

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Elie Cattan

Thèse dirigée par **François Bérard, Pascal Perrier, Amélie Rochet-Capellan**

préparée au sein du **Laboratoire d'Informatique de Grenoble, équipe Ingénierie de l'Interaction Homme-Machine** et de l'**Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Analyse de la latence et de sa compensation pour l'interaction au toucher direct : aspects techniques et humains

Thèse soutenue publiquement le **26 septembre 2017**, devant le jury composé de :

Mme Jocelyne Troccaz

Directrice de recherche à l'Université Grenoble Alpes , Présidente

M. Géry Casiez

Professeur à l'Université Lille 1 , Rapporteur

M. Jean-Louis Vercher

Directeur de recherche à l'Université Aix-Marseille , Rapporteur

Mme Petra Isenberg

Chargée de recherche à l'Inria Saclay , Examinatrice

M. Matthieu Rupin

Ingénieur de recherche à HAP2U , Examineur

M. François Bérard

Maître de conférences à Grenoble INP , Directeur de thèse

M. Pascal Perrier

Professeur à Grenoble INP , Co-Directeur de thèse

Mme Amélie Rochet-Capellan

Chargée de recherche à l'Université Grenoble Alpes , Co-Directeur de thèse



Abstract

Latency, the delay between a user input on a system and the corresponding response from the system, is a major issue for the usability of interactive systems. In direct-touch interaction, latency is particularly perceivable and alters user performance even at levels in the order of ten milliseconds. Yet, current touch devices such as smartphones or tablet-pc exhibit in general latencies over 70 ms.

Our goal is to improve the knowledge on latency (its causes, its effects) and to find strategies to compensate it or to decrease its negative effects. We present a review of the HCI literature on the topic, then we link this literature with the motor control research field that has studied human behaviour when facing visuomotor perturbations, and in particular the adaptation to feedback delay.

We then present our four contributions. We contribute both in a practical and a theoretical manner to the problem of latency in direct-touch interaction. Two of our contributions supplement the diagnosis of latency : the first one is a new latency measurement technique ; the second one is a study of the impact of latency on bimanual interaction, which is important when interacting on large tactile surfaces. We show that bimanual interaction is as much affected by latency as a single hand interaction, suggesting that more complex tasks, suppose to increase the cognitive load, do not necessarily reduce the effect of latency. Our two other contributions address the reduction of the effects of latency. On one hand, we introduce a low latency system (25 ms) associated with a predictive software compensation, and we show that the system enables users to improve their performances as if they were using a system with 9 ms of latency. On the other hand we study users' ability to adapt to latency in order to improve their performance on a tracking task, and we show that the negative impact of latency is reduced with long-term training thanks to human adaptability.

Résumé

La latence, c'est-à-dire le délai entre l'action d'un utilisateur en entrée d'un système et la réponse correspondante fournie par le système, est un problème majeur pour l'utilisabilité des dispositifs interactifs. La latence est particulièrement perceptible pour l'interaction au toucher et détériore la performance de l'utilisateur même à des niveaux de l'ordre de la dizaine de millisecondes. Or, la latence des écrans tactiles actuels (smartphones ou tablettes) est en général supérieure à 70 ms.

Notre objectif est d'améliorer nos connaissances sur la latence (ses causes, ses effets) et de trouver des méthodes pour la compenser ou en diminuer les effets négatifs. Nous proposons un état de l'art des travaux en IHM sur le sujet, puis nous effectuons un rapprochement avec la littérature du contrôle moteur qui a aussi étudié le comportement humain face à des perturbations visuomotrices et en particulier l'adaptation des mouvements à un retard du retour visuel.

Nous détaillons ensuite nos quatre contributions. Nos résultats contribuent de manière à la fois pratique et théorique à la résolution du problème de la latence lors de l'interaction au toucher direct. Deux contributions complètent le diagnostic de la latence : la première est une nouvelle technique de mesure de latence ; la seconde est une étude de l'effet de la latence sur l'interaction bimanuelle, importante pour l'interaction sur les grandes surfaces tactiles. Nous montrons que l'interaction bimanuelle est autant touchée par la latence que l'interaction à une main, ce qui suggère que des tâches plus complexes, qui augmenterait la charge cognitive, ne réduisent pas nécessairement l'effet de la latence. Nos deux autres contributions portent sur la réduction des effets de la latence. D'une part, nous proposons un système à faible latence (25 ms) associé à une compensation prédictive logicielle, et nous mettons en évidence que ce système permet d'améliorer la performance des utilisateurs comme s'ils utilisaient un système à 9 ms de latence. D'autre part nous étudions la capacité des utilisateurs à s'adapter à la latence pour améliorer leur performance sur une tâche de suivi de cible et nous montrons que l'impact négatif de la latence se réduit sur le long terme grâce aux capacités d'adaptation humaine.

Table des matières

Remerciements	11
Introduction	13
I Etat de l'art	17
1 Définitions et contexte de la thèse	19
1.1 Latence d'un système interactif	19
1.1.1 Définition	19
1.1.2 Panorama sur la latence	21
1.2 La latence : un facteur limitant pour l'IHM	21
1.2.1 Temporalité, cause et conséquence	22
1.2.2 Boucle ouverte vs boucle fermée	22
1.3 La latence au toucher	23
1.3.1 Interaction directe vs. indirecte	23
1.3.2 Dispositifs tactiles et technologies	25
1.4 Bilan	27
2 Diagnostiquer	29
2.1 Mesurer la latence	30
2.1.1 Sources de latence	30
2.1.2 Mesure de latence avec caméra et annotation d'images	33
2.1.3 Automatisation de la mesure de la latence	38
2.1.4 Bilan	44
2.2 Perception et effets de la latence	45
2.2.1 La latence en réalité virtuelle	45
2.2.2 Perception de la latence au toucher	46
2.2.3 Impact qualitatif de la latence au toucher	48
2.2.4 Impact quantitatif de la latence	50
2.2.5 Bilan	54
3 Compenser les effets de la latence	57
3.1 Réduction et compensation de la latence : aspects techniques	58
3.1.1 Capteurs et Affichage	58
3.1.2 Réduction de la latence au niveau de l'application	61
3.1.3 Optimisation du pipeline	65
3.1.4 Bilan	67
3.2 Adaptation humaine face à un retard du retour visuel	69
3.2.1 Définitions et concepts	69

3.2.2	Compensation, adaptation et latence	75
3.2.3	Bilan	80
II	Contributions	83
4	Contributions à la compréhension du phénomène de latence	85
4.1	Un système faible latence	86
4.1.1	Démarche	86
4.1.2	Le système Optitrack	87
4.1.3	Recréer l'interaction tactile	89
4.1.4	Optimisation du système : synchronisation entre la capture et l'affichage	93
4.1.5	Latence finale	95
4.1.6	Applications du système faible latence	96
4.1.7	Bilan	97
4.2	Une technique de mesure de la latence au toucher légère et précise	99
4.2.1	Problématique	99
4.2.2	La technique de mesure	99
4.2.3	Evaluation de la technique par une étude utilisateur	101
4.2.4	Discussion	105
4.2.5	Bilan	106
4.3	Effets de la latence sur l'interaction bimanuelle	108
4.3.1	Introduction	108
4.3.2	Etude utilisateur : effets de la latence sur une tâche simple vs composite	109
4.3.3	Discussion	115
4.3.4	Bilan	116
5	Compensation de la latence	117
5.1	Compensation logicielle : utilisation d'une prédiction	118
5.1.1	La prédiction, une question de compromis	118
5.1.2	Choix d'un modèle de prédiction	119
5.1.3	Evaluation de la prédiction linéaire - Etude utilisateur	123
5.1.4	Discussion	129
5.1.5	Bilan	134
5.2	Compensation motrice humaine	135
5.2.1	Problématique	135
5.2.2	Etude utilisateur 1 : Adaptation à la latence, transfert, rétention	136
5.2.3	Etude utilisateur 2 : Interférence entre l'apprentissage des compétences de suivi et l'adaptation à la latence	148
5.2.4	Discussion : Contrôle moteur	150
5.2.5	Discussion : conception des systèmes tactiles	153
5.2.6	Bilan	154
	Conclusion	155

Table des figures

1.1	Représentation de la boucle perception-action entre l'utilisateur et la machine	20
1.2	Interaction au toucher, direct vs indirect	24
1.3	Dispositifs tactiles de différentes tailles	26
2.1	Découpage des différentes sources de latence pour le cas de l'interaction au toucher	30
2.2	Effet de la discrétisation de l'affichage sur la latence	31
2.3	Latence introduite par un lissage des données des capteurs	32
2.4	Technique du chronomètre pour mesurer la latence d'un écran LCD	33
2.5	Méthode de mesure de latence de He et al. [57]	35
2.6	Méthode de mesure de latence de Liang et al. [90]	35
2.7	Méthode de mesure de latence de Teather et al. [137]	36
2.8	Méthode de mesure de latence de Swindells et al. [136]	36
2.9	Méthode de mesure de latence de Ng et al. [106]	37
2.10	Pendule utilisé par Steed pour mesurer la latence [133]	38
2.11	Méthode de mesure de la latence au toucher "high accuracy" de Bérard et Blanch [19]	39
2.12	Circuit électronique utilisé par Beyer et al. pour mesurer la latence d'un smartphone [23]	40
2.13	Méthode de mesure de la latence au toucher "low overhead" par Bérard et Blanch [19]	41
2.14	Bilan de quatre approches pour mesurer la latence	43
2.15	Perception de la latence sur un écran tactile	46
2.16	Tableau résumé des seuils de perception de la latence pour différentes tâches et modalité d'utilisation d'un écran tactile	47
2.17	Modélisation d'un mouvement de pointage par la loi de Fitts	50
2.18	Différentes droites de Fitts pour différents niveaux de latence [62]	52
2.19	Tableau récapitulatif des études sur l'impact de la latence sur la performance de temps de pointage pour différents dispositifs et différentes tâches	55
3.1	FMT, un capteur tactile capacitif haute performance [88]	58
3.2	Système faible latence utilisé par Ng et al. [106]	59
3.3	Deux exemples où la latence est intégrée dans le design de l'interaction	61
3.4	Utilisation d'une prédiction pour compenser la latence du premier contact [155]	63
3.5	Utilisation d'une prédiction pour compenser la latence sur un système de réalité augmentée	64
3.6	<i>Tearing</i>	65
3.7	Une optimisation possible de la latence : récupérer les données des capteurs les plus récentes possibles	66

3.8	Deux paradigmes expérimentaux pour étudier les recalibrations visuomotrices	70
3.9	Modèle d'un protocole expérimental pour tester l'adaptation à une perturbation	71
3.10	Modèle d'un protocole expérimental pour tester les <i>savings</i>	73
3.11	Modèle d'un protocole expérimental pour tester le transfert d'apprentissage et les interférences	74
3.12	Résultats de Foulkes et Miall sur l'adaptation à la latence pour une tâche de suivi de cible avec un joystick [52]	78
3.13	Effets du retard sur la trajectoire du retour visuel similaires aux effets d'un décalage spatial ou d'inertie	79
4.1	Photo de notre système faible latence	87
4.2	Des marqueurs réfléchissants de différentes tailles	88
4.3	Une caméra Optitrack Flex 13	88
4.4	Les deux calibrations nécessaires pour simuler l'interaction tactile avec l'Optitrack	90
4.5	La tige fine équipée d'un marqueur qui sert à effectuer la calibration écran-caméras	90
4.6	Etape 2 de la calibration de notre système	91
4.7	Etape 3 de la calibration de notre système	92
4.8	Différents types de synchronisation entre la fréquence de rafraîchissement de l'écran et la fréquence d'arrivée d'une nouvelle image à afficher ainsi que la latence qui en résulte	94
4.9	Découpage de la latence sur notre système faible latence	96
4.10	Une application du système faible latence : l'interaction au stylet	97
4.11	Un utilisateur réalisant une mesure de latence avec notre méthode prédictive	100
4.12	Un utilisateur réalisant une mesure de latence avec la méthode LO [19] . . .	102
4.13	Ordre de présentation des différents facteurs pour l'étude sur la mesure de latence	103
4.14	Erreur moyenne de mesure pour chaque participant à 30 ms et 80 ms de latence avec la méthode LO ou avec notre méthode prédictive	104
4.15	Erreur de la mesure LO la plus stable comparée avec l'erreur de la première mesure prédictive	105
4.16	Moyenne de l'erreur de mesure inter-participants, dans la condition LO et la condition prédictive	106
4.17	Affichage au début de la tâche simple et de la tâche composite	109
4.18	Affichage lors de pauses	110
4.19	Ordre de présentation des différents facteurs pour l'étude sur l'interaction bimanuelle	112
4.20	<i>impact_temps</i> et <i>impact_erreur</i> de la latence pour les tâches simple et composite avec les intervalles de confiance à 95% de la moyenne	113
4.21	Table des valeurs mesurées des variables dépendantes selon le facteur TACHE113	
4.22	Effet de MAIN et LATENCE sur le temps de déplacement pendant la tâche composite	114
5.1	Utilisation d'une prédiction pour compenser la latence	119
5.2	Un exemple d'une trajectoire de pointage en une dimension	120
5.3	Les trajectoires générées par différents modèles de prédiction pour la dernière phase d'un mouvement de pointage	121

5.4	Trajectoire du doigt, trajectoire retardée de 75 ms et trajectoire prédite sur l'intégralité d'un mouvement de pointage avec le modèle de prédiction linéaire	122
5.5	Erreur moyenne entre l'objet manipulé et le doigt, avec l'utilisation de la prédiction ou non	123
5.6	Un utilisateur en train de réaliser la tâche de pointage	124
5.7	Ordre de présentation des différents facteurs pour l'étude sur la prédiction .	125
5.8	Table des valeurs mesurées des variables dépendantes selon les facteurs PREDICTION et LATENCE	126
5.9	Effet de la latence sur le temps d'acquisition de cible quand aucune prédiction n'est utilisée	127
5.10	Temps d'acquisition de cible avec ou sans prédiction pour les différents ID et les différents niveaux de latence	128
5.11	Erreur moyenne sur les participants et les ID pour les différentes valeurs de latence avec ou sans prédiction	129
5.12	Bande passante moyenne sur les participants pour les différentes valeurs de latence avec ou sans prédiction	130
5.13	Proportion de participants ayant préféré la condition "avec prédiction" pendant l'étude qualitative	131
5.14	Extrapolation des données sans prédiction pour évaluer la latence équivalente d'un système avec prédiction	132
5.15	L'affichage à trois moments de l'étude.	139
5.16	Trajectoires et vitesses de la cible pour les deux tâches de suivi étudiées . . .	140
5.17	Diagramme quantile-quantile évaluant la pertinence du modèle linéaire mixte sélectionné pour modéliser l'erreur lors de l'entraînement sur la tâche de suivi de cible sur l'ellipse	143
5.18	Moyenne inter-participants de l'erreur de suivi pour chaque essai et session pour le groupe contrôle et le groupe test	144
5.19	Moyenne inter-participants de l'erreur de suivi sur l'ellipse de la première à la dixième session pour les deux groupes	145
5.20	Performances de suivi de cible sur la trajectoire de transfert	146
5.21	Moyenne inter-participants de l'erreur de suivi sur l'ellipse lors de la seconde étude pour le groupe contrôle et le groupe test	149

Remerciements

Je remercie les membres de mon jury d'avoir participé à la validation de ces trois ans de travail, et en particulier les deux rapporteurs pour leurs apports sur le manuscrit.

Je remercie François, Amélie et Pascal pour leur présence, leur implication et la motivation qu'ils m'ont apportée.

Je remercie mes collègues de l'équipe IIHM et de l'équipe PCMD et en particulier mes co-bureau sans qui ces trois ans de thèse auraient été bien plus mornes.

Et finalement, je remercie mon seul soutien indéfectible durant cette thèse : ma chaise de bureau.

Introduction

“Oh dear! Oh dear! I shall be too late!”

- Lewis Carroll, Alice au pays des merveilles

Motivation

Aujourd’hui, les interfaces homme-machine ne sont plus limitées au simple paradigme écran/clavier/souris mais se développent vers des interactions dites “naturelles”. Ces interactions sont conçues dans le but de mettre en œuvre des mécanismes toujours plus proches de ceux que nous exploitons lorsque nous interagissons dans le monde physique [151]. Par exemple, l’interaction tactile et la réalité augmentée ou virtuelle nous permettent de déplacer des objets virtuels ou de naviguer dans des environnements numériques en mimant les interactions du monde physique. Ce type d’interaction est de plus en plus présent dans notre vie quotidienne comme le montre le succès croissant des smartphones et des tablettes tactiles ainsi que l’essor des technologies de réalité virtuelle.

Cependant, les traitements informatiques qui gèrent ces environnements numériques nécessitent du temps. Depuis les capteurs du matériel, jusqu’au rendu à l’écran, en passant par l’application, il existe de multiples étapes qui cumulent du temps de calcul et qui forment ce qu’on appelle la latence du système, c’est-à-dire le délai entre le moment où un opérateur effectue une action sur le système, et le moment où le système fournit le retour sensoriel correspondant à cette action (retour visuel en général). Pour les actions comme “déplacer un objet” ou “explorer son environnement”, ce retard est inexistant dans le monde physique. La latence dégrade donc la similitude avec le monde physique : elle casse la sensation de naturel même à un niveau faible, ce qui a un impact négatif sur l’expérience et la performance de l’utilisateur.

La latence est un facteur particulièrement limitant pour l’interaction tactile directe. Lorsqu’un objet est déplacé sur un écran tactile, à cause de la latence, il traîne derrière le doigt au lieu d’être juste dessous. Cette transformation d’un retard temporel en un écart spatial entre le doigt et l’objet rend la latence très facilement perceptible. Celle-ci a alors un impact négatif sur les utilisateurs à des niveaux de l’ordre de la dizaine de millisecondes seulement [69, 106], ce qui est un seuil bien plus bas que les niveaux de latence autour de

75 ms des dispositifs actuellement disponibles dans le commerce.

Dans cette thèse, notre objectif est donc d’analyser la latence dans le but de la réduire ou d’en minimiser les effets afin de rendre l’interaction plus “naturelle”. Nous nous intéressons plus spécifiquement au contexte des écrans tactiles et de l’interaction au toucher.

Démarche

Nous examinons le phénomène de latence en deux étapes. Dans une première étape, nous effectuons un diagnostic de la latence, c’est-à-dire que nous examinons ses causes, les méthodes pour la mesurer, ainsi que les conséquences qu’elle a sur les utilisateurs. Dans une seconde étape, une fois le phénomène caractérisé, nous analysons différentes stratégies pour en compenser les effets. Pour chacune de ces deux étapes, nous analysons d’abord les travaux de la littérature puis nous proposons des contributions qui viennent compléter ces travaux ou qui offrent de nouvelles perspectives sur le diagnostic ou la compensation de la latence.

Cette thèse aborde chacune des questions de diagnostic et de compensation de la perturbation selon deux points de vue. Le premier est centré sur la machine avec des contributions matérielles et logicielles. Le diagnostic consiste alors à comprendre les sources de la latence et comment celle-ci peut être mesurée et optimisée. Pour la compensation, l’aspect machine intervient sur des améliorations logicielles qui permettent d’atténuer les effets de la latence. Le second point de vue est centré sur l’utilisateur. D’une part nous diagnostiquons les effets négatifs de la latence sur les utilisateurs. D’autre part, nous étudions la compensation de la latence par les utilisateurs en nous appuyant sur les théories et les paradigmes expérimentaux du domaine du contrôle moteur et en particulier de l’apprentissage sensorimoteur.

Structure du document

Dans ce document, nous proposons tout d’abord un état de l’art sur la latence pour l’interaction au toucher. Celui-ci nous permet de définir le contexte de la thèse et ses contours (Chapitre I) puis d’effectuer un diagnostic de la latence au niveau de la machine (causes et mesure) et de ses effets sur les utilisateurs (Chapitre II). Enfin, nous exposons différentes manières par lesquelles les effets de la latence peuvent être atténués, par améliorations matérielles ou logicielles ou par adaptation motrice humaine (Chapitre III).

Dans une seconde partie, nous détaillons nos contributions. Nos trois premiers apports permettent une meilleure compréhension des causes et effets de la latence : la création d’un système optimisé pour obtenir une interaction au toucher à faible latence, une nouvelle technique pour mesurer la latence et l’analyse des effets de la latence sur l’interaction bimanuelle (Chapitre IV). Nos deux autres contributions portent sur la compensation

INTRODUCTION

de la latence : l'étude d'une possible compensation logicielle grâce à une prédiction du mouvement de l'utilisateur et l'analyse de l'adaptation motrice humaine face à la latence (Chapitre V). L'ensemble de ces contributions nous offre une meilleure compréhension de la perturbation induite par le phénomène de latence et fournit des moyens pour en atténuer les effets. Enfin, nous proposons une conclusion ainsi qu'une réflexion sur les perspectives ouvertes par notre travail.

Première partie

Etat de l'art

Chapitre 1

Définitions et contexte de la thèse

Ce premier chapitre donne un aperçu global de ce qu'est la latence et permet d'introduire la problématique de la thèse et de définir son contexte. Il répond aux questions suivantes :

- Qu'est ce que la latence ?
- Pourquoi la latence est une limite pour l'interaction Homme-machine ?
- Pourquoi étudier la latence dans le contexte de l'interaction tactile ?

1.1 Latence d'un système interactif

1.1.1 Définition

Les systèmes interactifs permettent la communication entre la machine et ses utilisateurs. Grâce à différents types d'interfaces, l'utilisateur peut effectuer des actions en entrée du système qui peut alors lui répondre en renvoyant un retour sensoriel, souvent visuel, via un affichage sur un écran. L'utilisateur perçoit alors ce retour, l'analyse, et effectue l'action suivante (Figure 1.1). Le traitement des actions de l'utilisateur prend du temps et la réponse de la machine n'est pas instantanée. Ce délai entre l'action de l'utilisateur et la réponse fournie par la machine est la latence du système interactif.

La latence dite bout-à-bout d'un système interactif se définit comme **le temps entre l'action d'un utilisateur sur le système et le moment où le système fournit un retour sensoriel correspondant à cette action.**

Le terme *latence* est parfois utilisé pour désigner une sous-partie de la latence bout-à-bout. Par exemple, la latence d'une application sera le temps entre le moment où l'application reçoit un nouvel événement et le moment où elle envoie l'image à afficher à l'écran. Dans cette thèse, nous nous focalisons sur la latence bout-à-bout, c'est-à-dire la latence globale du système car c'est celle qui est perçue par l'utilisateur et dont nous souhaitons analyser les effets sur le comportement des utilisateurs. Sauf précision contraire, dans la

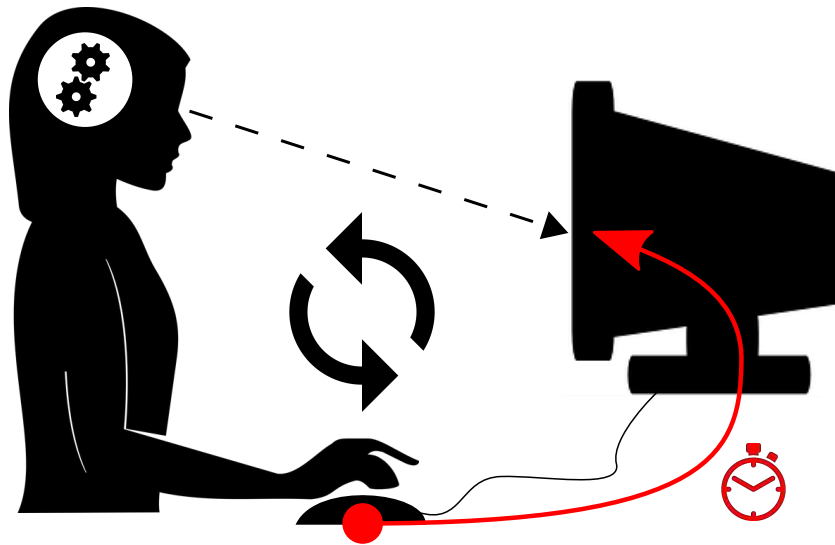


FIGURE 1.1 – Représentation de la boucle perception-action entre l'utilisateur et la machine. L'exemple de cette figure est un cas d'utilisation avec une souris et un écran. La latence (courbe rouge) est le temps qui s'écoule entre l'action de l'utilisateur sur le système (ici le déplacement de la souris) et le retour fourni par le système (ici l'affichage à l'écran).

suite du document, tout emploi du mot latence fera référence à la latence bout-à-bout du système.

Les interfaces dont l'utilisateur se sert pour interagir avec le système peuvent être de différentes natures. Elles peuvent prendre la forme d'un périphérique externe comme un clavier ou une souris, mais peuvent aussi permettre une interaction plus directe comme tourner la tête lorsqu'on est plongé dans un environnement virtuel via un casque de réalité virtuelle. Dans cette thèse nous étudions l'interaction au toucher qui consiste à poser ou déplacer son doigt sur une surface tactile. L'interaction au toucher est de plus en plus fréquente dans notre vie quotidienne à travers, par exemple, les smartphones ou les tablettes, et nous verrons qu'elle est particulièrement sensible à la latence.

Le retour du système peut faire appel aux différents sens de l'utilisateur en étant visuel, audio voire haptique. Par exemple, lorsqu'on tape au clavier sur un smartphone, les trois types de retour peuvent être disponibles. La lettre qu'on vient de toucher peut grandir (retour visuel), on peut entendre un "clic" (retour audio) et on peut sentir une vibration (retour haptique). Kaaresoja et al. (2014) ont montré que la sensibilité à la latence est différente selon le type de retour du système [71]. Les modalités audio et haptique sont moins fréquemment utilisées pour l'interaction au toucher et souvent seulement pour renforcer le retour visuel [28, 29]. Dans cette thèse, nous nous focalisons donc sur le retour visuel, c'est-à-dire l'affichage sur l'écran tactile.

1.1.2 Panorama sur la latence

Un des plus anciens domaines de l'IHM où ont été traités les problèmes introduits par la latence est celui de la téléopération [128]. Les systèmes de téléopération peuvent parfois atteindre des latences de quelques centaines de millisecondes, voire de l'ordre de plusieurs secondes, qui rendent leur contrôle difficile.

Un autre domaine où le problème de la latence a été étudié est celui de la communication réseau où la latence est un enjeu particulier pour les jeux en lignes [38, 27]. Le serveur central qui gère le jeu doit traiter différentes latences provenant des différents systèmes utilisés par les joueurs. Une mauvaise harmonisation de ces latences peut conduire à des différences de latence entre joueurs de l'ordre de la demi-seconde [27]. Cela peut entraîner des incohérences dans le jeu ou des biais en faveur de certains joueurs qui possèdent alors des informations mises à jour plus récemment.

Dans le domaine de la réalité virtuelle, la latence a rapidement été considérée comme un des facteurs limitant de l'immersion, même à des niveaux très faibles. Jerald (2010) suggère que la latence des casques de réalité virtuelle ne devrait pas dépasser 3 ms pour que l'utilisateur n'en ressente plus les effets [68]. D'autres études ont mesuré la perception de la latence et l'influence de la latence sur la performance lorsque l'utilisateur est plongé dans un environnement virtuel. Ces études sont détaillées au chapitre 2.

Le domaine du contrôle moteur s'est servi de la latence comme d'un outil pour la recherche sur les mécanismes cognitifs du contrôle du mouvement. Introduire un retard entre le mouvement de l'utilisateur et la réponse du système permet d'analyser les réactions des utilisateurs lorsqu'ils sont soumis à une contradiction entre leur proprioception et leur vision. Ce paradigme expérimental a été exploité dans plusieurs études qui sont détaillées au chapitre 3.

Les travaux sur la latence des écrans tactiles ne sont apparus que plus récemment [69, 106] et constituent un nouveau champ de recherche sur lequel nous nous focalisons dans cette thèse.

1.2 La latence : un facteur limitant pour l'IHM

La latence influe sur la temporalité entre une action effectuée par l'utilisateur et le retour fourni par le système. Au-delà de certaines valeurs, il devient difficile d'associer les retours du système aux actions effectuées et l'utilisateur doit alors opter pour des stratégies qui limitent l'utilisabilité du système.

1.2.1 Temporalité, cause et conséquence

Le cerveau gère la relation entre action et conséquence de cette action de manière asymétrique [121]. Tout retour d'un système arrivant avant même que l'action soit effectuée ne pourra être perçu comme une conséquence de cette action car les effets sont censés temporellement suivre leur cause et non l'inverse. Par contre, un retour visuel arrivant jusqu'à 500 ms après l'action pourra être perçu comme une conséquence de l'action [121]. Si la latence d'un système interactif fournissant un retour visuel est inférieure à 500 ms, elle ne devrait pas perturber la sensation pour l'utilisateur que ses actions sont bien la cause de ce qu'il voit à l'écran.

D'un point de vue qualitatif et subjectif, un retard entre une action et sa conséquence peut parfois être considéré comme une expérience positive par les utilisateurs. En effet, il peut permettre de mettre l'emphasis sur le moment de l'interaction ou de créer une prise de conscience de la technologie sous-jacente [89]. D'un autre côté, la latence peut créer un manque de confiance envers le système et être "irritante" [89].

La plupart du temps, l'utilisateur ne réalise pas une seule action sur le système, mais une série d'actions. Une action peut dépendre du résultat de la précédente et, dans ce cas, l'utilisateur doit attendre le retour du système pour agir. Si la latence est trop élevée, alors l'utilisateur est obligé d'adopter une stratégie d'attente entre chaque action comme l'ont montré les deux études décrites dans le paragraphe suivant.

1.2.2 Boucle ouverte vs boucle fermée

En 1963, Sheridan et Ferrell ont proposé une étude de l'influence de la latence sur la performance dans une tâche de positionnement pour un cadre de téléopération [128]. Le participant de l'étude doit déplacer un manipulateur maître de manière à positionner un manipulateur esclave à un endroit donné. Le manipulateur esclave suit les mouvements du maître, mais avec un retard pouvant aller jusqu'à 3.2 s. La latence affecte le temps de réalisation de la tâche, mais surtout, l'expérience montre qu'à des valeurs de retard de 1 s ou supérieures, les mouvements s'effectuent en boucle ouverte : à chacune de ses actions, le participant attend que le manipulateur esclave ait atteint sa position, sans tenter de corriger la trajectoire en cours. Lorsque la tâche est effectuée sans retard, par contre, les mouvements sont réalisés en boucle fermée, c'est-à-dire que l'utilisateur réagit en parallèle du retour fourni par le système.

L'étude de Sheridan et Ferrell n'incluant aucun niveau de latence entre 0 s et 1 s, Hill (1976) a mesuré avec plus de précision la transition entre réponse en boucle fermée ou en boucle ouverte et a mené une étude similaire mais en ajoutant un niveau de latence à 0.3 s [61]. Hill définit un seuil sur la vitesse du manipulateur qui lui permet de découper chaque trajectoire en plusieurs mouvements. Lorsque la vitesse tombe sous un certain seuil pendant 0.4 s, on considère qu'un mouvement est terminé. Un nouveau mouvement

démarre lorsque la vitesse repasse au-dessus du seuil pendant 0.4 s. Le nombre moyen de mouvements pour exécuter la tâche n'augmente pas entre 1 s ou 3 s de latence, ce qui démontre une stratégie en boucle ouverte de type "bouge et attend" similaire dans les deux conditions. A 300 ms, le nombre de mouvements est supérieur à celui de la condition sans retard, mais inférieur à celui de la condition à 1 s. En effet, lorsque le contrôle passe en boucle fermée, l'utilisateur corrige sa trajectoire au fur et à mesure sans passer sous le seuil de vitesse et effectue donc un nombre de mouvements réduit. Un nombre de mouvements intermédiaire à 300 ms comparé à 0 ms et 1 s indique que la stratégie de contrôle change de manière continue d'un contrôle en boucle fermée, quand il n'y a pas de latence, vers un contrôle en boucle ouverte quand la latence augmente.

Pour des latences supérieures à 1 s, les utilisateurs doivent donc fonctionner en boucle ouverte. Lorsque la latence devient inférieure à 1 s, on observe une transition progressive vers un contrôle en boucle fermée. Dans le cadre de cette thèse, les niveaux de latence considérés seront inférieurs à 100 ms, suffisamment faibles pour que les utilisateurs contrôlent leurs mouvements en boucle fermée. Le contrôle en boucle fermée n'exclut pas pour autant l'influence de la latence sur la performance des utilisateurs, comme nous le verrons au chapitre 2.

1.3 La latence au toucher

Comme décrit pour les deux études dans le paragraphe précédent, la latence a été d'abord considérée comme un facteur limitant dans certaines tâches où sa valeur était de l'ordre de plusieurs centaines de millisecondes. Pour certains types d'interaction, comme le toucher ou la réalité virtuelle, la sensibilité à la latence est beaucoup plus forte et même de très faibles retards peuvent altérer l'interaction [106]. Le développement commercial récent et à grande échelle, de dispositifs tactiles et de réalité virtuelle a ouvert un nouveau champ de recherche sur des niveaux de latence plus faibles, de l'ordre de quelques dizaines de millisecondes.

Dans cette section, nous considérons les différences entre l'interaction directe et l'interaction indirecte qui expliquent la sensibilité forte des utilisateurs à la latence sur les écrans tactiles. Nous proposons ensuite une vue d'ensemble des dispositifs tactiles actuels et de la façon dont la latence peut être un facteur limitant lorsqu'on interagit avec ses dispositifs.

1.3.1 Interaction directe vs. indirecte

L'interaction est **directe** lorsque l'espace moteur et l'espace d'affichage sont confondus. Cet espace peut être 2D, par exemple dans le cas de l'écran tactile d'un smartphone ou d'une tablette, mais aussi 3D, par exemple pour la réalité augmentée. Lorsque l'espace

d'interaction et l'affichage sont séparés, l'interaction est **indirecte**, comme lorsqu'on utilise une souris pour déplacer un curseur sur un écran. L'interaction tactile peut se faire, selon les dispositifs, directement sur l'écran, ou indirectement sur un trackpad.

Un des buts de l'Interaction Homme Machine est la conception d'interactions *naturelles*. Au sens de Bérard et Rochet-Capellan [22], l'interaction est "naturelle" si elle exploite les compétences déjà acquises dans le monde physique. L'interaction directe, à ce titre, en confondant l'espace d'interaction et l'espace d'affichage, mime le monde physique et répond à ce critère. Bérard et Rochet-Capellan (2015) ont montré que l'apprentissage se transfère mieux vers l'interaction physique lorsque les participants ont été entraînés en interaction au toucher directe plutôt qu'indirecte via un trackpad. Ces résultats confirment la similarité sensori-motrice entre l'interaction au toucher directe et l'interaction physique et en objectivent le caractère "naturel".

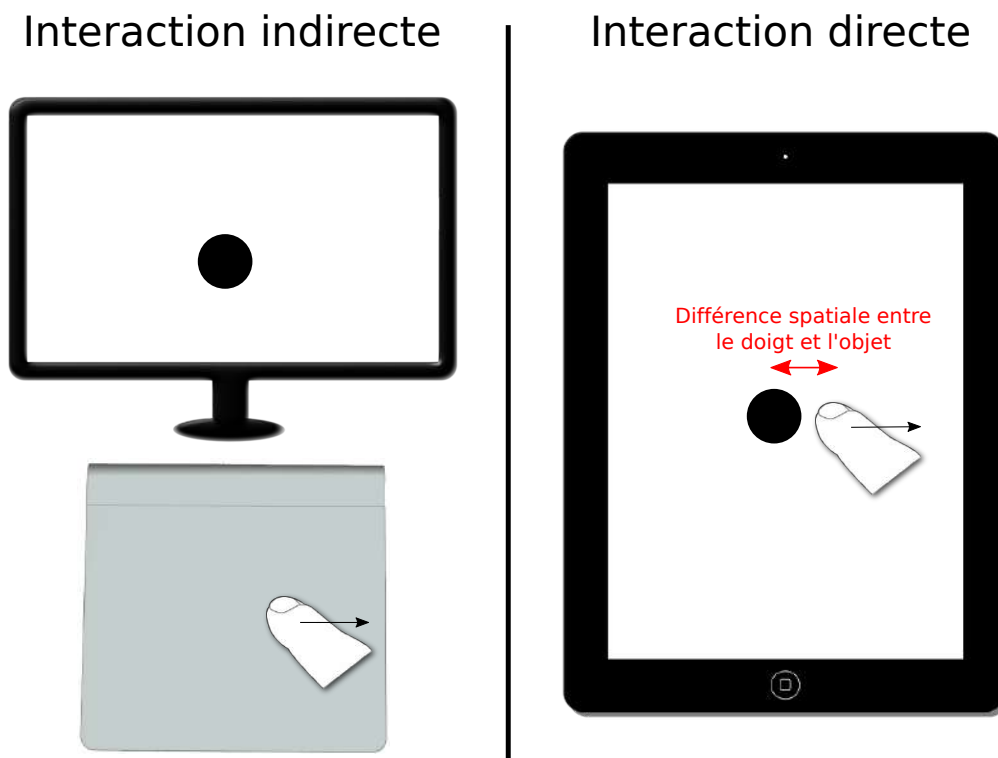


FIGURE 1.2 – Interaction au toucher, direct vs indirect. Lorsque l'interaction est directe, le retard se traduit par un écart spatial entre le doigt et le retour visuel fourni par l'écran. La latence est donc plus facilement perceptible lorsque l'interaction est directe.

En interaction indirecte, Bérard se base sur une estimation générale à 100 ms du temps de cycle humain de perception, cognition, et action ; il en déduit que les systèmes qui dépassent 50 ms de latence dégradent les performances des utilisateurs [20]. En 2015, Deber mesure le seuil de perception de la latence par les utilisateurs est de 55 ms pour le toucher indirect [46]. Les valeurs de latence actuelles de dispositifs d'interaction indirects tels que le trackpad ou la souris sont autour de 60 ms [32, 54] ce qui est assez proche des

seuils de perception. La latence n'a donc aujourd'hui qu'une influence minimale sur ce type de dispositifs.

L'interaction directe imite le monde physique et est particulièrement sensible aux déviations par rapport aux propriétés du monde physique qui peuvent dégrader la sensation de *naturel*, en particulier à la latence. En effet, lorsque l'interaction est directe, même un très faible retard temporel du retour visuel se transforme en une différence spatiale facilement perceptible (Figure 1.2). Ng et al. (2012) puis Deber et al. (2015) ont étudié les seuils de perception de la latence et ont montré qu'en déplaçant un objet au toucher direct, les utilisateurs peuvent percevoir un retard aussi faible qu'une dizaine de millisecondes [46, 106]. La latence des écrans tactiles actuels, autour de 75 ms [45], est donc bien plus haute que ce seuil de perception, et reste un facteur particulièrement important à considérer pour obtenir une interaction de qualité sur écran tactile.

Un moyen pour minimiser la latence des écrans tactiles est de proposer des technologies tactiles performantes. La variété des technologies tactiles va de pair avec une variété de type d'écrans avec lesquels les gens interagissent différemment. Il est donc aussi intéressant de comprendre comment la présence de la latence peut interférer avec l'interaction sur ces différents dispositifs.

1.3.2 Dispositifs tactiles et technologies

Technologies d'écrans tactiles

Le but de cette section est de comprendre la relation entre différentes technologies tactiles et la latence qui en découle. Pour une revue plus technique de l'ensemble des technologies, on pourra se reporter aux travaux de Walker [147].

La technologie d'écrans tactiles la plus utilisée aujourd'hui est la technologie capacitive [147]. Une matrice d'électrodes d'un matériau conducteur transparent est placée devant l'écran. Lorsqu'un doigt s'approche des électrodes, il en modifie la capacité. Des capteurs placés en bout de chaque ligne et colonne de la matrice permettent de détecter la position des électrodes dont la capacité a changé, et donc la position du/des doigt(s) posé(s) sur l'écran. Cette technologie est plus coûteuse que d'autres technologies tactiles, mais la matrice d'électrode ne détériore pas l'affichage contrairement à d'autres technologies.

Au début des années 2000, deux systèmes capacitifs multi-touch ont été proposés : SmartSkin [118] et DiamondTouch [47]. SmartSkin fournit des positions de toucher à 30 Hz, ce qui induit en moyenne 17 ms de latence (cf. "Information numérique discrète" au chapitre 2). DiamondTouch fonctionne entre 20 et 40 Hz, soit avec une latence entre 12.5 ms et 25 ms. Récemment, Leigh et al. (2014) ont proposé une optimisation de la technologie capacitive où les lignes et colonnes de la matrice d'électrodes sont scannées simultanément et non plus séquentiellement [88]. Cela permet d'obtenir un retard de seulement 40 μ s entre le moment du toucher et le moment où la donnée de position est disponible.

La technologie résistive est également largement employée. Elle consiste en deux couches dans lesquelles circule un courant électrique induit. Ces deux couches sont séparées par un faible espace mais rentrent en contact lorsqu'un objet vient exercer une pression sur l'écran. Le champ électrique dans les deux faces permet alors de déduire le point de contact. Ce type d'écran est peu cher et peut être activé par n'importe quel objet (conducteur ou non) exerçant une pression sur l'écran. Il y a dix ans, c'était la technologie majoritaire sur le marché [147] mais elle s'est rapidement faite supplanter par la technologie capacitive. En effet, la technologie résistive implique une dégradation importante de l'affichage à cause des couches placées devant l'écran qui laissent peu passer la lumière. Ce type d'écran est aussi moins sensible au contact du doigt à l'écran puisqu'une certaine pression est nécessaire pour l'activer.

Il existe aussi un certain nombre de technologies optiques. La technologie FTIR, par exemple, utilise des LEDs et une caméra infrarouge. Lorsque le doigt vient se poser sur l'écran, il change le coefficient de réfraction et laisse échapper la lumière infrarouge qui était capturée dans la surface de l'écran. La lumière est immédiatement réfléchiée par le doigt et captée par la caméra, permettant la détection du point de contact. La méthode FTIR possède plusieurs défauts dont l'utilisation d'une caméra qui implique un volume important, et le contrôle de l'environnement lumineux car la technologie est sensible aux sources de lumière.

Dispositifs et interaction

De nos jours, les dispositifs tactiles ont des facteurs de formes différents (Figure 1.3). Il existe des dispositifs de petites tailles comme les smartwatches, des dispositifs de taille moyenne comme les smartphones ou les tablettes tactiles, et enfin, des dispositifs de grande taille comme les tables tactiles.



FIGURE 1.3 – Dispositifs tactiles de différentes tailles. De gauche à droite : une smartwatch, un smartphone, une tablette et une table tactile

L'interaction sur ces différents dispositifs dépend de leur taille. En effet, l'interaction sur des dispositifs avec des écrans de petites tailles comme les smartwatches ou les smartphones se fait principalement avec des gestes de contacts discrets. Au contraire, les écrans de grandes tailles vont être plus propices aux actions continues comme le déplacement d'objets présents à l'écran ou l'écriture avec un stylet. La sensibilité des utilisateurs à la

latence étant plus forte pour les interactions continues [46], les grands écrans tactiles sont donc plus fortement touchés par la présence de latence. De plus, sur un écran de grande taille, les utilisateurs vont effectuer des mouvements plus amples et plus rapides, ce qui accroît l'écart spatial entre le doigt et le retour visuel et rend donc les effets de la latence plus perceptibles.

Les grandes surfaces tactiles sont aujourd'hui principalement utilisées pour un usage professionnel particulier (par exemple la tablette graphique pour un dessinateur) ou dans des lieux publics comme les galeries commerciales ou les aéroports, mais se sont assez peu développées par rapport aux smartphones et tablettes qui sont maintenant des objets du quotidien. Alors qu'il y a plus de 20 ans, Wellner envisageait le tactile comme un moyen d'interagir dans un monde augmenté sur une large surface [149], il est intéressant de noter que les grandes tables tactiles et les bureaux numériques n'ont pas connu le même essor que les dispositifs mobiles. On peut penser que la latence, souvent relativement élevée sur ces grandes surfaces tactiles, est un des facteurs qui en a freiné le développement.

Latences des dispositifs tactiles actuels

Il est difficile de connaître de manière exacte la valeur de la latence bout-à-bout des dispositifs actuellement dans le commerce. La principale raison est qu'il est compliqué de donner une valeur fixe de latence étant donné que la valeur de base pourra varier en fonction de plusieurs facteurs dont l'application en cours, ce qui tourne en tâche de fond, le niveau de batterie du dispositif etc... De plus, peu de constructeurs fournissent des données claires sur les différentes sources de latence de leurs dispositifs. La littérature scientifique qui propose des techniques de mesure de latence donne des valeurs pour certains dispositifs commerciaux actuels en essayant de contrôler un maximum de facteurs (batterie pleine et/ou pour une application donnée). Dans les résultats les plus récents, Deber et al. (2016) ont reporté des valeurs entre 52 ms (iPhone 6) et 92 ms (Nexus 6) [45] et Beyer et al. (2015) des valeurs entre 40 ms et 120 ms pour treize smartphones et tablettes [23].

La plupart des écrans tactiles atteignent donc des valeurs de latence inférieures à 150 ms, des valeurs suffisamment basses pour que le contrôle s'exécute en boucle fermée. Les valeurs de latence que nous considérerons dans ce mémoire seront donc de l'ordre de dizaines de millisecondes, avec une valeur typique, représentant le niveau de latence des dispositifs tactiles actuels, autour de 75 ms.

1.4 Bilan

En résumé, le contexte de la thèse est celui de l'interaction au toucher directe sur écran tactile. Ce type d'interaction est particulièrement sensible à la latence, celle-ci étant perceptible même autour de 10 ms, largement en deçà des valeurs des dispositifs actuels qui se situent en moyenne autour de 75 ms.

Bien que la thèse s’articule principalement autour de l’interaction tactile directe, les contributions que nous proposons s’inspirent aussi de travaux précédents sur la latence dans des domaines connexes comme la réalité virtuelle ou l’interaction à la souris. La bibliographie des deux chapitres suivants englobe donc un contexte plus général, mais dont les résultats servent à la compréhension du problème particulier de la latence au toucher.

Chapitre 2

Diagnostiquer

Pour pouvoir réduire la latence ou ses effets, il est important d'en comprendre les origines et d'en mesurer les effets. Diagnostiquer la latence nécessite d'abord une analyse au niveau de la machine : il faut savoir mesurer ce retard, ce qui inclut l'analyse des différentes sources de latence. Ensuite, il faut s'intéresser à l'aspect humain afin d'évaluer les effets que peut avoir la latence sur les utilisateurs autant d'un point de vue qualitatif (effet sur la frustration ou la présence) que quantitatif (performance sur une tâche).

Ce second chapitre procède donc à une revue des travaux existants sur la latence dans le but de répondre à ces questions :

- Quelles sont les sources de la latence ?
- Quelles techniques la littérature propose-t-elle pour mesurer la latence ?
- Quel est l'impact de la latence sur l'utilisateur ?

2.1 Mesurer la latence

Mesurer la latence bout-à-bout d'un système nécessite d'abord d'identifier ses différentes sources. La première partie de cette section présente donc une découpe du pipeline qui décrit les différentes causes de latence entre l'action d'un utilisateur et l'affichage à l'écran d'un retour visuel. Ensuite, nous présentons un ensemble de méthodes de mesure de latence qui consiste à annoter des images du système pendant son utilisation. Ce type d'annotation pouvant s'avérer fastidieux, nous explorons alors les autres méthodes de la littérature proposant des solutions plus automatisées pour mesurer la latence.

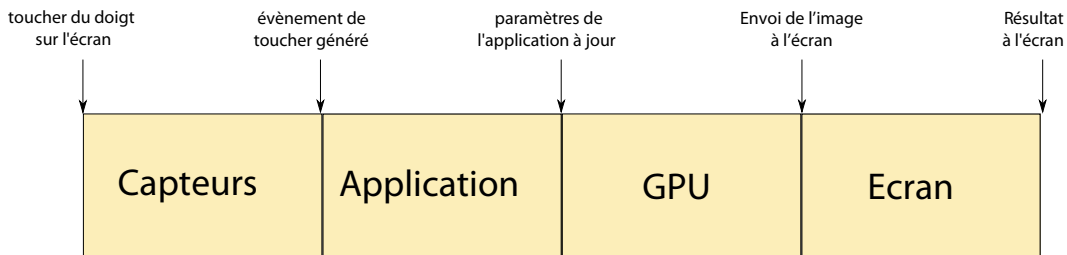


FIGURE 2.1 – Découpage des différentes sources de latence pour le cas de l'interaction au toucher.

2.1.1 Sources de latence

La latence bout-à-bout provient de plusieurs sources de retards qui se cumulent. En s'inspirant des travaux de Mine [98], Kamarainen et al. [74] et Tsoi et Xiao [1, 140], nous listons ci-dessous différentes sources potentielles de latence dans le pipeline entre l'entrée de l'utilisateur et un retour visuel affiché à l'écran. La description de ces sources est faite de manière générale, pouvant aussi bien s'appliquer, par exemple, aux écrans tactiles qu'à un système de réalité augmentée.

Information numérique discrète

La latence se définit comme le retard entre une action de l'utilisateur et le retour fourni par le système correspondant à cette action. Or, les actions des utilisateurs dans le monde physique sont continues alors que le système ne fonctionne que de manière discrète. La majorité des actions de l'utilisateur ne génèrent donc pas de retour du système et ce phénomène introduit de la latence. Par exemple, si l'utilisateur déplace son doigt sur un écran tactile dont le rafraichissement fonctionne à 60 Hz, alors la position ne sera pas mise à jour pendant toute la période d'un affichage, c'est-à-dire 16.7 ms. Au début de la période, lorsque l'écran vient d'afficher une nouvelle image, la latence est minimale. Mais le doigt continue à se déplacer et tant qu'un nouveau rafraichissement n'a pas lieu, l'image affichée devient de plus en plus caduque. A la fin de la période, juste avant le prochain rafraichissement, la latence est maximale, la dernière image affichée par le système est celle

datant de 16.7 ms. En intégrant ce retard sur la durée de la période, cela introduit une latence moyenne de $16.7/2=8.3$ ms (Figure 2.2).

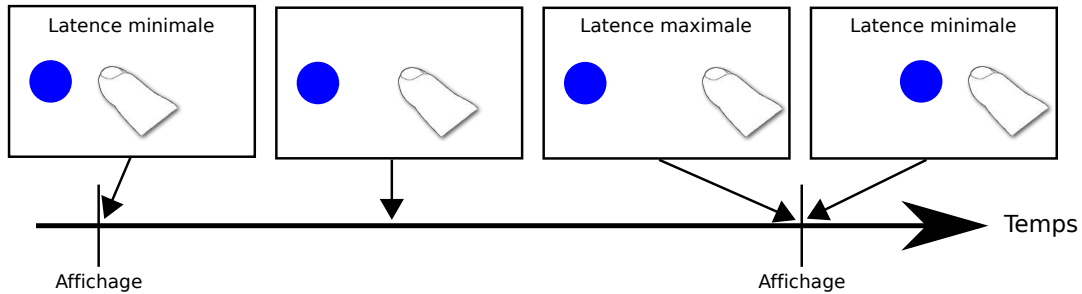


FIGURE 2.2 – Effet de la discrétisation de l’affichage sur la latence. Juste après l’affichage la latence est minimale. Pendant le cycle d’affichage, le doigt continue à se déplacer, mais le retour visuel n’est pas mis à jour jusqu’à l’affichage suivant ce qui génère une latence moyenne d’une demie période.

La latence introduite par ce phénomène est donc, en moyenne, la moitié de la période de l’écran. Mais ce phénomène se généralise à tous les processus du système, par exemple à la fréquence du capteur tactile. Dans le cas où ces processus sont synchronisés, c’est l’élément fonctionnant à la plus faible fréquence qui dicte la latence induite. En pratique, c’est souvent l’écran.

Il est donc important d’avoir des fréquences de fonctionnement élevés pour les différents processus du système afin de limiter la latence induite. Certains capteurs tels que les gyroscopes ou accéléromètres de casques de réalité virtuelle peuvent fonctionner à 1000 Hz [86]. Pour les systèmes tactiles, la plupart des dispositifs commerciaux ont une fréquence d’échantillonnage de capteur synchronisée avec celle de l’écran autour de 60 Hz, mais certaines tablettes récentes comme l’iPad Air 2 atteignent 120 Hz [140]. Échantillonner plus rapidement nécessite par contre des ressources énergétiques et de calcul plus importantes. Pour avoir un compromis entre forte consommation et faible latence, Apple a introduit sur l’iPad Pro un échantillonnage adaptatif : comme la latence est plus facilement perceptible avec un stylet qu’au toucher [104], l’échantillonnage est à 120 Hz pour le toucher, mais double à 240 Hz lorsque le stylet est utilisé [3].

Le principe d’une fréquence qui s’adapte aux besoins semble se généraliser à l’affichage : Apple communique sur le fait que la fréquence d’affichage chute sur ses dernières tablettes tactiles si aucune application ne demande de mise à jour de l’écran, dans un but de conservation d’énergie. Cette approche est très récente et semble assez unique pour le moment. À l’heure actuelle, la plupart des écrans ou des projecteurs fonctionnent à 60 Hz. Certains dispositifs d’affichage conçus pour la 3D ou pour les jeux vidéo ont une fréquence de 120 Hz voire 144 Hz. Cette gamme de fréquence commence à arriver sur les écrans tactiles avec le nouvel iPad Pro dont la fréquence d’affichage adaptative peut atteindre 120 Hz [2]. Une bonne fréquence de rafraîchissement est essentielle. En effet, Janzen et Teather (2014) montrent qu’en plus d’introduire de la latence, elle influe aussi

négalement sur la performance des utilisateurs [67].

Latence des capteurs

Les capteurs peuvent introduire une latence de traitement. Pour éviter le bruit dans les données, les capteurs peuvent attendre de collecter plusieurs données avant de fournir une sortie lissée. Par exemple, à 60 hz, faire la moyenne des trois derniers points de données introduit une latence équivalente à une période soit 16.7 ms (Figure 2.3). Enfin, le temps de transmission de la donnée du capteur jusqu'à l'application s'ajoute aussi à la latence globale du système.

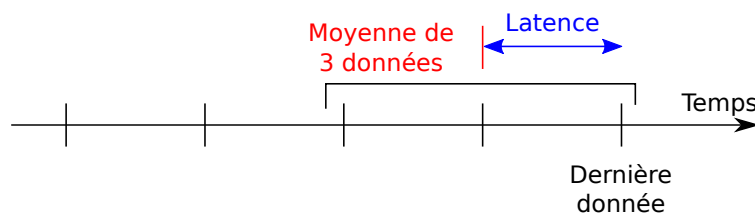


FIGURE 2.3 – Latence introduite par un lissage des données des capteurs. Dans cet exemple, si on fait la moyenne des trois dernières données au lieu de considérer uniquement la dernière, on introduit une latence d'une période.

Latence de l'application et pipeline graphique

Une fois les données des capteurs reçues, celles-ci sont traitées par une application qui elle aussi contribue à la latence. D'une part, l'application peut nécessiter un long temps de calcul. Par exemple, plusieurs dizaines de millisecondes peuvent être nécessaires pour mettre à jour un environnement virtuel complexe qui utilise un moteur physique sur de nombreux objets. Il appartient donc aux développeurs d'optimiser au maximum leurs applications pour minimiser cette source de latence.

Le système doit ensuite calculer l'image qu'il va devoir afficher à l'écran. Le temps de cette opération peut varier selon la complexité de l'environnement virtuel et la puissance de calcul fournie par la carte graphique.

Latence de l'écran

Les systèmes d'affichage ajoutent presque systématiquement un traitement du signal pour ajuster les couleurs, les contrastes ou adapter la résolution, ce qui génère de la latence. Une méthode courante pour mesurer la latence de ces pré-traitements sur les écrans LCD est d'utiliser la méthode du chronomètre [5]. L'écran à tester est mis côte-à-côte avec un écran à tube cathodique qui n'effectue pas ce type de pré-traitement de l'image. On affiche un chronomètre sur les deux écrans en clonant la sortie de la carte graphique (Figure 2.4).

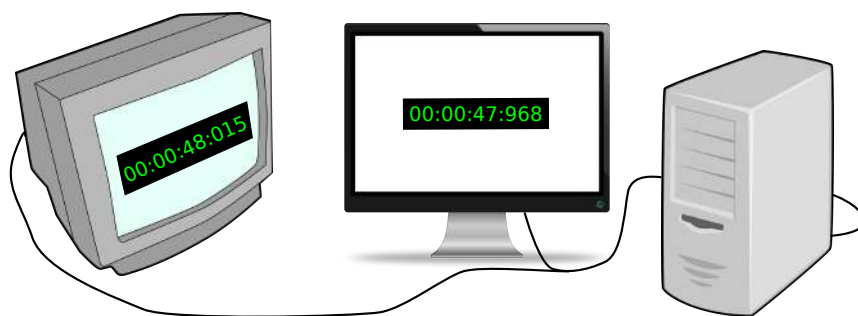


FIGURE 2.4 – Technique du chronomètre pour mesurer la latence d’un écran LCD. On duplique la sortie vidéo d’un ordinateur sur un écran à tube cathodique et sur l’écran LCD à tester, on affiche un chronomètre et on compare les temps sur une photo. Sur cette image, l’écran LCD (à droite) a un retard de 47 ms.

Avec une photo du dispositif, on compare la différence de temps entre les chronomètres de chaque écran, ce qui donne le retard de l’écran LCD sur l’écran cathodique.

De plus, l’affichage sur un écran LCD ne se fait pas de manière simultanée en tout point de l’écran. La mise à jour des pixels se fait progressivement, ce qui génère en réalité des latences différentes selon les parties de l’écran considérées. Nous avons capturé une vidéo haute fréquence (~ 2000 Hz) d’un écran Asus VG248QE fonctionnant à 120 Hz, et constaté que les pixels se mettent à jour ligne par ligne du haut vers le bas, le total de l’opération prenant environ 7 ms. Casiez et al. (2015) ainsi que Deber et al. (2016) observent ce même phénomène et concluent qu’une mesure de latence peut varier selon l’endroit de l’écran où on l’effectue [32, 45]. Les auteurs conseillent donc d’effectuer les mesures à une position standard comme le centre de l’écran.

Bilan

Les sources de latence sont diverses et peuvent provenir du matériel mais aussi du logiciel (Figure 2.1). Il appartient donc autant aux fabricants de systèmes interactifs qu’aux développeurs d’applications de minimiser la latence et de la contrôler en la mesurant. Suite à l’analyse et à la connaissance de ces différentes sources de retard, nous avons construit notre propre système tactile faible latence en essayant d’optimiser chaque couche du pipeline entre le toucher du doigt et l’affichage à l’écran. Ce système est décrit au chapitre 4.

2.1.2 Mesure de latence avec caméra et annotation d’images

Dans le découpage proposé ci-dessus, toute latence liée au logiciel, comme la latence de l’application, est facile à mesurer. En effet, il suffit de mémoriser la date précise à deux étapes du programme pour mesurer le temps qui s’écoule entre ces deux dates. Cependant, lorsqu’on s’intéresse à la latence bout-à-bout du système, le problème devient plus ardu

car la latence des composants d'entrée et de sortie du système (capteurs, écran) doit être intégrée à la mesure. Une possibilité est alors d'utiliser un outil d'observation extérieur avec lequel on détecte à la fois l'action de l'utilisateur et le retour visuel de l'écran. Par exemple, on peut prendre une vidéo du système en action et mesurer l'écart (en nombre d'images successives de la vidéo) entre le moment où l'action est exécutée et le moment où le système produit la sortie correspondante.

Pour la latence d'un écran tactile, on différencie la latence du premier contact, lorsqu'un doigt ou un stylet rentre en contact avec l'écran, de la latence en régime continu, lorsqu'un doigt ou un stylet glisse sur l'écran. Ces deux latences peuvent être différentes car elles n'impliquent pas forcément les mêmes processus, notamment au niveau des capteurs du système.

Latence du premier contact

En 2010, Kareesoja et Brewster proposent une méthode pour mesurer la latence lors du contact d'un stylet sur un téléphone mobile [70]. Une caméra capture à 300 images par seconde une séquence d'un stylet venant toucher l'écran du téléphone. Les images sont ensuite analysées par un opérateur pour mesurer la durée entre le moment où le stylet touche l'écran et le moment où l'écran du téléphone fournit un retour visuel. Pour ce type de mesure, le challenge est de détecter précisément le moment où le stylet entre en contact avec l'écran ce qui n'est pas facile à percevoir quand la vidéo est vue du dessus (ce qui est le cas, car on veut aussi voir l'écran). Les auteurs résolvent ce problème en accolant un miroir au téléphone qui fournit une image réfléchie du stylet vue de côté, ce qui permet ensuite sur la vidéo de clairement identifier le moment du contact. Des mesures sont réalisées sur quatre téléphones différents et deux applications. Les mesures démontrent que la latence varie selon les téléphones et l'application avec des valeurs allant de 60 ms à 200 ms.

Latence pour un mouvement continu

Pour mesurer la latence lors d'un mouvement continu, capturé par une caméra, il est utile d'identifier des points caractéristiques du mouvement. Cela permet de repérer les moments où l'objet dont on suit la position (le doigt pour un écran tactile, un objet sur lequel on ajoute de l'information pour de la réalité augmentée...) puis le retour visuel atteignent ces points particuliers sur les images de la vidéo.

He et al. (2000) proposent d'afficher une grille fixe en arrière plan (Figure 2.5). Cela fixe une référence spatiale sur le film pris par une caméra. Les auteurs repèrent alors deux images où l'objet et le retour visuel sont à la même position sur la grille [57]. La différence de temps entre la prise de ces deux images est une mesure de la latence du système.

Il est aussi possible d'utiliser des mouvements régis par des lois physiques connues pour convertir un décalage spatial observé sur une unique image de la vidéo en un déca-

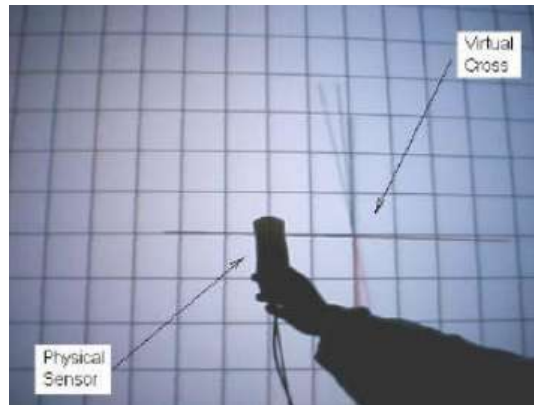


FIGURE 2.5 – Méthode de mesure de latence de He et al. [57].

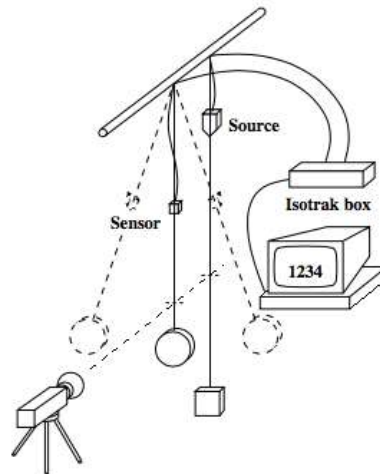


FIGURE 2.6 – Méthode de mesure de latence de Liang et al. [90]. La caméra filme à la fois le pendule, et l'écran qui affiche la date courante.

lage temporel. Par exemple, pour mesurer la latence d'un système de tracking (*Polhemus Isotrak*), Liang et al. (1991) proposent de filmer en même temps un pendule et un écran qui affiche la date actuelle du système [90] (Figure 2.6). Un capteur du *Polhemus Isotrak*, attaché au pendule, fournit au système la position du pendule. Sur une séquence vidéo enregistrée, les auteurs repèrent une image où le pendule est en position verticale, et notent la date affichée sur l'écran à ce moment là. Dans un fichier journal ayant enregistré tous les événements reçus par le système, ils retrouvent le dernier événement de position reçu à cette date. Cette position est en retard par rapport au pendule physique vertical. Connaissant l'amplitude et la période du pendule, ils en déduisent précisément l'écart de temps entre les deux positions, qui correspond à la latence du système.

Teather et al. (2009) utilisent aussi un mouvement pendulaire pour mesurer la latence d'un système de tracking optique (*Optitrack*) et d'une souris [137]. Le mouvement du pendule est suivi de deux manières : d'une part par l'*Optitrack*, et d'autre part par une



FIGURE 2.7 – Méthode de mesure de latence de Teather et al. [137]. La souris est fixe. La plaque de polystyrène entre l'écran et la souris effectue des mouvements pendulaires pendant la mesure. Les points gris sur la plaque sont les marqueurs réfléchifs suivis par le système Optitrack.

souris. Les deux trajectoires enregistrées par chacun des deux systèmes sont ensuite chacune retranscrites à l'écran par une ligne représentant les positions enregistrées. En filmant à la fois le pendule et l'écran, le nombre d'images de la vidéo entre le moment où le pendule rebrousse chemin à une position extrême, et celui où une des deux lignes rebrousse chemin à l'écran indique le retard de l'un ou de l'autre des deux systèmes. La caméra filme à 60 Hz, et la mesure n'est donc précise qu'à une période près (16.7 ms). Pour améliorer cette précision, les auteurs répètent la mesure 10 fois et calculent la moyenne des latences obtenues.

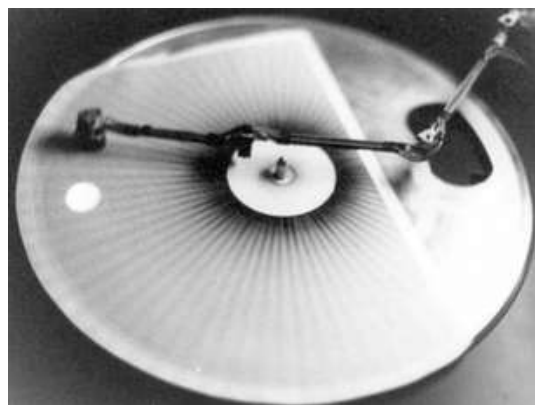


FIGURE 2.8 – Méthode de mesure de latence de Swindells et al. [136]. L'angle entre le retour visuel (disque blanc) et le bras rotatif du phonographe donne une mesure de la latence du dispositif de réalité augmentée.

Pour caractériser la latence d'un système de réalité virtuelle ou de réalité augmentée,

Swindells et al. (2000) proposent d'utiliser un mouvement de rotation à vitesse constante en se servant d'un phonographe [136]. Le mouvement d'un disque physique attaché au phonographe est suivi par un système Optotrack. Un disque virtuel est alors affiché par le système à la dernière position reçue. Lorsque le phonographe est à l'arrêt, le disque virtuel apparaît superposé au disque physique. Quand le phonographe tourne, à cause de la latence du système, le disque virtuel est en retard (Figure 2.8). Avec une photo du système, en mesurant l'angle de ce retard, et comme on connaît la vitesse de rotation du phonographe, le décalage angulaire peut être convertit en retard temporel.

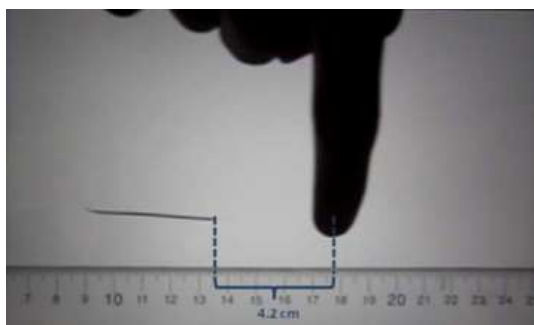


FIGURE 2.9 – Méthode de mesure de latence de Ng et al. [106].

Pour la latence d'un écran tactile, Ng et al. (2012) enregistrent une vidéo à 240 Hz d'un doigt traçant une ligne sur un écran tactile à côté d'une règle [106]. Sur les images de la vidéo, les auteurs analysent l'écart en cm entre le doigt et le trait affiché à l'écran (Figure 2.9). En observant quelques images successives, ils déduisent aussi la vitesse du doigt et convertissent alors la distance mesurée précédemment en un temps de latence entre le doigt et l'affichage du trait. Les auteurs mesurent aussi les meilleurs cas et pire cas de latence en cherchant les images du film où le trait est le plus proche ou le plus loin du doigt.

Limites de l'annotation d'images à la main

Ces différentes techniques de mesures de la latence d'un système comportent un inconvénient majeur : l'annotation humaine des vidéos. L'analyse des vidéos par des humains peut en effet introduire un biais subjectif (juger que tel objet est à telle position sur telle image) et demande beaucoup de temps, ce qui limite le nombre de mesures. Ainsi, He et al. reportent que l'annotation des vidéos leur a pris 10h [57], et Helps et Helps que l'annotation image par image nécessite plusieurs heures de travail fastidieux [59]. Le fait que l'annotation soit pénible réduit le nombre possible de mesures.

De plus, ce type de mesure ne fournit qu'une évaluation instantanée de la latence alors que celle-ci peut être variable. Cette variabilité ne pourrait être observée qu'à partir d'un nombre important de mesures successives ce qui est difficile avec l'annotation à la main. Certains travaux apportent donc des améliorations de ces techniques en automatisant la

mesure de la latence, soit par des techniques de vision par ordinateur pour analyser les images des vidéos, soit par l'intégration complète de la caméra (ou d'un capteur photosensible) dans un mécanisme électronique de mesure de latence.

2.1.3 Automatisation de la mesure de la latence

Traitement d'images

L'utilisation d'algorithmes de traitement d'images permet l'automatisation de l'annotation des images filmées par une caméra. Ces traitements extraient l'entrée de l'utilisateur et le retour visuel, les comparent et déterminent le retard entre les deux.



FIGURE 2.10 – Pendule utilisé par Steed pour mesurer la latence [133]. Grâce à La LED rouge accrochée au pendule, les auteurs extraient facilement la trajectoire du pendule sur les images prises par une caméra.

Steed (2008) réutilise le principe du pendule en l'automatisant. Une caméra filme à la fois un pendule et un retour visuel projeté sur un mur, généré par la position suivie du pendule [133]. Les positions du pendule et de l'image sont extraites automatiquement de la vidéo. Les auteurs font ensuite concorder une sinusoïde avec les deux trajectoires du pendule et de l'image virtuelle. Le déphasage entre les sinusoïdes correspond à la latence du système.

En utilisant l'extraction des deux trajectoires par traitement d'image, Friston et Steed (2014) proposent aussi une méthode pour automatiser le comptage du nombre d'images sur la vidéo entre les moments où les trajectoires de l'utilisateur et du retour visuel passent par un même point caractéristique du mouvement [55]. Les points caractéristiques choisis sont les pics d'accélération, lorsque la vitesse est nulle et que la trajectoire change de direction. Après un lissage gaussien, ces points sont extraits sur les deux trajectoires (utilisateur et retour visuel). Par soustraction des dates de deux points correspondant, les auteurs en

déduisent la latence. Le fait d’avoir de nombreux points caractéristiques lors d’une unique mesure permet d’obtenir une mesure moyenne plus précise (± 3 ms) que lorsque les images de la vidéo sont analysées à la main.

Pour tester la latence de leurs smartphones, la firme Sony utilise aussi des tests automatisés à l’aide de bras robotisés qui viennent effectuer un contact ou se déplacer sur l’écran tactile. Un film est enregistré avec une caméra puis analysé avec du traitement d’image pour calculer la latence du dispositif [7].

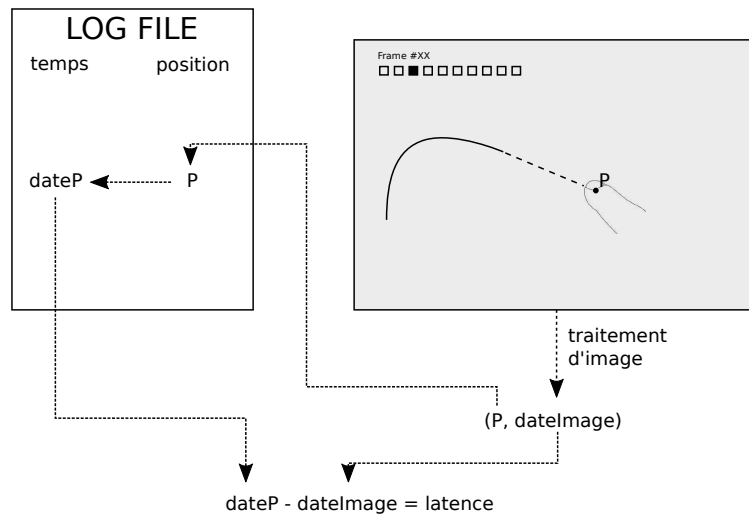


FIGURE 2.11 – Méthode de mesure de la latence au toucher “high accuracy” de Bérard et Blanch [19]. Sur une image de la vidéo on extrait la `dateImage`, codée à l’aide des carrés en haut à gauche, ainsi que la position `P` du doigt. Dans le fichier des événements, on retrouve la `dateP` à laquelle l’évènement de position `P` a été enregistré par le système. La différence entre les deux dates donne la latence du système.

Pour la latence d’un mouvement continu au toucher, en suivant la méthode de Liang [90], la solution proposée par Berard et Blanch (2013) encode et affiche la date courante à l’écran [19]. La mesure consiste à enregistrer à l’aide d’une caméra externe les mouvements du doigt de l’utilisateur sur l’écran ainsi que l’encodage de la date. Sur chaque image prise par la caméra, la position du doigt `P` et la date correspondante sont extraites grâce à des techniques de traitement d’images. En allant chercher dans la table des événements qui ont été enregistrés par l’application, on trouve l’évènement dont la position est `P` (si aucun évènement ne correspond, on interpole entre deux évènements). La comparaison de la date de cet évènement avec celle extraite de l’image donne alors la latence du système (Figure 2.11). Cette automatisation génère une mesure par image prise par la caméra, soit plusieurs dizaines de mesures par secondes. Cette grande quantité de mesures permet d’améliorer la précision de l’estimation, mais aussi de calculer un histogramme et donc d’avoir une idée de la distribution et de la variabilité de la latence.

Samsung a aussi déposé un brevet sur une technique de mesure de latence où une caméra filme à la fois le mouvement du doigt, et une diode qui s’allume lorsque la direction

du doigt est modifiée [125]. Un traitement d'image est ensuite effectué pour déduire le nombre d'images de la vidéo entre le changement de direction du doigt et l'allumage de la diode.

Circuit électronique

Une autre méthode pour automatiser la mesure de la latence est de concevoir un système électronique qui génère une entrée sur le système et capte le retour visuel fourni. La comparaison des signaux électriques générant l'entrée et généré par la sortie donne la latence du système.

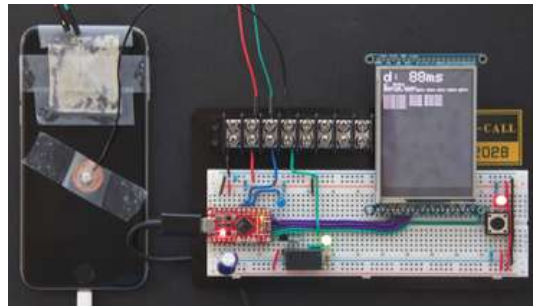


FIGURE 2.12 – Circuit électronique utilisé par Beyer et al. pour mesurer la latence d'un smartphone [23]. Une photodiode est positionnée sur le haut de l'écran pour percevoir le retour visuel. Un objet métallique est connecté au centre de l'écran pour simuler des contacts tactiles.

La caméra est alors remplacée par une simple photodiode capable de détecter un changement sur l'écran. La photodiode est intégrée dans un circuit électronique. Le signal d'entrée peut être généré en connectant un objet conducteur à l'écran ou détecté avec un capteur de vibration sur le doigt qui touche l'écran. Ce signal d'entrée est comparé au signal capté par la photodiode (Figure 2.12). Le retard entre le signal d'entrée et le moment où la photodiode capte un changement à l'écran indique la latence du système. Beyer et al. [23] (2015), Kamarainen et al. [74] (2016), Deber et al. [45] (2016) et Casiez et al. [33] (2017) se servent de cette méthode pour mesurer la latence sur différents dispositifs mobiles. Par contre, cette méthode est limitée à la mesure de latence du premier contact.

Papadakis et al. (2011) utilisent une méthode similaire pour évaluer la latence entre le suivi d'un objet et la réaction d'un environnement virtuel [108]. Un objet est déplacé automatiquement et est suivi par le système. Lorsque sa position franchit un certain seuil, une partie de l'affichage passe de noir à blanc. Un oscilloscope monitore à la fois la position de l'objet et la luminosité de l'écran (à l'aide d'une photodiode). La comparaison des deux signaux donne alors le retard avec lequel l'écran a répondu au franchissement du seuil.

Boucler la sortie sur l'entrée

Si on parvient à boucler la sortie du système sur sa propre entrée, alors on peut mesurer la latence de manière automatisée et sans avoir à disposer d'un dispositif externe. Créer une telle boucle forme un cycle, le retour fourni par le système générant une nouvelle entrée et ainsi de suite. Le placement d'un minuteur à n'importe quel moment de ce cycle permet alors de mesurer le temps d'un cycle complet, c'est-à-dire la latence du système.

En réalité augmentée, le système capture une image du monde réel avec une caméra, et en fonction de cette image, calcule un contenu numérique qui vient s'afficher sur un écran pour se superposer à l'image du monde réel. Sielhorst et al. (2007) proposent une méthode pour mesurer la latence sur un dispositif de réalité augmentée en utilisant la propre caméra du système qui filme l'écran du système [129]. L'écran affiche de manière visuelle (un disque qui se déplace) la date courante qui, captée par la caméra, est réinjectée dans le système (passant donc par tout le pipeline qui ajoute du retard) pour être réaffichée par l'écran. Plus en détail, un disque *date* se déplace à vitesse constante et représente le temps qui passe. La caméra détecte la position de ce disque *date*, en déduit la date actuelle, et demande au système de réafficher cette date avec un second disque *retour caméra*, sur une ligne au dessus de celui qui vient d'être capté. En fait, le temps que le système affiche ce second disque, le disque *date* se sera déplacé. La différence de position entre les deux disques, qu'on analyse sur chaque image prise par la caméra, indique la latence du système. Le fait d'obtenir une mesure à chaque image permet, comme dans la technique proposée par Bérard et Blanch [19], d'obtenir des informations sur la distribution et la variabilité de la latence.

D'une manière similaire, Casiez et al. (2015) utilisent une méthode pour mesurer la latence d'une souris qui ferme la boucle entrée/affichage en injectant la sortie de l'écran comme entrée du système [32]. La mesure consiste à poser la souris sur l'écran qui affiche une certaine texture. Un déplacement de cette texture génère alors un faux déplacement de la souris qui est ensuite enregistré par le système et traité comme tel. L'instant auquel la texture a été déplacée étant connu, il peut être comparé au moment où l'événement généré par la souris a été reçu ou à des moments ultérieurs dans le pipeline d'affichage, pour mesurer la latence à différentes étapes de traitement.

Pour la latence des écrans tactiles, reprendre cette approche nécessite de générer une entrée tactile qui mime l'affichage à l'écran. Bérard et Blanch (2013) proposent une approche de ce type en s'appuyant sur la participation d'un opérateur humain [19]. L'écran affiche un disque qui tourne à vitesse constante sur un cercle. Lors de la mesure, les auteurs demandent à l'opérateur de suivre ce disque au mieux avec son doigt. Si on suppose que l'opérateur suit parfaitement le disque, alors, à chaque fois qu'un événement de toucher est généré, la position du doigt à l'écran est connue car c'est la position où le disque est affiché. La position de l'évènement de toucher est alors comparée avec celle du doigt pour connaître la latence du système. En réalité, le suivi du disque par l'opérateur est impar-

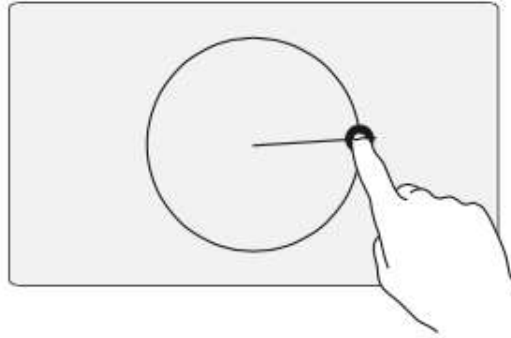


FIGURE 2.13 – Méthode de mesure de la latence au toucher “low overhead” par Bérard et Blanch [19]. Un opérateur humain doit suivre le plus précisément possible avec son doigt un disque qui tourne de manière régulière sur un cercle. Le détail de la méthode est précisé ci-dessous.

fait, mais les tests réalisés par les auteurs montrent que la variabilité autour du disque s'équilibre lorsqu'ils font la moyenne des latences sur plusieurs échantillons. Comme la fréquence du système échantillonne plusieurs dizaines d'évènements par secondes, quelques secondes seulement de suivi du disque résultent en une mesure assez précise de la latence. Dans l'étude proposée par les auteurs, cette méthode donne une mesure de latence à moins de 5 ms d'une valeur obtenue par une autre technique “haute précision” décrite dans les mêmes travaux [19].

Dans la technique précédente, pour aboutir la sortie du système avec son entrée, l'utilisateur est forcé à suivre la sortie du système. Une conception inverse du problème est de modifier artificiellement la sortie du système jusqu'à ce qu'elle corresponde parfaitement à l'entrée. Cette approche est évoquée par Swindells et al. (2000) qui décrivent une méthode prédictive pour un système de réalité augmentée [136]. Au lieu d'afficher la sortie normale du système, les auteurs affichent une sortie prédite. Lorsque la prédiction vient se superposer parfaitement avec l'entrée, alors le temps prédit correspond à la latence du système. Cependant, les auteurs ne testent pas la méthode car ils la jugent “probablement moins précise” que leur méthode du phonographe avec annotation d'une photo à la main (cf. paragraphe *Latence pour un mouvement continu*). Cette approche est reprise par Knibbe et al. (2015) pour de la projection virtuelle sur des balles de jonglage [78]. Une projection prédite est affichée, filmée par une caméra et analysée. Le temps de prédiction qui donne la meilleure superposition entre l'image virtuelle et les balles physiques est considéré comme la latence du système. Les auteurs ne testent la prédiction que par pas de 5 ms, ce qui limite la précision de la méthode. De plus, ils ne comparent pas les résultats de cette technique avec ceux obtenus par une autre technique de mesure.

Cette approche de boucler le système sur lui-même en raccordant sa sortie sur son entrée a l'avantage d'utiliser l'horloge interne du système informatique pour analyser la latence et ne nécessite aucun dispositif externe. Cependant les méthodes utilisées sont ad-hoc et doivent être pensées au cas par cas car elles dépendent beaucoup de la nature des

entrées/sorties.

2.1.4 Bilan

Dans cette section, nous avons considérées différentes approches pour mesurer la latence. Celles-ci sont résumées en Figure 2.14 pour l'exemple d'un écran tactile.

Mesurer la latence bout-à-bout d'un système nécessite d'avoir une horloge capable de capter à la fois le moment de l'interaction et celui du retour du système. Dans les travaux antérieurs, cette horloge est souvent celle d'un dispositif externe au système dont on veut mesurer la latence.

Il peut s'agir d'une caméra qui enregistre l'interaction avec le système. On analyse ensuite les images pour mesurer le temps écoulé (en nombre d'images) entre l'instant de l'interaction et la réaction du système. Cette méthode est relativement simple à mettre en place mais s'avère fastidieuse car elle nécessite soit une annotation des images à la main longue et pénible, soit la création d'algorithmes de traitement d'images pour capturer les positions du doigt et du retour visuel sur les images de la vidéo.

On peut aussi utiliser l'horloge d'un circuit électronique qui génère une entrée sur le système et en capte la sortie. Cela demande en revanche un minimum de connaissances dans le domaine de l'électronique et est donc peu accessible aux non-spécialistes.

Une dernière approche est de boucler le système sur lui-même. Dans cette catégorie de technique de mesure, des méthodes prédictives ont été décrites mais jamais évaluées. Une des contributions de notre travail a été de programmer et de tester une technique de mesure de latence au toucher pour un mouvement continu sur la base de cette méthode. Ces travaux sont détaillés au chapitre 4.

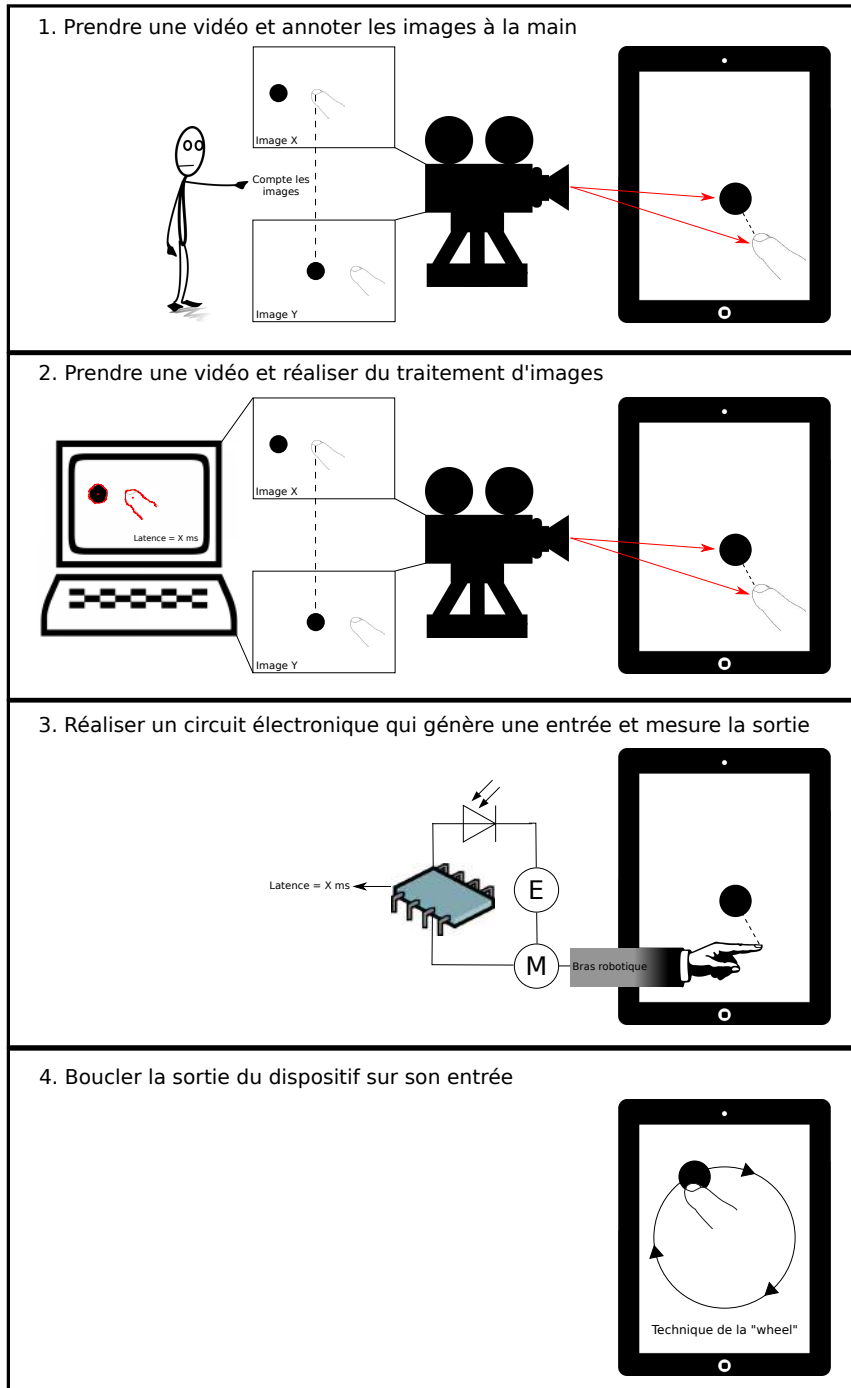


FIGURE 2.14 – Bilan de quatre approches pour mesurer la latence.

2.2 Perception et effets de la latence

Dans cette section, nous présentons les travaux de la littérature sur la perception et les effets de la latence en les organisant selon trois axes. Le premier traite de la perception : quel est le niveau minimal auquel, selon les conditions, les utilisateurs peuvent-ils percevoir un retard ? Le second axe traite des effets qualitatifs de la latence comme la frustration qu'elle peut entraîner chez les utilisateurs. Le troisième axe traite de l'impact quantitatif de la latence sur la performance de l'utilisateur : combien de temps perd-on, ou quelle précision perd-on lorsqu'on réalise une tâche sur un dispositif soumis à de la latence ?

En préambule, cette section explore les travaux sur la perception et les effets qualitatifs et quantitatifs de la latence en réalité virtuelle. Ce domaine où les problèmes liés à la latence sont apparus critiques très rapidement fournit des résultats que l'on peut apparenter avec ceux sur la latence au toucher qui sont détaillés par la suite.

2.2.1 La latence en réalité virtuelle

En réalité virtuelle, l'utilisateur est plongé dans un environnement numérique qui réagit, entre autres, aux mouvements de sa tête grâce au port d'un casque. Si le système souffre de latence, cela peut créer de l'oscillopsie, c'est-à-dire l'illusion que le monde autour de l'utilisateur se déplace alors que son regard ne bouge pas. Lorsqu'ils demandent aux participants de leur étude de bouger la tête en regardant sur les côtés, Allison et al. (2001) reportent que les participants perçoivent des instabilités de plus en plus fortes avec une plus grande latence ou de plus grands mouvements de la tête [13]. Jerald (2010) affine ce résultat en montrant que les seuils de perception de la latence sont reliés à l'inverse des pics d'accélération de la tête de l'utilisateur [68]. Selon ses résultats, l'auteur suggère que la latence des systèmes de réalité virtuelle devraient être inférieure à 3 ms pour que l'utilisateur ne perçoive plus de mouvement de la scène dû à la latence.

Les résultats de la littérature sont contradictoires sur le fait que cette oscillopsie due à la latence contribue à une sensation de malaise. Certaines études montrent une corrélation entre latence et nausée [48] alors que d'autres ne trouvent aucun effet [50]. Ce débat ne s'inscrit pas dans le cadre de cette thèse, mais un état de l'art plus précis est disponible dans les travaux de Buker et al. [30].

Certains auteurs étudient aussi l'impact de la latence de l'environnement virtuel sur la performance de pointage. Ware et Balakrishnan (1994) montrent que pour une tâche de pointage 3D dans un environnement virtuel, avec des latences variant entre 87 ms et 364 ms, la latence du système qui reproduit les mains de l'utilisateur dans l'environnement influe négativement sur la performance. Cependant, ce n'est pas le cas de la latence du système qui suit les mouvements de la tête, probablement car les sujets ne bougeaient presque pas la tête pour effectuer la tâche. So et Chung (2005) précisent ensuite ce résultat en montrant que, pour une tâche de pointage 3D, la latence de suivi des mains interagit

avec la taille de la cible, mais pas avec la distance de la cible [131].

2.2.2 Perception de la latence au toucher

Des études récentes se sont penchées sur le problème de la perception de la latence au toucher. Comme pour la réalité virtuelle, la latence reste perceptible à des niveaux très faibles (Figure 2.15).

Raaen et Eg [117] (2015) et Kaaresoja et al. [71] (2014) s'intéressent à la perception du retard par les utilisateurs entre l'appui sur un bouton et un retour visuel. Pour Raaen et Eg, le retour visuel est le changement de direction de rotation d'un disque à l'écran. Pour Kaaresoja et al., c'est l'allumage d'une LED sur le bouton. Dans les deux études, les participants jugent de la simultanéité entre l'appui sur le bouton et le retour visuel. Les résultats sont similaires dans les deux travaux : aux alentours de 80 ms, 75% des essais sont jugés comme étant simultanés.

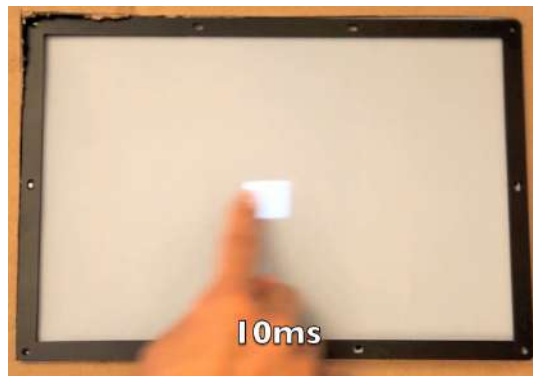


FIGURE 2.15 – Perception de la latence sur un écran tactile. Même à 10 ms de latence, on perçoit que le doigt est décentré de l'objet lorsque celui-ci est déplacé. Photographie issue de [106]

Ng et al. (2012) étudient la limite de perception de la latence par les utilisateurs pour une tâche de déplacement d'objet en utilisant un système tactile fabriqué sur mesure pour atteindre des latences très faibles [106]. Pour cela, les auteurs utilisent un protocole qui détermine un seuil de discrimination (ou "Just Noticeable Difference", JND) entre un niveau de latence fixe quasi nul (~ 1 ms) et un niveau plus élevé variable. Le protocole de JND propose plusieurs tests de discrimination au participant qui doit, après avoir déplacé un objet sur l'écran tactile dans deux conditions, choisir quelle condition lui semble avoir le niveau de latence le plus élevé. Le second niveau de latence varie en fonction des réponses du participant. Si le participant se trompe, le niveau variable devient plus élevé, et la discrimination est plus facile. Si le participant donne la bonne réponse, le niveau variable est diminué pour rendre la discrimination plus difficile. Le niveau de latence variable converge alors vers le seuil de perception du participant. Les auteurs montrent que le seuil de perception de la latence est très bas. Les participants à l'étude distinguent en moyenne des

retards aussi faibles que 6 ms, et jusqu'à 2 ms pour certains.

Ce seuil est très faible comparé aux 80 ms trouvées par Raeen et Eg et Kaaresoja et al. dont les résultats se basent sur une tâche de perception de la latence lors du premier contact pour une interaction indirecte. La perception de la latence est donc probablement dépendante de la tâche effectuée : toucher direct/indirect ou toucher premier contact/continu par exemple.











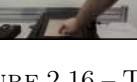
Image	Auteurs	Direct vs indirect	Toucher vs stylet	Tâche	Seuil de perception
	Ng et al. 2012	direct	toucher	déplacer un carré	6 ms
	Ng et al. 2014	direct	stylet	déplacer un carré	6 ms
	Ng et al. 2014	direct	stylet	déplacer un carré plus petit	2 ms
	Ng et al. 2014	direct	stylet	griffonner	40 ms
	Annett et al. 2014	direct	stylet	faire une ligne	53 ms
	Annett et al. 2014	direct	stylet	écrire	50 ms
	Annett et al. 2014	direct	stylet	dessiner	61 ms
	Deber et al. 2015	direct	toucher	déplacer un carré	11 ms
	Deber et al. 2015	direct	toucher	premier contact	69 ms
	Deber et al. 2015	indirect	toucher	déplacer un carré	55 ms
	Deber et al. 2015	indirect	toucher	premier contact	96 ms

FIGURE 2.16 – Tableau résumé des seuils de perception de la latence pour différentes tâches et modalité d'utilisation d'un écran tactile. Photographies issues de [15, 46, 104, 106]

Dans une seconde étude, Ng et al. (2014) généralisent leurs résultats en considérant le stylet de manière similaire au toucher [104]. Cette fois, le protocole de JND est répété, mais avec différentes tâches : déplacer un grand carré, déplacer un petit carré ou griffonner. Les résultats montrent alors que le seuil de perception est fortement dépendant de la tâche réalisée. En déplaçant le petit carré, il est plus facile de détecter un décalage entre la pointe du stylet et le centre du carré. Le seuil de perception tombe alors à 2 ms contrairement aux 6 ms obtenues avec le grand carré. Pour la tâche de griffonnage, le seuil est bien plus élevé,

à 40 ms. Les auteurs en déduisent que la charge cognitive pourrait influencer la perception du retard. La tâche de griffonnage est basique, mais demande une certaine attention sur la forme de la courbe et la longueur des traits, cela pourrait entraîner une charge cognitive et attentionnelle qui ferait chuter le seuil perceptif de la latence. Cette hypothèse est cohérente avec les résultats d'Annett et al. (2014) qui montrent des seuils encore plus élevés pour des tâches au stylet plus complexes comme un dessin de ligne (53 ms), l'écriture d'un mot (50 ms) ou le dessin d'une étoile (61 ms) [15].

Deber et al. (2015) fournissent une étude de synthèse sur la perception de la latence au toucher [46]. L'analyse inclut à la fois comme facteurs la tâche de premier contact et de déplacement d'objet ainsi que la modalité de l'interaction, directe ou indirecte. Les auteurs, s'appuyant encore une fois sur le protocole de JND, confirment que la latence est plus perceptible pour les tâches de déplacement d'objet et en interaction directe (JND=11 ms), qu'en interaction indirecte (JND=55 ms). Les seuils sont plus élevés pour le premier contact avec des seuils à 69 ms en interaction directe et 96 ms en interaction indirecte. Les auteurs étudient aussi à quel point une diminution de latence par rapport à un niveau d'origine donné est perceptible ou non. Leurs résultats indiquent si une diminution de latence sera suffisante ou non pour être perçue par les utilisateurs.

Les résultats de ces différentes études sur la perception de la latence au toucher ou avec un stylet sont résumés en Figure 2.16.

2.2.3 Impact qualitatif de la latence au toucher

La latence est donc perceptible à des niveaux très faibles pour l'interaction au toucher directe. On peut penser qu'en détectant que le système est peu réactif, les utilisateurs ressentent de la frustration ou de l'insatisfaction. Certaines études sont ainsi menées en interrogeant directement les utilisateurs sur l'utilisabilité d'un système à différents niveaux de latence.

En 2011, Anderson et al. testent l'utilisabilité d'un grand écran tactile sur trois applications différentes à différents niveaux de latence entre 180 ms et 780 ms [14]. Les trois applications incluent des mouvements de base comme le défilement d'une page, le zoom ou la navigation. Les participants de l'étude notent l'utilisabilité du système pour chaque type de mouvement et chaque niveau de latence. Les résultats indiquent une chute importante des notes à 580 ms. Les auteurs préconisent donc une latence inférieure à 580 ms pour assurer l'acceptation des systèmes tactiles par les utilisateurs.

Les résultats de Ritter et al. (2015) viennent appuyer ce seuil de 580 ms [119]. Dans leur étude, les auteurs séparent la tâche de premier contact de la tâche de déplacement continu. Les participants réalisent les deux tâches et notent leur acceptation de la latence entre 1 et 10. Les auteurs montrent que l'acceptation devient de plus en plus faible au fur et à mesure que la latence augmente. Les participants sont plus sensibles à la latence pour

le déplacement continu, les notes devenant mauvaises à partir de 450 ms, contre 650 ms pour la tâche de premier contact.

Miller et Mandryk (2016) comparent différentes frustrations générées lors d'un jeu de type *infinite runner* où un coureur doit éviter des obstacles [97]. Ils différencient la frustration positive qui peut naître d'un niveau de difficulté de jeu plus élevé (par exemple plus d'obstacles ou des puzzles plus compliqués) et la frustration négative qui peut être générée lorsque les commandes du jeu ne réagissent pas comme le joueur s'y attend. Ils montrent que les utilisateurs ne ressentent pas plus de frustration négative lorsque la difficulté provient d'une latence (aléatoire entre 0 et 300 ms) des commandes que lorsque le niveau du jeu devient plus difficile. En revanche, les participants prennent plus de plaisir dans la condition difficile que dans la condition avec latence.

Kang et al. (2016) constatent que des applications mobiles peu réactives obtiennent des notes basses sur leurs sites de vente et des plaintes d'utilisateurs telles que "Je déteste attendre plus longtemps que ce à quoi je m'attendais" ou "tout est super long" [75]. Les auteurs mènent donc une étude pour comprendre la corrélation entre la patience de l'utilisateur et la réactivité d'une application mobile. Les participants doivent noter leur patience et donner un retour qualitatif sur leur expérience lors de l'utilisation d'une application à trois niveaux de latence différents : 200 ms, 500 ms et 2 s. Les participants se sont plaints à 500 ms ("très peu réactif", "répond en retard"), voire énervés à 2 s, mais peu de commentaires négatifs émergent de la condition à 200 ms.

Ces résultats soulèvent un point intéressant quand à la corrélation entre : l'impact de la latence sur la perception et la performance de l'utilisateur, et la gêne ou la frustration engendrée sur l'utilisateur. Dans les études citées ci-dessus, on remarque que les seuils proposés par les auteurs, de l'ordre de plusieurs centaines de millisecondes, sont bien plus élevés que les seuils de perception décrits dans la section précédente ainsi que bien au dessus des seuils où la latence affecte encore la performance [69].

Les conditions de latence de plusieurs centaines de millisecondes que les utilisateurs jugent acceptables sont en revanche relativement proches de celles auxquelles les utilisateurs font face lorsqu'ils utilisent des écrans tactiles dans la vie de tous les jours. On peut donc supposer que bien que la latence soit perceptible et réduise la performance, elle n'engendre pas de frustration ou de mécontentement chez les utilisateurs car ceux-ci sont habitués à se servir d'écrans tactiles ayant de telles latences. Les résultats qualitatifs obtenus seraient donc fortement liés aux a priori des utilisateurs sur le comportement du système. Ng et al. font d'ailleurs remarquer que les gens ayant utilisé fréquemment leur système tactile basse latence et ayant pris l'habitude de faibles retards finissent par trouver la latence des écrans tactiles actuellement disponibles dans le commerce "complètement inacceptable" [106].

2.2.4 Impact quantitatif de la latence

En ce qui concerne l'évaluation quantitative de l'impact de la latence, les études portent principalement sur des tâches réalisées à la souris. Les études sur l'interaction au toucher sont plus récentes et moins nombreuses mais les résultats obtenus semblent indiquer que les conséquences de la latence sont similaires dans les deux domaines, avec des valeurs et des seuils différents. La suite de cette section fournit un état de l'art pour ces deux domaines. En préambule, nous décrivons la tâche de pointage qui est la plus utilisée pour évaluer la performance.

Pointage et loi de Fitts

L'influence de la latence a été étudiée sur différentes tâches comme le pilotage (qui consiste à se déplacer dans une chemin de largeur donnée [9]) ou le suivi de cible, mais la tâche la plus couramment considérée est la tâche de pointage. Un modèle fréquemment utilisé pour prédire la performance lors d'une tâche de pointage est la *loi de Fitts* que nous détaillons ci-dessous.

En 1954, Fitts propose un modèle de prédiction de performance pour le mouvement de pointage en 1D, c'est à dire le mouvement rapide du doigt vers une cible, en fonction de la distance à laquelle se situe cette cible et de sa taille [51]. Selon cette loi, le temps pour exécuter le mouvement (MT) est une fonction affine de l'index de difficulté (ID) de la tâche, défini comme le logarithme de la distance (D) sur la taille (W) de la cible (cf. Figure 2.17), les paramètres de la fonction affine étant déterminés empiriquement.

$$MT = a + b \times ID \text{ avec } ID = \log_2 \frac{2D}{W}$$

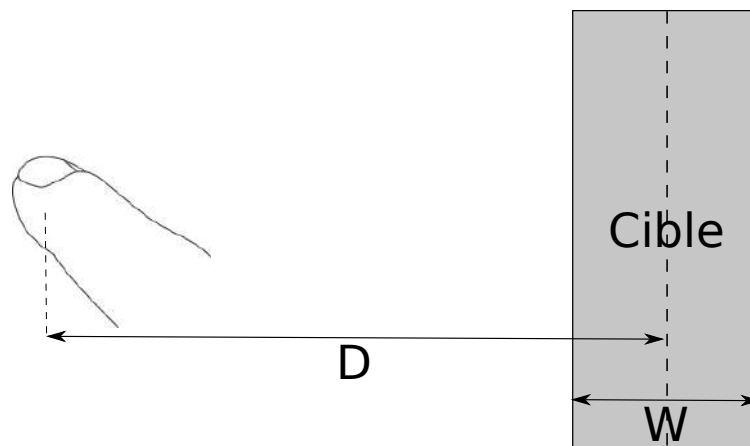


FIGURE 2.17 – Modélisation d'un mouvement de pointage par la loi de Fitts. Le mouvement de la main ou du doigt vers une cible, est décrit comme étant une fonction affine du logarithme de la distance D à la cible divisée par la largeur W de la cible.

La loi de Fitts a été adoptée dans le domaine de l'interaction homme machine car elle

prédit les temps de réalisation de pointages à la souris, mais aussi avec d'autres dispositifs comme le joystick, le stylet ou le doigt [132]. La loi de Fitts a plusieurs variations [114] mais la forme la plus utilisée dans le domaine de l'IHM est celle proposée par Soukoreff et MacKenzie [132]. Cette version inclut les améliorations précédemment introduites par MacKenzie [92]. L'ID est mesuré préférablement avec la formule dite de Shannon : $ID = \log_2(\frac{D}{W} + 1)$, qui fournit des régressions plus précises et des indices de difficulté toujours positifs. La formule est aussi généralisée pour des tâches en 2D.

L'IHM ajoute un paramètre supplémentaire qui n'est pas pris en compte par la loi de Fitts, qui est la possibilité de rater une cible (p. ex. si l'utilisateur clique alors que le pointeur est à côté de la cible). L'utilisateur doit alors trouver un juste milieu entre vitesse de pointage et nombre d'erreurs. La tâche peut être réalisée plus rapidement, mais au risque de faire plus d'erreurs et inversement. Ce compromis n'étant pas forcément le même pour tous les utilisateurs, MacKenzie propose une méthode pour normaliser l'erreur à 4% de cibles ratées. Une fois la tâche terminée, la taille théorique effective W_e de la cible est calculée pour que les données obtenues correspondent à 4% de cibles ratées puis l'index de difficulté effectif $ID_e = \log_2(\frac{D}{W_e} + 1)$. L'utilisation de ID_e à la place de ID compense pour les variations de la tolérance aux cibles ratées entre utilisateurs.

Soukoreff et MacKenzie introduisent aussi une mesure de bande passante ("throughput" en anglais), calculée comme une moyenne des indices de difficulté effectifs divisés par les temps de pointage correspondant. Cette bande passante offre une mesure globale de la performance combinant les effets de la pente et de l'ordonnée à l'origine de la droite fournie par la loi de Fitts. Le fait de disposer d'une unique mesure facilite la comparaison entre les conditions par exemple si la tâche est effectuée à différents niveaux de latence.

Les grandes surfaces tactiles, auxquelles nous nous intéressons particulièrement dans cette thèse, sont capables de détecter plusieurs points de toucher, et notamment une interaction bimanuelle. Kabbash et al. (1994) montrent que l'utilisation des deux mains peut améliorer la performance pour une tâche de manipulation directe [72], avec une performance presque doublée pour certaines tâches impliquant une interaction symétrique des deux mains [85]. Cependant l'interaction bimanuelle entraîne souvent un comportement plus complexe où la partition du travail entre la main dominante et non dominante n'est pas forcément symétrique [56, 73]. L'influence de la latence sur une tâche bimanuelle pourrait donc être différente d'un simple cumul de l'influence sur chacune des deux mains.

Impact de la latence sur la performance

On trouve des résultats sur les effets de la latence sur la performance dès 1959, avec une étude à propos des effets du délai sur l'écriture [143]. Celle-ci se fait grâce à un stylet électronique dont l'affichage est situé sur un écran séparé (interaction indirecte) et la qualité de la production écrite est mesurée en terme de vitesse, de clarté et d'erreur (omission, duplication ou substitution). L'étude montre que la latence introduit des perturbations qui

mènent à une production de lettres aux contours bruités, la clarté de l'écriture étant de moins en moins bonne au fur et à mesure que la latence augmente, de 0 à 520 ms. La vitesse d'écriture et le nombre d'erreurs quant à eux, sont liés par un compromis dépendant des instructions données.

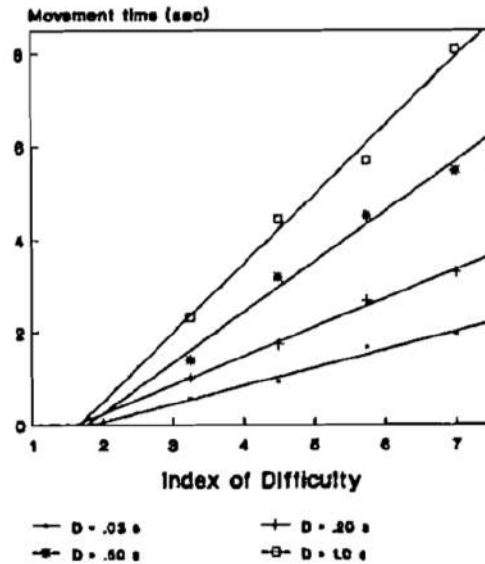


FIGURE 2.18 – Différentes droites de Fitts pour différents niveaux de latence révélées par les résultats de Hoffmann [62].

En 1992, Hoffmann expérimente les effets de la latence sur un dérivé de la tâche de pointage, aussi modélisable par la loi de Fitts [62]. La tâche consiste à tourner un bouton pour amener la pointe d'un enregistreur potentiométrique d'un point à un autre lorsque l'interaction est soumise à différents niveaux de latence allant de 30 ms à 1 s. Les résultats montrent que pour chaque niveau de latence, on retrouve la droite qu'indique la loi de Fitts. Cependant la pente de ces droites varie selon la latence, indiquant une corrélation entre l'influence de la latence sur le temps de pointage et la difficulté de la tâche (Figure 2.18). Hoffmann propose alors une extension de la loi de Fitts en incluant la latence :

$$MT = a + b \times ID + c \times LATENCE \times ID$$

Les paramètres a , b et c sont déterminés expérimentalement. Le dernier terme est ajouté par rapport à la version classique de la loi de Fitts. Il prend en compte l'influence de la latence ainsi que son interaction avec la difficulté de la tâche.

L'année suivante, MacKenzie et Ware obtiennent des résultats similaires pour une tâche de pointage en une dimension à la souris [93]. Une étude est menée sur 8 sujets qui réalisent des tâches de pointage de difficultés variables avec des latences allant de 8.3 ms à 225 ms. Le temps de pointage se modélise alors par la même loi que celle proposée par Hoffmann. Cette étude montre aussi que la différence de performance entre 8.3 ms et 25 ms de latence est négligeable. Cela indique qu'il pourrait exister un effet de seuil, et un niveau

de latence en deçà duquel la performance de l'utilisateur ne serait plus affectée.

Pavlovyh et Stuerzlinger (2009) examinent aussi le phénomène de la latence sur le pointage à la souris [111] à des niveaux variant entre 33 ms et 133 ms. Ils montrent que la performance en pointage (caractérisées par la bande passante) décroissent lorsque la latence devient supérieure à 58 ms. Une étude semblable est aussi menée par Teather et al. [137] conduisant à des résultats similaires : la bande passante diminue avec l'augmentation de la latence. Ces résultats sont ensuite généralisés à une interaction 3D, grâce à une tâche de positionnement de cube, version étendue de la tâche de pointage 2D. La différence de performance qui était visible en 2D entre 35 ms et 75 ms de latence s'efface et n'est plus significative pour la tâche en 3D. Cependant les auteurs font l'hypothèse que cela est probablement dû à une latence plus élevée et variable que prévue dans la condition 35 ms lors de la tâche 3D. Ivkovic et al (2015) étudient aussi les effets de la latence sur le pointage à la souris et obtiennent des résultats qui concordent avec les travaux précédents [65]. La latence n'a pas d'effet significatif sur le temps de pointage entre 11 ms et 41 ms, mais un effet devient visible entre 41 ms et 71 ms.

Ces travaux sur les tâches de pointage à la souris sont affinés en 2015 par Friston et al. qui, équipés d'un système à très faible latence (~ 6 ms), estiment l'impact de la latence sur un intervalle plus restreint : entre 6 ms et 36 ms par pas de 10 ms puis à 56 ms et 86 ms. Leurs résultats sont cohérents avec ceux des études précédentes, la latence ayant un impact sur la performance lorsqu'elle dépasse 56 ms. Les auteurs découpent le geste de pointage en trois phases : accélération (avant le pic de vitesse), acquisition (après le pic de vitesse et jusqu'à ce que la cible soit atteinte pour la première fois) et correction (après avoir atteint la cible et jusqu'à la validation par l'utilisateur). La phase d'accélération n'est pas touchée par la latence alors que le temps d'acquisition lui est réduit quand la latence augmente. On peut penser qu'en percevant le retard, les participants décelèrent moins rapidement après le pic de vitesse. La latence a aussi un effet négatif important sur la phase de correction. Ce résultat est cohérent avec celui de So et Chung [131] décrit précédemment pour la réalité virtuelle : il y a interaction entre latence et largeur de cible (qui influence la phase de correction) mais pas entre latence et distance de la cible (qui influence la phase d'acquisition). Friston et al. montrent que la latence a un impact sur le comportement des utilisateurs dès 16 ms même si les effets sur la phase d'acquisition et la phase de correction se compensent et qu'on ne voit donc pas d'effet global. Seulement une fois passé 56 ms (ce qui est cohérent avec les 58 ms trouvées par Pavlovyh et Stuerzlinger [111]), un clair impact de la latence apparaît sur les trois phases cumulées.

Certaines études examinent aussi les effets de la latence sur la performance d'autres tâches à la souris. Pavlovyh et Stuerzlinger (2011) considèrent une tâche de suivi de cible [112]. Ils montrent que la latence affecte l'erreur longitudinale (dans la direction du mouvement) entre 50 ms et 110 ms, cet effet interagit avec la vitesse de la cible : plus la cible est rapide, plus les effets négatifs de la latence se font ressentir. Dans la continuité, Pavlovyh et Gutwin (2012) étendent ces résultats en ajoutant une acquisition à la fin de

la tâche et en rendant les chemins de suivi plus ou moins imprévisibles [110]. Ils montrent que la latence influence aussi le nombre de clics nécessaire à l'acquisition et que l'effet de la latence sur l'erreur interagit avec la complexité du chemin. Friston et al. étudient aussi la tâche de pilotage [54]. Les résultats obtenus sont similaires de ceux pour les autres tâches. La latence a un effet négatif sur la performance à partir de 56 ms et cet effet interagit avec la difficulté de la tâche.

En résumé, pour l'interaction indirecte à la souris, les travaux antérieurs s'accordent sur un impact de la latence à partir d'un seuil compris entre 58 ms et 71 ms (Tableau récapitulatif 2.19). Au-delà de ce seuil, lorsque la latence augmente, elle fait progressivement diminuer la performance. Pour une tâche de pointage à une dimension, l'effet de la latence peut se modéliser avec une version modifiée de la loi de Fitts.

Impact quantitatif de la latence au toucher

Les études concernant la latence sur les écrans tactiles ne sont apparues que plus récemment. Jota et al., en 2013, sont les premiers à étudier l'effet négatif de la latence sur la performance utilisateur lors d'une tâche de positionnement d'objet [69]. Les niveaux de latence examinés par les auteurs (entre 1 ms et 50 ms) sont plus faibles que ceux des études à la souris mais un effet de la latence est démontré entre les conditions 10 ms et 25 ms. Un effet de la latence à des valeurs de cet ordre n'était pas visible pour la souris. Cela est cohérent avec les résultats sur la perception qui montrent que les utilisateurs sont plus à même de percevoir un retard lorsque l'interaction est directe [46]. Les résultats obtenus par Jota et al. suggèrent aussi qu'il n'y a pas forcément d'effet de seuil, même pour des valeurs de latence jusqu'à 1 ms. Comme pour l'étude de Friston et al. [54], un découpage du mouvement de pointage en trois phases démontre que c'est la phase finale de l'acquisition qui est la plus touchée par la latence.

2.2.5 Bilan

Lorsque l'interaction est directe, la latence est perçue à des niveaux plus faibles que lorsque l'interaction est indirecte. En particulier, pour un mouvement de toucher continu, les utilisateurs peuvent discerner des latences aussi basses que 6 ms lors de la réalisation d'une tâche simple telle que le déplacement d'un objet. Cependant, ce seuil de perception est différent selon les tâches effectuées. Plus la tâche est complexe et la charge cognitive augmente, moins les utilisateurs perçoivent la latence.

La latence affecte aussi la performance des utilisateurs. Quand l'interaction est indirecte, à la souris, la latence dégrade la performance à partir d'un seuil situé entre 50 ms et 60 ms. Ce seuil est fortement réduit pour l'interaction directe au toucher, puisque la performance sur une tâche de positionnement d'objet est affectée à des niveaux inférieurs à 25 ms. Sachant que pour des tâches plus complexes la latence est moins perceptible, on

Auteurs/Date	Dispositif	Tâche	Seuil de l'influence de la latence (S)
Bergeijk et D. 1959	Styler	Ecriture	$S < 40$ ms
Hoffmann 1992	Bouton rotatif	Pointage	$S < 30$ ms
MacKenzie et W. 1993	Souris	Pointage	$25 \text{ ms} < S < 75 \text{ ms}^{(1)}$
Pavlovych et S. 2009	Souris	Pointage	$58 \text{ ms} < S < 83 \text{ ms}$
Teather et al. 2009	Souris	Pointage	$S < 75$ ms
Teather et al. 2009	Souris	Pointage 3D	$75 \text{ ms} < S$
Ivkovic et al. 2015	Souris	Pointage	$41 \text{ ms} < S < 71 \text{ ms}$
Friston et al. 2015	Souris	Pointage	$56 \text{ ms} < S$
Friston et al. 2015	Souris	Pilotage	$56 \text{ ms} < S$
Pavlovych et S. 2011	Souris	Suivi de cible	$50 \text{ ms} < S < 110 \text{ ms}$
Jota et al. 2013	Toucher	Positionnement	$S < 25$ ms

FIGURE 2.19 – Tableau récapitulatif des études sur l'impact de la latence sur la performance de temps de pointage pour différents dispositifs et différentes tâches. ⁽¹⁾Ces valeurs sont discutées par Pavlovych et Stuerzlinger qui suggèrent un biais négatif de 60 ms [111].

peut supposer que la performance est alors aussi moins affectée quand la charge cognitive est plus lourde.

La forte présence de l'interaction au toucher directe dans notre vie quotidienne via les smartphones et les tablettes a donc amené la latence comme une question centrale de l'utilisabilité de ces dispositifs. Comme le montrent les résultats des études qualitatives, les d'utilisateurs prennent conscience de l'impact de la latence seulement à partir de niveaux assez élevés, autour de 500 ms. Cela pourrait être dû à une habitude à des niveaux de latences élevés sur les dispositifs actuellement disponibles dans le commerce. Dans tous les cas, leur performance est amoindrie par la latence et cette réduction pourrait interagir avec la charge cognitive de la tâche effectuée. Une des contributions de cette thèse, présentée au chapitre 4, est d'examiner cette interaction dans le cadre d'une tâche de positionnement bimanuelle.

Chapitre 3

Compenser les effets de la latence

Les effets négatifs de la latence ont motivé plusieurs travaux destinés à les réduire ou à les compenser. Du côté de la machine, la réduction de la latence peut émaner des fabricants de systèmes tactiles pour créer du matériel plus performant mais aussi des concepteurs d'applications pour avoir des logiciels plus réactifs. Du côté des utilisateurs, la réduction des effets de la latence existerait aussi sans amélioration matérielle ni logicielle si les utilisateurs développent des stratégies motrices pour maintenir leur performance malgré la présence de retard.

Ce troisième chapitre explore les travaux antérieurs portant sur la compensation de la latence d'abord du côté machine puis du côté humain afin de répondre aux questions suivantes :

- Quelles améliorations matérielles et logicielles sont réalisables pour réduire la latence ?
- Quelles stratégies les utilisateurs sont-ils capable de développer pour compenser les effets négatifs de la latence ?

3.1 Réduction et compensation de la latence : aspects techniques

La latence bout-à-bout d'un système est le cumul de retards provenant de différentes couches du système (cf. chapitre 2). Toute possibilité de réduire la latence dans chacune de ces couches et dans leur communication contribue ainsi à réduire la latence bout-à-bout. Nous passons donc en revue les optimisations de différentes sources de latence réalisées dans différents travaux de la littérature, en nous focalisant principalement sur l'interaction au toucher.

3.1.1 Capteurs et Affichage

Réduction de la latence des capteurs

Les écrans tactiles capacitifs fonctionnent à l'aide d'une matrice d'électrode (cf. chapitre 2). Une première étape pour minimiser la latence des capteurs tactiles est d'optimiser la manière dont le signal électrique est capté sur cette matrice pour détecter le point de contact du doigt. Les capteurs capacitifs traditionnels utilisent une approche *time-division multiplexing* (TDM) où chaque rang de la matrice est activé séquentiellement. Leigh et al. (2014) proposent un capteur "Fast Multi-Touch" (FMT) où les rangs sont tous scannés en parallèle [88] (Figure 3.1). FMT fonctionne à 4000 Hz avec une latence entre le toucher physique et le calcul de la donnée de position de seulement $40 \mu\text{s}$. L'intérêt du capteur est démontré au travers d'un dispositif tactile complet, aussi optimisé sur le traitement des données et l'affichage, et qui parvient à une latence bout-à-bout de seulement $110 \mu\text{s}$. Cependant, le fonctionnement haute fréquence de FMT implique une très forte consommation d'énergie.

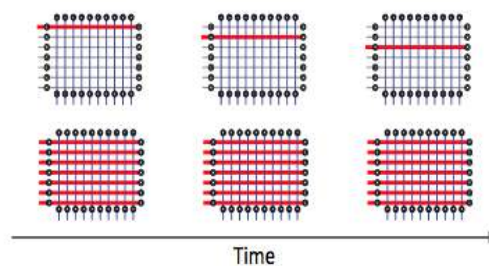


FIGURE 3.1 – FMT, un capteur tactile capacitif haute performance [88]. Dans une approche classique (haut), les rangées d'électrodes sont activées séquentiellement. Dans FMT toutes les rangées sont activées simultanément à chaque pas de temps.

Ng et al. (2015) utilisent aussi une approche de parallélisation pour réduire la latence d'un dispositif tactile résistif [105]. La surface tactile est une matrice d'interrupteurs connectant des colonnes potentiellement alimentées avec des lignes reliées à un multiplexeur. La présence d'un doigt sur l'écran ferme l'interrupteur et connecte une ligne et

une colonne. Si une colonne C est alimentée et que le multiplexeur reçoit un signal sur la ligne L , alors c'est qu'un doigt est posé à l'intersection de C et L . Si rien n'est reçu par le multiplexeur, cela signifie qu'aucun doigt n'est présent sur les colonnes activées. Il est possible d'alimenter plusieurs colonnes simultanément, et si le multiplexeur ne reçoit rien, éliminer beaucoup de positions potentielles en un seul cycle. Les auteurs se servent aussi du fait qu'entre deux cycles, le déplacement d'un point de contact est limité et, pour gagner du temps, ils ne scannent qu'un minimum d'intersections autour de la dernière position connue. Le capteur fonctionne à plus de 1000 Hz et a donc une latence inférieure à 1 ms.

Il existe aussi d'autres algorithmes conçus pour réduire la complexité de la détection d'un toucher unique [76] ou de plusieurs points de contact [12].

Réduction de la latence de l'affichage

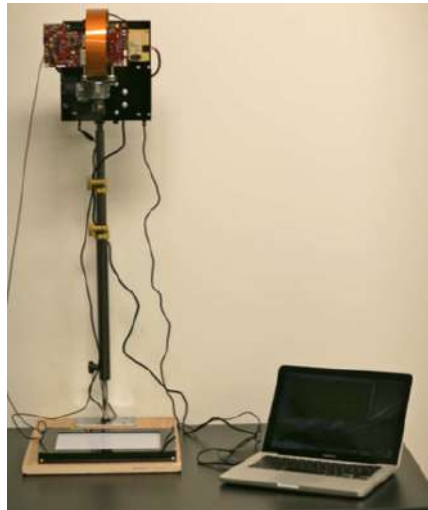


FIGURE 3.2 – Système faible latence utilisé par Ng et al. [106]. L'affichage se fait grâce au projecteur DLP en haut à gauche de l'image qui projette directement sur la surface tactile en bas à gauche.

Les capteurs utilisés par Ng et al. [105] et Leigh et al. [88] ont été tous deux testés en étant intégrés dans un dispositif tactile entièrement optimisé pour la faible latence. Dans les deux cas, l'affichage est réalisé en utilisant un projecteur DLP (*Digital Light Processing*, Figure 3.2). Cette technologie utilise une puce constituée d'une matrice de miroirs mobiles. Ces miroirs pivotent de manière à laisser passer (pixel blanc) ou non (pixel noir) la lumière. De cette manière, la fréquence de fonctionnement des miroirs, qui peut atteindre 32 kHz, résulte en une latence inférieure à $40 \mu\text{s}$. En concédant une fréquence de fonctionnement plus faible et donc une latence supérieure, les miroirs peuvent pivoter plusieurs fois pendant un même cycle d'affichage, ce qui projette des niveaux de gris, proportionnels au temps écoulé avec le miroir passant ou bloquant pendant la durée du cycle.

L'utilisation d'écrans à tube cathodique diminue aussi la latence de l'affichage. En effet,

ces écrans n'effectuent pas de traitement sur l'image à afficher comme le font les écrans LCD actuels. La seule latence sur ce type d'affichage est donc celle qui est nécessaire au balayage de l'écran pour afficher chaque pixel. Les écrans à tube cathodique sont d'ailleurs utilisés pour mesurer la latence d'écrans LCD en comparant un affichage cloné sur les deux écrans (cf. chapitre 2). Cependant la technologie à tube cathodique a l'inconvénient d'être limitée à des écrans lourds et encombrants qui ne conviennent pas, par exemple, aux dispositifs mobiles.

Parmi les écrans disponibles dans le commerce, ceux qui gèrent la 3D ou qui sont vendus pour les joueurs de jeux vidéo ont des fréquences de rafraîchissement supérieures ou égales à 120 Hz. Utiliser de tels écrans réduit la latence due à la fréquence de l'affichage de 8.3 ms en moyenne pour un écran 60 Hz à 4.2 ms pour un écran 120 Hz. Ces écrans possèdent aussi des modes pour limiter la latence provenant du traitement des images avant l'affichage. Pour le tactile, seuls quelques appareils comme le dernier modèle d'iPad Pro sont capables d'afficher à 120 Hz [2].

Limites des approches matérielles

La latence des capteurs et de l'affichage provient en partie de la fréquence de fonctionnement de ces deux dispositifs. Comme montré ci-dessus, certaines technologies testées en laboratoire permettent des fréquences supérieures à 1000 Hz et induisent un retard de moins d'1 ms, imperceptible par les utilisateurs. Cependant, ce fonctionnement haute fréquence fait face à un double problème.

D'une part, la création de tels composants est techniquement complexe. Pour l'instant, les méthodes utilisées pour diminuer la latence le sont au détriment d'autres facteurs. Par exemple, le dispositif faible latence proposé par Ng et al. a 1 ms de retard seulement, mais ne permet pas d'afficher en couleur, elle reste limité à une petite surface d'interaction (24 cm × 16 cm), et elle nécessite de développer entièrement le code d'affichage sur puce programmable (FPGA). De plus la stratégie d'utilisation d'un projecteur, bien que viable pour un cadre d'expérimentation en laboratoire, semble difficilement applicable aux dispositifs mobiles.

D'autre part, pour que l'œil perçoive une continuité de l'animation dans la succession des images, un affichage à 24 Hz est suffisant. Le rafraîchissement des écrans actuels atteint le taux de 60 Hz pour lisser l'animation et éviter les effets de scintillement. Passer de 60 Hz aux 1000 Hz nécessaires pour produire une latence inférieure à la milliseconde demanderait une plus grande quantité de ressources énergétiques, avec un rapport coût-bénéfice discutable.

Une solution intermédiaire pour un meilleur compromis entre fréquence d'affichage et consommation d'énergie est implémentée sur le nouvel iPad [2]. La fréquence n'est plus fixe mais s'adapte selon les besoins en réactivité. Par exemple, lors de l'utilisation du stylet,

l'écran rafraîchit à 120 Hz pour diminuer la latence au maximum. Mais pour regarder un film, l'écran ne rafraîchit qu'à 48 Hz, une fréquence suffisante pour percevoir la continuité des images.

3.1.2 Réduction de la latence au niveau de l'application

La latence des capteurs et de l'affichage est optimisée par les fabricants de dispositifs tactiles. Les concepteurs d'applications, qui travaillent au niveau du logiciel, doivent donc composer avec une latence fixe imposée par le matériel. Dans cette section, nous explorons trois stratégies pour gérer la latence à un niveau logiciel. La première consiste à amener l'utilisateur à percevoir la latence comme une caractéristique souhaitée de l'application. La seconde est de compenser directement les conséquences de la latence spécifiques à une tâche. La troisième est de compenser la latence par des algorithmes de prédiction.

Intégrer la latence comme un élément de design de l'interaction

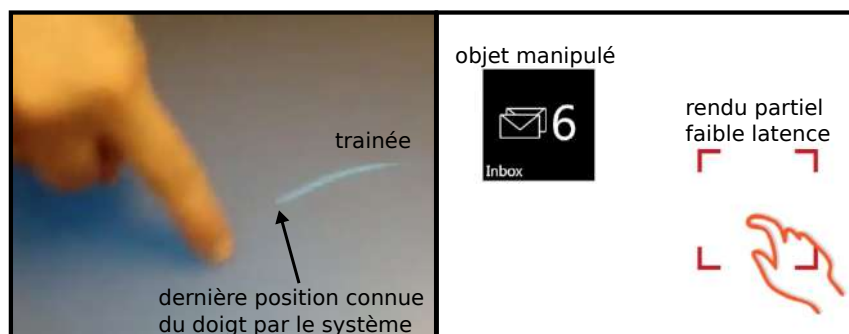


FIGURE 3.3 – Deux exemples où la latence est intégrée dans le design de l'interaction.

A gauche : *Ripples* de Wigdor et al. [150].

A droite : Un design multicouche proposé par Ng et al. [106]. Lorsque le doigt se déplace, un rendu partiel “fil de fer” de l'objet manipulé est affiché avec une faible latence pour avoir un premier retour visuel, même s'il est incomplet.

Wigdor et al. (2009) ajoutent un élément visuel à l'interaction pour donner l'impression à l'utilisateur que la présence de la latence est intentionnelle [150]. En affichant une traînée visuelle qui suit la trajectoire du doigt, les concepteurs intègrent le retard du curseur comme un élément du design (Figure 3.3).

Pour limiter l'effet de la latence de l'application, Ng et al. (2012) proposent de combiner deux couches de retour visuel [106]. La première couche fournit un rendu partiel pour fonctionner à faible latence. La seconde couche comporte le rendu final de l'application avec toutes ses fonctionnalités, mais donc avec plus de latence (Figure 3.3).

Compenser les conséquences de la latence spécifiquement à la tâche réalisée

Pour une tâche d'acquisition de cible à la souris, Ivkovic et al. (2015) constatent qu'avec de la latence, les utilisateurs tendent à pointer plus loin que la cible [65]. Les auteurs proposent donc un mécanisme de cibles "collantes" : le gain de la souris (le ratio entre le mouvement de la souris et le mouvement du curseur) est réduit lorsque le curseur passe sur une cible. C'est une reproduction du principe du pointage sémantique proposé par Blanch et al. [25], spécifiquement dédié à la compensation de la latence. Cette technique évite le dépassement de la cible lors du pointage. Pour une tâche de suivi de cible avec de la latence, la plus grosse perte de performance advient au moment d'un changement de direction de la cible. Ivkovic et al. proposent donc un système d'assistance du suivi où le curseur est attiré dans la direction du mouvement de la cible. Grâce à une étude utilisateur, les auteurs montrent que l'ajustement adéquat de ces deux fonctions (gain diminué sur la cible et curseur attiré vers les cibles mouvantes) en fonction de la latence réduit les effets négatifs de la latence sur la performance des utilisateurs pour des tâches de pointage et de suivi de cible. En revanche, cette technique est limitée aux applications où les cibles potentielles sont connues.

Compenser la latence par une prédiction

Pour contrecarrer un retard, il semble logique d'essayer de prendre de l'avance ; par exemple en prédisant les actions futures de l'utilisateur.

Avec la souris, différentes techniques de prédiction ont été utilisées pour diminuer le temps nécessaire pour pointer une cible parmi plusieurs cibles possibles. Ces techniques tentent de prédire l'intention finale de l'utilisateur et peuvent aussi servir à diminuer la latence d'un système. Asano et al. (2005) utilisent une prédiction basée sur l'amplitude du pic de vitesse du curseur [16]. Le modèle utilisé suppose que la distance que va parcourir l'utilisateur est une fonction affine de l'amplitude du pic de vitesse. Les paramètres de cette fonction affine sont appris pour chaque utilisateur pendant une session de calibration. L'interface détecte le pic de vitesse du mouvement et fait sauter le curseur à la distance prédite par la fonction affine. Une étude utilisateurs de cette technique d'interaction montre que la prédiction réduit le temps de mouvement pour les cibles éloignées (>800 pixels) mais n'a pas d'effet, voire accroit, le temps de mouvement pour les cibles plus proches. Des articles plus récents proposent aussi des modèles de prédiction plus complexes : des lois cinématiques [84], des réseaux de neurones et des filtres de Kalman [24] ou du *template matching* [109]. Dans ces propositions, des modèles complexes sont choisis de manière à pouvoir prédire plus longtemps en avance. La prédiction n'étant jamais exacte, ces techniques d'interaction se contentent donc de proposer la cible prédite à l'utilisateur et attendent ensuite une confirmation ou un ajustement de sa part.

Dans le cas de l'interaction tactile, Xia et al. (2014) utilisent une prédiction pour

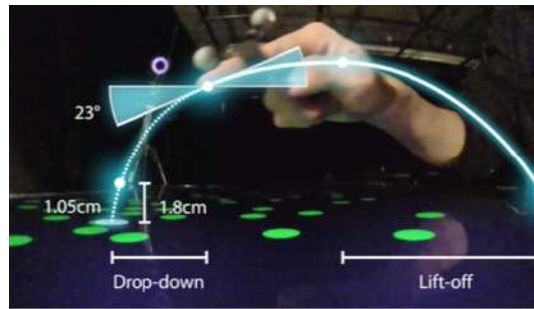


FIGURE 3.4 – Utilisation d'une prédiction pour compenser la latence du premier contact [155]. Xia et al. prédisent la position d'arrivée du doigt (en pointillé) sur l'écran grâce à un modèle parabolique basé sur l'angle du mouvement avec une précision de 1.05 cm.

réduire la latence du premier contact [155]. La position du doigt est suivie par un système de suivi optique et il s'agit de prédire, à partir de la trajectoire du doigt en l'air, le moment et la position où celui-ci va se poser sur l'écran (Figure 3.4). La prédiction est faite en extrapolant une parabole sur la trajectoire du doigt et résulte en une réponse immédiate lorsque le doigt touche finalement l'écran. Ahmad et al. (2016) ont aussi étudié l'effet d'une prédiction pour le toucher d'un écran tactile situé dans l'habitacle d'une voiture [11]. L'écran tactile affiche plusieurs cibles. La main de l'utilisateur est suivie grâce à un dispositif *Leap Motion*, et la prédiction suppose qu'à tout instant, la cible visée par l'utilisateur est celle qui est la plus proche de sa main. Quand la probabilité que l'utilisateur sélectionne une cible donnée dépasse un certain seuil, cette cible est automatiquement activée par le système, souvent même sans que le doigt ait besoin d'aller jusqu'au contact avec l'écran. Les auteurs montrent que l'utilisation de la prédiction réduit le temps de sélection et induit une charge de travail subjective moins importante pour les utilisateurs. Ces résultats sont encore plus significatifs lorsque le système est utilisé dans un contexte où la conduite est soumise à des perturbations (mouvements ou vibrations du véhicule) qui rendent difficile de toucher l'écran avec précision.

Ces travaux reposent sur la prédiction du premier contact de l'utilisateur sur l'écran. Lorsque l'utilisateur manipule un objet à l'écran, le problème est plus complexe. Il ne s'agit plus de prédire seulement un point d'arrivée mais de pouvoir prédire en continu la trajectoire future du doigt pour afficher l'objet virtuel au plus près de la position du doigt. Ce type de prédiction est particulièrement adapté au domaine de la réalité virtuelle ou augmentée où une prédiction continue est aussi nécessaire pour gérer le déplacement du contenu virtuel lors des mouvements de l'utilisateur.

En réalité augmentée, Knibbe et al. (2015) montrent qu'un filtre de Kalman avec un modèle balistique permet de prédire le mouvement de balles lancées en l'air et d'améliorer la projection d'une augmentation virtuelle sur ces balles [78] (Figure 3.5). Miyafuji et al. (2016) améliorent ce modèle pour gérer le rebond de la balle sur le sol [99]. Cependant, cette technique de prédiction ne s'applique qu'aux objets dont le mouvement est défini

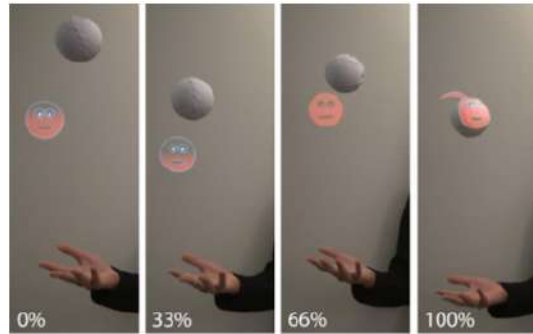


FIGURE 3.5 – Utilisation d'une prédiction pour compenser la latence sur un système de réalité augmentée. Knibbe et al. améliorèrent la projection sur des balles de jonglage, détériorée par la latence (à gauche), grâce à l'utilisation d'une prédiction (longueur de prédiction croissante de gauche à droite) [78].

par une loi connue, comme une parabole pour la chute libre, ce qui la rend difficilement applicable aux mouvements d'un utilisateur dont il est difficile de prévoir l'intention.

Le domaine de la réalité virtuelle donne des pistes à suivre pour un bon modèle de prédiction continue. LaViola (2003) montre qu'un modèle de prédiction à 50 ms ou 100 ms de la position de la tête d'un utilisateur se basant sur une simple régression linéaire fonctionne plus rapidement tout en étant aussi précis que des modèles plus complexes basés sur des filtres de Kalman [87]. Bucker et al. (2012) utilisent une prédiction avec un réseau de neurones pour compenser la latence d'un système de réalité virtuelle [30]. Leurs résultats indiquent que la prédiction permet de réduire la sensation de malaise parfois ressentie en réalité virtuelle. Des travaux plus récents de LaValle et al. (2014) examinent la méthode prédictive utilisée pour le casque de réalité virtuelle *Oculus rift* [86]. Grâce à un échantillonnage du gyroscope très rapide (1 kHz), un modèle d'ordre 2 avec une accélération supposée constante permet de prédire la position de la tête de manière précise 20 ms ou 40 ms en avance. Les auteurs insistent sur les exigences nécessaires pour avoir une bonne prédiction : des données capteur denses et précises et une longueur de prédiction limitée.

Pour la latence lors du glissement continu du doigt sur l'écran, de nombreux brevets et rapports techniques ont été déposés récemment sur l'utilisation de prédictions pour compenser le retard [31, 91, 100, 124, 141]. Cependant les informations fournies par ces brevets sont souvent limitées et on ne connaît pas les modèles de prédiction utilisés. Les tablettes produites par Apple [140] et Microsoft [8] se servent aussi d'algorithmes de prédiction pour compenser la latence du toucher mais il n'y a pas de description de ces algorithmes ni des effets réels qu'ont ces prédictions sur les utilisateurs.

Les différentes prédictions exposées ci-dessus illustrent que le modèle choisi dépend fortement du cadre dans lequel la prédiction est utilisée : pour une tâche de pointage, le modèle peut utiliser une connaissance de l'interface et de la position des différentes cibles de l'utilisateur [11] ou en être ignorant. Le modèle peut avoir été entraîné sur le comportement d'un utilisateur spécifique [16] ou au contraire être général. Le modèle peut être créé pour

un type de mouvement particulier [78] ou non. Enfin, la qualité de la prédiction dépend de la longueur de prédiction souhaitée. En effet, prédire une trajectoire future est sujet à erreurs, et plus la prédiction est longue, plus les erreurs sont amplifiées. Il faut alors trouver un compromis où la prédiction réduit les effets de la latence sans toutefois introduire trop d'erreur. Des données de bonne qualité, fréquentes et précises, sont alors souhaitables car elles améliorent le modèle de prédiction et permettent de prédire plus loin pour une erreur de prédiction équivalente.

3.1.3 Optimisation du pipeline

Une fois la latence optimisée au niveau matériel et logiciel, la latence bout-à-bout peut être encore réduite par une amélioration de la communication entre ces deux couches.

Synchroniser le calcul GPU et l'application avec l'affichage

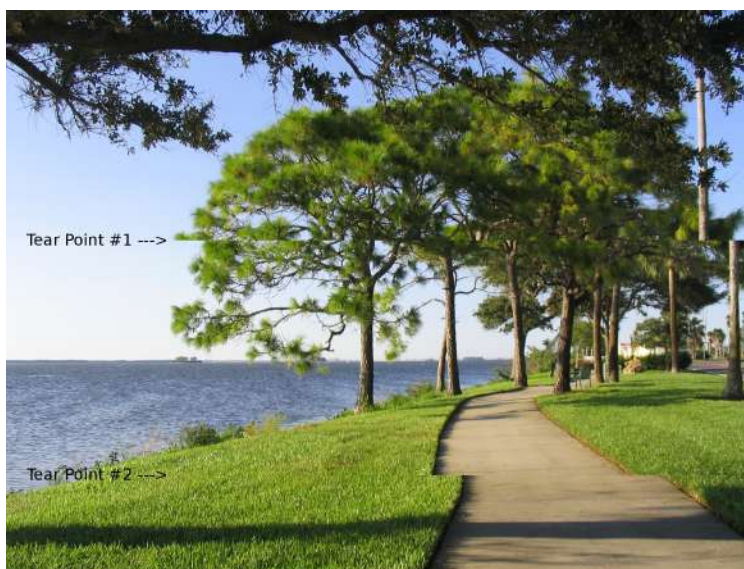


FIGURE 3.6 – Exemple d'un affichage où l'image à afficher a changé deux fois au cours du rafraîchissement de l'écran. On observe alors deux zones où l'image est "déchirée" (*tearing*)

Pour éviter les problèmes de déchirure de l'image dus à l'affichage simultané d'images successives à l'écran (*screen tearing*, Figure 3.6), la carte graphique utilise un système de double mémoire (*double buffering*). Une mémoire stocke l'image actuellement affichée à l'écran, tandis que l'autre est disponible en écriture pour le calcul de l'image suivante. Lorsque la mémoire de travail est prête à être affichée, le système attend le signal de synchronisation verticale (VSync) de l'écran, c'est à dire le moment où l'écran va débuter un nouveau cycle d'affichage, pour faire l'échange des mémoires. La déchirure est évitée, mais l'attente du signal VSync alors qu'une image est prête à être affichée contribue à la latence.

Généralement, le temps que vont prendre l'application et la carte graphique pour traiter les données et préparer la nouvelle image est inconnu. Les données qui arrivent des capteurs sont donc traitées dès que possible, c'est-à-dire juste après l'échange du contenu des mémoires. Si le temps de calcul de la nouvelle image est négligeable, ce qui est un cas fréquent, alors l'attente de VSync est une source importante de latence.

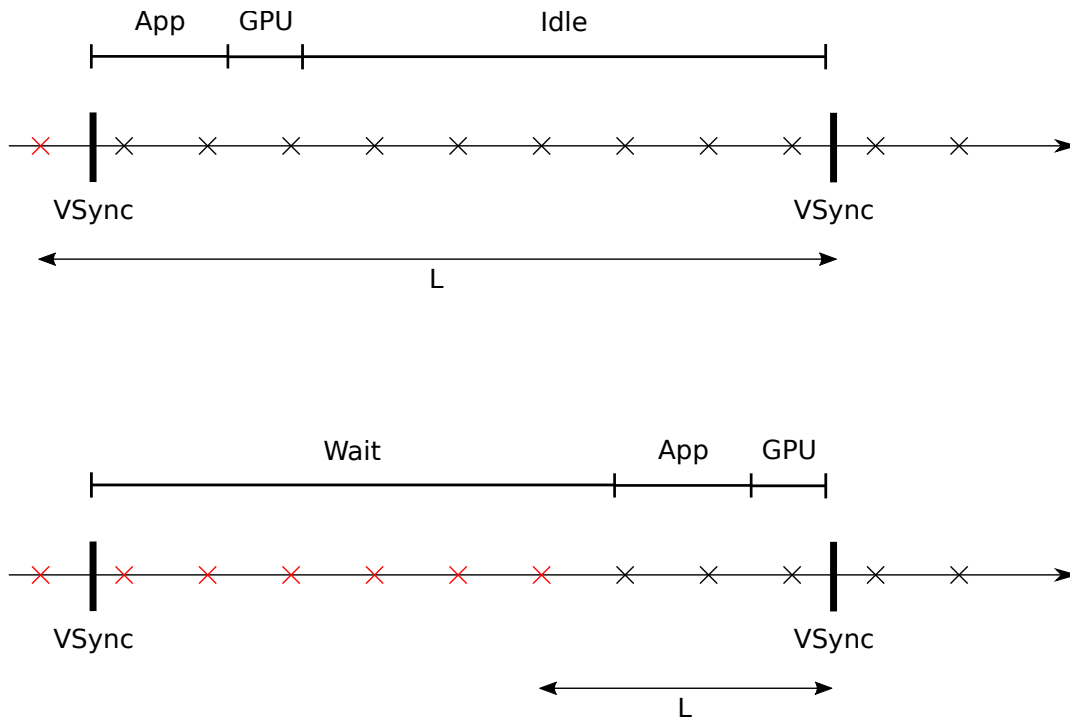


FIGURE 3.7 – Une optimisation possible de la latence : récupérer les données des capteurs les plus récentes possibles. Sur la plupart des systèmes, l'application s'exécute juste après que la dernière image ait été envoyée pour affichage (VSync). Les données de capteurs considérées (croix rouges) sont celles qui sont arrivées avant le précédent signal de synchronisation (schéma du haut). Si le moment du prochain signal de synchronisation et la durée que vont prendre l'application et le GPU pour générer une nouvelle image sont connus, le système peut attendre pour lancer ces opérations aussi tard que possible. Des données plus récentes sont utilisées et on réduit la latence L entre les données du capteur et l'envoi de l'image à l'écran (schéma du bas).

Si la durée de calcul du pipeline application-carte graphique est peu variable et connue, alors il est possible d'optimiser le temps d'attente avant la synchronisation verticale. Il s'agit de programmer le système pour qu'il attende aussi longtemps que possible les données des capteurs. Ainsi, il dispose des données les plus récentes au moment de calculer le prochain retour graphique. Le système garde juste assez de temps pour faire ce calcul avant le prochain signal VSync. De cette manière, les données les plus récentes sont récupérées et la latence bout-à-bout du système est réduite (Figure 3.7). Cette méthode se retrouve dans les travaux de la société Valve [145] (2015) pour la réalité virtuelle, et, dès 1997, dans les travaux de Jacobs et al., pour minimiser la latence relative entre plusieurs sources de

données pour des systèmes de réalité augmentée [66].

Vers une disparition de la synchronisation verticale ?

Les problèmes de latence introduits par la synchronisation verticale ont conduit Nvidia à proposer une technique de synchronisation adaptative : G-Sync [4]. Dans G-Sync, ce n'est plus le GPU qui s'adapte au rafraîchissement de l'écran mais l'écran qui s'adapte à la fréquence du GPU. Cette technique permet d'éviter les effets de déchirure de l'image en évitant la perte de temps due à l'attente de la synchronisation verticale.

Dans des travaux récents, Yun et al. (2017) discutent l'utilité des mécanismes de synchronisation avec les dispositifs actuels [156]. Les auteurs notent que beaucoup d'applications utilisant le tactile ne modifient qu'une partie locale de l'écran à chaque nouvelle image (par exemple, le dessin, ou le déplacement d'un objet sous le doigt) et sont donc peu sensibles au problème de déchirure de l'image. De plus, le décalage du retour visuel créé par la latence due aux techniques utilisées pour contrer l'effet de déchirure peut être plus important que le décalage de la déchirure elle-même. Les auteurs proposent alors un design asynchrone (qui inclut une technique similaire à celle qui est utilisée par Nvidia pour G-Sync) qui permet de réduire de 32 ms en moyenne la latence de plusieurs applications.

3.1.4 Bilan

La réduction de la latence au toucher par seule voie matérielle est possible, mais très coûteuse en terme de dépense énergétique quand il s'agit d'atteindre des niveaux de latence de l'ordre de la milliseconde. Au niveau logiciel, intégrer la latence comme une caractéristique de l'application est un moyen de faire accepter la latence à l'utilisateur et l'utilisation d'une prédiction pourrait diminuer significativement ses effets négatifs. Cependant, pour la latence au toucher, la plupart des modèles de prédiction utilisés proviennent de brevets et aucune étude ne vient évaluer ces modèles pour quantifier clairement l'effet bénéfique de la prédiction et ses limites. Le domaine de la réalité virtuelle pour lequel l'effet d'une prédiction a été évalué laisse penser que des prédictions trop longues amènent trop d'instabilités.

La conclusion que nous pouvons tirer des travaux présentés dans cette section est que la meilleure diminution de la latence devrait s'obtenir avec des efforts parallèles au niveau matériel et logiciel, avec les étapes suivantes : 1 - Une réduction de la latence au niveau matériel dans des proportions raisonnables et peu coûteuse énergétiquement ; 2 - Une optimisation de la synchronisation entre les capteurs, l'application et l'affichage ; et 3 - Une compensation de la latence restante avec une prédiction des comportements de l'utilisateur. La 3ème étape ne peut être bénéfique que si la longueur de prédiction est courte, ce qui est possible si la majeure partie du retard a déjà été supprimée par l'optimisation matérielle et par la synchronisation entre les couches matérielle et logicielle.

Dans cette thèse, nous avons suivi cette approche. Nous avons construit un dispositif tactile capable d'atteindre 25 ms de latence tout en ayant un affichage en couleur sur une surface de 24 pouces (61 cm) de diagonale. Ce dispositif est présenté au chapitre 4. Les 25 ms restantes sont ensuite compensées par une prédiction dont nous avons évalué les effets sur la performance utilisateur lors d'une étude présentée au chapitre 5.

3.2 Adaptation humaine face à un retard du retour visuel

A l'heure actuelle, les avancées matérielles et logicielles ne produisent pas des dispositifs à la latence suffisamment faible pour que celle-ci n'ait plus de conséquence sur les utilisateurs. Les possesseurs d'écrans tactiles se servent alors de dispositifs imparfaits et font face aux perturbations induites par la latence. Ces perturbations ayant des effets négatifs sur leur performance (cf. chapitre 2), on peut considérer la latence comme une perturbation visuomotrice. Or, le cerveau humain est capable de s'adapter à certaines perturbations visuomotrices et de compenser leurs effets négatifs [127]. Est-ce aussi le cas pour la latence ?

Les capacités adaptatives sensorimotrices des utilisateurs font l'objet de nombreux travaux dans la recherche sur les apprentissages sensorimoteurs et le contrôle moteur en général. Dans cette section, nous introduisons tout d'abord différents concepts généraux qui correspondent à des processus d'apprentissage sensorimoteur en action, notamment lorsqu'une personne fait face à une perturbation visuomotrice : l'adaptation, l'apprentissage de compétence, la rétention et le transfert d'apprentissage. Nous nous intéressons ensuite aux travaux antérieurs ayant traité plus particulièrement de l'adaptation au retard du retour visuel dont nous nous inspirerons dans nos études utilisateurs pour comprendre la compensation de la latence par les utilisateurs.

3.2.1 Définitions et concepts

Adaptation à une perturbation

Notre cerveau possède une représentation de la manière dont les commandes motrices qu'il envoie vont influencer sur les états de notre corps. Selon certains auteurs, cette représentation serait basée sur un modèle interne de la dynamique inverse entre le résultat désiré et la commande motrice à exécuter [154]. Quelque soit la nature de la représentation qui le supporte, le lien entre les commandes motrices et les états de notre corps évolue au cours du temps avec les modifications anatomiques qui adviennent au cours de la croissance et/ou des modifications de la masse musculaire. Cette représentation doit donc être mise à jour en continu en s'appuyant sur les retours sensoriels du résultat des différentes commandes qui réussissent ou échouent [127]. D'après Krakauer et Mazzoni (2011), l'adaptation sensorimotrice est une forme d'apprentissage, caractérisée par "une réduction graduelle de l'erreur causée par une perturbation via une recalibration sensorimotrice progressive" [80]¹. Pour comprendre les mécanismes sous-jacents à cette adaptation, de nombreuses études ont été menées qui analysent les comportements humains quand une perturbation contrôlée est introduite. Il existe alors deux paradigmes principaux illustrés en Figure 3.8. Le premier est d'introduire la perturbation au niveau moteur comme par exemple une tâche de

1. "incremental reduction in sensory prediction errors caused by a perturbation in the world through trial-by-trial modification of a motor-to-sensory mapping"

déplacement d'un bras mécanique avec une force supplémentaire qui vient perturber le mouvement [126]. Le second est de perturber le retour sensoriel, par exemple en déplaçant la position d'un retour visuel. Dans le cadre de notre travail, c'est ce second paradigme qui correspond au type de perturbation introduit par la latence et que nous explorons donc ci-dessous.

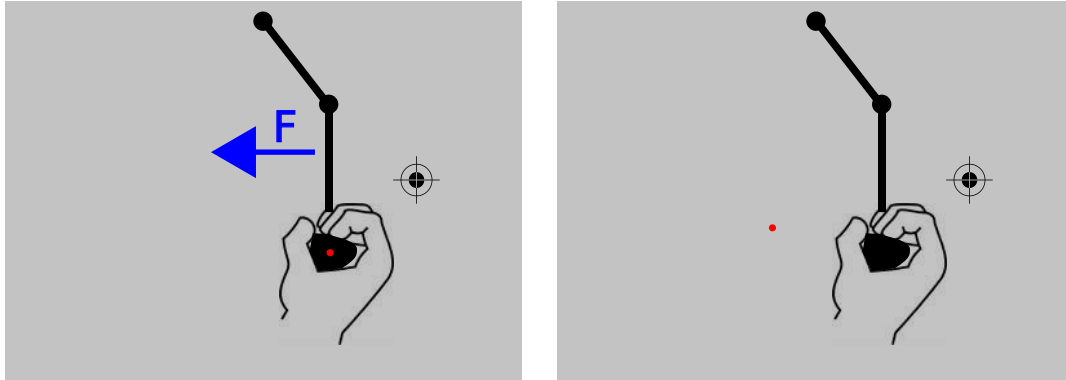


FIGURE 3.8 – Deux paradigmes expérimentaux pour étudier les recalibrations visuomotrices. Dans cet exemple, le participant doit amener un curseur représentant sa main (point rouge) sur une cible en déplaçant la poignée d'un bras robotisé. Le participant ne voit ni sa main ni le bras qui sont cachés par l'écran (l'écran est représenté par le fond gris). Le premier paradigme (à gauche) est d'introduire une perturbation motrice via une force exercée par le bras robotisé. Le second paradigme (à droite) est de déplacer le retour visuel.

Dès 1867, Helmholtz réalise une étude sur l'adaptation visuomotrice [146]. L'auteur teste sur lui-même des lunettes avec des prismes qui décalent le champ visuel vers la gauche. Lorsque l'auteur pointe vers une cible, ils pointe alors trop loin vers la gauche, mais au fur et à mesure de la répétition du geste, l'auteur finit par compenser le décalage visuel pour pointer vers la cible. Lorsqu'on retire les lunettes, et que la vision n'est plus perturbée, l'auteur remarque qu'il pointe alors vers la droite de la cible. Ces effets opposés qui subsistent lorsque la perturbation est brutalement retirée sont appelés *after-effects*. Les études sur la distorsion de la vision ont été complétées avec le temps en montrant que l'adaptation existe pour différentes tâches (e.g. jeter un objet vers une cible [94]) ou pour différents types de transformations du champ visuel [18].

Avec l'utilisation de la souris et des ordinateurs, un nouveau paradigme s'est développé pour étudier les recalibrations visuomotrices. Les sujets contrôlent un curseur à l'écran avec un appareil (souris, joystick...) et doivent atteindre une cible. Le mouvement habituel du curseur peut alors être modifié par voie logicielle pour introduire une perturbation des relations visuomotrices. Krakauer et al. (2000), entre autres auteurs, montrent que les sujets de leur étude peuvent s'adapter à un changement de gain ou à une perturbation de la direction du mouvement [82].

Le phénomène d'adaptation est distinct de celui d'une simple compensation. Mazzoni et Krakauer (2006) montrent qu'une stratégie de compensation explicite peut ramener l'er-

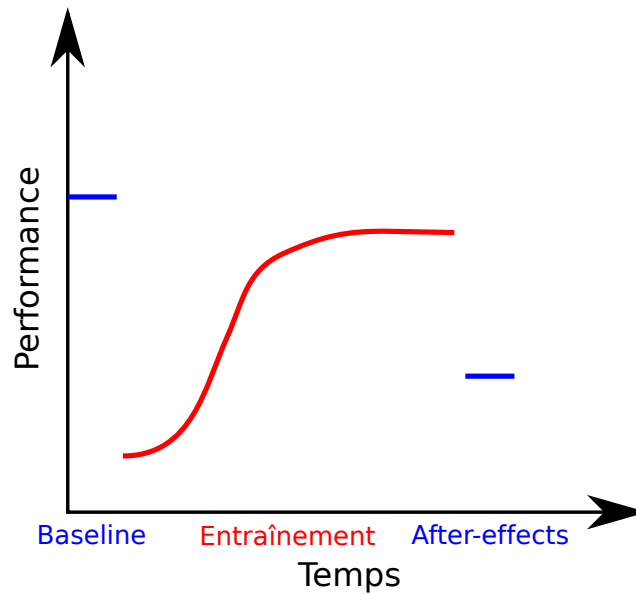


FIGURE 3.9 – Modèle d’un protocole expérimental pour tester l’adaptation à une perturbation. On mesure d’abord la performance sur la tâche non perturbée (bleu, *baseline*). Ensuite on introduit la perturbation (rouge) et on entraîne le participant sur la tâche. Finalement on remesure sa performance sur la tâche non perturbée (test d’*after-effects*). Si la performance est plus faible que la *baseline* c’est qu’il y a eu recalibration et donc adaptation.

reur d’une tâche à zéro sans pour autant qu’il y ait adaptation [95]. Dans leur étude, les participants doivent partir d’une position centrale et pointer avec la main pour amener un curseur sur des cibles disposées en cercle et écartées de 45 degrés. Une perturbation est introduite : le curseur se déplace dans une direction orientée de 45 degrés par rapport à la direction de la main. Les auteurs observent alors une phase d’adaptation classique (Figure 3.9, “entraînement”) avec d’abord une grosse erreur directionnelle corrigée progressivement, puis des *after-effects* lorsque la perturbation est retirée. Les auteurs étudient une seconde condition dans laquelle, lorsque la perturbation est introduite, les expérimentateurs font remarquer explicitement au sujet qu’il lui suffit de pointer vers la cible voisine pour compenser la perturbation. Dans cette condition l’erreur est réduite à zéro instantanément grâce à l’utilisation de la stratégie. Cependant, au fur et à mesure des essais, l’erreur grandit au même rythme que l’erreur s’est réduite pour ceux qui n’exploitent pas cette stratégie. Les auteurs discutent ce phénomène et concluent que l’utilisation de la stratégie proposée permet de compenser la perturbation, en réduisant l’erreur à zéro, mais ne permet pas de corriger la contradiction entre la proprioception et la vision des sujets. La contradiction est corrigée lors d’une phase d’adaptation qui survient malgré l’augmentation de l’erreur qu’elle engendre avec l’utilisation parallèle de la stratégie.

Une compensation des effets d’une perturbation visuomotrice est donc possible sans qu’il y ait adaptation, c’est-à-dire sans recalibration des mécanismes visuomoteurs. La

preuve d'une adaptation peut être faite en montrant les effets de cette recalibration lorsque la perturbation est retirée (*after-effects*, Figure 3.9) ou sur une tâche différente (transfert d'adaptation, cf. ci-dessous).

Apprentissage d'une compétence vs adaptation

La littérature sur l'apprentissage sensorimoteur distingue en général l'apprentissage de compétences sensorimotrices de l'adaptation sensorimotrice. L'apprentissage d'une compétence sensorimotrice est une progression des stratégies motrices pour réaliser cette compétence de la façon la plus stable, la plus régulière ou la plus efficace possible, avec une certaine optimalité. L'adaptation (ou *Error-based learning* d'après Wolpert et al. (2011) [152]) n'intervient que lorsque le contexte d'apprentissage de cette compétence est perturbé et que la perturbation introduit de l'erreur. L'adaptation permet alors de réduire l'erreur dans ce nouveau contexte.

Il existe différentes perspectives sur la notion d'optimalité dans l'apprentissage d'une compétence motrice. Krakauer et Mazzoni (2011) définissent, par exemple, l'apprentissage de compétences sur une tâche motrice comme un décalage de la fonction de compromis précision-vitesse sur cette tâche [80]. Ils suggèrent ainsi que l'apprentissage de compétences motrices permettrait de réaliser la tâche de manière plus rapide et plus précise. Cependant l'apprentissage est aussi limité par l'effort que l'utilisateur doit fournir pour réaliser la tâche. D'après les résultats de Nagengast et al. (2009), il semblerait que la théorie de l'*Optimal feedback control* [139], qui optimise une fonction de coût associant effort et précision, puisse prédire les résultats d'un apprentissage de compétences sur la manipulation d'outils à la dynamique complexe [102]. D'autres résultats indiquent cependant que le cadre de l'inférence Bayésienne serait plus approprié que celui du contrôle optimal pour comprendre les comportements moteurs [53].

Rétention de l'apprentissage

L'adaptation à une perturbation et l'apprentissage de compétences motrices ne s'effectuent pas sur les mêmes échelles temporelles : l'apprentissage de compétences est en général bien plus long que l'adaptation [80] et peut s'étaler sur plusieurs jours. Walker et al. (2002) montrent l'effet bénéfique du sommeil sur la progression de la réalisation d'une tâche motrice (taper au clavier une séquence de chiffres) [148]. Leur étude indique qu'après une période d'éveil de 12 h, la performance reste inchangée alors qu'après une période de 12 h incluant une nuit de sommeil la performance est significativement améliorée.

Doyon et al. (2009) [49] réalisent une étude similaire à celle de Walker et al. mais en ajoutant une tâche d'adaptation visuomotrice (pointage avec un joystick inversé, c'est-à-dire que le mouvement du curseur va dans le sens opposé à celui du joystick) en plus de la tâche de séquence de frappe au clavier. Comme Walker et al., pour la tâche de frappe, les

auteurs trouvent une influence positive du sommeil sur la performance. En revanche cela n'est pas le cas pour l'adaptation visuomotrice qui ne suggère aucun gain quelle que soit la condition. Cela indique que l'adaptation et l'apprentissage de compétences seraient deux phénomènes distincts.

Alors que l'apprentissage de compétences motrices a une très bonne rétention [10], l'adaptation peut s'oublier facilement. Par exemple, Criscimagna et Shadmehr (2008) montrent que l'adaptation à une perturbation mécanique s'oublie progressivement avec le temps [39]. Les sujets sont re-testés entre 2 min et 24 h après la session d'adaptation. Au bout de 24 h, les sujets ne compensent plus que 25% de la perturbation contre plus de 60% à la fin de l'adaptation. Une disparition de l'adaptation peut aussi être observée quand on retire le retour visuel, quand on crée l'illusion d'une bonne performance (paradigme *error-clamp* [39]) ou qu'on effectue une phase de désadaptation (ou *washout*, c'est-à-dire refaire la tâche sans la perturbation) [77].

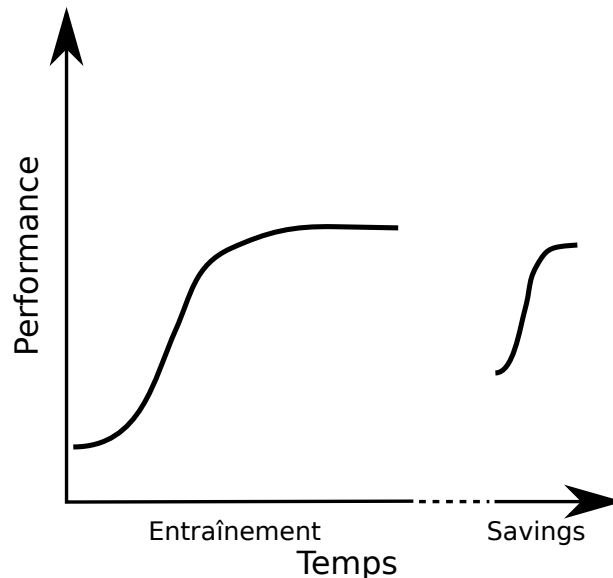


FIGURE 3.10 – Modèle d'un protocole expérimental pour tester les *savings*. On entraîne un participant sur une tâche puis on le teste à nouveau sur cette même tâche après avoir laissé s'écouler un certain laps de temps. Si on constate que la perte de performance due à ce laps de temps est vite compensée par un réapprentissage rapide, on observe des *savings*

Cependant, si la perturbation est réintroduite, il est possible d'observer une nouvelle adaptation plus rapide que la précédente, un phénomène nommé *savings* [153] (Figure 3.10). Ces *savings* dépendent de la manière dont s'est déroulée la première phase d'adaptation. Par exemple, dans le cas où celle-ci a été graduelle (perturbation introduite petit à petit), alors, après une phase de désadaptation, on n'observe pas de *savings* [120]. De la même manière, une disparition de l'adaptation due à une suppression du retour visuel ou l'illusion d'une bonne performance ne mène à aucun saving [77].

Transfert d'apprentissage

Quand il y a de l'apprentissage sur une tâche (adaptation ou compétences), cet apprentissage peut avoir des conséquences sur d'autres tâches qui indiquent un processus de généralisation. Les conséquences négatives sont des interférences et les conséquences positives du transfert d'apprentissage [81] (Figure 3.11).

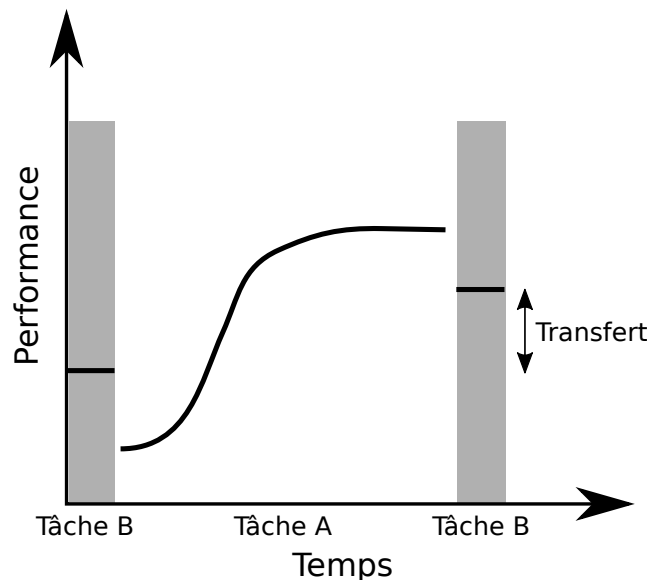


FIGURE 3.11 – Modèle d'un protocole expérimental pour tester le transfert d'apprentissage et les interférences. Après une première évaluation de la performance sur la tâche de transfert B, le participant est entraîné sur la tâche A. Après apprentissage, le participant repasse la tâche B. Le transfert est mesuré en comparant la performance sur la tâche B avant et après apprentissage. Dans cet exemple la performance est meilleure ce qui démontre un transfert d'apprentissage de la tâche A vers la tâche B.

Le transfert peut servir à quantifier la similarité, à un niveau sensorimoteur, entre deux tâches. Kozak et al. (1993) observent qu'un entraînement à une tâche dans un environnement virtuel ne se transfère pas à la même tâche effectuée dans le monde physique [79]. Les auteurs concluent que la tâche réalisée en réalité virtuelle est trop différente de la tâche dans le monde physique à la fois à cause de la technique (ils évoquent le problème de la latence) et du design (leur tâche dans l'espace virtuel n'est pas exactement la même que celle du monde physique). Bérard et Rochet-Capellan (2015) se servent du transfert pour mesurer la similarité sensorimotrice entre une interaction numérique et une interaction physique, un transfert d'apprentissage important indiquant une forte similarité [22]. Cette similarité est alors interprétée comme une indication objective quantifiant le caractère "naturel" de l'interaction au niveau sensorimoteur.

La présence ou l'absence de transfert d'une tâche à une autre peut aussi apporter une meilleure compréhension des phénomènes sensorimoteurs impliqués dans la réalisation de la tâche. Par exemple, l'adaptation à des lunettes prismatiques pour une tâche de lancer

effectuée avec le bras droit ne se transfère pas au bras gauche [94]. L'adaptation ne se fait donc pas qu'au niveau du système visuel, auquel cas il y aurait dû y avoir transfert pour tout autre mouvement testé. Bedford (1993) montre qu'une adaptation à deux décalages différents par des lunettes à prismes se généralise sur des valeurs de perturbation situées entre les bornes d'entraînement, mais que l'extrapolation de l'adaptation à des valeurs plus extrêmes semble faible voire absente [18]. Pour une tâche de pointage en interaction indirecte avec une tablette, les résultats de Krakauer et al. (2000) révèlent que l'adaptation à une modification du gain (relation entre la distance parcourue sur la tablette et la distance parcourue par le curseur à l'écran) se transfère à différentes directions de pointage [82]. Par contre l'adaptation à une perturbation qui modifie la direction du curseur par une rotation dans le plan ne se généralise pas à des cibles situées dans des directions différentes. Ces résultats différents pour les deux types de transfert suggèrent donc que les notions d'erreur d'amplitude et d'erreur de direction ne sont pas traitées de la même manière par le cerveau.

3.2.2 Compensation, adaptation et latence

Les études citées précédemment focalisent sur l'adaptation à des perturbations visuomotrices spatiales alors que la latence introduit une perturbation visuomotrice temporelle. Nous allons voir que l'adaptation à la latence se déroule différemment. Dans cette section, nous présentons des recherches effectuées dans le domaine du contrôle moteur : d'abord sur l'utilisation du retour visuel puis une revue des travaux sur l'adaptation à un retard du retour visuel. A la fin de chaque paragraphe nous mettons les résultats du contrôle moteur en parallèle avec certains résultats obtenus en IHM détaillés au chapitre 2. Enfin, nous exposons la différence principale qui existe entre les deux domaines et qui limite l'applicabilité des résultats obtenus en contrôle moteur pour le cadre de la latence au toucher direct.

Utilisation du retour visuel

Certaines études menées sur l'adaptation à des perturbations visuomotrices nous permettent d'avoir une meilleure compréhension de l'importance du retour visuel dans différentes phases du mouvement. Prablanc et Martin (1992) montrent que les mouvements de pointage ne s'appuient pas uniquement sur des commandes motrices préprogrammées, mais prennent aussi en compte le retour visuel et son erreur par rapport à la cible [115]. En effet, lorsque la cible que les participants doivent atteindre est légèrement déplacée au moment de la saccade oculaire qui annonce le début du mouvement de la main, les auteurs observent une correction en ligne de la trajectoire vers la nouvelle position de la cible. Cela ne serait pas le cas si les participants ne s'appuyaient pas sur le retour visuel.

Saunders et Knill (2003) étudient si le retour visuel, dans une tâche de pointage, a une importance seulement en fin de mouvement ou durant toute la durée du mouvement [123].

Ils imposent une perturbation (le retour visuel est décalé comparé à la position de la main) aux participants à différents stades du mouvement (25% ou 50% de la distance à la cible parcourue). Les résultats montrent que des corrections pour contrer la perturbation sont visibles très tôt dans le mouvement et vont donc dans le sens d'un contrôle visuel continu depuis le début du mouvement.

D'un point de vue IHM, ces résultats montrent bien l'importance du retour visuel tout au long de la trajectoire. La latence, qui perturbe directement ce retour visuel, a donc une influence pendant tout le mouvement de l'utilisateur.

Le retard du retour visuel peut empêcher une autre adaptation

Le retard du retour visuel peut avoir une influence sur l'adaptation à d'autres perturbations. Held et al. (1966) constatent que l'adaptation à un déplacement spatial constant (*offset*) du retour visuel n'est plus possible quand le dispositif présente une latence supérieure à 0.3 s [58]. Honda et al. (2012) montrent que le retard du retour visuel influe également sur l'adaptation à une rotation de la trajectoire d'un curseur, mais que cette influence peut être réduite si les participants ont eu une habitude préalable au retard [63].

Pour l'IHM, de tels phénomènes impliqueraient que la présence de latence pourrait empêcher d'autres apprentissages. Nous verrons dans une de nos contributions (chapitre 5) que ce n'est pas le cas pour l'apprentissage d'une compétence de suivi de cible, qui reste le même qu'il y ait ou non de la latence.

Perception du retard du retour visuel lors du premier contact et adaptation

L'interaction au toucher peut se diviser en deux sous-actions, le premier contact du doigt sur l'écran et le mouvement continu du doigt qui se déplace sur l'écran. Nous avons vu dans le chapitre précédent que selon l'action, la latence peut être différente et que de plus, elle n'est pas perçue de la même manière. Pour l'adaptation à un retard du retour visuel, nous considérons donc séparément le premier contact et le mouvement continu.

La manière dont les participants jugent la simultanéité d'un retour visuel et d'une action qu'ils viennent de réaliser a été étudiée dans différents travaux. Kaaresoja et al. (2014) demandent à des participants de juger de la simultanéité entre l'appui sur un bouton tactile et l'allumage d'une diode [71]. Les résultats indiquent que pour un retard de 85 ms, 75% des essais sont jugés simultanés. Les participants ont donc un biais qui leur fait juger simultané un retour visuel qui arrive en réalité en retard. Ce biais est appelé asynchronie négative et s'observe aussi quand on demande à des participants de synchroniser un toucher avec un signal externe régulier (un flash ou un son). On remarque alors que le contact du toucher précède le signal de plusieurs dizaines de millisecondes [17].

Stetson et al. (2006) étudient le déplacement du point de simultanéité subjective (PSS,

point of subjective simultaneity) après une habitude au retard [134]. Les participants doivent appuyer sur une touche et un flash s'allume. Les participants jugent alors si le flash est arrivé après ou avant l'appui sur la touche. On considère le PSS comme le retard auquel les participants jugent le flash comme étant arrivé après l'appui sur 50% des essais. Sur un bloc où la majorité des retards présentés est autour de 35 ms, les auteurs observent un PSS un peu au-dessus de zéro. Dans un second bloc où les retards présentés sont autour de 135 ms, le PSS observé est alors 44 ms plus grand. Cela indique que face à un retard plus grand, les participants s'adaptent et recalibrent leur jugement d'ordre.

Sugano et al. (2012) montrent aussi que l'asynchronie négative observée sur une tâche de synchronisation d'un clic de souris au moment d'un flash lumineux est modifiée par une habitude au retard [135]. Sur un pre-test, l'asynchronie négative est évaluée à 54.8 ms. Les participants sont ensuite habitués à la présence d'un retard lors d'une phase où un flash apparaît 150 ms après les clics de souris. Un post-test sur la tâche de synchronisation révèle que l'asynchronie négative s'est décalée de 12.3 ms et est maintenant évaluée à 67.1 ms.

Ces études ne présentent pas une adaptation motrice au sens où nous l'avons définie puisque les phases d'habitude ne sont pas conçues pour permettre aux participants d'améliorer leur performance. Mais il s'agit tout de même d'une recalibration des relations visuomotrices après exposition au retard du retour visuel.

L'asynchronie négative décelée par les travaux du contrôle moteur pourrait expliquer un résultat de l'IHM sur la perception de la latence. Si les utilisateurs perçoivent un retour visuel retardé comme simultané avec leur action de toucher, il semble logique qu'ils soient peu sensibles à la latence du premier contact par rapport à la latence lors d'un mouvement continu [46].

Adaptation à un retard du retour visuel lors d'un déplacement continu

Cunningham et al. (2001) explorent l'adaptation au retard du curseur d'une souris [40]. Les participants de leur étude jouent à un jeu dans lequel ils doivent manipuler un avion horizontalement à l'écran en se servant d'une souris avec un gain de 1 (la distance parcourue par la souris est retranscrite exactement à l'écran). Il y a trois phases : un pre-test avec retard minimal (35 ms), une phase d'adaptation avec retard à 235 ms puis un post-test à retard minimal. Lors de l'introduction du retard le plus important, les participants perçoivent le délai et doutent de pouvoir réussir le jeu. Cependant, à la fin de la phase d'adaptation, la plupart des participants atteignent une performance similaire à celle qu'ils ont obtenue dans le pre-test, et certains déclarent même que les mouvements de la souris et de l'avion sont simultanés. Lors du post-test, les auteurs observent de forts *after-effects* qui confirment qu'il y a eu adaptation. Dans un deuxième article (2000), les auteurs font une étude similaire pour évaluer si les participants peuvent s'adapter au retard du retour visuel d'un simulateur de conduite [41]. Trois retards sont étudiés : 130 ms, 230 ms, 430 ms. L'adaptation n'est visible que pour 230 ms. Les auteurs supposent que le retard de 130 ms

est trop similaire à l'absence de retard pour observer de forts *after-effects*. Pour 430 ms, l'absence d'adaptation est attribuée au temps accordé aux participants pour l'entraînement qui ne serait pas suffisant pour permettre une adaptation à un retard aussi élevé. Dans un troisième article (2001), les auteurs montrent aussi que l'adaptation à un retard n'est pas spécifique à la conduite sur une route donnée mais peut se transférer à d'autres routes [42].

De la Malla et al. (2012) étudient l'influence du retard du retour visuel sur une tâche d'interception de cible. Leur première étude [43] se concentre sur l'importance du retour visuel pour s'adapter à un retard entre le mouvement de la main, et celui d'un curseur à l'écran. Les résultats montrent que la performance pour la tâche d'interception de cible avec du retard est similaire lorsque le curseur est visible durant tout le mouvement ou seulement lorsque la main atteint la cible. Cela indique que le retour visuel est important surtout dans la dernière partie du mouvement quand il s'agit de faire face à un retard. Lors d'une seconde étude [44], les auteurs montrent que l'adaptation au retard du retour visuel ne semble pas se transférer entre différentes tâches sensorimotrices comme l'interception et le suivi de cible.

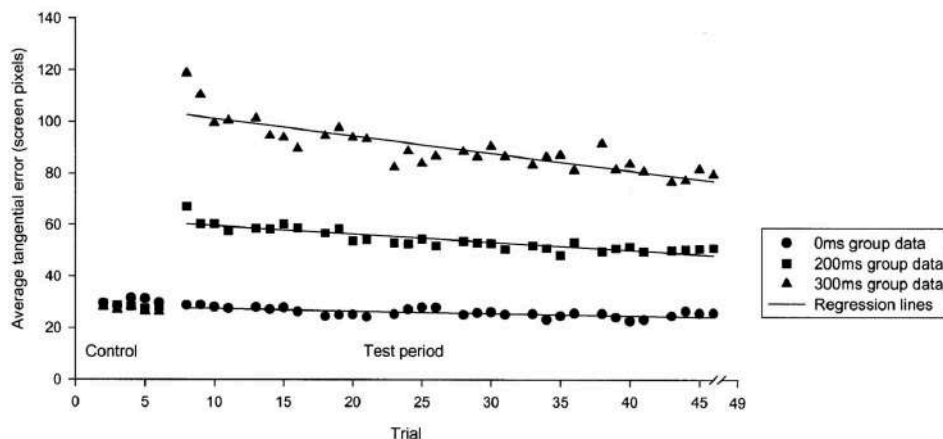


FIGURE 3.12 – Résultats de Foulkes et Miall sur l'adaptation à la latence pour une tâche de suivi de cible avec un joystick [52]. Avec l'entraînement, la performance des participants effectuant la tâche de suivi de cible avec un retard du retour visuel se rapproche de la performance du groupe effectuant la tâche sans retard.

Vercher et Gauthier (1992) considèrent la compensation du retard pour une tâche de suivi de cible sur une trajectoire prévisible, en interaction indirecte [144]. Les auteurs mettent en évidence que les participants de l'étude peuvent compenser un retard du retour visuel tant que celui-ci ne dépasse pas un seuil autour de 200-250 ms. Foulkes et Miall (2000) [52] s'intéressent au cas où la trajectoire est imprévisible. Les effets de trois retards sont observés : 0 ms, 200 ms et 300 ms. Les participants effectuent du suivi de cible à l'aide d'un joystick avec des essais étalés sur 2 à 3 jours. Les performances des groupes effectuant la tâche avec du retard se rapprochent petit à petit de la performance du groupe réalisant la tâche sans retard (Figure 3.12). Cependant, la vitesse de ce progrès est très

lente comparée aux résultats d'études sur l'adaptation à des perturbations visuomotrices spatiales. Miall et Jackson (2006) [96] prolongent ces travaux avec une étude similaire mais qui s'étend cette fois sur 5 jours avec un seul groupe test effectuant le suivi de cible avec 300 ms de retard. Sur certains essais, le retard est retiré pour observer la présence ou non d'*after-effects* qui pourraient être la preuve d'une adaptation. Les résultats témoignent encore d'une réduction de l'écart entre le groupe test et le groupe contrôle (qui fait la tâche sans retard) au fur et à mesure des sessions, avec des *after-effects* présents dès le premier jour. Cependant, la présence de ces *after-effects* pourrait être seulement due au fait qu'avec le retard les participants ralentissent leur mouvements. Dans ce cas, les *after-effects* ne seraient dus qu'à une stratégie de compensation de l'erreur et non à une recalibration sensorimotrice. L'adaptation est donc incertaine.

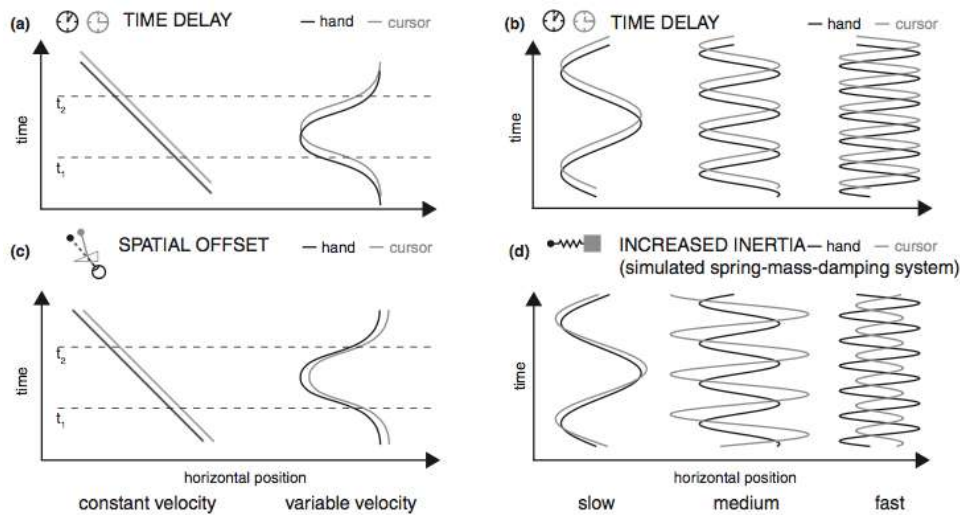


FIGURE 3.13 – Effets du retard sur la trajectoire du retour visuel similaires aux effets d'un décalage spatial ou d'inertie. Pour Rohde et Ernst, cette ambiguïté pourrait empêcher l'adaptation au retard du retour visuel [121].

Pour Rohde et al. (2014), l'adaptation au retard du retour visuel pour une tâche de suivi de cible n'est possible que si la trajectoire de suivi est prévisible et que le participant peut anticiper les mouvements de la cible [122]. Les auteurs mènent une étude qui indique que seuls les participants ayant été entraînés sur la tâche de suivi prévisible ont des *after-effects* sur le post-test. Dans des travaux plus récents Rohde et Ernst (2016) discutent aussi des similitudes entre retours visuels perturbés par un retard ou par d'autres perturbations comme le décalage spatial (*offset*) ou l'inertie [121] (Figure 3.13). Par exemple, le fait que dans plusieurs études, les participants ralentissent leurs mouvements face à un retard [96, 122] pourrait en fait être une stratégie pour compenser de l'inertie. Les auteurs expliquent que cette ambiguïté entre les effets de différentes perturbations peut être la cause de l'absence d'adaptation observée dans certaines études.

Dans le cadre de l'IHM, les résultats de De la Malla et al. sont cohérents avec les résultats de Friston et al. [54] pour la souris ou Jota et al. [69] pour le toucher qui suggèrent

une importance plus grande du retour visuel dans la dernière partie du mouvement de pointage. Pour l'adaptation à la latence au toucher direct pour un mouvement continu, à notre connaissance, il n'existe aucun travail de recherche dans le domaine de l'IHM. Cela sera donc une de nos contributions.

Applicabilité des résultats du contrôle moteur pour la latence en IHM

Le contrôle moteur se sert de systèmes interactifs (souris, tablettes) pour interroger le fonctionnement humain de manière générale. Les paradigmes utilisés ne se soucient donc pas des conditions réelles d'utilisation des systèmes interactifs en jeu, qui, en revanche, nous intéressent pour l'IHM.

En comparant les études du contrôle moteur sur le retard du retour visuel décrites ci-dessus avec notre cadre de la latence au toucher direct, on constate effectivement deux différences majeures. D'une part, les retards considérés dans ces études sont beaucoup plus élevés que les niveaux que nous considérons pour la latence au toucher. D'autre part, dans chacune de ces études, la main des participants est soit cachée, soit l'interaction est indirecte. Cette dissociation entre la main et le retour visuel met alors en jeu une contradiction entre la proprioception et la vision des participants. Dans le cas de la latence au toucher, à la fois la main et un retour visuel (un objet déplacé sur l'écran) sont visibles. La contradiction est alors entre le comportement attendu du retour visuel (sensé rester sous le doigt) et son comportement réel. L'aspect direct de l'interaction au toucher peut donc avoir une influence spécifique sur l'adaptation à la latence.

3.2.3 Bilan

Notre revue de l'adaptation sensorimotrice dans la littérature du contrôle et de l'apprentissage moteur établit que l'être humain est capable de s'adapter à des perturbations visuomotrices. Alors que les résultats sur différents types de perturbations spatiales montrent une adaptation rapide, le résultat est moins clair pour l'adaptation au retard du retour visuel qui correspond au type d'adaptation que devrait induire la latence d'un système. Lorsque le retard intervient de manière discrète dans le temps, lors d'un simple contact, on observe un effet d'adaptation et un décalage du point de simultanéité subjective (PSS) entre le toucher et le retour visuel qui indique une recalibration visuomotrice. En revanche lorsque le retard intervient dans une interaction continue (comme le suivi de cible), on observe toujours une compensation de l'erreur, mais selon les cas, celle-ci peut être seulement due à des stratégies de compensation sans recalibration visuomotrice. L'adaptation au retard pour un mouvement continu pourrait alors dépendre de la tâche effectuée et de si cette tâche permet une identification claire du retard par rapport à d'autres types de perturbations aux effets similaires (comme l'inertie).

Le cadre de la latence au toucher que nous étudions diffère du cadre des études

conduites par la recherche sur l'apprentissage sensorimoteur car l'interaction est directe et l'utilisateur d'un système tactile est donc confronté à deux retours visuels parallèles : celui de sa main, et celui à l'écran (d'un objet déplacé par exemple). Dans ces conditions, on peut se demander si l'utilisateur est capable d'apprendre à compenser la latence et si cette compensation conduit à un phénomène d'adaptation. Une de nos contributions, exposée au chapitre 5, a été d'étudier l'adaptation à la latence pour une tâche de suivi de cible au toucher direct.

Deuxième partie

Contributions

Chapitre 4

Contributions à la compréhension du phénomène de latence

Pour comprendre le phénomène de la latence, il faut pouvoir la mesurer et la contrôler. Il faut aussi pouvoir tester des niveaux de latence les plus bas possible afin d'observer les comportements des utilisateurs sans latence (ou à des niveaux négligeables). Ces prérequis sont nécessaires pour étudier dans le détail les effets négatifs qu'elle peut avoir sur la performance des utilisateurs. Notre analyse de la littérature a illustré les limites des solutions actuelles pour ces trois problèmes : les dispositifs à faible latence ne permettent d'afficher qu'en noir et blanc sur une petite surface, les techniques de mesures de latence sont souvent lourdes à mettre en place et les effets de la latence n'ont jamais été testés pour l'interaction bimanuelle, qui est pourtant centrale dans le cadre de l'interaction sur grandes surfaces tactiles. Dans ce chapitre nous présentons nos contributions qui répondent aux questions suivantes :

- Peut-on construire un système à faible latence permettant un affichage de grande taille et en couleur à partir du matériel disponible dans le commerce ?
- Comment concevoir une technique de mesure de latence qui ne nécessite pas de dispositif extérieur, tout en étant précise ?
- Quels sont les effets de la latence sur l'interaction bimanuelle ?

4.1 Un système faible latence

Pour mener des études permettant de mieux comprendre les effets de la latence, il est en général nécessaire de comparer un groupe test qui subit les effets de la latence avec un groupe contrôle (ou groupe témoin) qui exécute les mêmes tâches mais sans la latence. Pour cela il faut disposer d'un système à latence négligeable. Nous avons vu au chapitre 3 que Ng et al. (2012) ont proposé un système d'interaction au toucher à 1 ms de latence. Cependant ce système présente deux limites quant à nos objectifs : 1 - il implique du matériel spécifique non disponible dans le commerce ; 2 - il n'affiche qu'en noir et blanc sur une surface réduite (24 cm×16 cm), ce qui limite son utilisation à des mouvements de faibles amplitudes. Afin de fournir un système aisément reproductible et de pouvoir étendre les études des effets de la latence à des tâches pouvant être réalisées sur de plus grandes surfaces, nous avons élaboré notre propre système à faible latence. Celui-ci, au prix d'une latence finale supérieure à 1 ms, est construit à partir de matériel disponible dans le commerce et d'un OS standard (et avec une grande disponibilité de bibliothèques logicielles). Il supporte aussi un affichage en couleur et sur une grande surface.

4.1.1 Démarche

Nous avons vu au chapitre 2 que la fréquence d'échantillonnage des capteurs ainsi que la fréquence de rafraîchissement de l'affichage sont deux facteurs cruciaux qui influent sur la latence d'un système. En s'appuyant sur les écrans tactiles capacitifs disponibles dans le commerce, la latence de notre système serait bornée par les capacités de ce matériel. Notre démarche a donc été de se défaire de la technologie tactile capacitive classique et de recréer l'interaction tactile avec du matériel différent et plus performant. La latence et le temps-réel étant des problèmes importants pour la gestion d'environnements virtuels, le matériel de suivi des mouvements des utilisateurs dont on se sert en réalité virtuelle ou en réalité augmentée est souvent conçu pour être très rapide. Le système de tracking Optitrack, de la société Natural point [6], par exemple, permet de suivre la position de points dans l'espace à une fréquence de 120 Hz et avec une faible latence.

Notre démarche est donc d'utiliser ce système Optitrack pour suivre la position des doigts sur un écran et détecter le toucher. Pour l'affichage, nous utilisons un écran Asus VG248QE renommé pour sa faible latence¹ et capable d'afficher à 120 Hz. Grâce aux 120 Hz de l'Optitrack et de l'écran, la contribution à la latence due à la discrétisation de l'information digitale (cf. chapitre 2) est réduite à une demi-période soit 4.2 ms. Dans la suite de cette section, nous décrivons en détail le fonctionnement et l'optimisation de notre système faible latence qui nous permet de ne pas dépasser 25 ms de latence.

1. La latence des écrans est appelée "input lag" dans la communauté technique.



FIGURE 4.1 – Photo de notre système faible latence. L’ordinateur portable est connecté aux caméras (rouges, en haut à gauche et en haut à droite) et exécute le logiciel de suivi. Les données de position du doigt sont transmises via le réseau au PC principal (en bas à droite). Celui-ci calcule le retour visuel et l’affiche sur l’écran posé horizontalement sur le bureau.

4.1.2 Le système Optitrack

Le système Optitrack fonctionne principalement avec 3 éléments. Des *marqueurs* sont fixés sur l’élément que l’on veut suivre, qui dans notre cas sera un doigt de l’utilisateur. Ces marqueurs sont détectés sur des images 2D prises par plusieurs *caméras*. Les données de toutes les *caméras* sont ensuite centralisées par le logiciel de traitement *Motive* qui en déduit la position 3D des marqueurs dans l’espace.

Marqueurs

Pour rendre un objet visible par les caméras, on y fixe des marqueurs réfléchissants. Plusieurs types de marqueurs sont fournis avec l’Optitrack et ont été utilisés à différentes fins. Ceux-ci sont représentés en Figure 4.2

Lors de nos premières études, nous avons utilisé des boules réfléchissantes de diamètre 1.4 cm (Figure 4.2, gauche). La taille importante de la boule et sa forme sphérique garantissent qu’elle soit bien vue des caméras, quelles que soient sa distance et son orientation et assurent la robustesse de son suivi. Cependant, fixée sur l’ongle des participants avec du ruban adhésif double face, cette boule est encombrante et empêche aussi partiellement



FIGURE 4.2 – Des marqueurs réfléchissants de différentes tailles. De gauche à droite : sphère de 14 mm de diamètre et demi-sphères de 4 mm et 3 mm de diamètre.

l'utilisateur de voir ce qu'il y a derrière son doigt. Dans les études suivantes, nous avons donc remplacé ce marqueur par un petit marqueur, demi-boule de 4 mm de diamètre, fixé sur l'ongle du participant avec du papier adhésif double face (Figure 4.2, centre). Grâce à un réglage fin de Motive, ce marqueur est suivi de manière aussi robuste que la sphère. Lors des phases de calibration (cf. paragraphe "Recréer l'interaction tactile"), pour plus de précision, un marqueur encore plus petit, de 3 mm de diamètre est utilisé (Figure 4.2, droite).

Caméras



FIGURE 4.3 – Une caméra Optitrack Flex 13. On note le cercle de LEDs infrarouges autour de l'objectif.

Les caméras Optitrack fonctionnent principalement avec un rayonnement infrarouge. L'objectif des caméras est entouré d'un cercle de LED émettant de la lumière infrarouge qui est réfléchi par les marqueurs puis captée par la caméra (Figure 4.3). Il existe plusieurs modèles de caméras Optitrack. Lors de nos études, nous avons utilisé exclusivement les caméras de la gamme Flex 13 qui fonctionnent par USB et peuvent capturer 120 images par secondes de 1280×1024 pixels. Nous avons aussi testé les caméras Prime 13, dont le taux d'échantillonnage peut monter jusqu'à 240 Hz. Cependant, avec notre système synchrone (cf. section 4.1.4) le rafraîchissement de l'écran impose toujours une discrétisation du retour

visuel à 120 Hz (cf. chapitre 2) et une plus haute fréquence d'échantillonnage des caméras ne peut suffire pour diminuer la latence. Nous verrons par contre au chapitre 5 qu'une quantité de données doublée pourrait permettre de réaliser une meilleure prédiction pour compenser la latence.

Un problème important à considérer est aussi le positionnement des caméras. Bien que deux caméras seulement soient nécessaires pour pouvoir calculer la position 3D d'un marqueur, nous avons toujours utilisé 3 ou 4 caméras en fonction du lieu où le dispositif a été installé. De cette manière nous augmentons la robustesse du suivi car, même si une caméra perd le marqueur à cause d'un problème d'occultation ou de luminosité, le suivi du marqueur n'est pas interrompu tant qu'au moins deux caméras peuvent le détecter. Le positionnement des caméras est illustré sur la photo de la Figure 4.1. Les caméras sont placées derrière l'écran par rapport à l'utilisateur (Figure 4.1), sur des trépieds, à environ 40 cm de hauteur, de manière à avoir un point de vue en plongée sur l'écran. Deux caméras sont placées sur la gauche de l'écran, une autre sur la droite (ou deux si on utilise 4 caméras), de manière à ce que chacune ait l'écran complet dans son champ de vision.

Motive

Les caméras Optitrack sont reliées à un concentrateur USB connecté à un ordinateur sur lequel tourne le logiciel Motive (version 1.6.0). Motive est le logiciel fourni par Natural Point [6] qui permet d'agréger les données des caméras pour fournir une position 3D du marqueur. Le logiciel fonctionne sous Windows sur un PC dédié, et diffuse les données de position 3D du marqueur vers le PC principal via le protocole UDP sur un réseau filaire dédié (i.e. non partagé avec l'internet) à 1Gbit/s. Cette communication n'est pas une source conséquente de latence puisque le temps de communication aller-retour a été mesuré à 0.5 ms. Les fonctionnalités de Motive permettent de gérer les réglages des caméras (puissance des LED, temps d'exposition) et de calibrer leurs positions relatives pour permettre le calcul de la position 3D.

4.1.3 Recréer l'interaction tactile

Motive ne fournit que la position 3D du marqueur dans l'espace avec une origine arbitraire. Pour connaître la position du doigt sur l'écran et déterminer si celui-ci est en contact ou non avec l'écran, deux calibrations sont requises afin de :

- mesurer la transformation entre les coordonnées 3D Optitrack et les coordonnées du repère de l'écran en pixel ;
- mesurer la transformation entre la position du marqueur (sur l'ongle) et le point de contact (sous le doigt).

Nous avons écrit un programme pour mesurer ces deux transformations illustrées en Figure 4.4.

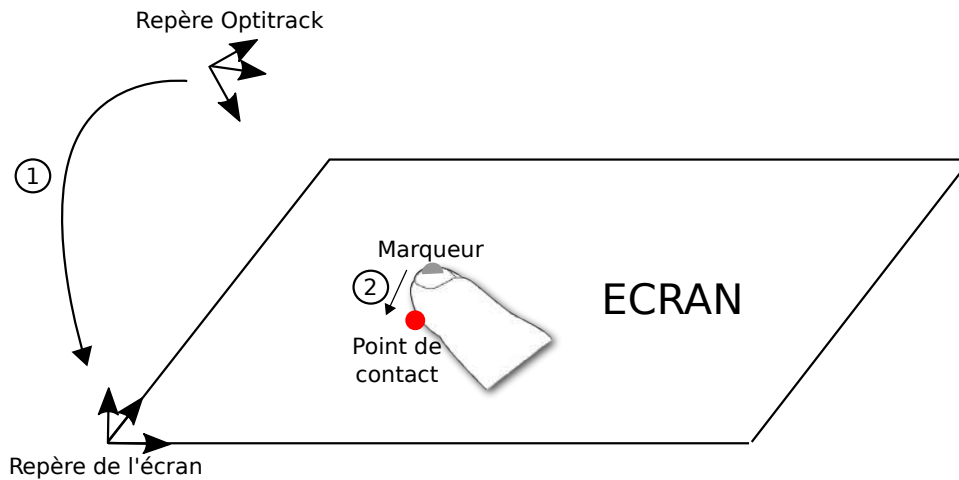


FIGURE 4.4 – Les deux calibrations nécessaires pour simuler l’interaction tactile avec l’Optitrack : 1 - Calibrer la transformation entre le repère Optitrack et le repère de l’écran ; 2 - Calibrer la transformation entre la position du marqueur sur le doigt de l’utilisateur et le point de contact sur l’écran où on veut afficher un retour visuel.

Calibration écran/caméras

La calibration entre les caméras et l’écran est indépendante de l’utilisateur du système. Elle est donc nécessaire uniquement lorsque les caméras ou l’écran sont déplacés.

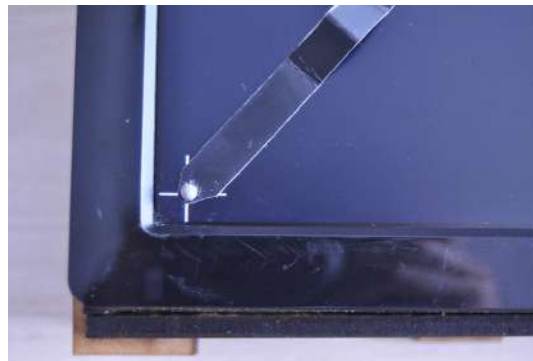


FIGURE 4.5 – La tige fine équipée d’un marqueur qui sert à effectuer la calibration caméras/écran. Sur cette photo, le marqueur est positionné pour la calibration du point en bas à gauche de l’écran.

La calibration écran/caméras s’effectue en deux étapes :

Étape 1 : Une calibration pour pouvoir projeter les coordonnées 3D de l’Optitrack sur l’écran et obtenir un couple de coordonnées x et y en pixel.

Étape 2 : Une calibration afin de connaître la distance du marqueur à l’écran.

Pour réaliser ces calibrations, nous nous servons d’un marqueur hémisphérique de diamètre 3 mm collé au bout d’une tige fine. De cette manière nous pouvons guider le marqueur à des endroits précis sur la surface de l’écran (Figure 4.5).

Pour la calibration en x et y notre programme affiche successivement une croix dans chacun des quatre coins de l'écran. Pour chaque croix, un opérateur vient positionner le marqueur situé au bout de la tige fine sur la croix et valide. Cette opération enregistre une correspondance entre les 4 positions 2D à l'écran en pixel, et les 4 positions 3D du marqueur données par l'Optitrack en millimètres. Tant que le marqueur reste à la surface ou au-dessus de l'écran, par interpolation, on peut faire correspondre n'importe quelle position du marqueur dans l'espace à une position 2D, en pixels, projetée sur l'écran.

Grâce à la calibration précédente, on connaît quatre points situés sur l'écran dans les coordonnées Optitrack. Avec ces points on pourrait calculer l'équation du plan de l'écran et déterminer la distance à ce plan de n'importe quelles coordonnées de marqueur envoyées par l'Optitrack. Cependant cela mènerait à des problèmes d'imprécision sur la détection du contact [34]. En effet l'écran n'est pas totalement plan mais un peu concave. Pour obtenir plus de précision, nous réalisons une seconde calibration en divisant l'écran en plusieurs sous-parties pour lesquelles une transformation indépendante est calculée.

Pour réaliser cette seconde étape, un opérateur déplace le marqueur sur la surface de l'écran de manière à couvrir le maximum d'espace. L'écran est virtuellement découpé en une grille de 4×3 rectangles. Pour chaque section de l'écran, le programme applique un algorithme RANSAC en sélectionnant aléatoirement des triplets de points capturés par l'Optitrack dans cette partie d'écran et en déduit une équation de plan dans les coordonnées Optitrack. Une fois les équations obtenues pour chaque section de l'écran, l'opérateur éloigne le marqueur de l'écran, ce qui *oriente* chacun des plans. Grâce à ces équations, on peut alors, à partir des coordonnées d'un marqueur dans le repère Optitrack, calculer sa distance à l'écran comme illustré en Figure 4.6.

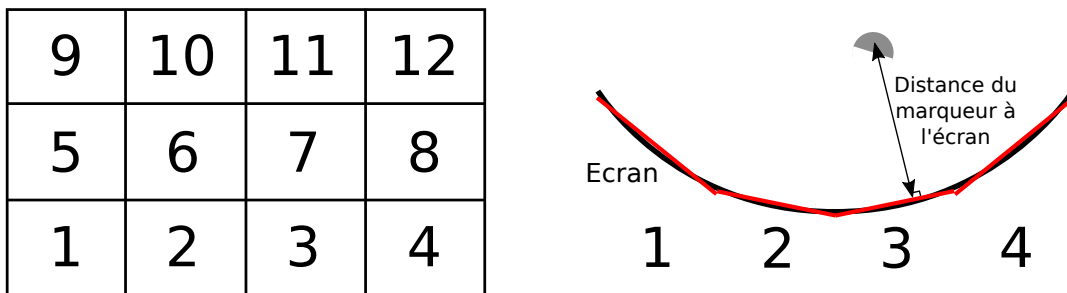


FIGURE 4.6 – Etape 2 de la calibration de notre système. A gauche : le découpage de l'écran vu du dessus en 12 sous-parties. A droite : Une vue en coupe de l'écran dont la concavité a été exagérée pour plus de clarté. Dans chaque sous-partie, un plan modélise la surface de l'écran (vu en coupe, trait rouge). On calcule alors la distance d'un marqueur à l'écran comme la distance du marqueur au plan modélisant l'écran dans la sous-partie que le marqueur survole.

Calibration doigt/écran

Cette calibration est réalisée à chaque fois qu'un marqueur est placé sur le doigt d'un utilisateur. Elle se déroule aussi en deux étapes :

Etape 3 : Le calcul de l'écart (*offset*) entre la position du marqueur projetée sur l'écran et la position de la pulpe du doigt qui touche l'écran.

Etape 4 : L'enregistrement de la hauteur maximale du marqueur pour les différentes sections de l'écran.

Pour l'étape 3, l'utilisateur doit poser son doigt, équipé du marqueur, sur une croix au centre de l'écran. On enregistre alors le décalage 2D entre la position projetée à l'écran du marqueur (obtenue grâce à la calibration de l'étape 1) et la position d'affichage de la croix sur laquelle se trouve la pulpe du doigt de l'utilisateur. Cette étape est illustrée sur la Figure 4.7.

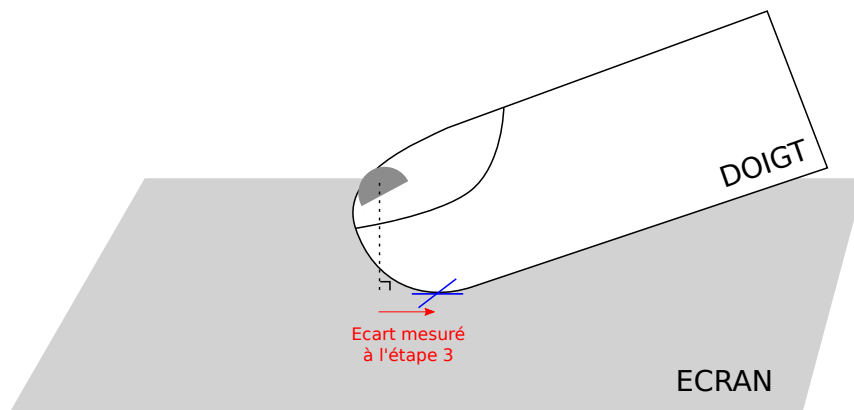


FIGURE 4.7 – Etape 3 de la calibration de notre système. Elle consiste à calculer l'écart entre la position projetée du marqueur sur l'écran et la position à laquelle la pulpe du doigt touche l'écran.

Pour l'étape 4, l'utilisateur doit déplacer son doigt sur l'écran sans le lever, de manière aléatoire mais en essayant de couvrir toute la surface. A chaque échantillonnage, le programme calcule la distance du marqueur à l'écran (grâce à la calibration de l'étape 2) et enregistre la hauteur maximale h_{max} du marqueur pour chacune des 12 sous parties de l'écran. Ces hauteurs maximales vont servir à définir des seuils de contact avec l'écran. Définir des seuils différents pour chaque sous-partie de l'écran est nécessaire car l'utilisateur ne place pas son doigt dans la même position selon que celui-ci est proche ou loin de lui sur l'écran, et cela influe sur l'altitude du marqueur collé sur l'ongle.

Fonctionnement

Lors de l'utilisation du système, l'Optitrack envoie les coordonnées 3D d'un marqueur au programme principal chargé d'afficher le retour visuel à l'écran. Grâce à l'étape 1, le programme déduit un couple (x,y) de coordonnées projetées du marqueur sur l'écran et

donc aussi la sous-partie de l'écran au dessus de laquelle se trouve le marqueur. Avec cette sous-partie et la calibration de l'étape 2, le programme calcule alors la distance du marqueur à l'écran. Si cette distance est inférieure à $1.1 \times h_{max}$ alors le programme considère que le doigt touche l'écran. Le programme considère que le doigt ne touche plus l'écran lorsque la distance devient supérieure à $1.2 \times h_{max}$, l'hystérésis permettant la stabilité du toucher. Lorsque le doigt est en contact avec l'écran, on peut calculer la position à laquelle afficher le retour visuel. Il s'agit des coordonnées (x,y) auxquelles on ajoute le décalage calibré lors de l'étape 3.

Le programme qui récupère les événements de toucher et qui affiche le retour visuel à l'écran s'exécute sur le PC principal sous MacOS (par opposition au PC dédié à Motive). Il est écrit en C++ et utilise la librairie OpenGL pour les rendus graphiques. Contrairement au système de Ng et al. dont le programme s'exécute sur une carte électronique dédiée [106], le fait d'utiliser une suite logicielle standard donne accès à un grand choix de bibliothèques qui rendent possible la programmation de multiples applications.

La résolution du système Optitrack est supérieure à celle de l'écran. La limite de précision de notre système est donc la taille d'un pixel de l'écran soit 0.28 mm. Les calibrations sont aussi suffisamment robustes pour effectuer des sessions de plusieurs heures sans perdre en précision.

4.1.4 Optimisation du système : synchronisation entre la capture et l'affichage

Dans le cas où l'image à afficher à l'écran est générée de manière désynchronisée avec l'affichage, la nouvelle image est prête à un moment aléatoire du cycle d'affichage, et on doit attendre le prochain rafraichissement pour l'afficher. Cela ajoute une latence variable comme illustré en Figure 4.8, en haut. Dans le meilleur des cas l'image est prête juste avant le rafraichissement ce qui n'ajoute pas de latence, mais dans le pire des cas, il faut attendre un cycle complet avant de pouvoir l'afficher. Cette latence variable se traduit par un tremblement du retour visuel.

Il est possible d'effectuer une synchronisation pour obtenir une latence constante (Figure 4.8, bas). Dans notre cas, paramétrer Motive pour une capture à 120 Hz (fréquence de rafraichissement de notre écran) n'est pas suffisant. En effet, cette fréquence de 120 Hz est seulement indicative et la moindre différence entre la fréquence de capture et la fréquence d'affichage résulte en une dérive progressive du déphasage entre les deux fréquences (Figure 4.8, milieu). Cette dérive génère une augmentation ou une réduction progressive de la latence jusqu'à un "saut" qui se traduit en une saccade du retour visuel clairement perceptible. Il est donc souhaitable de synchroniser précisément capture et affichage.

Pour réaliser cette synchronisation, nous utilisons le mécanisme de barrière (*fence*) d'OpenGL. Après avoir demandé l'échange des 2 mémoires graphiques (cf. chapitre 3, "Op-

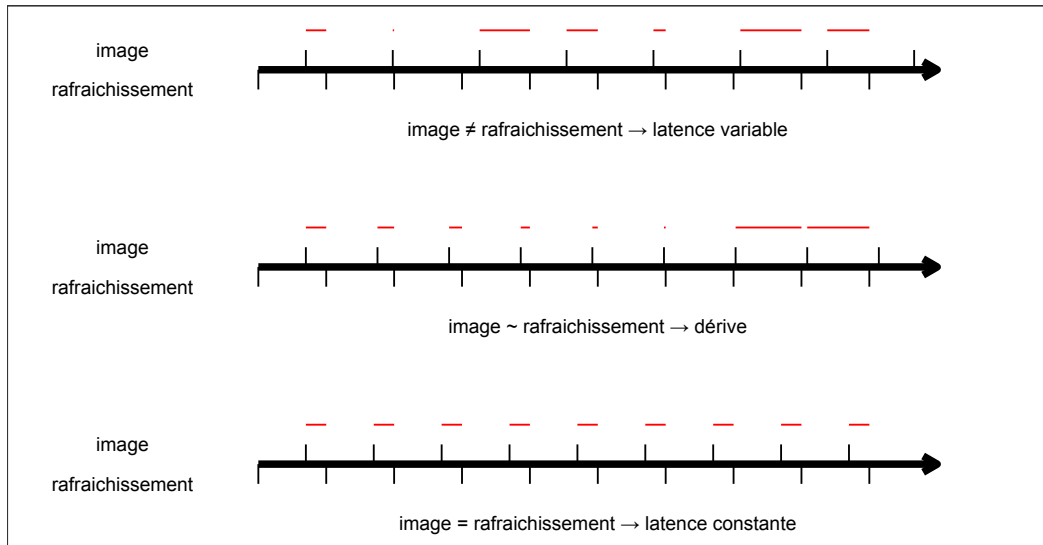


FIGURE 4.8 – Différents types de synchronisation entre la fréquence de rafraichissement de l'écran et la fréquence d'arrivée d'une nouvelle image à afficher ainsi que la latence qui en résulte. Les traits rouges indiquent la latence entre l'instant où une image à afficher est générée par le programme (marque vers le haut sur l'axe temporel) et l'instant où le rafraichissement de l'écran a lieu (marque vers le bas). Si la fréquence de production d'image est désynchronisée de celle de l'affichage alors la latence est variable (en haut). Si les fréquences sont presque identiques on observe une dérive progressive de la latence (au milieu). Si les deux fréquences sont synchrones, la latence est constante (en bas).

timisation du pipeline”) par un appel `swap_buffer`, nous posons une barrière pour bloquer l'exécution du programme. Lorsque l'exécution est libérée par OpenGL, nous savons que l'échange vient d'avoir lieu. Nous sommes donc à un instant proche du début d'un cycle d'affichage. Nous envoyons immédiatement un signal de synchronisation au système Optitrack par l'intermédiaire de l'interface série. Ceci permet donc de générer un signal de synchronisation en s'assurant que celui-ci est toujours envoyé à la fréquence de l'affichage et au même instant du cycle.

Grâce à cette synchronisation, la latence entre la disponibilité d'une nouvelle image et l'affichage reste constante ce qui évite les tremblements ou les saccades du retour visuel. Nous avons en plus optimisé cette synchronisation pour que les caméras Optitrack capturent et envoient un nouvel événement de position le plus tard possible mais de manière à ce que le programme puisse traiter l'évènement et générer la nouvelle image à afficher juste avant le rafraichissement (cf. chapitre 3 et Figure 3.7). Pour cela, nous bornons tout d'abord le temps de traitement de l'application et du GPU. Dans notre cas, ce temps de génération de la nouvelle image à partir des événements de position est toujours inférieur à 4 ms. Notre programme, arrête donc de récupérer des événements et commence à les traiter 4 ms avant la prochaine date de rafraichissement de l'écran estimée grâce à la barrière évoquée plus haut. Ensuite, nous utilisons l'interface de synchronisation du logiciel

Motive pour régler un temps d'attente (*Global sync offset*) entre le moment où le signal de synchronisation est reçu et le moment où les caméras capturent une position. Nous réglons ce temps d'attente de manière à ce que l'évènement provenant de l'Optitrack soit reçu par le programme juste avant le seuil des 4 ms. De cette manière, nous utilisons l'évènement Optitrack le plus récent possible affichable au prochain rafraichissement.

4.1.5 Latence finale

En utilisant la méthode haute précision de Bérard et Blanch [19] (cf. chapitre 2), nous mesurons la latence moyenne bout-à-bout de notre système à 25 ms. Cette valeur moyenne a été mesurée de nombreuses fois au cours de la thèse avec peu de variabilité entre ces mesures (± 1 ms) ce qui prouve la robustesse de notre dispositif. Quant à la fluctuation autour de cette valeur moyenne, la synchronisation du dispositif implique que la seule source de variabilité de la latence est celle due à la discrétisation du système, soit une variabilité de ± 4.2 ms autour de la valeur moyenne.

Sources des 25 ms de latence

Nous avons essayé de comprendre d'où proviennent les 25 ms de latence de notre système.

En premier lieu, nous savons que la discrétisation du système à 120 Hz impose une latence moyenne de 4.2 ms. De plus, notre programme prend 4 ms pour traiter les évènements Optitrack et générer le retour visuel.

Nous avons aussi mesuré la latence de notre écran Asus VG48QE. Pour cela, nous avons écrit un programme qui, en même temps qu'il affiche un fond uni blanc à l'écran génère un signal USB qui allume une diode sur un convertisseur USB série. En filmant l'écran et la diode avec une caméra haute fréquence (~ 2000 Hz), on peut compter le nombre d'images séparant l'allumage de la diode de l'apparition de l'image blanche à l'écran. Le retard entre les deux évènements fournit une mesure de la latence de l'écran. Nous trouvons 10.5 ms pour l'apparition des premiers pixels blancs et 14 ms pour que l'apparition de pixels blanc se propage jusqu'au milieu de l'écran (on rappelle que cet écran prend 7 ms pour afficher tous ses pixels, cf. chapitre 2).

Cela donne un total de $4.2+4+14=22.2$ ms de latence. Nous attribuons les 2.8 ms de latence restante aux traitements nécessaires à Motive pour calculer les positions 3D des marqueurs ainsi qu'au temps de transmission USB des caméras vers Motive. Le découpage de la latence de notre système est illustrée sur la Figure 4.9.

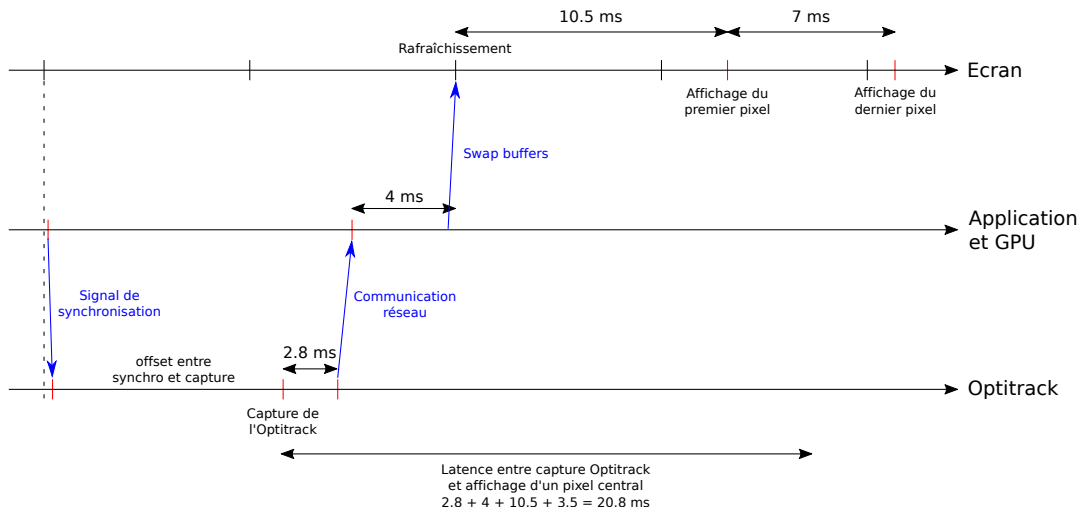


FIGURE 4.9 – Découpage de la latence sur notre système faible latence sur trois axes temporels de l'Optitrack, de l'application et de l'écran. La latence entre la capture de l'Optitrack et l'affichage d'un pixel au centre de l'écran est de 20.8 ms. En y ajoutant 4.2 ms dues à la discretisation du système on retrouve bien la valeur mesurée de 25 ms.

Contrôle de la latence

En ajoutant un temps d'attente dans l'application, par exemple, à la réception des données des capteurs, on peut artificiellement augmenter la latence de $N \times 8.33$ ms, où N est entier. En effet, comme notre dispositif fonctionne de manière synchrone, la latence que l'on peut ajouter de cette manière est discrétisée par les rafraîchissements de l'écran qui sont fixes (soit on affiche au prochain rafraîchissement, soit à l'un des suivants mais pas entre les deux).

Pour régler la latence avec un grain plus fin, entre deux rafraîchissements, on peut par contre toujours moduler le temps d'attente de la synchronisation de l'Optitrack. Si par exemple on attend 1 ms de moins après le signal de synchronisation pour capturer une donnée, alors cette donnée sera plus vieille de 1 ms et on ajoute donc 1 ms de latence au système. De cette manière, le système nous permet de simuler n'importe quelle latence supérieure à 25 ms.

4.1.6 Applications du système faible latence

L'objectif premier de notre système est de permettre l'interaction tactile directe à faible latence, avec un seul doigt. Cependant, deux fonctionnalités supplémentaires ont été ajoutées.

En particulier, nous avons besoin de l'interaction bimanuelle car nous voulons étudier les effets de la latence sur cette interaction. En plaçant un second marqueur sur l'autre index de l'utilisateur et grâce à un algorithme de suivi, l'interaction de chaque doigt sur

l'écran est gérée de manière indépendante. On utilise la même calibration pour les deux doigts en supposant alors que la transformation entre la position du marqueur et le point de contact pour le second doigt est la même que celle du doigt qui a servi pour la calibration. Cette hypothèse est raisonnable si on place le marqueur de manière symétrique sur le second doigt mais fait perdre un peu de précision. La fonctionnalité bimanuelle est aussi moins robuste car une confusion entre les deux marqueurs peut tout de même arriver dans de rares cas (par exemple, les bras qui se croisent rapidement). L'étude sur les effets de la latence sur l'interaction bimanuelle est décrite au paragraphe 4.3.

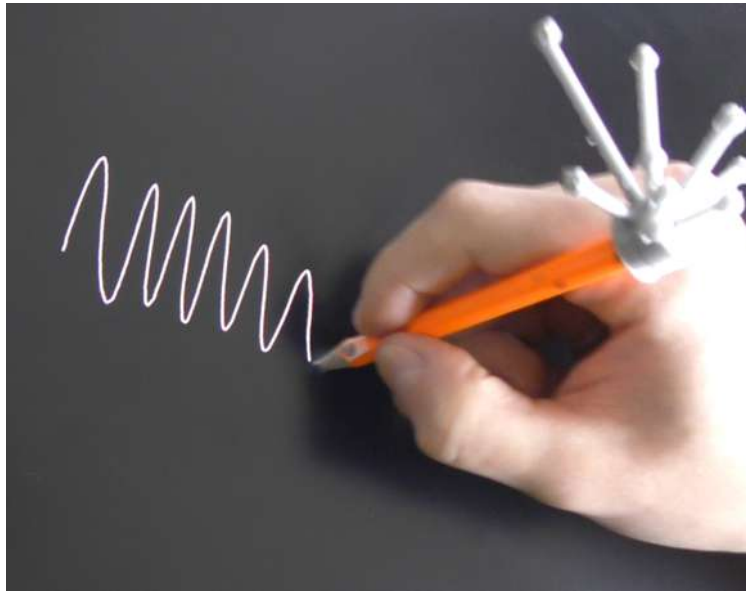


FIGURE 4.10 – Une application du système faible latence : l'interaction au stylet.

Un avantage de ce système, est la possibilité de créer d'autres interactions à faible latence en équipant n'importe quels objets avec des marqueurs. Ainsi, pour démontrer cette flexibilité, nous avons créé une interaction au stylet qui est illustrée sur la Figure 4.10. Trois marqueurs fixes les uns par rapport aux autres sont utilisés pour suivre les mouvements d'un crayon. Une fonctionnalité de Motive permet de définir, à partir de ces marqueurs, une structure rigide dont le logiciel nous fournit l'orientation en plus de la position. Nous utilisons ces informations pour mesurer la position de la mine du crayon par rapport aux trois marqueurs. Ensuite, le système est utilisé de la même manière qu'avec le doigt.

4.1.7 Bilan

Grâce à un système de suivi optique, un écran à faible latence, et la mise en œuvre d'une synchronisation optimisée entre les deux, nous avons obtenu un système simulant l'interaction au toucher avec 25 ms de latence seulement. Le système permet un affichage en couleur sur une grande surface. En utilisant un dispositif de suivi optique et une suite logicielle standard nous rendons notre système flexible et nous pouvons ajouter facilement

d'autres types d'interactions à faible latence telles que l'interaction au stilet. Ce système nous permet aussi de simuler n'importe quelle latence supérieure aux 25 ms de base.

Ce système nous a permis de mener plusieurs études en pouvant contrôler précisément la latence. Ces études sont présentées dans les sections suivantes ainsi que dans le dernier chapitre.

4.2 Une technique de mesure de la latence au toucher légère et précise

Cette section reprend nos travaux publiés à la conférence Interactive Tabletops and Surfaces (ITS) en 2015 : “A Predictive Approach for an End-to-End Touch-Latency Measurement” [35].

4.2.1 Problématique

La littérature propose de nombreuses techniques de mesure de latence, mais nous avons vu au chapitre 2 que la plupart d’entre elles nécessitent des efforts conséquents à mettre en place, avec notamment l’utilisation de matériel additionnel comme une caméra, un pendule ou un oscilloscope, et parfois une analyse image par image. En 2013, Bérard et Blanch ont introduit une nouvelle approche qui ne nécessite pas de matériel autre que le système dont on veut mesurer la latence et qui requiert l’intervention d’un opérateur humain [19]. La mesure repose sur la capacité de l’opérateur à réaliser une tâche de suivi de cible sur un cercle à vitesse constante. Cette tâche n’est pas si simple et requiert de l’entraînement pour atteindre de bonnes performances. Si la tâche n’est pas correctement exécutée, cela peut biaiser la mesure.

Nous mettons en œuvre une approche similaire avec un opérateur humain, mais en exécutant une tâche où les utilisateurs seront capables de davantage de précision. Nous savons que l’humain est très performant pour percevoir un très faible décalage entre son doigt et un retour visuel car il peut détecter des latences inférieures à 10 ms [106]. En associant cette capacité avec la technique de mesure proposée par Knibbe et al. [78] à base de prédiction, il en découle une nouvelle technique de mesure de la latence à la fois légère (ne nécessite pas de matériel supplémentaire) et potentiellement plus précise que la technique de Bérard et Blanch.

4.2.2 La technique de mesure

Notre technique de mesure de la latence s’appuie sur une prédiction de la trajectoire du doigt et suit la logique suivante. Lorsqu’on utilise une prédiction dans le but de compenser la latence, on affiche le retour visuel non pas à la dernière position connue, mais à une position prédite. En théorie, si on dispose d’un modèle de prédiction parfait de la trajectoire du doigt, alors il suffit de faire une prédiction dont la longueur est équivalente à la latence pour la compenser complètement. Pour aboutir à une mesure de latence, nous inversons ce raisonnement : si on ajuste la longueur de la prédiction jusqu’à ce que la position du retour visuel corresponde à la position du doigt, alors la longueur de la prédiction devient une mesure de la latence du système.

Notre approche consiste à afficher une ligne verticale qui suit les mouvements de

l'index de l'utilisateur sur la surface tactile. L'utilisateur doit bouger son doigt sur un guide représenté par deux lignes horizontales affichées à l'écran. Le mouvement doit se faire à vitesse la plus constante possible, mais l'amplitude de la vitesse n'est pas contrainte. Quand le doigt se déplace, à cause de la latence, la ligne est en retard sur le doigt (Figure 4.11, droite). L'utilisateur peut alors moduler la valeur de la prédiction à l'aide du clavier jusqu'à ce qu'il observe la meilleure correspondance entre la position de son doigt et la position de la ligne lorsqu'il se déplace. Lorsque l'utilisateur est satisfait de la correspondance, la longueur courante de la prédiction est la mesure de latence. Nous faisons l'hypothèse que la contrainte de vitesse constante est plus facile à satisfaire pour l'utilisateur, et génère moins d'erreur lorsqu'elle est mal respectée, que la contrainte de tracé de cercles à vitesse constante imposée par la méthode de Bérard et Blanch [19].

Ng et al. (2014) ont montré que les utilisateurs remarquent d'autant plus la latence que l'objet est de petite taille [104]. Nous minimisons donc la taille de l'objet déplacé dans le but de maximiser la précision de la perception des utilisateurs. La ligne affichée à l'écran mesure 1 pixel de large (0.28 mm). Nous favorisons le repérage de la correspondance entre la ligne très fine et le doigt épais en traçant une ligne sur l'index de l'utilisateur dans la continuité de la ligne affichée à l'écran lorsque le doigt est immobile (ce trait est visible sur la Figure 4.11, à gauche).

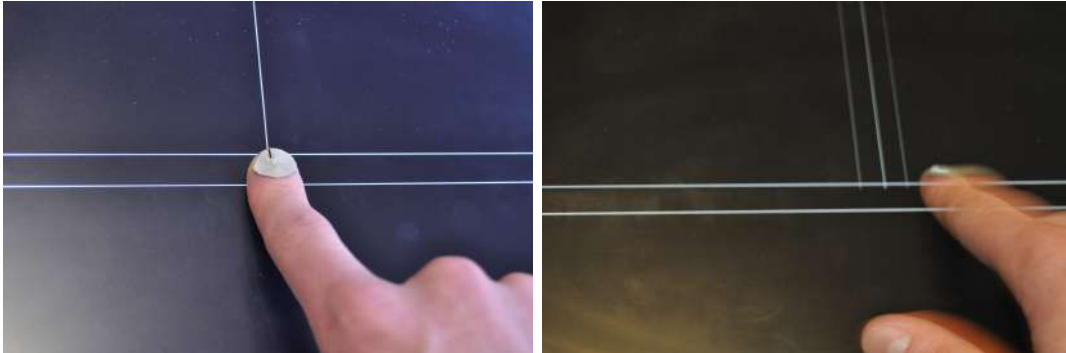


FIGURE 4.11 – Un utilisateur réalisant une mesure de latence avec notre méthode prédictive. A gauche, lorsque l'utilisateur ne bouge pas, la ligne verticale à l'écran est alignée avec la ligne sur son doigt. A droite, quand l'utilisateur se déplace (ici vers la droite), la ligne est en retard du fait de la latence. L'utilisateur doit alors modifier la longueur de prédiction pour que la ligne soit alignée même lorsqu'il est en mouvement. Sur la photo de droite, on aperçoit plusieurs lignes verticales : c'est un artefact dû au temps d'exposition de la photo qui capture plusieurs cycles d'affichage de l'écran.

Cette approche prédictive ne peut fournir une estimation précise de la latence que si le modèle utilisé pour la prédiction est correct. Nous utilisons un modèle de mouvement à vitesse constante pour la prédiction. A chaque cycle d'affichage, une position prédite est calculée selon l'équation :

$$\hat{\mathbf{x}}_i = \mathbf{x}_i + L \times \hat{\mathbf{s}}_i \quad \hat{\mathbf{s}}_i = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\Delta t} \quad (4.1)$$

où \mathbf{x}_i représente les positions du doigt observées, $\hat{\mathbf{s}}_i$ la vitesse estimée et Δt la période de rafraîchissement de l'écran. La longueur de prédiction ajustable L est utilisée pour mesurer la latence.

Mathématiquement, ce modèle prédit parfaitement les futures positions du doigt seulement si celui-ci bouge en ligne droite à vitesse constante ce qui n'est jamais vraiment le cas. Pour la ligne droite, nous demandons aux utilisateurs de faire des mouvements de translation horizontale sur l'écran en essayant de rester entre les deux lignes blanches horizontales qui forment un guide (Figure 4.11). Comme tout mouvement de translation de l'utilisateur d'un côté à l'autre de l'écran inclut une phase d'accélération (au début) ainsi qu'une phase de décélération (à la fin) qui induisent respectivement une prédiction trop courte et trop longue, nous demandons aux utilisateurs de se concentrer sur la partie centrale de leur mouvement et de stabiliser au maximum leur vitesse lors de cette phase. Même de cette manière la vitesse n'est jamais parfaitement constante. Toutefois, les utilisateurs ne choisissent le meilleur alignement entre le doigt et la ligne qu'au bout de plusieurs mouvements de translation *dans les deux sens*. Nous supposons que les erreurs de prédiction s'équilibrent sur l'ensemble de ces mouvements et permettent une estimation la plus juste (proche de la valeur réelle) et précise (variabilité autour de la valeur réelle) possible de la latence.

Les systèmes souffrant de latences plus élevées nécessitent aussi des longueurs de prédiction plus grandes pour la mesure, qui résultent en de plus grandes instabilités. La précision de notre technique pourrait donc décroître avec l'augmentation de la latence du système. La justesse et la précision de notre technique sont donc évaluées à deux niveaux de latence différents dans une étude utilisateur décrite dans la section suivante.

4.2.3 Evaluation de la technique par une étude utilisateur

Une étude utilisateur nous permet d'évaluer la justesse et la précision de notre nouvelle technique de mesure. Nous comparons aussi notre approche à celle de Bérard et Blanch, "Low Overhead" (LO), qui est la mesure de la littérature la plus avancée pour les mesures ne nécessitant pas de matériel externe.

Les 8 personnes qui participent à l'étude ont un âge moyen de 26.75 [21-33] et incluent 3 femmes. Tous les participants proviennent de notre laboratoire et ont des connaissances en informatique, mais aucun n'a d'expérience ni avec la méthode LO ni avec notre approche prédictive.

Notre étude a été réalisée avec notre propre système faible latence décrit en début de chapitre. Cependant, notre approche ne se limite pas à ce système particulier et pourrait être intégrée sur n'importe quel smartphone ou tablette. Pour s'assurer de la latence du système aux deux niveaux étudiés (30 ms et 80 ms), celle-ci est vérifiée avec la méthode précise "High Accuracy" décrite par Bérard et Blanch [19].

Tâche

Pour la méthode LO, nous avons utilisé une forme de roue semblable à celle de Bérard et Blanch [19] (Figure 4.12). L'écran affiche un cercle de rayon 8.8 cm avec un rayon en rotation à vitesse constante $s = 3rad.s^{-1}$. Les participants doivent suivre l'intersection du rayon et du cercle. Le début et la fin de l'enregistrement sont contrôlés par le participant en appuyant sur la barre espace d'un clavier. Les participants doivent exécuter au moins un tour et demi sur le cercle avant de terminer l'enregistrement.

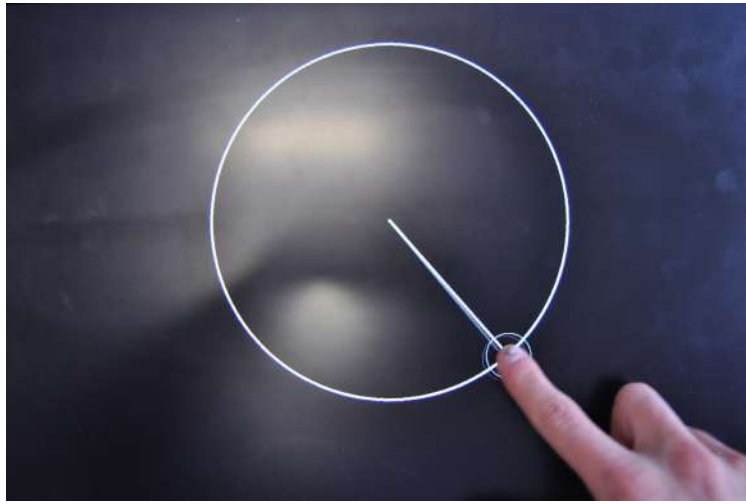


FIGURE 4.12 – Un utilisateur réalisant une mesure de latence avec la méthode LO [19].

Pour la méthode prédictive, les participants exécutent la technique décrite en Section 4.2.2. Ils peuvent ajuster la valeur de la prédiction à l'aide d'un clavier : deux touches permettent des incréments de ± 1 ms, deux autres des incréments de ± 0.1 ms. Les participants doivent ajuster la valeur de la prédiction jusqu'à ce qu'ils considèrent que la ligne affichée à l'écran et la ligne sur le doigt correspondent, lors de mouvements horizontaux à une vitesse aussi constante que possible. Aucune contrainte de temps n'est donnée et le participant décide que sa mesure est terminée lorsqu'il est satisfait de l'alignement.

Protocole

Le design de l'étude inclut deux facteurs intra-sujet : METHODE avec deux niveaux, LO (méthode de Bérard et Blanch [19]) et PRED (notre méthode prédictive) et LATENCE avec deux niveaux, 30 ms et 80 ms.

L'ordre de présentation des facteurs est illustré en Figure 4.13. Les participants exécutent 5 mesures dans chacune des 4 conditions (2 METHODE \times 2 LATENCE). Nous avons donc 160 mesures (8 participants \times 4 conditions \times 5 répétitions). Les conditions sont ordonnées d'abord par METHODE puis par LATENCE. Les ordres de présentation de METHODE et LATENCE sont équilibrés entre les participants pour compenser un effet

d'ordre potentiel. Au début de l'étude, les deux méthodes sont présentées aux participants ainsi qu'une démonstration par l'expérimentateur.

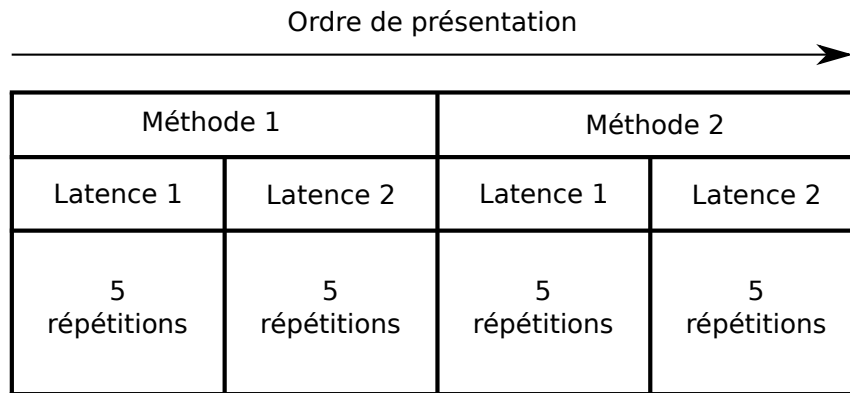


FIGURE 4.13 – Ordre de présentation des différents facteurs pour l'étude sur la mesure de latence (voir détails dans le texte).

Avec la méthode LO, les participants ont 1 min pour s'entraîner à suivre la rotation. Ensuite, ils exécutent cinq mesures dans la première condition de latence puis cinq dans la seconde. Pour encourager les participants à être plus précis sur le suivi de cible, l'erreur standard est affichée à la fin de chaque mesure.

Avec la méthode prédictive, on demande aux participants de se concentrer sur la phase centrale d'un mouvement d'un côté à l'autre de l'écran et d'y maintenir une vitesse la plus constante possible. La longueur de prédiction, non accessible aux participants, démarre à 0 ms. La précision de la mesure peut être faussée si les participants ne font que répéter la même séquence de touche au clavier à chaque nouvelle mesure. En fait, comme la mesure nécessite plusieurs dizaines d'appui sur les touches, les participants ont tous utilisé la fonction "key repeat" du clavier. En maintenant enfoncée la touche pour modifier la longueur de la prédiction, celle-ci se modifie pas à pas jusqu'à ce que la touche soit relâchée. Il est très difficile de reproduire exactement le même temps d'appui sur la touche, et l'utilisation de cette fonction par les participants nous assure donc qu'ils ne font pas que reproduire une séquence. Lorsque les participants sont satisfaits de la correspondance entre les deux lignes, la longueur de prédiction finale est enregistrée puis remise à 0 ms pour la mesure suivante.

Analyse de l'erreur de mesure

Pour chacune des 160 mesures de latence, nous enregistrons l'erreur de mesure, c'est-à-dire la différence entre la valeur de la latence théorique (30 ms ou 80 ms) et la valeur mesurée. Nous retirons 10 outliers intra-sujets qui sont à plus de 1.5 fois l'écart interquartile au dessus du dernier quartile ou en dessous du premier quartile. Les mesures de notre second participant à 80 ms avec la méthode LO sont en moyenne à 35 ms et très variables. Nous considérons les cinq mesures de cette condition comme outliers et nous les retirons des

analyses. Il reste alors 145 mesures.

L'analyse de l'erreur de mesure et de sa variabilité intra et inter participants est faite graphiquement et avec des calculs de moyenne et d'intervalle de confiance.

Résultats

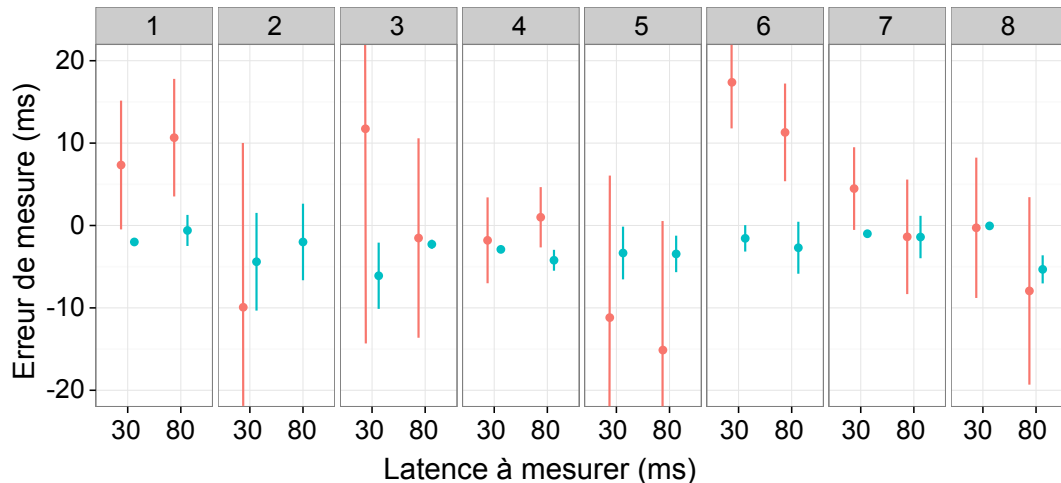


FIGURE 4.14 – Erreur moyenne de mesure pour chaque participant (de 1 à 8) à 30 ms et 80 ms de latence avec la méthode LO (rouge, gauche) ou avec notre méthode prédictive (bleu, droite). Les intervalles de confiance à 95% indiquent la variabilité intra sujet. (Figure traduite de [35])

La variabilité intra sujet de la mesure est plus grande avec la méthode LO comparée à la méthode prédictive (Figure 4.14). L'écart type moyen des cinq mesures, sur les sujets et les deux niveaux de latence est plus petit avec la méthode prédictive (1.6 ms) qu'avec la méthode LO (8.4 ms).

Estimer la latence avec la méthode prédictive prend plus de temps (entre 30 s et 1 min selon les participants) qu'une mesure avec la méthode LO (moins de 10 s). Cela est dû aux nombreuses étapes d'ajustements et d'observations nécessaires avec l'approche prédictive. Bérard et Blanch recommandent qu'avec la méthode LO, plusieurs mesures soient effectuées et que soit sélectionné *le meilleur essai*, c'est-à-dire celui où le suivi de cible montre la plus faible erreur-standard. Pour essayer de compenser ces différences entre les deux méthodes nous avons comparé une unique mesure de la méthode prédictive (arbitrairement, la première) avec le meilleur des cinq essais de la méthode LO, pour que les temps opérateur nécessaires pour chaque technique soient rééquilibrés.

La méthode LO se montre très juste lorsqu'on moyenne les résultats des 8 participants : la moyenne des 8 meilleures mesures donne 33.5 ms à 30 ms et 80.4 ms à 80 ms. La méthode prédictive donne des résultats similaires avec 28.1 ms à 30 ms et 78.4 ms à 80 ms (Figure 4.15). Mais, l'intervalle de confiance pour la moyenne de l'erreur de mesure (en

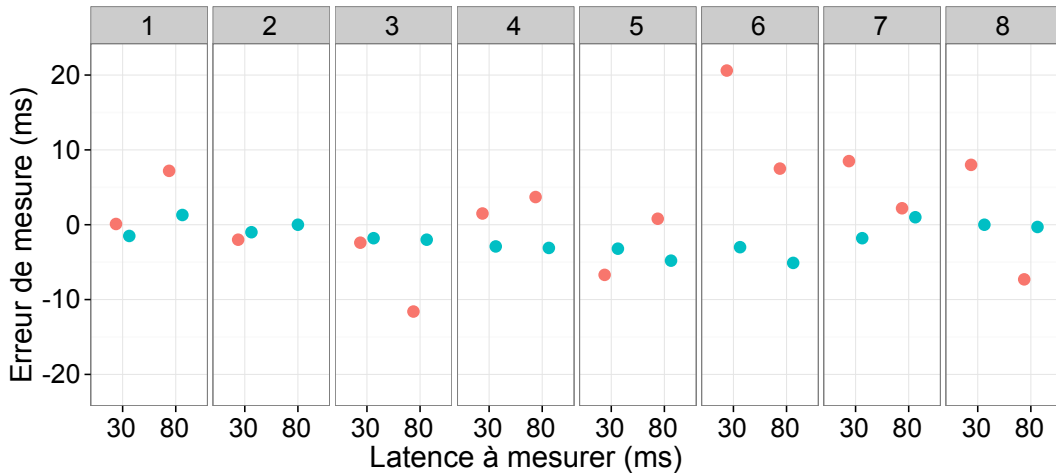


FIGURE 4.15 – Erreur de la mesure LO la plus stable (rouge, gauche) comparée avec l’erreur de la première mesure prédictive (bleu, droite) pour chaque participant à 30 ms et 80 ms de latence. (Figure traduite de [35])

regroupant les deux niveaux 30 ms et 80 ms) est plus petit avec la méthode prédictive qu’avec la méthode LO (Figure 4.16). Cela suggère qu’une mesure rapide de la latence en 2 min serait aussi juste mais plus précise (car moins soumise à la variabilité inter-sujets) avec la méthode prédictive qu’avec la méthode LO.

A la question “Quelle méthode était la plus facile”, 6 participants sur 8 répondent qu’ils préfèrent l’approche prédictive. Ils trouvent la tâche de suivi de cible difficile car contrainte par le système alors que la méthode prédictive leur permet d’adopter leur propre rythme.

4.2.4 Discussion

Avec la méthode LO, la mesure de la latence présente une grande variabilité intra et inter participants. Cela pourrait être dû au faible temps d’entraînement, la tâche de rotation nécessitant sûrement un certain temps pour être maîtrisée. La méthode LO donne une bonne estimation de la latence en moyenne sur plusieurs utilisateurs, mais utiliser la mesure d’un unique utilisateur peut mener à de larges biais. Au contraire, notre méthode prédictive est plus juste et plus précise quand on considère la mesure d’un unique opérateur.

Il semble que les mesures obtenues avec la méthode prédictive souffrent d’un léger biais négatif (la mesure sous-estime la valeur réelle). Nous supposons que cela est dû au fait de toujours démarrer la mesure avec une longueur de prédiction à 0 ms. La convergence de la prédiction par valeurs basses peut mener à une sous-estimation de la valeur réelle de la latence. En conséquence, nous recommandons d’exécuter plusieurs mesures et de faire varier aléatoirement la longueur initiale de la prédiction.

Contrairement à notre hypothèse, notre méthode prédictive ne montre pas une précision plus faible à 80 ms qu’à 30 ms. Il semble donc que les utilisateurs puissent s’ac-

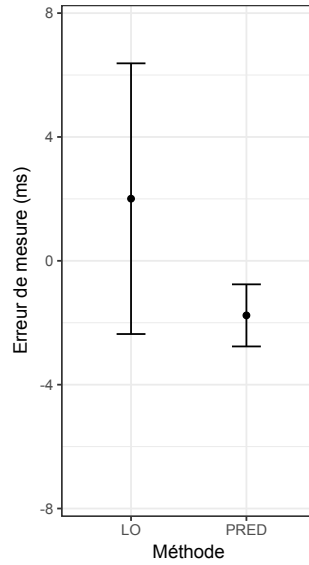


FIGURE 4.16 – Moyenne de l’erreur de mesure inter-participants, dans la condition LO (en gardant la mesure la plus stable) et la condition prédictive (avec la première mesure). La variabilité inter-participants est plus grande avec la mesure LO.

commoder de l’effet de flou qui apparaît lorsque la longueur de prédiction augmente. De même, l’effet de flou créé par la variabilité de la latence du système (± 4.2 ms autour de la valeur moyenne, cf. section 4.1.5) semble bien intégré et la méthode prédictive permet de mesurer la valeur de la latence moyenne avec justesse. Des études supplémentaires seraient néanmoins nécessaires pour évaluer plus en profondeur la robustesse de la mesure à la variabilité de la latence.

Une étude plus poussée sera nécessaire pour valider notre approche sur un système avec des capteurs à fréquence plus faible. Bien que des capteurs à 120 Hz existent déjà dans le commerce (sur les Apple iPad depuis l’iPad Air 2), la plupart des écrans tactiles vendus actuellement ont des fréquences d’échantillonnage dans l’intervalle [60 Hz-85 Hz]. Une fréquence plus faible peut mener à une prédiction moins stable et donc à une mesure moins précise. Cependant, les résultats similaires de notre étude à 30 ms et 80 ms indiquent que les utilisateurs sont peu sensibles aux instabilités de prédiction dans le cadre de la mesure.

4.2.5 Bilan

En conclusion, notre contribution est une nouvelle technique de mesure de latence se basant sur l’utilisation d’une prédiction et l’assistance d’un opérateur humain. Cette technique est facile à développer, et ne requiert aucun matériel additionnel en plus du système dont la latence est évalué. Nous avons réalisé une étude utilisateur qui montre que notre approche est aussi juste, plus précise et plus facile à réaliser pour l’opérateur qu’une technique de mesure proposée par le passé pour les mêmes conditions. Alors que

des techniques de mesures lourdes et complexes pouvaient décourager les concepteurs de systèmes interactifs, notre technique légère devrait les inciter à diagnostiquer la latence de leurs systèmes.

4.3 Effets de la latence sur l'interaction bimanuelle

Une fois que nous pouvons contrôler et mesurer la latence de notre système, nous sommes en mesure d'étudier les effets négatifs que les perturbations qu'elle induit peuvent avoir sur la performance des utilisateurs. En particulier, nous nous concentrons sur l'interaction bimanuelle, utilisée sur les grandes surfaces tactiles, et pour laquelle l'influence de la latence n'a jamais été évaluée à notre connaissance.

Cette section reprend nos travaux publiés à la conférence Interactive Surfaces and Spaces (ISS) en 2016 : "Effect of Touch Latency on Elementary vs. Bimanual Composite Tasks" [36].

4.3.1 Introduction

Nous avons vu au chapitre 2 que l'impact de la latence a été étudié pour diverses modalités d'interaction (souris, réalité virtuelle, tactile...). Dans le cas du tactile, il a été montré que la latence se perçoit même à des valeurs inférieures à 10 ms et que la performance des utilisateurs est détériorée en dessous de 25 ms. Ces études ont été menées sur des tâches simples comme l'acquisition de cibles, le suivi de cible ou le pilotage. Ces tâches de base sont implicitement considérées comme des sous-tâches d'une interaction plus globale.

Cependant, l'interaction au toucher dans la vie de tous les jours, s'effectue dans un environnement plus complexe avec potentiellement de multiples objets et/ou cibles. Quand elle est disponible, comme sur les grandes surfaces tactiles, les utilisateurs peuvent aussi utiliser l'interaction bimanuelle pour paralléliser leurs actions et améliorer leur performance [72]. Dans ce cas, l'interaction au toucher devient un ensemble de tâches composites où les tâches simples étudiées par la littérature n'apparaissent pas en séquence mais sont entrelacées. L'environnement complexe et l'interaction bimanuelle ajoute une charge cognitive pour : 1 - gérer les distracteurs [26] (i.e. les objets qui n'interviennent pas dans la sous-tâche), 2 - planifier l'ordre des sous-tâches et 3 - choisir une main ou l'autre pour telle ou telle tâche.

Des études récentes ont montré que lorsque la charge cognitive augmente, la perception de la latence décroît [15] (cf. chapitre 2). De plus, si la main non dominante doit réaliser des actions précises, ses mouvements seront plus lents que si ces mêmes actions sont réalisées par la main dominante [73]. Or, le ralentissement du geste rend aussi la latence moins perceptible. Si la latence est moins perceptible, on peut supposer que son impact sur la performance de l'utilisateur est moins fort. La motivation de notre travail est d'étudier si l'effet négatif de la latence observée sur des tâches simples existe aussi sur des tâches plus complexes, plus proches d'une interaction en situation réelle.

Une étude *in situ* sur l'influence de la latence sur la performance permettrait de fournir des résultats sur des tâches de la vie de quotidienne mais il serait difficile de

contrôler l'ensemble des autres facteurs pouvant influencer sur la performance. Au contraire, une étude en laboratoire permet de contrôler les différents facteurs entrant en jeu et de vérifier que tous les facteurs sont les mêmes dans toutes les conditions, sauf le facteur testé. Les conditions de laboratoires sont alors plus favorables à l'émergence de résultats théoriques permettant de comprendre les effets de la latence sur la performance.

Nous avons réalisé une étude contrôlée pour comparer l'effet de la latence sur une tâche simple de positionnement et sur une tâche composite plus complexe. La tâche composite se divise en plusieurs sous-tâches de positionnement semblables à la tâche simple, mais réalisées à deux mains. Nous estimons que la tâche composite augmente la charge cognitive comparée à la tâche simple puisqu'elle demande des décisions pour le tri et de la planification pour la coordination des deux mains. Notre hypothèse est que l'effet de la latence devrait être réduit pour la tâche composite comparativement à la tâche simple.

4.3.2 Etude utilisateur : effets de la latence sur une tâche simple vs composite

Tâche simple et tâche composite

La tâche simple, est une tâche de positionnement qui consiste à amener un objet d'un point de départ vers une cible (Figure 4.17, gauche).

Afin de pouvoir comparer les tâches, la tâche composite est construite avec les mêmes cibles que la tâche simple. La différence est que plusieurs objets sont présentés simultanément au lieu d'un unique objet, et ces objets doivent être triés sur deux cibles selon leur couleur (Figure 4.17, droite).

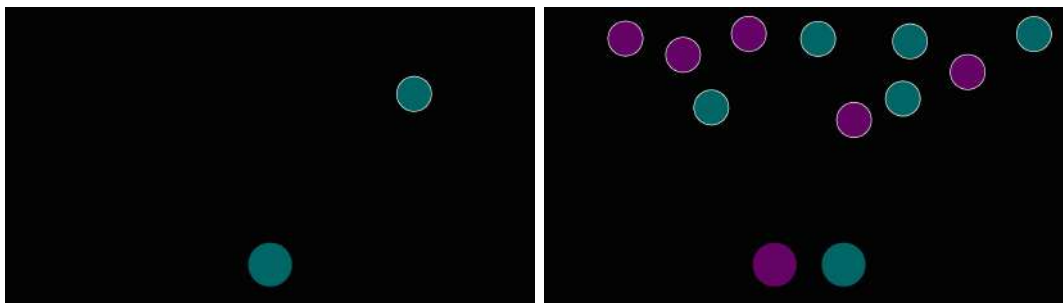


FIGURE 4.17 – Affichage au début de la tâche simple (gauche) et au début de la tâche composite (droite). Les utilisateurs doivent amener les disques situés dans la moitié haute de l'écran dans les cibles au bas de l'écran. La tâche composite est effectuée avec les index des deux mains et nécessite la correspondance des couleurs entre les objets et les cibles. (Figure issue de [36])

Relativement à la tâche simple, la tâche composite ajoute une sous-tâche de tri (amener les objets sur la cible de la bonne couleur), une sous-tâche de planification (optimiser la séquence de positionnement des différents objets), l'allocation de chaque main à un sous-

ensemble d'objets et la coordination motrice entre les deux mains. La tâche composite est conçue explicitement pour accroître la charge cognitive des utilisateurs. Il s'agit d'un "assemblage orthogonal" [72] où chaque main réalise une sous-tâche indépendante ce qui peut "imposer une lourde charge cognitive" [72]. Le fait de paralléliser les actions à deux mains ajoute aussi de la diversion visuelle car le regard des utilisateurs doit constamment naviguer entre les deux mains, ce qui peut aussi accroître la charge cognitive [72] et réduire les ressources attentionnelles ayant pour conséquence une moins bonne perception de la latence.

Les deux tâches sont réalisées sur notre système décrit au début de ce chapitre à deux niveaux de latences différents : une valeur de latence faible, 25 ms, qui est le minimum de notre système décrit ci-dessus et une valeur plus élevée, 100 ms, où un effet de la latence devrait être nettement visible, au moins sur la tâche simple [34, 69]. Les deux niveaux de latences utilisés sont vérifiés régulièrement à l'aide de la technique prédictive décrite dans la section précédente. En choisissant deux niveaux assez différents (facteur 4 entre les deux valeurs), nous renforçons l'effet de la latence pour rendre une potentielle différence entre les deux types de tâches clairement visible. Pour chaque niveau, nous mesurons la dégradation de performance due à la latence.

Protocole



FIGURE 4.18 – Affichage lors de pauses. Les carrés verts sont les cibles à toucher pour commencer l'essai suivant. A gauche, le prochain essai sera la tâche simple dans la condition de latence minimale. A droite, le prochain essai sera la tâche composite avec de la latence.

Le facteur principal de l'étude est TACHE, intra-sujet, avec deux conditions : *simple* et *composite*. Dans les deux conditions, les participants doivent déplacer un *objet* (disque, $R = 1.78\text{ cm}$) dans une *cible* (disque, $R = 2.20\text{ cm}$) de la même couleur. Chaque fois que l'objet est relâché, c'est-à-dire, lorsque le participant lève son doigt de l'écran tactile, un retour visuel est fourni : la couleur de la cible émet un flash de 0.1s vert (resp. rouge) si la tâche de positionnement est réussie (resp. ratée). La tâche est réussie si l'objet est relâché alors qu'il est entièrement dans la cible. La consigne donnée aux participants est de réaliser les tâches "aussi vite que possible, mais en évitant de faire trop d'erreurs". Des pauses sont proposées régulièrement pour éviter les effets de fatigue. Pendant les pauses, une tortue ou

un lapin est affiché à l'écran pour indiquer si le bloc suivant sera avec faible ou forte latence (Figure 4.18). Le but est de réduire l'effet de surprise d'un brusque changement de latence et d'éviter les outliers au début de chaque nouveau bloc. Les participants choisissent quand reprendre en touchant une cible carrée verte pour la tâche simple, ou deux cibles de ce type pour la tâche composite. Ces cibles se situent à la même position que les cibles de la tâche qui va suivre et garantissent que les doigts du participant sont à la même position au début de chaque essai (Figure 4.18).

Pour la tâche simple, une cible bleue est affichée à une position fixe au centre bas de l'écran. Des objets bleus apparaissent les uns après les autres à des positions pseudo-aléatoires de la moitié haute de l'écran (Figure 4.17, gauche). Une réalisation consiste à toucher l'objet avec l'index de la main dominante, à venir le positionner sur la cible et à relâcher l'objet (lever le doigt) pour valider. Dès que l'objet est lâché et le retour visuel donné, un nouvel objet apparaît pour la répétition suivante de la tâche. Les répétitions sont groupées par blocs de 40, chaque répétition ayant une position de départ unique (index de difficulté de Fitts entre 5.08 et 6.19). Les blocs de 40 répétitions sont présentés d'abord avec la latence faible puis la latence haute. Cette séquence est répétée cinq fois pour un total de 400 positionnements : $[40 \text{ (positions de départ)} \times 2 \text{ (latences)}] \times 5 \text{ (répétitions)}$. Les deux conditions de latence sont donc entrelacées ce qui permet d'éviter un effet d'ordre, la latence faible arrivant 5 fois avant et 4 fois après la latence haute et inversement. Les positions initiales des objets sont aussi aléatoires dans chaque bloc pour éviter les apprentissages de séquences. Une pause est possible entre chaque bloc.

Pour la tâche composite, deux cibles, une violette et une bleue, sont affichées en bas de l'écran, à 7 cm l'une de l'autre. Au début de chaque répétition, 10 objets, 5 violets et 5 bleus, apparaissent simultanément dans la moitié haute de l'écran. Une répétition de la tâche consiste à amener les 10 objets dans leur cible correspondante (Figure 4.17, droite). Les index des deux mains sont utilisés. La tâche se termine lorsque les 10 objets ont été déplacés et relâchés. Il y a 4 positions de départ différentes pour l'ensemble des 10 objets, qui rassemblées correspondent aux 40 positions relatives objet-cible utilisées pour la tâche simple. Les 4 positions sont réalisées d'abord à faible latence puis à latence élevée. Cette séquence est répétée cinq fois pour un total de 400 positionnements : $[10 \text{ (objets)} \times 4 \text{ (positions de départ)} \times 2 \text{ (latences)}] \times 5 \text{ (répétitions)}$. Les deux conditions de latence sont donc entrelacées comme pour la tâche simple. Une pause est possible entre chaque réalisation pour garantir que les deux index du participant reviennent en position initiale avant la réalisation suivante.

L'ordre de présentation des facteurs est illustré en Figure 4.19.

Les 12 participants qui ont pris part à l'étude ont un âge moyen de 28.3 [24-38] et incluent 3 femmes. Tous les participants étaient droitiers et chacun a réalisé les deux tâches dans les deux conditions de latence. La moitié des participants ont démarré par la tâche simple, les autres par la tâche composite.

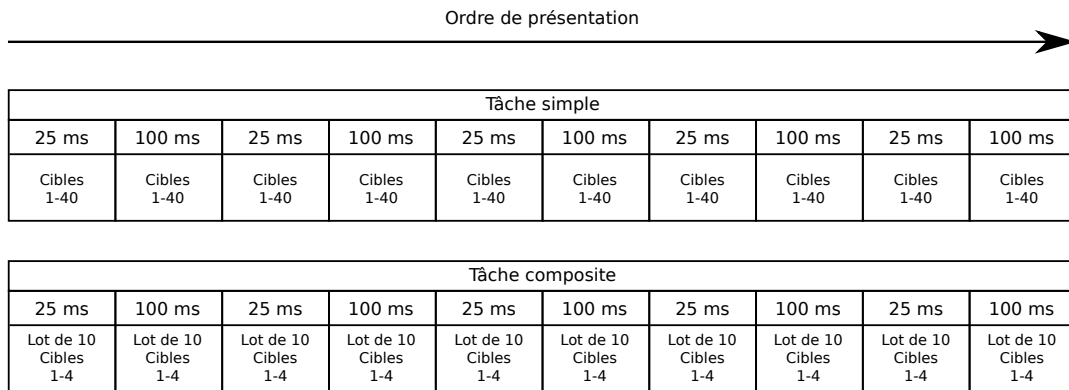


FIGURE 4.19 – Ordre de présentation des différents facteurs pour l'étude sur l'interaction bimanuelle (voir détails dans le texte). La moitié des participants démarrent avec la tâche simple, l'autre avec la tâche composite.

Mesures et analyses

Notre but est de mesurer *l'impact de la latence* selon la TACHE. Les variables habituellement mesurées pour quantifier la performance sur une tâche de positionnement sont le temps passé à déplacer l'objet et le nombre d'erreurs. Le fait que l'on veuille comparer deux tâches requiert un choix délicat de la variable mesurée puisque des tâches différentes amènent intrinsèquement des temps de réalisation différents. Si la même augmentation du temps de réalisation, en secondes, est observée pour une tâche longue ou courte, alors l'impact de la latence est en fait plus fort sur la tâche courte que sur la tâche longue. Pour comparer l'impact de la latence sur différentes tâches de durées variables, nous devons donc mesurer l'augmentation relative du temps de réalisation, en pourcentage, entre la faible et la haute latence.

Pour l'erreur, dans notre cas, mesurer une valeur absolue ou un pourcentage revient au même car le nombre de cible acquises dans chaque condition "LATENCE×TACHE" est identique (200 cibles). Nous mesurons donc l'impact de la latence sur l'erreur comme une différence absolue entre les deux niveaux de latence ($\#$ cible ratées à 100ms - $\#$ cibles ratées à 25ms), que nous comparons ensuite selon la TACHE.

Nos deux variables dépendantes sont donc :

- *L'impact_temps* de la latence : l'augmentation *relative* du temps de déplacement d'un objet quand la latence augmente de 25 ms à 100 ms, mesurée en pourcentage.
- *L'impact_erreur* de la latence : L'augmentation du nombre de cibles ratées quand la latence augmente de 25 ms à 100 ms.

L'effet de TACHE sur les deux variables dépendantes est testé avec des t-tests appariés.

En plus de ces deux tests, nous effectuons aussi une ANOVA à mesures répétées pour vérifier les effets de TACHE et de LATENCE sur le temps de déplacement des objets et sur l'erreur.

Résultats

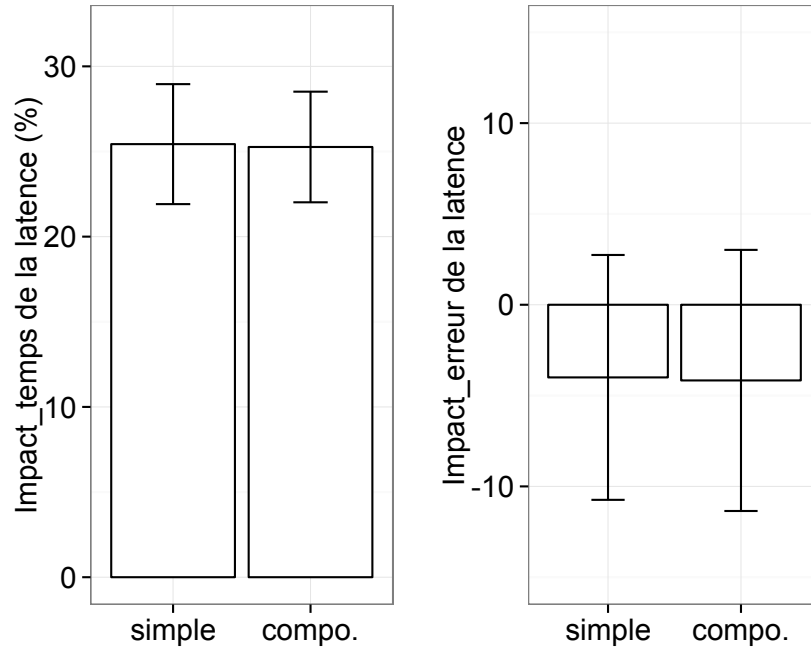


FIGURE 4.20 – *impact_temps* et *impact_erreur* de la latence pour les tâches simple et composite avec les intervalles de confiance à 95% de la moyenne. (Figure traduite de [36])

	<i>impact_temps</i>		<i>impact_erreur</i>	
	simple	composite	simple	composite
moyenne	25.4	25.3	-4.0	-4.2
min	21.1	16.0	-21	-25
max	40.8	34.9	13	13
stddev	5.54	5.11	10.6	11.3

FIGURE 4.21 – Table des valeurs mesurées des variables dépendantes selon le facteur TACHE (simple vs. composite). (Table traduite de [36])

Les statistiques (moyenne, min, max, stddev) de l'*impact_temps* et de l'*impact_erreur* pour les deux tâches sont résumées dans la Figure 4.21.

Le temps moyen de déplacement d'un objet est plus long lors de la tâche composite que lors de la tâche simple (1.44 s contre 0.94 s, $F_{1,11} = 169, p < .001$). Il y a aussi un net effet de la LATENCE sur le temps de déplacement des objets ($F_{1,11} = 255, p < .001$). Pour la tâche simple, l'*impact_temps* moyen est de 25.4% ce qui est cohérent avec les résultats de la littérature [34, 69]. L'*impact_temps* pour la tâche composite est très similaire : 25.3%

(Figure 4.20, gauche). Un t-test ne permet pas de déduire une différence significative de l'*impact_temps* de la latence entre les deux conditions ($t(11) = .097, p = .92$).

Le nombre d'erreurs est significativement plus grand pour la tâche composite que pour la tâche simple (25.6 erreurs contre 20.9 erreurs, $F_{1,11} = 5.4, p = .04$), mais il n'y a pas d'effet significatif de la LATENCE sur le nombre d'erreurs ($F_{1,11} = 2.0, p > .18$) ni d'interaction entre TACHE et LATENCE ($F_{1,11} = .005, p > .94$). Il est alors logique qu'un t-test ne puisse déduire aucune différence de l'*impact_erreur* selon la TACHE ($t(11) = .07, p = .95$).

En résumé, TACHE ne semble pas avoir d'effet ni sur l'*impact_temps* ni sur l'*impact_erreur*. Une absence d'effet ne peut être prouvée statistiquement, mais ces résultats indiquent que même si une différence significative de l'impact de la latence pouvait être observée, la taille d'effet serait petite.

Ces résultats ne vont pas dans le sens de notre hypothèse selon laquelle l'impact de la latence serait moindre pour la tâche composite. Cette influence plus forte qu'attendue de la latence sur la tâche composite pourrait être causée par l'introduction de la main non dominante, peut être plus sensible à la latence que la main dominante.

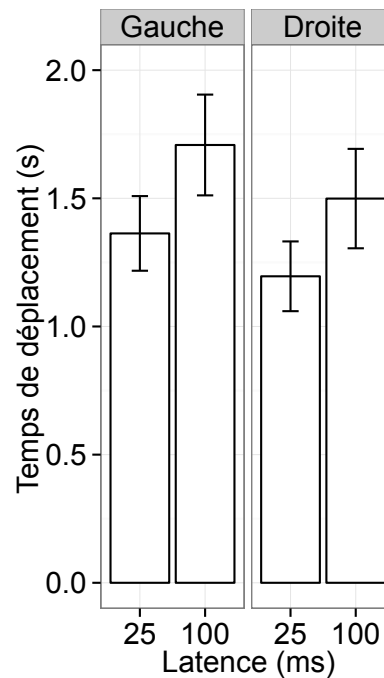


FIGURE 4.22 – Effet de MAIN et LATENCE sur le temps de déplacement pendant la tâche composite. (Figure traduite de [36])

Nous avons donc examiné l'influence de la main utilisée sur les résultats. En ne gardant que les données de la tâche composite, une ANOVA à mesures répétées est réalisée pour comprendre comment les facteurs MAIN et LATENCE influencent le temps de déplacement

des objets. Comme tous nos participants sont droitiers, nous utilisons les valeurs *gauche* et *droite* pour désigner respectivement la main non dominante et dominante. Comme la tâche de positionnement requiert de la précision, le temps de déplacement est plus grand pour la main gauche (1.54 s) que pour la main droite (1.35 s) ($F_{1,11} = 10, p = .009$, Figure 4.22). Plus intéressant, la LATENCE augmente le temps de déplacement pour les deux mains ($F_{1,11} = 142, p < .001$) mais on ne trouve pas d'interaction entre MAIN et LATENCE ($F_{1,11} = 1.3, p > 0.25$). L'effet de la latence sur la performance est donc similaire sur les mains gauche et droite. L'introduction de la main non dominante ne peut donc pas être une explication de l'impact plus fort qu'attendu de la latence sur la tâche composite.

4.3.3 Discussion

La littérature contient plusieurs études qui montrent que la latence réduit la performance des utilisateurs, mais ces études portent uniquement sur des tâches élémentaires réalisées à une main. Des travaux récents ont montré que la perception de la latence par les utilisateurs est réduite lorsque la charge cognitive de la tâche augmente [15]. Dans cette étude, nous avons supposé que quand l'utilisateur perçoit moins la latence, sa performance devrait être moins affectée. Nous avons essayé d'observer cette réduction de l'influence de la latence, mais les résultats n'appuient pas notre hypothèse. Les temps de déplacement d'objets plus longs dans la tâche composite semblent bien indiquer une charge cognitive plus importante que dans la tâche simple. Cependant, la latence a une influence semblable dans les deux tâches : un temps de déplacement qui augmente de 25.4% contre 25.3% lorsque la latence passe de 25 ms à 100 ms.

Nous avons aussi interviewé les participants à la fin de l'étude pour connaître leur avis sur les deux tâches. Nous leur avons demandé de noter quatre phrases sur une échelle de Likert entre 1 (Pas du tout d'accord) et 5 (Tout à fait d'accord).

- J'ai clairement remarqué la différence entre les conditions de latence sur la tâche simple.
- J'ai clairement remarqué la différence entre les conditions de latence sur la tâche composite.
- La latence a fortement affecté ma performance sur la tâche simple.
- La latence a fortement affecté ma performance sur la tâche composite.

Les participants remarquent la différence entre les deux conditions de latence dans les deux tâches, mais perçoivent mieux la latence lors de la tâche simple ($M = 4.83, SD = 0.39$) que lors de la tâche composite ($M = 4.5, SD = 0.52$), $t(11) = 2.35, p = .039$. Cela est cohérent avec les résultats d'Annett et al. sur la perception de la latence [15] : avec une charge cognitive plus grande, la latence devient plus difficile à percevoir. Les participants sentent que la latence affecte leur performance sur la tâche simple ($M = 4.5, SD = 0.80$) et sur la tâche composite ($M = 4, SD = 1.0$) mais la différence entre les deux n'est pas significative ($t(11) = 1.11, p = 0.29$). Les résultats du questionnaire sont

donc cohérents avec les résultats quantitatifs : même si la latence est moins perceptible sur la tâche composite que sur la tâche simple, il n'y a pas de preuve claire d'un effet plus faible sur la performance de la tâche composite.

4.3.4 Bilan

Dans le cadre du diagnostic du phénomène de la latence, cette étude révèle un fort effet de la latence sur des tâches plus complexes et ouvre la voie vers une recherche plus fondamentale sur les aspects nuisibles de la latence. En particulier, nous devons comprendre comment la latence peut garder la même influence sur la performance alors qu'elle est moins bien perçue.

Chapitre 5

Compensation de la latence

Au chapitre 4, nous avons décrit nos contributions en ce qui concerne le diagnostic de la latence et la mise au point d'un dispositif nous permettant de mieux comprendre ce phénomène. Dans ce chapitre, nous nous intéressons plus spécifiquement à la compensation de la latence au niveau logiciel et utilisateur.

Au niveau logiciel, il est possible de compenser la latence avec une prédiction du mouvement de l'utilisateur. Nous avons vu au chapitre 3 que l'utilisation d'une prédiction pour compenser la latence au toucher est un problème ardu car il s'agit de prédire la position du doigt à tout instant tout au long du mouvement. Dans ce chapitre, notre première contribution est l'évaluation d'une prédiction linéaire sur la vitesse du doigt pour réduire les effets de la latence au toucher.

Lorsque les améliorations matérielles et logicielles ne suffisent pas à réduire la latence au point qu'elle n'ait plus d'effet sur les utilisateurs, ceux-ci se servent d'un dispositif imparfait. La compensation de la latence peut alors aussi provenir d'une adaptation motrice humaine. Les effets de la latence que nous avons décrits au chapitre 2 n'ont été évalués que sur de courtes durées d'utilisation. Nous faisons l'hypothèse que les capacités d'adaptation humaines peuvent mener, à plus long terme, à une réduction des effets négatifs de la latence sur la performance. La deuxième contribution de ce chapitre est une étude sur l'adaptation à la latence. L'étude d'un tel phénomène contribue à caractériser l'impact sur la performance motrice d'une exposition prolongée à ce type de perturbation visuomotrice et permet d'émettre des hypothèses sur la façon dont les sujets intègrent ces perturbations dans le contrôle moteur de la tâche à exécuter. Une telle étude indique aussi des pistes à suivre en terme de conception des dispositifs tactiles pour minimiser les effets de la latence.

Ce chapitre répond aux questions suivantes :

- Dans quelle mesure une prédiction linéaire compense-t-elle les effets négatifs de la latence ?
- L'utilisateur développe-t-il, avec de l'entraînement, des mécanismes pour minimiser les effets de la latence sur sa performance ?

5.1 Compensation logicielle : utilisation d'une prédiction

Cette section reprend nos travaux publiés à la conférence Interactive Tabletops and Surfaces (ITS) en 2015 : “Reducing latency with a continuous prediction : Effects on users’ performance in direct-touch target acquisitions” [34].

5.1.1 La prédiction, une question de compromis

Nous avons vu au chapitre 3 que la compensation de la latence par la seule voie matérielle est souvent faite au dépens d'autres qualités du système (par exemple, sa consommation énergétique, ou la taille de l'affichage). Il faut donc aussi développer des solutions logicielles afin de minimiser les effets négatifs de la latence. Notamment, il est possible d'utiliser une prédiction : lors de l'interaction au toucher, l'objet manipulé n'est alors plus affiché à la plus récente position connue du doigt, mais à une position prédite en fonction de la trajectoire du doigt (Figure 5.1). Des prédictions ont déjà été utilisées pour inférer l'objectif final du mouvement d'un utilisateur (cf. chapitre 3). Ces techniques peuvent être employées pour supprimer la latence du premier contact [155]. Ici, nous nous attaquons au problème plus complexe du déplacement du doigt sur l'écran. Dans ce cas, un objet est déplacé par le doigt, et la prédiction doit alors agir de manière *continue* pour réduire l'écart entre l'objet et le doigt. Prédire la trajectoire d'un mouvement humain est une tâche complexe et peut introduire elle-même une nouvelle source d'erreur entre la position de l'objet et celle du doigt. Une approche prédictive n'est donc efficace que si le bénéfice de la réduction des effets de la latence dépasse les nouveaux effets négatifs potentiellement introduits par la prédiction.

Certaines tablettes comme la Surface Pro 4 de Microsoft [8] ou l'iPad Pro d'Apple [140] utilisent déjà une prédiction pour compenser la latence de leur système. Cependant, le choix du modèle de prédiction est inconnu, la longueur de la prédiction est limitée (16 ms [140]) et on ne connaît pas les effets de la prédiction sur les utilisateurs. À notre connaissance, aucune étude antérieure à nos travaux ne fournit une évaluation objective des bénéfices apportés par une telle prédiction sur la performance des utilisateurs.

Nous avons donc étudié la compensation de la latence d'un système grâce à une prédiction. La prédiction rapproche le retour visuel du doigt de l'utilisateur, mais plus on prédit loin dans le temps, plus on est susceptible d'introduire du bruit dû à l'inexactitude de la prédiction. Il s'agit donc de trouver un compromis entre longueur de prédiction et bruit qui permette d'améliorer la performance et le confort des utilisateurs. Dans notre étude, la mesure des bénéfices de la prédiction s'est faite en évaluant la performance des utilisateurs sur des tâches de positionnement d'objets. Nous avons utilisé notre système faible latence (décrit au chapitre 4) pour étudier la prédiction à quatre niveaux de latence : 25 ms, 42 ms, 58 ms et 75 ms, en essayant à chaque fois de compenser totalement la latence (longueur de prédiction = latence du système).

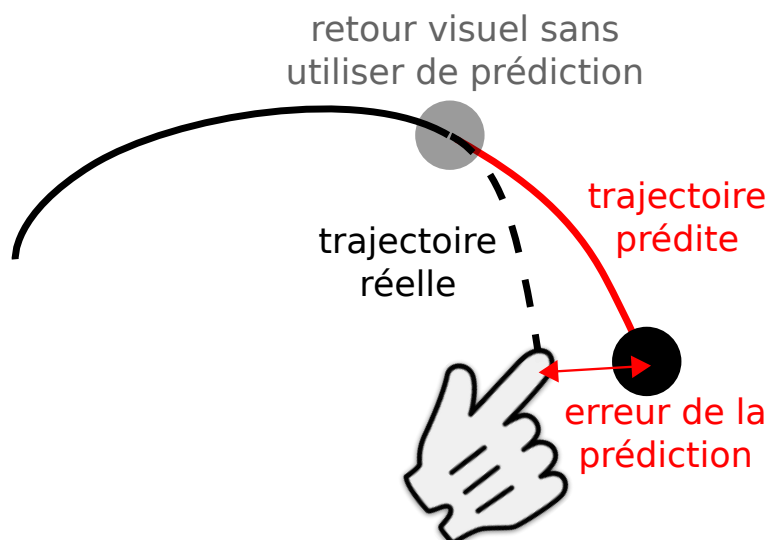


FIGURE 5.1 – Utilisation d'une prédiction pour compenser la latence. A un instant donné, le système connaît seulement une partie de la trajectoire du doigt (trait noir continu) à cause de la latence. Grâce à une prédiction de la trajectoire future, on affiche le retour visuel plus proche du doigt. Mais si la prédiction ne correspond pas exactement à la trajectoire réelle (pointillés noirs), elle crée une erreur qui perturbe l'utilisateur. Une prédiction utile est une prédiction dont l'effet de l'erreur est moins néfaste que l'effet de la latence seule.

Nous observons en particulier les effets de la prédiction sur une tâche de base pour l'interaction au toucher : le déplacement d'un objet vers une cible. Dans la suite, nous appelons cette tâche "tâche de pointage" par similarité avec les tâches de ce type utilisées en IHM pour l'évaluation de différents dispositifs.

5.1.2 Choix d'un modèle de prédiction

Jota et al. (2013) ont montré que la latence au toucher affecte la performance principalement dans la dernière phase du mouvement de pointage, lors des ajustements fins sur la cible [69]. Cela pourrait s'expliquer par le fait que pour la première phase, balistique, l'utilisateur se contente du retour visuel de son doigt pour contrôler son mouvement, la position de l'objet ne devenant nécessaire que lors du positionnement précis sur la cible. Pour mieux comprendre ce phénomène, nous avons étudié une vingtaine de trajectoires de différents participants pointant sur des cibles petites et distantes (taille=0.42 cm, amplitude=33.2 cm).

Nous avons regardé et annoté à la main sur les profils de vitesse la durée des mouvements d'ajustement dans la phase finale. Nous avons observé que des mouvements de moins de 70 ms, presque saccadés étaient assez fréquents, mais erratiques et donc semblant imprévisibles. Ces mouvements sont entrecoupés de périodes d'immobilité de durées qui semblent aussi imprévisibles (Figure 5.2). Or, la prédiction est surtout nécessaire dans cette dernière phase erratique du mouvement. La quantité de latence qu'on peut compen-

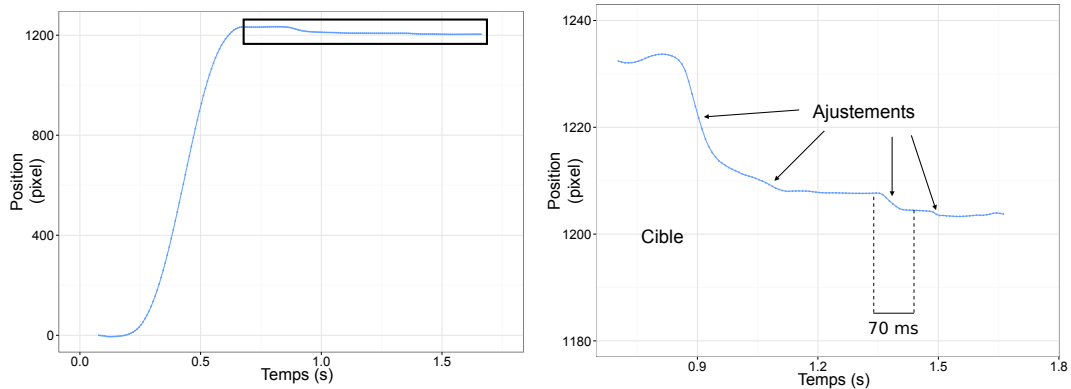


FIGURE 5.2 – Un exemple d’une trajectoire de pointage en une dimension. La première partie du mouvement (à gauche) est lisse et régulière. C’est la phase balistique, amenant le doigt dans le voisinage de la cible, sans nécessairement considérer l’objet à déplacer. La partie finale du mouvement (encadrée à gauche et agrandie à droite) est composée de plusieurs ajustements pour atteindre la zone de la cible et nécessite le retour visuel de l’objet. Ces ajustements sont souvent de durées inférieures à 70 ms.

ser est donc bornée par la durée de ces mouvements d’ajustement. Un système avec 70 ms de latence ne sera informé du début de tels mouvements qu’alors que ceux-ci seront déjà terminés, ce qui rend leur prédiction impossible.

Ces observations induisent qu’un bon système de prédiction doit être le plus réactif possible et donc ne s’appuyer que sur les données les plus récentes, ce qui correspond aux approches utilisées en réalité virtuelle [86]. Nous avons d’abord considéré un modèle parabolique (les coordonnées x et y sont modélisées de manière indépendante) en extrapolant les trois dernières positions connues du doigt. Pour évaluer la précision de la prédiction, nous avons comparé la trajectoire prédite à une trajectoire “physique” obtenue à partir des données retardées en ôtant le délai du système à la date de chaque donnée (ce qui revient à translater la courbe des données retardées selon l’axe temporel). Nous avons utilisé une latence de 75 ms pour amplifier les effets de la prédiction. Les résultats sont illustrés sur la Figure 5.3.

Une première version du modèle parabolique utilise un minimum de trois points, échantillonnés à 120 Hz, pour réaliser la prédiction. La Figure 5.3-A montre des oscillations de la prédiction qui sont le résultat du bruit des capteurs dans le modèle de prédiction utilisé. Cette instabilité peut être réduite par l’ajout de données (Figure 5.3-B, avec 10 échantillons). Ajouter des points lisse la prédiction mais n’empêche pas de fortes erreurs. De plus, chaque point de donnée rajouté introduit de l’information de plus en plus vieille et donc de moins en moins pertinente pour prédire la suite du mouvement. En d’autres termes, lisser sur des points plus anciens augmente l’inertie et réduit la réactivité de la prédiction.

Nous avons aussi évalué un modèle de prédiction linéaire du premier ordre qui fait

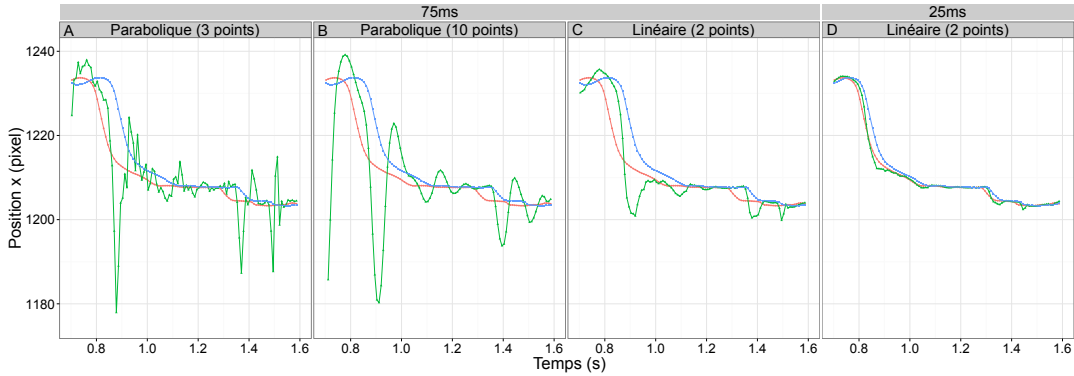


FIGURE 5.3 – Les trajectoires générées par différents modèles de prédiction pour la dernière phase d'un mouvement de pointage avec 75 ms de latence (A, B, C) et 25 ms de latence pour le modèle linéaire (D). En bleu, la trajectoire retardée. En rouge, la trajectoire du doigt, obtenue par translation de la courbe bleue. En vert, la trajectoire prédite. (Figure traduite de [34])

L'hypothèse d'une vitesse constante. L'avantage d'utiliser un modèle du premier ordre est qu'il ne requiert que deux points de données pour produire une estimation et il est donc réactif. De plus, il ne nécessite qu'une étape d'intégration sur le temps, et ainsi, amplifie moins le bruit des capteurs. Les résultats de ce modèle sont de meilleure qualité que ceux du modèle de second ordre.

Ce modèle de prédiction linéaire du premier ordre a donc été retenu pour nos études de la compensation de la latence par prédiction. Nous utilisons des longueurs de prédiction inférieures ou égales à 75 ms et nous essayons de compenser entièrement la latence du système, c'est-à-dire que nous faisons une prédiction à X ms si le système a X ms de latence. La latence l du système a été mesurée et régulièrement contrôlée par la méthode "high accuracy" proposée par Bérard et Blanch [19]. Nous avons observé une certaine variabilité de la latence (± 8 ms) autour de la valeur moyenne. Cependant, nous n'avons pas de méthode qui puisse estimer la latence à tout instant afin d'ajuster la prédiction et nous avons considéré la latence comme constante. La fréquence de notre système faible latence est très stable et nous avons donc aussi considéré la période d'échantillonnage Δt constante.

La position 2D *prédite* du doigt $\hat{\mathbf{x}}_i$ au cycle i est donc calculée par la formule suivante :

$$\hat{\mathbf{x}}_i = \mathbf{x}_i + l * \hat{\mathbf{s}}_i \quad \hat{\mathbf{s}}_i = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{\Delta t} \quad (5.1)$$

\mathbf{x} est la position *observée* du doigt, $\hat{\mathbf{s}}$ et l'estimation de la vitesse, l (exprimée en ms) est la latence du système et $\Delta t = 8.33$ ms. Un avantage de ce modèle est qu'il est indépendant de l'utilisateur et de la position des cibles potentielles du mouvement. Il est donc possible de le mettre en œuvre facilement sur la plupart des systèmes tactiles.

Pour évaluer la précision de ce modèle de prédiction, nous avons comparé l'erreur

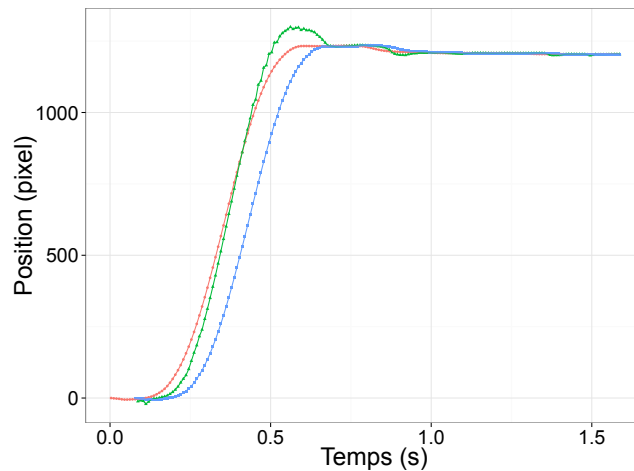


FIGURE 5.4 – Trajectoire du doigt (rouge), trajectoire retardée de 75 ms (bleue) et trajectoire prédite (verte) sur l'intégralité d'un mouvement de pointage avec le modèle de prédiction linéaire. Globalement, la trajectoire prédite est plus proche de celle du doigt que la trajectoire retardée mais elle introduit une perturbation avec une sorte d'oscillation autour de la trajectoire du doigt.

de position moyenne d'une part entre la trajectoire physique du doigt et la trajectoire retardée et, d'autre part, entre la trajectoire physique du doigt et la trajectoire prédite (Figure 5.4). Cela revient à calculer l'écart moyen entre l'objet déplacé et le doigt dans les cas avec ou sans prédiction. Les erreurs calculées découlent d'une moyenne sur 7428 trajectoires enregistrées pendant l'étude utilisateur décrite dans la section suivante. Les résultats sont visibles sur la Figure 5.5.

L'erreur moyenne diminue de 0.66 cm à 0.11 cm à 25 ms de latence et de 1.6 cm à 0.61 cm à 75 ms, soit une amélioration respective de 84% et 63%. Le modèle de prédiction utilisé semble donc performant mais il faut pondérer ce résultat par ses effets sur la trajectoire (Figures 5.3-C-D et 5.4). La prédiction sous-estime la position réelle du doigt lors des phases d'accélération et la surestime lors des phases de décélération. Cela se voit particulièrement pour la condition à 75 ms de latence pour les ajustements finaux sur la cible. Dans ce cas particulier, prédire, à partir de positions antérieures, des corrections qui évoluent rapidement apparaît très difficile, la prédiction introduisant plus de perturbations que ne le fait la trajectoire retardée. La prédiction rapproche donc le retour visuel du doigt de l'utilisateur mais donne aussi une trajectoire plus chaotique au retour visuel, notamment dans la phase de correction.

Nous avons donc mené une étude utilisateur pour évaluer les effets de notre modèle de prédiction linéaire sur la performance des utilisateurs.

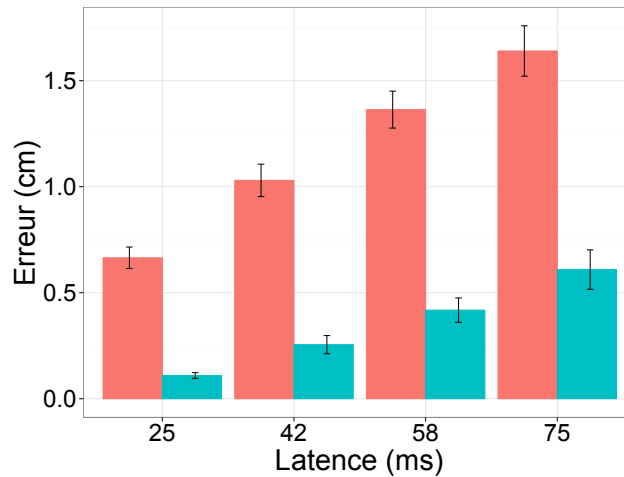


FIGURE 5.5 – Erreur moyenne en centimètre entre l’objet manipulé et le doigt, avec l’utilisation de la prédiction (en bleu) ou non (en rouge), calculé sur un ensemble de 7428 trajectoires. (Figure traduite de [34])

5.1.3 Evaluation de la prédiction linéaire - Etude utilisateur

Le but de cette étude est d’évaluer à quel point une approche prédictive contrebalance les effets négatifs de la latence au toucher sur une tâche de pointage. Plus précisément, nous souhaitons déterminer les niveaux de latence pour lesquels l’utilisation du modèle de prédiction linéaire améliore l’évaluation quantitative (performance) et qualitative (préférence) des comportements utilisateurs. Un but secondaire de notre étude est d’étendre les résultats de Jota et al. [69]. Ces auteurs ont montré l’effet négatif de la latence au toucher sur le positionnement d’objet, mais seulement pour des indices de difficulté de Fitts (cf. chapitre 2) inférieurs à 3 alors que l’intervalle d’ID conseillé pour le pointage est entre 2 et 8 [132]. Notre étude inclut des indices allant jusqu’à 6.34.

Nous avons invité 24 volontaires provenant de notre environnement universitaire (âge=27.4 [22,39], 8 femmes) à prendre part à l’étude. Les participants commencent par une courte session de familiarisation (10-15 min) avec le dispositif et la tâche de positionnement. Ensuite, ils réalisent des tâches de positionnement d’indices de difficulté variables sur lesquelles leur performance est évaluée (25 min). Finalement, on demande aux participants d’exprimer leurs préférences en testant un déplacement d’objet avec ou sans prédiction, avec différents niveaux de latence.

Protocole de la phase d’évaluation de performance

La tâche sur laquelle est évaluée la performance des participants est une tâche de pointage réalisée avec notre système faible latence (cf. chapitre 4). Les participants doivent toucher un objet représenté par un disque rouge (rayon=1.38 cm) et l’amener sur une cible représentée par un disque blanc de rayon supérieur. La taille de la cible, en terme d’indice



FIGURE 5.6 – Un utilisateur en train de réaliser la tâche de pointage. L’essai se termine avec succès si le disque rouge est relâché alors qu’il est totalement compris dans la cible blanche. (Photo issue de [34])

de difficulté de Fitts, est la différence entre le diamètre du disque cible et le diamètre du disque objet, et varie selon la difficulté de la tâche. L’affichage présenté aux utilisateurs est visible en Figure 5.6.

Lorsque les participants lâchent l’objet (c’est-à-dire qu’ils lèvent le doigt de la surface de l’écran), la tâche se termine. La cible devient verte (resp. rouge) si l’essai est réussi (resp. raté), c’est-à-dire si l’objet est relâché alors qu’il est totalement compris dans la cible. Après un délai aléatoire compris entre 0.2 s et 1.2 s, la cible suivante apparaît. L’imprévisibilité de ce délai empêche les participants d’anticiper leur mouvement vers la cible suivante. Entre deux essais, l’objet reste à la position où le participant l’a lâché.

Nous avons demandé aux participants d’être aussi rapides que possible tout en essayant de limiter leur taux d’erreur à “une erreur entre chaque pause” ce qui correspond à un taux de 5%. Toutes les cibles sont situées sur la ligne horizontale au milieu de l’écran pour éviter au maximum les problèmes d’occultation. Toutes les 20 cibles, le message “Vous pouvez prendre une pause” s’affiche. La pause se termine lorsque le participant reprend l’objet.

Le plan d’expérience comprend trois facteurs :

- PREDICTION. Deux niveaux : *faux* (pas de prédiction) ou *vrai* (avec prédiction). Quand *vrai*, la longueur de prédiction est égale à la latence. Afin de limiter le nombre de conditions de l’étude, nous n’avons pas considéré les compensations partielles.
- LATENCE. Quatre niveaux : 25 ms, 42 ms, 58 ms et 75 ms. Le niveau le plus bas est défini par la latence minimale de notre système. Pour la valeur haute, 75 ms correspond à la durée typique des mouvements correctifs que nous avons observés lors de tâches de positionnement.
- ID. Quatre niveaux mesurés en bits (avec distance et largeur de cible en cm) : 2.32 ($d=11$, $w=2.77$), 3.64 ($d=22.14$, $w=1.94$), 5.04 ($d=22.14$, $w=0.69$), 6.34 ($d=33.2$, $w=0.42$). Ces ID couvrent donc des petites et grandes cibles à distances proches ou

éloignées.

L'ordre de présentation des facteurs est illustré en Figure 5.7. Au total, l'étude comprend donc 32 conditions (2 PREDICTION \times 4 LATENCE \times 4 ID). Avec 20 répétitions pour chaque condition, soit 15360 tâches réalisées (24 participants \times 32 conditions \times 20 répétitions). La présentation de PREDICTION et LATENCE est équilibrée sur les différents participants pour contrebalancer tout effet d'ordre. A l'intérieur de chaque bloc PREDICTION \times LATENCE, la présentation des 80 cibles (4 ID \times 20 répétitions) est pseudo aléatoire, le même ordre aléatoire étant utilisé pour chaque bloc et pour tous les participants. Les conditions sont ordonnées d'abord par PREDICTION puis par LATENCE.

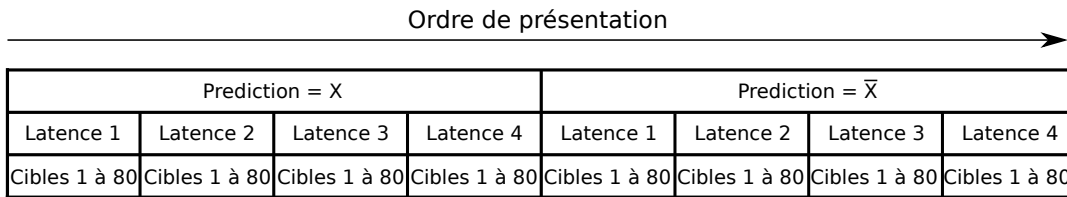


FIGURE 5.7 – Ordre de présentation des différents facteurs pour l'étude sur la prédiction (voir détails dans le texte).

Mesures et analyses

Pour chaque sujet et condition, nous avons retiré les essais dont le temps de positionnement est à plus de 1.5 fois l'écart interquartile en dessous du premier quartile ou au-dessus du troisième quartile, ce qui représente 3.125% de tous les essais.

Pour chaque sujet et condition, trois variables dépendantes sont mesurées :

- *temps* (en secondes) : la durée entre le moment où l'objet est touché et celui où il est relâché.
- *erreur* (en pourcentage) : le ratio du nombre d'échecs sur le nombre total d'essais.
- *TP* : ou "throughput", la bande passante, comme définie par Soukoreff et MacKenzie [132], en bits par seconde (cf. chapitre 2 "Pointage et loi de Fitts"). En particulier, nous avons calculé les IDE pour pouvoir établir TP.

Les effets de PREDICTION, LATENCE et ID et leurs interactions sont étudiés avec des ANOVAs intra-sujet. Une diminution de *temps* et *erreur* ou une augmentation de *TP* traduisent des effets positifs des différents facteurs et inversement. Des tests de Student appariés sont utilisés pour une analyse Post-hoc pour tester nos deux hypothèses : (1) un effet négatif de la latence sur la performance des utilisateurs et (2) un effet positif de la prédiction pour des latences assez faibles.

Résultats

Pour comparer notre étude avec les résultats de la littérature [93] ainsi que pour tester les effets de la latence sur un intervalle d'ID plus large que dans Jota et al. [69],

	Sans prédiction				Avec prédiction			
	25 ms	42 ms	58 ms	75 ms	25 ms	42 ms	58 ms	75 ms
<i>temps</i> (s)	0.92	0.98	1.02	1.10	0.86	0.93	1.03	1.14
<i>erreur</i> (%)	4.77	4.56	4.91	3.86	6.70	9.54	12.44	14.63
<i>TP</i> (bit/s)	5.01	4.79	4.45	4.16	5.16	4.79	4.18	3.65

FIGURE 5.8 – Table des valeurs mesurées des variables dépendantes selon les facteurs PREDICTION et LATENCE.

nous avons analysé les effets de LATENCE et ID sur *temps*, *erreur* et *TP*. Cette première analyse ne concerne que le sous-ensemble de données où PREDICTION=*faux*. Comme les travaux auxquels nous nous comparons n'avaient pas de condition avec prédiction dans leur protocole, dans un premier temps, nous avons contrôlé l'effet de l'ordre de présentation de PREDICTION en ajoutant ce facteur inter-sujets aux ANOVA. L'effet de l'ordre de présentation sur *temps*, *erreur* et *TP* n'est pas significatif et n'interagit pas avec les effets de LATENCE et ID. Nous avons donc fusionné les deux groupes indépendamment de l'ordre de présentation de PREDICTION pour les analyses qui suivent. Dans la suite, ce sous-ensemble de données est appelé "les données sans prédiction". Des ID croissants font augmenter *temps* et *erreur* de manière très significative ($p < .001$) dans toutes les conditions de latence. Ce résultat est peu surprenant puisqu'il confirme simplement la loi de Fitts. En conséquence, les analyses qui suivent se concentrent plutôt sur les effets de LATENCE et son interaction avec ID.

La moyenne sur les ID des variables dépendantes pour les différentes conditions de LATENCE et de PREDICTION sont répertoriées sur la Figure 5.8.

La variable *temps* augmente significativement avec LATENCE ($F_{3,69} = 13.6, p < .001$, Figure 5.9) entre 25 ms et 42 ms (+61 ms, $t(23) = 2.9, p = .017$) et entre 58 ms et 75 ms (+80 ms, $t(23) = 3.1, p = .005$). La différence entre 42 ms et 58 ms n'est pas significative (+40 ms, $t(23) = 1.3, p > .2$). Les effets de LATENCE sur *temps* sont aussi plus grands pour les ID élevés que pour les bas ($F_{9,207} = 4.7, p < .001$) : les pentes d'une régression linéaire de *temps* en fonction d'ID augmentent avec LATENCE ($F_{3,69} = 7.3, p < .001$) entre 25 ms et 42 ms ($t(23) = 2.4, p = .023$) et entre 58 ms et 75 ms ($t(23) = 2.6, p = .015$), la différence de pente entre 42 ms et 58 ms n'est pas significative ($t(23) = 0.1, p > .9$). *TP* décroît quand LATENCE augmente ($F_{3,69} = 21.0, p < .001$, Figure 5.12, points de gauche), avec un changement significatif entre 25 ms et 42 ms (-0.22 bit/sec, $t(23) = 2.1, p < .05$), entre 42 ms et 58 ms (-0.34 bit/sec, $t(23) = 3.0, p < .006$) et entre 58 ms and 75 ms (-0.28 bit/sec, $t(23) = 3.2, p < .005$). Il n'y a pas d'effet global de LATENCE sur *erreur* ($F_{3,69} = 0.99, p > .4$, Figure 5.11, points de gauche), ni d'interaction entre LATENCE et ID ($F_{9,207} = 1.7, p > .1$).

Les effets de la prédiction sur la performance des participants ont ensuite été analysés

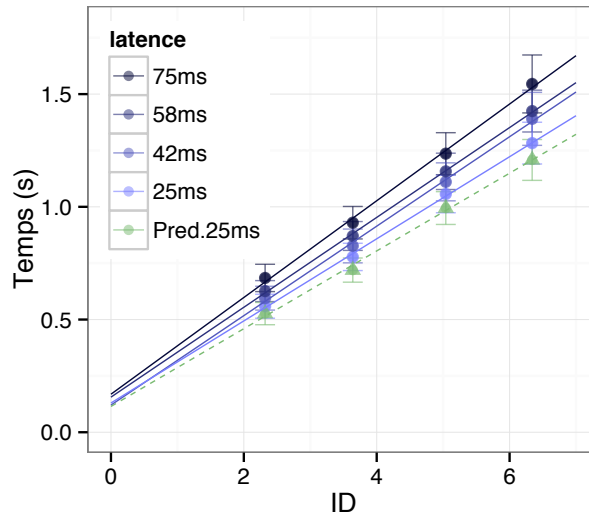


FIGURE 5.9 – Effet de la latence sur le temps d’acquisition de cible quand aucune prédiction n’est utilisée. La condition à 25 ms avec prédiction est ajoutée pour comparaison. Les points représentent la valeur moyenne pour les 24 participants et les barres d’erreur donnent les intervalles de confiance à 95% de la moyenne. Les lignes sont les régressions linéaires dans chaque condition de latence. (Figure traduite de [34])

en utilisant l’ensemble des données. Les effets de ID et LATENCE sont comparables à ceux observés sur les données “sans prédiction”. Les résultats suivants se focalisent donc sur les effets liés à PREDICTION et sont représentés sur la Figure 5.10.

Il n’y a pas d’effet significatif global de PREDICTION sur *temps* ($F_{1,23} = 0.4, p > .5$), mais cela est dû aux différents effets de PREDICTION selon LATENCE et ID ($F_{9,207} = 4.0, p < .001$, Figure 5.10). PREDICTION a un effet négatif à 75 ms, neutre à 58 ms et un effet positif à 25 ms et 42 ms. Pour mieux comprendre ces interactions, des ANOVAs indépendantes sont réalisées pour chaque condition de LATENCE avec PREDICTION et ID en facteurs. A 25 ms de latence, le *temps* moyen sur les IDs et les participants est réduit significativement de 6.2% en utilisant la prédiction (-57 ms, $F_{1,23} = 6.4, p = .02$). A 42 ms, il y a une tendance, mais non significative ($F_{1,23} = 3.9, p = .06$) et il n’y a pas d’effet de PREDICTION détecté ni à 58 ms ni à 75 ms ($F_{1,23} < 1.1, p > .3$ dans les deux cas). L’effet de PREDICTION ne dépend pas clairement de ID à 25 ms ($F_{3,69} = 1.8, p = .16$), alors qu’il tend à augmenter avec les ID plus élevés pour le trois autres niveaux de LATENCE ($F_{3,69} > 3.2, p < .03$).

L’influence de PREDICTION et LATENCE sur *erreur* est illustrée en Figure 5.11. *erreur* est significativement touchée par PREDICTION ($F_{1,23} = 41.6, p < .001$). Cependant, PREDICTION a des effets différents sur *erreur* selon ID ($F_{3,69} = 29.9, p < .001$) et LATENCE ($F_{3,69} = 12.7, p < .001$). Des ANOVAs indépendantes pour chaque niveau de latence montrent que, à 25 ms, *erreur* n’est pas significativement dépendante de PREDICTION ($F_{1,23} > 3.3, p = .08$). Les effets de PREDICTION sur *erreur* sont en revanche

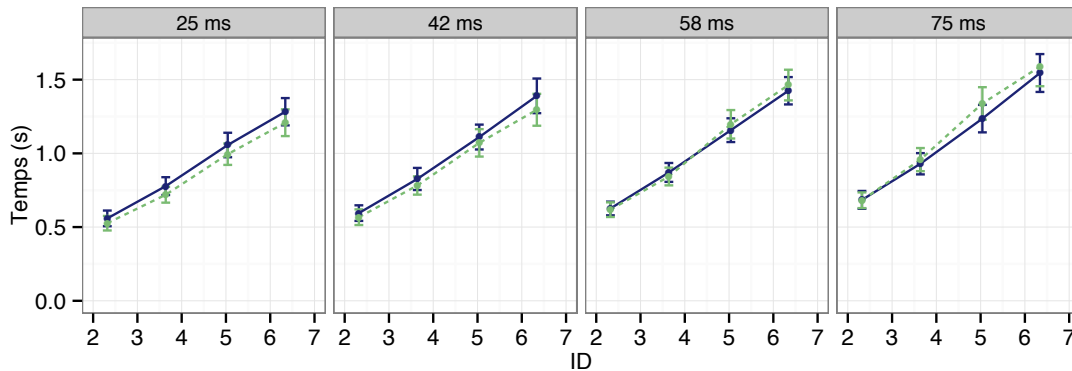


FIGURE 5.10 – Temps d’acquisition de cible (*temps*) avec PREDICTION *vrai* (vert) ou *faux* (bleu) pour les différents ID et les différents niveaux de latence. Les barres d’erreur donnent les intervalles de confiance à 95% de la moyenne sur l’ensemble des participants. (Figure traduite de [34])

significatifs à 42 ms, 58 ms et 75 ms ($F_{1,23} > 18.5, p < .001$).

L’influence de PREDICTION et LATENCE sur *TP* est illustrée en Figure 5.12. Les effets de LATENCE sur *TP* sont significativement différents selon PREDICTION ($F(3, 69) = 9.3, p < .001$). La prédiction augmente la bande passante à 25 ms (+0.15) mais pas significativement ($t(23) = 1.4, p > .17$), à 42 ms, la bande passante reste stable. La prédiction diminue la bande passante à 58 ms et 75 ms ($t(23) > 2.5, p < .03$).

Etude qualitative

La performance objective sur une tâche n’est pas l’unique critère d’évaluation pour concevoir une bonne interaction : le confort de l’utilisateur est aussi un aspect important. Nous avons donc évalué les préférences des participants lors de l’interaction avec ou sans la prédiction.

A chaque essai, les participants interagissent avec le disque virtuel dans deux conditions A et B. Les deux conditions ont la même latence, mais l’une utilise de la prédiction, l’autre non. L’assignation de la prédiction à A ou B est faite aléatoirement. Comme dans la première partie de l’étude, la prédiction est configurée pour compenser totalement la latence du système. En appuyant sur la barre espace du clavier, les participants peuvent, autant de fois qu’ils le veulent, changer entre les conditions A et B. La fin de chaque essai se fait quand le participant le souhaite, lorsqu’il exprime quelle condition, A ou B, il trouve “la plus confortable”. Les participants sont contraints de faire un choix, même s’ils trouvent la différence entre les deux conditions difficile à percevoir.

Comme les préférences peuvent dépendre de la tâche réalisée, la session est divisée en deux parties. Sur les quatre premiers essais, les quatre niveaux de latence sont testés sans instruction de tâche particulière : les participants peuvent explorer librement l’interaction avec le disque. Lors des seize choix suivants, les participants doivent réaliser une tâche de

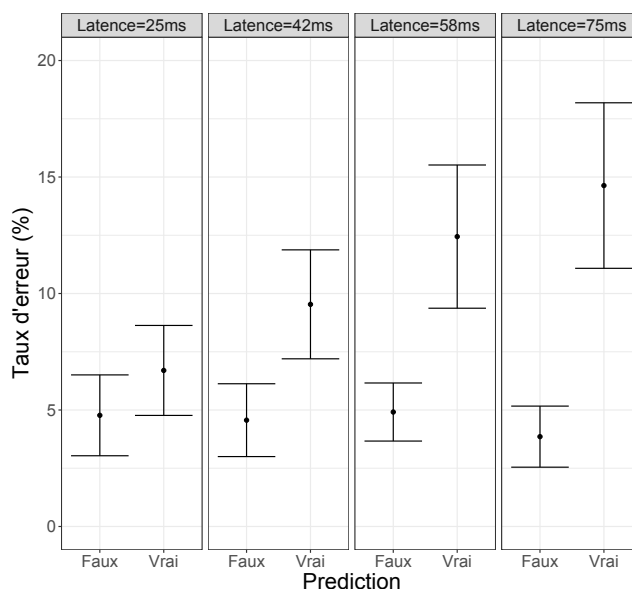


FIGURE 5.11 – *erreur* moyenne sur les participants et les ID pour les différentes valeurs de LATENCE et PREDICTION, et intervalles de confiance à 95% de la moyenne.

positionnement identique à celles utilisées dans la première partie, avec les 4 IDs et les 4 latences possibles. L'ordre des latences est équilibré sur les participants. Pour une latence donnée, les ID sont présentés dans l'ordre suivant : 2.32, 5.04, 3.64, 6.34.

Pour chaque condition de tâche et de latence, nous mesurons la proportion de participants ayant préféré la condition avec la prédiction. Les résultats sont illustrés sur la Figure 5.13. Chaque score de préférence est vérifié avec un test du χ_2 , pour tester si l'hypothèse nulle d'un choix aléatoire entre les deux conditions peut être rejetée. Lors de la tâche de positionnement d'objet, les participants ont préféré la condition sans prédiction pour les trois latences les plus hautes ($\chi_2(1, 96) > 13, p < .001$), mais l'hypothèse d'un choix aléatoire ne peut être rejetée à 25 ms ($\chi_2(1, 96) = 0.67, p > 0.41$). Pour l'interaction libre, 17 participants sur 24 (70.8%) ont préféré la condition avec prédiction à 25 ms de latence ($\chi_2(1, 24) = 4.17, p < 0.05$) alors que le choix aléatoire ne peut être rejeté pour les trois latences plus élevées ($\chi_2(1, 24) < 1.50, p > 0.22$).

5.1.4 Discussion

Effet de la latence sur la performance

L'analyse des données "sans prédiction" a permis la réplification des résultats de la littérature qui montraient le fort impact négatif de la latence sur la performance de l'utilisateur lors de tâches de positionnement d'objets [69]. La latence agit comme un facteur multiplicatif de l'indice de difficulté : lorsque la latence augmente, la bande passante diminue et la pente des droites de régression sur le temps d'acquisition augmentent, montrant la dé-

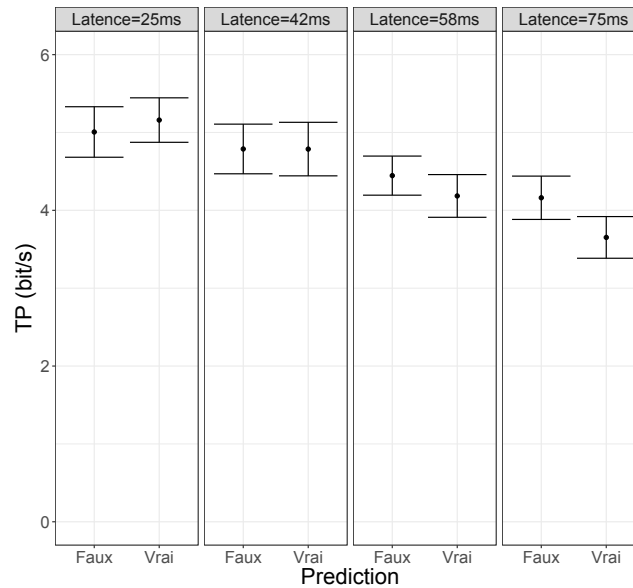


FIGURE 5.12 – TP moyen sur les participants pour les différentes valeurs de LATENCE et PREDICTION, et intervalles de confiance à 95% de la moyenne.

gradation de performance (Figure 5.9). La performance s'améliore continûment lorsque la latence décroît jusqu'à notre valeur minimale de 25 ms. Cela est cohérent avec les résultats de Jota et al. [69], que nous étendons pour l'acquisition de cibles avec des ID supérieurs à 2.6 bits, jusqu'à 6.3 bits. Le fait que les résultats soient cohérents avec les travaux antérieurs accreditte la validité de notre système de contrôle de latence et la validité de notre protocole expérimental.

Compenser la latence par une prédiction

Utiliser la prédiction avec des latences supérieures à 42 ms n'a pas amélioré la performance des utilisateurs. A 58 ms et 75 ms de latence la prédiction augmente même le temps d'acquisition pour les deux cibles avec les deux ID les plus élevés. La petite taille de ces cibles (0.7 cm et 0.4 cm) requiert un contrôle précis et davantage d'ajustements difficilement prévisibles lors de la phase finale de l'acquisition par rapport à des grandes cibles. Ces résultats renforcent ceux des études précédentes suggérant que la latence a une influence plus forte lors de cette dernière phase [54, 69]. De plus, notre étude révèle que le seuil de longueur de prédiction en dessous duquel les performances sont améliorées est plutôt bas, en tout cas lorsqu'il s'agit d'effectuer une tâche qui nécessite un positionnement précis. Une prédiction de 42 ms pourrait sembler être une faible valeur comparée aux longueurs de prédiction utilisées pour estimer le point final d'un mouvement de pointage [155]. Cependant, la prédiction du point d'arrivée se base essentiellement sur le mouvement initial de l'acquisition, qui est bien plus régulier que les mouvements d'ajustement finaux. De plus cette prédiction vise souvent à déterminer une zone d'arrivée, la cible, et non un point

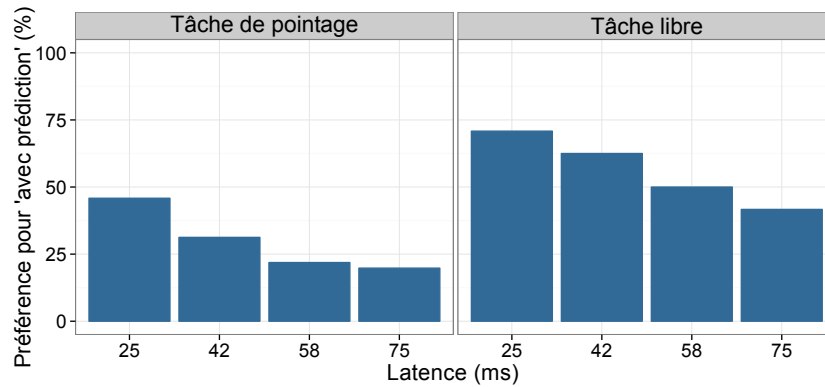


FIGURE 5.13 – Proportion de participants ayant préféré la condition “avec prédiction” pendant l’étude qualitative. (Figure traduite de [34])

précis. La précision requise est donc moindre comparée à notre objectif de compensation de la latence.

En incluant dans notre étude des ID élevés, avec des petites tailles de cible, nous avons probablement sélectionné des tâches parmi les plus sensibles aux erreurs de prédiction. D’autres tâches couramment utilisées en toucher direct comme défiler dans un menu ou naviguer sur une carte ne demandent pas une précision inférieure au centimètre. Notre étude peut donc être considérée comme un des pires cas d’utilisation d’une prédiction en interaction au toucher. D’autre part, cette étude utilise la première version du système faible latence décrit au chapitre 4. Notons que dans cette première version, le seuil de détection du toucher sur l’écran est défini de manière uniforme pour l’ensemble de l’écran ce qui mène à des imprécisions. Des mouvements parasites lors du lâcher du disque augmentent probablement le taux d’erreur. De plus, la prédiction amplifie les mouvements parasites. Sans la présence de ceux-ci, l’écart de taux d’erreur entre les conditions avec ou sans prédiction serait certainement moindre.

Même avec des positionnements d’objets précis, la prédiction a amélioré la performance des utilisateurs lorsque la latence du dispositif était à 25 ms. Pour évaluer l’importance de cette amélioration, nous avons utilisé les données “sans prédiction”. Sur la Figure 5.9, l’amélioration entre 42 ms et 25 ms sans prédiction est similaire à celle entre 25 ms sans prédiction et 25 ms avec la prédiction. En première approximation, on peut donc estimer que la prédiction a eu un effet bénéfique similaire à la réduction de latence de 42 à 25 (17ms). Pour une approximation plus robuste, nous avons calculé, de la même manière que Jota et al. [69], une régression linéaire du temps moyen (sur les participants et les ID) pour réaliser la tâche de positionnement selon la latence (Figure 5.14). Le modèle correspond bien aux données ($R^2=0.98$) et nous nous en servons comme un indicateur de la performance que nous pouvons attendre pour différents niveaux de latence. Nous avons identifié la latence qui amènerait à un temps moyen de 0.795 s, ce qui correspond à la performance des participants à 25 ms avec la prédiction. La régression indique que la

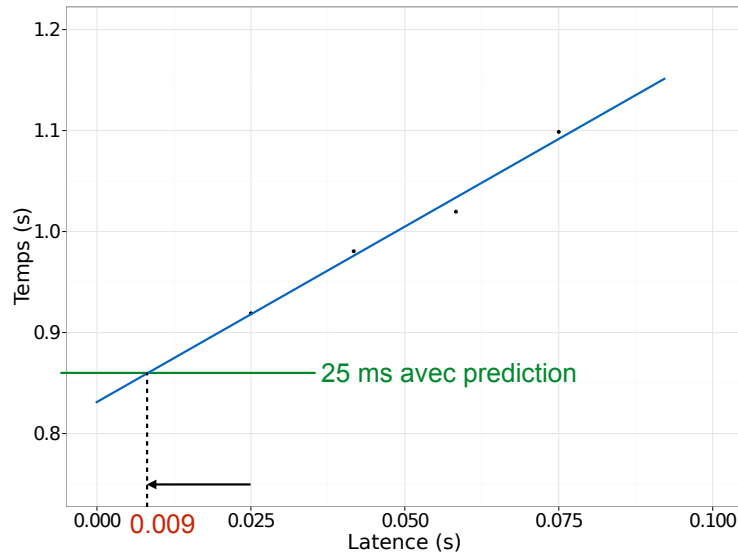


FIGURE 5.14 – Régression linéaire du temps d'acquisition de cible (groupé sur tous les ID) en fonction de la latence pour les données sans prédiction. Si on place le résultat obtenu à 25 ms avec prédiction, la régression indique une latence équivalente de 9 ms.

prédiction a permis aux utilisateurs d'avoir une aussi bonne performance que si la latence avait été de 9 ms, soit une amélioration de 64% par rapport à 25 ms.

Pour résumer, en supposant que la latence du système est assez basse, l'étude indique qu'une prédiction continue rapproche la latence apparente d'un système de la condition idéale sans latence, ce qui permet aux utilisateurs d'améliorer leur performance. De plus, notre extrapolation suppose que l'évolution du gain de performance reste linéaire en fonction de la réduction de la latence. Or, il est possible qu'il existe un effet de seuil. Jota et al. n'ont pas détecté de preuve d'un tel seuil en descendant à 1 ms de latence [69] mais nous savons que celui-ci existe pour l'interaction à la souris [93]. De plus, Ng et al. ont aussi montré qu'en moyenne, la latence au toucher pour le déplacement d'un objet devient imperceptible sous 6 ms [106]. Il est donc possible que la latence ait moins d'influence sur les performances à de tels niveaux. Si un tel effet de seuil existe, alors la régression s'incurverait pour les très faibles latences et notre prédiction pourrait nous avoir amené plus près que ce que nous avons évalué d'un système parfait. Ce n'est bien sûr qu'une hypothèse qui nécessitera d'être validée expérimentalement, mais elle indique que notre estimation d'une équivalence à 9 ms de latence est conservative.

Préférences des utilisateurs

Lorsque les participants sont libres d'interagir avec le disque, ils font plutôt des mouvements lents et lisses. Cela favorise la prédiction puisqu'il n'y a alors pas de variation brutale de la vitesse et l'erreur générée par la prédiction reste faible. Dans ce contexte, la préférence des participants va clairement à la présence de prédiction à 25 ms comme visible

sur la Figure 5.13 (à droite).

Lorsque les participants doivent réaliser la tâche de positionnement, les résultats sur les préférences sont cohérents avec ceux observés sur la performance. Les participants rejettent la prédiction pour les valeurs de latence élevées. Pour justifier leur choix, les participants disent qu'ils sentent des "instabilités" et une "absence de contrôle" sur l'objet manipulé. Dans la condition à 25 ms, certains participants ont eu des difficultés à percevoir la différence entre les conditions avec ou sans prédiction. Cela est cohérent avec le non-rejet de l'hypothèse nulle dans cette condition : face à deux conditions non différenciables, on s'attend à de l'aléatoire. Par contre, les résultats sur la performance montrent que même si les utilisateurs ne perçoivent pas de différence entre les deux conditions, réduire la latence avec la prédiction améliore tout de même leur performance sur la tâche.

Généralisation de la méthode

Les dispositifs tactiles commerciaux actuels ont pour la plupart des latences bien supérieures à 25 ms, valeur pour laquelle notre prédiction apporte des résultats probants. Cependant, 25 ms est une valeur accessible, qui ne nécessite pas d'améliorations matérielles significatives. Certaines tablettes comme les iPads depuis l'iPad Air 2 ont des capteurs pour le tactile fonctionnant à 120 Hz. Les écrans à 120 Hz ont été démocratisés par le développement de la 3D et commencent maintenant à apparaître sur les tablettes [2]. Un fonctionnement à 120 Hz est source de 4.17 ms latence, ce qui laisse plus de 20 ms pour les autres sources (application, traitement avant affichage, synchronisation...).

Notre étude devrait donc motiver les créateurs d'applications à garder une latence inférieure à 20 ms. Lorsque c'est le cas, un modèle prédictif simple, tel que le modèle linéaire que nous avons utilisé, peut alors contrer en partie la latence restante pour se rapprocher d'un dispositif idéal. Au contraire, atteindre 9 ms de latence ou moins avec une pure approche matérielle semble plus difficile et plutôt coûteux en terme de dépense énergétique [88, 106].

Cependant, il semble que l'utilité d'une approche prédictive ne soit pas limitée aux systèmes n'ayant pas plus de 25 ms de latence. Deber et al. ont montré que des petites réductions de latence, de l'ordre de 16 ms, sont facilement perçues par les utilisateurs pour une tâche de déplacement d'un objet [46]. Combiné avec nos résultats, cela ouvre la voie vers des études sur la compensation *partielle* de la latence avec une prédiction sur des dispositifs à latence plus élevée.

Travaux ultérieurs

Après la publication de nos travaux en 2015 [34], d'autres chercheurs ont continué d'explorer les effets d'une prédiction dans le but de compenser la latence au toucher. Ces travaux sont parus en 2016 :

Nancel et al. explorent les effets secondaires générés par différents modèles de prédiction [103]. Les auteurs font tester aux participants différents modèles de prédiction (premier ordre, second ordre, filtre de Kalman...) sur différentes tâches (dessin, déplacement d'objet, déplacement d'une grande zone de l'affichage). Avec les commentaires des participants, les auteurs regroupent les effets secondaires dans 7 catégories (retard, anticipation, mauvaise distance, mauvaise orientation, tremblements, sauts et effet de ressort). Ensuite, grâce aux trajectoires produites par les participants sur les différentes tâches, les auteurs proposent des métriques pour quantifier chacun de ces effets. Ces métriques permettront de comparer de nouveaux modèles de prédiction.

Henze et al. [60] ont aussi mis en évidence qu'un modèle de prédiction à 33.3 ms, s'appuyant sur un réseau de neurones, réduit l'erreur de manière plus importante qu'une prédiction linéaire du premier ordre et qu'elle améliore la performance des utilisateurs sur une tâche de pointage. Comme dans nos résultats, les auteurs constatent qu'une longueur de prédiction trop élevée (66.6 ms) perd les bénéfices qu'elle apporte à 33.3 ms en faveur d'un bruit qui fait alors décroître la performance.

5.1.5 Bilan

Nous avons montré qu'utiliser une prédiction de la trajectoire future du doigt afin de compenser la latence du toucher lors d'un mouvement continu est un problème complexe. Les modèles de prédiction sont sensibles au bruit des données. Dans notre cas, parmi les modèles testés, c'est un modèle linéaire simple qui nous permet d'obtenir les meilleurs résultats. Ce modèle est très réactif car il est basé seulement sur les deux plus récentes positions connues. Par ailleurs, il est peu sensible au bruit car il ne nécessite qu'une seule intégration du temps. Lors d'une expérimentation utilisateur, nous avons constaté que le modèle de prédiction améliore l'utilisabilité du système uniquement si la longueur de prédiction est inférieure à 42 ms. Lorsque la prédiction est utilisée sur notre système à 25 ms de latence pour effectuer des tâches de pointage elle permet de compenser l'équivalent de 64% de ces 25 ms.

5.2 Compensation motrice humaine

5.2.1 Problématique

Nous avons montré dans les chapitres précédents la sensibilité des utilisateurs à la latence au toucher et l'impact négatif qu'elle a sur leur performance. Pourtant, l'interaction tactile est de plus en plus présente dans notre quotidien que ce soit sur les smartphones et les tablettes ou sur des surfaces plus grandes comme les écrans tactiles publics. Au-delà de l'aspect naturel de l'interaction tactile [150], on peut se demander si les utilisateurs ne s'habituent pas à un monde numérique qui répond à leurs actions avec du retard. En revanche, il n'est pas évident de déterminer si cette habitude implique aussi de la compensation, c'est-à-dire, est ce qu'avec une utilisation prolongée d'écrans tactiles longs à réagir, les utilisateurs deviennent finalement capables de réaliser des tâches comme s'il n'y avait pas de latence ?

Diverses études du domaine du contrôle moteur montrent les capacités du cerveau humain à s'adapter à des perturbations visuomotrices spatiales, mais aussi temporelles (cf. chapitre 3). Cependant, ces études se focalisent sur de l'interaction indirecte, où la main des participants est cachée, afin de créer un décalage entre la proprioception de l'utilisateur et sa vision d'un curseur représentant la position de sa main. Ces conditions sont différentes de l'interaction directe où sont visibles dans un même espace, à la fois la main et l'objet déplacé qui est en retard à cause de la latence.

Les études proposées dans le domaine de l'IHM ne peuvent non plus répondre à cette question sur l'habitude à la latence pour l'interaction au toucher, car elles ont toutes été menées sur des périodes très courtes (souvent moins d'une heure). Elles ne permettent donc pas d'étudier la progression des participants avec la pratique. Dans la vie quotidienne, les gens utilisent leurs appareils tous les jours, durant des périodes plus longues et en répétant certaines tâches plusieurs fois par jour, ce qui offre donc un entraînement bien plus important qu'une session expérimentale, et offre ainsi une plus grande opportunité de développer des stratégies compensatrices. Les effets de la latence sur une personne entraînée pourraient alors être différents de ceux révélés lors de courtes sessions expérimentales.

Nous avons mené deux études expérimentales afin de répondre aux questions suivantes :

- Est ce que les utilisateurs compensent les effets de la latence grâce à l'entraînement ? Si oui, la compensation est-elle totale ou seulement partielle ? Combien de temps cela prend-il ? Est ce que cela persiste avec le temps ?
- S'agit-il d'une compensation ou d'une adaptation ?
- Cette compensation est-elle spécifique à une tâche donnée ou se transfère-t-elle à des tâches similaires ?
- La présence de latence altère-t-elle l'apprentissage de compétences motrices ?

Nos études se concentrent sur une tâche de suivi de cible. La première étude suit

l'évolution des performances de deux groupes sur dix sessions réparties sur au moins deux semaines. Un groupe est soumis à une latence, l'autre non. Cette première étude inclut aussi des tests sur le transfert d'apprentissage d'une tâche à une autre, et sur la rétention de l'apprentissage. La deuxième étude étudie l'influence du retard sur l'apprentissage des connaissances motrices nécessaires au suivi de cible. Nous discutons ensuite des résultats de ces deux études d'un point de vue sensorimoteur. En particulier, nous considérons ce que ces études nous apprennent sur les stratégies mises en œuvre lors d'une perturbation visuomotrice temporelle dans le cas de l'interaction au toucher directe. Puis nous abordons le point de vue de l'IHM et de la conception des interactions.

5.2.2 Etude utilisateur 1 : Adaptation à la latence, transfert, rétention

Les travaux présentés dans cette section ont fait l'objet d'une publication à la conférence Human Factors in Computing Systems (CHI) en 2017 : "Does Practice Make Perfect? Learning to Deal with Latency in Direct-Touch Interaction" [37]. Nous proposons ici une analyse statistique différente et plus approfondie des résultats que celle qui a été exploitée dans cet article. Cette nouvelle analyse a été soumise pour publication dans un journal de neurosciences.

Cette première étude a été conçue de manière à tester ces quatre hypothèses :

- Les utilisateurs peuvent s'adapter à la latence sur une tâche de suivi de cible, avec de l'entraînement. (H1 - adaptation)
- En prolongeant l'apprentissage, la performance avec de la latence devient aussi bonne que s'il n'y avait pas de latence. (H2 - adaptation totale)
- Cet apprentissage se transfère sur une tâche similaire. (H3 - transfert)
- Cet apprentissage persiste à long terme. (H4 - rétention)

Nous avons invité 20 participants (droitiers, âge=28 [23-37], 5 femmes) à prendre part à l'étude, principalement des étudiants des différentes universités et laboratoires de notre campus. La plupart possèdent un smartphone et/ou une tablette et l'utilisent tous les jours.

Protocole

Les participants ont réalisé une tâche de suivi de cible pendant dix sessions d'entraînement réparties sur au moins deux semaines. Notre approche est de comparer les effets de la latence lors de la première exposition et après l'entraînement. Nous attendons une amélioration de la performance avec la pratique. Sur cette amélioration, nous devons distinguer la part due à l'acquisition de compétences spécifiques à la tâche (apprendre la trajectoire et la vitesse de la cible, ou comment faire un suivi de manière plus lisse), de la part due à la compensation du retard. Nous avons donc comparé les performances d'un *groupe contrôle* et d'un *groupe test* entraînés sur la même tâche respectivement dans une condition de faible latence (aussi basse que notre système le permet) et une condition avec latence plus élevée.

Durant l'entraînement, l'amélioration de la performance du groupe contrôle est principalement due aux compétences spécifiques au suivi de cible. Pour le groupe test, l'amélioration est un cumul de l'amélioration sur le suivi de cible et de la compensation de la latence. Par différence entre les deux groupes, on évalue donc l'amélioration due à la compensation de la latence. Le groupe test, qui doit faire face à la latence, devrait avoir une performance plus faible que le groupe contrôle à la première session. Une comparaison des groupes à la dernière session évalue aussi si la latence est compensée totalement (la performance du groupe test rattrape celle du groupe contrôle), partiellement, ou pas du tout.

La division en deux groupes impose donc un design inter-sujets. En effet, chaque participant ne peut passer qu'une seule condition car son entraînement dans une condition empêche sa participation à l'autre. Ce design introduit une difficulté : la différence de performance entre les deux conditions pourrait être expliquée en partie par une différence générale de performance des participants de chaque groupe. Nous contrôlons donc soigneusement l'assignation des participants au groupe contrôle ou test de manière à équilibrer les groupes. La performance individuelle de chaque participant est testée sur dix répétitions de la tâche de suivi sur la trajectoire d'entraînement avec latence élevée, avant même la première session d'entraînement. Cette *tâche d'équilibrage* permet de constituer des groupes équivalents en termes de capacité à suivre une cible et à gérer la latence.

Le groupe test est entraîné avec 75 ms de latence, ce qui correspond à la latence de nombreux systèmes disponibles dans le commerce. Dans le cas idéal, le groupe contrôle devrait être entraîné avec une latence nulle ou négligeable. Nous approximons cette condition en associant notre système faible latence (décrit au chapitre 4) avec la technique de prédiction à 25 ms décrite dans la section précédente. Nous faisons alors l'hypothèse que la latence restante ou les légères instabilités introduites par la prédiction ont des effets négligeables comparés aux effets des 75 ms de latence du groupe test. Par simplification, dans la suite, on nommera les conditions "avec latence" pour le groupe test et "sans latence" pour le groupe contrôle.

La tâche principale sur laquelle les participants se sont entraînés durant les 10 sessions a été choisie de manière à être suffisamment difficile pour pouvoir observer un apprentissage mesurable grâce à une amélioration de performance. Nous avons aussi choisi une tâche nouvelle pour éviter de trop grosses disparités dans les performances des participants avant l'expérience. En particulier, nous avons évité la tâche classique de positionnement d'objet : la plupart des participants utilisent des écrans tactiles au quotidien et ont donc des niveaux d'entraînement variés sur cette tâche avec des niveaux de latence différents. La tâche de suivi de cible a déjà été utilisée dans la littérature pour étudier l'apprentissage sensorimoteur [52, 96] et satisfait nos deux prérequis. Lors d'une étude pilote, nous avons constaté qu'une trajectoire complexe n'était pas nécessaire pour observer de l'apprentissage. Cependant, utiliser un simple cercle avec vitesse constante ne conviendrait pas à nos objectifs. En effet, une vitesse constante impliquerait un écart spatial constant entre le doigt et l'objet déplacé. C'est un cas particulier qui ne rend pas bien compte de la nature

temporelle du phénomène de latence ce qui pourrait, selon Rohde et Ernst [121], empêcher une adaptation potentielle. C'est pourquoi nous avons choisi une trajectoire ellipsoïdale sur laquelle la vitesse de la cible est liée à la courbure par la loi de la puissance 2/3. Selon Lacquaniti et al. [83], ce profil de vitesse correspond aux mouvements naturels de la main.

L'affichage, tel qu'il est présenté aux participants, est illustré en Figure 5.15. Des captures d'écran y sont représentées lors de différents moments de l'étude.

Les participants contrôlent un disque rouge (1.78 cm de rayon) avec leur index. L'écran affiche le chemin suivi par un disque blanc de même taille qui sert de cible. Afficher la trajectoire de la cible assure que son mouvement est prédictible, ce qui, selon Rohde et al., est une condition pour s'adapter au délai [122]. Le mouvement de la cible (Figure 5.16) suit les équations suivantes :

$$\begin{cases} x(t) = 840 + 720 * \cos(t) \\ y(t) = 640 + 270 * \sin(t) \end{cases} \begin{cases} x'(t) = -720 * \sin(t) \\ y'(t) = 270 * \cos(t) \end{cases}$$

x et y sont les coordonnées exprimées en pixels pour une taille d'écran de 1920×1080 pixels, de diagonale de 24 pouces (taille d'un pixel=0.277 mm), sur un repère direct (origine en bas à gauche). Le temps t est exprimé en secondes. La cible suit donc une trajectoire ellipsoïdale, faisant un tour de l'ellipse en 2π secondes. L'ellipse est légèrement décentrée en haut à gauche pour éviter que la main droite des participants vienne taper contre les bords en bas et à droite de l'écran lorsqu'ils réalisent la tâche.

Les participants doivent tout d'abord amener le disque rouge dans une zone de départ (disque blanc, rayon=2.62 cm) centrée à l'extrême droite de l'ellipse. Une fois l'objet dans la zone pendant une demi-seconde, celle-ci disparaît, la cible apparaît et commence à tourner sur l'ellipse dans le sens inverse des aiguilles d'une montre. Le but du participant est de suivre la cible avec le disque rouge aussi précisément que possible, c'est à dire d'essayer de recouvrir au maximum la cible avec le disque rouge. Un recouvrement complet est difficile, et une portion de la cible en forme de croissant de lune reste souvent visible. Le participant peut facilement interpréter ce retour visuel : il est en retard si le croissant blanc est devant le disque rouge, en avance s'il est derrière.

Pour chaque essai, la cible effectue trois tours de l'ellipse ce qui dure environ 20 s ($3 \times 2\pi$). Quand le point à l'extrême droite de l'ellipse est franchi à la fin du troisième tour, l'essai se termine. Le système déplace alors automatiquement le disque rouge à une position fixe en bas à gauche de l'écran, prêt à démarrer l'essai suivant.

Les participants peuvent suivre la progression de leurs performances grâce à des barres verticales qui affichent l'erreur pour chaque essai. La première barre représente la moyenne du participant lors de sa session précédente. Pour le cas particulier de la première session, la première barre indique juste la performance sur le premier essai. Cette première barre a toujours la même taille et est appuyée par une barre horizontale rouge. Les barres suivantes indiquent l'erreur pour les essais de la session courante, leur taille étant calculée

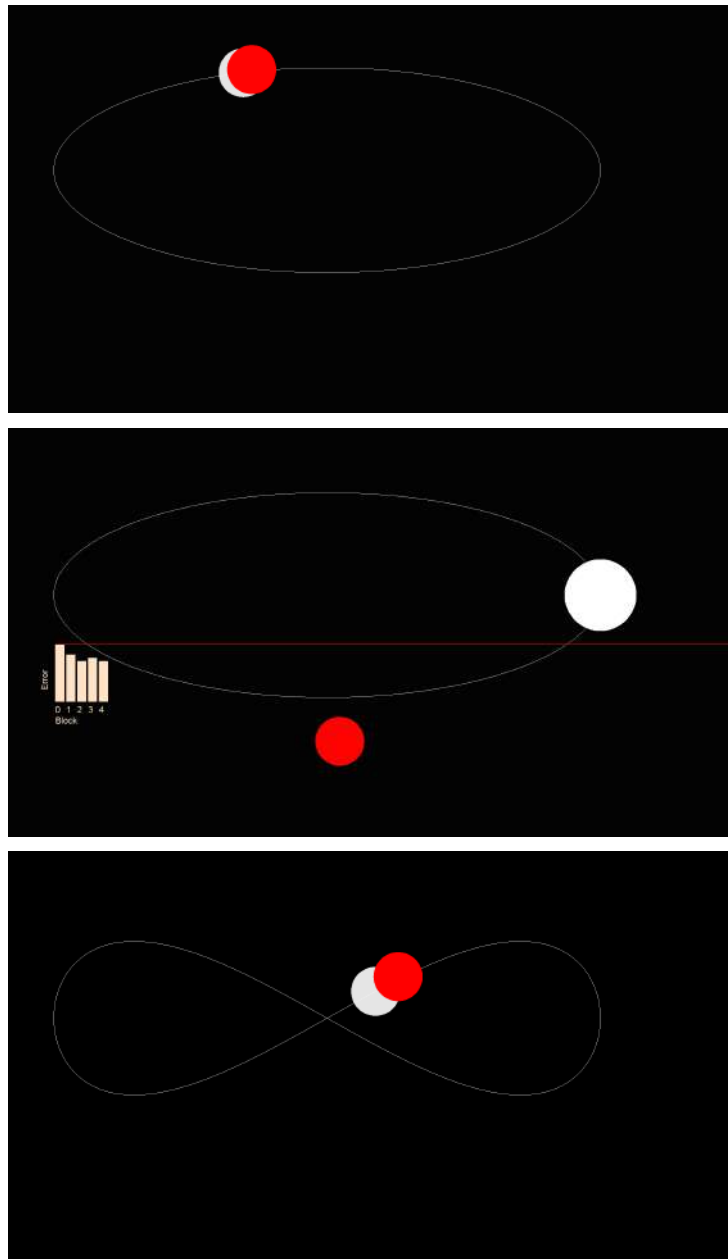


FIGURE 5.15 – L’affichage à trois moments de l’étude.

Haut : Affichage durant un essai. Le disque rouge est contrôlé par le doigt du participant. Le disque blanc (partiellement recouvert) est la cible, qui suit la trajectoire de l’ellipse dans le sens inverse des aiguilles d’une montre. Ici, la cible n’est pas totalement recouverte avec un croissant de lune sur le devant qui indique que le participant est en retard.

Milieu : Affichage entre deux essais. Le grand disque blanc est la zone de départ pour l’essai suivant. Des barres verticales présentent les résultats des essais précédents de la session. La ligne rouge horizontale correspondant à l’essai 0 est la moyenne de l’erreur lors de la session précédente et fixe un objectif d’erreur à essayer de ne pas dépasser pour les participants.

Bas : Affichage durant un essai sur la tâche de transfert.

(Images issues de [37])

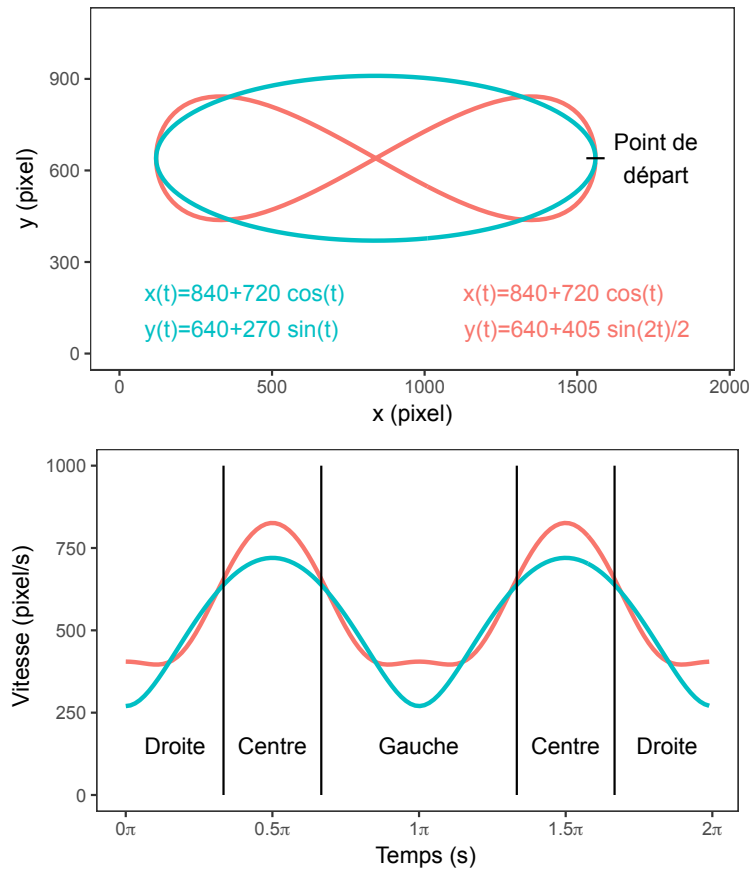


FIGURE 5.16 – En haut : Equations et représentations de la trajectoire d’entraînement (ellipse, bleu) et de la trajectoire de transfert (symbole infini, rouge). En bas : Profils de vitesse des deux trajectoires et découpage en trois parties du symbole infini (les détails sont donnés dans le texte).

relativement à la première. Les participants, tacitement, tentent de rester sous la barre rouge, c’est-à-dire de faire mieux que leur moyenne précédente. Les barres sont visibles uniquement entre les essais : elles disparaissent dès que le disque rouge est amené dans la zone de départ de l’essai suivant.

Il a été prouvé que le sommeil est un facteur important dans l’amélioration des performances motrices [148]. Nous avons donc choisi de répartir nos sessions d’entraînement sur différents jours de manière à mesurer l’adaptation sur le long terme. L’entraînement se déroule sur 10 sessions de 60 essais chacune. Le nombre d’essais par session a été calculé de manière à ce que la session dure environ une demi-heure. Une étude pilote nous a indiqué que des sessions plus longues peuvent introduire un fort effet de fatigue. Un laps de temps semi-régulier est imposé entre les sessions : chaque session est faite un jour différent (avec donc au moins une nuit de sommeil entre deux sessions), deux sessions ne peuvent être séparées par plus de trois jours, et la totalité des dix sessions doit être réalisée en l’espace de moins de trois semaines. De plus, chaque participant doit faire ses sessions à peu près

à la même heure de la journée. Finalement, à cause de contraintes d'emploi du temps, deux participants du groupe test ont fait deux de leurs sessions le même jour (séparées par quelques heures). Nous n'avons pas observé d'effet de la distribution des sessions : les courbes d'apprentissage sont semblables pour les participants faisant trois sessions par semaine un jour sur deux et pour les participants venant tous les jours de la semaine (week-ends exclus).

Durant les sessions, de courtes pauses sont autorisées entre chaque essai. Des pauses plus longues de 30 s sont imposées tous les dix essais pour éviter que les participants enchaînent trop rapidement et soient fatigués sur les derniers essais.

Pour comprendre si la compensation de la latence est spécifique à la tâche, nous évaluons aussi le transfert d'apprentissage avec une seconde tâche que nous appelons *tâche de transfert*. L'étude de De La Malla et al. suggère que le transfert d'apprentissage du retard dans leurs conditions expérimentales, avec la main cachée, n'intervient que quand les tâches sont très similaires [44]. Il est probable qu'aucun transfert ne serait visible pour une tâche trop différente du suivi de cible sur l'ellipse. Nous avons donc choisi également une tâche de suivi de cible mais sur une trajectoire différente. La forme est celle d'un symbole infini, avec un profil de vitesse différent. La position et la vitesse de la cible (Figure 5.16) sont régies par les équations suivantes :

$$\begin{cases} x(t) = 840 + 720 * \cos(t) \\ y(t) = 640 + 405 * \sin(2t)/2 \end{cases} \quad \begin{cases} x'(t) = -720 * \sin(t) \\ y'(t) = 405 * \cos(2t) \end{cases}$$

La trajectoire de cette tâche est intéressante pour évaluer le transfert car la partie droite est très semblable à la partie droite de l'ellipse. La partie gauche est similaire en forme, mais avec un déplacement de la cible dans la direction opposée. Et la partie centrale est un mouvement diagonal, très différent du profil de l'ellipse, avec deux directions différentes. Il est donc possible que le transfert d'apprentissage de l'ellipse vers cette nouvelle trajectoire soit différent selon la partie considérée.

Pour étudier la rétention de l'apprentissage, 12 participants (6 contrôles, 6 tests) reviennent entre 7 et 9 semaines après leur dernière session d'entraînement. Les participants doivent réaliser 30 essais sur la tâche de suivi de cible sur l'ellipse, dans les mêmes conditions de latence que durant la phase d'apprentissage. Nous n'avons pas pu tester tous les participants du fait d'indisponibilités, mais 6 participants dans chaque groupe donnent tout de même des indications sur la rétention à long terme de l'apprentissage.

Mesures

La performance sur la tâche de suivi de cible est mesurée avec l'erreur de suivi spatiale : c'est la distance entre le centre de l'objet et le centre de la cible. L'erreur n'est pas mesurée pendant le premier tour pour laisser le participant rattraper la cible lors du départ et

reprendre le rythme. On calcule la moyenne de l'erreur échantillonnée à 120 Hz sur les deux derniers tours autour de l'ellipse. Cela donne un total de 12360 mesures d'erreur (60 essais \times 10 sessions \times 20 participants + 30 essais \times 1 post-session \times 12 participants). Nous enlevons 317 outliers (pour un participant et une session, un essai où l'erreur de suivi est en dehors de 1.5 fois l'écart interquartile au-dessus du troisième quartile ou en dessous du premier quartile). Il reste 12043 mesures à analyser.

La vitesse de la cible selon les différentes parties de l'ellipse pourrait aussi influencer l'apprentissage. Dans une analyse séparée, l'ellipse est découpée en 100 sections de temps égaux et on fait la moyenne inter-participants de l'erreur de suivi sur chaque section pour chaque session et groupe.

Hypothèses et analyse statistique

D'après les travaux de la littérature, on s'attend à ce que la latence augmente l'erreur de suivi au début de l'apprentissage (erreur plus grande pour le groupe test comparé au groupe contrôle). Si une compensation de la latence est possible, la performance du groupe test devrait se rapprocher progressivement de la performance du groupe contrôle. Pour vérifier cette hypothèse, nous évaluons l'évolution globale de la performance suivant les sessions ainsi que la dynamique de la progression à l'intérieur de chaque session et les pertes ou les gains inter-sessions. Nous avons construit des modèles linéaires mixtes (*Linear Mixed Model*, LMM) à partir des données en utilisant le logiciel R [116] et la fonction *lme* [113]. L'erreur de suivi est la variable dépendante ; l'essai (facteur numérique, de 0 à 59), la session et le groupe sont les effets fixes. Le participant est un effet aléatoire. Le LMM décrit alors l'évolution de l'erreur de suivi en fonction de l'essai, en faisant l'hypothèse que cette variation est linéaire dans chaque session et pour chaque groupe. Le meilleur LMM est choisi en utilisant des tests de rapport de vraisemblance et une approche de suppression inversée [101]. C'est-à-dire qu'avec un test de modèles emboîtés, on évalue la différence entre un modèle complet et un modèle auquel on enlève un facteur ou une interaction. Si ces deux modèles ne montrent pas de différence significative pour expliquer la variable dépendante alors on supprime le facteur ou l'interaction du modèle. Des autocorrélogrammes sont ensuite tracés pour vérifier l'absence d'autocorrélation dans les résidus à l'intérieur d'une session. Le diagramme quantile-quantile de la Figure 5.17 valide la pertinence du modèle. On note que le graphe dévie légèrement de la régression. Certains sujets ayant obtenu une performance moins bonne sont plus difficilement modélisés par le LMM.

Des tests post-hoc sont ensuite effectués afin de comparer :

- Les ordonnées à l'origine et les pentes entre les groupes à chaque session. On utilise la fonction *glht* [64] avec les paramètres par défaut pour les ajustements nécessaires pour les comparaisons multiples
- Le point final du modèle dans une session (i.e. à l'essai 59) avec l'ordonnée à l'origine de la session suivante. Cela est fait pour chaque session et chaque groupe. On

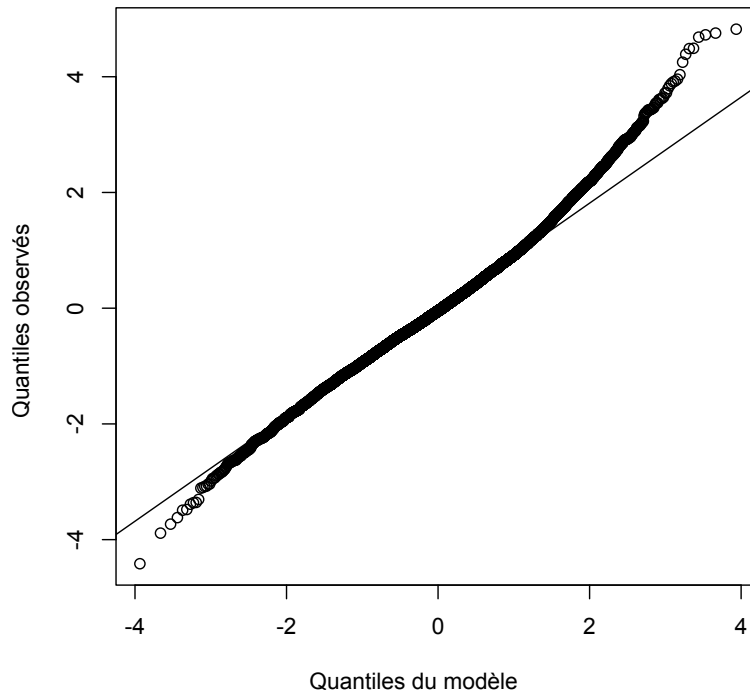


FIGURE 5.17 – Diagramme quantile-quantile évaluant la pertinence du LMM sélectionné pour modéliser l’erreur lors de l’entraînement sur la tâche de suivi de cible sur l’ellipse.

utilise une correction de Bonferroni pour compenser les comparaisons multiples. Ces comparaisons sont faites afin de détecter les discontinuités entre les sessions qui peuvent être des pertes ou des gains

Le transfert d’apprentissage est analysé séparément, en comparant la performance sur la trajectoire de transfert avant et après l’apprentissage sur l’ellipse. Afin de retirer les outliers dus au changement brutal de condition, on ne considère que les cinq derniers essais du pre-test et du post-test dont on fait les moyennes par sujet. Comme on ne veut qu’un seul indice de performance pour le pre-test et le post-test, on fait la moyenne de ces cinq derniers essais. Les données sont aussi analysées séparément pour les trois parties du symbole infini (droite : $time \in [0; \frac{\pi}{3}] \cup [\frac{5\pi}{3}; 2\pi]$, centre : $time \in [\frac{\pi}{3}; \frac{2\pi}{3}] \cup [\frac{4\pi}{3}; \frac{5\pi}{3}]$, gauche : $time \in [\frac{2\pi}{3}; \frac{4\pi}{3}]$, Figure 5.16). L’effet de *session*, *partie* et *groupe* sur l’erreur de suivi est évalué avec une ANOVA mixte, avec *session* et *partie* en facteurs intra-sujet et *groupe* en facteur inter-sujets. Pour le groupe contrôle, la comparaison évalue seulement le transfert des compétences de suivi de cible, alors que pour le groupe test la comparaison évalue à la fois le transfert de compétences de suivi de cible et de l’adaptation potentielle à la latence. La différence de transfert entre les deux groupes révèle alors si la compensation à la latence est transférée. Si c’est le cas, c’est qu’il y a recalibration visuomotrice chez les participants et donc adaptation.

Tous les effets sont considérés significatifs pour $p < .05$.

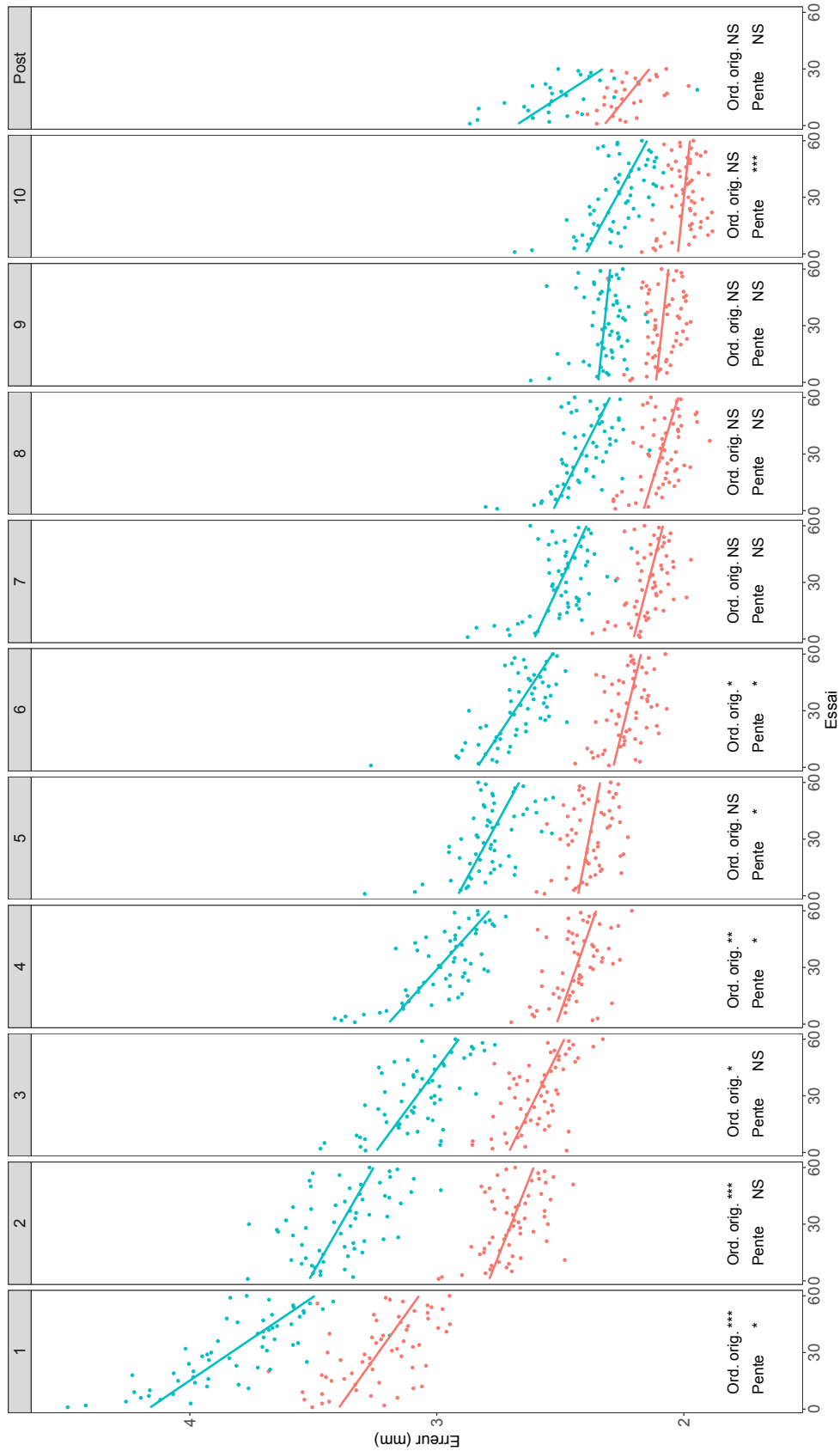


FIGURE 5.18 – Moyenne inter-participants de l'erreur de suivi pour chaque essai et session pour le groupe contrôle (rouge) et le groupe test (bleu). Les régressions du modèle linéaire sont tracées pour chaque groupe et chaque session. Pour plus de clarté, l'axe vertical n'a pas son origine à 0. Les comparaisons des ordonnées à l'origine et des pentes du modèle entre les deux groupes sont indiquées en bas de la figure. Les étoiles indiquent le niveau de significativité de la p-valeur : 0 < *** < 0.001 < ** < 0.01 < * < 0.05 < NS < 1

Résultats

Comme on le voit sur la Figure 5.18, l'erreur de suivi est plus grande pour le groupe test que pour le groupe contrôle. Avec l'entraînement, l'erreur décroît pour les deux groupes ainsi que l'écart entre les groupes. En accord avec cette description, le LMM sélectionné inclut des effets de *groupe*, *essai* et *session* sur l'erreur de suivi ainsi qu'une triple interaction entre les facteurs (test de modèles emboîtés sans l'interaction, $p < .025$) qui montre : un impact négatif de la latence sur la performance des participants ; une amélioration de l'erreur selon les jours mais aussi selon les essais ; une évolution différente de l'erreur en fonction des essais pour les différentes sessions et selon le groupe.

Les comparaisons multiples indiquent que l'ordonnée à l'origine du groupe test est significativement plus grande que celle du groupe contrôle aux sessions 1 à 4 et 6 (Figure 5.18). La différence diminue de 0.76 mm lors de la session 1 (test-contrôle, $p < .001$) à 0.35 mm lors de la session 10 ($p > .4$). Avec l'entraînement, la performance du groupe test se rapproche donc de celle du groupe contrôle. La comparaison des pentes entre les deux groupes donne des résultats plus variables. La pente du groupe test est toujours plus grande que celle du groupe contrôle mais la différence n'est significative qu'aux sessions 1,4,5,6 et 10 ($p < .04$).

La Figure 5.18 indique aussi que le groupe test tend à démarrer chaque session avec une erreur plus grande comparée à la fin de la session précédente. Cependant, les tests de discontinuité entre les sessions ne sont pas significatifs.

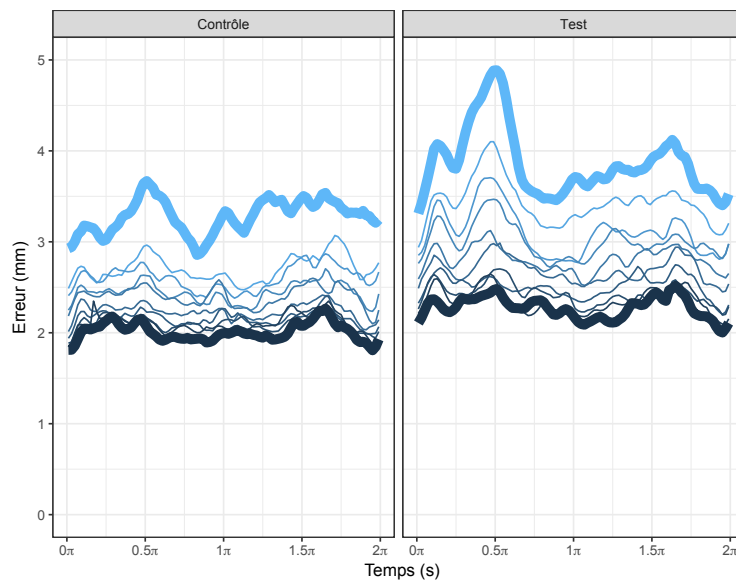


FIGURE 5.19 – Moyenne inter-participants de l'erreur de suivi sur l'ellipse de la première (bleu clair) à la dixième (bleu foncé) session pour les deux groupes. La première et la dernière session sont dessinées avec un trait plus épais pour faciliter la comparaison.

L'analyse de l'erreur selon les différentes parties de l'ellipse fait ressortir des profils

hétérogènes (Figure 5.19). On observe deux pics d'erreur à $\pi/2$ et $3\pi/2$, qui correspondent aux pics de vitesse de la cible. Le premier pic est plus grand pour le groupe test mais diminue progressivement au cours des sessions. Pour le groupe contrôle, il semble que la plus grande diminution de l'erreur advienne pendant les deux premières sessions alors que la diminution pour le groupe test s'étale de manière plus régulière sur les 10 sessions.

Les participants testés 7 à 9 semaines après leur dernière session d'entraînement ont une bonne rétention de l'apprentissage et des *savings* illustrés par un réapprentissage rapide (Figure 5.18, session "Post"). Nous avons considéré l'erreur de suivi moyenne sur les 10 derniers essais de cette session, après une courte refamiliarisation sur 20 essais. Une comparaison (ANOVA avec *session* et *groupe* en facteurs) montre que la performance est moins bonne que celle de la session 10 ($F(1, 10) = 6.39, p = .03$) mais avec une petite taille d'effet (+0.2 mm) comparée au gain de performance global entre la session 1 et 10 (-1.39 mm). Il n'y a pas d'influence du *groupe* ($F(1, 10) = 0.46, p > .5$) ni d'interaction *groupe* \times *session* ($F(1, 10) = 0.16, p > .69$).

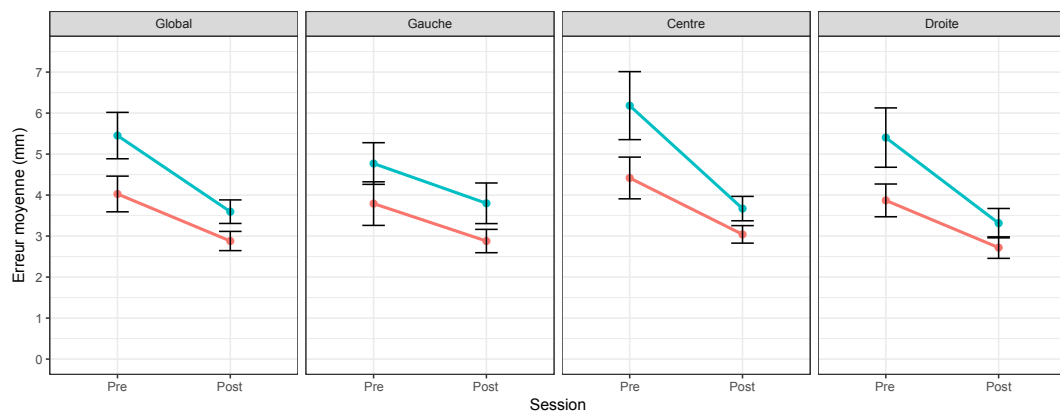


FIGURE 5.20 – Performances de suivi de cible sur la trajectoire de transfert, avant (pre-test) et après (post-test) l'entraînement sur l'ellipse. Le groupe contrôle est en rouge, le groupe test est en bleu. Le graphique de gauche illustre la progression globale; les trois autres découpent ce résultat pour les trois parties de la trajectoire en forme de symbole infini. (Figure traduite de [37])

Les effets du transfert sont détaillés sur la Figure 5.20. Les performances sont globalement meilleures pour le groupe contrôle que le groupe test (effet de *groupe* : $F(1, 18) = 25, p < .001$) et pour le post-test que le pre-test (effet de *session* $F(1, 18) = 123, p < .001$). Le groupe test montre une meilleure amélioration de sa performance que le groupe contrôle (*groupe* \times *session* : $F(1, 18) = 6.9, p < .02$) qui varie selon la partie de la trajectoire (*groupe* \times *session* \times *partie* : $F(2, 36) = 5.1, p < .02$). L'effet du transfert est similaire sur les trois parties pour le groupe contrôle mais pas pour le groupe test. Deux ANOVAs post-hoc (avec correction de Bonferroni) montrent en effet une forte interaction entre *session* et *partie* pour le groupe test ($F(2, 18) = 12, p < .001$) mais pas pour le groupe contrôle ($F(2, 18) = 4.5, p = .052$).

Discussion de la première étude

Si on reprend les 4 hypothèses évaluées dans cette première étude (H1 : adaptation, H2 : adaptation totale, H3 : transfert, H4 : rétention), les résultats indiquent que :

- Les participants s'adaptent à la latence. Pendant l'entraînement, les deux groupes améliorent leur performance, mais la dynamique de l'amélioration dépend du groupe. Nous supposons que le groupe contrôle apprend principalement la tâche de suivi alors que le groupe test apprend une combinaison de la tâche de suivi et d'une compensation de la latence. Le transfert de cette compensation sur une autre tâche suggère qu'il y a bien eu une recalibration visuomotrice et donc adaptation, ce qui va dans le sens de H1 et H3.
- Lorsque les participants effectuent un nouveau test de performance 7 à 9 semaines après leur dernière session d'entraînement, il y a un bon effet de rétention pour les deux groupes, ce qui va dans le sens de H4.

L'absence d'une différence significative entre les deux groupes à la session 10 va dans le sens de H2. Cependant la performance du groupe test reste inférieure à celle du groupe contrôle sur l'ensemble des sessions, une cohérence qui n'est pas prise en compte dans les comparaisons intra-session. L'absence de significativité statistique pourrait être due à un nombre trop faible de participants. Une étude avec davantage de participants et de sessions serait nécessaire mieux évaluer H2.

L'analyse détaillée de l'erreur en fonction de la position sur l'ellipse et le découpage en trois parties de la trajectoire de transfert suggèrent que la latence a un effet plus important sur la partie haute de l'ellipse et sur la partie droite et centrale du symbole infini. Ces parties correspondent à des positions où l'objet manipulé est souvent caché par la main du participant. Cet effet disparaît avec l'entraînement, ce qui pourrait s'expliquer par le fait que les participants intégreraient la relation temporelle entre le doigt et l'objet et s'appuieraient de moins en moins sur le retour visuel de l'objet.

Les analyses intra-sessions suggèrent aussi que le groupe test a une progression plus importante que le groupe contrôle (la pente du modèle linéaire dans chaque session est systématiquement plus grande pour le groupe test que pour le groupe contrôle, bien que cette différence ne soit pas toujours statistiquement significative). Bien qu'aucune discontinuité entre sessions n'ait pu être statistiquement observée, la Figure 5.18 indique que le groupe test a tendance à commencer chaque session avec une erreur plus grande que celle de la fin de la session précédente ce qui n'est pas le cas du groupe contrôle. On sait que la mémoire motrice régresse avec le temps mais peut aussi être réactivée [39] ; on peut donc supposer que la latence pourrait augmenter cette régression, et que la compensation de la latence nécessiterait quelques essais avant d'être réactivée.

L'ensemble des résultats de cette première étude pourrait indiquer que l'apprentissage dans le groupe test comporterait deux processus parallèles : un processus lent d'apprentissage de la tâche de suivi de cible (que l'on retrouve chez le groupe contrôle), avec une

bonne rétention ; et un processus rapide pour s'adapter à la latence, avec une mauvaise rétention entre chaque session, mais un réapprentissage rapide.

La seconde étude a été construite pour tester cette hypothèse et mieux comprendre l'interaction entre ces deux processus. Notre démarche est d'utiliser un test de transfert pour évaluer si la latence interfère avec l'apprentissage des compétences de suivi. Si apprendre la tâche de suivi dans une condition avec latence se transfère à la condition sans latence, alors c'est probablement que les deux processus fonctionnent indépendamment l'un de l'autre.

5.2.3 Etude utilisateur 2 : Interférence entre l'apprentissage des compétences de suivi et l'adaptation à la latence

Les travaux présentés dans cette section ont été soumis dans un journal en neurosciences en addition à l'analyse détaillée de la première étude.

Cette seconde étude a été conçue de manière à tester cette hypothèse :

- L'apprentissage des compétences pour réaliser une tâche de suivi se fait indépendamment de l'adaptation à la latence (H5).

Nous observons si les utilisateurs sont capables de transférer des compétences apprises dans une condition où la latence est présente vers une condition sans latence. Nous avons invité 20 participants (droitiers, âge=26 [21-40], 5 femmes) à prendre part à l'étude, principalement des étudiants de différentes universités et laboratoires de notre campus. La plupart possèdent un smartphone et/ou une tablette et l'utilisent tous les jours. Aucun de ces participants n'a pris part à la première étude.

Protocole

Le protocole de cette seconde étude est similaire à celui de la première, mais limité à une unique session. Les participants doivent effectuer la tâche de suivi sur l'ellipse et sont divisés en deux groupes. Le groupe contrôle fait 80 essais dans la condition sans latence. Le groupe test fait 10 essais sans latence (pre-test), puis 60 essais avec latence (entraînement) et finalement 10 essais à nouveau sans latence (post-test pour le transfert). La performance d'un participant sur les 10 premiers essais effectués dans la condition sans latence sert aussi à l'affecter dans un des deux groupes de manière à ce que ceux-ci soient équilibrés.

Mesures et analyse

La comparaison entre le premier et le dernier bloc du groupe test évalue le transfert d'apprentissage de la condition avec latence vers la condition sans latence. La comparaison de la progression du groupe test entre ces deux blocs avec la progression du groupe contrôle évalue la qualité du transfert. Un transfert complet indiquerait que les compétences de suivi

de cible sont apprises séparément de la gestion de la latence, sans interférence. Une absence de transfert suggérerait que l'apprentissage des compétences de suivi pour une latence donnée est un processus spécifique qui ne se généralise pas vers d'autres conditions de latence. Un transfert partiel serait compatible avec les deux alternatives : soit le produit d'interférences entre deux processus indépendants, soit le produit d'une similarité entre deux processus spécifiques.

On analyse la moyenne de l'erreur de suivi des 10 premiers puis des 10 derniers essais (premier et dernier bloc) de chaque groupe, pour obtenir un indicateur de performance avant et après entraînement. Les effets de bloc et groupe sur l'erreur sont évalués avec une ANOVA mixte, bloc étant un facteur intra-sujet et groupe un facteur inter-sujets.

Résultats

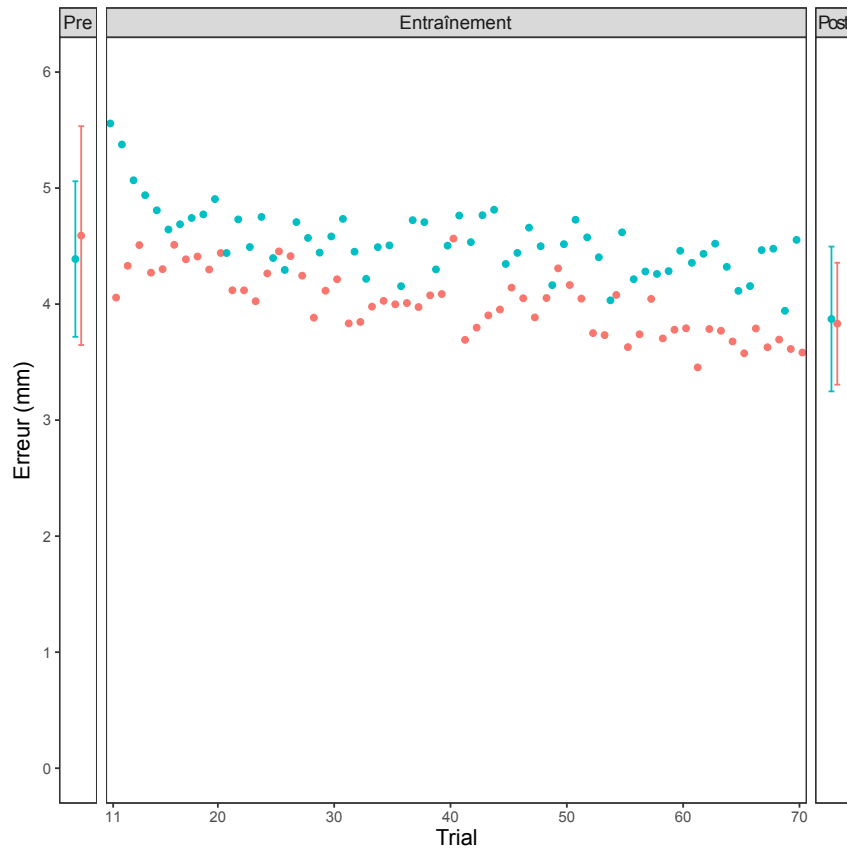


FIGURE 5.21 – Moyenne inter-participants de l'erreur de suivi sur l'ellipse lors de la seconde étude pour le groupe contrôle (rouge) et le groupe test (bleu). Chaque point dans les cadres "Pre" et "Post" représentent l'erreur moyenne, avec les intervalles de confiance à 95%, sur 10 essais, dans le premier et le dernier bloc, pour lesquels il n'y a pas de latence quel que soit le groupe. Chaque point dans le cadre "Training" représente un essai pendant l'entraînement. Durant cette phase, le groupe test est soumis à 75 ms de latence tandis que le groupe contrôle réalise la tâche sans latence.

L'erreur pour le premier et le dernier bloc ainsi que la performance sur les essais d'entraînement sont reproduits en Figure 5.21. Les deux groupes améliorent leur performance entre le pre-test et le post-test (effet de bloc : $F(1, 18) = 23.7, p < .001$) et sont globalement à des niveaux similaires (effet de groupe : $F(1, 18) = .04, p > .85$), avec une évolution comparable entre le premier et le dernier bloc (bloc \times groupe : $F(1, 18) = .86, p > .36$). Ces résultats indiquent que la performance des deux groupes sur le dernier bloc est similaire en dépit de conditions d'entraînement différentes. Nous ne pouvons conclure précisément sur une absence d'effet, mais les résultats suggèrent que s'il y a interférence de la latence sur l'apprentissage de compétence alors elle n'est pas flagrante. Une absence d'interférence indiquerait que la présence de latence n'est pas un obstacle pour développer des compétences de suivi de cible et serait en accord avec l'hypothèse H5 selon laquelle l'adaptation à la latence pour le toucher direct est apprise séparément des compétences de suivi.

5.2.4 Discussion : Contrôle moteur

Dans cette section, nous discutons et interprétons les résultats du point de vue du contrôle moteur. C'est-à-dire que nous essayons de comprendre les mécanismes d'adaptation du contrôle du mouvement et le fonctionnement des recalibrations visuomotrices qui permettent de s'adapter à un phénomène tel que la latence.

Apprentissage des compétences de suivi de cible sans latence

Dans cette section, nous ne considérons que le comportement du groupe contrôle pour comprendre l'acquisition des compétences de suivi de cible au toucher lorsque la latence est très faible.

Bien que la tâche soit répétée jour après jour, les effets de transfert vers la seconde trajectoire lors de la dernière session suggèrent que les participants du groupe contrôle n'ont pas seulement appris une compétence spécifique pour suivre la cible au rythme imposé sur l'ellipse. Ils semblent avoir plutôt développé des compétences de suivi plus générales. Cette hypothèse est aussi supportée par l'observation d'un transfert similaire quelle que soit la partie du symbole infini.

Dans une tâche de suivi de cible avec une interaction indirecte au joystick, Miall et Jackson [96] n'avaient pas observé d'amélioration claire de la performance de suivi sur cinq jours pour leur groupe contrôle. Dans notre étude, cinq jours suffisent pour réduire l'erreur de plus de 25% (d'une moyenne de 3.28 mm à la session 1 à 2.39 mm à la session 5). Au moins deux facteurs ont pu accélérer l'apprentissage dans notre étude comparée à celle de Miall et Jackson : (1) le retour sur la performance donné aux participants après chaque essai ; et (2) le fait que l'interaction au toucher partage des mécanismes communs avec des compétences de la vie de tous les jours et pourrait bénéficier de ces compétences, ce qui serait moins le cas de l'interaction indirecte utilisée par Miall et Jackson. Les résultats de

Bérard et Rochet-Capellan [22] vont dans le sens de cette seconde hypothèse. Les auteurs ont montré que le transfert d'apprentissage est important entre l'interaction au toucher et le déplacement d'un objet dans le monde physique ce qui indique un partage de mécanismes sensorimoteurs entre les deux interactions. Des études plus poussées seront nécessaires pour déterminer si l'interaction au toucher direct accélère en effet l'apprentissage de compétences comparée à l'interaction indirecte.

Apprentissage de compétences de suivi de cible et adaptation à la latence : deux processus indépendants ?

Les travaux antérieurs ont montré que le retard du retour visuel a un impact négatif sur une tâche de suivi de cible en interaction indirecte [52, 96]. Ces résultats se retrouvent dans notre première étude à la session 1 pour l'interaction au toucher direct. A notre connaissance, seuls Miall et Jackson (2006) ont étudié l'adaptation à un retard du retour visuel sur une tâche de suivi sur plus de 3 jours [96]. Les auteurs ont reporté une adaptation au retard, visible par une réduction de l'erreur dans le groupe test (avec 300 ms de latence), celle-ci se rapprochant progressivement de celle du groupe contrôle (sans latence). Notre première étude indique que cette progression lente au fil des sessions est aussi caractéristique de l'adaptation à la latence pour l'interaction au toucher direct, avec une latence plus faible (75 ms).

Alors que les compétences de suivi de cible s'améliorent progressivement avec l'entraînement, il semblerait que l'adaptation à la latence suive une évolution différente (avec une légère tendance à une perte inter-sessions et un réapprentissage rapide intra-session). Cela suggère qu'un processus d'apprentissage différent serait à l'œuvre dans cette condition, en parallèle avec l'apprentissage des compétences de suivi de cible. Cette interprétation est cohérente avec les résultats de Smith et al. (2006) qui ont étudié l'adaptation à un champ de force imposé par un manipulateur [130]. Les auteurs modélisent leurs données en cumulant deux processus d'apprentissage : un lent avec une bonne rétention et un rapide avec une faible rétention. Alors que Smith et al. ne discutent pas l'indépendance de ses deux processus, Krakauer et Mazzoni (2011) séparent clairement l'adaptation, qui impliquerait le *cerebellum*, et l'apprentissage de compétences, qui serait associé au cortex [80].

La séparation des deux processus dans le cerveau est cohérente avec les résultats de la seconde étude selon lesquels les participants faisant face à la latence semblent développer des compétences de suivi de cible équivalentes à ceux n'ayant pas de latence. Les participants du groupe test pourraient développer une stratégie de contrôle de leur mouvement qui serait indépendante de la latence. En parallèle, ils essaieraient de recalibrer cette stratégie pour s'adapter au retour visuel incohérent. Telgen et al. (2014) [138] considèrent que l'automatisation d'une nouvelle stratégie de contrôle devrait générer des gains inter-sessions alors que la recalibration d'une stratégie existante devrait générer des pertes. Nos résultats peuvent être interprétés à la lumière de ces hypothèses : nous observons dans la

première étude une tendance à la rétention voire au gain inter-session (entre la session 1 et 2) pour les compétences de suivi de cible, et une tendance à la perte pour l'adaptation à la latence.

Le transfert plus fort pour le groupe test que pour le groupe contrôle dans la première étude indique qu'en plus des compétences de suivi de cible, les participants sont aussi capables de transférer leur adaptation à la latence. Contrairement au groupe contrôle, le transfert pour le groupe test est différent selon les parties de la trajectoire considérées. Cela pourrait être dû à une influence de la vitesse ou de la position sur le transfert de l'adaptation à la latence. Une autre explication serait que dans les parties hautes et centrales de la trajectoire, la main droite des participants cache le retour visuel de l'objet manipulé. Or, lors du post-test, après l'adaptation, la stratégie de contrôle repose moins fortement sur le retour visuel de l'objet et plus sur celui du doigt (cf. "Discussion de la première étude"). Un retour visuel caché aurait donc moins d'influence sur la performance des participants. D'autres études seront nécessaires pour clarifier les origines de ce phénomène. Par exemple, un test de performance sans le retour visuel de l'objet pourrait permettre de quantifier l'importance de ce retour visuel avant et après la phase d'adaptation.

Interaction de l'apprentissage sensorimoteur entre le monde physique et le monde numérique

Le développement récent de systèmes ubiquitaires (téléphones portables, tablettes, réalité virtuelle et augmentée) a orienté les concepteurs d'interaction homme-machine vers des modalités d'interaction "naturelle". Les nouvelles interactions dérivent les gestes que nous effectuons dans le monde physique pour créer des interactions avec du contenu numérique [107], comme par exemple pour l'interaction directe au toucher [151]. Cela représente un défi important pour les sciences du mouvement : comment le cerveau des utilisateurs transfère-t-il les aptitudes sensorimotrices du monde physique au monde numérique et inversement [22] ?

Dans notre étude, la tendance à la perte de performance entre les sessions pour les participants géant la latence pourrait être en accord avec l'hypothèse bayésienne selon laquelle les gens associent un a priori sur les états possibles du monde qui les entoure avec l'évaluation de l'état courant fourni par les sens [142]. L'a priori d'un utilisateur consiste à penser qu'un objet va se déplacer sans latence car il a développé depuis l'enfance ses aptitudes sensorimotrices dans un monde physique sans latence. Il est possible que les utilisateurs transfèrent cet a priori vers le monde numérique lorsqu'ils interagissent avec un écran tactile. Alors, lorsque le retour visuel ne correspond pas à cet a priori, ils recalibreraient leurs commandes motrices pour compenser la latence. Quand les utilisateurs arrêtent d'utiliser leur dispositif, ils retourneraient au monde physique et désapprendraient la recalibration. Selon Kitago et al. (2013) [77], le désapprentissage est "un processus actif

où le comportement adapté revient progressivement aux habitudes”¹, qui, dans le monde physique ne comportent pas de latence. Cependant, les *savings* indiquent que le processus de recalibration est réactivable très rapidement.

Les résultats de Miall et Jackson [96] indiquent aussi que cette recalibration est généralisable entre deux niveaux de latence. Sur une tâche de suivi de cible, les participants qui se sont adaptés à 300 ms de latence ont une bien meilleure performance sur un post-test à 400 ms que les participants s’étant entraînés sans latence. Cette capacité à recalibrer rapidement pour différents niveaux de latence pourrait expliquer pourquoi les utilisateurs de nouvelles technologies peuvent passer si facilement d’un dispositif à l’autre. De manière générale, plus d’études sur le transfert seront nécessaires pour déterminer si ces résultats sont spécifiques au suivi de cible ou si cela s’applique à d’autres tâches.

5.2.5 Discussion : conception des systèmes tactiles

Dans cette section, nous discutons les résultats sous l’angle de l’interaction homme-machine. Quelles implications nos résultats ont-ils sur la conception des systèmes tactiles ? Quelles études devront être menées pour compléter nos résultats ?

L’adaptation à la latence lors des 10 sessions d’entraînement a réduit l’impact négatif de la latence de 54% sur la tâche de suivi de cible et on observe une rétention de cette adaptation, même après sept semaines sans entraînement. De plus, il semble que la présence de latence n’empêche pas l’apprentissage d’une compétence motrice telle que le suivi de cible. Nos études indiquent donc que l’impact de la latence sur la performance des utilisateurs pour l’interaction au toucher a peut-être moins d’effet négatif sur les comportements utilisateurs que ne le décrivent les travaux précédents.

Nous avons choisi une tâche avec une trajectoire prévisible et pour laquelle les effets de la latence ne peuvent être confondus avec les effets d’une autre perturbation (spatiale ou inertielle). Notre but était de favoriser l’adaptation [122, 121]. Des tâches plus classiques de l’IHM devront être étudiées pour évaluer la généralité de nos résultats. La tâche d’acquisition de cible, par exemple, contient des mouvements erratiques durant la phase d’ajustement pour lesquels la latence a de forts effets négatifs (cf. section 5.1.2). Il est probable que l’adaptation à la latence pour ce type de tâche soit plus difficile.

Le transfert d’apprentissage entre différents niveaux de latence est aussi un champ d’investigation futur. En effet, les utilisateurs interagissent avec différents dispositifs et différentes applications avec des niveaux de latence variables. Une étude sur ce type de transfert indiquerait si l’adaptation développée pour un niveau particulier de latence se transfère à d’autres activités ou si de nouvelles phases d’apprentissage sont nécessaires. Une absence de transfert entre différents niveaux de latence encouragerait le développement de systèmes et d’applications aux mêmes niveaux de latence.

1. “an active process whereby adapted behavior gradually reverts to baseline habits”

Enfin, bien que ce travail indique que l'impact de la latence sur les utilisateurs semble moins important que ne le laissaient présager les études court terme, cela ne doit pas empêcher le développement de dispositifs à faible latence. S'adapter à la latence prend du temps et pourrait se transférer uniquement à des tâches très proches. De plus, l'adaptation pourrait aussi avoir un coût. Les utilisateurs atteindraient une performance similaire aux résultats sans latence mais il est possible que cela demande plus d'efforts ou une charge cognitive plus importante. Ces effets devront aussi être évalués dans de futures études.

5.2.6 Bilan

Nous avons observé l'influence de la latence sur l'interaction au toucher sur une plus grande période de temps que les précédentes études. Notre première étude montre que les utilisateurs peuvent s'adapter à la latence au toucher direct et donc que l'impact de la latence se réduit avec de l'entraînement. Couplées aux résultats de la seconde étude, les données suggèrent aussi que les processus d'apprentissage du suivi de cible et d'adaptation à la latence n'interféreraient pas l'un avec l'autre.

Ces résultats appuient l'hypothèse d'une séparation des processus d'apprentissage de compétence et d'adaptation à une perturbation qui est exprimée dans certains travaux du domaine du contrôle moteur. Quant aux interfaces tactiles, ils suggèrent que la latence aurait un impact plus faible que ce qu'indiquaient les travaux antérieurs dans la mesure où les utilisateurs sont capables de développer des stratégies pour être efficaces malgré la latence. Toutefois, des études complémentaires seront nécessaires pour compléter ces résultats, notamment des études sur le transfert entre différents niveaux de latences ou sur des tâches plus communes en IHM, tel que le positionnement d'objet.

Conclusion

“Que de temps perdu à gagner du temps”

- Paul Morand, Journal inutile

Bilan de nos contributions

Nous avons traité le problème de la latence au toucher tout d’abord en diagnostiquant ses causes et ses effets puis en étudiant les différents moyens et stratégies qui peuvent être mis en œuvre pour la compenser.

Contributions sur les aspects matériel et logiciel

Le diagnostic et la compensation ont d’abord été traités d’un point de vue matériel et logiciel.

Nous avons réalisé un système qui présente uniquement 25 ms de latence sans imposer de contrainte forte sur son exploitation : il permet un affichage standard en couleur sur une grande surface, et l’utilisation de logiciels standards pour le programmer. Nous avons exploité ce système pour mener plusieurs études utilisateurs. L’utilisation de matériel du commerce standard rend ce système facilement reproductible, de même que nos études.

Au niveau logiciel, nous avons évalué le gain de performance utilisateur que peut apporter une prédiction de la trajectoire du doigt pour compenser la latence. En effet, bien que de plus en plus de systèmes tactiles du commerce utilisent des algorithmes de prédiction pour compenser la latence, aucune étude n’en avait évalué les effets sur le comportement des utilisateurs. Nous avons montré que pour une tâche de pointage au toucher, l’utilisation d’une prédiction linéaire sur une courte fenêtre temporelle améliore significativement la performance des utilisateurs. Grâce à cette prédiction, pour une tâche de déplacement d’objets, notre système à 25 ms de latence est comparable à un système à 9 ms de latence soit un gain de 64%.

Considérant que les techniques de mesure de latence au toucher développées dans les travaux précédents n’étaient pas adaptées pour une mesure rapide, juste, précise et facile à

mettre en œuvre, nous avons aussi développé notre propre méthode de mesure de la latence au toucher. De plus, la technique se généralise facilement pour mesurer la latence d'autres types de dispositifs. Elle a par exemple été utilisée depuis pour mesurer la latence d'un système de réalité augmentée [21].

Contributions sur l'aspect utilisateur

Le diagnostic et la compensation de la latence ont aussi été explorés d'un point de vue de l'utilisateur.

Notre revue de la littérature nous a permis de soulever les spécificités de la latence pour le domaine de l'interaction tactile directe et d'insister sur l'importance de ses effets négatifs sur les utilisateurs. Une de nos contributions vient appuyer ces conclusions : en étudiant l'impact de la latence sur l'interaction bimanuelle nous avons montré que les effets négatifs du retard sur la performance des utilisateurs sont tout aussi importants que dans une interaction à une main.

Nous avons enfin effectué un rapprochement entre les études sur la latence dans le cadre de l'IHM et les études sur l'adaptation aux perturbations visuomotrices du domaine du contrôle moteur. Ce rapprochement a permis de réutiliser les méthodes expérimentales et le cadre théorique des études sur l'apprentissage sensorimoteur pour les appliquer au cadre de la latence au toucher direct. Nous avons alors montré la capacité des participants à s'adapter à la latence sur le long terme pour une tâche de suivi de cible. Cette première étude utilisant les paradigmes expérimentaux du contrôle moteur nous permet donc d'apporter une nuance aux études précédemment conduites en IHM. Nos résultats indiquent aussi que des études long terme sur la latence sont nécessaires pour en appréhender correctement les effets.

Liste des publications

Ces différentes contributions ont fait l'objet des quatre publications suivantes :

- Cattan, E., Rochet-Capellan, A., Perrier, P., & Bérard, F. (2015, November). Reducing latency with a continuous prediction : Effects on users' performance in direct-touch target acquisitions. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (pp. 205-214). ACM.
- Cattan, E., Rochet-Capellan, A., & Bérard, F. (2015, November). A predictive approach for an end-to-end touch-latency measurement. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (pp. 215-218). ACM.
- Cattan, E., Rochet-Capellan, A., & Bérard, F. (2016, November). Effect of Touch Latency on Elementary vs. Bimanual Composite Tasks. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces* (pp. 103-108). ACM.
- Cattan, E., Rochet-Capellan, A., Perrier, P., & Bérard, F. (2017, May). Does Prac-

... Make Perfect?. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (pp. 5619-5629). ACM.

Un article est également en cours de révision pour un journal de neurosciences.

Perspectives

Nos contributions sur le diagnostic et la compensation de la latence ouvrent la voie vers plusieurs perspectives de recherches.

Nous avons montré des effets similaires de la latence pour l'interaction au toucher à une ou deux mains. Ces résultats semblent indiquer que la perception de la latence et l'influence de la latence sur la performance sont deux phénomènes décorrélés puisque la première décroît quand la charge cognitive augmente mais pas la seconde. De futures études seront nécessaires pour préciser ces résultats, par exemple en étudiant l'influence de la latence sur les performances de dessin ou d'écriture.

La méthode de compensation de la latence avec l'aide d'une prédiction a connu un essor important pendant la durée de cette thèse avec plusieurs travaux de recherches [34, 60, 103] mais aussi des intégrations dans des dispositifs commerciaux comme l'iPad Pro ou la Microsoft Surface. Le développement d'écran tactiles fonctionnant à 120 Hz [2] encourage d'autant plus l'utilisation d'une prédiction qui pourra être plus précise avec une plus grande quantité de données. Les modèles de prédiction ne doivent alors pas seulement être proposés mais aussi évalués sur les utilisateurs pour différentes tâches afin de quantifier le gain réel qu'ils apportent.

Enfin, notre étude sur l'adaptation à la latence nous a montré que l'influence de la latence varie avec le temps et l'entraînement. Ce travail de recherche montre l'importance de connecter la recherche en IHM à la recherche sur l'apprentissage sensorimoteur. Il est aussi une première étape vers des études à long terme sur la latence, idéalement *in situ*, qui fourniraient des données bien plus complètes sur le comportement des utilisateurs face à cette perturbation.

Bibliographie

- [1] Advanced touch input on ios. <https://developer.apple.com/videos/play/wwdc2015/233/>.
- [2] Apple keynote 2017. https://p-events-delivery.akamaized.net/17qopibbefvoihbsefbsefvopihb06/m3u8/hls_vod_mvp.m3u8.
- [3] Apple pencil. <https://www.apple.com/fr/apple-pencil/>.
- [4] Comment fonctionne nvidia g-sync? <http://www.nvidia.fr/object/how-does-g-sync-work-fr.html>.
- [5] Input lag testing. http://www.tftcentral.co.uk/articles/input_lag.htm#stopwatch.
- [6] Optitrack website. <http://optitrack.com>.
- [7] System performance verification – touchscreen technology series. <https://developer.sonymobile.com/2014/12/04/system-performance-verification-touchscreen-technology-series-4-video/>.
- [8] Touch prediction in microsoft api. [https://msdn.microsoft.com/en-us/%20en-%20us/library/windows/desktop/hh969214\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/%20en-%20us/library/windows/desktop/hh969214(v=vs.85).aspx).
- [9] Johnny Accot and Shumin Zhai. Beyond fitts' law : models for trajectory-based hci tasks. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 295–302. ACM, 1997.
- [10] Jack A Adams. Historical review and appraisal of research on the learning, retention, and transfer of human motor skills. *Psychological Bulletin*, 101(1) :41, 1987.
- [11] Bashar I Ahmad, Patrick M Langdon, Simon J Godsill, Richard Donkor, Rebecca Wilde, and Lee Skrypchuk. You do not have to touch to select : A study on predictive in-car touchscreen with mid-air selection. In *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pages 113–120. ACM, 2016.
- [12] Humza Akhtar and Qian Kemao. An accurate and efficient sampling algorithm for capacitive touch panels. In *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, pages 1–6. IEEE, 2016.

- [13] Robert S Allison, Laurence R Harris, Michael Jenkin, Urszula Jasiobedzka, and James E Zacher. Tolerance of temporal delay in virtual environments. In *Virtual Reality, 2001. Proceedings. IEEE*, pages 247–254. IEEE, 2001.
- [14] Glen Anderson, Rina Doherty, and Subhashini Ganapathy. User perception of touch screen latency. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice*, pages 195–202. Springer, 2011.
- [15] Michelle Annett, Albert Ng, Paul Dietz, Walter F. Bischof, and Anoop Gupta. How low should we go? : Understanding the perception of latency while inking. In *Proceedings of Graphics Interface (GI)*, 2014.
- [16] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino. Predictive interaction using the delphian desktop. In *Proceedings of ACM UIST*, pages 133–141, 2005.
- [17] Gisa Aschersleben. Temporal control of movements in sensorimotor synchronization. *Brain and cognition*, 48(1) :66–79, 2002.
- [18] FL Bedford. Perceptual learning. *The psychology of learning and motivation*, 30 :1–60, 1993.
- [19] François Bérard and Renaud Blanch. Two touch system latency estimators : High accuracy and low overhead. In *ACM ITS*, pages 241–250. ACM, 2013.
- [20] François Bérard. *Vision par ordinateur pour l'interaction homme-machine fortement couplée*. PhD thesis, Université Joseph-Fourier-Grenoble I, 1999.
- [21] Francois Berard and Thibault Louis. The object inside : Assessing 3d examination with a spherical handheld perspective-corrected display. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4396–4404. ACM, 2017.
- [22] François Bérard and Amélie Rochet-Capellan. The transfer of learning as hci similarity : Towards an objective assessment of the sensory-motor basis of naturalness. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1315–1324. ACM, 2015.
- [23] Justus Beyer, Richard Varbelow, Jan-Niklas Antons, and Steffen Zander. A method for feedback delay measurement using a low-cost arduino microcontroller : Lesson learned : Delay influenced by video bitrate and game-level. In *Quality of Multimedia Experience (QoMEX), 2015 Seventh International Workshop on*, pages 1–2. IEEE, 2015.
- [24] Pradipta Biswas, Gokcen Aslan Aydemir, Pat Langdon, and Simon Godsill. Intent recognition using neural networks and kalman filters. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, pages 112–123. Springer, 2013.
- [25] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. Semantic pointing : improving target acquisition with control-display ratio adaptation. In *Proceedings*

- of the SIGCHI conference on Human factors in computing systems, pages 519–526. ACM, 2004.
- [26] Renaud Blanch and Michael Ortega. Benchmarking pointing techniques with distractors : adding a density factor to fitts’ pointing paradigm. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1629–1638. ACM, 2011.
- [27] Mike Bond. Boom! headshot!
- [28] Stephen Brewster. Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing*, 6(3) :188–205, 2002.
- [29] Stephen Brewster, Faraz Chohan, and Lorna Brown. Tactile feedback for mobile interactions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159–162. ACM, 2007.
- [30] Timothy J Buker, Dennis A Vincenzi, and John E Deaton. The effect of apparent latency on simulator sickness while using a see-through helmet-mounted display reducing apparent latency with predictive compensation. *Human Factors : The Journal of the Human Factors and Ergonomics Society*, 54(2) :235–249, 2012.
- [31] KIM Byoung-eul and Yeongkyu Lim. Mobile terminal and touch coordinate predicting method thereof, January 23 2014. US Patent App. 14/760,065.
- [32] Géry Casiez, Stéphane Conversy, Matthieu Falce, Stéphane Huot, and Nicolas Roussel. Looking through the eye of the mouse : A simple method for measuring end-to-end latency using an optical mouse. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 629–636. ACM, 2015.
- [33] Géry Casiez, Thomas Pietrzak, Damien Marchal, Sébastien Poulmane, Matthieu Falce, and Nicolas Roussel. Characterizing latency in touch and button-equipped interactive systems. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2017.
- [34] Élie Cattan and François Bérard. Reducing latency with a continuous prediction : Effects on users’ performance in direct-touch target acquisitions. In *ACM ITS*. ACM, 2015.
- [35] Elie Cattan, Amélie Rochet-Capellan, and François Bérard. A predictive approach for an end-to-end touch-latency measurement. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, pages 215–218. ACM, 2015.
- [36] Elie Cattan, Amélie Rochet-Capellan, and François Bérard. Effect of touch latency on elementary vs. bimanual composite tasks. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, pages 103–108. ACM, 2016.
- [37] Elie Cattan, Amélie Rochet-Capellan, Pascal Perrier, and François Bérard. Does practice make perfect ? In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5619–5629. ACM, 2017.

- [38] Mark Claypool and Kaja Claypool. Latency and player actions in online games. *Communications of the ACM*, 49(11) :40–45, 2006.
- [39] Sarah E Criscimagna-Hemminger and Reza Shadmehr. Consolidation patterns of human motor memory. *Journal of Neuroscience*, 28(39) :9610–9618, 2008.
- [40] Douglas W Cunningham, Vincent A Billock, and Brian H Tsou. Sensorimotor adaptation to violations of temporal contiguity. *Psychological Science*, 12(6) :532–535, 2001.
- [41] Douglas W Cunningham, Astros Chatziastros, Markus von der Heyde, and Heinrich H Bülthoff. Temporal adaptation and the role of temporal contiguity in spatial behavior. 2000.
- [42] Douglas W Cunningham, Astros Chatziastros, Markus Von der Heyde, and Heinrich H Bülthoff. Driving in the future : temporal visuomotor adaptation and generalization. *Journal of vision*, 1(2) :3–3, 2001.
- [43] Cristina de la Malla, Joan López-Moliner, and Eli Brenner. Seeing the last part of a hitting movement is enough to adapt to a temporal delay. *Journal of Vision*, 12(10) :4–4, 2012.
- [44] Cristina de la Malla, Joan López-Moliner, and Eli Brenner. Dealing with delays does not transfer across sensorimotor tasks. *Journal of vision*, 14(12) :8, 2014.
- [45] Jonathan Deber, Bruno Araujo, Ricardo Jota, Clifton Forlines, Darren Leigh, Steven Sanders, and Daniel Wigdor. Hammer time! : A low-cost, high precision, high accuracy tool to measure the latency of touchscreen devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2857–2868. ACM, 2016.
- [46] Jonathan Deber, Ricardo Jota, Clifton Forlines, and Daniel Wigdor. How much faster is fast enough? : User perception of latency & latency improvements in direct and indirect touch. In *Proceedings of ACM CHI*, 2015.
- [47] Paul Dietz and Darren Leigh. Diamondtouch : a multi-user touch technology. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226. ACM, 2001.
- [48] Paul DiZio and James R Lackner. Circumventing side effects of immersive virtual environments. In *HCI (2)*, pages 893–896, 1997.
- [49] Julien Doyon, Maria Korman, Amélie Morin, Valérie Dostie, Abdallah Hadj Tahar, Habib Benali, Avi Karni, Leslie G Ungerleider, and Julie Carrier. Contribution of night and day sleep vs. simple passage of time to the consolidation of motor sequence and visuomotor adaptation learning. *Experimental Brain Research*, 195(1) :15–26, 2009.
- [50] Mark H Draper, Erik S Viirre, Thomas A Furness, and Valerie J Gawron. Effects of image scale and system time delay on simulator sickness within head-coupled virtual environments. *Human factors*, 43(1) :129–146, 2001.

- [51] Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6) :381–391, 1954.
- [52] Alexander J McC Foulkes and R Chris Miall. Adaptation to visual feedback delays in a human manual tracking task. *Experimental Brain Research*, 131(1) :101–110, 2000.
- [53] Karl Friston. What is optimal about motor control? *Neuron*, 72(3) :488–498, 2011.
- [54] Sebastian Friston, Per Karlstrom, and Anthony Steed. The effects of low latency on pointing and steering tasks. 2014.
- [55] Sebastian Friston and Anthony Steed. Measuring latency in virtual environments. *IEEE transactions on visualization and computer graphics*, 20(4) :616–625, 2014.
- [56] Yves Guiard. Asymmetric division of labor in human skilled bimanual action : The kinematic chain as a model. *Journal of motor behavior*, 19(4) :486–517, 1987.
- [57] Ding He, Fuhu Liu, Dave Pape, Greg Dawe, and Dan Sandin. Video-based measurement of system latency. In *International Immersive Projection Technology Workshop*, 2000.
- [58] Richard Held, Aglaia Efstathiou, and Martha Greene. Adaptation to displaced and delayed visual feedback from the hand. *Journal of Experimental Psychology*, 72(6) :887, 1966.
- [59] Richard Helps and Clarissa Helps. Measuring stylus and tablet performance for usability in sketching. In *Proceedings of the 5th Annual Conference on Research in Information Technology*, pages 19–24. ACM, 2016.
- [60] Niels Henze, Markus Funk, and Alireza Sahami Shirazi. Software-reduced touchscreen latency. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 434–441. ACM, 2016.
- [61] John W Hill. Comparison of seven performance measures in a time-delayed manipulation task. *IEEE Transactions on Systems, Man, and Cybernetics*, (4) :286–295, 1976.
- [62] Errol R Hoffmann. Fitts’ law with transmission delay. *Ergonomics*, 35(1) :37–48, 1992.
- [63] Takuya Honda, Masaya Hirashima, and Daichi Nozaki. Habituation to feedback delay restores degraded visuomotor adaptation by altering both sensory prediction error and the sensitivity of adaptation to the error. *Frontiers in psychology*, 3 :540, 2012.
- [64] Torsten Hothorn, Frank Bretz, and Peter Westfall. Simultaneous inference in general parametric models. *Biometrical journal*, 50(3) :346–363, 2008.
- [65] Zenja Ivkovic, Ian Stavness, Carl Gutwin, and Steven Sutcliffe. Quantifying and mitigating the negative effects of local latencies on aiming in 3d shooter games. In *Proceedings of ACM CHI*, 2015.

- [66] Marco C Jacobs, Mark A Livingston, et al. Managing latency in complex augmented reality systems. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 49–ff. ACM, 1997.
- [67] Benjamin F Janzen and Robert J Teather. Is 60 fps better than 30? : the impact of frame rate and latency on moving target selection. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems*, pages 1477–1482. ACM, 2014.
- [68] Jason J Jerald. *Scene-motion-and latency-perception thresholds for head-mounted displays*. The University of North Carolina at Chapel Hill, 2009.
- [69] Ricardo Jota, Albert Ng Ng, Paul Dietz, and Daniel Wigdor. How fast is fast enough? a study of the effects of latency in direct-touch pointing tasks. In *ACM CHI*. ACM, 2013.
- [70] Topi Kaaresoja and Stephen Brewster. Feedback is... late : measuring multimodal delays in mobile device touchscreen interaction. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, page 2. ACM, 2010.
- [71] Topi Kaaresoja, Stephen Brewster, and Vuokko Lantz. Towards the temporally perfect virtual button : touch-feedback simultaneity and perceived quality in mobile touchscreen press interactions. *ACM Transactions on Applied Perception (TAP)*, 11(2) :9, 2014.
- [72] Paul Kabbash, William Buxton, and Abigail Sellen. Two-handed input in a compound task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 417–423. ACM, 1994.
- [73] Paul Kabbash, I Scott MacKenzie, and William Buxton. Human performance using computer input devices in the preferred and non-preferred hands. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, pages 474–481. ACM, 1993.
- [74] Teemu Kämäräinen, Matti Siekkinen, Antti Ylä-Jääski, Wenxiao Zhang, and Pan Hui. Dissecting the end-to-end latency of interactive mobile video applications. *arXiv preprint arXiv :1611.08520*, 2016.
- [75] Yu Kang, Yangfan Zhou, Min Gao, Yixia Sun, and Michael R Lyu. Experience report : Detecting poor-responsive ui in android applications. In *Software Reliability Engineering (ISSRE), 2016 IEEE 27th International Symposium on*, pages 490–501. IEEE, 2016.
- [76] Youngchun Kim and Ahmed H Tewfik. Low energy and latency touch detection using group testing. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 21st European*, pages 1–5. IEEE, 2013.

- [77] Tomoko Kitago, Sophia L Ryan, Pietro Mazzoni, John W Krakauer, and Adrian M Haith. Unlearning versus savings in visuomotor adaptation : comparing effects of washout, passage of time, and removal of errors on motor memory. 2013.
- [78] Jarrod Knibbe, Hrvoje Benko, and Andrew D Wilson. Juggling the effects of latency : Motion prediction approaches to reducing latency in dynamic projector-camera systems. 2015.
- [79] JJ Kozak, PA Hancock, EJ Arthur, and ST Chrysler. Transfer of training from virtual reality. *Ergonomics*, 36(7) :777–784, 1993.
- [80] John W Krakauer and Pietro Mazzoni. Human sensorimotor learning : adaptation, skill, and beyond. *Current opinion in neurobiology*, 21(4) :636–644, 2011.
- [81] John W Krakauer, Pietro Mazzoni, Ali Ghazizadeh, Roshni Ravindran, and Reza Shadmehr. Generalization of motor learning depends on the history of prior action. *PLoS Biol*, 4(10) :e316, 2006.
- [82] John W Krakauer, Zachary M Pine, Maria-Felice Ghilardi, and Claude Ghez. Learning of visuomotor transformations for vectorial planning of reaching trajectories. *Journal of Neuroscience*, 20(23) :8916–8924, 2000.
- [83] Francesco Lacquaniti, Carlo Terzuolo, and Paolo Viviani. The law relating the kinematic and figural aspects of drawing movements. *Acta psychologica*, 54(1) :115–130, 1983.
- [84] Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. Endpoint prediction using motion kinematics. In *Proceedings of ACM CHI*, pages 637–646. ACM, 2007.
- [85] Celine Latulipe, Craig S Kaplan, and Charles LA Clarke. Bimanual and unimanual image alignment : an evaluation of mouse-based techniques. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 123–131. ACM, 2005.
- [86] Steven M LaValle, Anna Yershova, Max Katsev, and Michael Antonov. Head tracking for the oculus rift. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 187–194. IEEE, 2014.
- [87] Joseph J. LaViola. Double exponential smoothing : An alternative to kalman filter-based predictive tracking. In *Proceedings of the Workshop on Virtual Environments 2003*, EGVE '03, pages 199–206, New York, NY, USA, 2003. ACM.
- [88] Darren Leigh, Clifton Forlines, Ricardo Jota, Steven Sanders, and Daniel Wigdor. High rate, low-latency multi-touch sensing with simultaneous orthogonal multiplexing. In *Proceedings of ACM UIST*, pages 355–364. ACM, 2014.
- [89] Eva Lenz, Sarah Diefenbach, and Marc Hassenzahl. Exploring relationships between interaction attributes and experience. In *Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces*, pages 126–135. ACM, 2013.

- [90] Jiandong Liang, Chris Shaw, and Mark Green. On temporal-spatial realism in the virtual reality environment. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*, pages 19–25. ACM, 1991.
- [91] Jan Staffan Lincoln. Position lag reduction for computer drawing, April 11 2012. US Patent App. 13/444,029.
- [92] I Scott MacKenzie. Movement time prediction in human-computer interfaces. In *Proceedings of Graphics Interface*, volume 92, page 1, 1992.
- [93] I Scott MacKenzie and Colin Ware. Lag as a determinant of human performance in interactive systems. In *Proceedings of ACM CHI*, pages 488–493, 1993.
- [94] TA Martin, JG Keating, HP Goodkin, AJ Bastian, and WT Thach. Throwing while looking through prisms. *Brain*, 119(4) :1183–1198, 1996.
- [95] Pietro Mazzoni and John W Krakauer. An implicit plan overrides an explicit strategy during visuomotor adaptation. *Journal of neuroscience*, 26(14) :3642–3645, 2006.
- [96] RC Miall and JK Jackson. Adaptation to visual feedback delays in manual tracking : evidence against the smith predictor model of human visually guided action. *Experimental Brain Research*, 172(1) :77–84, 2006.
- [97] Matthew K Miller and Regan L Mandryk. Differentiating in-game frustration from at-game frustration using touch pressure. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, pages 225–234. ACM, 2016.
- [98] Mark R Mine. Characterization of end-to-end delays in head-mounted display systems. *The University of North Carolina at Chapel Hill, TR93-001*, 1993.
- [99] Shio Miyafuji, Masato Sugasaki, and Hideki Koike. Ballumiere : Real-time tracking and spherical projection for high-speed moving balls. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces*, pages 33–37. ACM, 2016.
- [100] Farshid Moussavi. Methods and apparatus for incremental prediction of input device motion, July 1 2014. US Patent 8,766,915.
- [101] Roger Mundry and Charles L Nunn. Stepwise model fitting and statistical inference : turning noise into signal pollution. *The American Naturalist*, 173(1) :119–123, 2008.
- [102] Arne J Nagengast, Daniel A Braun, and Daniel M Wolpert. Optimal control predicts human performance on objects with internal degrees of freedom. *PLoS Comput Biol*, 5(6) :e1000419, 2009.
- [103] Mathieu Nancel, Daniel Vogel, Bruno De Araujo, Ricardo Jota, and Géry Casiez. Next-point prediction metrics for perceived spatial errors. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 271–285. ACM, 2016.
- [104] Albert Ng, Michelle Annett, Paul Dietz, Anoop Gupta, and Walter F. Bischof. In the blink of an eye : Investigating latency perception during stylus interaction. In *ACM CHI*, pages 1103–1112. ACM, 2014.

- [105] Albert Ng and Paul H Dietz. The effects of latency and motion blur on touch screen user experience. *Journal of the Society for Information Display*, 2015.
- [106] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. Designing for low-latency direct-touch input. In *Proceedings of ACM UIST*, pages 453–464. ACM, 2012.
- [107] DA Norman. The way i see it : Natural user interfaces are not natural. *interactions* 17, 3, 2010.
- [108] Giorgos Papadakis, Katerina Mania, and Eftichios Koutroulis. A system to measure, control and minimize end-to-end head tracking latency in immersive simulations. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 581–584. ACM, 2011.
- [109] Phillip T Pasqual and Jacob O Wobbrock. Mouse pointing endpoint prediction using kinematic template matching. In *Proceedings of ACM CHI*, pages 743–752. ACM, 2014.
- [110] Andriy Pavlovyh and Carl Gutwin. Assessing target acquisition and tracking performance for complex moving targets in the presence of latency and jitter. In *Proceedings of the 2012 Graphics Interface Conference*, pages 109–116. Canadian Information Processing Society, 2012.
- [111] Andriy Pavlovyh and Wolfgang Stuerzlinger. The tradeoff between spatial jitter and latency in pointing tasks. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, pages 187–196. ACM, 2009.
- [112] Andriy Pavlovyh and Wolfgang Stuerzlinger. Target following performance in the presence of latency, jitter, and signal dropouts. In *Proceedings of Graphics Interface 2011*, pages 33–40. Canadian Human-Computer Communications Society, 2011.
- [113] Jose Pinheiro, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, and R Core Team. *nlme : Linear and Nonlinear Mixed Effects Models*, 2017. R package version 3.1-131.
- [114] Réjean Plamondon and Adel M Alimi. Speed/accuracy trade-offs in target-directed movements. *Behavioral and brain sciences*, 20(02) :279–303, 1997.
- [115] Claude Prablanc and Olivier Martin. Automatic control during hand reaching at undetected two-dimensional target displacements. *Journal of neurophysiology*, 67(2) :455–469, 1992.
- [116] R Development Core Team. *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.
- [117] Kjetil Raaen and Ragnhild Eg. Instantaneous human-computer interactions : Button causes and screen effects. In *International Conference on Human-Computer Interaction*, pages 492–502. Springer, 2015.

- [118] Jun Rekimoto. Smartskin : an infrastructure for freehand manipulation on interactive surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120. ACM, 2002.
- [119] Walter Ritter, Guido Kempter, and Tobias Werner. User-acceptance of latency in touch interactions. In *International Conference on Universal Access in Human-Computer Interaction*, pages 139–147. Springer, 2015.
- [120] Ryan T Roemmich and Amy J Bastian. Two ways to save a newly learned motor pattern. *Journal of neurophysiology*, 113(10) :3519–3530, 2015.
- [121] Marieke Rohde and Marc O Ernst. Time, agency, and sensory feedback delays during action. *Current Opinion in Behavioral Sciences*, 8 :193–199, 2016.
- [122] Marieke Rohde, Loes CJ van Dam, and Marc O Ernst. Predictability is necessary for closed-loop visual feedback delay adaptation. *Journal of vision*, 14(3) :4, 2014.
- [123] Jeffrey A Saunders and David C Knill. Humans use continuous visual feedback from the hand to control fast reaching movements. *Experimental Brain Research*, 152(3) :341–352, 2003.
- [124] Benoit Schillings, Feng Qian, Jonathan Kaldor, and George Djabarov. Predicting touch input, January 24 2013. US Patent App. 13/749,543.
- [125] Frank C Seto. System and method of measuring continuous touch controller latency, September 23 2015. US Patent App. 14/863,323.
- [126] Reza Shadmehr and Ferdinando A Mussa-Ivaldi. Adaptive representation of dynamics during learning of a motor task. *Journal of Neuroscience*, 14(5) :3208–3224, 1994.
- [127] Reza Shadmehr, Maurice A Smith, and John W Krakauer. Error correction, sensory prediction, and adaptation in motor control. *Annual review of neuroscience*, 33 :89–108, 2010.
- [128] Thomas B Sheridan and William R Ferrell. Remote manipulative control with transmission delay. *IEEE Transactions on Human Factors in Electronics*, (1) :25–29, 1963.
- [129] Tobias Sielhorst, Wu Sa, Ali Khamene, Frank Sauer, and Nassir Navab. Measurement of absolute latency for video see through augmented reality. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2007.
- [130] Maurice A Smith, Ali Ghazizadeh, and Reza Shadmehr. Interacting adaptive processes with different timescales underlie short-term motor learning. *PLoS Biol*, 4(6) :e179, 2006.
- [131] Richard HY So and German KM Chung. Sensory motor responses in virtual environments : Studying the effects of image latencies for target-directed hand movement. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 5006–5008. IEEE, 2005.

- [132] R. William Soukoreff and I. Scott MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies*, 61(6) :751 – 789, 2004.
- [133] Anthony Steed. A simple method for estimating the latency of interactive, real-time graphics simulations. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 123–129. ACM, 2008.
- [134] Chess Stetson, Xu Cui, P Read Montague, and David M Eagleman. Motor-sensory recalibration leads to an illusory reversal of action and sensation. *Neuron*, 51(5) :651–659, 2006.
- [135] Yoshimori Sugano, Mirjam Keetels, and Jean Vroomen. The build-up and transfer of sensorimotor temporal recalibration measured via a synchronization task. *Frontiers in psychology*, 3 :246, 2012.
- [136] Colin Swindells, John C Dill, and Kellogg S Booth. System lag tests for augmented and virtual environments. In *Proceedings of ACM UIST*, 2000.
- [137] Robert J Teather, Andriy Pavlovyh, Wolfgang Stuerzlinger, and I Scott MacKenzie. Effects of tracking technology, latency, and spatial jitter on object movement. In *3D User Interfaces, 2009. 3DUI 2009. IEEE Symposium on*, pages 43–50. IEEE, 2009.
- [138] Sebastian Telgen, Darius Parvin, and Jörn Diedrichsen. Mirror reversal and visual rotation are learned and consolidated via separate mechanisms : recalibrating or learning de novo? *Journal of Neuroscience*, 34(41) :13768–13779, 2014.
- [139] Emanuel Todorov and Michael I Jordan. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11) :1226–1235, 2002.
- [140] Peter Tsoi and Jacob Xiao. Advanced touch input on ios. Technical report, Apple Inc., 2015.
- [141] Xiao Tu, Fei Xiong, Jianfeng Lin, and Patrick Jee-An Poon. Wet ink predictor, September 16 2015. US Patent App. 14/856,167.
- [142] Edward JA Turnham, Daniel A Braun, and Daniel M Wolpert. Inferring visuomotor priors for sensorimotor learning. *PLoS Comput Biol*, 7(3) :e1001112, 2011.
- [143] Willem A Van Bergeijk and Edward E David Jr. Delayed handwriting. *Perceptual and Motor Skills*, 9(3) :347–357, 1959.
- [144] J-L Vercher and GM Gauthier. Oculo-manual coordination control : ocular and manual tracking of visual targets with delayed visual feedback of the hand motion. *Experimental Brain Research*, 90(3) :599–609, 1992.
- [145] A Vlachos. Advanced vr rendering. In *Game Developers Conference*, 2015.
- [146] Hermann Von Helmholtz. *Handbuch der physiologischen Optik*, volume 9. Voss, 1867.
- [147] Geoff Walker. A review of technologies for sensing contact location on the surface of a display. *Journal of the Society for Information Display*, 20(8) :413–440, 2012.

- [148] Matthew P Walker, Tiffany Brakefield, Alexandra Morgan, J Allan Hobson, and Robert Stickgold. Practice with sleep makes perfect : sleep-dependent motor skill learning. *Neuron*, 35(1) :205–211, 2002.
- [149] Pierre Wellner. Interacting with paper on the digitaldesk. *Communications of the ACM*, 36(7) :87–96, 1993.
- [150] Daniel Wigdor, Sarah Williams, Michael Cronin, Robert Levy, Katie White, Maxim Mazeev, and Hrvoje Benko. Ripples : utilizing per-contact visualizations to improve user interaction with touch displays. In *Proceedings of ACM UIST*, UIST '09, pages 3–12. ACM, 2009.
- [151] Daniel Wigdor and Dennis Wixon. *Brave NUI world : designing natural user interfaces for touch and gesture*. Elsevier, 2011.
- [152] Daniel M Wolpert, Jörn Diedrichsen, and J Randall Flanagan. Principles of sensorimotor learning. *Nature Reviews Neuroscience*, 12(12) :739–751, 2011.
- [153] Daniel M Wolpert and J Randall Flanagan. Computations underlying sensorimotor learning. *Current opinion in neurobiology*, 37 :7–11, 2016.
- [154] Daniel M Wolpert, R Chris Miall, and Mitsuo Kawato. Internal models in the cerebellum. *Trends in cognitive sciences*, 2(9) :338–347, 1998.
- [155] Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. Zero-latency tapping : using hover information to predict touch locations and eliminate touchdown latency. In *Proceedings of ACM UIST*, pages 205–214. ACM, 2014.
- [156] Min Hong Yun, Songtao He, and Lin Zhong. Reducing latency by eliminating synchrony. In *Proceedings of the 26th International Conference on World Wide Web*, pages 331–340. International World Wide Web Conferences Steering Committee, 2017.