suite couplées afin de proposer une simulation complète de la procédure.

TABLE 2.1 – Tableau récapitulatif du cahier des charges de notre simulateur et chapitres traitant du sujet

Exigence	Signification	Implémentation
EC ₁	Représenter précisément le dos du patient	Section 2.2.3
EC ₂	Représenter précisément les outils utilisés	
EC _{2.1}	Représenter la longueur de l'aiguille utilisée	Chapitre 4
EC _{2.2}	Représenter la seringue utilisée	Section 2.2.2
EF ₁	Permettre la palpation du dos	Section 2.2.3
EF _{1.1}	Représenter les épineuses	
EF _{1.2}	Permettre de localiser le point d'insertion de l'aiguille	
EF ₂	Permettre de paramétrer les éléments de la simulation pour reproduire plusieurs types de patients	Chapitre 4
EF ₃	Reproduire les sensations haptiques d'insertion d'une aiguille dans l'espace intervertébral	Chapitre 4
EF _{3.1}	Simuler le cheminement de l'aiguille dans l'espace intervertébral	
EF _{3.2}	Simuler et reproduire les efforts d'insertion d'aiguille	
EF _{3.3}	Reproduire la sensation de toucher un os	
EF ₄	Simuler le comportement d'une seringue	Chapitre 3
EF _{4.1}	Reproduire la sensation de la perte de résistance	
EF _{4.2}	Simuler la diffusion de sérum dans les tissus	
EF _{4.3}	Maintenir le piston en position si l'utilisateur n'appuie pas dessus	
EF _{4.4}	Pouvoir bloquer le mouvement du piston de la seringue	
EF ₅	proposer un système d'évaluation de la performance	Chapitre 5

2.2.1 Représentation de l'insertion de l'aiguille

L'insertion d'aiguille dans un cas réel

Dans un cas réel, une fois le point d'insertion choisi, l'anesthésiste doit choisir une orientation, un angle d'attaque, afin d'atteindre l'espace péri-dural. Dans un cas réel, le choix de cette trajectoire peut se révéler complexe. En effet sans moyen de visualisation préalable, il est impossible de connaître la position exacte des deux vertèbres et il peut donc arriver d'en rencontrer une lors d'une insertion. Dans ce cas le praticien devra ressortir l'aiguille pour modifier son orientation avant de recommencer. Il faut donc permettre à l'utilisateur de notre simulateur de choisir son angle d'insertion avec le plus de liberté possible.

Au niveau des efforts ressentis, la résistance à l'insertion dépend de la couche traversée et peut atteindre environ 10 N. Cette valeur d'effort résistant relativement importante est un des points clés pour représenter la complexité de la procédure. Les deux aspects principaux concernant l'insertion de l'aiguille sont donc les degrés de mobilité et l'effort maximal continu qui pourra être produit par l'interface.

Enfin en pratique, une fois que l'aiguille atteint les premiers ligaments, la trajectoire de celle-ci est par la suite nécessairement rectiligne et il devient impossible de pivoter l'aiguille. De ce fait, notre solution pour reproduire l'insertion de l'aiguille doit permettre de limiter la trajectoire à une trajectoire rectiligne une fois les premiers ligaments atteints.

Solution choisie

Nous avons décidé d'utiliser une interface haptique électrique existante afin de réaliser cette partie, ainsi plusieurs choix étaient disponibles. Parmi eux, il est possible de citer les interfaces haptiques proposées par Haption®, 3D System ou Force Dimension. Notre premier choix a été d'utiliser une interface haptique de type bras haptique comme par exemple la série des VirtuoseTM d'Haption ou les Geomagics de 3D System (Figure 2.1).



FIGURE 2.1 – (a) interface Force Dimension Delta.3 [35] (b) interface 3DSystem Geomagic Touch [36] (c) interface Haption® VirtuoseTM 6D [37]

L'ensemble de ces solutions remplit les attentes en termes de degrés de mobilité de la poignée qui pourraient être utilisés pour représenter l'insertion de l'aiguille. Toutefois, nous avons par la suite choisi les interfaces haptique Haption car celles-ci proposent un actionnement des degrés de mobilité de la poignée sur tous les modèles de la gamme VirtuoseTM.

La gamme d'interfaces haptiques Force Dimension delta.3 [35] (Figure 1.6(a)) ou delta.6 [35] annoncent des efforts en translation plus importants du fait de l'architecture parallèle du robot. Cela est un des points important de notre simulation et les valeurs d'efforts en translation annoncés sont de 20 N pour les deux versions de l'interface. Cependant, le contrôle d'effort en rotation au niveau de la poignée n'est présent que sur la version la plus coûteuse de l'interface.

La gamme d'interfaces Geomagic Touch (Figure 1.6(b)) et Touch X [36] présente elle un effort en translation bien plus faible que les interfaces Force Dimension avec seulement 3.3 N pour la version Touch et 7.9 N pour la version Touch X. Pour ce qui est de couple résistant au niveau de la poignée ceux-ci sont absents ce qui complexifierait le rendu haptique dans notre application.

Pour ce qui est de la gamme Haption® VirtuoseTM 6D [37] (Figure 1.6(c)) ou VirtuoseTM 6D Desktop [37], les efforts en translation sont relativement similaires à ceux de la solution proposée par 3DSYSTEM avec 3 N pour le 6D Desktop et 9.5 N pour la version 6D classique. Toutefois, toutes les versions de cette interface proposent six degrés de liberté actifs à savoir les trois translations mais aussi les trois rotations de la poignée.

L’interface que nous avons choisie d’utiliser est le VirtuoseTM 6D de Haption®. Cette interface dispose de six degrés de mobilité au total et permet de contrôler les efforts produits au niveau de l’orientation de la poignée. Les degrés de mobilité en rotation de la poignée nous serviront à assurer un bon suivi de trajectoire rectiligne et d’éviter que l’on puisse pivoter autour de l’axe de l’aiguille. De plus cette interface est capable de produire des efforts importants en continu (10 N) ce qui améliore grandement le réalisme général de la simulation. Enfin, cette interface est fournie avec un API simplifiant son contrôle au travers d’une application C++.

2.2.2 Représentation de la seringue et simulation de la perte de résistance

La perte de résistance en pratique

La perte de résistance est l’élément central de notre simulation du fait de son importance dans la réalisation de l’anesthésie péridurale. Si ce principe est souvent laissé de côté [7, 6] (ou tout du moins traité très sommairement [5, 8, 9]) par les simulateurs existants, notre simulateur se devra de proposer une simulation plus complète permettant plus de variabilité dans les rendus haptiques proposés.

L’un des problèmes principaux concernant la perte de résistance est qu’il n’existe que très peu d’études présentant des mesures d’efforts ressentis. L’étude la plus complète sur le sujet a été réalisée par TRAN *et al.* [10] et présente à la fois les efforts sur l’aiguille et les pressions à l’intérieur de la seringue. Cette étude a mis en avant des efforts de l’ordre de la dizaine de newtons maximum sur le piston de la seringue une fois convertis à partir des valeurs de pressions fournies. Ces valeurs ont d’ailleurs pu être confirmées lors des premiers tests sur banc d’essais haptique avec les médecins associés au projet. Cependant, cette étude ne présente que les résultats d’un jeu de mesures réalisées sur cadavre porcin. Il est donc impossible d’en tirer des conclusions concernant des possibles différences découlant de la morphologie des patients. Nous avons donc dû avoir recours à l’expérience des anesthésistes travaillant avec nous sur le projet afin de définir la variabilité à mettre en œuvre.

À partir de ce constat, nous avons mis en avant plusieurs points à reproduire dans notre représentation haptique de la perte de résistance. Il faut être en mesure de produire des efforts résistants d’intensité variable afin d’être en mesure de représenter la perte de résistance. De plus, selon le type de patient que nous souhaitons représenter, les niveaux d’efforts maximums peuvent être différents selon les caractéristiques des ligaments traversés. La présence éventuelle de graisse ou une calcification partielle fait varier la résistance ressentie sur le piston de la seringue. Enfin un autre point essentiel dans la représentation de la perte de résistance est la soudaineté de cette

perte de résistance. Selon le type de patient, la perte de résistance peut être plus ou moins soudaine. C'est particulièrement le cas des personnes en sur-poids. Dans ce cas, il est possible de rencontrer de la graisse à l'interface entre le ligament jaune et l'espace péri-dural. Dans ce cas la perte de résistance pourra être qualifiée de «douce». Pour l'ensemble de ces raisons, reproduire la sensation de perte de résistance ressentie au travers de l'utilisation de la seringue nécessite un contrôle relativement précis de l'effort résistant que l'on produit.

Solution choisie

Afin d'être fidèle à la réalité, il faut choisir un moyen de reproduire des efforts pouvant aller jusqu'à une dizaine de newtons en continu pendant toute une procédure complète. Les solutions envisagées étaient les suivantes :

- Utiliser un système à moteur électrique associé à un mécanisme de pignon crémaillère
- Utiliser un vérin hydraulique,
- Utiliser un vérin pneumatique.

Souhaitant proposer une interface finale relativement proche de la pratique réelle en terme de forme et d'utilisation, il a été choisi d'utiliser une solution à base de vérin. De plus, compte tenu des efforts nécessaires, afin de limiter au maximum le poids de l'ensemble, la solution électrique a été très vite éliminée du fait de son rapport poids puissance jugé trop faible par rapport à d'autres types d'actionnement. Nous avons alors choisi d'utiliser un vérin pneumatique plutôt qu'une version hydraulique, principalement pour des raisons de maintenance (fuite d'huiles par exemple), de gamme d'effort et de prix. De plus les solutions pneumatiques permettent de tirer profit de la compliance naturelle de l'actionneur pour fournir un rendu haptique assez réaliste. Ces différents avantages et caractéristiques ont permis l'utilisation d'actionneurs pneumatiques dans divers projets haptiques [38], y compris au sein du laboratoire Ampère [13].

Afin de se rapprocher au plus de l'apparence et de la prise en main d'une seringue, nous avons ajouté trois pièces obtenues par fabrication additive pour les ailettes et l'extrémité du piston. Pour ce qui est du modèle de vérin choisi et de son instrumentation, l'objectif était de se rapprocher au plus d'une seringue de perte de résistance et de rester suffisamment léger pour pouvoir se monter sur l'interface haptique électrique choisie. Le vérin utilisé est un vérin double effet Airpel® à faible friction (Airpel modèle M16D100D [39]). Afin de contrôler le fonctionnement de ce vérin nous utilisons une servovalve Festo MPYE®-5-M5-010 B. Concernant le capteur de position, notre choix s'est porté sur un capteur magnétique (MPS-064TSNU0 [40]) de SICK et pour les capteurs de pressions nous avons utilisés des capteurs miniatures FESTO (EPRB-1 [41]). Ces deux choix de capteurs étaient la solution la plus compacte afin de garder des dimensions raisonnables. Le vérin équipé et instrumenté est présenté en Figure 2.2. Le contrôle de l'ensemble est effectué avec une interface dSPACE (DS1103) et une application ControlDesk.

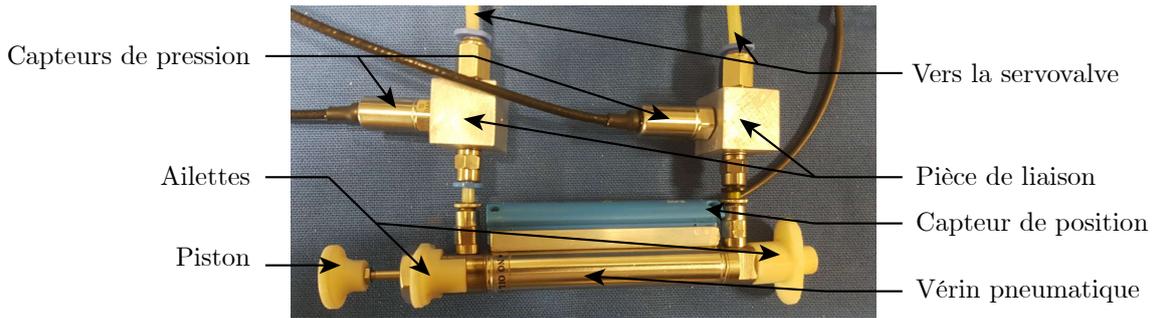


FIGURE 2.2 – Vérin pneumatique équipé de ses capteurs et des pièces ajoutées afin de se rapprocher de la prise en main et de l'apparence d'une seringue de perte de résistance

Cette solution permet de rester relativement proche des dimensions d'une seringue utilisée en pratique et permet donc d'avoir une prise en main semblable à la prise en main d'une seringue réelle.

2.2.3 Vers une meilleure représentation du patient

Afin de coller au plus proche de la réalité, nous ajoutons des pièces imprimées en 3D au vérin pneumatique pour le rapprocher au plus d'une véritable seringue en termes d'apparence et de prise en main. Un autre point important est la représentation du dos du patient. Pour cela nous avons ajouté un dispositif représentant physiquement le dos du patient. Nous avons utilisé des pans de mousses fixés sur un assemblage de plaques d'aluminium et maintenues en position par deux plaques d'aluminium supplémentaires. Ce dispositif a été réalisé en se basant sur des mesures de tailles dos de femmes enceintes, que ce soit de la courbure ou de la hauteur. La CAO correspondante ainsi qu'un aperçu du dispositif final sont présentés en Figure 2.3.

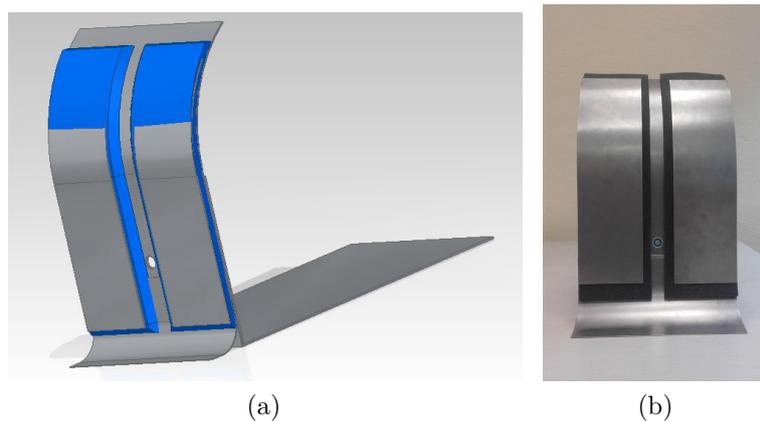


FIGURE 2.3 – (a) design du dispositif représentant le dos du patient sous CAO (b) dispositif réel vu de face (point de vue de l'utilisateur)

La séparation entre les plaques représente la colonne vertébrale du patient. Sur la moitié basse du dispositif, il est possible de distinguer ce qui servira de

point d'insertion de l'aiguille. Une photographie de cet ensemble est présentée sur la Figure 2.4. Cette partie est constituée de deux pièces obtenue par fabrication additive, séparées par une couche de mousse. Cela permet de maintenir le point d'insertion fixe tout en permettant de laisser libre toute les rotations nécessaires afin d'avoir une liaison linéaire annulaire. L'ajout de mousse permet ensuite de limiter la rigidité de la liaison, ce qui ajoute au ressenti réaliste de l'ensemble.



FIGURE 2.4 – Détail de la partie représentant le point d'insertion de l'aiguille

Pour améliorer la représentation du dos en particulier au toucher, nous avons par la suite ajouté des reproductions d'épineuses afin de permettre de repérer où se situe le point d'insertion choisi. Puis, la zone proche du point d'insertion a été couverte d'un patch de silicone habituellement utilisé sur les mannequins médicaux pour permettre la palpation de la zone concernée. Enfin, le reste du simulateur est couvert par des champs médicaux similaires à ceux utilisés en pratique afin de pousser un peu plus le sentiment d'immersion.

Avec ce dispositif, l'orientation du dos ou la position précise des épineuses n'est pas modifiable. Cependant, ce dispositif répond surtout aux exigences EF_1 et EC_1 . En effet, pour réaliser une pose de péridurale, l'anesthésiste repose le dos d'une main sur le dos du patient. Cet appui lui assure en particulier un meilleur contrôle de l'avancée de l'aiguille. Il est possible de distinguer cet appui de la main gauche sur la photo Figure 2.5a. C'est particulièrement utile lorsque l'aiguille traverse le ligament jaune et finit par arriver dans l'espace péridural. À ce moment précis, l'anesthésiste ressent la perte de résistance et il devient bien plus facile de poursuivre l'insertion de l'aiguille. Cependant, comme l'espace péridural reste relativement peu profond, il est crucial que l'anesthésiste stoppe au plus vite l'avancée de l'aiguille. Ainsi en s'appuyant sur le dos du patient avec sa main, l'anesthésiste peut éviter d'avancer trop profondément dans l'espace péridural. Le simulateur complet est présenté en utilisation en Figure 2.5.

2.2.4 Génération d'archétypes de patients

Dans l'optique de proposer une alternative pertinente aux mannequins, un des points essentiel de notre simulateur doit être la possibilité de proposer à l'utilisateur différents types de patients afin de coller au mieux aux situations que l'apprenant pourra rencontrer en pratique. Pour cela, nous avons travaillé avec Christian BAUER et Florent SIGWALT, en se basant principalement sur les mesures de TRAN *et al.* [10]

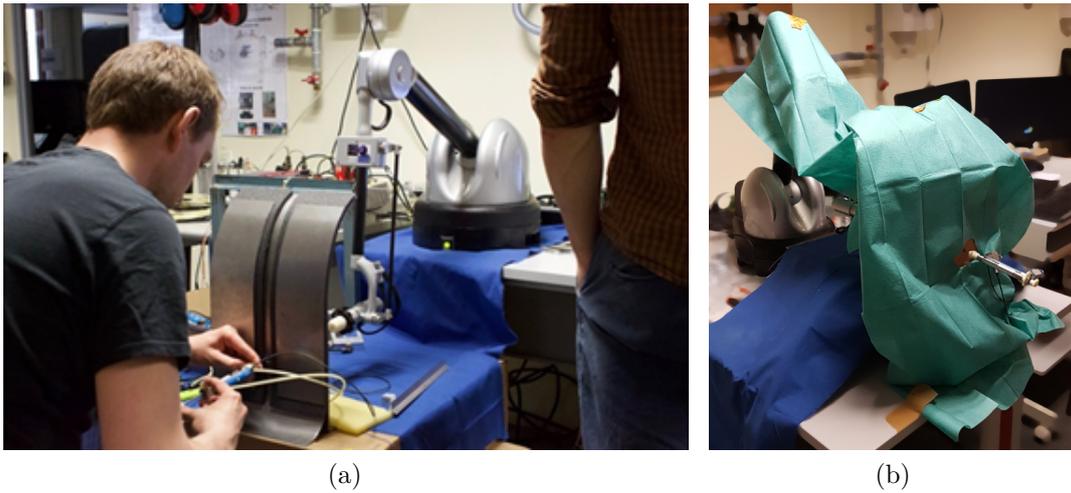


FIGURE 2.5 – (a) simulateur complet en utilisation, (b) aspect final du simulateur (champs compris)

comme point de départ. Suite à nos discussions, nous avons décidé de commencer en proposant trois types de patients afin de couvrir la majorité des cas possibles. Bien entendu, une fois les archétypes mis en place, il est tout à fait possible de générer des patients plus ou moins complexes hors de ces archétypes. En effet, générer ces premiers archétypes était aussi un moyen pour nous d'identifier les paramètres de l'anatomie d'un patient à rendre modifiables par l'utilisateur.

Les trois archétypes mis en œuvre sont : un patient moyen, un patient âgé et un patient en sur-poids. En terme de paramètres modifiables, cela correspond à l'épaisseur du derme, la profondeur de l'espace péri-dural, l'effort produit par le ligament jaune, la présence ou non de frottements secs lors de la traversée du ligament jaune et la soudaineté de la perte de résistance. L'ensemble des valeurs retenues pour chacun de ces paramètres en fonction du type de patient simulé est reporté dans la Table 2.2 ci-dessous.

TABLE 2.2 – Archétypes de patients prédéfinis (F_{dura} correspond à l'effort de coupe du ligament jaune)

Patient	Épaisseur du derme (cm)	Épaisseur de l'espace péri-dural (mm)	F_{liga} (N)	«Soudaineté» de perte de résistance	Frottements
Moyen	1,0	6,0	8,0	moyen	non
Surpoids	3,0	4,0	6,0	lente	non
Agé	1,0	4,0	14,0	rapide	oui

Chaque type de patient apporte un intérêt de formation particulier. Si le patient dit moyen présente une première approche de la péri-durale permettant ainsi de se

familiariser avec le ressenti de la perte de résistance par exemple, les autres archétypes proposent de se former à des cas plus complexes. Par exemple, le cas d'un patient en sur-poids implique que la longueur d'insertion de l'aiguille est plus longue et la perte de résistance plus subtile donc plus difficile à repérer avec certitude. Enfin le cas d'un patient âgé a été associé avec la possibilité de présenter une calcification du ligament jaune. Dans ce cas, il devient plus difficile d'insérer l'aiguille dans le ligament ce qui a trois effets principaux : la résistance à l'insertion de l'aiguille est plus importante, la résistance à l'injection de fluide est, elle aussi, plus importante et la perte de résistance est extrêmement marquée. Du fait de ces efforts importants, il peut donc être difficile de s'arrêter à temps. Ce cas particulier requiert donc une plus grande précision.

Conclusion

La solution de simulation proposée se compose donc de deux types d'actionnements différents afin de proposer une simulation complète et réaliste de la procédure. Cette conception a pour objectif de tirer profit des avantages de chaque type d'actionnement.

D'une part, l'interface électrique permet d'avoir une interface efficace pour reproduire une insertion d'aiguille, proposant suffisamment de degrés de liberté actifs. De plus, ce type d'interface est couramment utilisé et dispose d'outils de développement et de contrôles adaptés. Il est donc relativement simple pour nous de développer une application dédiée à la procédure d'insertion d'aiguille.

D'autre part l'utilisation d'un vérin pneumatique en complément nous autorise à proposer un rendu haptique suffisamment réaliste tout en produisant des efforts résistants relativement importants. Cette solution avec vérin pneumatique nous permet de rester relativement proche de la forme et de la taille d'une véritable seringue. Cela améliore surtout le réalisme de la simulation en s'éloignant d'une représentation de la perte de résistance «tout-ou-rien». Bien entendu, l'utilisation de deux interfaces impose de devoir les synchroniser pendant leur fonctionnement.

Enfin, l'ajout de pièces permettant de représenter le dos du patient à la manière d'un mannequin laisse la possibilité aux apprenants de palper la zone d'insertion et de se familiariser avec le bon positionnement de l'aiguille. Cela permet donc de garder quelques avantages des mannequins tout en proposant une simulation plus détaillée et personnalisable de la procédure en elle-même. Toutefois, l'inconvénient de ce système est qu'il ne permet pas à l'utilisateur de choisir son point d'insertion.

Chapitre 3

Reproduire la sensation de perte de résistance : contrôle du vérin pneumatique

Introduction

Dans ce chapitre nous allons aborder les travaux concernant le fonctionnement de la partie pneumatique de notre simulateur. La notion de *LOR* étant primordiale dans la réalisation d'une pose de péridurale réussie, nous avons décidé de faire de sa représentation un des points clés de notre simulateur. Pour cela nous nous sommes éloignés du fonctionnement classique, avec valve tout-ou-rien, de ce type de simulation pour se diriger vers une solution certes plus complexe mais offrant aussi plus de personnalisation des rendus haptiques générés.

Les solutions présentées précédemment prenaient en effet le parti d'une représentation simpliste de la *LOR* [9, 5, 8] que l'on peut qualifier de perte de résistance tout ou rien. Ce type de perte de résistance, bien que possible ne représente qu'une petite partie des cas envisageables, c'est pourquoi VAUGHAN *et al.* [3] met en avant la nécessité de disposer d'un certain degré de personnalisation de la sensation produite par la perte de résistance.

Pour notre solution, nous avons choisi l'utilisation d'un vérin pneumatique double effet afin de représenter la seringue de perte de résistance utilisée en pratique. Dans ce chapitre nous nous pencherons sur la modélisation utilisée pour contrôler notre vérin, puis nous présenterons les lois de commandes utilisées ainsi que leur implémentation, que ce soit en pratique ou en simulation.

3.1 Modélisation du vérin : la transformée A-T

Pour modéliser le vérin pneumatique, le choix s'est porté sur la transformée A-T présentée par ABRY *et al.* [42, 43] et utilisée sur le simulateur BirthSIM [14, 13]. Cette modélisation se base sur une approche thermodynamique permettant de décrire le fonctionnement du vérin pneumatique. Nous allons ici reprendre cette modélisation en la détaillant rapidement sans reprendre pour autant l'ensemble des calculs intermédiaires. Le schéma en Figure 3.1 présente le vérin pneumatique ainsi que les différentes variables associées aux deux chambres. Pour la suite nous utiliserons donc les conventions suivantes :

- le point central de la course du vérin est considéré comme étant $y = 0$, la valeur de y étant considérée croissante si la tige du vérin sort ;
- la chambre côté tige (et utilisateur) est notée chambre N. Toute variable associée à cette chambre sera notée avec un «N» en indice dans l'ensemble des équations de ce chapitre ;
- la chambre opposée est notée chambre P. Toute variable associée à cette chambre sera notée avec un «P» en indice dans l'ensemble des équations de ce chapitre.

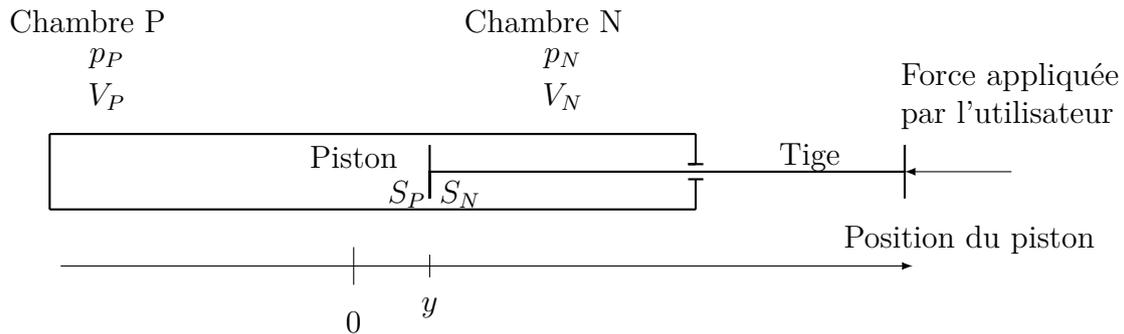


FIGURE 3.1 – Schéma du vérin et des variables utilisées

Partant de là, les travaux de ABRY *et al.* [42] proposent une approche thermodynamique de la modélisation du comportement du vérin débouchant dans un premier lieu sur le système d'équations ci-dessous :

$$\left\{ \begin{array}{l} \frac{dy}{dt} = v \\ \frac{dv}{dt} = \frac{-b \cdot v + F_{pneu}}{M} \\ \frac{dp_P}{dt} = \frac{k \cdot R \cdot T}{V_P(y)} \cdot q_{mP} - \frac{S_P}{R \cdot T} \cdot p_P \cdot v \\ \frac{dp_N}{dt} = \frac{k \cdot R \cdot T}{V_N(y)} \cdot q_{mN} - \frac{S_N}{R \cdot T} \cdot p_N \cdot v \end{array} \right. \quad (3.1)$$

Ici, y est la position du vérin. Nous noterons de plus l la longueur totale de la course du piston, v sa vitesse, b le coefficient de frottement visqueux, M la masse attachée à la tige, p_P et p_N les pressions dans chacune des chambres, V_P et V_N les volumes des chambres, S_P et S_N les sections de chaque chambre, q_{mP} et q_{mN} les débits massiques entrant dans chacune des chambres, T la température, k la constante polytropique, R la constante des gaz parfaits et F_{pneu} la force pneumatique créée par le vérin. Cette force pneumatique est définie comme suit :

$$F_{pneu} = p_P \cdot S_P - p_N \cdot S_N \quad (3.2)$$

Dans cette modélisation nous n'utiliserons que la force produite par les frottements visqueux. Les frottements secs seront négligés du fait de l'utilisation d'un vérin spécifique *Low stiction* de Airpel (modèle M16D100D [39]).

Partant de ce premier système d'équation, ABRY *et al.* [42] définissent une transformation linéaire basée sur la transformée de PARK [44]. Cette transformée, appelée transformée A-T, consiste en une matrice de transformation Λ permettant de transformer le vecteur des débits massiques entrant dans chacune des chambres (notés q_{mN} et q_{mP}) et q_{mA} et q_{mT} nommés respectivement débit massique actif et débit massique de pressurisation. Cette matrice est définie comme suit :

$$\begin{bmatrix} q_{mA} \\ q_{mT} \end{bmatrix} = \Lambda(y) \cdot \begin{bmatrix} q_{mP} \\ q_{mN} \end{bmatrix} \quad (3.3)$$

et $\Lambda(y)$ s'écrit :

$$\Lambda(y) = V_0 \cdot \begin{bmatrix} \frac{1}{V_P} & -\frac{1}{V_N} \\ \frac{1}{V_P} & \frac{1}{V_N} \end{bmatrix} \quad (3.4)$$

avec V_0 le volume total du vérin. De ce fait, nous avons :

$$\det(\Lambda(y)) = \frac{2 \cdot V_0^2}{V_P(y) \cdot V_N(y)} \neq 0, \forall y \in]-\frac{l}{2}; \frac{l}{2}[\quad (3.5)$$

garantissant donc l'inversibilité de cette matrice quelle que soit la position du piston. Par la suite en utilisant cette transformée sur les pressions p_P et p_N , nous obtenons le système d'équation suivant :

$$\left\{ \begin{array}{l} \frac{dy}{dt} = v \\ \frac{dv}{dt} = \frac{-b \cdot v + F_{pneu}}{M} \\ \frac{dF_{pneu}}{dt} = -K_{pneu} \cdot v + B_1 \cdot q_{mA} \\ \frac{dK_{pneu}}{dt} = \frac{f_K(y, v, K_{pneu}, q_{mA}, q_{mT})}{V_N(y) \cdot V_P(y)} \end{array} \right. \quad (3.6)$$

Ici, K_{pneu} est la raideur pneumatique et se définit comme suit :

$$K_{pneu}(y) = \left(\frac{p_P}{V_P(y)} + \frac{p_N}{V_N(y)} \right) \cdot k \cdot S^2 \quad (3.7)$$

et f_K comme suit :

$$f_K(y, v, K_{pneu}, q_{mA}, q_{mT}) = A_1 \cdot K_{pneu} \cdot y \cdot v - A_2 \cdot F_{pneu} \cdot v - B_2 \cdot y \cdot q_{mA} + B_3 \cdot q_{mT} \quad (3.8)$$

Dans cette fonction les constantes A_1 , A_2 , B_1 , B_2 et B_3 sont définies comme suit :

$$\begin{aligned} A_1 &= 2 \cdot S^2 \cdot (k + 1) & A_2 &= S^2 \cdot k \cdot (k + 1) \\ B_1 &= \frac{k \cdot R \cdot T \cdot S}{V_0} & B_2 &= \frac{S^3 \cdot k^2 \cdot T \cdot R}{V_0} \\ B_3 &= S^2 \cdot k^2 \cdot T \cdot R \end{aligned} \quad (3.9)$$

L'avantage de cette transformée est particulièrement visible dans l'équation (3.6). En effet, contrairement à l'équation (3.1), une fois la transformée A-T utilisée, le débit virtuel q_{mA} est maintenant clairement identifié comme le moyen de contrôler l'effort pneumatique généré. Toutefois, le modèle présenté ci-dessus reste un modèle nécessitant la présence de deux servovalves, une pour contrôler q_{mA} et une pour contrôler q_{mT} . À ce sujet, HERZIG *et al.* [45] précise qu'il est possible de n'utiliser que le débit massique q_{mA} et ainsi n'utiliser qu'une seule servovalve pour contrôler la position ou l'effort pneumatique généré par le vérin. Pour cela, les auteurs démontrent que la dynamique de K_{pneu} , habituellement contrôlée avec q_{mT} , reste bornée si nous ne la contrôlons plus. Le système d'équation que nous utiliserons donc par la suite est donc le suivant :

$$\left\{ \begin{array}{l} \frac{dy}{dt} = v \\ \frac{dv}{dt} = \frac{-b \cdot v + F_{pneu}}{M} \\ \frac{dF_{pneu}}{dt} = -K_{pneu} \cdot v + B_1 \cdot q_{mA} \end{array} \right. \quad (3.10)$$

3.2 Contrôle du vérin

Une fois la modélisation du vérin correctement définie, il est possible de s'intéresser à son contrôle à proprement parler. Il s'agit de simuler une seringue et nous avons pour cela choisi d'utiliser deux lois de commandes différentes. L'une des deux lois sera en charge de maintenir le piston du vérin en position. Cette loi sera utilisée pour garder le vérin en position lorsque l'utilisateur n'applique aucun effort dessus. La seconde loi, elle, sera en charge de produire un effort pneumatique donné. Elle

sera active lorsque l'utilisateur appliquera un effort sur le piston. Elle aura donc pour objectif de produire les effets haptiques souhaités. Nous présenterons aussi la méthode utilisée pour déterminer la valeur de tension de commande transmise à la servovalve.

3.2.1 Contrôle en position

Nous allons ici nous pencher sur la loi de commande en position (Figure 3.2) utilisée pour le vérin pneumatique lorsque l'utilisateur n'appuie pas dessus et qu'il faut donc garder le piston en position. Nous présenterons sa synthèse et sa stabilité asymptotique.

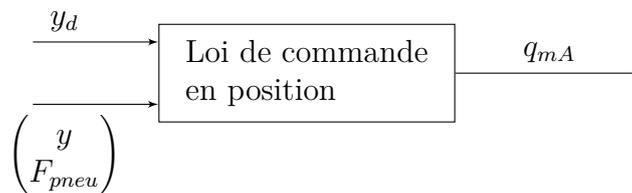


FIGURE 3.2 – Schéma bloc de la loi de commande en position

Synthèse de la commande

Pour le contrôle en position, nous avons repris la commande développée par ABRY [42] et utilisée par la suite par HERZIG *et al.* [13]. Cette loi de commande utilise le modèle à une entrée présentée dans l'équation (3.10) et se définit grâce à une approche par *backstepping*. Pour commencer, nous définissons donc une trajectoire en position que l'on note y_d . Partant de cette trajectoire, on calcule les dérivées suivantes : la vitesse v_d , l'accélération a_d et le jerk j_d . Nous définissons ensuite l'effort pneumatique souhaité F_d à l'aide de l'équation suivante :

$$F_d = M \cdot (a_d + z_1 \cdot (C_1^2 - 1) - z_2 \cdot (C_1 + C_2)) + b \cdot v \quad (3.11)$$

Une fois ces valeurs souhaitées définies, il est nécessaire de définir un certain nombre de signaux d'erreur afin d'appliquer la loi de commande. Ces signaux sont définis comme suit :

$$\begin{cases} z_1 &= y - y_d \\ z_2 &= v - v_d + C_1 \cdot z_1 \\ z_3 &= F_{pneu} - F_d \\ z_{3i} &= \int z_3 dt \end{cases} \quad (3.12)$$

Il est alors possible de définir la valeur de q_{mA} à appliquer au vérin afin d'atteindre la position y_d souhaitée. Il faut bien sûr s'assurer du fait que cette trajectoire doit nécessairement C^3 sur \mathbb{R}_+^1 afin de pouvoir en définir les trois dérivées

1. Une fonction C^3 sur \mathbb{R}_+ est au moins trois fois dérivable et ses trois dérivées sont continues sur \mathbb{R}_+ .

nécessaires à son fonctionnement. Nous obtenons alors la commande q_{mA} suivante :

$$q_{mA} = f_0 + f_1 \cdot z_1 + f_2 \cdot z_2 + f_3 \cdot z_3 + f_{3i} \cdot z_{3i} \quad (3.13)$$

avec :

$$\begin{aligned} f_0 &= \frac{M^2 \cdot j_d + M \cdot K_{pneu} \cdot v - v \cdot b^2 + F_{pneu} \cdot b}{M \cdot B_1} \\ f_1 &= -\frac{M \cdot (C_1^3 - 2 \cdot C_1 - C_2)}{B_1} \\ f_2 &= \frac{M^2 \cdot (C_1^2 + C_1 \cdot C_2 + C_2^2 - 1) - 1}{M \cdot B_1} \\ f_3 &= -\frac{C_1 + C_2 + C_3}{B_1} \\ f_{3i} &= \frac{K_{int}}{B_1} \end{aligned}$$

Dans ces équations, l'ensemble des C_i ainsi que K_{int} sont des constantes à définir et sont nécessairement positives.

Cette loi de commande présente un avantage très agréable pour notre application en proposant de régler la raideur en boucle fermée, K_{cl} , ainsi que l'amortissement en boucle fermée, B_{cl} du système contrôlé. Cela offre un meilleur contrôle sur le ressenti qu'a l'utilisateur lorsqu'il presse la tige du piston et que la loi de commande en position est active. Dans leurs travaux ABRY *et al.* [42] définissent ces variables comme suit :

$$\begin{aligned} K_{cl} &= M \cdot (C_1 \cdot C_2 + 1) \\ B_{cl} &= M \cdot (C_1 + C_2) \end{aligned} \quad (3.14)$$

Ces deux variables, fixées par les constantes C_1 et C_2 , permettent donc de personnaliser le ressenti haptique mais aussi de simplifier la mise en œuvre de la loi de commande en donnant des significations physiques aux constantes d'une loi de commande par backstepping.

Stabilité asymptotique

En appliquant la loi de commande ci-dessus au vérin, ainsi que le choix de la trajectoire en effort le système d'équation (3.1) devient :

$$\begin{cases} \frac{dz_1}{dt} = -C_1 \cdot z_1 + z_2 \\ \frac{dz_2}{dt} = -z_1 - C_2 \cdot z_2 + \frac{z_3}{M} \\ \frac{dz_3}{dt} = -\frac{z_2}{M} - C_3 \cdot z_3 \end{cases} \quad (3.15)$$

Pour la suite nous poserons $Z = [z_1, z_2, z_3]$, en posant la fonction de LYAPOUNOV suivante :

$$V_p(Z) = \frac{1}{2} \cdot z_1^2 + \frac{1}{2} \cdot z_2^2 + \frac{1}{2} \cdot z_3^2 \quad (3.16)$$

Nous avons donc bien $V_p(Z) \geq 0, \forall Z \in \mathbb{R}^3$ et $V_p(Z) = 0$ si et seulement si $Z = [0, 0, 0]$. De plus la dérivée s'écrit :

$$\dot{V}_p(Z) = \dot{z}_1 \cdot z_1 + \dot{z}_2 \cdot z_2 + \dot{z}_3 \cdot z_3 \quad (3.17)$$

soit avec l'équation (3.15) :

$$\dot{V}_p(Z) = -C_1 \cdot z_1^2 - C_2 \cdot z_2^2 - C_3 \cdot z_3^2 \quad (3.18)$$

Nous avons donc bien $\dot{V}_p(Z) \leq 0, \forall Z \in \mathbb{R}^3$ et $\dot{V}_p(Z) = 0$ si et seulement si $Z = [0, 0, 0]$ et on prouve ainsi la stabilité asymptotique de cette loi de commande.

3.2.2 Contrôle en effort

Nous allons maintenant nous intéresser à la seconde loi de commande utilisée sur le vérin pneumatique, à savoir la loi de commande en effort (Figure 3.3). Cette loi est en charge de reproduire les rendus haptiques souhaités et en particulier de reproduire l'effet ressenti par la perte de résistance.

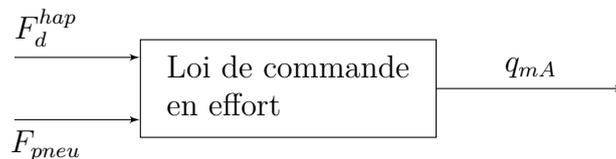


FIGURE 3.3 – Schéma bloc de la loi de commande en effort

Dans un premier temps et afin de produire ces effets, trois approches ont été testées [46] :

- une loi de commande par *backstepping*,
- une loi de commande par *sliding mode*,
- une loi de commande par *backstepping* générant une raideur K_{cl} variable.

Nous avons choisi de garder la première solution. En effet la solution par *sliding mode* donne aussi de bon résultats, mais lors de nos essais avec des médecins, les problèmes de *chattering*, inhérents à un fonctionnement par *sliding mode* se sont révélés gênants. Quant à la commande par *backstepping* générant une raideur en boucle fermée variable, cette loi de commande permet bien de reproduire une

perte de résistance mais sa mise en place était extrêmement complexe et n'offre pas suffisamment de personnalisation de la perte de résistance. En effet, celle-ci ne permettait que de produire des pertes de résistance très douces. Le temps de réponse en K_{cl} n'étant pas assez court, il était impossible de produire une perte de résistance très soudaine. Nous nous intéresserons donc ici à la loi de commande par *backstepping* permettant de générer une trajectoire en effort donnée. Pour plus de détail sur les autres approches il est possible de se référer à l'article [46] présenté à la conférence IFAC 2017.

Synthèse de la loi de commande

Pour cette loi de commande, nous ne cherchons à contrôler que la force pneumatique F_{pneu} produite par le vérin et nous ne nous intéressons qu'à l'équation suivante :

$$\frac{dF_{pneu}}{dt} = -K_{pneu} \cdot v + B_1 \cdot q_{mA} \quad (3.19)$$

Nous allons donc chercher à obtenir une valeur de q_{mA} permettant de ramener F_{pneu} vers une valeur F_d^{hap} souhaitée. Il faut donc fournir une trajectoire en effort F_d^{hap} . Une fois celle-ci définie nous utilisons l'erreur z_4 définie comme :

$$z_4 = F_{pneu} - F_d^{hap} \quad (3.20)$$

Nous choisissons alors q_{mA} tel que :

$$q_{mA} = \frac{1}{B_1} (K_{pneu} \cdot v + \dot{F}_d^{hap}) - C_4 \cdot z_4 \quad (3.21)$$

avec C_4 une constante strictement positive choisie par l'utilisateur. Nous remarquons que le débit massique actif ainsi défini nécessite la connaissance de la dérivée temporelle de la trajectoire en effort. Il faudra donc s'assurer que lorsque cette loi de commande est utilisée, la trajectoire en effort souhaité sera au moins \mathcal{C}^1 .

Stabilité asymptotique

En remplaçant cette valeur de q_{mA} dans l'équation (3.19) nous obtenons alors l'équation suivante :

$$\frac{dz_4}{dt} = -C_4 \cdot B_1 \cdot z_4 \quad (3.22)$$

Ainsi en posant :

$$V_e(z_4) = \frac{1}{2} \cdot z_4^2 \quad (3.23)$$

nous vérifions bien que $V_e(z_4) \geq 0, \forall z_4 \in \mathbb{R}$ et $V_e(z_4) = 0$ si et seulement si $z_4 = 0$. De plus :

$$\dot{V}_e(z_3) = \dot{z}_3 \cdot z_3 \quad (3.24)$$

soit, en utilisant l'équation (3.22) :

$$\dot{V}_e(z_3) = -C_4 \cdot B_1 \cdot z_3^2 \quad (3.25)$$

ce qui donne $\dot{V}_e(z_3) \geq 0, \forall z_3 \in \mathbb{R}$ et $\dot{V}_e(z_3) = 0$ si et seulement si $z_3 = 0$ garantissant donc la stabilité asymptotique du système contrôlé.

3.2.3 Algorithme de sélection de la loi de commande

Afin de permettre au vérin de se comporter à la manière d'une seringue de perte de résistance, il a donc été choisi d'utiliser à tour de rôle les deux lois de commande définies précédemment. Il a donc fallu définir un algorithme permettant de sélectionner la loi de commande à utiliser selon le cas et faire en sorte que leurs sélections successives n'entraînent pas d'instabilité du système.

Définition de l'algorithme

Dans une volonté de garder le système final le plus compact et léger possible, aucun capteur d'effort n'a été ajouté au vérin pour détecter la force d'appui sur le piston par l'utilisateur (Figure 2.2). De ce fait il nous faut donc n'utiliser que les informations à notre disposition pour définir la loi de commande à sélectionner. En amont de l'algorithme complet, le système est initialisé pour que le piston du vérin soit en position sortie. Pour cela il est possible de sélectionner la loi de commande en position et fixer la position y_d de sorte à sortir le piston ou alternativement, il est possible de forcer l'utilisation de la loi de commande en effort et de fixer un effort pneumatique donné (positif) ce qui aura pour effet de sortir le piston complètement. Le schéma Figure 3.4 présente schématiquement le cas où la loi de commande choisie au départ est la loi de commande en position. Dans ce schéma, t_{del} est une constante représentant un temps d'attente et y_s correspond à la position pour laquelle la tige du vérin est complètement sortie. Enfin, y_{past} correspond au signal de position y retardé de t_{del} .

Une fois l'algorithme implémenté le schéma bloc du système contrôlé devient celui présenté en Figure 3.5

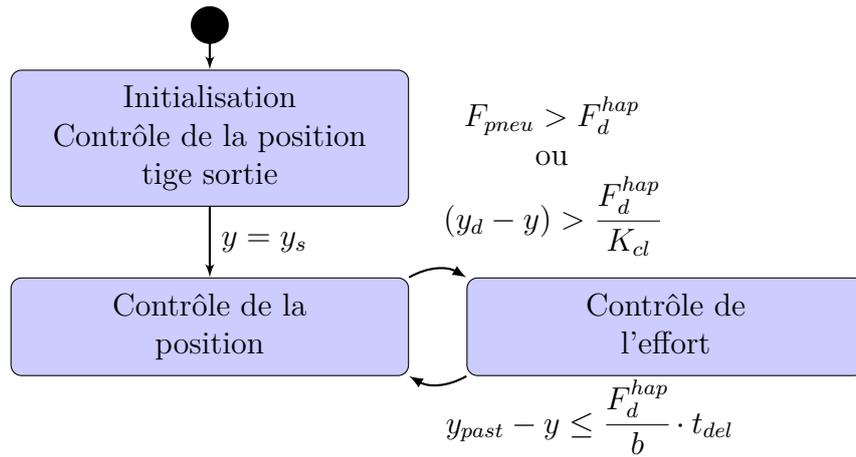


FIGURE 3.4 – Diagramme d'état simplifié de l'algorithme gouvernant le choix de la loi de commande à utiliser

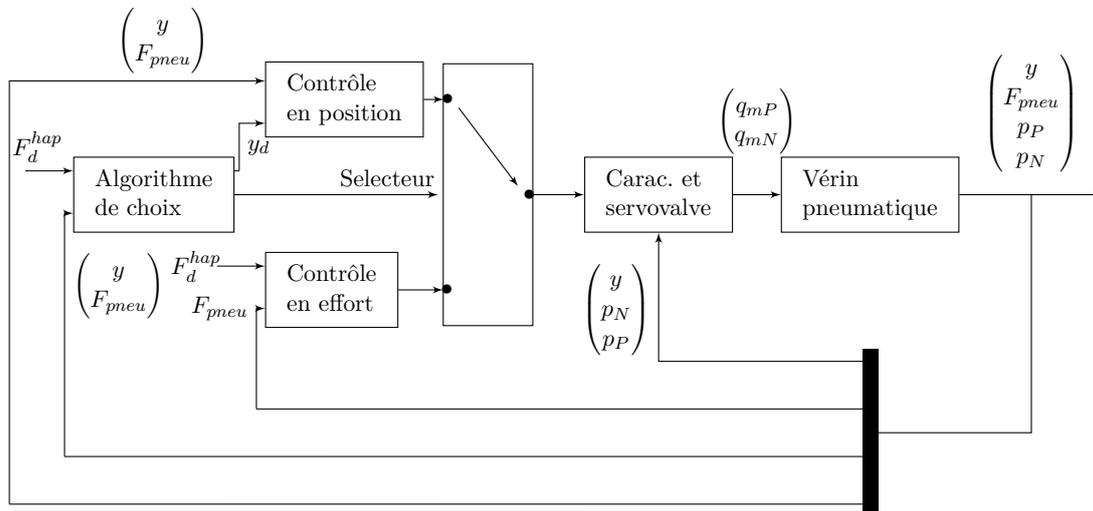


FIGURE 3.5 – Schéma bloc de l'architecture du contrôle du vérin pneumatique

Nous allons commencer en expliquant la logique de sélection mise en œuvre afin de définir l'algorithme complet. Nous présenterons en même temps certains exemples de son fonctionnement obtenus en simulation.

Passage de commande en position à commande en effort

Dans le cas où la loi de commande active est la commande en position, l'utilisateur n'appuie pas sur le piston. Dans ce cas l'algorithme définit, quoiqu'il advienne, la trajectoire d'effort souhaité F_d^{hap} qui correspond à la résistance que nous souhaitons faire ressentir à l'utilisateur. Cet effort dépend de la couche physique traversée à cet instant par l'aiguille. Afin de détecter une utilisation, et donc de passer sur la loi de commande en effort, l'algorithme vérifie donc si l'effort pneumatique généré est

supérieur à cette valeur de F_d^{hap} . Ce qui donne donc la condition suivante :

$$F_{pneu} \geq F_d^{hap} \quad (3.26)$$

Si cette condition est validée l'algorithme assure passage de la loi de commande en position à la loi de commande en effort, et ce avec le moins de vibrations possible pour l'utilisateur car l'effort alors produit par le vérin correspond exactement à l'effort haptique souhaité. A cette condition a été ajouté une seconde condition qu'il est possible d'interpréter comme une condition de sécurité. Cette condition est la suivante :

$$(y_d - y) > \Delta y_{p \rightarrow f} \quad (3.27)$$

avec :

$$\Delta y_{p \rightarrow f} = \frac{F_d^{hap}}{K_{cl}} \quad (3.28)$$

En effet, lorsque la commande en position est active, ABRY *et al.* [42] définit une raideur boucle fermée K_{cl} ainsi qu'un amortissement en boucle fermée B_{cl} . Ainsi en considérant que le système est quasi statique, et en négligeant donc l'importance de B_{cl} , il est possible d'associer le fonctionnement du système masse ressort amortissement. Dans ce cadre, cette seconde condition vérifie si la position courante de la tige du vérin dévie trop fortement de la position souhaitée au point de devoir générer un effort correspondant à F_d^{hap} . Il s'agit cependant d'une sécurité car elle permet surtout de forcer un passage à la loi de commande en effort si jamais la loi de commande en position mettrait trop de temps à réagir à une pression de l'utilisateur.

Un exemple du fonctionnement de cette condition de sécurité est présenté en Figure 3.6 et permet de voir que si cette condition de sécurité est active, le sursaut en effort ressenti restera relativement faible pour l'utilisateur.

Passage de commande en effort à commande en position :

Ce passage doit avoir lieu lorsque l'utilisateur cesse d'appliquer une pression sur la tige du vérin ou lorsque son effort est trop faible. Il faut en effet lui permettre de mettre en pause son mouvement s'il le souhaite. Cela permet aussi aux personnes ayant pour habitude d'appliquer de la force par «à-coups» (cf. Chapitre 1 et [33]) de le faire s'ils le souhaitent. Comme le système final ne dispose pas de capteur d'effort il faut donc se contenter dans ce cas de la donnée en position.

Lorsque le contrôle en effort est actif et que l'utilisateur relache son effort sur la tige, le vérin doit alors suivre le modèle suivant :

$$\begin{cases} \frac{dy}{dt} = v \\ \frac{dv}{dt} = \frac{-b \cdot v + F_d^{hap}}{M} \end{cases} \quad (3.29)$$

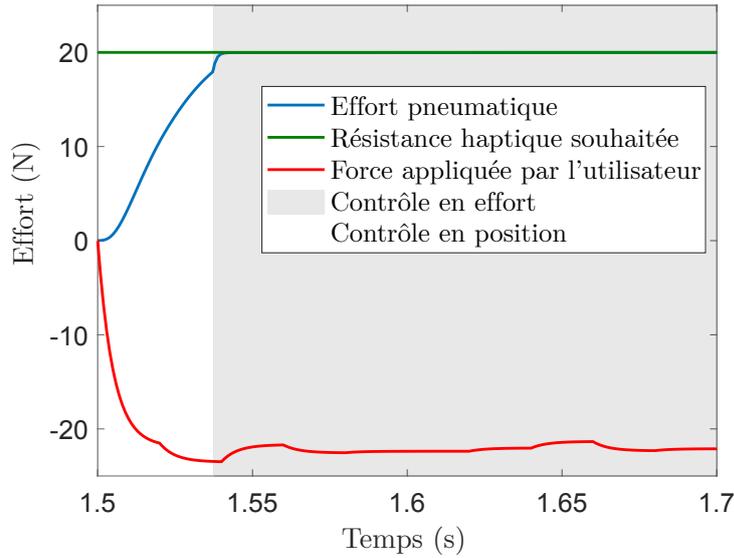


FIGURE 3.6 – Simulation présentant : la force appliquée par un utilisateur, l'effort pneumatique produit par le vérin et la résistance haptique souhaitée F_d^{hap} . Dans ce cas, la condition de sécurité basée sur la position est déclenchée en premier créant un petit sursaut de la force pneumatique produite à $t \approx 1.55$ s

Dans le cas où l'utilisateur relâche la pression sur la tige du vérin nous allons considérer F_d^{hap} constant pour simplifier les calculs. Ainsi, lorsque la tige n'est plus maintenue, l'effort pneumatique généré a pour effet de faire reculer cette dernière. Au moment où la tige est lâchée, elle a donc une vitesse nulle. A partir de cet instant, considéré comme $t = 0$, la vitesse de la tige sera donc solution de l'équation différentielle (3.29) et peut donc s'écrire :

$$v(t) = \frac{F_d^{hap}}{b} \cdot (1 - e^{-\frac{b}{M} \cdot t}) \quad (3.30)$$

ce qui donne donc :

$$y(t) - y_0 = \frac{F_d^{hap}}{b} \cdot (t + \frac{M}{b} e^{-\frac{b}{M} \cdot t}) \quad (3.31)$$

où y_0 correspond alors à la position à laquelle la tige a été relâchée. Nous pouvons donc fixer un délai t_{del} après lequel on souhaite repérer le recul et en choisissant ce délai tel que $t_{del} \ll \frac{M}{b}$ on a donc :

$$y(t_{del}) - y_0 = \frac{F_d^{hap}}{b} \cdot t_{del} \quad (3.32)$$

Il est donc possible d'utiliser la condition $y(t) - y(t + t_{del}) \geq \frac{F_d}{b} \cdot t_{del}$ pour savoir quand effectuer notre changement. Il faudra cependant ne pas utiliser une valeur

trop faible ou trop importante pour t_{del} . Dans notre cas, une valeur de $t_{del} = 0.1s$ s'est révélée satisfaisante. Si cette valeur est trop faible, l'algorithme pourra avoir tendance à être trop sensible et inversement. Des exemples du manque de sensibilité sont présentés en Figure 3.7.

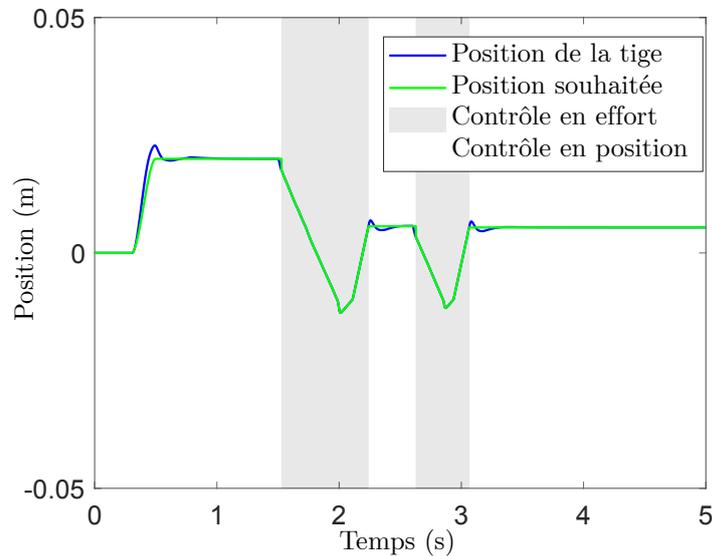


FIGURE 3.7 – Simulation présentant : la position y la position souhaitée y_d utilisant un temps t_{del} trop important, retardant la détection par l'algorithme de la nécessité de changer de loi

Par la suite il faudra fournir une position d'équilibre souhaitée pour la loi de commande en position. Pour cela, deux solutions sont envisageables :

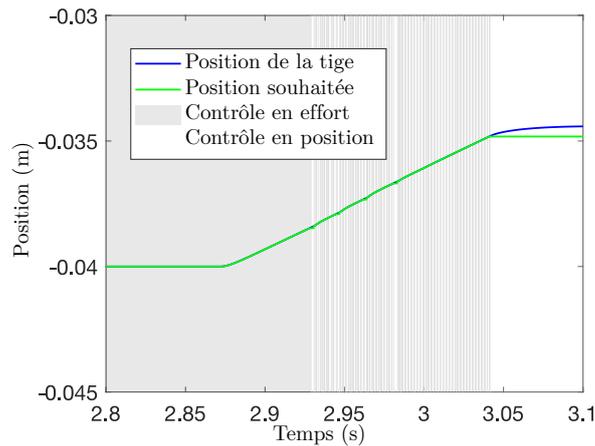
- choisir ce qui correspondrait à y_0 , le moment où l'on pense que l'utilisateur a relâché la pression sur la tige ;
- choisir la position à laquelle le changement de loi a lieu.

La première solution permet à l'utilisateur de retrouver la tige exactement à la position lorsqu'il a cessé d'appuyer. Cependant pendant un court instant après le changement de loi, la tige va se repositionner et donc avancer sans que l'utilisateur n'appuie dessus. Cela peut paraître irréaliste pour l'utilisateur. La seconde solution limite ce changement de position à un court temps de retour à l'équilibre de la loi de commande en position. C'est donc la solution qui a été choisie et implémentée.

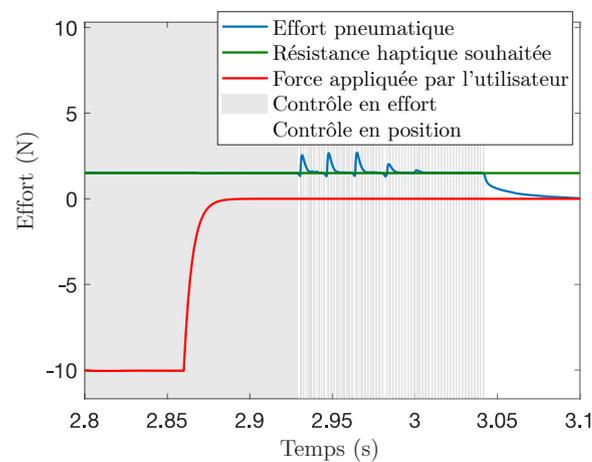
Gestion du changement de loi de commande et stabilité

Une fois les conditions de changement de loi de commande établies, il est intéressant de se pencher sur l'effet de ces changements de lois sur le fonctionnement du système. En effet sans poser de limitation particulière au changement de loi de commande, l'algorithme peut avoir tendance à effectuer des changements de lois trop fréquents ou successifs ce qui peut avoir un effet négatif pour l'utilisateur.

Un exemple d'un tel fonctionnement est présenté en Figure 3.8. Dans cet exemple, aucune restriction n'est mise en place pour limiter le nombre de changements de lois de commande au cours du temps. Cela a pour effet de produire une succession de changements de lois en un temps très réduit. Si cela ne semble pas déboucher sur une instabilité du système, cela reste gênant pour l'utilisateur qui pourra ressentir une vibration due au fait que l'algorithme «hésite» entre deux lois de commande pendant cette période de temps.



(a)



(b)

FIGURE 3.8 – Simulation présentant des cas où aucune restriction n'est mise en place pour limiter le nombre de changements de loi de commande

Pour remédier à cela et assurer la stabilité asymptotique de notre système, nous avons mis en place une temporisation basée sur la théorie de temps de stabilisation ou *dwell time*. D'après HESPANHA *et al.* [47], il est possible d'assurer la stabilité d'un tel système pourvu que les sous-systèmes soient stables et que la commutation soit suffisamment lente. Afin de définir la vitesse limite de cette commutation, il faut s'intéresser aux temps caractéristiques des sous-systèmes.

Dans notre cas nous avons deux sous-systèmes :

$$\begin{cases} \frac{dz_1}{dt} = -C_1 \cdot z_1 + z_2 \\ \frac{dz_2}{dt} = -z_1 - C_2 \cdot z_2 + \frac{z_3}{M} \\ \frac{dz_3}{dt} = -\frac{z_2}{M} - C_3 \cdot z_3 \end{cases} \quad (3.33)$$

quand la loi de commande en position est active et

$$\begin{cases} \frac{dy}{dt} = v \\ \frac{dv}{dt} = \frac{-b \cdot v + F_{pneu}}{M} \\ \frac{dz_4}{dt} = -C_4 \cdot B_1 \cdot z_4 \end{cases} \quad (3.34)$$

quand la loi de commande en effort est active, z_4 est assimilée à z_3 pour simplifier. Pour obtenir les temps de stabilisation, nous considèrerons donc les valeurs propres de la matrice :

$$\mathbf{A}_1 = \begin{pmatrix} -C_1 & 1 & 0 \\ -1 & -C_2 & 1/M \\ 0 & -1/M & -C_3 \end{pmatrix} \quad (3.35)$$

pour la loi de commande en position et pour la loi de commande en effort, la stabilité n'est importante que pour l'équation :

$$\frac{dz_3}{dt} = -C_4 \cdot B_1 \cdot z_3 \quad (3.36)$$

et il est donc possible de considérer la matrice :

$$\mathbf{A}_2 = \begin{pmatrix} -C_4 \cdot B_1 & 0 & 0 \\ 0 & -C_4 \cdot B_1 & 0 \\ 0 & 0 & -C_4 \cdot B_1 \end{pmatrix}, \quad (3.37)$$

En effet, z_1 et z_2 sont des signaux d'erreur en position et en vitesse qui ne sont pas affectés par le contrôle de l'effort pneumatique. Il est donc possible de les fixer arbitrairement à 0 en choisissant volontairement $y_d = y$ et $v_d = v$ lorsque la loi de commande en effort est active. Ainsi, d'après HESPANHA *et al.* [47], il est possible de définir un temps τ_D à patienter entre deux commutations. Ce temps ne sera alors dépendant que des valeurs de C_1 , C_2 , C_3 et C_4 . En ajoutant cet effet de «temporisation» dans l'algorithme, nous obtenons le logigramme présenté en Figure 3.9.

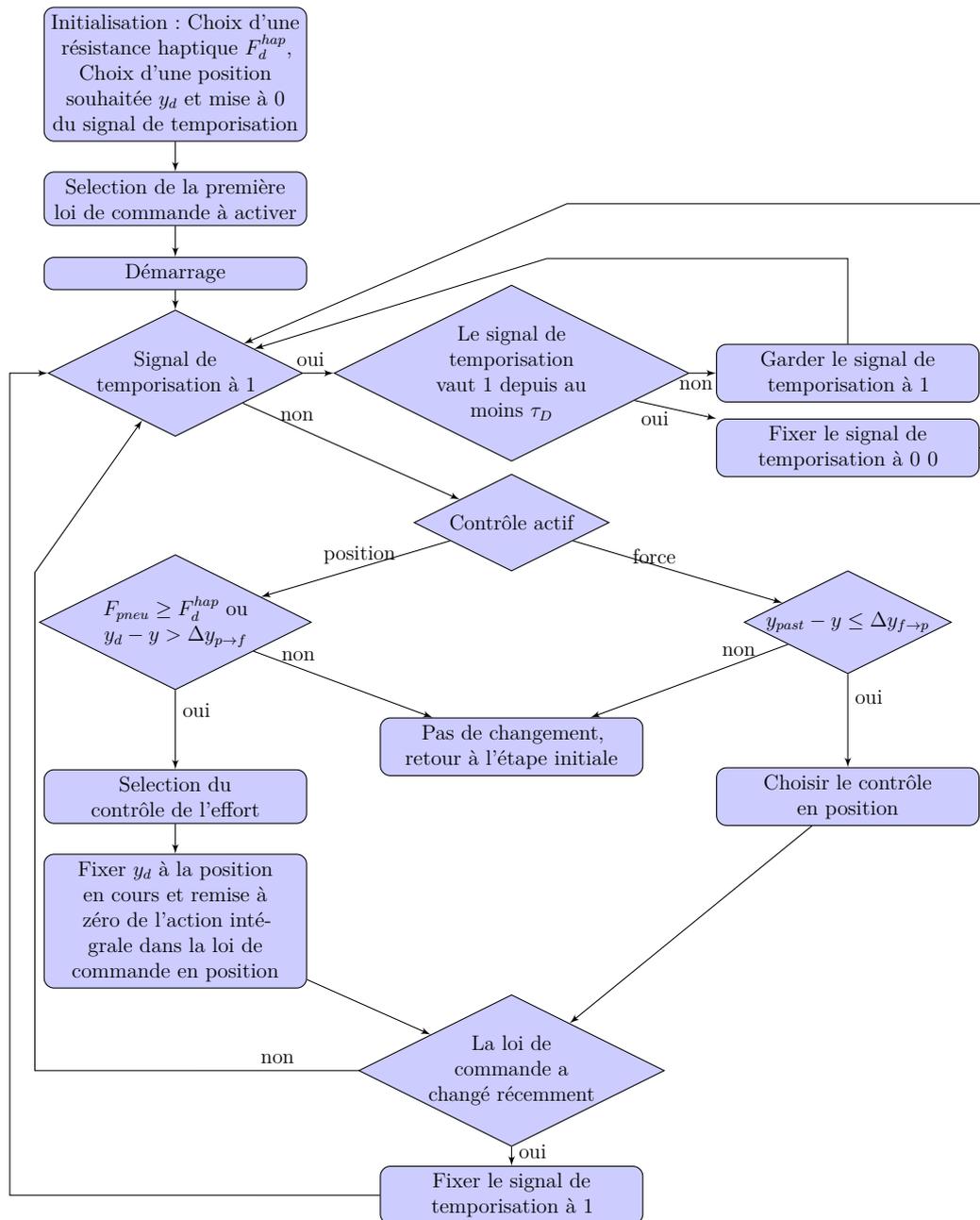


FIGURE 3.9 – Logigramme complet de l’algorithme de contrôle du vérin pneumatique

3.3 Implémentation et essais

3.3.1 Implémentation sur banc d'essai dédié

L'interface a pu être testée sur un banc d'essai dédié. Ce banc d'essai est présenté en Figure 3.10. Il est constitué d'un vérin pneumatique double action Airpel® modèle M16D100D contrôlé avec une servovalve Festo MPYE®-5-M5-010 B. Ce vérin a une course de 100 mm et 16 mm de diamètre interne. Pour ce qui est des capteurs, nous avons utilisé deux capteurs de pression MEAS® U5136 et un capteur de position linéaire à faible frottement (LVDT) modèle DC-EC 2000 de *Measurement Specialties*. Le tout est contrôlé grâce à un contrôleur dSPACE (DS1103) cadencé à 1 kHz.

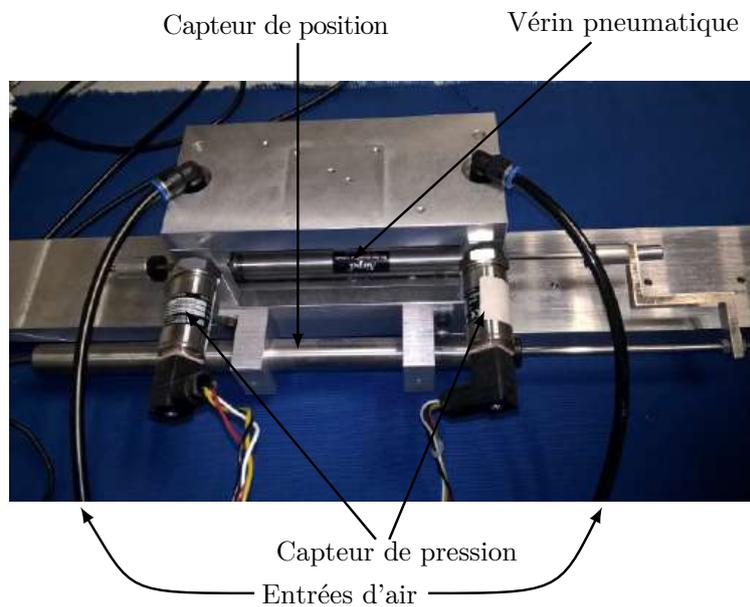


FIGURE 3.10 – Banc d'essai utilisé pour tester la simulation de la seringue

Les valeurs des paramètres sont disponibles dans la Table 3.1. Ce choix assure $\tau_D \approx 0.1$ s d'après [47].

TABLE 3.1 – Paramétrage des lois de commandes

K_d	B_d	C_3	C_4	K_i
1000 N/m	50 N/(m.s)	20	$4 \cdot 10^{-4}$	10

Concernant le choix de K_d , en faisant l'hypothèse que $F_d^{hap} = 10$ N, celui-ci assure un changement de loi de commande après un déplacement d'environ 1 cm. Pour le choix du paramètre B_d , il est nécessaire de vérifier une condition mathématique supplémentaire énoncée par ABRY *et al.* [43] afin d'être en mesure de définir la loi

de commande en position par *backstepping*, à savoir :

$$B_{cl} \geq 2\sqrt{M \cdot (K_{cl} - M)} \quad (3.38)$$

Il est d'ailleurs souhaitable de choisir une valeur relativement proche de cette limite. En effet, utiliser une valeur trop grande pour B_{cl} a tendance à rendre le système sensiblement oscillant. De plus, dans cette application, les mouvements sont relativement lents et l'influence de B_{cl} reste marginale.

Lors de nos premiers essais, nous avons choisi de démarrer l'algorithme en activant la loi de commande en effort en premier. Dans ces tests, nous générions une trajectoire en effort simple en choisissant un point de la course du vérin correspondant à la perte de résistance. Afin de tester les possibilités de personnalisation du système nous avons généré trois trajectoires en effort différentes pour représenter plusieurs types de patients, à savoir :

- patient «facile» : perte de résistance très soudaine et effort maximum relativement faible (équivalent d'un patient standard) ;
- patient «intermédiaire» : perte de résistance soudaine et effort maximum relativement élevé (équivalent d'un patient calcifié) ;
- patient «difficile» : perte de résistance douce et effort maximum relativement élevé (équivalent d'un patient en sur-poids).

Afin de créer une perte de résistance soudaine ou rapide, nous utilisons un filtre du second ordre dont nous choisissons le temps de réponse pour reproduire un patient donné.

Finalement, afin de définir la tension de commande à envoyer à la servovalve, nous avons recours à la table de données issues d'une caractérisation de la servovalve et d'un algorithme développé par ABRY *et al.* [42]. Grâce à cette caractérisation, il est possible de connaître la valeur de tension de commande à envoyer connaissant le volume actuel des chambres du vérin (d'après la position du piston) et la pression dans chacune des chambres. La caractéristique utilisée est présentée en Figure 3.11.

3.3.2 Résultats

L'utilisation de ce banc d'essai a permis de tester le fonctionnement de l'algorithme et de valider l'approche choisie. Les tests ont été réalisés avec les deux médecins travaillant sur le projet. La Figure 3.12 présente un cas «facile» et représente une situation simple où l'utilisateur n'a fait qu'appliquer une pression constante sur la tige du vérin jusqu'à reconnaître la perte de résistance à $t \approx 25$ s. La Figure 3.14 présente un cas «intermédiaire» avec une petite hésitation de l'utilisateur aux alentours de $t \approx 35$ s. À ce moment là l'algorithme passe temporairement sur la loi de commande en position pour quelques secondes. Enfin, la Figure 3.13 présente le cas «difficile» qui est aussi une situation simple à l'instar de l'essai présenté en Figure 3.12.

Dans l'ensemble de ces essais, le réalisme de la simulation s'est révélé tout à fait satisfaisant, d'après les anesthésistes experts travaillant sur le projet. Il est toutefois

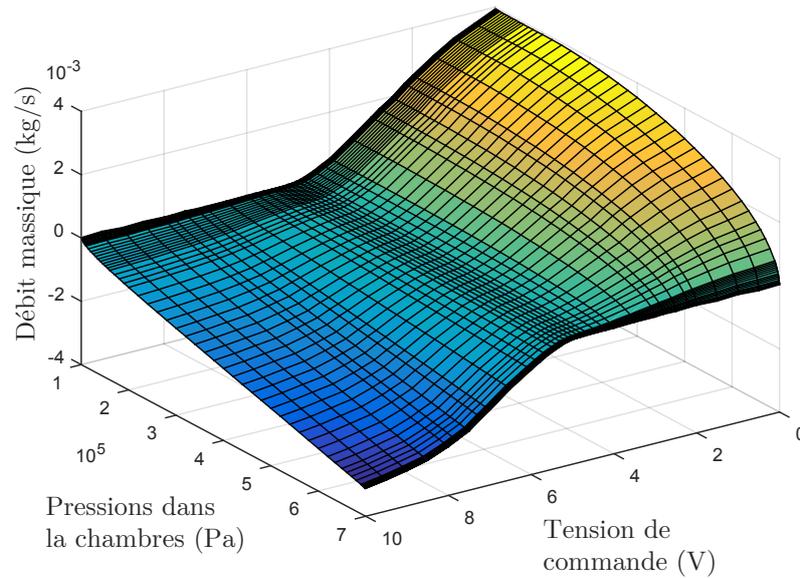


FIGURE 3.11 – Caractérisation entrée sortie de la servovalve

possible de relever des saccades dans le mouvement de la tige comme par exemple dans la Figure 3.12 aux alentours de $t \approx 12$ s. Ces saccades sont majoritairement imputables à la présence de frottements secs dans le vérin qui ne se retrouveront pas sur le vérin utilisé sur le simulateur complet. En suppléments à ces mesures, l'Annexe B présente quelques mesures supplémentaires réalisées lors d'une campagne d'essais.

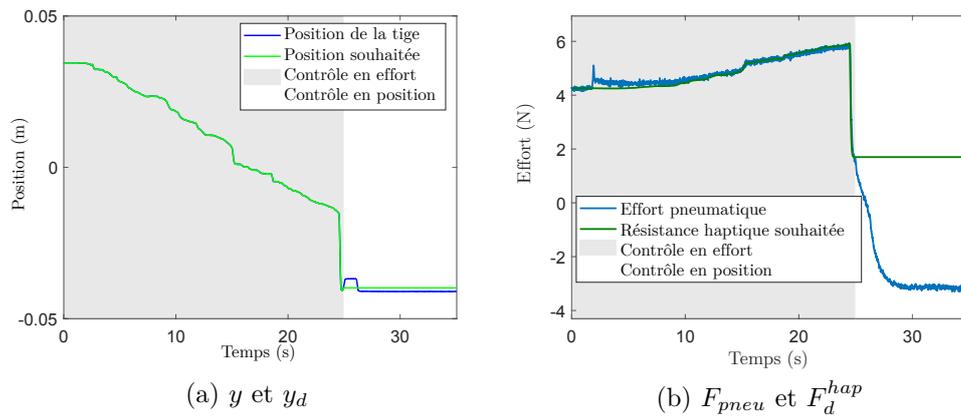


FIGURE 3.12 – Résultats expérimentaux obtenus en simulant un patient «classique»

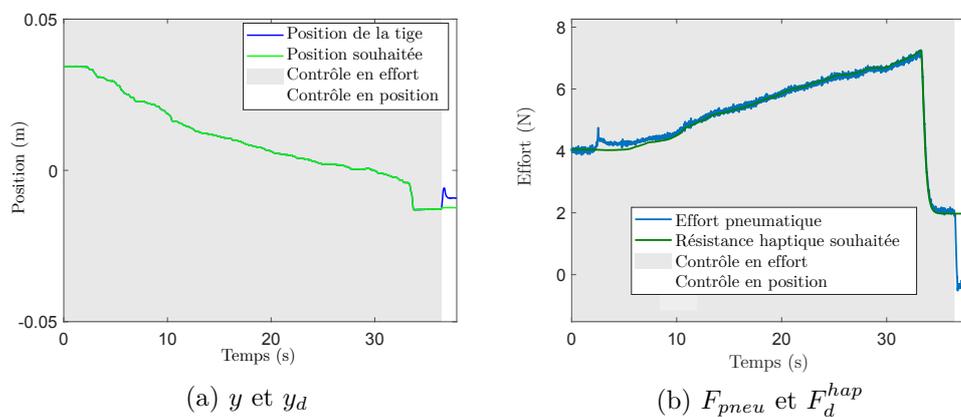
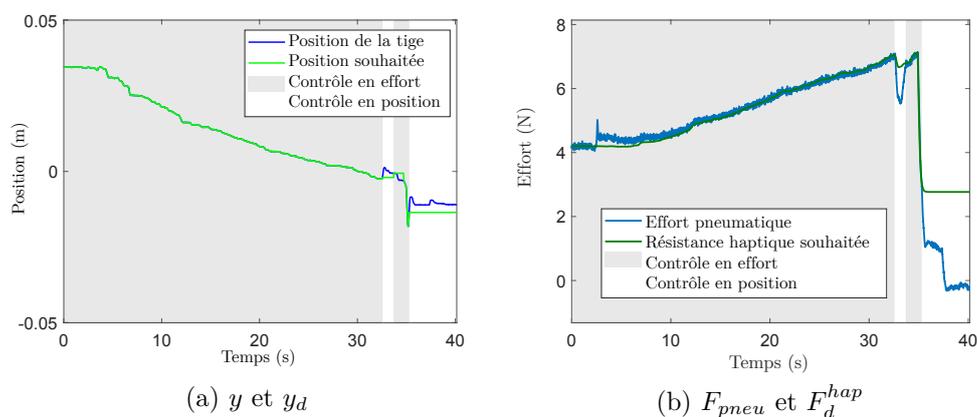


FIGURE 3.13 – Résultats expérimentaux présentant un patient «difficile»

FIGURE 3.14 – Résultats expérimentaux présentant un patient «intermédiaire» et une hésitation de l'utilisateur à $t \approx 35$ s

Conclusion

Dans ce chapitre, nous avons présenté la mise en œuvre d'une interface haptique permettant de reproduire le comportement de la seringue de perte de résistance au cours de la procédure d'anesthésie péridurale. À travers les premiers essais réalisés sur un banc d'essai dédié, les premiers résultats ont permis de valider l'approche et l'utilisation d'un vérin pneumatique pour cette partie de la simulation. Les retours de la part des médecins ont validé le réalisme général de l'interface.

Les lois de commandes et l'algorithme utilisés, permettent de garder le contrôle sur les rendus haptiques produits et il est donc possible de proposer une simulation de la perte de résistance bien plus précise et réaliste que ce qui l'a été par des simulateurs précédents. Il est en effet possible de s'éloigner de la reproduction d'une perte de résistance tout-ou-rien.

Chapitre 4

Reproduire l'insertion de l'aiguille à l'aide d'un bras haptique

Introduction

Dans ce chapitre, nous allons nous intéresser à l'implémentation de la simulation haptique pour reproduire la sensation d'insertion d'une aiguille dans la colonne vertébrale d'un patient. Afin de reproduire les sensations haptiques attendues nous avons eu recours à une interface haptique Virtuose 6D développée par Haption. Il ne s'agit donc pas d'établir le modèle de fonctionnement de l'interface, ni des lois de commande précises de la-dite interface. En effet, tout le travail de modélisation et de contrôle de l'interface et de ses différents degrés de mobilité a été réalisé au préalable par le fabricant qui fournit une API de contrôle de sa machine.

Il sera donc question des choix d'architecture informatiques liés à ce dispositif. Ce chapitre a aussi pour objectif de présenter la modélisation de l'espace traversé par l'aiguille au cours d'une insertion ainsi que des choix d'implémentation des différents rendus haptiques mis en œuvre. L'objectif de l'implémentation est de fournir une simulation satisfaisante au sens de VAUGHAN *et al.* [3] tout en permettant que les modèles utilisés soient facilement adaptables pour de potentielles futures applications.

Le premier point abordé traitera donc de la manière dont sera représenté l'espace intervertébral puis nous nous attarderons plus en détails sur les rendus haptiques précis mis en œuvre dans la simulation.

4.1 Architecture de l'application et modélisation de l'espace intervertébral et de la pointe d'aiguille

Pour commencer, il convient de s'intéresser à la construction de l'application en charge de la simulation de l'insertion de l'aiguille dans l'espace intervertébral. La première partie sera l'occasion de préciser l'architecture logicielle à proprement parler, puis nous préciserons la modélisation des différents éléments constituant l'espace intervertébral.

4.1.1 Architecture logicielle et fonctionnements basiques

Architecture logicielle de la simulation

La solution implémentée utilise l'architecture simplifiée présentée en Figure 4.1 pour l'implémentation de la simulation. Dans ce schéma nous avons mis en avant les interactions principales de l'application ainsi que les noms de classes utilisées. Il est possible de séparer cette architecture en trois grandes parties :

- une partie responsable de l'affichage que verra l'utilisateur final ;
- une partie responsable de la simulation haptique ;
- une classe centrale, ici `CDemo`, faisant office d'interface entre les différents éléments.

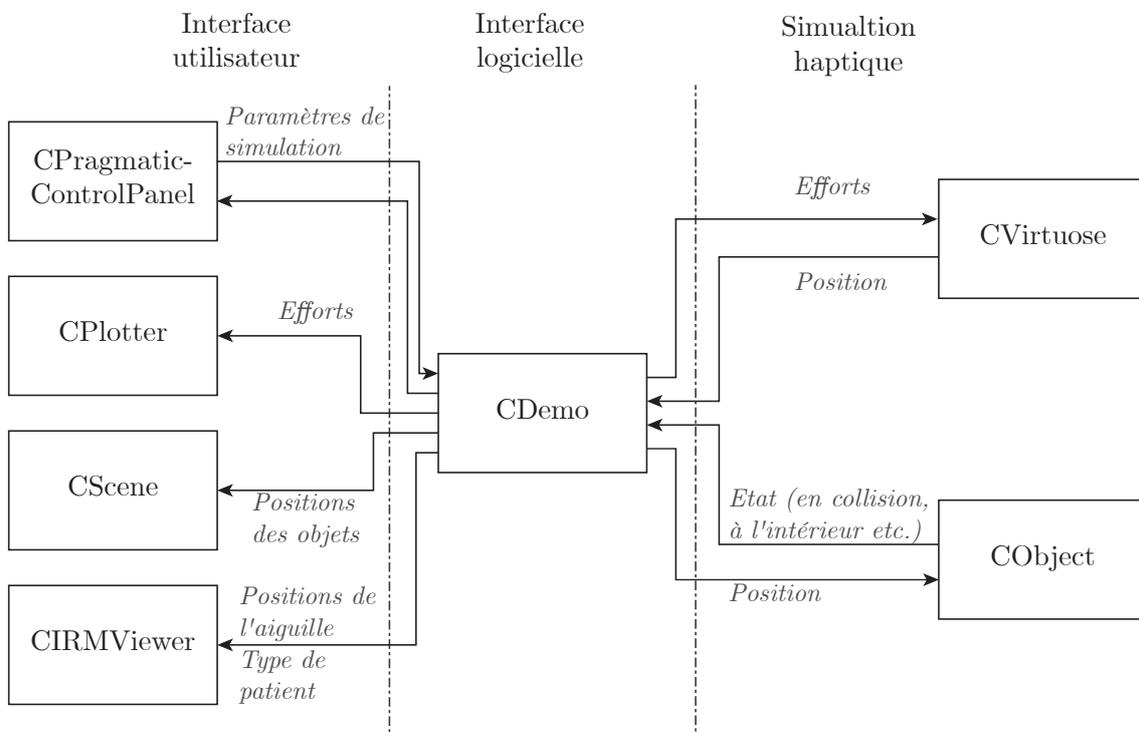


FIGURE 4.1 – Architecture logicielle simplifiée de l'application et des interactions principales

Dans notre architecture, la classe `CDemo` est donc l'élément central en charge de gérer l'ensemble des éléments essentiels de l'application et de servir d'interface pour le reste. Cette classe contient en partie les divers éléments comme les objets représentant l'espace intervertébral ou la pointe de l'aiguille. Ces différents objets sont implémentés sous la forme de classes dérivées de la classe `CObject`. Elle contient aussi la classe assurant la communication avec l'interface haptique (`CVirtuose`). Côté interface utilisateur la classe contient une référence au panneau de configuration (`CPragmaticControlPanel`), une référence à l'utilitaire en charge de tracer les courbes en effort en temps réel (`CPlotter`), la classe gérant la construction de la scène 3D (`CScene` qui représente l'espace intervertébral simulé) et l'utilitaire de visualisation d'images IRM (`CIRMViewer`). La classe `CDemo` est accessible dans l'ensemble de l'application à travers une instance unique qui pourra être appelée pour faire l'interface entre l'interface haptique et la scène 3D par exemple.

En ce qui concerne des interactions, le panneau de configuration `CPragmaticControlPanel` permet de fixer un certain nombre de paramètres qui seront ensuite transmis à la classe centrale. Il reçoit des informations à transmettre à l'utilisateur concernant l'état de la simulation (sauvegarde de fichier de mesure par exemple). La classe `CPlotter` ne fait que recevoir des informations à afficher. La classe `CScene` reçoit des modifications à appliquer à l'environnement (taille des éléments par exemple). Quant à l'utilitaire de visualisation d'IRM, il reçoit les informations concernant la position actuelle de la pointe de l'aiguille ou encore le type de patient à représenter.

Côté simulation haptique, la classe `CVirtuose` reçoit des informations d'efforts à produire et met à disposition la position actuelle de l'interface. Les `CObject` reçoivent les modifications de tailles ou de caractéristiques à modifier et transmettent les événements et les efforts obtenus par la simulation physique.

Cette représentation ne fait pas figurer les utilitaires de gestion des fichiers de mesure qui peuvent intervenir sur l'ensemble des éléments présentés ici en fonction des besoins et de choix de valeurs à enregistrer. Plus de précisions quant à son utilisation seront apportées dans la suite de ce manuscrit.

Simulation physique, position physique et objet virtuel

La plupart du temps les systèmes haptiques ont recours à un système de proxy ou *god-object* dont un aperçu du fonctionnement a été présenté dans le chapitre 1 et rappelé en Figure 4.2. Dans ce type de fonctionnement, l'objet virtuel lié à l'interface sert à explorer simplement un environnement virtuel qui pourra ensuite soumettre cet objet à des efforts ou des contraintes particulières. Dans l'exemple de la Figure 4.2, l'objet virtuel entre par exemple en contact avec un solide au sein de la simulation. L'action de contact n'est bien entendu que l'une des possibilités mais pour les rendus possibles il est nécessaire d'avoir recours à un moteur physique (implémenté par la librairie *Open Source ODE* [48] pour *Open Dynamics Engine*) qui sera en charge de détecter d'éventuelles collisions, d'appliquer une résistance donnée, une pesanteur, etc.

Par la suite, en simulant un mécanisme ressort-amortisseur entre l'interface et l'objet virtuel, la simulation sera en mesure de déterminer l'effort que doit produire

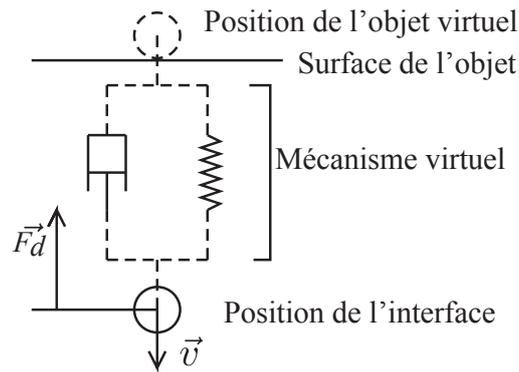


FIGURE 4.2 – Exemple d’un système de proxy où l’utilisateur vient toucher une surface virtuelle

l’interface haptique. Il est donc possible de schématiser ce fonctionnement global par la Figure 4.3.

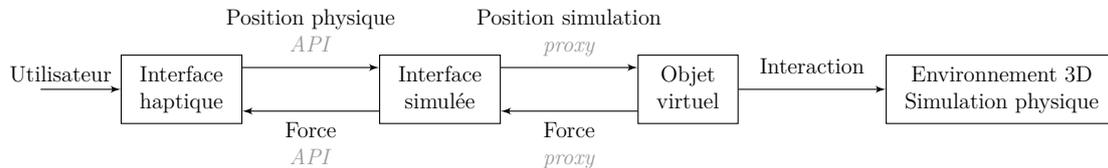


FIGURE 4.3 – Principe de fonctionnement schématique d’un algorithme de proxy / «god-object»

Si le système de proxy a initialement été mis en place pour répondre à la problématique de l’exploration d’une surface en 3D [22], il peut tout à fait être généralisé à toute sorte d’interactions. Il faut toutefois garder à l’esprit que la simulation d’un mécanisme ressort amortisseur agit comme un filtre passe bas sur les trajectoires en effort. Ainsi, si nous souhaitons reproduire des variations soudaines d’efforts il sera nécessaire de s’affranchir de ce proxy et communiquer l’effort souhaité «brut» directement à l’interface.

Concernant la logique générale de la simulation, nous avons donc trois éléments clés, à savoir : la position physique de l’interface, la position de l’objet virtuel lié à l’interface et un environnement de simulation. Partant de ces éléments d’architecture nous allons maintenant pouvoir nous intéresser à l’implémentation des différents rendus haptique nécessaire à notre simulation.

Description de l’environnement 3D

Avant d’aller plus loin, il convient de poser les conventions qui seront utilisées dans la simulation et dans l’implémentation de certains effets. Il est ici particulièrement question des repères utilisés. Il est possible d’en dénombrer cinq cohabitants dans la simulation.

- le premier est le repère global de la simulation ou repère d’environnement. Celui-ci est fixe et le reste tout au long de la simulation ;

- ensuite vient le repère d’observation correspondant au repère de l’écran, défini par rapport au repère de l’environnement 3D ;
- le troisième est celui de la base de l’interface haptique, et donc celui utilisé par l’API de contrôle de l’interface. Celui-ci est défini par rapport au repère d’observation. Ce repère reste fixe dans notre cas car le repère d’observation ne bouge pas ;
- le quatrième correspond au repère outil, celui-ci représente la position de la poignée de l’interface haptique. Ce repère est défini par rapport au repère d’environnement ;
- enfin, le dernier est le repère d’avatar qui correspond à la position de la main de l’utilisateur par rapport au système haptique, et qui permet de prendre en compte la géométrie de l’outil fixé à son extrémité ; il est positionné par rapport au repère outil.

Dans notre cas, le repère d’observation est fixe (la caméra observant le modèle 3D ne bouge pas) nous pouvons donc considérer que le repère de base de l’interface est défini par rapport au repère global directement. Pour commencer, il est plus simple de présenter le positionnement du repère de base de l’interface haptique. Selon la documentation de référence de l’API d’Haption, le repère de base est défini comme suit :

- l’origine se situe au centre du système à retour d’effort, c’est-à-dire au point d’intersection entre l’axe 1 (rotation de la base) et l’axe 2 (premier mouvement vertical) ;
- axe X horizontal, aligné selon la position médiane de l’axe 1 (rotation de la base) et orienté vers l’utilisateur ;
- axe Y horizontal, perpendiculaire à l’axe X et orienté vers la droite de l’utilisateur ;
- axe Z vertical et orienté vers le haut.

Comme ce repère est fixe par rapport au repère global, il convient de le considérer comme repère de base afin de simplifier le propos. Nous pouvons donc utiliser le schéma Figure 4.4 pour résumer la position des différents repères utilisés dans notre cas. La figure Figure 4.8 présente les repères outils et avatar sur le simulateur.

4.1.2 Représentation de l’espace intervertébral

Concernant la représentation de l’espace intervertébral, afin de pouvoir modifier simplement et rapidement la configuration, nous avons choisi de conserver une modélisation simplifiée de l’espace en terme de géométrie. En effet, la très grande majorité des éléments présents est représentée sous la forme d’«espaces» parallélépipédiques et de «boîtes». Un schéma du modèle utilisé est visible en Figure 4.5. Si cette modélisation s’éloigne de la réalité physique, elle permet de modifier très facilement et rapidement les paramètres de simulation et la géométrie de l’espace. De plus comme la vocation première de la simulation est de produire des rendus haptiques,

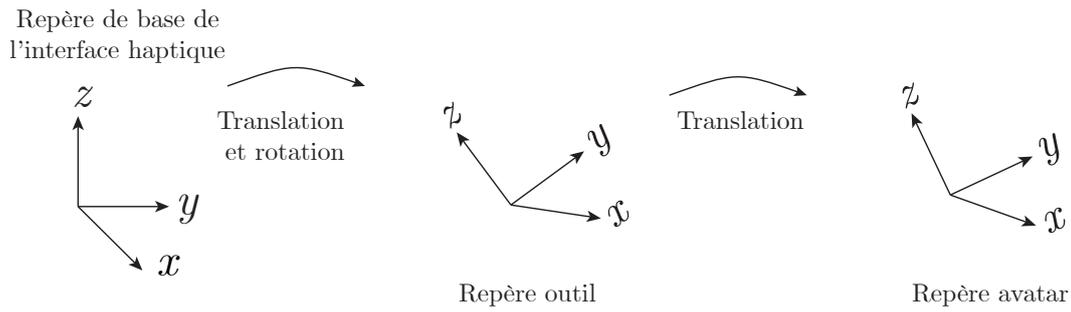


FIGURE 4.4 – Repères utilisés dans la simulation

il n'est pas nécessaire d'être trop précis dans la représentation de l'espace traversé. De plus l'apprenant n'aura pas accès à ce modèle 3D, celui-ci servant uniquement à l'implémentation et au calcul des efforts. Simplifier le modèle permet donc d'assurer un fonctionnement à une fréquence d'échantillonnage suffisamment grande et de simplifier certains algorithmes comme la détection de collisions.

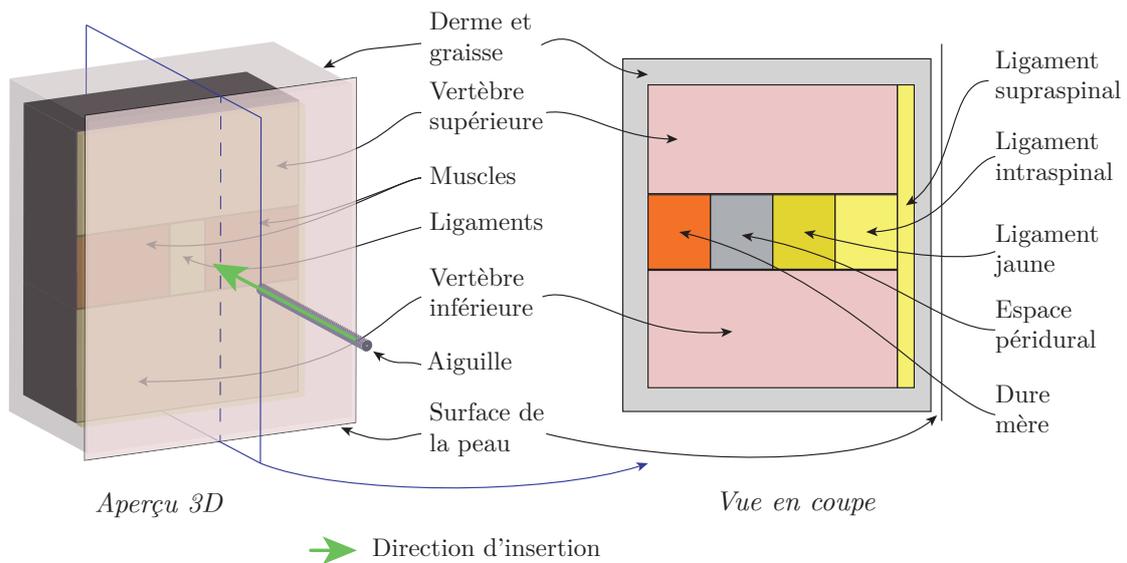


FIGURE 4.5 – Modèle de l'environnement de simulation 3D utilisé pour représenter l'espace intervertébral (les dimensions ne sont pas à l'échelle)

Pour rappel l'objectif est de reproduire les efforts définis dans l'équation (2.2) soit :

$$\vec{F}_{ins} = \vec{F}_{raideur} + \vec{F}_{frottement} + \vec{F}_{coupe} + \vec{F}_{traj} + \vec{F}_{contact} \quad (4.1)$$

Pour ce qui est des éléments utilisés nous avons recours aux classes C++ suivantes :

- `CResistantBoxSpace` : cette classe est utilisée pour représenter le derme et la graisse, les différents ligaments, l'espace périural, la dure mère et les muscles.

Il s'agit d'un parallélépipède rectangle produisant une résistance constante à l'avancée (soit \vec{F}_{coupe} dans l'équation (4.1)). Tout objet pénétrant un tel espace se voit donc appliqué l'effort de l'espace en question :

- **CFrictionBoxSpace** : cette classe est utilisée pour représenter l'effet de *stick-slip* présent en particulier dans le ligament jaune (soit $\vec{F}_{frottement}$ dans l'équation (4.1)). Il s'agit alors de l'utiliser en conjonction avec une **CResistantBoxSpace** afin d'ajouter l'effet de frottement à l'effet de résistance simple ;
- **CShakyBox** : cette classe est utilisée pour représenter les os. C'est là aussi un parallélépipède rectangle permettant de reproduire une sensation de contact dur (soit $\vec{F}_{contact}$ dans l'équation (4.1)) ;
- **CSoftPlane** : cette classe est utilisée pour représenter la surface de la peau. Elle reproduit la sensation de percer la peau au début de l'insertion d'aiguille (soit $\vec{F}_{raideur}$ dans l'équation (4.1)) ;
- **CSoftTrajColBoxSpace** : cette classe est utilisée pour générer la première partie des efforts dûs au suivi d'une trajectoire (soit \vec{F}_{traj} dans l'équation (4.1)). Il s'agit ici des efforts spécifiques qui apparaissent lorsque l'aiguille se situe dans le derme. Ces efforts ayant pour objectif de ramener l'utilisateur à sa trajectoire d'insertion original, ils sont dirigés de l'aiguille vers la trajectoire, perpendiculairement à cette dernière. À ce moment l'utilisateur peut encore choisir de modifier légèrement sa trajectoire, le derme conservant une certaine élasticité. Cette partie de la trajectoire vient ensuite contraster avec l'effort de suivi de trajectoire imposé par les ligaments.

Par la suite, chacun de ces éléments a un effet haptique. L'effet est alors activé différemment selon le type d'objet. S'il s'agit des **CResistantBoxSpace** alors, l'effet est activé lorsque l'objet virtuel attaché à l'interface haptique entre à l'intérieur de l'espace en question. S'il s'agit de la **CShakyBox** ou du **CSoftPlane**, l'effet s'active lorsque l'objet virtuel entre en contact avec la boîte. Nous avons aussi voulu éviter les inconsistances dans l'effort haptique reproduit qui pourraient être générées par une interface imparfaite entre deux espaces. Pour cela les ligaments, le derme, l'espace péri-dural ainsi que la dure mère sont constitués en imbriquant les **CResistantBoxSpace** comme présenté sur le schéma Figure 4.6. Avec ce modèle, la résistance appliquée est alors une somme de résistances. Par exemple en se plaçant dans le ligament intraspinal, en plus de la résistance du ligament concerné l'utilisateur est soumis à la résistance du derme et la résistance du ligament supraspinaux.

4.1.3 Représentation de l'aiguille

En ce qui concerne l'aiguille, la trajectoire adoptée par celle-ci est nécessairement rectiligne. La seule partie pertinente pour la simulation est donc la pointe de l'aiguille et il suffira de représenter cette pointe. Pour cela, nous utilisons une sphère générée par la classe **CNeedleTip**. C'est en utilisant la position de cette sphère par rapport aux autres éléments de la simulation que nous générerons les rendus haptiques souhaités. Dans notre simulation l'objet virtuel sera donc représenté par un objet de type **CNeedleTip** dont le graphe d'héritage est présenté en Figure 4.7.

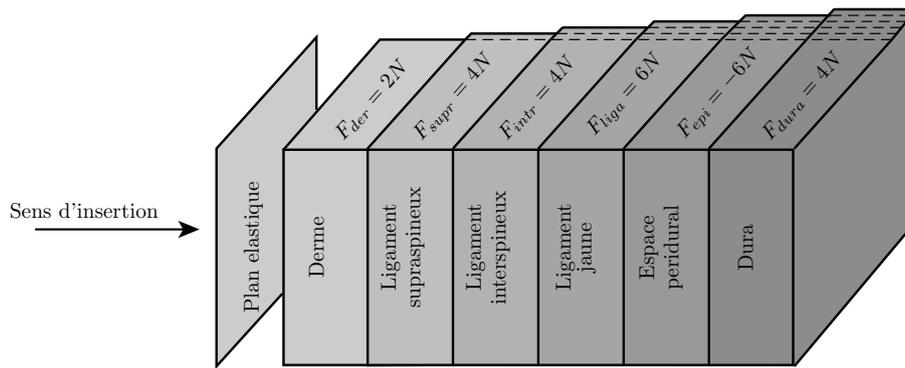


FIGURE 4.6 – Modélisation du derme, ligaments, espace péri-dural et dure mère

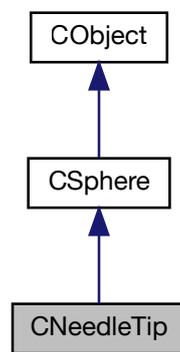


FIGURE 4.7 – Graphe d'héritage de la classe CNeedleTip

Il est aussi nécessaire de faire en sorte que le repère d'avatar se situe au niveau de la pointe de l'aiguille une fois celle-ci attachée à l'interface. Pour cela, nous translatons donc le repère de l'outil pour obtenir le repère d'avatar souhaité. Un exemple de ce positionnement est présenté sur la Figure 4.8. L'objectif est de faire en sorte que lorsque la pointe de l'aiguille est à la surface de la peau, le centre de rotation de la tige du robot corresponde à la pointe de l'aiguille dans l'environnement virtuel. En déplaçant le repère avatar, nous permettons donc de prendre en compte le fait que la pointe de l'aiguille ne correspond pas directement à la position de la poignée mais se trouve décalée à la surface de la peau comme montrée sur la Figure 4.8. Le repère avatar étant par la suite dépendant du repère outil, il est possible de choisir simplement son angle d'insertion sans pour autant faire bouger le point d'insertion car on fait alors en sorte que le centre de rotation se situe au niveau du point d'insertion défini.

4.2 Détails d'implémentation des rendus haptiques

4.2.1 Restriction à une trajectoire rectiligne

La restriction à une trajectoire rectiligne est réalisée en deux parties distinctes. En effet, lorsque la pointe de l'aiguille se situe dans le derme ou dans un muscle, il est nécessaire de représenter une trajectoire relativement «douce». Le but est de

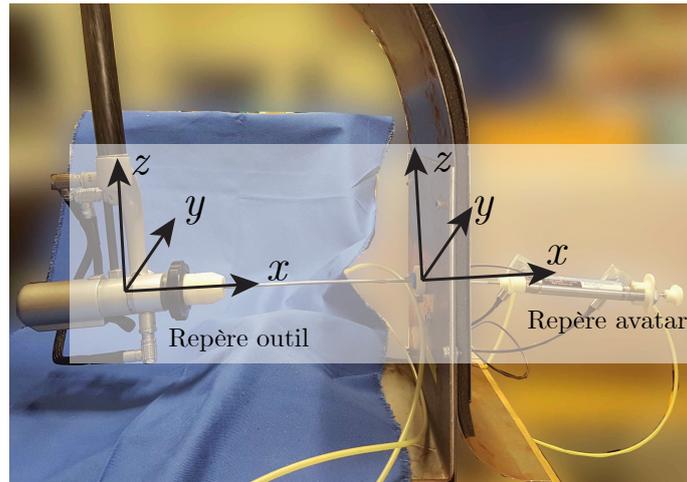


FIGURE 4.8 – Positionnement du repère avatar par rapport au repère outil

représenter les muscles et le derme comme des zones relativement élastiques et donc permettre à l'utilisateur de se réorienter légèrement dans ces espaces. Pour cela, lorsque la pointe de l'aiguille pénètre dans le derme ou les muscles pour la première fois, l'orientation de l'aiguille à ce moment est enregistrée pour servir de référence sous la forme d'un quaternion et la position du point d'insertion, ce qui permet de définir une première droite d'insertion. Par la suite, nous utilisons un correcteur proportionnel afin de générer un effort pour ramener la position la pointe de l'aiguille vers la droite d'insertion d'origine et l'orientation de l'aiguille vers l'orientation originale. L'effort peut donc s'écrire comme suit :

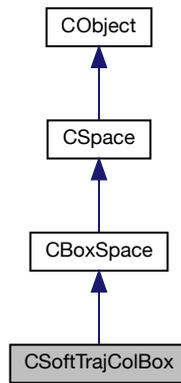
$$\vec{F}_{traj} = K_{soft} \cdot \frac{\overrightarrow{OH}}{\|\overrightarrow{OH}\|} \quad (4.2)$$

où K_{soft} représente le coefficient du correcteur et donc une raideur, O est la position courante de l'objet virtuel et H son projeté sur la trajectoire rectiligne initiale. L'effort ainsi calculé est ensuite appliqué à l'objet virtuel attaché à l'interface. Ensuite concernant l'orientation de la poignée nous utilisons la valeur suivante pour la génération d'un couple :

$$\vec{C}_{traj} = C_{soft}^k \cdot \vec{z} \quad (4.3)$$

où C_{soft}^k est un coefficient positif, \vec{z} le vecteur attaché à la poignée et \vec{d} le vecteur directeur de la droite trajectoire. Au niveau implémentation, nous avons recours à un espace de type `CSoftTrajColBoxSpace` dont le graphe d'héritages est présenté en Figure 4.9.

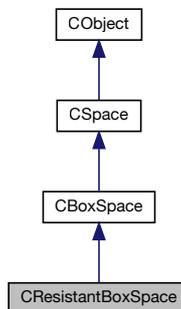
En effet, en pénétrant dans la peau nous générons une trajectoire «souple» autour' de laquelle l'utilisateur peut naviguer pour modifier légèrement son angle d'insertion, puis lorsque la pointe de l'aiguille finit par atteindre le premier ligament, la pointe de l'aiguille est alors bloquée par le mécanisme virtuel. Par la suite lorsque

FIGURE 4.9 – Graphe d’héritage de la classe `CSoftTrajColSpace`

la pointe de l’aiguille pénètre le premier ligament, nous utilisons une fonctionnalité de l’API de contrôle de l’interface haptique qui permet de contraindre la trajectoire de l’aiguille de manière plus rigide. Cela a pour effet de limiter les déplacements de la pointe de l’aiguille à ceux permis par une liaison glissière entre la pointe de l’aiguille et la trajectoire. Le fonctionnement à l’aide de proxy génère automatiquement les efforts haptiques nécessaires. La position de la pointe de l’aiguille dans l’environnement virtuel étant contraint, si la position physique de l’interface s’éloigne de cette trajectoire, un effort est généré. La trajectoire est générée au moment où la pointe de l’aiguille pénètre dans les ligaments la première fois. Nous fixons la direction à l’aide du quaternion d’orientation de l’aiguille à ce moment précis et la position lors de la première insertion.

L’avantage d’utiliser ces deux types de trajectoire permet à l’utilisateur d’influer un peu sur la trajectoire finale lorsqu’il se situe dans le derme comme c’est le cas en pratique.

4.2.2 Résistance des couches physiques

FIGURE 4.10 – Graphe d’héritage de la classe `CResistantBoxSpace`

Les `CResistantBoxSpace` (Figure 4.10) permettent de générer un effort constant opposé à la direction de déplacement de la pointe de l’aiguille. Cet effort est donc

appliqué à l'objet virtuel et son expression mathématique est la suivante :

$$\vec{F}_{coupe} = -F_{couche} \cdot \frac{\vec{v}}{\|\vec{v}\|} \text{ si } \|\vec{v}\| \neq 0 \quad (4.4)$$

Ici, \vec{v} correspond à la vitesse instantanée de l'objet virtuel. Cette vitesse est calculée par la librairie ODE en temps réel et elle est ramenée à zéro si elle devient trop faible. F_{couche} correspond à une constante définie par la couche physique traversée. En outre l'effort à produire est filtré à l'aide d'un filtre de premier ordre sur l'ensemble de ses directions. L'ajout de ce simple filtre est nécessaire pour éviter les sursauts d'efforts et changements de résistance abrupts indésirables lors de l'insertion de l'aiguille.

4.2.3 Représentation du contact dur

Dans notre cas, si la pointe de l'aiguille entre en contact avec un os, celle-ci va se planter dans l'os et le médecin sera alors contraint de reculer. Il faut donc dans un premier temps empêcher le glissement de la pointe de l'aiguille sur la surface de l'os ce que permet de faire la librairie ODE.

Pour la représentation d'un contact dur, la simulation utilise deux fonctionnalités principales. Dans un premier temps, la plus grande partie de la simulation utilise simplement le fonctionnement par proxy pour générer un effort correspondant à un appui constant sur la surface. Un schéma de ce fonctionnement est présenté en Figure 1.8 . Avec ce principe, il est possible de faire de l'exploration de surface. Cependant les contacts ont tendance à paraître «mous» car la raideur du mécanisme virtuel est limitée.

Ensuite, et pour donner l'impression d'un contact dur sur l'os, nous avons eu recours à la méthode développée par KUCHENBECKER *et al.* [23] que nous avons adaptée en 3D. La méthode proposée dans ces travaux consiste à générer une trajectoire en effort particulière qui se déclenche au moment du contact. Pour cela, au moment du premier contact entre la pointe de l'aiguille et l'os, nous générons la trajectoire à appliquer puis, pendant les échantillons de temps suivants, la trajectoire en effort est appliquée cette fois directement à l'interface haptique, sans passer par le mécanisme de proxy. Celle-ci est fondée sur la version sinusoidale décroissante proposée par KUCHENBECKER *et al.* [23] et s'écrit comme suit :

$$\vec{F}_{contact}(t) = A \cdot e^{\log(0.01) \cdot \frac{t}{d}} \cdot \sin(2 \cdot \pi \cdot f \cdot t) \cdot \vec{v}_{in} \text{ pour } 0 \leq t \leq d \quad (4.5)$$

Ici, A , d , f sont des constantes positives définies dans [23]. Leurs valeurs ont pu être obtenues par mesures réalisées lors d'une collision avec une surface dure. Dans notre application nous avons utilisé les valeurs obtenues en enregistrant l'accélération d'une pointe lors d'un contact avec du bois. Ce qui donne les valeurs suivantes : $A = 15.9 \text{ N.s/m}$, $d = 0.055 \text{ s}$ et $f = 55 \text{ Hz}$. Ce rendu haptique particulier a été implémenté dans la classe `CShakyBox` (Figure 4.11). Un aperçu de la trajectoire en effort produite par l'équation (4.5) est présenté en Figure 4.12.

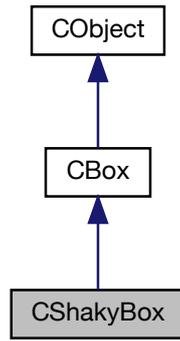


FIGURE 4.11 – Graphe d’héritage de la classe CShakyBox

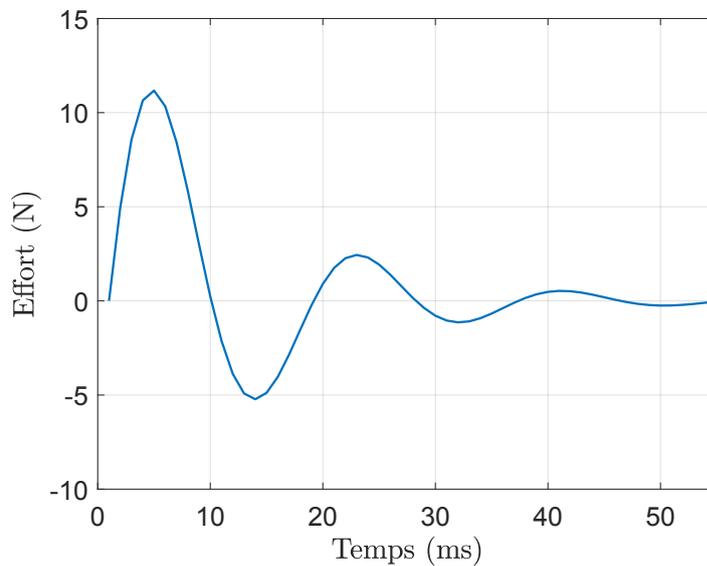


FIGURE 4.12 – Trajectoire en effort aidant à la représentation du contact dur [23]

4.2.4 Représentation du perçage de la peau

À l’instar de l’effort de contact, la force représentant le perçage d’un tissu (ou perçage de cupule) est un retour haptique lié à un évènement discret et nécessitant d’être représenté avec une dynamique relativement importante. En effet l’objectif est de simuler le perçage de la peau qui intervient très soudainement. Schématiquement, il s’agit de reproduire un effort croissant sur quelques millimètres pour ensuite faire apparaître une rupture brutale. Dans notre cas, ce retour haptique n’est utilisé que lors du perçage de la peau. Nous avons développé la classe `CSoftPlane` (Figure 4.13). Il s’agit d’un plan déformable qui déclenche le rendu haptique lorsque la pointe de l’aiguille entre en contact avec celui-ci. L’effort appliqué a la forme suivante :

$$\vec{F}_{raideur} = (K_{peau}^a + K_{peau}^b \cdot OP) \cdot \vec{OP}, \text{ si } 0 < OP < d_{per} \quad (4.6)$$

où K_{peau}^a est une raideur spécifique du matériau, K_{peau}^b une constante positive, O la position de la pointe de l’aiguille, P le projeté orthogonal de O sur le plan de

la peau et d_{per} , la distance après laquelle le perçage de la peau a lieu . Il s'agit donc d'une raideur (K_{peau}^a) d'une part et d'un effet similaire à une raideur (K_{peau}^b), appliquée dans une portion limitée de l'espace. Dans notre cas, $K_{peau}^a = 40$ N/m et $K_{peau}^b = 4000$ N/m² et la distance de perçage est choisie telle que $d_{per} = 5mm$. Ces valeurs sont adaptée de l'étude de OKAMURA *et al.* [34] qui ont constaté une variabilité assez importante selon l'échantillon testé. Nous avons donc choisi des valeurs cohérentes avec ces travaux, permettant de rester relativement réaliste.

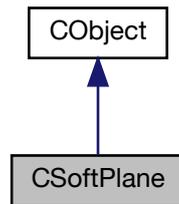


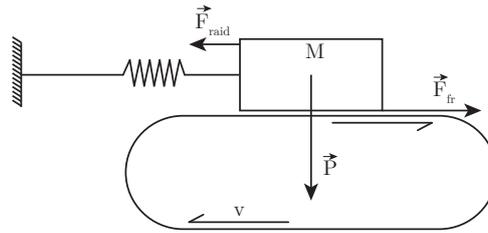
FIGURE 4.13 – Graphe d'héritage de la classe CSoftPlane

Dans le modèle, ce plan est positionné en avant par rapport à l'ensemble des espaces CResistantBoxSpace représentant le dermes et les ligaments de telle sorte à ce que l'utilisateur ne pénètre le premier espace qu'au moment où la perçage a lieu. De cette manière, tant que l'utilisateur n'a pas percé la peau, il ne ressent que cet effort de perçage.

4.2.5 Représentation du frottement sec de type *stick-slip*

Le dernier rendu haptique spécifique nécessaire dans la simulation reproduit un effet de frottements secs de type *stick-slip*. Contrairement à l'ensemble des rendus haptiques présentés précédemment, cet effet n'est pas présent sur tous les types de patients. Le phénomène apparaît lorsque l'aiguille pénètre un ligament jaune calcifié. Cela est souvent le cas sur les personnes âgées mais un effet similaire peut aussi apparaître sur des patients jeunes pratiquant du sport relativement fréquemment.

Une simulation de cet effet a pu être proposée dans des simulateurs développés précédemment comme celui présenté par MAGILL *et al.* [9]. Cependant, il n'est pas possible de réutiliser ces travaux directement car ceux-ci tirent profit de l'architecture et du contrôle spécifique de leur simulateur haptique (robot à câbles) pour reproduire l'effet, en imposant un mouvement précis aux moteurs électriques responsables des retours haptiques. Ce n'est donc pas directement transposable dans notre cas. Afin de développer une simulation du phénomène, nous sommes donc revenus aux modèles existants. Ce phénomène provient de l'existence de deux coefficients de frottements différents lorsque deux objets concernés sont en mouvement l'un par rapport à l'autre ou non. Nous avons alors un coefficient de frottement μ_a si les deux éléments concernés sont en adhérence et un coefficient μ_g si l'un des objets est en mouvement par rapport à un autre. Généralement, lorsqu'il s'agit de modéliser ce phénomène le type de mécanisme utilisé est celui présenté en Figure 4.14.

FIGURE 4.14 – Mécanisme servant à l'étude du phénomène *stick-slip*

Le principe de ce dispositif expérimental est de placer une masse M sur un tapis roulant et de la fixer à un ressort. Ainsi, il est possible de produire l'effet souhaité. Il y a alors deux possibilités :

- la masse glisse sur le tapis. Dans ce cas, la masse est soumise à une force de frottement sec telle que $F_{fr} = \mu_g \cdot M \cdot g$, ainsi qu'à l'effort produit par le ressort ;
- la masse est immobile par rapport au tapis. Elle a alors une accélération nulle car le tapis défile à vitesse constante. La masse est alors soumise à l'effort du au ressort auquel s'oppose une force de frottement sec.

Les efforts tangentiels appliqués par le tapis sur la masse :

$$F_{fr} = \begin{cases} k \cdot x & \text{si } k \cdot x < \mu_a \cdot M \cdot g \\ \mu_g \cdot M \cdot g & \text{si } k \cdot x > \mu_g \cdot M \cdot g \end{cases} \quad (4.7)$$

Dans l'équation (4.7) la première ligne correspond à une phase où la masse est «collée» au tapis, dans ce cas la force de frottement entre le tapis et la masse vient compenser la force due à la raideur pour permettre à la masse d'avoir une accélération nulle. Le passage en glissement a lieu lorsque la force produite par la raideur surpasse l'effort tangentiel maximum, à savoir $\mu_a \cdot M \cdot g$. La seconde ligne correspond elle à une phase de glissement. Dans ce cas la force de frottement est constante et vaut $\mu_g \cdot M \cdot g$. Durant cette phase, l'effort produit par la raideur est supérieur à l'effort tangentiel dû aux frottements et la masse se déplace donc vers son point d'attache. Pour repasser sur une phase où la masse est collée au tapis, il faut alors que l'effort dû à la raideur diminue suffisamment pour que $k \cdot x > \mu_g \cdot M \cdot g$. Nous obtenons alors une alternance entre des phases d'adhérence et des phases de glissement. Ainsi pour reproduire ce phénomène et en particulier la sensation haptique qu'il produit, nous nous sommes basés sur ce modèle en y apportant certaines modifications.

En effet, dans notre cas, il n'est pas question de tapis roulant pour induire l'adhérence et le glissement, toutefois avec le mécanisme de proxy et le mouvement induit par l'utilisateur, il est possible d'obtenir sensiblement le même mécanisme. Pour cela nous avons recours à un espace de type `CFrictionBoxSpace` (Figure 4.15) qui aura pour but de générer les efforts. De manière schématique, le mécanisme mis en œuvre est celui présenté en Figure 4.16. Le fonctionnement de ce mécanisme est globalement inversé par rapport au fonctionnement avec un tapis roulant présenté en

Figure 4.14 du fait que c'est la poignée de l'interface haptique qui applique la vitesse à l'ensemble. Cette vitesse n'est donc pas appliquée directement à l'objet subissant les frottements.

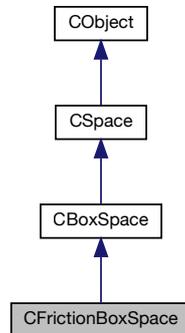


FIGURE 4.15 – Graphe d'héritage de la classe CFrictionBoxSpace

Dans un premier temps l'idée était de se rapprocher au plus du phénomène réel en bloquant le mouvement de la pointe de l'aiguille pour reproduire la phase d'adhérence et en relâchant ce blocage pour reproduire la phase de glissement. Cependant ce fonctionnement n'était pas acceptable dans notre simulation. En effet, le phénomène de *stick-slip* doit être appliqué en plus d'un certain nombre d'autres interactions. Or bloquer complètement les mouvements de la pointe de l'aiguille a pour effet de couper la simulation physique pour cet objet. Cela a pour principale conséquence de produire un rendu haptique haché et non réaliste. Nous avons donc pris le parti de ne jamais bloquer complètement le mouvement de la pointe de l'aiguille mais plutôt d'ajouter un effort supplémentaire. De plus du fait de l'architecture de la simulation et du fonctionnement de son moteur physique, l'effet de *stick-slip* est simplement un rendu supplémentaire, produisant un effort supplémentaire et il n'est pas possible de déduire facilement l'effort de frottement à appliquer. La conséquence principale est qu'il n'est pas possible d'utiliser une condition de transition entre une phase de glissement et une phase d'adhérence en se basant uniquement sur l'effort comme c'est le cas en réalité. En effet, si l'on souhaite bloquer réellement l'objet dans notre simulation, celui-ci ne peut plus être soumis à aucun effort à cause du fonctionnement spécifique du moteur physique de l'application. Il devient alors impossible de vérifier simplement la condition de passage d'un état d'adhérence à un état de glissement.

Nous avons donc choisi de baser la condition de transition soit sur l'effort soit sur la vitesse de l'objet. L'objectif étant de tenter de reproduire une sensation de craquement sans pour autant coller parfaitement au phénomène physique réel. Ainsi si l'objet se déplace lentement, nous appliquons un effort maximal correspondant à une phase d'adhérence. En revanche, si l'objet se trouve dans une phase d'adhérence, nous déterminons si nous basculons sur une phase de glissement en étudiant cette condition :

$$K_{proxy} \cdot d \geq F_{adh} \quad (4.8)$$

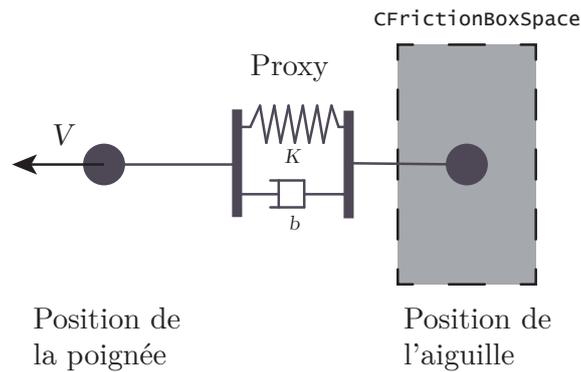


FIGURE 4.16 – Mécanisme équivalent pour représenter le phénomène de *stick-slip* dans notre simulation

où K_{proxy} est la raideur du proxy et F_{adh} est la force maximale due à l'adhérence. Dans cette condition, nous ne considérons pas l'amortissement du proxy dû au fait que l'ensemble de la procédure se fait avec une vitesse relativement faible et la variation de vitesse est elle aussi très faible. C'est donc naturellement que la vitesse relative entre la poignée et la pointe de l'aiguille ne se révèle jamais assez importante pour que l'amortissement soit une valeur à prendre en compte. Cette condition peut être vue comme le fait que l'utilisateur applique suffisamment d'effort pour passer à un état de glissement.

Pour résumer, l'algorithme de création de l'effort à appliquer est présenté en Figure 4.17. Dans ce schéma v_{lim} est une vitesse obtenue par essai-erreur en cherchant à produire un rendu réaliste. Dans la représentation du phénomène de *stick-slip* présenté en Figure 4.14, la force de frottement est une force tangentielle découlant du poids de la masse reposant sur le tapis roulant ($M \cdot g$). Dans notre cas ce poids n'a pas de sens car le frottement sec à l'origine du phénomène de *stick-slip* est dû à l'action des tissus sur le corps de l'aiguille. Cette interaction est donc représentée par F_{max} qui est l'équivalent de $M \cdot g$ dans le cas réel.

Cet algorithme propose une version très simplifiée du phénomène réel, en particulier lorsque l'on se situe dans un état d'adhérence. En effet, dans cet état la force de frottement n'est pas sentée être constante mais devrait croître jusqu'à atteindre sa valeur maximale. Il s'agissait ici de proposer une première approche de la représentation du phénomène *stick-slip* à l'origine de la sensation de craquement. Pour plus de précision et de réalisme, il pourra être nécessaire d'en améliorer le fonctionnement dans de prochaines versions du simulateur. Du point de vue de l'implémentation, ce rendu haptique est appliqué directement à l'interface pour préserver son aspect soudain.

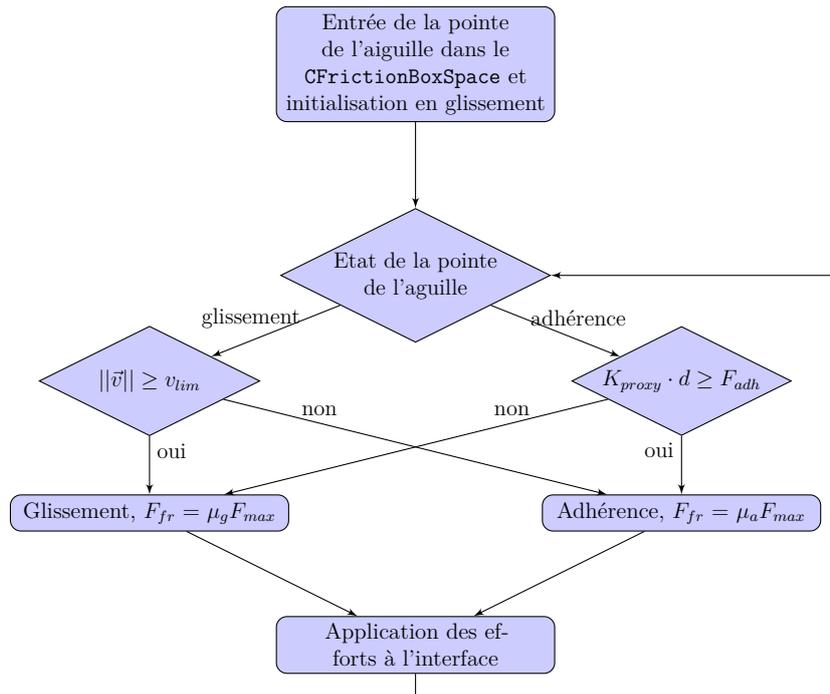


FIGURE 4.17 – Algorithme de choix entre une phase de glissement et une phase d’adhérence et efforts associés

4.3 Fonctionnalités supplémentaires : vers un simulateur idéal

Nous allons ici traiter des différents éléments mis en place pour rapprocher le simulateur proposé du simulateur idéal proposé par VAUGHAN *et al.* [3]. De plus nous traiterons le sujet de la synchronisation des deux parties du simulateur.

4.3.1 Paramétrages du modèle : représentation simplifiée de différents types de patients

Afin de rendre le simulateur suffisamment versatile, il est nécessaire de permettre à l'utilisateur la modification de certains paramètres pour simuler différents types de patients. Plutôt que de ne proposer que des simulations fixes spécifiques pour chaque type de patients, nous avons choisi de mettre à disposition un certain nombre de paramètres et de proposer des archétypes pour servir de base.

En collaboration avec les médecins partenaires, les différences entre les patients ont été identifiées comme suit :

- la profondeur du derme : le «derme» de la simulation regroupe le derme en lui même mais aussi d'éventuelles masses graisseuse qui peuvent se retrouver chez les patients en sur-poids. Il faut donc permettre la modification de la profondeur du derme pour pouvoir proposer la simulation de patient en sur-poids ;
- la résistance à la coupe des différentes couches : chaque couche physique a sa

résistance spécifique qui peut se révéler différente d'un patient à l'autre. Toutes sont donc modifiables, cependant les couches proposant le plus de variation de résistance sont le derme (avec la présence ou non de masses graisseuses) et le ligament jaune. La résistance de ces deux couches doit donc être facilement modifiable ;

- la présence de l'effet de *stick-slip* dans le ligament jaune : si cet effet est souvent présent il peut ne pas être ressenti. Il est particulièrement sensible sur des personnes présentant une calcification du ligament. Il est donc nécessaire de pouvoir supprimer ce rendu haptique le cas échéant ;
- la profondeur de l'espace péri-dural : selon l'anatomie du patient cette profondeur peut varier de 4 à 8 mm. Contrairement à la plupart des autres paramètres celle-ci n'est pas vraiment liée à une morphologie particulière. De plus, cette information n'est généralement pas disponible pour l'anesthésiste avant de poser la péri-durale. Il faut donc pouvoir la modifier et garder sa valeur cachée pour l'utilisateur.

La totalité de ces paramètres est conservée dans un fichier de configuration à modifier de préférence avant le lancement de la simulation. Le fichier contient un paramétrage par type de patient (moyen, sur-poids, calcifié) et un jeu de paramètres supplémentaire pour un patient nommé «patient *custom*». Ce type de patient est complètement laissé à l'appréciation de l'utilisateur. Il est alors libre de proposer des simulations de cas irréalistes. Toutefois cela peut permettre de générer à la volée un patient en sur-poids présentant une calcification du ligament jaune par exemple. Par la suite l'application permet la modification de certains paramètres en cours de fonctionnement à travers son interface utilisateur. Toute modification réalisée par l'interface ne sera pas sauvegardée dans le fichier de configuration. L'interface utilisateur permet l'accès direct aux paramètres suivants :

- le type de patient à représenter. l'utilisateur peut donc sélectionner un type de patient prédéfini (moyen, sur-poids, calcifié) ou un patient «*custom*». Le choix du type de patient choisi restreindra alors l'accès à certaines valeurs de paramètres,
- l'épaisseur du derme. Ce paramètre est limité selon le type de patient représenté,
- l'épaisseur de l'espace péri-dural,
- l'effort de coupe pour le ligament jaune. Ce paramètre est aussi limité selon le patient représenté,
- la présence éventuelle de frottement secs dans le ligament jaune. Ce paramètre est modifiable uniquement lorsque l'on sélectionne le type de patient «*custom*»

La modification des paramètres de la simulation via l'interface utilisateur est limitée de telle sorte à s'assurer que le patient reproduit reste réaliste. De manière générale, ces paramètres sont les paramètres qui présentent les différences les plus notables dans un cas réel. Ils ont donc un grand intérêt pédagogique d'où la simplicité de modification. Dans l'ensemble, seul le type de patient «*custom*» permet d'accéder à tous les paramètres tout en assurant le réalisme du patient simulé.

4.3.2 Ajout d'une visualisation IRM

Une des caractéristiques principales du simulateur idéal de VAUGHAN *et al.* [3] est la possibilité de mettre à disposition des apprenants une visualisation de la procédure grâce à des images IRM ou CT scan par exemple. Grâce au CHU de la Croix Rousse de Lyon, nous avons pu avoir accès à quelques images IRM anonymisées que nous avons intégrées à notre simulateur. Cependant, comme il s'agit de véritables images et non d'une simulation, les dimensions observées sur celles-ci sont fixes et ne peuvent donc pas représenter n'importe quel type de patient.

Toutefois, une telle visualisation n'étant pas disponible en pratique, elle doit surtout permettre aux novices de se faire une idée pratique de ce qu'il se passe pendant leurs premiers essais. Ainsi en calquant la géométrie de notre modèle sur les dimensions observées sur ces images il est possible de faire coller la représentation à l'image et ainsi de reporter la progression de l'aiguille directement sur l'image IRM. Les deux images utilisées sont présentées en Figure 4.18.

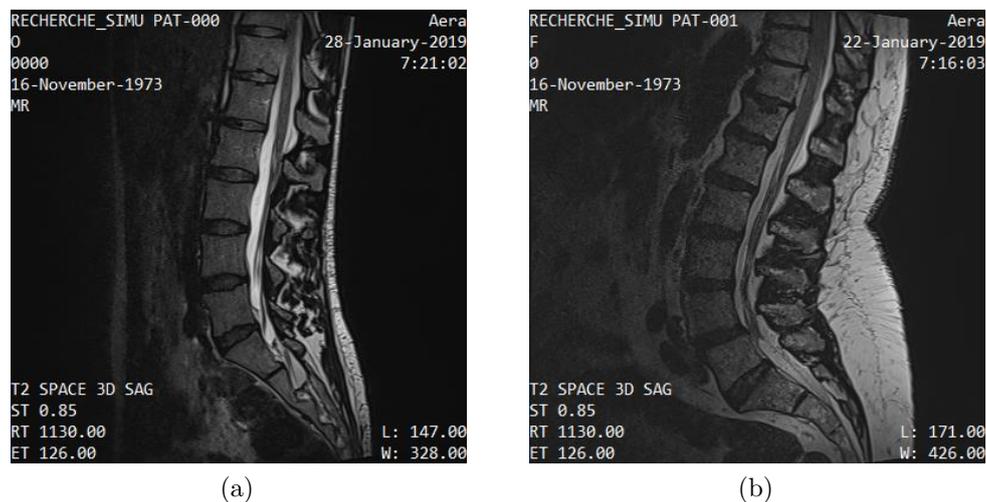


FIGURE 4.18 – Images IRM prises dans le plan sagittal utilisées dans l'application : (a) Patient «moyen», (b) Patient en sur-poids

Nous avons donc à disposition l'image Figure 4.18a qui sert pour le patient moyen mais aussi pour le patient calcifié, les deux ayant des mensurations relativement similaires, et l'image Figure 4.18b pour le patient en sur-poids. Pour ce qui est de la visualisation dans l'application, elle est réalisée à travers la classe `CIRMViewer`. Elle consiste en une simple projection de la position de l'aiguille sans tenir compte de la profondeur dans le plan coronal.

4.3.3 Acquisition des données de mesures

L'un des principaux avantages d'un simulateur informatisé est la possibilité d'enregistrer chaque essai. Ceci est bien entendu possible sur notre simulateur grâce à la classe `CFileWriter`. Cette classe fournit l'accès à une instance de la classe partout dans le code ce qui permet d'enregistrer les efforts produits ou la position d'un objet

par exemple. Il est possible d'ajouter toute donnée souhaitée à tout moment dans l'implémentation. Pour finir, cette classe permet l'enregistrement des données sous la forme de fichiers de mesures.

La création des fichiers de mesures est réalisée lorsque l'utilisateur a fini son essai, ce qui permet de ne pas bloquer la simulation en cours d'essai.

4.3.4 Design de l'interface utilisateur

L'interface utilisateur est très simpliste cependant certaines caractéristiques sont essentielles au fonctionnement de l'application. L'interface est constituée de plusieurs fenêtres présentant chacune son intérêt. Premièrement, une fenêtre centrale est utilisée pour gérer les paramètres de simulation. Cette fenêtre est idéalement cachée à l'utilisateur car elle contient les valeurs précises de chaque dimension importante, elle sera donc utilisée préférentiellement par le superviseur de l'essai. Une deuxième fenêtre permet de visualiser l'état du modèle 3D en cours de simulation. Elle aussi doit être cachée à l'utilisateur et son intérêt principal est de permettre de comprendre l'état de la simulation afin de débloquent une situation incomprise par l'utilisateur. Cette visualisation permet par exemple de vérifier s'il y a contact avec un os et permet au superviseur de donner des conseils à l'utilisateur si celui-ci se retrouve bloqué. Enfin une troisième fenêtre affiche la visualisation IRM. Cette fenêtre sert donc particulièrement au débutant pour lui donner ses premiers repères. L'ensemble de ces fenêtres sont visibles dans la Figure 4.19.

4.3.5 Synchronisation du Virtuose et du vérin pneumatique

Les deux parties du simulateur n'ont pas de moyen de synchronisation *a priori*, chacune étant contrôlée par une application différente. Afin de mettre en place cette synchronisation nous avons simplement utilisé une carte Arduino Uno. Les raisons de ce choix ont été la simplicité de son intégration. Il est possible de communiquer avec lui depuis l'application C++ à travers la connexion série USB de l'Arduino et celui-ci peut ensuite communiquer avec l'interface dSPACE à travers un signal *PWM* (*Pulse Width Modulation*). C'est une solution provisoire apportant son lot de désavantages liés pour la plupart à la vitesse de communication que l'on peut maintenir avec l'Arduino. Il fallait donc trouver un moyen de réduire la quantité de données à lui transmettre pour ne pas le surcharger.

L'application contrôlant l'interface haptique électrique est donc utilisée pour envoyer les informations de synchronisation nécessaires à l'application ControlDesk en charge du contrôle du vérin. Les informations de synchronisation restent assez simples. Il suffit de transmettre l'identifiant de la couche traversée par l'aiguille à un instant t . En effet, l'effort résistant au niveau de la seringue ne dépend que de cette variable. De plus, cette connexion est aussi utilisée pour synchroniser le type de patient choisi et donc les niveaux d'efforts à reproduire au niveau de la seringue. Cette information est envoyée une fois depuis l'application C++ vers ControlDesk au début de chaque essai. L'utilisation d'un Arduino Uno est une solution de prototypage rapide et n'a pas vocation à rester telle quelle.

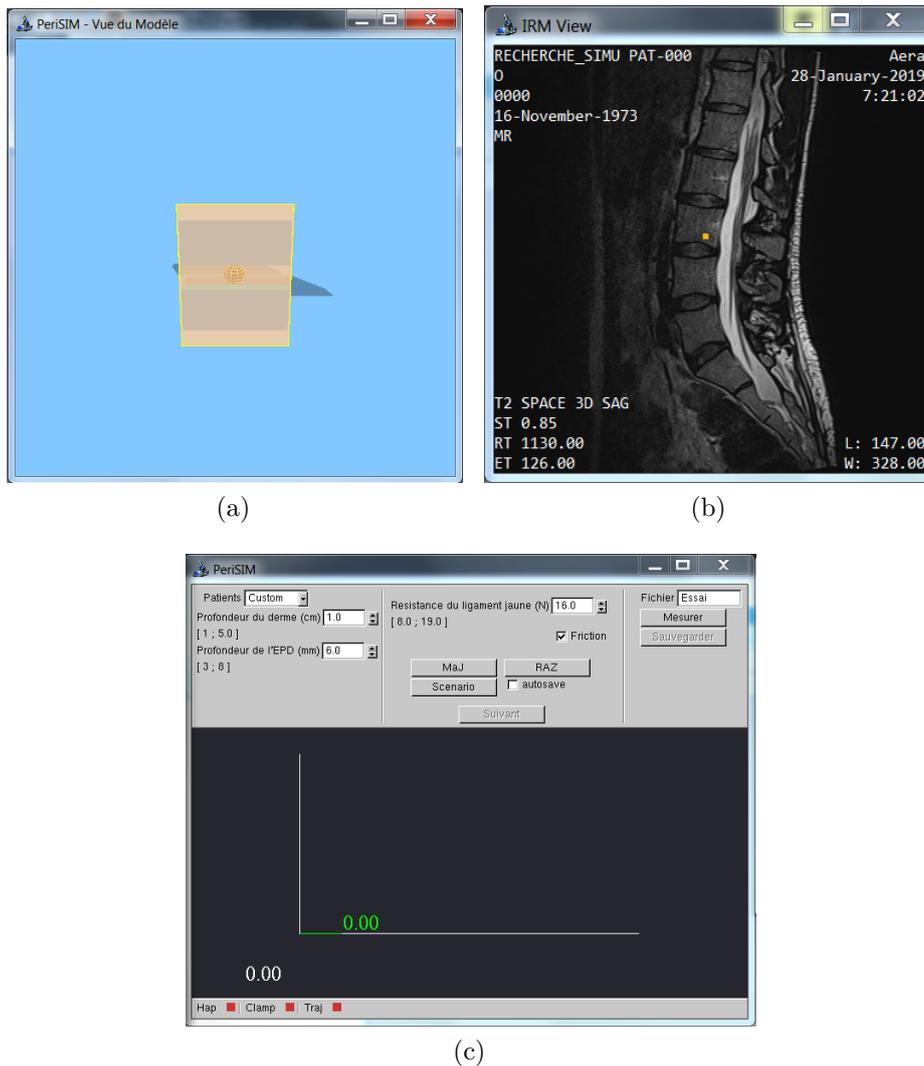


FIGURE 4.19 – Fenêtres de l’application contrôlant l’interface haptique électrique : (a) visualisation de l’environnement 3D, (b) visualisation IRM, (c), interface utilisateur centrale

Pour cela nous utilisons la communication par USB qui est un processus bloquant dans le programme. Afin d’éviter de bloquer la simulation haptique l’ensemble des communications ont été implémentées dans un processus séparé. Cette communication n’est activée que lorsqu’un changement de couche physique a été détecté.

Nous sommes ainsi en mesure de garantir que la carte Arduino sera disponible pour récupérer l’information. De plus ce mécanisme de transfert est utilisé pour transmettre les données concernant le type de patient à reproduire et en particulier pour préciser le type de *LOR* à reproduire. Ceci est fait ponctuellement avant chaque essai. Un schéma récapitulatif de l’ensemble des applications est présenté en Figure 4.20.

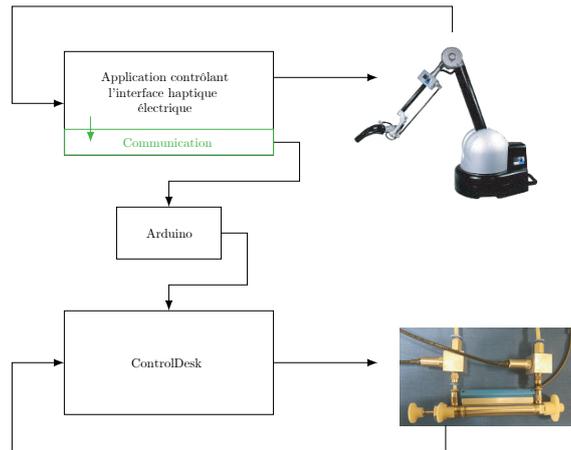


FIGURE 4.20 – Architecture complète de contrôle du simulateur

4.4 Exemple commenté

Pour finir, nous allons ici présenter le déroulement classique d'un essai du point de vue de la simulation haptique afin de passer en revue la plupart des éléments présentés dans ce chapitre et permettre une meilleure compréhension de l'ensemble. Avant toute chose, l'utilisateur ou le formateur choisit les paramètres essentiels de sa simulation en sélectionnant un type de patient parmi les quatre mis à sa disposition, ou en fixant lui-même les valeurs des paramètres clés à sa disposition sur l'interface utilisateur. Une fois cette première étape terminée, il peut ensuite démarrer son essai.

En début d'essai, la pointe de l'aiguille est située à la surface de la peau. Dans cette simulation, le point d'insertion est fixe et correspond à un point d'insertion «parfait». Ce point se situe légèrement en dessous du centre de l'espace intervertébral et permet donc d'adopter une trajectoire légèrement montante afin d'éviter de toucher un os. Il est en effet possible de voir sur la Figure 4.18 que les vertèbres présentent une légère inclinaison. En début de procédure, la pointe de l'aiguille entre donc en contact avec le `CSoftPlane` représentant la surface de la peau. Ce contact permet de créer l'effet haptique représentant F_{raid} puis après avoir parcouru 4 mm, la pointe de l'aiguille parvient dans le derme, ce qui a plusieurs conséquences :

- l'entrée dans le premier `CResistantBoxSpace` qui entraîne donc l'apparition de la résistance F_{coupe} caractéristique du derme ;
- l'entrée dans un `CSoftTrajColBoxSpace` ce qui se traduit par la génération d'une première trajectoire rectiligne pour créer la première partie de la force F_{traj} ;
- la mise à zéro de la valeur de F_{raid} , le point de perçage de la peau ayant été atteint.

Si l'aiguille continue d'avancer dans la bonne direction, elle rencontrera alors un second `CResistantBoxSpace` représentant cette fois-ci le premier ligament, à savoir le ligament supraspinal. À ce moment là, la pointe de l'aiguille sortira de la `CSoftTrajColBoxSpace` mettant donc fin à cette première trajectoire pour arriver

sur une nouvelle plus rigide dont la direction et le point de départ sont fixées au moment de l'entrée dans le ligament.

Ensuite, l'insertion se poursuit en entrant dans trois autres **CResistantBoxSpace** représentant dans l'ordre le ligament intraspinal, le ligament jaune et enfin l'espace péridural. Si l'insertion se poursuit après ce point, la pointe de l'aiguille peut alors atteindre le **CResistantBoxSpace** représentant la dure mère et l'utilisateur est alors averti que son essai a donné lieu à une ponction de la dure mère.

Tout au long de cette insertion, les données transmises à l'Arduino et donc au contrôleur du vérin pneumatique correspondent simplement aux moments où la pointe de l'aiguille pénètre dans une **CResistantBoxSpace**. Si l'aiguille n'est dans aucun de ces espaces, le contrôleur ne reçoit donc pas d'information et considère que l'aiguille est en dehors du patient. En outre, nous envoyons en plus une information si l'aiguille touche un os. En effet dans ce cas il est normalement impossible d'injecter du liquide, et il faut donc le faire ressentir à l'utilisateur. Enfin une fois l'essai fini, l'utilisateur interrompt son geste et a la possibilité d'enregistrer les données de son essai pour une analyse *a posteriori* via des algorithmes de classification présentés dans le Chapitre suivant.

Conclusion

Suivant la plupart des exigences énoncées par VAUGHAN *et al.* [3] reprises dans le cahier des charges section 2.1, nous avons pu développer une application permettant de contrôler un Virtuose 6D afin de simuler l'insertion d'une aiguille dans l'espace intervertébral afin d'atteindre l'espace péridural. La simulation proposée a pour avantage d'être simple et suffisamment réaliste. De ce fait il est très facile de représenter différents types de patients que ce soit en utilisant les archétypes fournis dans l'application ou en personnalisant entièrement le patient représenté. L'implémentation à bases de briques simples produisant chacune un effet haptique spécifique permet d'entreprendre simplement des modifications sur le modèle représenté et peut simplement être adapté à d'autres procédures similaires. La solution mise en œuvre permet un fonctionnement tout à fait satisfaisant de la simulation, d'après les retours d'usage des médecins partenaires.

La solution actuelle ne permet cependant pas à l'utilisateur de modifier quelles variables sont enregistrées car ceci est défini à la compilation de l'application. De plus, concernant la visualisation IRM, si le modèle utilisé en conjonction est directement basé sur les dimensions relevées sur les images IRM, il peut y avoir de légères imprécisions dues à la simplicité de la modélisation utilisée comparée à la complexité de la réalité. Il s'agit aussi simplement d'une image de fond qui n'est pas modifiée par le passage de l'aiguille ce qui devrait être le cas en réalité. Ainsi si nous souhaitons plus de réalisme, il est nécessaire d'apporter plus d'effort dans cet aspect de la simulation.

Concernant le modèle, celui-ci est pour le moment extrêmement rudimentaire et ne permet pas à l'utilisateur final de modifier certains paramètres de la simulation. En effet, le point d'insertion de l'aiguille est choisi au lancement de la simulation et ne dépend pas d'un choix de l'apprenant. Il s'agit simplement du point d'insertion «idéal» dans le cas présenté. Enfin, s'il est possible dans l'implémentation de pivoter les différents éléments du modèle 3D, cela n'est pas non plus facilement accessible à l'utilisateur. Ainsi la procédure simulée présente un cas «idéal» en ce sens. Pour plus de réalisme et pour prendre en compte plus de cas possibles, il serait intéressant de proposer au moins plusieurs postures possibles conformément à ce que l'on peut rencontrer en réalité. Le patient peut en effet ce tenir assis, couché sur le côté ou plus ou moins courbé.

Chapitre 5

Essais, résultats et étude du geste

Introduction

Dans ce chapitre nous allons détailler les essais réalisés, en dérivant le protocole expérimental, les critères d'analyse et les résultats obtenus. Le simulateur développé et présenté dans ce manuscrit a permis de mettre en place des sessions d'essais ayant pour but de le tester dans des situations diverses afin d'en étudier les performances. Un des objectifs de ces premiers essais a été de tester la possibilité d'utiliser le simulateur dans un contexte d'apprentissage.

Nous avons pu regrouper douze experts et douze novices afin de mettre le simulateur à l'épreuve et obtenir les résultats présentés dans ce chapitre. Nous avons donc pu tester différentes analyses des données acquises afin d'établir un moyen de «noter» la performance des utilisateurs et ainsi pouvoir étudier du geste de l'anesthésie péridurale. Cette partie présentera l'ensemble des méthodes testées et finira sur une proposition de méthode afin d'évaluer une éventuelle progression des apprenants.

L'approche ne consiste pas à développer un nouvel algorithme de classification mais plutôt de mettre en œuvre des algorithmes existants. Nous avons pour objectif de ne pas simplement nous limiter à une ou deux variables pour établir le niveau de performance mais plutôt d'avoir une approche plus globale.

5.1 Essais : protocole et données brutes

Dans cette section, nous allons présenter les essais réalisés et les mesures acquises. L'objectif est de présenter la logique du protocole mis en place durant la période d'essai. Les essais ont été réalisés en présence de 12 experts et 12 novices extérieurs au projet, dans l'optique d'obtenir des mesures et des données sur lesquelles travailler mais aussi pour obtenir des avis externes d'experts sur le simulateur.

5.1.1 Protocole expérimental

Le but des essais est d'obtenir suffisamment de points de mesure avec des participants de niveaux différents mais aussi de tester différentes configurations de simulation. Afin de garder une certaine répétabilité dans les essais nous avons donc décidé de réaliser le protocole suivant :

- chaque utilisateur dispose de deux essais «d'échauffement» non enregistrés ;
- chaque utilisateur doit effectuer douze essais différents ;
- parmi les douze essais, chaque utilisateur teste quatre patients standards, quatre patients en sur-poids et quatre patients calcifiés ;
- l'ordre des patients est choisi aléatoirement et le type de patient est annoncé avant l'essai à l'utilisateur ;
- l'ensemble de ces essais est réalisé en succession ;
- chaque utilisateur remplit un questionnaire sur le réalisme de la simulation pour les experts et sur l'intérêt pédagogique perçu pour les novices

Chaque essai dure quelques dizaines de secondes et chaque utilisateur a eu l'occasion de tester le simulateur (sans enregistrer de mesure) sur deux patients afin de se familiariser avec sa prise en main. Les novices devaient apprendre les rudiments de la procédure et les experts devaient adapter leur pratique à l'architecture du simulateur. Nous avons aussi fait en sorte que chaque utilisateur n'assiste à aucune autre utilisation du simulateur avant de le tester lui-même et ne pas fausser le temps de prise en main.

À chaque essai, la simulation redémarre dans une position similaire, à savoir avec la pointe de l'aiguille positionnée à la surface de la peau. Il a donc été demandé à l'utilisateur de repositionner le simulateur pour que l'origine du repère de l'avatar corresponde au point d'insertion physique du simulateur.

Les essais présentés ont eu lieu entre novembre et décembre 2018. De ce fait, et pour que tous les utilisateurs aient accès à la même version du simulateur, certains effets n'étaient pas encore présents. Le simulateur utilisé ne présentait donc pas les frottements secs de type *stick-slip* présentée dans la section 4.2.5, ni la première partie de la trajectoire implémentée à travers les `CSoftTrajColBoxSpace` présentée en section 4.2.1. Dans cette version, les frottements secs ont été remplacés par une valeur de résistance plus importante et la trajectoire rectiligne était imposée à l'utilisateur dès l'entrée dans le derme.

Une fois la série d'essais terminée chaque utilisateur a du répondre à un questionnaire selon qu'il soit novice (Annexe C) ou expert (Annexe D). Concernant les retours, les utilisateurs ont fait état d'un grand intérêt pour le simulateur, le considérant comme simple d'utilisation et intéressant pour la formation (Figure 5.1). De plus le réalisme a été globalement très apprécié, que ce soit au niveau de la représentation des différents patients que dans la représentation des différents éléments anatomiques. Enfin, les novices ayant pu le tester ont indiqué se sentir moins stressés à l'approche de leur première véritable péridurale après avoir pu s'entraîner une première fois sur le simulateur.

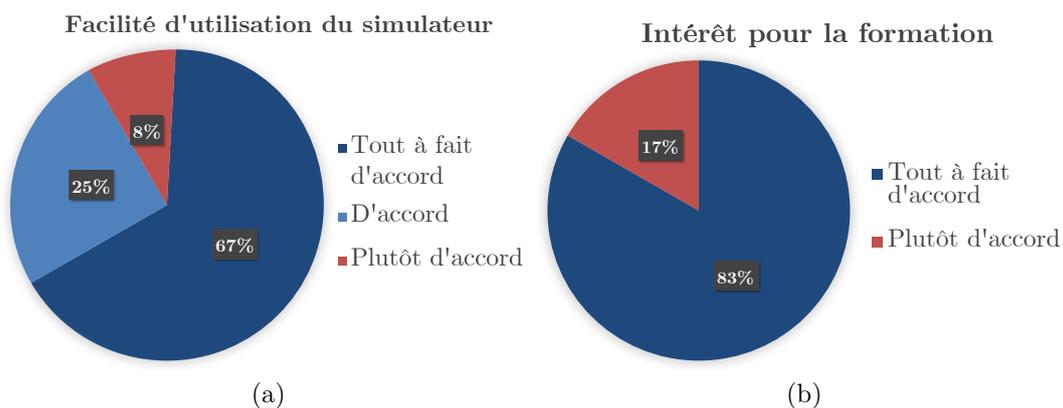


FIGURE 5.1 – Réponses obtenues des utilisateurs concernant la facilité d'utilisation du simulateur et son intérêt en tant qu'outil de formation d'après l'ensemble des utilisateurs

5.1.2 Données enregistrées et traitements préalables

Lors des essais, le simulateur enregistre les paramètres de la simulation, à savoir le type de patient simulé, l'effort maximal dans le ligament jaune, la profondeur du derme et la profondeur de l'espace péri-dural. Pour ce qui est des mesures, les données enregistrées sont les suivantes :

- la position de la pointe de l'aiguille dans les trois dimensions du repère de base de l'interface haptique en fonction du temps ;
- les efforts appliqués à la pointe de l'aiguille selon les trois dimensions du repère de l'interface haptique en fonction du temps ;
- l'angle d'insertion pris par la trajectoire de l'aiguille ;
- le nombre d'insertions nécessaires ;
- le nombre de contacts osseux ;
- la position finale de l'aiguille une fois que l'utilisateur décide avoir fini la procédure ;
- la position du piston du vérin pneumatique ;
- l'effort pneumatique produit par le vérin pneumatique.

Pour étudier le geste réalisé, il faut définir des variables qui peuvent par la suite servir de métriques. Dans un premier temps, nous avons défini un jeu de variables dans le but d'obtenir le plus d'éléments à analyser quitte à réduire ce nombre par la suite si certaines ne se révélaient pas pertinentes. Le choix de ces variables a pu être réalisé avec l'aide d'experts afin de proposer l'utilisation des variables cruciales d'un point de vue médical (*overshoot* par exemple), à quoi nous avons choisi d'ajouter des variables plus caractéristiques de l'étude de geste en général (vitesses et accélérations par exemple). Nous avons donc choisi les variables suivantes pour chaque essai :

- le temps utilisé pour réaliser la procédure ;
- la vitesse moyenne de la pointe de l'aiguille selon les directions du repère de base de l'interface haptique ;
- la variance de cette vitesse ;
- l'accélération moyenne de la pointe de l'aiguille selon les directions du repère de base de l'interface haptique ;
- la variance de cette accélération ;
- l'angle d'insertion de l'aiguille ;
- le nombre d'insertion ;
- le nombre de contacts osseux ;
- la distance parcourue dans l'espace péri-dural (l'*overshoot*) ;
- la réussite ou non de l'essai ;
- la vitesse moyenne du piston du cylindre pneumatique ;
- la variance de cette vitesse sur une procédure ;
- l'accélération moyenne du piston du cylindre pneumatique ;
- la variance de cette accélération sur une procédure ;
- «la quantité de seringue vidée» : en quelque sorte la distance totale parcourue par le piston du vérin pneumatique ;
- un indicateur d'utilisation de la seringue. Cet indicateur correspond simplement à une normalisation de la quantité de seringue vidée en fonction de la distance à parcourir côté aiguille. Étant donné que pour une personne en sur-poids, l'espace péri-dural se situe bien plus en profondeur que dans un cas classique, l'utilisateur pourrait donc avoir tendance à vider une plus petite quantité de la seringue sur un patient classique que sur un patient en sur-poids.

Le nombre de ces variables se porte donc à un total de 26, toutes numériques, mise à part l'information de réussite. Ne disposant que des données de position au cours du temps, il est nécessaire de dériver ces valeurs et pour cela nous avons utilisé la méthode des différences finies de CHAWDA *et al.* [49] sur une fenêtre de cinq échantillons de temps afin d'obtenir un résultat peu bruité et de qualité. Cela nous permet non seulement de limiter l'influence du bruit mais aussi de pouvoir accéder à la variance de la vitesse et de l'accélération.

Il est aussi nécessaire de préciser que pour assurer la précision de la valeur d'*overshoot*, il a été nécessaire d'implémenter une méthode spécifique. En effet,

si on se fie uniquement à la position de la pointe de l'aiguille, l'*overshoot* sera relativement imprécis. Comme la simulation a recours à un algorithme de *proxy*, la pointe de l'aiguille dans la simulation se retrouve «tractée» par l'interface. Ainsi lorsque l'utilisateur cesse d'avancer, la pointe de l'aiguille aura tendance à continuer pendant quelques échantillons supplémentaires pour se rapprocher de la position réelle de l'interface. Nous obtenons alors un positionnement imprécis. Pour éviter cela, lorsque la pointe de l'aiguille arrive dans l'espace péri-dural, nous nous intéressons à la position réelle de l'interface. Nous enregistrons donc la position physique de l'interface de l'entrée dans l'espace péri-dural jusqu'à la fin de la mesure. De ce fait lorsque l'utilisateur arrête son geste, la position de l'interface se retrouve elle aussi à l'arrêt.

5.1.3 Vers une analyse du geste

L'objectif de tout simulateur d'apprentissage est bien évidemment de permettre à des élèves de progresser dans leurs pratiques. Pour faire état de cette progression, il est possible de chercher à produire des métriques pour établir le niveau de performance. Comme présenté dans le Chapitre 1, il existe différentes métriques selon les gestes étudiés. Dans le cas de l'anesthésie péri-durale, la donnée principale étant l'*overshoot* [33]. Toutefois, utiliser simplement une variable pourrait se révéler insuffisant pour une évaluation pertinente.

Il nous a donc fallu essayer de mettre en avant une méthode d'étude plus poussée. N'ayant pas accès à une signification précise de la performance en dehors de la réussite ou non du geste, notre approche a donc été de considérer que les experts ayant participé aux tests ont une maîtrise suffisante du geste. Ainsi la maîtrise du geste a été définie à partir de l'analyse des gestes d'experts. Tout essai similaire à un essai d'expert est alors considéré comme un succès. Partant de là nous avons pu tester différentes approches pour déterminer la «similarité» d'un essai donné avec un essai d'expert.

Dans la suite du manuscrit, les analyses proposées auront donc pour objectif de déterminer un niveau de maîtrise à partir des mesures effectuées sur un essai particulier. Afin de tester les méthodes proposées nous aurons donc recours à la mise en place d'un groupe d'apprentissage qui permettra d'entraîner les algorithmes proposés à déterminer un niveau de maîtrise. Par la suite, nous testerons cet algorithme entraîné sur les données restantes. Nous définirons donc un échantillon d'entraînement correspondant à deux tiers des mesures effectuées et nous garderons le tiers restant pour servir d'échantillon de test.

5.2 Analyses «naïves»

Dans cette section, nous allons présenter des analyses «naïves» pour ensuite passer sur des analyses plus détaillées et plus précises. Dans une approche «naïve», il s'agira ici de faire une étude variable par variable des différents essais afin de tenter d'établir un niveau d'expertise.

5.2.1 Notations

Dans les sections à venir, les matrices seront notées en majuscules et en gras, les vecteurs en minuscule et en gras et les éléments seront eux en minuscule tout au long du manuscrit. Les vecteurs et éléments d'une même matrice conserveront de plus le même nom.

Nous noterons la matrice de mesure $\mathbf{X} \in \mathbb{R}^{n \times m}$ pour représenter une matrice contenant n observations et m variables. Nous noterons \mathbf{v}_i avec $i \in \llbracket 1 ; m \rrbracket$ le vecteur de base de l'espace des mesures \mathbb{R}^m correspondant à la i^{eme} variable. Enfin la notation \mathbf{X}^T correspondra à la matrice \mathbf{X} transposée.

5.2.2 Variables intéressantes

Nous commençons par présenter les résultats variable par variable parmi les 26 à notre disposition, en mettant l'accent sur les variables les plus importantes. Nous avons donc essayé de déterminer les variables significativement différentes en moyennes entre novices et experts. Nous présenterons ensuite un premier algorithme d'évaluation simplement basé sur les valeurs de certaines de ces variables. La Figure 5.2 présente des variables mesurées en séparant les essais réalisés par les experts de ceux réalisés par les novices. Les variables présentées figurent parmi les plus intéressantes car elle permettent d'entrevoir la différence entre novice et expert.

Parmi ces variables, le temps de réalisation de la procédure (ou *TCT*) est la variable la plus «étonnante», car si habituellement, le temps diminue avec l'expertise [30] dans notre cas les observations nous donnent le résultat inverse. La différence reste toutefois significative avec une valeur p inférieure à 0,01 pour un test de WILCOXON sur la moyenne des deux distributions.

Concernant la quantité de seringue vidée, nous pouvons noter que dans la plupart des essais, les experts utilisent quasiment exactement une seringue complète. À l'inverse, les novices présentent aussi une moyenne aux environs d'une seringue complète, mais la valeur varie beaucoup d'un essai à l'autre. Nous avons pu vérifier au cours des essais, que quoi qu'il arrive, lorsque la perte de résistance est détectée par un utilisateur expert, celui-ci aura tendance à vider complètement le contenu restant dans la seringue. Ce n'est pas forcément le cas des utilisateurs novices. Il s'agit d'une différence notable qui est cependant assez difficile à utiliser telle quelle pour évaluer la performance sur un geste en particulier. En effet, en définitive la moyenne des novices se situe également aux alentours d'une seringue complète.

Pour la vitesse moyenne de l'aiguille selon l'axe x du repère de l'interface présenté en section 4.1.1, les résultats se recourent avec les observations que nous pouvons faire vis-à-vis du temps nécessaire pour réaliser la procédure. La direction x correspondant quasiment à la direction principale d'insertion, nous notons que les experts vont en moyenne moins vite que les novices pour réaliser la procédure (la vitesse selon x est négative). Cela vient sans doute du fait qu'en s'autorisant à aller moins vite un expert s'assure de réussir et il est donc possible qu'un novice ne parvienne pas à évaluer le risque d'échec correctement. Encore une fois la différence de moyenne entre le groupe d'expert et le groupe de novice est significative avec une

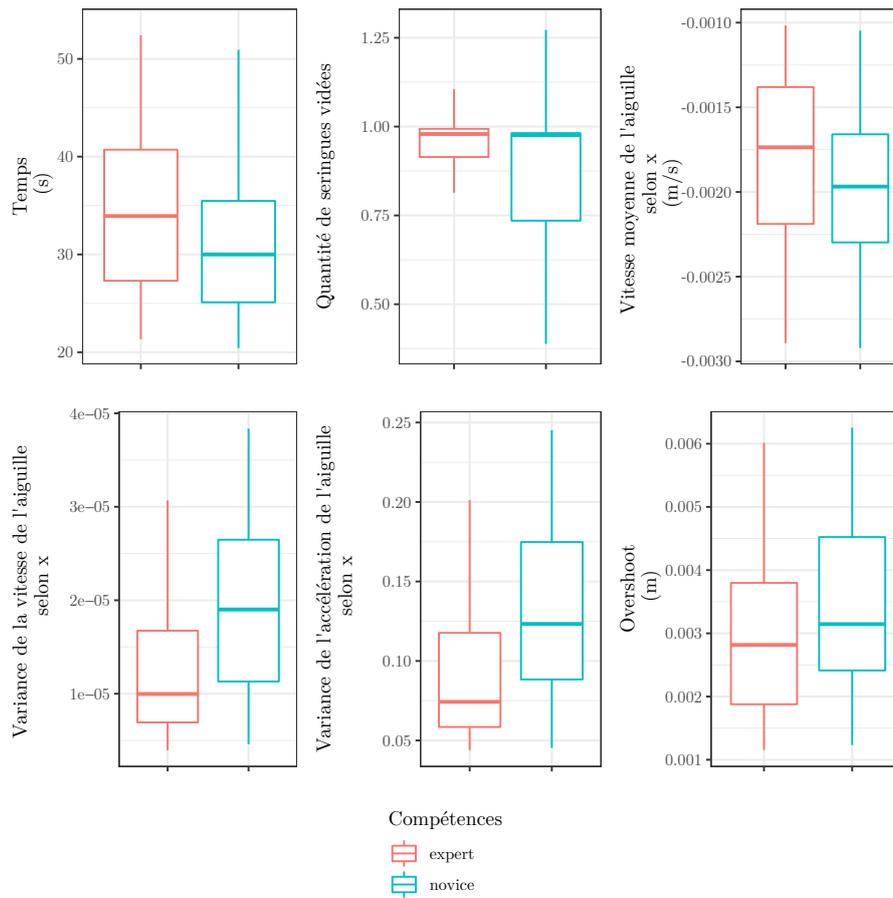


FIGURE 5.2 – Variables présentant une différence notable entre novices et experts

valeur p inférieure à 0,01.

Pour ce qui est des deux variances (vitesse et accélération) mises en avant dans la Figure 5.2 nous pouvons noter que dans les deux cas, les variances des experts sont bien plus faibles en moyenne que celles des novices. Nous retrouvons donc ici des résultats relativement classiques [50, 51, 52] avec l'expertise se traduisant par une certaine régularité du mouvement. Il est d'ailleurs possible d'analyser la vitesse moindre des experts comme moyen de s'assurer une vitesse et une accélération relativement constante. Là encore, dans les deux cas, la différence de moyenne est significative avec une valeur p très inférieure à 0,01.

Enfin, il convient de mettre en avant l'*overshoot* qui reste l'étalon de performance le plus courant [33] concernant le geste de la pose de péridurale. Sur cette variable, les experts ont une valeur d'*overshoot* inférieure à celle des novices ce qui correspond à ce que nous pouvons retrouver dans l'état de l'art et montre donc bien un niveau de maîtrise différent. Concernant cette variable la différence de moyenne est aussi significative avec une valeur p très inférieure à 0,01.

5.2.3 Une première méthode de classement

Vu qu'il existe un certain nombre de variables qui semblent indiquer en elle-même des différences de maîtrise par leur simple valeur. Il est possible d'établir un moyen d'évaluation en se basant simplement sur des mesures étudiées une à une afin d'essayer de mettre en avant une maîtrise particulière. Nous allons donc ici étudier une méthode de classification se basant simplement sur les différentes valeurs des variables. L'algorithme de classification proposé ici est un algorithme dit d'arbre binaire. Le principe de ce type d'algorithme est de définir des hyperplans successifs afin de séparer l'espace des mesures en plusieurs sous espaces représentant chacun une classe particulière. Il est à noter que ce type de méthode fonctionne de manière successive en définissant des seuils discriminant sur différentes variables. L'algorithme définit les seuils de manière à obtenir la meilleure séparation entre les deux classes définies. Les variables choisies par l'algorithme sont donc celles qui permettent successivement d'obtenir les meilleures classifications sur l'échantillon d'entraînement et dépendent donc de cet échantillon. Par exemple pour un espace de mesure $M \in \mathbb{R}^n$:

- l'algorithme cherche un seuil sur la variable v_i en deçà duquel une observation est dans la classe \mathcal{C}_1 et au delà duquel une observation est de la classe \mathcal{C}_2 ;
- l'algorithme cherche ensuite un seuil dans chacun de ces sous ensembles pour définir de nouveaux sous-ensembles et ainsi de suite.

Suite à cela, nous obtenons donc un certain nombre de sous ensembles qui peuvent par la suite être regroupés pour former un nombre de classes donné. S'il est possible de simplement définir un nombre arbitraire de sous ensembles à partir d'une telle méthode, nous allons plutôt l'utiliser afin de définir des variables de décisions afin d'évaluer la performance réalisée. Pour effectuer ce classement nous avons utilisé des algorithmes développés sous R, à savoir ici à travers la librairie `rpart` [53]. En appliquant cette méthode de classification, on parvient à obtenir des arbres hiérarchiques dont certains sont présentés en Figure 5.3. Chaque étiquette de cette figure présente la classe la plus probable (expert ou novice), la probabilité associée (expert en premier puis novice¹) puis le pourcentage de l'échantillon représenté. Enfin plus la couleur de l'étiquette est saturée (ou sombre), plus la confiance est grande concernant la prédiction effectuée. Par exemple sur la Figure 5.3a l'étiquette 2 indique qu'elle représente des experts avec une probabilité de 0,87 et contient 23% de l'ensemble des échantillons d'entraînement. Ensuite, chaque embranchement présente la variable testée et le seuil choisi. L'arbre se lit donc de haut en bas. Il faut aussi noter la répartition entre novices et experts dans certains groupes créés. Nous pouvons par exemple citer le groupe 28 de la Figure 5.3f qui est constitué de quasiment autant de novices que d'experts. Ainsi, si ce groupe est reconnu expert car ceux-ci y sont en majorité, il s'agirait plutôt d'un groupe indéterminé.

Nous notons que dans ces arbres une des façons de déterminer l'expertise est d'analyser la variance de l'accélération de l'interface haptique électrique selon x

1. En toute logique, si le groupe est un groupe expert la probabilité associée aux experts sera supérieure à 0.5 et inversement

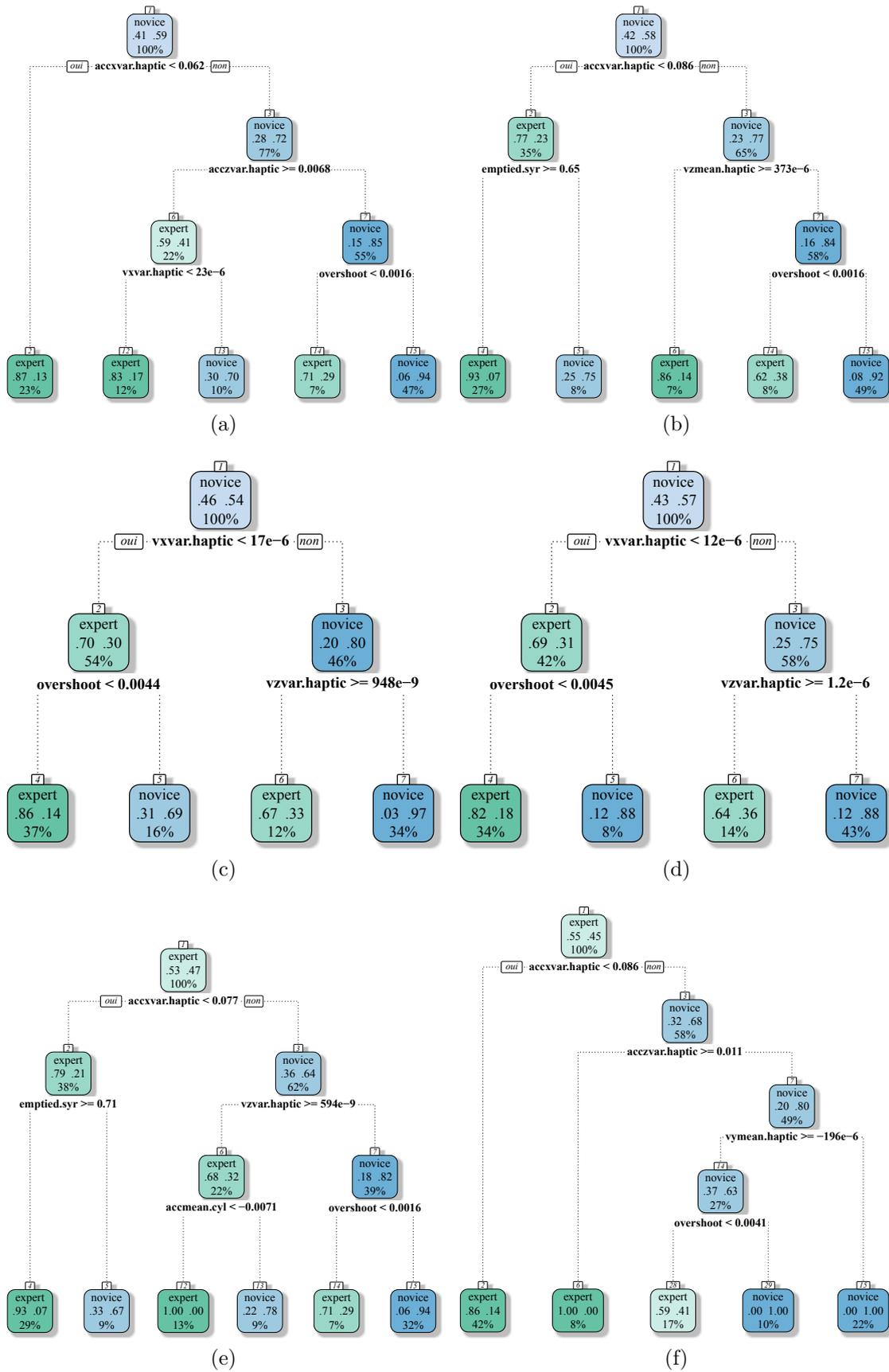


FIGURE 5.3 – Échantillon d'arbres hiérarchiques obtenus d'après les mesures

(`accxvar.haptic`) ainsi que la variance de la vitesse de l'interface haptique électrique selon x (`vxvar.haptic`). Nous retrouvons donc ici une certaine notion de régularité de la trajectoire. Une autre façon de déterminer l'expertise est de regarder la variance de vitesse de l'interface haptique électrique selon z (`vzvar.haptic`) ou la variance de l'accélération de l'interface haptique électrique selon z (`acczvar.haptic`). Cette condition est plus difficile à analyser. Ceci est peut être dû au fait que la bonne pratique est d'incliner légèrement l'aiguille vers le haut au moment de l'insertion. Avec cette inclinaison, nous obtenons alors une part de la trajectoire de l'interface haptique électrique selon z . Ce type de trajectoire étant surtout connu des experts il paraît logique que nous puissions envisager une séparation autour de cette variable en particulier. Enfin un autre élément essentiel concernant l'expertise est bien entendu la valeur de l'*overshoot*.

En utilisant cette méthode de classification nous obtenons alors les résultats présentés en Figure 5.4. Pour les obtenir, nous comparons la classe prédite par l'algorithme par rapport à la classe véritable de la mesure. Les résultats sont ensuite rassemblés en présentant les taux de bons classements par classe d'origine. Ces résultats sont loin d'être satisfaisants et se rapprocheraient presque d'une classification aléatoire pour certains d'entre eux. Toutefois des points importants sont notables. Ainsi, l'avantage de ce type de méthodes est la possibilité de comprendre rapidement le mode de classement et ainsi de pouvoir l'expliquer et utiliser cette connaissance pour comprendre la performance d'un utilisateur. Les variables mises en avant par ce type d'algorithme vont donc permettre de construire une pédagogie du geste sur les bonnes pratiques à adopter.

En conclusion, cette méthode «naïve» ne se révèle pas être suffisamment efficace en ce qui concerne les classifications obtenues. Bon nombre d'essais se retrouvent, en effet, mal classés en utilisant simplement cette méthode. Toutefois l'utilisation de ce type d'algorithme a pour avantage de mettre en avant un certain nombre de variables importantes dans la compréhension et l'analyse du geste de la périodurale. Il est, par exemple, possible de noter l'importance de l'*overshoot* ou des variances de vitesses d'insertion d'aiguille dans l'analyse du geste. Cette première analyse nous permet donc d'acquérir une connaissance *a priori* du geste qui pourra se révéler utile pour la mise en œuvre des méthodes de classification suivantes.

5.3 Réductions de dimensions et classification

Nous allons maintenant aborder des méthodes plus avancées afin d'établir avec plus de certitudes l'expertise d'un utilisateur. Pour mettre en place ces méthodes nous allons commencer par chercher à réduire la dimension de l'espace de mesure afin de ne garder que les variables pertinentes pour ensuite mettre en œuvre des moyens de classifications connus sur notre jeu de données.

5.3.1 Algorithmes de réductions de dimension utilisés

Nous commençons en présentant rapidement les deux méthodes de réduction de dimension utilisés dans nos travaux afin de comprendre et de pouvoir expliquer

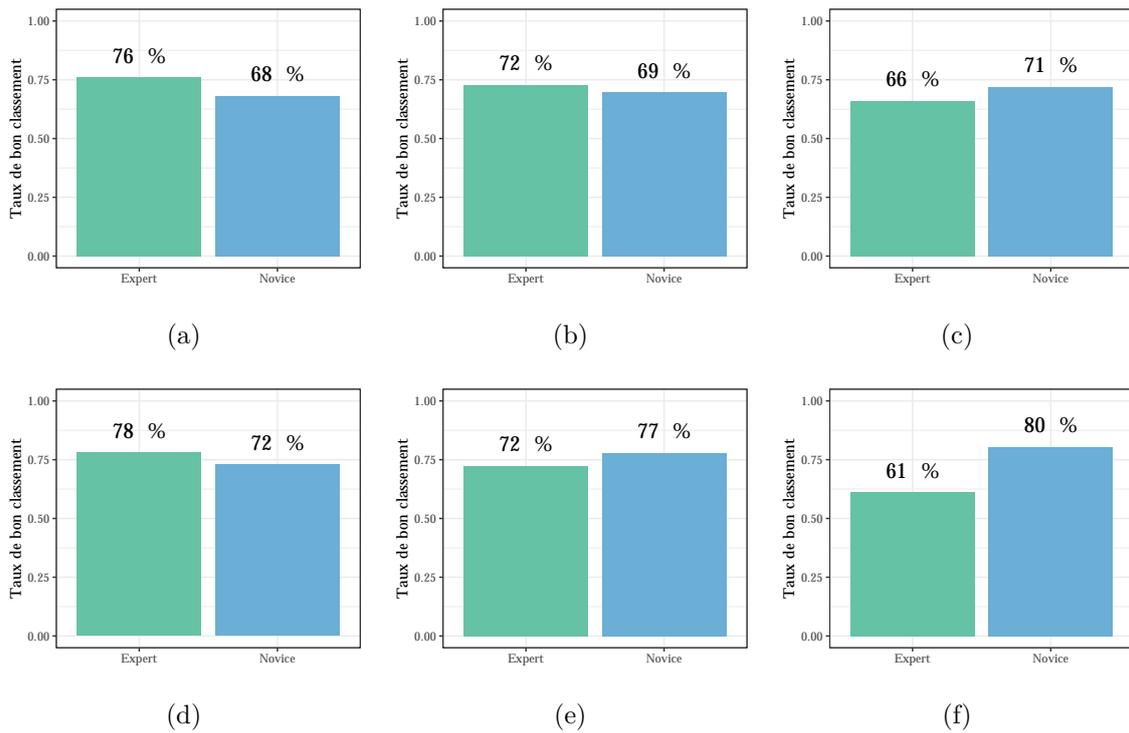


FIGURE 5.4 – Taux de bons classements correspondant aux arbres de classification présentés en Figure 5.3

plus simplement les résultats qui en découleront.

Le principe de l'Analyse en Composantes Principale (ACP) : objectifs et obtention

L'ACP [54, 55] est une méthode de réduction de dimension qui vise à établir les composantes principales d'un nuage de point en étudiant la variance selon les différentes dimensions de départ. Pour cela, cet algorithme cherche à réduire le nombre de dimension en conservant les plus possible «l'inertie» du nuage de point. Pour cela, nous commençons par effectuer une réduction et un centrage des données. Cela a pour but de permettre de comparer des valeurs exprimées dans des dimensions différentes et possiblement avec un ordre de grandeur différent. Nous utiliserons donc de préférence la matrice $\tilde{\mathbf{X}}$ définie telle que :

$$\tilde{\mathbf{x}}_i = \frac{\mathbf{x}_i - \bar{\mathbf{X}}}{\sigma(\mathbf{X})} \quad \forall i \in \llbracket 1 ; n \rrbracket \quad (5.1)$$

où \mathbf{x}_i et $\tilde{\mathbf{x}}_i$ correspondent aux lignes de la matrice de mesures et de la matrice centrée réduite respectivement, $\bar{\mathbf{X}}$ le vecteur des moyennes de chaque variable et $\sigma(\mathbf{X})$ le vecteur des écarts-types de chaque variable. On a donc une matrice dont les variables ont une moyenne nulle et un écart-type de 1. On peut alors définir la

décomposition en valeur singulières [56] de la matrice de mesure telle que :

$$\tilde{\mathbf{X}} = \mathbf{P}\mathbf{\Delta}\mathbf{Q}^T \quad (5.2)$$

où \mathbf{P} est la matrice $n \times l$ de vecteurs singuliers à gauche \mathbf{Q} la matrice $m \times l$ de vecteurs singuliers à droite, avec l le rang de la matrice $\tilde{\mathbf{X}}$ et $\mathbf{\Delta}$ la matrice diagonale des valeurs singulières. Par définition les matrices \mathbf{P} et \mathbf{Q} sont orthonormales. Dans l'ACP, nous notons que :

$$\mathbf{X}\mathbf{Q} = \mathbf{P}\mathbf{\Delta}\mathbf{Q}^T\mathbf{Q} = \mathbf{P}\mathbf{\Delta} = \mathbf{F} \quad (5.3)$$

où \mathbf{F} est appelée la matrice des scores de facteurs. Nous voyons ici que \mathbf{Q} fait office de matrice de projection et est aussi appelée matrice de charges. Cette projection décrit les observations contenues dans \mathbf{X} selon ce que l'on appelle les composantes principales. Une fois cette décomposition faite, nous pouvons utiliser la matrice \mathbf{Q} obtenue pour écrire de nouvelles observations dans la base définie par les composantes principales. Soit \mathbf{x}_{sup} un vecteur colonne de données supplémentaires. Nous pouvons définir ses facteurs dans la base de composantes principales comme suit :

$$\mathbf{f}_{sup}^T = \mathbf{x}_{sup}^T\mathbf{Q} \quad (5.4)$$

C'est ensuite sur cette décomposition que nous allons pouvoir effectuer une réduction de dimension et quelques analyses supplémentaires

Interprétations de l'ACP

L'ACP a l'avantage de donner l'accès à un certain nombre de métriques intéressantes pour analyser la qualité de la représentation des données ou encore pour comprendre le sens de ce qui est présenté. Nous allons décrire ici trois éléments intéressants que nous utiliserons par la suite. Le premier est l'importance des différentes composantes. Tout compte fait, l'ACP ne fait que donner une base différente à l'espace des données par rapport aux variables originales. Dans un premier temps, il est possible d'organiser les composantes principales par ordre d'*importance*. Pour cela, il est intéressant de se pencher sur la matrice de corrélation \mathbf{C} :

$$\mathbf{C} = \mathbf{X}^T\mathbf{X} \quad (5.5)$$

Nous pouvons noter que $\mathbf{X}^T\mathbf{X} = \mathbf{P}\mathbf{\Delta}\mathbf{Q}^T\mathbf{Q}\mathbf{\Delta}^T\mathbf{P}^T = \mathbf{P}\mathbf{\Delta}^2\mathbf{P}^T$ car \mathbf{Q} est orthogonale et $\mathbf{\Delta}$ est diagonale. Ainsi $\mathbf{\Delta}^2$ est la matrice des valeurs propres de $\mathbf{X}^T\mathbf{X}$ que nous noterons $\mathbf{\Lambda}$. Les vecteurs propres correspondent donc aux composantes

principales de notre décomposition. Nous pouvons ensuite définir l'inertie γ_j des colonnes de la matrice \mathbf{X} (les variables) telle que :

$$\gamma_j^2 = \sum_{i=1}^n x_{i,j}^2 \quad (5.6)$$

Et nous pouvons noter que $\sum_j \gamma_j = \text{tr}(\mathbf{\Lambda})$. En effet :

$$\sum_j \gamma_j = \text{tr}(\mathbf{X}^T \mathbf{X}) = \text{tr}(\mathbf{\Lambda}) \quad (5.7)$$

où $\text{tr}(\cdot)$ représente la trace de la matrice. Ainsi en notant λ_i la i^{eme} valeur propre et en étudiant la valeur de :

$$P_i^{\text{inert}} = \lambda_i / \sum_j \lambda_j \quad (5.8)$$

Nous obtenons la proportion d'inertie expliquée par le i^{eme} vecteur propre. Ainsi, nous pourrions ne garder que les composantes avec la valeur de P_i^{inert} la plus grande pour effectuer une réduction de dimension et nous nous assurerons de garder le plus d'information possible.

Ensuite, il est possible de remonter à la corrélation entre variable et composante. Pour cela, il suffit de s'intéresser à la matrice \mathbf{Q} . En effet, comme nous avons pu le voir pour représenter un élément dans la base de nos composantes principales il suffit de multiplier ce vecteur par \mathbf{Q} à droite. Ainsi pour obtenir la représentation de la j^{eme} variable avec les nouvelles composantes principales, il suffit d'observer le vecteur :

$$\mathbf{cor}_j = \mathbf{q}_j \quad (5.9)$$

où \mathbf{q}_j correspond à la j^{eme} ligne de \mathbf{Q} . Nous obtenons donc un vecteur \mathbf{cor}_j de norme 1 par définition de \mathbf{Q} et nous pouvons donc les représenter sous la forme d'une sphère de corrélation. Cela nous permettra d'analyser le résultat de l'ACP en fonction des variables d'origine et ainsi nous pourrions mettre en avant les variables expliquant la plus grande partie de l'inertie définie précédemment.

Enfin, il est possible de définir la contribution des observations aux axes de l'ACP. Pour cela, il faut regarder les coordonnées de l'observation données exprimées dans la base de projection. Plus cette contribution est élevée, plus l'observation en question est responsable de l'existence de la composante principale. Pour cela, nous utilisons les éléments de la matrice des facteurs \mathbf{F} ainsi que les valeurs propres. Nous définissons alors la contribution d'une observation i à la composante l comme :

$$ctr_{i,l} = \frac{f_{i,l}^2}{\lambda_l} \quad (5.10)$$

Ainsi si l'une des contributions vient à être trop importante cela signifie la plupart du temps que l'observation en question peut faire figure de point aberrant et il peut donc être intéressant de la retirer le temps de définir les composantes principales de notre ACP.

L'algorithme *Stochastic Neighbor Embedding (t-SNE)*

Le second algorithme de réduction de dimension utilisé est appelé *t-SNE* [57] pour *Stochastic Neighbor Embedding*. Cet algorithme a pour but de réduire les dimensions d'un espace euclidien en se basant sur l'étude de probabilités conditionnelles. En prenant deux points \mathbf{x}_i et \mathbf{x}_j de \mathbb{R}^n , l'algorithme calcule les probabilités conditionnelles $p_{j|i}$ que \mathbf{x}_j ait pour voisin \mathbf{x}_i . Ainsi deux points voisins auront une forte valeur de probabilité $p_{j|i}$ et inversement deux points éloignés auront une faible valeur de probabilité $p_{j|i}$. Pour calculer ces probabilités l'algorithme fait l'hypothèse d'une certaine distribution statistique des variables. Dans la version que nous utilisons les variables seront supposées appartenant à une loi de STUDENT. Ainsi la probabilité $p_{j|i}$ peut s'écrire :

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_k - \mathbf{x}_i\|^2/2\sigma_i^2)} \quad (5.11)$$

où $\|\cdot\|$ est la distance euclidienne et σ_i le vecteur des variances centrées sur \mathbf{x}_i qu'il faut déterminer. Avec cette analyse on obtient donc une représentation de l'espace d'origine en termes de similarité et nous nous éloignons de la simple distance pour prendre en compte les répartitions statistiques. On définit ensuite le même jeu de probabilités pour les équivalents de \mathbf{x}_i et \mathbf{x}_j une fois la réduction de dimension effectuée notés \mathbf{y}_i et \mathbf{y}_j :

$$q_{j|i} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_k - \mathbf{y}_i\|^2)} \quad (5.12)$$

Afin de réorganiser les points de données dans un espace de dimension réduite et définir une valeur pour σ_i , il faut définir la *perplexité* :

$$Perp(P_i) = 2^{H(P_i)} \quad (5.13)$$

où P_i est la distribution de probabilités conditionnelles d'un \mathbf{x}_i par rapport à l'ensemble des autres points (nous définissons de même Q_i pour \mathbf{y}_i) et H l'entropie de SHANNON définie comme telle :

$$H(P_i) = \sum_j p_{j|i} \log_2 p_{j|i} \quad (5.14)$$

La *perplexité* définit en quelque sorte une mesure du nombre de voisins [57] et sera fixée par l'utilisateur. Cette valeur de *perplexité* permettra à l'algorithme de

définir les probabilités $p_{j|i}$. Enfin, nous définissons une fonction de coût correspondant à la somme des divergences de KULLBACK-LEIBLER que nous minimisons par un algorithme de descente de gradient.

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \quad (5.15)$$

Pour l'implémentation de l'algorithme, nous avons eu recours à la librairie `Rtsne` [58]. En utilisant cet algorithme, nous obtenons un espace de moindre dimension dans lequel les points sont arrangés en fonction des densités de probabilités de telle sorte que chaque point soit à proximité d'autres points similaires. Appliqué sur notre jeu de données et en ne gardant que trois dimensions, nous obtenons le résultat présenté dans la Figure 5.5. Se limiter à trois dimensions nous permet de conserver la possibilité de visualiser relativement facilement les données tout en conservant plus d'informations qu'en deux dimensions. Dans cette figure nous avons fait apparaître les novices en bleu et les experts en rouge et nous pouvons constater que la séparation en deux classes semble envisageable visuellement. Toutefois le problème principal de cet algorithme est qu'il ne propose pas beaucoup d'éléments de compréhension de son résultat contrairement à une ACP et il n'est pas possible de donner un sens aux axes obtenus. Nous utiliserons donc son résultat simplement pour tenter de classifier les données récoltées.

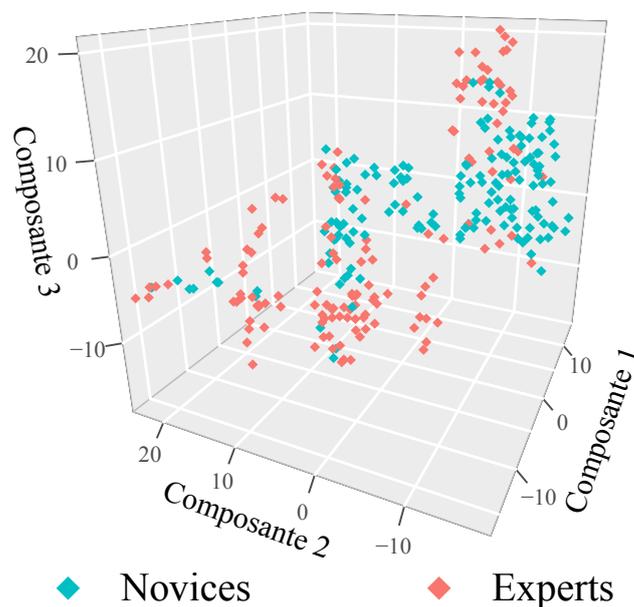


FIGURE 5.5 – Visualisation des données après utilisation d'un algorithme *t-SNE*

5.3.2 Sélection de variables et pré-traitement des données

Contrairement à la première méthode de classification proposée en section 5.2.3, les méthodes présentées ci-dessous reposent sur l'ensemble des variables simultanément et non séquentiellement. Afin d'obtenir un meilleur taux de bon classement,

nous avons donc cherché à améliorer les données disponibles en essayant de ne garder que les variables les plus pertinentes et en s’attellant à éliminer les essais aberrants.

Pour cela, nous appliquons une ACP à nos données pour essayer de repérer d’éventuels points aberrants et limiter le nombre de variables utilisées à celles ayant une importance significative dans l’explication de la variabilité de la matrice \mathbf{X} . L’algorithme d’ACP utilisé est implémenté dans la librairie `FactoMineR` [59]. Pour commencer, la Figure 5.6 présente le pourcentage de l’inertie totale expliquée par les composantes successives de l’ACP. Nous pouvons voir qu’en ne conservant que les cinq premières composantes, le jeu de données conserve plus de 60% de son inertie initiale. De plus, la méthode du «coude» (ou critère de CATTELL [60]) couramment employée, qui consiste à s’arrêter lorsque l’on note la présence d’un coude dans la décroissance de l’inertie expliquée, nous fait considérer en priorité les cinq premières composantes.

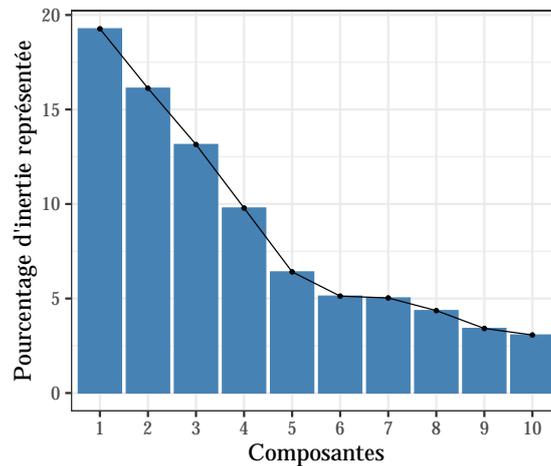


FIGURE 5.6 – Pourcentage d’inertie contenu dans les différentes composantes résultant de l’ACP sur le jeu de données complet

La Figure 5.7 présente alors l’importance cumulée des différentes variables dans la construction de ces cinq composantes principales. Sur cette figure, la ligne en trait interrompu court rouge correspond à la contribution moyenne attendue compte tenu du nombre de variables, à savoir $100/24 = 4.17\%$. Ainsi toute variable ayant une contribution nettement inférieure à cette moyenne n’a qu’une importance mineure sur les composantes en question. Il est donc possible de s’arrêter à cette moyenne limite. Toutefois, connaissant le geste en question et les résultats individuels nous allons ajouter l’*overshoot* qui reste un bon indicateur de performance et un facteur de réussite déterminant. De plus nous avons choisi de ne pas garder l’indicateur de seringue que nous avons défini dans la section 5.1.2 pour ne pas que celui-ci fasse doublon avec la quantité de seringue vidée.

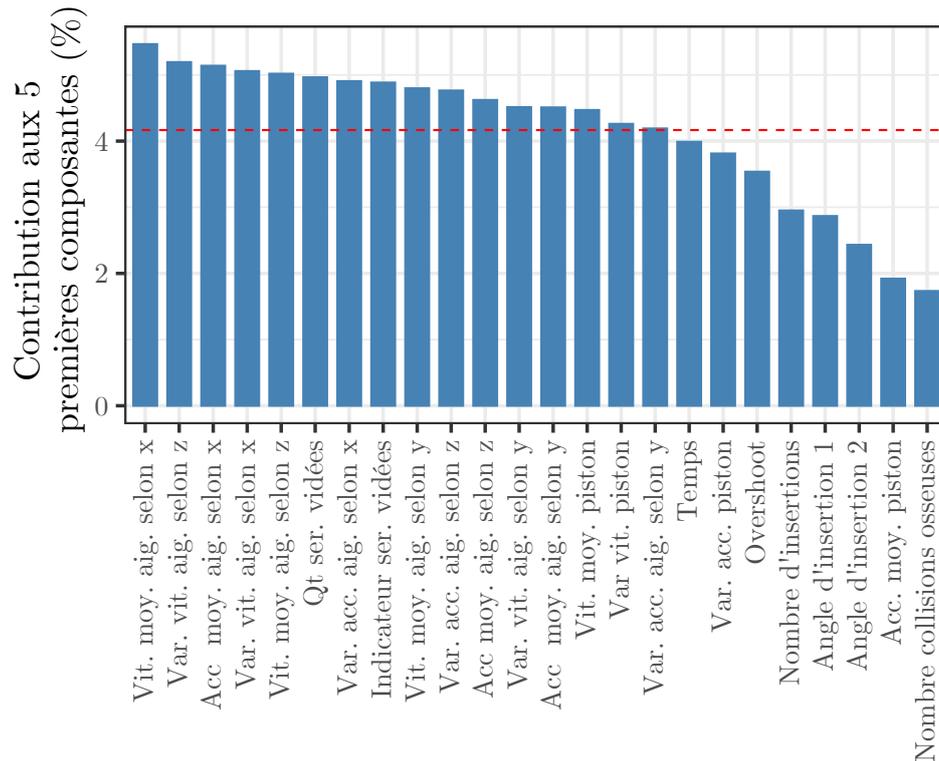


FIGURE 5.7 – Contributions des variables de bases aux 5 axes majeurs de l'ACP sur les données initiales

Une fois le nombre de variables utilisées réduit, nous pouvons étudier les mesures elles-mêmes pour voir si certaines d'entre elles se révèlent aberrantes. La Table 5.1 présente les essais ayant les contributions les plus importantes dans les cinq premières composantes de l'ACP. L'objectif n'est pas de mettre de côté la totalité des essais ayant une contribution trop importante mais d'en limiter la présence afin de faire en sorte que le reste des données ne se retrouve pas comprimé en un petit espace. Il sera bien entendu toujours possible de traiter ces observations à part. Dans notre cas, les données les plus problématiques sont :

- l'Expert 5, essai 1 selon la composante 3 ;
- l'Expert 5, essai 7 selon la composante 3.

Il aurait aussi été pertinent d'exclure le novice 7 essai 5 et 6 pour leur contribution selon l'axe 5. Cependant nous avons choisi de les conserver car leur contribution à cette composante est inférieure à la contribution à la composante 3 des deux essais de l'expert 5 et que la cinquième composante commence à ne représenter qu'une très faible partie de l'inertie totale du jeu de données.

Finalement, nous obtenons la représentation de données présentée en Figure 5.8 en deux dimensions avec le cercle de corrélation associé. Il paraît assez difficile de différencier *a priori* une zone correspondant à des experts d'une zone correspondant à des novices sur cette figure. Il ne faut toutefois pas oublier qu'il ne s'agit ici que des deux premières composantes de l'ACP et lors de notre analyse préliminaire

TABLE 5.1 – Tableau des individus à fortes contributions aux cinq premières composantes de l'ACP et contributions associées

	Comp. 1	Comp. 2	Comp 3.	Comp 4.	Comp 5.
Novice 7, essai 5	0,31	11,83	0,17	7,39	21,15
Novice 7, essai 6	0,73	5,55	0,32	7,00	18,18
Novice 9, essai 3	0,85	5,90	0,64	0,05	0,01
Expert 5, essai 1	0,34	0,17	25,30	0,17	0,05
Expert 5, essai 4	0,98	8,67	0,08	0,43	0,99
Expert 5, essai 6	0,65	8,67	0,01	3,43	2,12
Expert 5, essai 7	0,35	0,29	27,37	1,55	0,05
Expert 5, essai 12	1,03	9,33	0,14	0,96	2,23

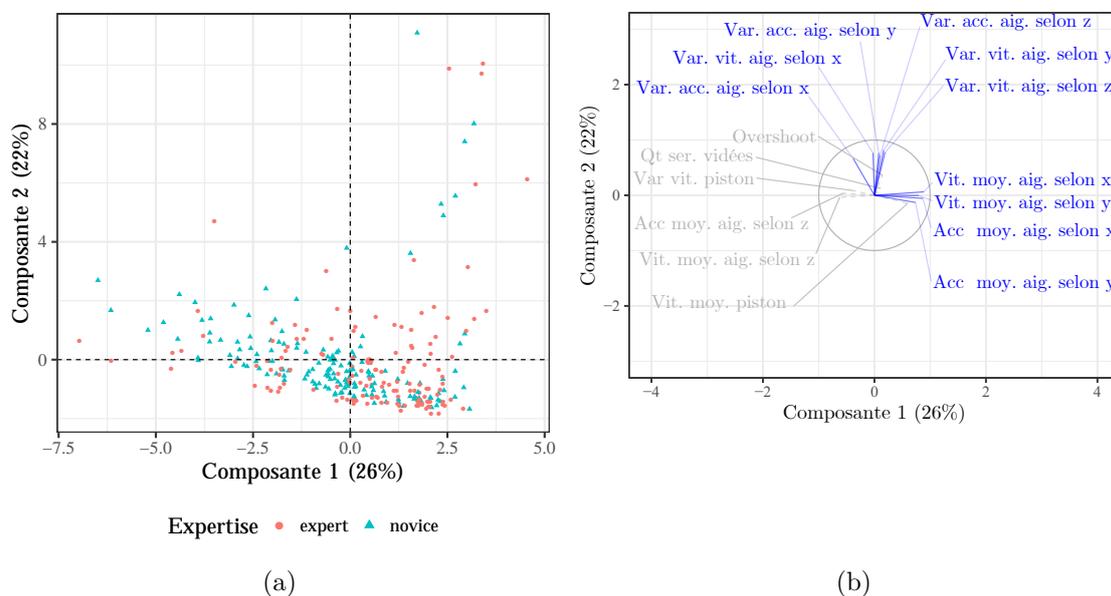


FIGURE 5.8 – (a) Ensembles des essais présentés sur les deux premières composantes de la l'ACP, (b) Cercle de corrélations pour les deux premières dimensions

nous avons choisi de garder au moins cinq composantes pour conserver suffisamment d'informations sur les données d'origine. Réaliser l'ACP dans un premier temps a surtout été un moyen de s'assurer de ne garder que les éléments pouvant aider à notre classification. Nous notons donc, qu'une fois les sélections faites, notre jeu de données ne semble plus contenir de points trop éloignés et nous pouvons déjà faire quelques remarques sur l'influences des différentes variables. Nous voyons, par exemple, que la première composante de l'ACP est très liée aux vitesses et accélérations moyennes de l'aiguille alors que la seconde composante présentée semble elle plus liées aux variances de ces vitesses et accélération. En conclusion, les éléments apportant le plus de disparités dans le jeu de données semblent être dans ces variables, ce qui confirme les observations faites en section 5.2.2.

5.4 Algorithmes de classification proposés

5.4.1 Algorithme des k-plus proches voisins (kPPV)

Le premier algorithme de classification proposé ici est un algorithme de k-plus proches voisins ou kPPV et implémenté dans la librairie `class` [61]. Afin de l'utiliser sur un jeu de données, il suffit de définir une distance dans l'espace de mesures considéré. Dans notre cas, la distance utilisée est simplement la distance euclidienne. Pour deux points \mathbf{x} et \mathbf{y} de \mathbb{R}^n nous définissons la distance de \mathbf{x} à \mathbf{y} comme suit :

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2} \quad (5.16)$$

L'algorithme de kPPV est ensuite assez simple puisqu'il suffit de fournir un ensemble d'entraînement et de préciser combien de voisins considérer. Ainsi, lorsque nous cherchons à classer un nouvel élément, l'algorithme détermine quels sont ses plus proches voisins parmi le *set* d'entraînement au sens de la distance euclidienne et peut ainsi faire une prédiction en fonction de la nature des voisins récupérés. Cet algorithme peut donc fournir une précision sur la certitude du classement en donnant le nombre de voisins de chaque type.

Dans notre application nous utilisons cet algorithme sur trois versions des données. Premièrement les données brutes (une fois les variables sélectionnées comme précisé en section 5.3.2), ensuite sur les données projetées sur les cinq premières composantes de l'ACP et enfin sur les trois dimensions issues de l'application de l'algorithme de réduction *t-SNE*. Dans notre cas la valeur de k a pu être choisie en essayant plusieurs valeurs différentes pour obtenir le meilleur résultat concernant le taux de bon classement. Pour cela, nous testons un l'algorithme avec k variant de 1 à 20 puis nous gardons le k donnant le meilleur résultat. Le résultat de ce test est présenté en Figure 5.9. Cette figure présente le taux de bon classement obtenu en fonction du nombre de voisins considéré. Ces tests sont effectués avec un *set* d'entraînement identique quelle que soit la valeurs de k . Les taux de succès obtenus en utilisant le meilleur k sont présentés en Figure 5.10. Nous notons que selon le type de données sélectionné, il est nécessaire de considérer plus ou moins de voisins pour améliorer la classification. Le point le plus intéressant est que nous avons besoin de sélectionner beaucoup plus de voisins pour obtenir de bons résultats en utilisant les données de l'ACP ($k = 16$) qu'avec les données issues de la réduction *t-SNE* ($k = 3$). Cela semble cependant cohérent car l'algorithme *t-SNE* semble rassembler les experts et les novices dans des zones de l'espaces assez distinctes (Figure 5.5) contrairement à l'algorithme d'ACP (Figure 5.8).

Nous remarquons que cette méthode de classification est bien plus efficace que les arbres hiérarchiques présentés plus haut. Toutefois, si elle est réalisée directement sur les données sans effectuer une réduction de dimension préalable, sa précision est amoindrie. Il faut donc avoir recours soit à une ACP soit à un algorithme de type *t-SNE* au préalable.

Un extrait des classifications obtenues présente, dans la Table 5.2, les confiances

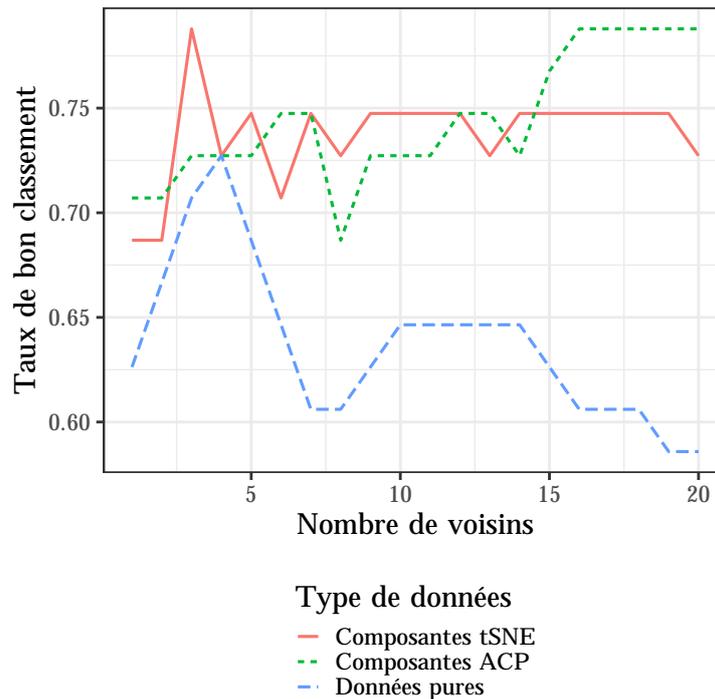


FIGURE 5.9 – Taux de bons classement selon le type de données utilisé et la valeur de k

obtenues sous la forme de probabilités. Nous pouvons voir dans ces quelques exemples que la confiance est parfois relativement proche voire égale à 0,5 et l'individu a alors quasiment autant de chance d'être un novice qu'un expert. De manière générale, l'utilisation des données issues d'un algorithme de *t-SNE* produit de meilleurs résultats. Cependant, du fait de l'utilisation de seulement 3 voisins, l'indice de confiance est moins précis que lorsque nous utilisons les données de l'ACP qui considère dans ce cas 16 voisins.

5.4.2 Algorithme Machine à Vecteurs de Support (SVM)

Un autre algorithme de classification relativement commun est l'algorithme de Machine à Vecteurs de Support [62] ou SVM pour *Support Vector Machine*. Cet algorithme définit un ensemble d'hyperplans pour découper l'espace de mesures et ainsi définir des zones correspondant à une classe en particulier. Pour définir l'hyperplan de séparation, il s'agit de résoudre un problème d'optimisation dépendant de la version de l'algorithme que nous souhaitons utiliser et du type de noyau que nous souhaitons utiliser. Nous ne rentrerons pas dans les détails de l'implémentation de cet algorithme mais il est à noter que ce type d'algorithme peut aussi fournir une estimation de la certitude de classification. Pour cela, une fois les hyperplans de séparation définis, il suffit de calculer la distance entre un point et l'hyperplan en question. Plus la distance est grande et plus la certitude augmente. La librairie utilisée pour cet algorithme est `e1017` [63].

Les taux de bons classement présentés en Figure 5.11 montrent d'assez bon

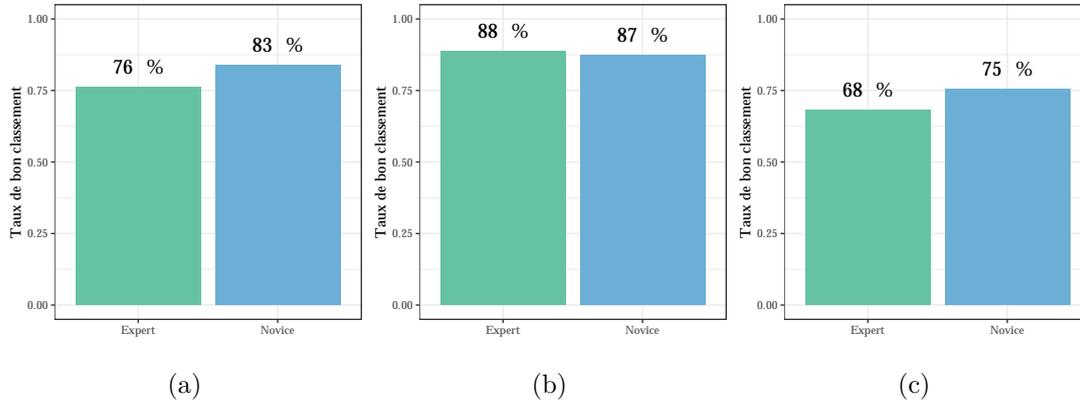


FIGURE 5.10 – Classifications obtenues par algorithme des kPPV sur : (a) composantes d'ACP avec $k=16$, (b) composantes de t -SNE avec $k=3$, (c) données brutes avec $k=4$

TABLE 5.2 – Extrait des résultats de classification obtenus par algorithme des kPPV avec les valeurs de k proposées en Figure 5.10, données utilisées, prédictions et confiances associées

	ACP		t -SNE		Données	
	Préd.	Confi.	Préd.	Confi.	Préd.	Confi.
Nov. 8, e. 5	Expert	0,75	Novice	1,00	Indéf.	0,50
Exp. 9, e. 8	Novice	0,81	Novice	1,00	Novice	0,75
Nov. 2, e. 5	Expert	0,69	Novice	0,67	Novice	0,75
Nov. 3, e. 10	Novice	0,62	Novice	1,00	Indéf.	0,50
Exp. 9, e. 1	Expert	0,88	Expert	0,67	Expert	1,00
Nov. 2, e. 10	Novice	0,56	Novice	1,00	Novice	1,00
Nov. 9, e. 9	Novice	0,94	Novice	1,00	Novice	1,00
Exp. 4, e. 8	Expert	0,88	Expert	0,67	Expert	1,00
Exp. 8, e. 5	Expert	0,81	Expert	1,00	Expert	0,75
Nov. 11, e. 9	Novice	0,94	Novice	1,00	Indéf.	0,50

résultats et ce quel que soit le type de données utilisées. Les résultats restent toutefois un peu moins bons que ceux obtenus à l'aide d'un algorithme de kPPV. Pour ce qui est de la confiance dans la classification un extrait est présenté dans la Table 5.3. Les taux de confiances sont en général assez bons lorsque nous utilisons les données de l'ACP ou de la *t-SNE*.

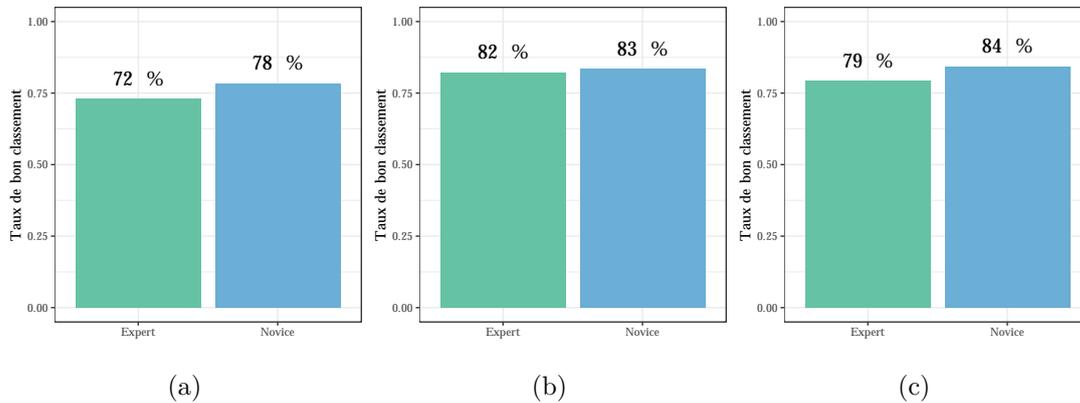


FIGURE 5.11 – Classifications obtenues par algorithme SVM sur : (a) composantes d'ACP, (b) composantes de *t-SNE*, (c) données brutes

TABLE 5.3 – Extrait des résultats de classification obtenus par algorithme SVM linéaire, données utilisées, prédictions et confiances associées

	ACP		<i>t-SNE</i>		Données	
	Préd.	Confi.	Préd.	Confi.	Préd.	Confi.
Exp. 9, e. 6	Expert	0,75	Novice	0,91	Expert	0,68
Nov. 8, e. 10	Novice	0,89	Novice	0,73	Novice	0,90
Exp. 5, e. 12	Expert	0,81	Expert	0,86	Expert	1,00
Nov. 8, e. 5	Expert	0,81	Novice	0,90	Expert	0,50
Nov. 1, e. 2	Novice	0,59	Novice	0,90	Novice	0,52
Nov. 3, e. 7	Novice	0,98	Novice	0,90	Novice	0,97
Exp. 5, e. 4	Expert	0,70	Expert	0,86	Expert	0,99
Nov. 2, e. 12	Novice	0,84	Novice	0,81	Novice	0,80
Nov. 2, e. 8	Novice	0,68	Novice	0,92	Novice	0,66
Nov. 6, e. 7	Novice	0,57	Novice	0,77	Novice	0,57

5.4.3 Réseau de neurones

La dernière méthode de classification testée est une méthode par réseau de neurones [64]. Ce type de méthode connaît un assez fort développement et il a donc été intéressant d'en tester l'efficacité sur notre jeu de données. Ce type d'algorithme présente un certain nombre d'avantages mais a aussi certains inconvénients. Il est en effet très difficile d'interpréter le résultat obtenu grâce à un réseau de neurones. Toutefois, nous avons pu tester un réseau implémenté sous R dans la librairie `neuralnet` [65] avec la même logique que les deux algorithmes précédents. L'utilisation

de R a été préférée à celle de Matlab par soucis de simplicité car nos données ont été étudiées dans ce langage. Il s'agit ici d'un réseau de neurones à rétro-propagation du gradient. Afin d'établir le réseau de neurone il faut en préciser l'architecture, à savoir le nombre de couches cachées et le nombre de nœuds par couches. Une des architectures choisies est présentée en Figure 5.12.

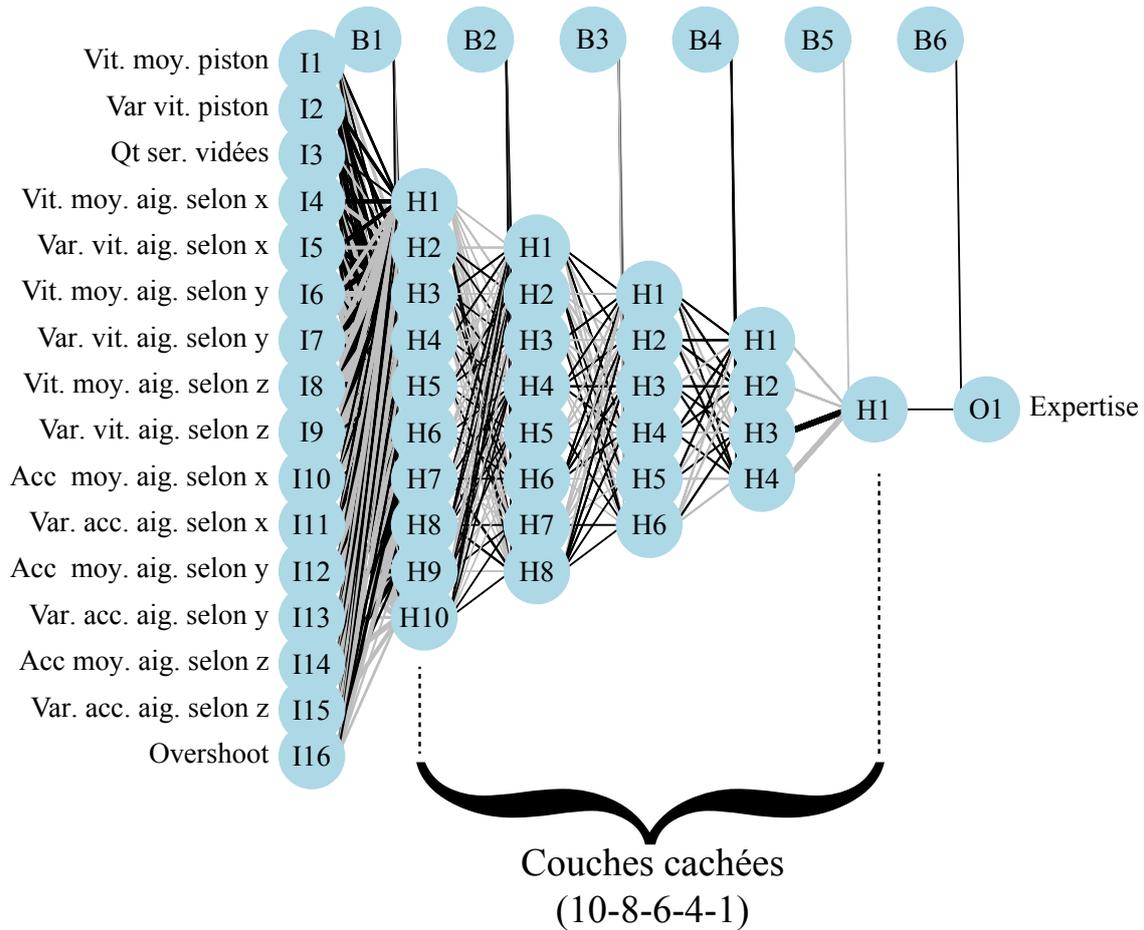


FIGURE 5.12 – Aperçu du réseau de neurones utilisés sur les données brutes

Pour chaque type de données plusieurs tests ont pu être effectués en cherchant à obtenir un résultat satisfaisant. Nous obtenons alors les résultats de classifications présentés en Figure 5.13

Les résultats de classification restent bons mais pas suffisamment pour être préférés aux méthodes précédentes. De plus il est assez complexe de trouver une architecture efficace simplement. Enfin ce type de classification ne donne pas accès directement à une évaluation de la confiance en la classification faite, contrairement aux algorithmes présentés précédemment. Cet algorithme présente cependant l'avantage de détecter avec une assez bonne réussite un novice même en utilisant les données brutes. Ces résultats restent peu poussés et une telle approche mériterait plus d'approfondissement afin d'obtenir de meilleurs résultats.

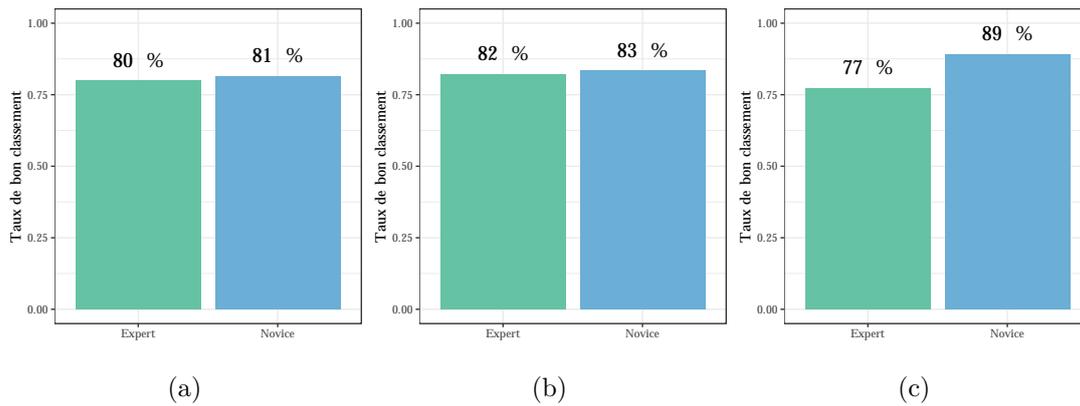


FIGURE 5.13 – Classifications obtenues par réseau de neurones sur : (a) composantes d’ACP (couches cachées 6-4-1), (b) composantes de t -SNE (couches cachées 10-8-6-4-1), (c) données brutes (couches cachées 10-8-6-4-1)

5.5 Méthode de classification et d’évaluation choisies

Parmi les méthodes testées l’algorithme de kPPV présente l’avantage de rester relativement simple et efficace pour évaluer la performance et la maîtrise du geste. Nous avons donc choisi de le garder pour proposer une évaluation des utilisateurs. Toutefois il est encore possible d’en améliorer la précision. Pour cela nous allons rassembler les classifications d’essais par individu afin de donner une appréciation globale de la performance plutôt qu’une étude mesure par mesure. En procédant ainsi nous pouvons améliorer grandement la précision de l’évaluation et ainsi obtenir les Tables 5.4 et 5.5.

Nous pouvons noter quelques classements hésitants même en utilisant cette méthode. De plus nous pouvons voir que l’utilisation des composantes de l’ACP donne souvent des réponses avec une moindre confiance contrairement à l’utilisation des données de l’algorithme t -SNE. Cela se comprend assez intuitivement du fait que le t -SNE rassemble les essais par similarité et a donc tendance à rassembler les essais d’une même personne ensemble. Ainsi si un expert se trouve trop similaire à un novice sur un essai, il y a de forte chance que ce soit le cas pour ses autres essais et inversement.

Enfin pour compléter l’aspect pédagogique de l’évaluation, il est intéressant de se pencher sur les variables mises en avant par la classification par arbre hiérarchique qui donne de moins bons résultats mais apporte des clés de compréhension plus pragmatiques de l’ensemble. En associant les variables mises en avant par ces arbres et le système d’évaluation présenté ici, nous pourrions proposer un début de formation pour de futurs médecins. Pour cela, les travaux de thèses de CANDALH-TOUTA [66] concernant la formation en laparoscopie pourraient servir de référence à transposer sur le geste de la péridurale.

TABLE 5.4 – Évaluation obtenue en agrégeant les classifications de chaque individu en utilisant un algorithme de kPPV sur les données de l’ACP avec $k = 16$

	Classements		Évaluation	
	Novice	Expert	Prédiction	Confiance
Novice 1	3	0	Novice	1
Novice 2	4	4	<i>Indéf.</i>	-
Novice 3	2	0	Novice	0,60
Novice 4	2	1	Novice	0,67
Novice 5	4	0	Novice	1
Novice 6	3	1	Novice	0,75
Novice 7	1	1	<i>Indéf.</i>	-
Novice 8	3	1	Novice	0,75
Novice 9	5	2	Novice	0,71
Novice 10	2	0	Novice	1
Novice 11	5	0	Novice	1
Novice 12	4	0	Novice	1
Expert 1	0	2	Expert	1
Expert 2	2	2	<i>Indéf.</i>	-
Expert 3	1	2	Expert	0,67
Expert 4	0	4	Expert	1
Expert 5	0	4	Expert	1
Expert 6	0	2	Expert	1
Expert 7	0	2	Expert	1
Expert 8	0	5	Expert	1
Expert 9	2	4	Expert	0,67
Expert 10	0	6	Expert	1
Expert 11	0	1	Expert	1
Expert 12	1	2	Expert	0,67

5.6 Discussion

Les résultats obtenus en classification sont très encourageants si nous souhaitons utiliser le simulateur en tant que moyen de formation. Il semble en effet qu’il soit possible de déterminer une maîtrise du geste à l’aide des données fournies par le simulateur avec une précision tout à fait satisfaisante. Il s’agit cependant d’une première étude et les résultats ne sont pas exempts de commentaires.

En effet, pour commencer, la version du simulateur ayant changé depuis la session d’essais enregistrés, il est possible que si nous souhaitons appliquer les méthodes de classifications mises en œuvre dans ce manuscrit, les résultats ne soient plus satisfaisants. Il est en effet difficile de prévoir l’impact des dernières modifications sur les algorithmes de classification. De plus les données acquises ne permettent pas de savoir si le simulateur permet l’apprentissage du geste de la pose de périodurale à proprement parler. Il n’y a pas de moyen de savoir si les utilisateurs ont pu progresser en utilisant le simulateur. Un protocole expérimental avec groupe témoin et beaucoup

TABLE 5.5 – Évaluation obtenue en agrégeant les classifications de chaque individu en utilisant un algorithme de kPPV sur les données de la *t-SNE* avec $k = 3$

	Classements		Évaluation	
	Novice	Expert	Prédiction	Confiance
Novice 1	3	0	Novice	1
Novice 2	7	1	Novice	0,88
Novice 3	2	0	Novice	1
Novice 4	3	0	Novice	1
Novice 5	4	0	Novice	1
Novice 6	3	1	Novice	0,75
Novice 7	1	1	<i>Indéf.</i>	-
Novice 8	4	0	Novice	1
Novice 9	5	2	Novice	0,71
Novice 10	2	0	Novice	1
Novice 11	5	0	Novice	1
Novice 12	4	0	Novice	1
Expert 1	0	2	Expert	1
Expert 2	1	3	Expert	0,75
Expert 3	0	3	Expert	1
Expert 4	0	4	Expert	1
Expert 5	1	3	Expert	0,75
Expert 6	0	2	Expert	1
Expert 7	0	2	Expert	1
Expert 8	0	5	Expert	1
Expert 9	3	3	<i>Indéf.</i>	-
Expert 10	1	5	Expert	0,83
Expert 11	1	7	Expert	0,88
Expert 12	1	2	Expert	0,67

plus d'apprenants est à prévoir, si le simulateur a pour vocation d'aider à la formation. En l'état il permet de donner un état des lieux de la maîtrise du geste, ce qui est un premier pas mais pas une finalité.

Concernant la classification en elle-même, les deux tiers des mesures servant à l'établissement du *set* d'entraînement, il ne reste qu'une faible partie des mesures pour servir de tests. Ainsi avec la méthode proposé ci-dessus, certains utilisateurs ne présentent que peu d'essais. Il est donc assez difficile de tirer des conclusions sur la qualité intrinsèque du résultat présenté et de nouvelles mesures pourraient être intéressantes afin d'étudier plus précisément une telle méthode. Enfin, une autre piste concernant l'évaluation de performance pourrait être de considérer les essais en les rassemblant par utilisateur mais en discriminant par type de patient. Cela réduirait cependant le nombre d'essais dans le *set* d'entraînement pour chaque patient. Toutefois, le système d'évaluation proposé semble à même de produire un résultat satisfaisant même sans traiter les types de patients séparément.

Conclusion

Les essais effectués sur le simulateur développé tout au long de ces travaux ont permis de mettre à l'épreuve la conception du simulateur ainsi que les différents choix qui ont pu être fait. Cela a été l'occasion de mettre en avant des pistes d'amélioration dont certaines ont pu être implémentées comme par exemple la présence d'un suivi de trajectoire plus souple lorsque l'aiguille est située dans la peau ou encore les frottements secs dans le ligament jaune.

Les essais réalisés ont aussi permis d'acquérir suffisamment de données pour commencer à envisager la mise en place d'algorithmes de classification et d'évaluation de la performance. Les premières analyses ont permis de mettre en avant les points importants concernant la maîtrise du geste comme l'*overshoot*, confirmant ce que l'on peut trouver dans la littérature sur l'anesthésie péridurale, ou encore la régularité du mouvement (à travers l'étude des variances de vitesses et d'accélération) que nous retrouvons aussi classiquement lorsque l'on s'intéresse à l'étude du geste.

Enfin des analyses plus poussées ont permis de proposer les prémices d'un système d'évaluation basées sur les mesures. Celui-ci n'est pas parfait et pourrait bien entendu être approfondi, toutefois les résultats obtenus sont tout à fait encourageants et peuvent d'ores et déjà être testés sur de nouvelles mesures.

Conclusion générale

Les travaux présentés dans ce manuscrit ont permis le développement d'un prototype de simulateur permettant d'apprendre l'anesthésie péridurale. Pour ce simulateur, il a été nécessaire de travailler au contrôle d'un vérin pneumatique afin de reproduire une sensation de perte de résistance précise et réaliste mais aussi personnalisable.

Ces travaux ont aussi été l'occasion de développer une application permettant l'utilisation d'une interface haptique dans le but de simuler l'insertion d'une aiguille. Du fait de la simplicité des classes mises en œuvre, les développements faits sur cette application en particulier permettent de reproduire un vaste échantillon de morphologies différentes et pourraient tout à fait être utilisés pour d'autres types d'insertion d'aiguilles.

Le simulateur ainsi proposé met à disposition une simulation complète et réaliste de la procédure et se démarque de nombreuses autres applications haptiques par l'utilisation de deux type d'actionnements différents.

Le simulateur a pu être mis à l'épreuve lors de tests réalisés en présence de futurs médecins mais aussi d'experts qui ont donc pu attester du réalisme de la simulation proposée mais aussi fournir des premières mesures afin d'étudier la viabilité d'un tel outil dans un contexte de formation. Il a donc été possible de mettre en œuvre des algorithmes d'évaluation de la maîtrise du geste de la pose de péridurale. Les premiers résultats ont été très encourageants et de bon augure pour l'avenir de ce simulateur malgré les réserve vis-à-vis des résultats obtenus.

En résumé les contributions de ces travaux sont les suivants :

- développement d'une interface haptique hybride (électrique pneumatique) spécialisée dans la reproduction du geste de l'anesthésie péridurale ;
- développement de lois de commandes et d'un algorithme pour reproduire le comportement d'une seringue à l'aide d'un vérin pneumatique ;
- développement d'une application permettant le contrôle d'une interface haptique Haption Virtuose 6D pour simuler l'insertion d'aiguille ;
- acquisition de données d'essais novices/experts et tests d'algorithmes de classifications classiques pour déterminer le niveau de maîtrise du geste ;
- proposition d'une première méthode d'évaluation de la performance basée sur ces méthodes de classification.

Concernant les améliorations à apporter et les perspectives de ces travaux, plusieurs pistes restent à explorer. Le simulateur développé s'est limité à l'utilisation d'une interface haptique électrique Haption Virtuose 6D. Cette solution a permis l'implémentation de la simulation mais l'architecture du bras haptique pose certaines contraintes. Les efforts maximums en continu, bien que suffisants sont utilisés à leur valeur maximale. Il pourrait être intéressant de se pencher sur l'utilisation d'une solution à architecture parallèle comme le Force Dimension (Figure 2.1a). Ce type de solution peut produire des efforts bien plus importants et pourrait simplifier certaines parties de la simulation comme le rendu du contact dur ou le frottement sec de type *stick-slip*.

Il pourrait aussi être intéressant de trouver un moyen de rendre le point d'insertion modifiable afin que l'utilisateur puisse le choisir lui-même. Dans le même ordre d'idée il serait nécessaire de permettre la représentation de plusieurs positions du patient (assis ou couché par exemple). Il serait aussi pertinent de complexifier le modèle du dos utilisé dans l'application pour le rapprocher de la réalité. Ce serait un bon moyen de simuler les cas où le chemin à suivre pour atteindre l'espace péridural n'est pas trivial.

Finalement, un certain nombre d'améliorations esthétiques et pratiques pourraient être nécessaires pour rendre le simulateur et son application plus simple d'utilisation.

Publications et communications

Ce doctorat a donné lieu à plusieurs publications et communications parmi lesquelles :

- une communication au congrès de l'IFAC 2017 [46] à Toulouse concernant les lois de commande utilisées sur le vérin pneumatique pour reproduire l'effet de perte de résistance,
- une communication au congrès ICRA 2019 [67] à Montréal concernant les premiers résultats obtenus sur le simulateur complet et la conception générale de celui-ci,
- un article paru au journal CEP [68] concernant la mise en œuvre de l'algorithme de commande du vérin pneumatique,
- un article paru dans *Multimodal Technologies Interact.* [69] concernant l'état de l'art de la simulation médicale et en particulier l'utilisation d'actionneurs pneumatiques dans les simulations haptiques.

Parmi ces articles, la publication de l'IFAC a été ajoutée en Annexe A de ce manuscrit car elle présente des résultats de lois de commandes qui n'ont pas été présentées au cœur de ce manuscrit.

Bibliographie

- [1] C. KONRAD, G. SCHUPFER, M. WIETLISBACH, and H. GERBER, “Learning Manual Skills in Anesthesiology : Is There a Recommended Number of Cases for Anesthetic Procedures?” *Anesthesia & Analgesia*, vol. 86, no. 3, p. 635, Mar. 1998. [Online]. Available : https://journals.lww.com/anesthesia-analgesia/Abstract/1998/03000/Learning_Manual_Skills_in_Anesthesiology__Is_There.37.aspx
- [2] J.-C. GRANRY and M.-C. MOLL, “État de l’art (national et international) en matière de pratiques de simulation dans le domaine de la santé,” Haute Autorité de la Santé (HAS), Tech. Rep., 2012.
- [3] N. VAUGHAN, V. N. DUBEY, M. Y. WEE, and R. ISAACS, “A review of epidural simulators : Where are we today?” *Medical Engineering & Physics*, vol. 35, no. 9, pp. 1235–1250, Sep. 2013. [Online]. Available : <http://linkinghub.elsevier.com/retrieve/pii/S135045331300057X>
- [4] B. L. LEIGHTON, “A greengrocer’s model of the epidural space,” *Anesthesiology*, vol. 70, no. 2, pp. 368–369, Feb. 1989.
- [5] J. C. MAGILL, M. F. BYL, M. F. HINDS, W. AGASSOUNON, S. D. PRATT, and P. E. HESS, “A Novel Actuator for Simulation of Epidural Anesthesia and Other Needle Insertion Procedures :,” *Simulation in Healthcare : The Journal of the Society for Simulation in Healthcare*, vol. 5, no. 3, pp. 179–184, Jun. 2010. [Online]. Available : <https://insights.ovid.com/crossref?an=01266021-201006000-00009>
- [6] V. Manoharan, D. van Gerwen, J. J. van den Dobbelsteen, and J. Dankelman, “Design and validation of an epidural needle insertion simulator with haptic feedback for training resident anaesthesiologists,” in *Haptics Symposium (HAPTICS), 2012 IEEE*. IEEE, 2012, pp. 341–348.
- [7] L. HIEMENZ, J. McDONALD, D. STREDNEY, and D. SESSANNA, “A physiologically valid simulator for training residents to perform an epidural block,” in *Proceedings of the 1996 Fifteenth Southern Biomedical Engineering Conference*, Mar. 1996, pp. 170–173.
- [8] D. THAO, t. T. M, and t. M. A, “Development and Evaluation of an Epidural Injection Simulator with Force Feedback for Medical Training,” *Studies in Health Technology and Informatics*, pp. 97–102, 2001. [Online]. Available : <http://www.medra.org/servlet/aliasResolver?alias=iospressISSNISBN&issn=0926-9630&volume=81&spage=97>

- [9] J. MAGILL, B. ANDERSON, and D. ROSEN, “A Multi-Axis Mechanical Simulator for Epidural Needle Insertion,” Cambridge, MA, 2nd International Conference on Medical Simulation, 2004.
- [10] D. TRAN and KING-WEI Hor, “Instrumentation of the Loss-of-Resistance Technique for Epidural Needle Insertion,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3. [Online]. Available : <http://ieeexplore.ieee.org/document/4760227/>
- [11] M. BOUZIT, G. BURDEA, G. POPESCU, and R. BOIAN, “The Rutgers Master II-new design force-feedback glove,” *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 256–263, Jun. 2002.
- [12] V. DUBEY, N. VAUGHAN, M. Y. K. WEE, and R. ISAACS, “Biomedical Engineering in Epidural Anaesthesia Research,” in *Practical Applications in Biomedical Engineering*, A. Andrade, Ed. InTech, Jan. 2013. [Online]. Available : <http://www.intechopen.com/books/practical-applications-in-biomedical-engineering/biomedical-engineering-in-epidural-anaesthesia-research>
- [13] N. HERZIG, R. MOREAU, and T. REDARCE, “A new design for the BirthSIM simulator to improve realism,” in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 2065–2068.
- [14] N. HERZIG, “De la Conception à la Commande d’une Nouvelle Interface Haptique 4 Axes Hybride Pneumatique Électrique pour la Simulation d’Accouchement : Le BirthSIM,” Ph.D. dissertation, 2016.
- [15] R. MOREAU, “Le simulateur d’accouchement BirthSIM : un outil complet pour la formation sans risque en obstétrique,” Dec. 2007. [Online]. Available : <https://tel.archives-ouvertes.fr/tel-00267771>
- [16] “Accueil - HAPTION SA,” Jul. 2019. [Online]. Available : <https://www.haption.com/fr/>
- [17] “Scanners and Haptics,” Jul. 2019. [Online]. Available : <https://www.3dsystems.com/scanners-haptics>
- [18] S. CHOI and H. TAN, “Effect of update rate on perceived instability of virtual haptic texture,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 4. Sendai, Japan : IEEE, 2004, pp. 3577–3582. [Online]. Available : <http://ieeexplore.ieee.org/document/1389970/>
- [19] R. KILCHENMAN and M. GOLDFARB, “Force saturation, system bandwidth, information transfer, and surface quality in haptic interfaces,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 2, May 2001, pp. 1382–1387 vol.2.
- [20] N. DIOLAITI, G. NIEMEYER, F. BARBAGLI, and J. SALISBURY, “Stability of Haptic Rendering : Discretization, Quantization, Time Delay, and Coulomb Effects,” *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 256–268, Apr. 2006. [Online]. Available : <http://ieeexplore.ieee.org/document/1618739/>
- [21] J. E. COLGATE and G. SCHENKEL, “Passivity of a class of sampled-data systems : Application to haptic interfaces,” *J. Field Robotics*, vol. 14, pp. 37–47, 1997.

- [22] C. B. ZILLES and J. K. SALISBURY, “A constraint-based god-object method for haptic display,” in *Intelligent Robots and Systems 95. Human Robot Interaction and Cooperative Robots*, *Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1995, pp. 146–151.
- [23] K. KUCHENBECKER, J. FIENE, and G. NIEMEYER, “Improving contact realism through event-based haptic feedback,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 2, pp. 219–230, Mar. 2006. [Online]. Available : <http://ieeexplore.ieee.org/document/1580456/>
- [24] M. t. R. MINSKY, “Computational haptics : the sandpaper system for synthesizing texture for a force-feedback display,” PhD Thesis, Massachusetts Institute of Technology, 1995.
- [25] J. P. FRITZ and K. E. BARNER, “Stochastic models for haptic texture,” in *Telemanipulator and Telepresence Technologies III*, vol. 2901. International Society for Optics and Photonics, 1996, pp. 34–45.
- [26] “OSATS.” [Online]. Available : <http://www.csats.com/osats>
- [27] “GOALS.” [Online]. Available : <http://www.csats.com/goals>
- [28] S. COTIN, N. STYLOPOULOS, M. OTTENSMEYER, P. NEUMANN, D. RATTNER, and S. DAWSON, “Metrics for Laparoscopic Skills Trainers : The Weakest Link !” in *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2002*, G. Goos, J. Hartmanis, J. van Leeuwen, T. Dohi, and R. Kikinis, Eds. Berlin, Heidelberg : Springer Berlin Heidelberg, 2002, vol. 2488, pp. 35–43. [Online]. Available : http://link.springer.com/10.1007/3-540-45786-0_5
- [29] C. FENG, J. W. ROZENBLIT, and A. HAMILTON, “Fuzzy Logic-Based Performance Assessment in the Virtual, Assistive Surgical Trainer (VAST),” in *15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ecbs 2008)*, Mar. 2008, pp. 203–209.
- [30] S. K. MISHRA, A. GANPULE, A. KURIEN, V. MUTHU, and M. R. DESAI, “Task completion time : Objective tool for assessment of technical skills in laparoscopic simulator for urology trainees,” *Indian journal of urology : IJU : journal of the Urological Society of India*, vol. 24, no. 1, pp. 35–38, Jan. 2008.
- [31] J. A. CIFUENTES QUINTERO, “Development of a new technique for objective assessment of gestures in mini-invasive surgery,” phdthesis, INSA de Lyon, Jul. 2015. [Online]. Available : <https://tel.archives-ouvertes.fr/tel-01368173/document>
- [32] L. van ADRICHEM, “Avoiding overshoot,” *DSPE Mikroniek*, vol. 3, pp. 36–40, 2009.
- [33] Anthony COMETA, “When Millimeters Count, Epidural Loss of Resistance Techniques Differ : A Simulator Study,” Miami, May 2018.
- [34] A. OKAMURA, C. SIMONE, and M. O’LEARY, “Force Modeling for Needle Insertion Into Soft Tissue,” *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 10, pp. 1707–1716, Oct. 2004. [Online]. Available : <http://ieeexplore.ieee.org/document/1337139/>
- [35] Force Dimension, “Force Dimension - delta.3 haptic device.” [Online]. Available : <http://www.forcedimension.com/downloads/specs/specsheet-delta.3.pdf>

- [36] 3D Systems, “Haptic Devices.” [Online]. Available : <https://fr.3dsystems.com/haptics-devices/touch>
- [37] Haption SA, “Spécification détaillée Virtuose 6d Desktop.” [Online]. Available : <https://downloads.haption.com/marketing/datasheet/Datasheet-Virtuose6DDesktop-2018.pdf>
- [38] A. TALHAN and S. JEON, “Pneumatic Actuation in Haptic-Enabled Medical Simulators : A Review,” *IEEE Access*, vol. 6, pp. 3184–3200, 2018.
- [39] Airpot, “Double-Acting Archives.” [Online]. Available : <http://airpot.com/product-category/product-lines/pneumatic-actuation/airpel-anti-stiction-air-cylinders/metric-models/double-acting-metric-models/>
- [40] SICK, “MPS-064tsnu0 | Magnetic cylinder sensors SICK.” [Online]. Available : <https://www.sick.com/ag/en/magnetic-cylinder-sensors/position-sensors/mps-t/mps-064tsnu0/p/p230381>
- [41] TE Connectivity, “Miniature Ratiometric Pressure Transducer | EPRB Series | TE Connectivity.” [Online]. Available : <http://www.te.com/usa-en/product-CAT-PTT0007.html>
- [42] F. ABRY, “Contribution à la commande et l’observation des actionneurs électropneumatiques : de l’intérêt de la transformée A-T,” Ph.D. dissertation, L’institut national des sciences appliquées de Lyon, 2013.
- [43] F. ABRY, X. BRUN, S. SESMAT, and E. BIDEAUX, “Non-linear position control of a pneumatic actuator with closed-loop stiffness and damping tuning,” in *Proc. of the 2013 European Control Conference (ECC)*, Zürich, Switzerland, 2013, p. 1089. [Online]. Available : <http://www.nt.ntnu.no/users/skoge/prost/proceedings/ecc-2013/data/papers/0463.pdf>
- [44] R. PARK, “Two reaction theory of synchronous machines,” *AIEE Transactions*, vol. 48.
- [45] N. HERZIG, R. MOREAU, A. LELEVÉ, and M. T. PHAM, “Stiffness Control of Pneumatic Actuators to Simulate Human Behavior on Medical Haptic Simulators,” in *IEEE International Conference on Advanced Intelligent Mechatronics*, ser. Proc. of the IEEE International Conference on Advanced Intelligent Mechatronics, IEEE, Ed. Banff, Canada : IEEE, Jul. 2016. [Online]. Available : <https://hal.archives-ouvertes.fr/hal-01333383>
- [46] T. SÉNAC, A. LELEVÉ, and R. MOREAU, “Control laws for pneumatic cylinder in order to emulate the loss of resistance principle,” in *20th IFAC World Congress*. IFAC, 2017.
- [47] J. P. HESPANHA and A. S. MORSE, “Stability of switched systems with average dwell-time,” in *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*, vol. 3, 1999, pp. 2655–2660 vol.3.
- [48] RUSSELL Smith, “OPEN DYNAMICS ENGINE V0.5 USER GUIDE,” May 2004.
- [49] V. CHAWDA, O. CELIK, and M. K. O’MALLEY, “Application of Levant’s differentiator for velocity estimation and increased Z-width in haptic interfaces,” in *2011 IEEE World Haptics Conference*. Istanbul : IEEE, Jun. 2011, pp. 403–408. [Online]. Available : <http://ieeexplore.ieee.org/document/5945520/>

- [50] M. K. CHMARRA, S. KLEIN, J. C. F. de WINTER, F.-W. JANSEN, and J. DANKELMAN, “Objective classification of residents based on their psychomotor laparoscopic skills,” *Surgical Endoscopy*, vol. 24, no. 5, pp. 1031–1039, May 2010. [Online]. Available : <http://link.springer.com/10.1007/s00464-009-0721-y>
- [51] A. GHASEMLOONIA, Y. MADDAHI, K. ZAREINIA, S. LAMA, J. C. DORT, and G. R. SUTHERLAND, “Surgical Skill Assessment Using Motion Quality and Smoothness,” *Journal of Surgical Education*, vol. 74, no. 2, pp. 295–305, Mar. 2017. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S1931720416302100>
- [52] S. ESTRADA, C. DURAN, D. SCHULZ, J. BISMUTH, M. D. BYRNE, and M. K. O’MALLEY, “Smoothness of surgical tool tip motion correlates to skill in endovascular tasks,” *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 5, pp. 647–659, Oct. 2016.
- [53] T. THERNEAU, B. ATKINSON, B. RIPLEY, and M. B. RIPLEY, “Package ‘rpart,’” *Available online : cran. ma. ic. ac. uk/web/packages/rpart/rpart. pdf (accessed on 20 April 2016)*, 2015.
- [54] H. ABDI and L. J. WILLIAMS, “Principal component analysis : Principal component analysis,” *Wiley Interdisciplinary Reviews : Computational Statistics*, vol. 2, no. 4, pp. 433–459, Jul. 2010. [Online]. Available : <http://doi.wiley.com/10.1002/wics.101>
- [55] F. HUSSON, S. LÊ, and J. PAGÈS, *Exploratory multivariate analysis by example using R*, second edition ed., ser. CRC computer science and data analysis series. Boca Raton : CRC Press, 2017.
- [56] H. ABDI, “Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition (GSVD),” *Encyclopedia of Measurement and Statistics.*, Jan. 2007.
- [57] L. v. d. MAATEN and GEOFFREY Hinton, “Visualizing Data using t-SNE,” *Journal of Machine Learning Research*, no. 9, 2008.
- [58] J. KRIJTHE and LAURENS van der Maaten (Author of original C++, “Rtsne : T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation,” Nov. 2018. [Online]. Available : <https://CRAN.R-project.org/package=Rtsne>
- [59] F. HUSSON, J. JOSSE, S. LE, and J. MAZET, “FactoMineR : Multivariate Exploratory Data Analysis and Data Mining,” Jul. 2019. [Online]. Available : <https://CRAN.R-project.org/package=FactoMineR>
- [60] R. B. CATTELL, “The Scree Test For The Number Of Factors,” *Multivariate Behavioral Research*, vol. 1, no. 2, pp. 245–276, Apr. 1966. [Online]. Available : https://doi.org/10.1207/s15327906mbr0102_10
- [61] B. RIPLEY and W. VENABLES, “class : Functions for Classification,” Jan. 2019. [Online]. Available : <https://CRAN.R-project.org/package=class>
- [62] M. HEARST, S. DUMAIS, E. OSUNA, J. PLATT, and B. SCHOLKOPF, “Support vector machines,” *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, Jul. 1998. [Online]. Available : <http://ieeexplore.ieee.org/document/708428/>

- [63] D. MEYER, E. DIMITRIADOU, K. HORNIK, A. WEINGESSEL, F. LEISCH, t. l. C++-code), and t. l. C++-code), “e1071 : Misc Functions of the Department of Statistics, Probability Theory Group (Formerly : E1071), TU Wien,” Jun. 2019. [Online]. Available : <https://CRAN.R-project.org/package=e1071>
- [64] C. M. BISHOP, *Neural networks for pattern recognition*. Oxford : New York : Clarendon Press ; Oxford University Press, 1995.
- [65] S. FRITSCH, F. GUENTHER, and M. F. GUENTHER, “Package ‘neuralnet’,” *Training of Neural Networks*. Available online : *ftp ://64.50*, vol. 236, no. 1, 2019.
- [66] N. Candalh-Touta, “Assistance to the Learning of Laparoscopic Gesture Dexterity,” Theses, Sorbonne Université, Oct. 2018. [Online]. Available : <https://tel.archives-ouvertes.fr/tel-01966556>
- [67] T. SÉNAC, A. LELEVÉ, R. MOREAU, L. KRÄHENBÜHL, F. SIGWALT, C. BAUER, and Q. ROUBY, “Designing an accurate and customizable epidural anesthesia haptic simulator,” 2019.
- [68] T. SÉNAC, A. LELEVÉ, R. MOREAU, L. KRAHENBUHL, F. SIGWALT, and C. BAUER, “Simulating a syringe behavior using a pneumatic cylinder haptic interface,” *Control Engineering Practice*, vol. 90, pp. 231–240, Sep. 2019. [Online]. Available : <https://linkinghub.elsevier.com/retrieve/pii/S0967066119301042>
- [69] T. SÉNAC, A. LELEVÉ, R. MOREAU, C. NOVALES, L. NOUAILLE, M. T. PHAM, and P. VIEYRES, “A Review of Pneumatic Actuators Used for the Design of Medical Simulators and Medical Tools,” *Multimodal Technologies and Interaction*, vol. 3, no. 3, p. 47, 2019.

Annexe A

**Article présenté à la conférence
IFAC 2017 à Toulouse**

Control laws for pneumatic cylinder in order to emulate the Loss Of Resistance principle

Thibault SÉNAC* Arnaud LELEVÉ* Richard MOREAU*

* *Laboratoire Ampère, UMR CNRS 5005 INSA-Lyon, Université de Lyon*

Abstract: Medicine requires advanced teaching methods in order to reach an efficient student training without having to train them directly on patients. In France, the Haute Autorité de la Santé (H.A.S.) has stated to "never [do] the first time with a patient" as a requirement for the training of new doctors. The goal of this work is to offer a novel robotic solution to teach students how to perform an epidural anaesthesia. This medical operation can be divided into two different gestures: first the insertion of a needle between two vertebrae and second the application of pressure on the plunger throughout the insertion of the needle. This work aims at simulating this second part as the first part has already been simulated in previous studies. We introduce a way to emulate the principle of loss of resistance felt by the anaesthetist when the needle reaches the appropriate depth. In order to reproduce the real syringe and its behaviour we propose to use a pneumatic cylinder because of the inherent quality of such an actuator. Indeed, pneumatic actuators have an inherent compliance that is interesting when creating an haptic interface. For instance, it has been used before to create a childbirth simulator.

Keywords: Simulators, Nonlinear control system

1. INTRODUCTION

Epidural anaesthesia, as many other medical specialties, requires a lot of training to be mastered enough to practice it on a real patient. A study by Vaughan et al. (2013) on epidural simulation reported that it required at least 80 anaesthesia to reach a 90% success rate. Thus, such a gesture requires advanced learning process. This is why this article introduces a way of simulating a part of the gesture, in order to train new anesthetist in the future.

During an epidural anaesthesia, the anaesthetist applies a constant pressure on the plunger of the syringe to indirectly locate the epidural space to inject the anaesthetic. During the insertion of the needle between two vertebrae of a patient the anaesthetist may feel an important resistance force which greatly decreases as soon as the needle reaches the epidural space. This is called the principle of *Loss Of Resistance* (LOR). This work introduces various control laws that aim at emulating this principle.

Some epidural simulators developed prior to this work also included a simulation of the LOR principle (Dang et al. (2001) and Magill et al. (2004)) but these simulations did not offer a precise emulation of the LOR principle, only using a valve with a binary type control. The valve is half-closed to create a high resistance and opened to create the loss of resistance. To emulate the phenomenon, we propose to use a pneumatic cylinder controlled by a servo-valve, using two different control patterns. This will enable a more precise control over the haptic feedback the simulator provides. This paper depicts the experimental results obtained with both control laws and it compares both solutions. In order to keep the paper short, stability

demonstrations are not detailed (they can be provided on demand).

2. MATERIAL AND MODELING

2.1 Material

In order to emulate the LOR principle, we use a low friction Airpel[®] pneumatic cylinder coupled with a Festo MPYE[®]-5-M5-010 B servo-valve. In order to set up our control laws we use a position sensor and two pressure transducers (MEAS[®] U5136), one for each cylinder chamber. In this context, the pneumatic cylinder is the equivalent of the syringe used in the original operation and the plunger is equivalent to the piston of this cylinder. In order to emulate the LOR principle we needed to have a precise model of the system so that it is possible to render the feeling of this loss of resistance. A simplified diagram of the pneumatic cylinder is depicted in Fig.1. In this diagram, S is the piston surface which, in our case, is the same in both chambers, both q_m represent mass flows, V_N and V_P volumes, P_N and P_P pressures and T temperatures. The subscripts refer to the chamber of the cylinder. Finally F_{dist} represents the force applied on the cylinder by the user as if he pushed the plunger of a syringe.

2.2 System modeling

The model of the system is mainly based on the one proposed by Abry et al. (2013). This model is based on a classical thermodynamical modeling in which the author added an A-T transform based on Park transform usually used in electrical design. This transform is presented in eq.(1).

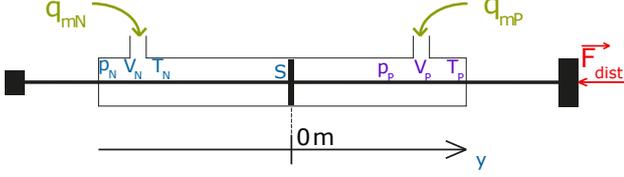


Fig. 1. Simplified diagram of the test bench pneumatic cylinder variables

$$\begin{bmatrix} q_{mA} \\ q_{mT} \end{bmatrix} = \Lambda(y) \cdot \begin{bmatrix} q_{mP} \\ q_{mN} \end{bmatrix} \quad (1)$$

where

$$\Lambda(y) = V_0 \cdot \begin{bmatrix} \frac{1}{V_P} & -\frac{1}{V_N} \\ \frac{1}{V_P} & \frac{1}{V_N} \end{bmatrix} \quad (2)$$

In eq.(1), q_{mA} represents the active flow (which provokes a pneumatic force due to a difference of pressurization of both chambers), q_{mT} represents the pressurization flow (which provides a different stiffness to the piston when pushed by the user), q_{mP} and q_{mN} are the input mass flows in both chambers. V_N and V_P are the volumes of each chambers, y is the position of the piston and V_0 is the total volume of the cylinder.

This modeling provides an easy mean to render a position control and also a force control. This model, presented in eq.(3), allows for two different types of control. First, the control law offered by Abry et al. (2013) controls the position of the piston over time with two parameters: a closed-loop damping and a closed loop stiffness. For our simulation we needed to control the damping of the motion. A high damping can create a high resistance force and a low damping can render a low resistance force. However the damping provided by the control law was not sufficient to achieve this. Then the control law has been utilized with a tweak that allows us to emulate a variable damping using the closed loop stiffness parameter. The second type of control law aims at reproducing a pneumatic force equivalent to the resistance force felt by the user when pushing the plunger.

$$\begin{cases} \frac{dy}{dt} = v \\ \frac{dv}{dt} = \frac{-b \cdot v - F_{fr}(v) + F_{pneu}}{M} \\ \frac{dF_{pneu}}{dt} = -K_{pneu} \cdot v + B_1 \cdot q_{mA} \end{cases} \quad (3)$$

In eq.(3), q_{mA} represents the control input, v the velocity of this piston, b the physical damping of the cylinder, F_{fr} the friction force, F_{pneu} the pneumatic force, M the mass of the piston, K_{pneu} the pneumatic stiffness, $B_1 = \frac{k \cdot R \cdot T \cdot S}{V_0}$ a constant, k the polytropic constant, R the gas constant, V_0 the total volume. These variables are represented in Fig.1. K_{pneu} expression is provided in eq.(3bis)

$$K_{pneu} = \left(\frac{P_P}{V_P} + \frac{P_N}{V_N} \right) \cdot k \cdot S^2 \quad (3bis)$$

Here P_P and P_N represent the pressure in each chambers, as stated in Fig.1.

3. CONTROL LAWS

Based on the model presented in eq.(3), two control approaches have been created, one based on the work of Abry et al. (2013) and using a position control to emulate a variable damping and the other controlling the pneumatic force, F_{pneu} in order to reproduce the real resistance force.

3.1 Emulation of a variable damping

Abry et al. (2013) introduced a position control law for pneumatic cylinders. This law is a backstepping law that has some parameters tied to the closed loop stiffness and damping of the final system. Thanks to these parameters, this control position law has inherent interest for our particular case. Indeed we can use the closed loop stiffness in order to generate the appropriate haptic response. This work presents the following control law:

$$q_{mA} = f_0 + f_1 \cdot z_1 + f_2 \cdot z_2 + f_3 \cdot z_3 \quad (4)$$

with:

$$\begin{aligned} f_0 &= \frac{M^2 \cdot j_d + M \cdot K_{pneu} \cdot v - v \cdot b^2 - b \cdot F_{sec} + F_{pneu} \cdot b}{M \cdot B_1} \\ f_1 &= -\frac{M \cdot (C_1^3 - 2 \cdot C_1 - C_2)}{B_1} \\ f_2 &= \frac{M^2 \cdot (C_1^2 + C_1 \cdot C_2 + C_2^2 - 1) - 1}{M \cdot B_1} \\ f_3 &= -\frac{C_1 + C_2 + C_3}{B_1} \end{aligned}$$

Here C_1 , C_2 , and C_3 are three positive parameters that have to be tuned depending on the expected performance of our law, j_d the desired jerk and $z_1 = y - y_d$, $z_2 = v - v_d + C_1 \cdot z_1$, $z_3 = F_{pneu} - F_{pneu_d}$ the errors needed by the backstepping method. This control law sets a closed loop stiffness, as well as a closed loop damping, through the various parameters that has to be tuned. In Abry et al. (2013) it is shown that with this control law, the closed loop stiffness and damping can be respectively written:

$$K_{cl} = M \cdot (C_1 \cdot C_2 + 1) \quad (5)$$

and

$$B_{cl} = M \cdot (C_1 + C_2) \quad (6)$$

At first we tried to control only the damping by way of the value of B_{cl} but we could not get a proper control of the damping of the system during all the piston motion. But, when combining this equation with the fact that it is possible to get the stiffness of a system with:

$$K_{cl} = \frac{F_{dist}}{\Delta y} \quad (7)$$

(here F_{dist} is the disturbance force applied by the user while pushing the cylinder piston), it is possible to design a variable stiffness that renders a physical damping. Indeed with a physical damping we would get:

$$y = a \cdot t + b \quad (8)$$

as an expression of the piston position over time when the piston is pushed by the user with a constant force, and

where a can be assimilated to a physical damping, t is the time and b a constant that can be related to the initial conditions of the system. It is then possible to obtain this particular stiffness trajectory for our system:

$$K_{cl_d} = \frac{1}{\frac{1}{B_{cl_d}} \cdot (t - t_{in}) + \frac{1}{K_{cl_0}}} \quad (9)$$

Here K_{cl_d} is the desired closed loop stiffness, B_{cl_d} is the desired closed loop damping, t_{in} the time at which we record the pushing on the piston, K_{cl_0} the initial value for the closed loop stiffness, this is usually a high value around 3000 N/m.

To ensure stability of this law we need to make sure that K_{cl_d} has a minimum value strictly positive in time. We can for example choose this value to be around 200 N/m without hindering the results of this law. However, this law remains quite complex, due to the amount of parameters that need to be set in order for the law to work properly. This complexity turns out to be major setback when using it. In the end, simpler control laws have been developed to bypass this tuning issue.

3.2 A pneumatic force control

To avoid the tuning drawbacks of the previous law, pneumatic force control laws have been implemented to emulate the LOR principle. In this respect, two control laws have been developed using two different methods, a sliding mode control, based on a previous work from Shen and Goldfarb (2007), and a backstepping control. Both laws are solely based on this particular equation:

$$\frac{dF_{pneu}}{dt} = -K_{pneu} \cdot v + B_1 \cdot q_{mA} \quad (10)$$

These laws control the pneumatic force to render the resistance force needed to feel the loss of resistance. For these two laws we then define a desired pneumatic force, F_{pneu_d} as the objective of the control. Using eq.(10) we define a backstepping control law with this active flow:

$$q_{mA} = \frac{1}{B_1} \left(K_{pneu} \cdot v + \dot{F}_{pneu_d} \right) - C_{bsf} \cdot z_f \quad (11)$$

Here C_{bsf} is a positive parameter of our control law and $z_f = F_{pneu} - F_{pneu_d}$ is the error for the backstepping control. Finally, based on Shen and Goldfarb (2007), we define a similar kind of control using a sliding mode approach:

$$q_{mA} = B_1^{-1} (\dot{F}_{pneu_d} + K_{pneu} \cdot v - \kappa \cdot \text{sgn}(s)) \quad (12)$$

Here s is the sliding surface defined as such: $s = F_{pneu} - F_{pneu_d}$ and κ a parameter to choose. The final idea with these two laws is to choose the pneumatic force according to the resistance force we need to render the LOR haptic feedback.

4. PROTOCOL AND EXPERIMENTAL RESULTS

4.1 Protocol

Both control laws have been thoroughly tested in simulation prior to being implemented on our test bench. Various tests have been performed, such as step response, response

to a sine wave, disturbance rejection and finally simulation of the LOR principle. All results are quite encouraging for both step response, response to a sine wave and disturbance rejection with the pneumatic force control. It is not the case for the damping emulation for which it has been difficult to get good results for stiffness control.

We introduce the results of the LOR simulation, for the force control by sliding mode in Fig.3, the force control by backstepping in Fig.4 and the damping simulation control by backstepping in Fig.5. To simulate the LOR principle, the cylinder length is divided in two parts. The first part (from 0.035 m to -0.02 m for the force control and from 0.035 m to 0 m for the damping emulation) represents the highly resistant part and the second part (from -0.02 m to -0.04 m for the force control and from 0 m to -0.04 m for the damping emulation) is the less resistant part. At first the piston is fully pulled into the resistant part. When the user pushes on the piston, he feels a strong resistance force. Then, when the piston arrives at frontier position, the desired force or damping changes to create a lower resistance force. At this moment the user must realize that he is at the point of loss of resistance and stop pushing on the piston. If the moment of LOR triggering is clear to the user then the simulation is successful.

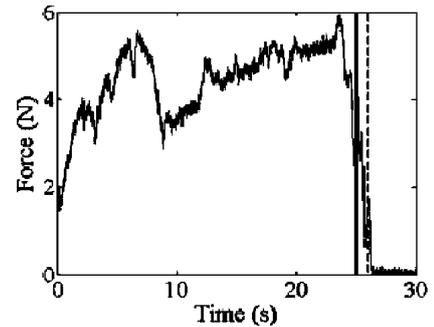


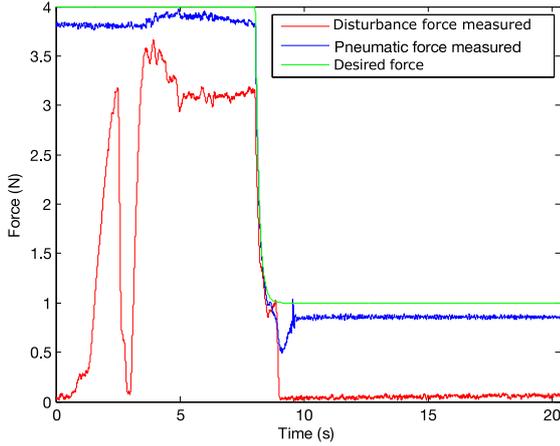
Fig. 2. Experimental measures of the resistance force applied on the plunger of an epidural syringe (Tran et al. (2009))

4.2 Discussion

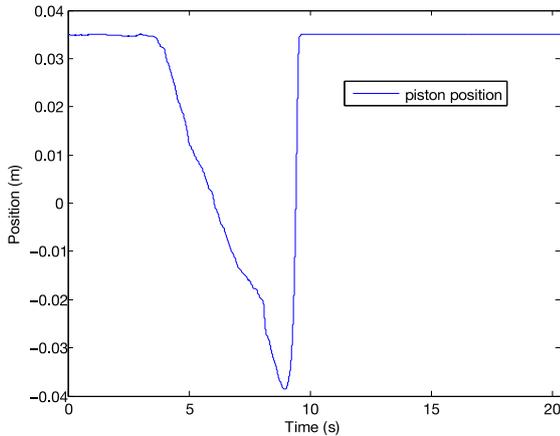
The results presented in Fig. 3 and Fig. 4 show a very good response both in terms of step response and disturbance rejection for the force control laws.

In these figures, we present the disturbance force applied by the user on the cylinder piston, the desired pneumatic force during the test, and the actual pneumatic force generated by the pneumatic cylinder. These results show that these laws are perfectly able to cope with a disturbance force without creating a high relative error. The different tests have shown the sliding mode chattering was an issue so the backstepping law provides a better quality haptic feedback.

The damping control law provides the results presented in Fig. 5. The stiffness control is not accurate with this type of control which hinders the accuracy of the damping control. Though, observing the pneumatic force generated in response to the disturbance force, it remains close to the experimental force measured in Tran et al. (2009) and depicted in Fig. 2. Moreover the haptic feedback also allows



(a)



(b)

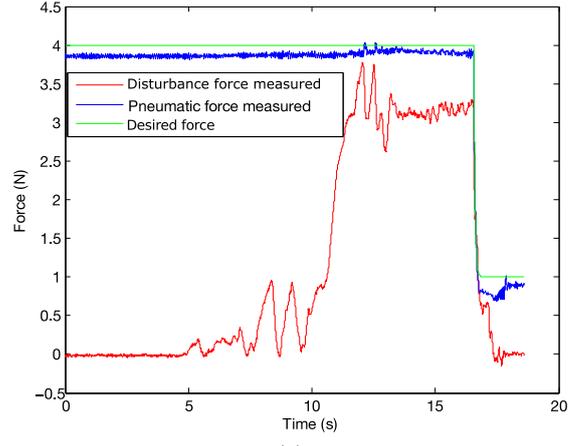
Fig. 3. (a) Disturbance force, pneumatic force recorded, and desired force for the Sliding mode control law (b) Position of the piston during this test

the user to detect the occurrence of the LOR principle. The difference between this law and the others is that the damping control law allows only a smooth transition between the high resistance and the low resistance parts. It still bears interest to render some of the hardest cases when the transition must be rendered unclear and the LOR triggering may be more subtle.

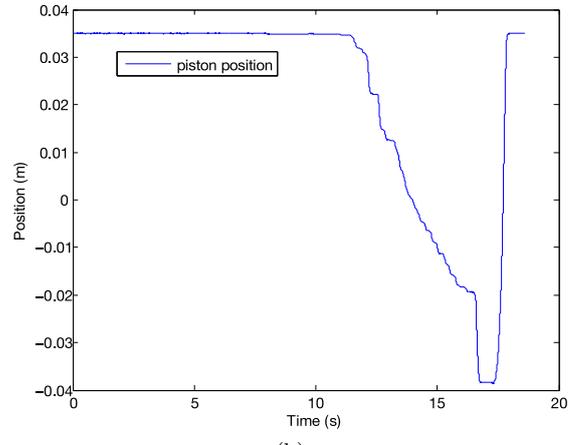
Finally, these control laws have been tested by an anaesthetist in order to validate the haptic feedback, to state whether the haptic feedback is close enough to the real haptic feedback of a syringe. His intervention also helped to tune some control parameters such as the norm of the resistance force, F_{pneu_d} , or the damping, B_{cl_d} , needed to emulate the feeling of the real syringe. The tests have brought very interesting results showing that this hardware simulation was close enough to the reality to be considered usable as a part of a complete simulator.

5. CONCLUSION

Both force control laws ended up giving very good experimental results and providing a high quality haptic feedback for the user. As far as the simulation of the LOR principle is concerned these laws provide a very good control of the haptic feedback. In the example depicted



(a)



(b)

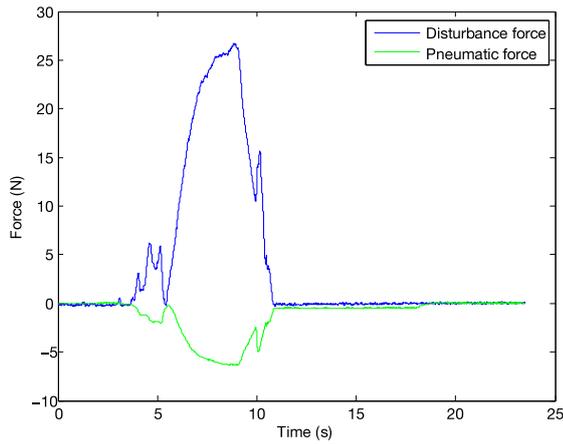
Fig. 4. (a) Disturbance force, pneumatic force recorded, and desired force recorded for the Backstepping control law (b) Position of the piston during this test

in this paper, for the force control laws we used filtered step inputs as a way to prevent an overshoot in the pneumatic force response. This filtering allows to control the pneumatic force decrease speed and thus to control the haptic feedback efficiently. For example, it can be used to emulate different kinds of patients (aged, calcified, ...) and create a more realistic LOR simulation. To widen the range of application of these results, this force control law may also be used for any simulation of a syringe. In our case it should be used in the future to simulate the syringe of various articular puncture operations.

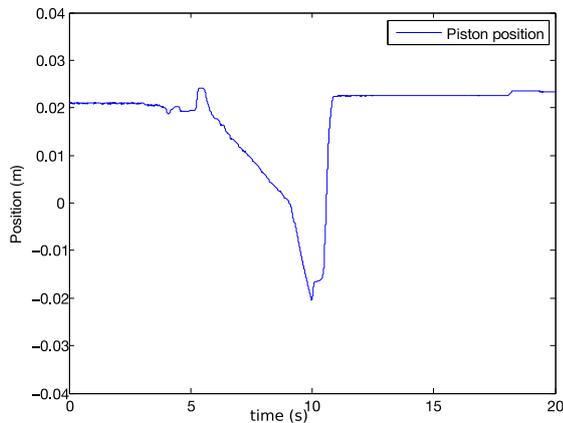
$$\begin{aligned} z_1 &= y - y_d \\ z_2 &= v - v_d + C_1 \cdot z_1 \\ z_3 &= F_{pneu} - F_{pneu_d} \end{aligned}$$

REFERENCES

- Abry, F., Brun, X., Sesmat, S., and Bideaux, E. (2013). Non-linear position control of a pneumatic actuator with closed-loop stiffness and damping tuning. In *European Control Conference (ECC)*, 1089. Zürich, Switzerland.
- Dang, T., Annaswamy, T.M., and Srinivasan, M.A. (2001). Development and evaluation of an epidural injection simulator with force feedback for medical training. *Studies in Health Technology and Informatics*, 81, 97–102.



(a)



(b)

Fig. 5. (a) Disturbance and pneumatic force recorded for the Backstepping control law (b) Piston position during the test (the most resistant part is for a position between 0.04 m and 0 m and the least resistant part is between 0 m and -0.04 m)

Magill, J., Anderson, B., Anderson, G., Hess, P., and Pratt, S. (2004). Multi-axis Mechanical Simulator for Epidural Needle Insertion. In *Medical Simulation*, 267–276. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-25968-8_30.

Shen, X. and Goldfarb, M. (2007). Simultaneous Force and Stiffness Control of a Pneumatic Actuator. *Journal of Dynamic Systems, Measurement, and Control*, 129(4), 425. doi:10.1115/1.2745850.

Tran, D., King-Wei Hor, Kamani, A., Lessoway, V., and Rohling, R. (2009). Instrumentation of the Loss-of-Resistance Technique for Epidural Needle Insertion. *IEEE Transactions on Biomedical Engineering*, 56(3), 820–827. doi:10.1109/TBME.2008.2011475.

Vaughan, N., Dubey, V.N., Wee, M.Y., and Isaacs, R. (2013). A review of epidural simulators: Where are we today? *Medical Engineering & Physics*, 35(9), 1235–1250. doi:10.1016/j.medengphy.2013.03.003. URL <http://linkinghub.elsevier.com/retrieve/pii/S135045331300057X>.

Name Definition

HAS	Haute Autorité de la Santé
F_{pneu}	Pneumatic force (N)
S	Piston area (m)
p_N, p_P	Pressure in both cylinder chamber (bar)
p_a	Atmospheric pressure (bar)
F_{fr}	Friction force (N)
y	Piston position (m)
v	Piston velocity (m/s)
V_N et V_P	Chamber volumes (m ³)
V_0	Total volume (m ³)
T_P, T_N	Temperature in each chamber (K)
k	Polytropic constant
r	Gas constant
q_{mP}, q_{mN}	Air flows in each chamber (kg/s)
q_{mA}	Active massic air flows (kg/s)
q_{mT}	Pressurization massic air flow (kg/s)
K_{pneu}	Pneumatic stiffness (N/m)
K_{cl}	Closed loop stiffness (N/m)
$j_d, a_d, v_d,$ y_d, F_{pneu_d}	Desired jerk (m/s ³), acceleration (m/s ²), velocity (m/s), position (m) and force (N)

NOMENCLATURE

Annexe B

Résultats d'essais supplémentaires utilisant l'algorithme de contrôle de vérin pneumatique

Cette annexe présente quelques mesures supplémentaires obtenues en utilisant l'algorithme de contrôle présenté dans le chapitre 3 sur le simulateur final. Le vérin instrumenté utilisé est donc celui présenté en Figure B.1. Les résultats présentés sont constitués de 6 essais réalisés par des experts et de 6 essais réalisés par des novices.

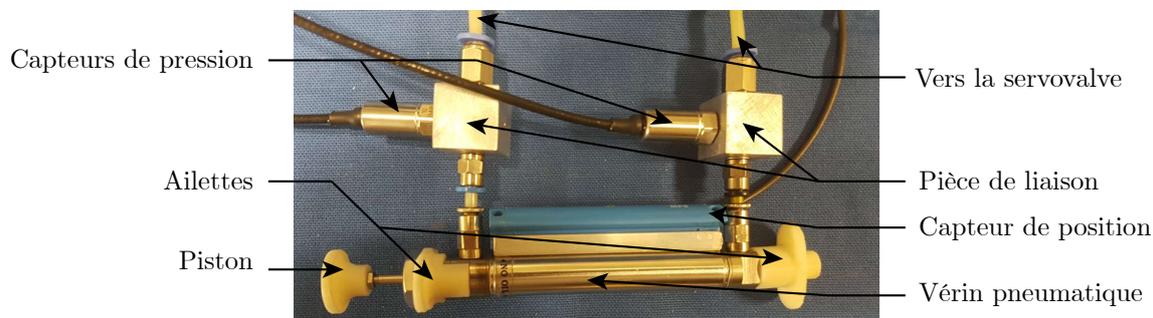
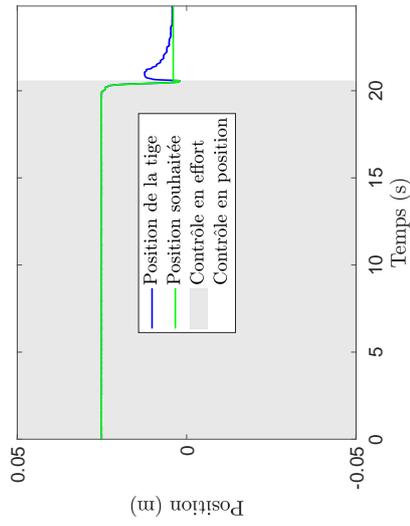
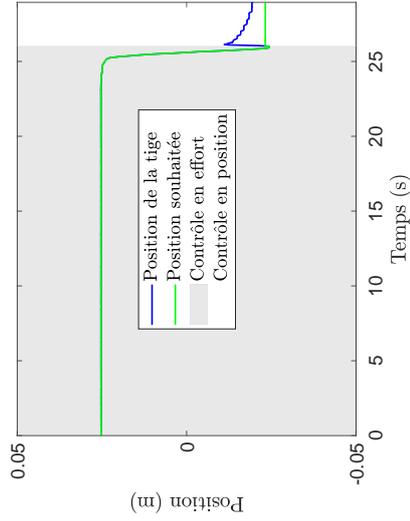


FIGURE B.1 – Vérin instrumenté utilisé pour reproduire le comportement d'une seringue

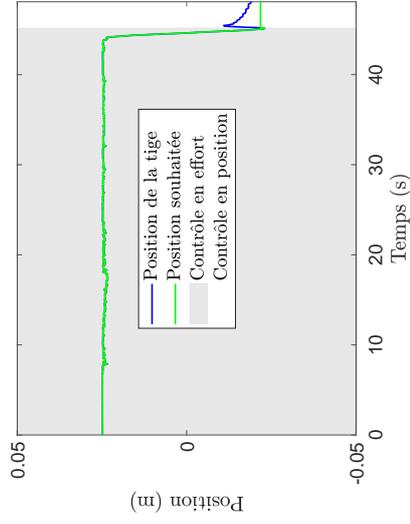
Essais d'experts



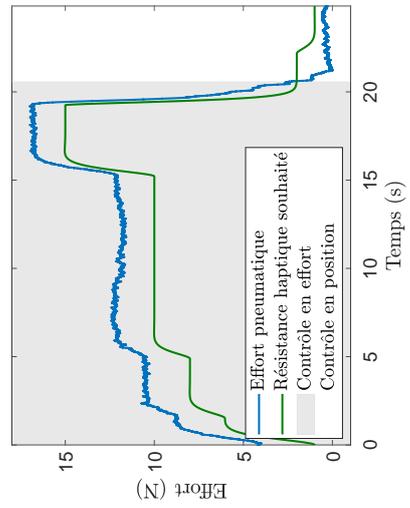
(a) y et y_d



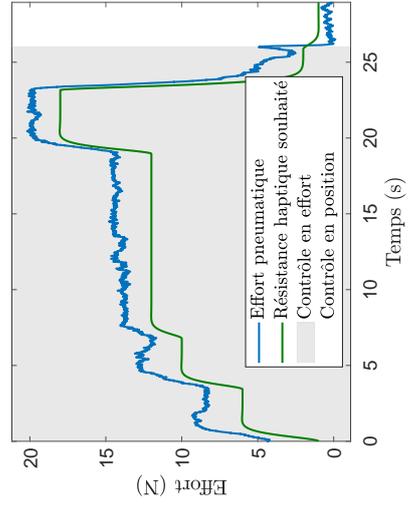
(b) y et y_d



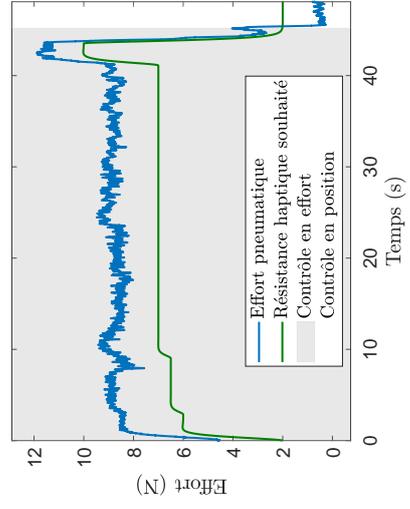
(c) y et y_d



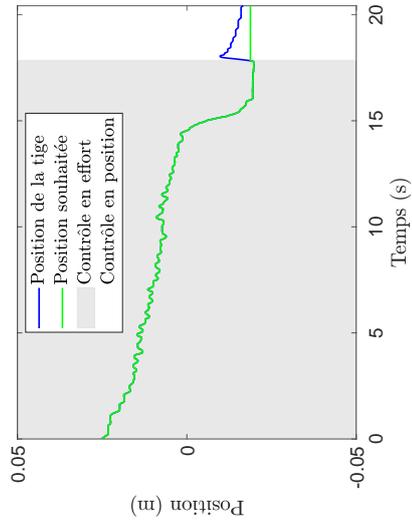
(d) F_{pneu} et F_d^{hap}



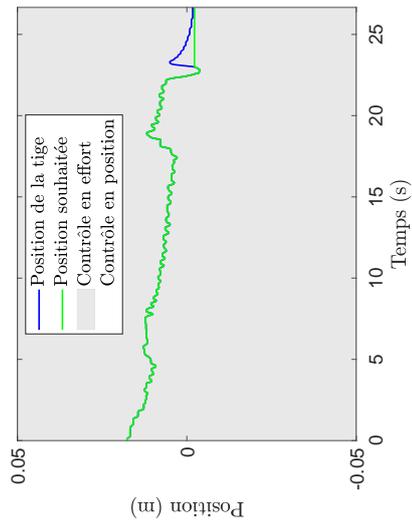
(e) F_{pneu} et F_d^{hap}



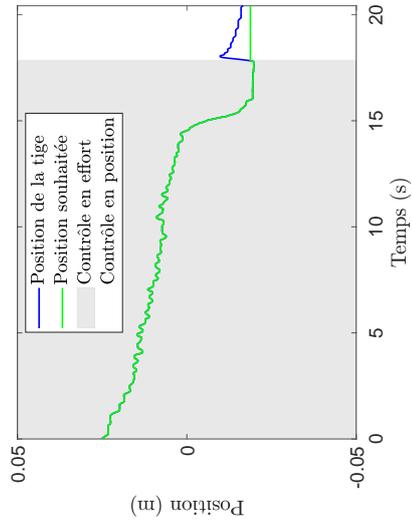
(f) F_{pneu} et F_d^{hap}



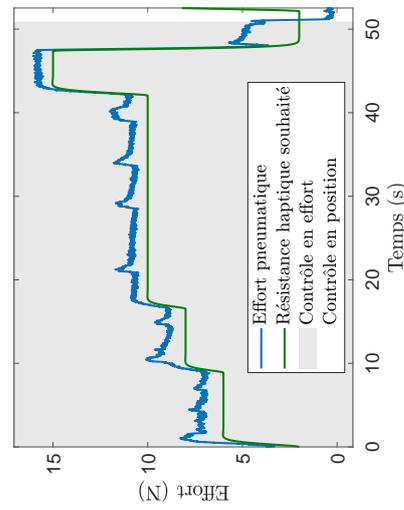
(a) y et y_d



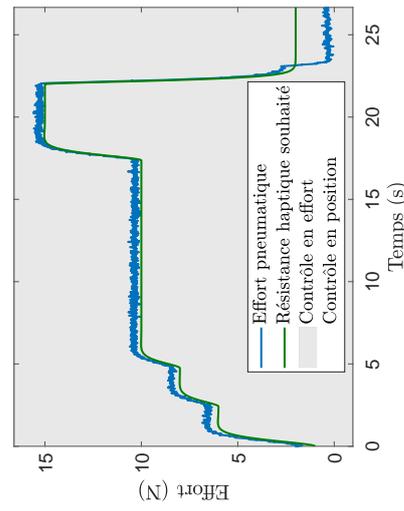
(b) y et y_d



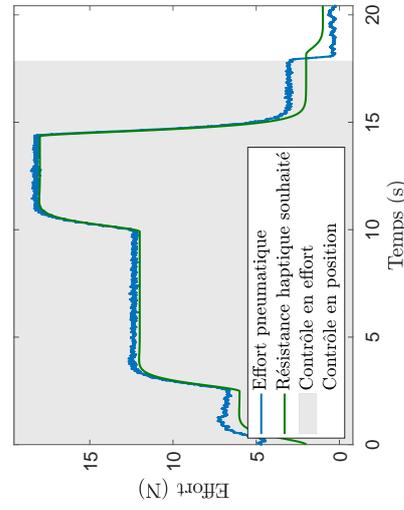
(c) y et y_d



(d) F_{pneu} et F_d^{hap}

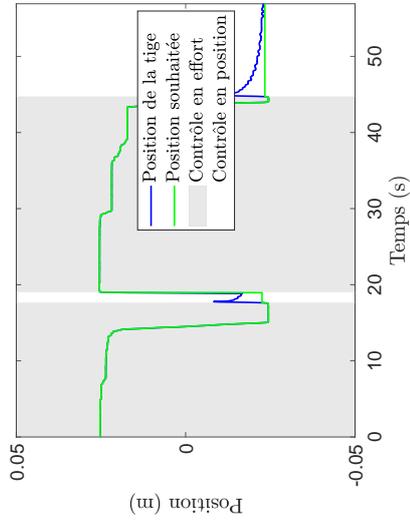


(e) F_{pneu} et F_d^{hap}

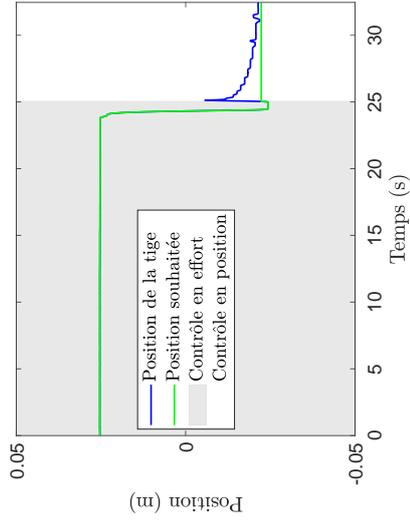


(f) F_{pneu} et F_d^{hap}

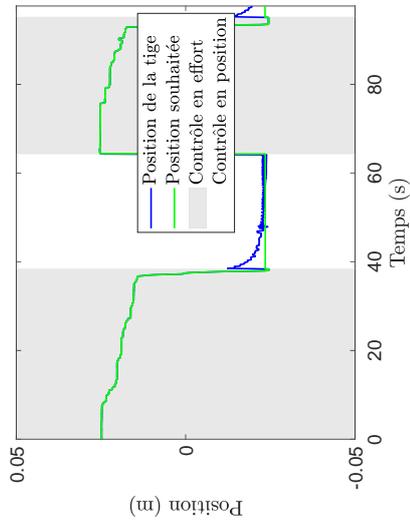
Essais de novices



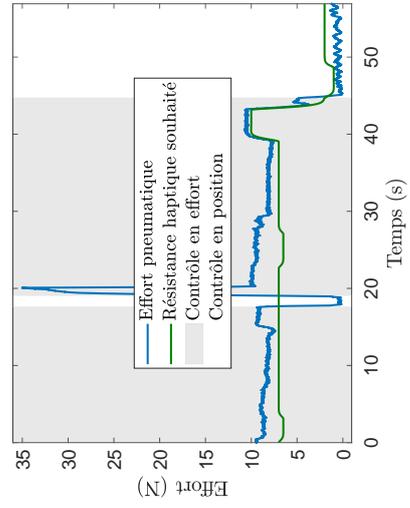
(a) y et y_d



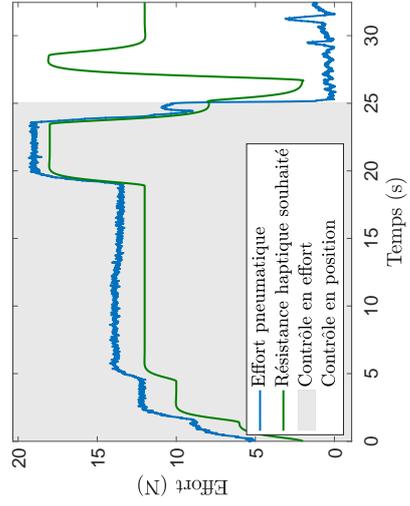
(b) y et y_d



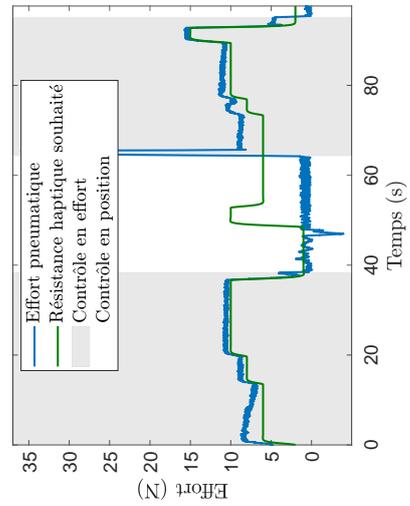
(c) y et y_d



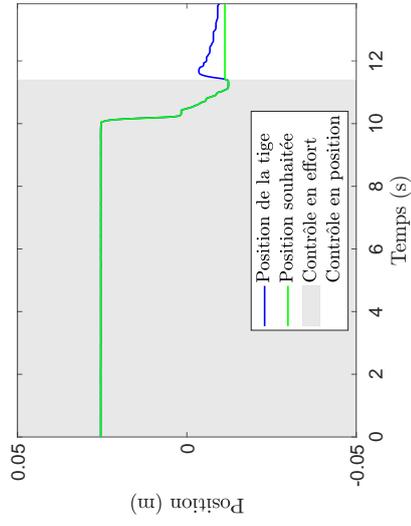
(d) F_{pneu} et F_d^{hap}



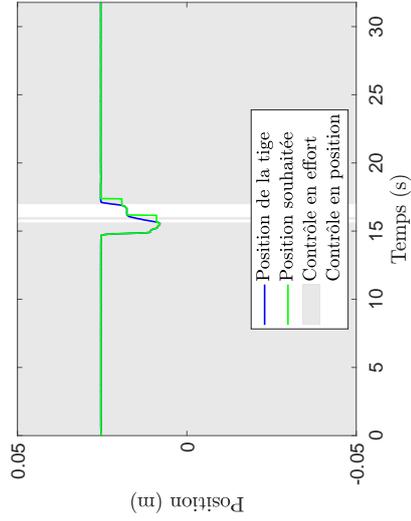
(e) F_{pneu} et F_d^{hap}



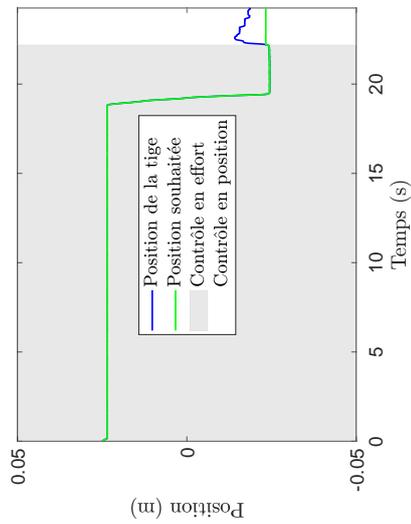
(f) F_{pneu} et F_d^{hap}



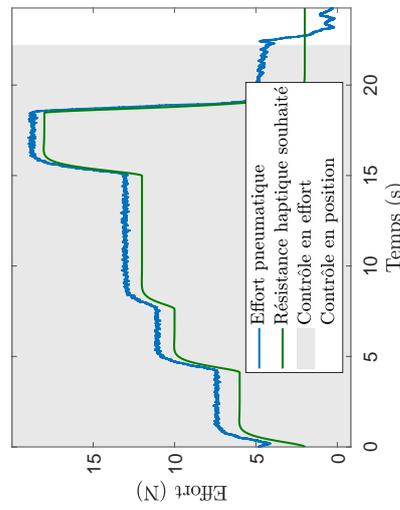
(a) y et y_d



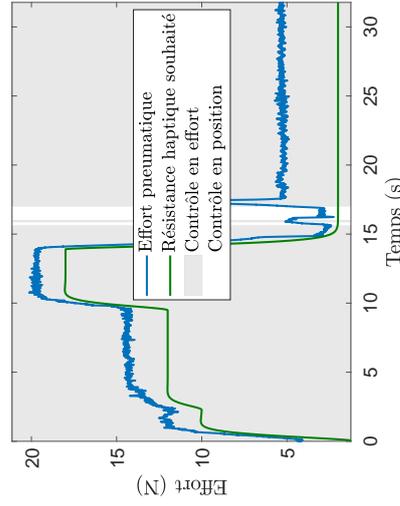
(b) y et y_d



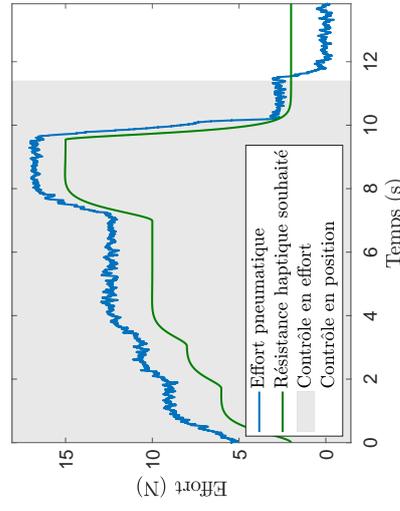
(c) y et y_d



(d) F_{pneu} et F_d^{hap}



(e) F_{pneu} et F_d^{hap}



(f) F_{pneu} et F_d^{hap}

Annexe C

Questionnaire soumis aux novices après l'utilisation du simulateur

Annexe D

Questionnaire soumis aux experts après l'utilisation du simulateur

QUESTIONNAIRE

Age: __ __

Genre : F / H

A. Jugez-vous que les sensations ressenties correspondent à la réalité ...:**1. Lors du franchissement par l'aiguille des différentes couches anatomiques :**

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7
Tout à fait d'accord	D'accord	Plutôt d'accord	Indifférent	Plutôt pas d'accord	Pas d'accord	Pas du tout d'accord

2. Lors du franchissement du ligament jaune :

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7
Tout à fait d'accord	D'accord	Plutôt d'accord	Indifférent	Plutôt pas d'accord	Pas d'accord	Pas du tout d'accord

3. Lors de la perte de résistance :

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7
Tout à fait d'accord	D'accord	Plutôt d'accord	Indifférent	Plutôt pas d'accord	Pas d'accord	Pas du tout d'accord

4. Lors de la manipulation / réorientation de l'aiguille :

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7
Tout à fait d'accord	D'accord	Plutôt d'accord	Indifférent	Plutôt pas d'accord	Pas d'accord	Pas du tout d'accord

5. Lors de la pression sur le piston de la seringue :

<input type="checkbox"/> 1	<input type="checkbox"/> 2	<input type="checkbox"/> 3	<input type="checkbox"/> 4	<input type="checkbox"/> 5	<input type="checkbox"/> 6	<input type="checkbox"/> 7
Tout à fait d'accord	D'accord	Plutôt d'accord	Indifférent	Plutôt pas d'accord	Pas d'accord	Pas du tout d'accord

