PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à MINES ParisTech

Dynamic Adaptation of Human-Computer Interfaces Using Cognitive Load Tracking

Adaptation Dynamique des Interfaces Homme-Machine en Utilisant du Suivi de Charge Cognitive

Soutenue par

Bruno MASSONI SGUERRA

Le 19 décembre 2019

École doctorale n°621

Ingénierie des systèmes, matériaux, mécanique, énergétique

Spécialité

Informatique temps-réel, robotique et automatique

Composition du jury :

Andrew HOWES Professeur, University of Birmingham	*Rapporteur*
Charles TIJUS Professeur, Universié Paris 8	*Rapporteur*
Isis Truck Professeur, Universié Paris 8	*Présidente du jury*
Samuel Benveniste Docteur, CEN STIMCO	*Examinateur*
Pierre JOUVELOT Directeur de recherche, MINES ParisTech	*Directeur de thèse*

MINES ParisTech | PSL

Any fool can know. The point is to understand.

Albert Einstein

Acknowledgements

This thesis is not only the fruit of my work. It's a result of the sum of efforts and of support I received from many people.

First, I'd like to thank my supervisors, Pierre Jouvelot and Samuel Benveniste. Thank you for your availability and all the useful (and not so useful) discussions we had. Thank you for your rigor and for allowing me to develop my own ideas and to have fun.

I'd like to thank the two interns I had the pleasure to supervise, Amine Benamara and Soha Lagneb. This thesis is significantly supported by your work and the ideas we discussed. Thank you very much.

I also would like to thank my colleagues at the Centre de recherche en informatique of MINES ParisTech (starting by the center head) François Irigoin, Corinne Ancourt, Fabien Coelho, Laurent Daverio, Emilio Gallego Arias, Olivier Hermant, Claire Medrala and Claude Tadonki; thank you for your welcome and help. I would like to thank the PhD students, Maksim Berezov, Pierre Guillou, Patryk Kiepas, Lucas Massoni Sguerra, Adila Susungi and Pierre Wargnier; thank you for all the good times we had in Fontainebleau.

I'd like to thank everyone from the LUSAGE and CEN STIMCO team. I really appreciated the warm environment of the everyday life at the Broca Hospital. Thank you, Benoit Charlieux, Manon Demange, Anaëlle Durand, Philippe De Oliveira Lopes, Baptiste Isabet, Maribel Pino, Claudia Sehnal and Maria José Urbiola Gallegos.

I'd like to thank all my friends that supported me during these three years of thesis. Life in Paris would not be the same without you: a special thank you, then, to Marie Vialle, Mickael Pierre, Manon Demange, Amanda Fernandes, Gabriella Bettonte, Maryna Savchenko, Maksim Berezov, Nelson Gomes and Yeongran Kim; you have my deepest appreciation.

Lastly, I'd like to thank my family, my mother and father and my brothers for all the great support they have given me, for always pushing me forward and for all their love and incentive. This work would not have been possible without you: thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"En général, nous sommes moins conscients de ce que nos esprits font le mieux."

Marvin Minsky

En 1967, l'écrivain américain John M. Culkin a déclaré : "Nous façonnons nos outils, et ceux-ci, à leur tour, nous façonnent." En 2019, 52 ans plus tard, cela ne pourrait pas être plus vrai. Nous vivons actuellement une révolution technologique où la technologie progresse à un rythme exponentiel. Il n'a jamais été aussi rapide, ni facile, de créer de nouveaux outils et de les mettre entre les mains des utilisateurs. Ces nouveaux outils technologiques remodèlent les canaux d'information et les connexions humaines, tout en remodelant les emplois, les loisirs et, par conséquent, le sens de l'être humain. Un flot de nouveaux gadgets, smartphones, montres intelligentes, applications, chatbots et autres déferle sur les consommateurs tous les jours. Cette vague de nouveaux outils exige à terme une adaptation des utilisateurs, une lutte constante pour s'adapter afin de ne pas devenir obsolète et conserver sa place dans la société. Jamais autant de nouveaux outils ont été créés à une telle vitesse ; jamais auparavant la vie humaine n'a changé aussi rapidement.

La majeure partie de ce développement technologique révolutionnaire est aveuglée par l'objectif d'être plus précis, plus rapide et plus efficace. Il y a un certain nombre de raisons qui rendent ces mesures dignes d'être poursuivies. Pourtant, les développeurs ont tendance à oublier de mettre le facteur humain dans la fonction de perte qu'ils ciblent. Quelle que soit la résilience des êtres humains, capables de s'adapter à des environnements difficiles et d'apprendre à gérer des interfaces complexes, il existe des limitations intrinsèques qui sont tout simplement inévitables, des limitations telles comme la vitesse de traitement du cerveau humain, l'énergie

de l'attention disponible, les habitudes de sommeil, la capacité de mémoire, la vitesse d'apprentissage et bien d'autres. Celles-ci peuvent être considérées comme les limitations matérielles du cerveau.

Les technologies destinées aux humains doivent être conçues autour de l'homme, ce qui signifie qu'il faut tenir compte du fonctionnement de l'esprit humain lors de la conception d'un nouvel outil, tout en tenant compte des limitations humaines. C'est un moyen d'assurer de meilleures adaptation et acceptation humaines à une nouvelle technologie, un moyen de lisser la courbe d'apprentissage et, finalement, de rendre ces outils plus efficaces. Dans son livre "Sum: Forty Tales from the Afterlives" de 2009, le neuroscientifique américain David Eagleman écrit, à propos d'une race fictive en proie à des questions comme "pourquoi sommes-nous ici ?" ou "quel est le but de l'existence ?", qu'ils ont décidé d'investir pendant des générations dans le développement d'une machine de calcul intensif dédiée à trouver des réponses. Cependant, le projet a lamentablement échoué, car la machine résultante était bien trop avancée pour interagir avec ces êtres. Lorsque vous développez une machine plus complexe et intelligente que vous, votre capacité à comprendre la machine commence à diminuer. La technologie la plus avancée peut être sous-utilisée si l'utilisateur ne parvient pas à interagir correctement avec elle, car "un outil est aussi bon que l'est son utilisateur", comme on le dit souvent.

Les limitations de l'esprit humain représentent le point de rupture au-delà duquel les technologies ne peuvent plus se déployer. Il n'est pas souhaitable de concevoir des interfaces qui clignotent le maximum d'informations pouvant être affichées sur un écran à la vitesse la plus élevée possible. De fait, ce ne sont pas des paramètres visés lors du développement d'un nouvel outil. Instinctivement, les concepteurs, étant eux-mêmes des humains, savent qu'il existe des limites cognitives à la quantité d'informations pouvant être traitées par les utilisateurs. La plupart des bonnes interfaces restent simples ; une quantité minimaliste de boutons et d'options affichés aide les utilisateurs à concentrer leur attention et à décider quelle commande ils doivent utiliser. Cependant, l'affichage de la quantité minimale d'informations n'est pas toujours souhaitable non plus. Par exemple, la quantité "minimale" d'informations est subjective. Selon la maîtrise de l'utilisateur sur le sujet concernant les données affichées, il ou elle peut avoir plus ou moins de facilité à réfléchir sur différentes quantités de données. Par exemple, le jeu de Go consiste en deux joueurs posant des pierres blanches et noires sur un plateau. Le tablier de Go traditionnel a une grille de 19×19*, et à chaque intersection une pierre peut être placée. Puisqu'il y a 3 états possibles pour chaque intersection (vide, noir ou blanc), le nombre de configurations possibles du tablier est* $3^{361} \approx 1,7 \times 10^{172}$ *(pas toutes les configurations ne sont possibles en raison des règles du jeu ; John Tromp a déterminé qu'il existe environ* $2,08 \times 10^{170}$ *configurations légales [2]). Cette quantité incroyable de configurations possibles est une*

des raisons pour lesquelles les machines ont eu du mal à battre les joueurs de Go jusqu'à que l'AlphaGo de Google apparaisse en 2016. Cependant, les joueurs de Go avancés peuvent jouer une variante du jeu à une seule couleur, où les joueurs utilisent des pierres de la même couleur, ce qui signifie que les joueurs doivent se rappeler quelle pierre est laquelle en se souvenant de la progression du jeu dans son ensemble. Cet exploit impressionnant peut sembler impossible pour un débutant, mais il ne fait que montrer comment le niveau de maîtrise affecte la perception d'informations complexes.

Trouver le bon équilibre entre la complexité d'une interface et la façon dont l'utilisateur perçoit cette complexité peut être la clé pour étendre les performances humaines et technologiques au maximum. Cela nécessite une adaptation continue des canaux de communication entre homme et machine. En intégrant des interfaces avec une connaissance approfondie du fonctionnement de l'esprit humain, la technologie pourrait compenser les limitations humaines, faciliter l'apprentissage et accroître l'accessibilité. Cependant, non seulement en servant de béquille pour l'esprit, mais en adaptant régulièrement les systèmes tout en tenant compte de l'apprentissage humain, une nouvelle technologie, une fois assimilée, pourrait avancer en permanence les performances humaines en compensant les limitations là où elles apparaissent, mais en stimulant les domaines où l'utilisateur est déjà compétent. Cela signifie que les interfaces des outils doivent être façonnées par le fonctionnement de l'esprit. La conception de nouveaux outils et de leurs interfaces doit être centrée autour de ces limitations et avec une connaissance approfondie de la façon dont nous "travaillons", en tant qu'êtres humains. Cela pourrait être un facteur clé dans cette ère fondée sur les données, permettant aux nouvelles technologies de nous changer, non pas en allant contre les utilisateurs mais en les guidant.

1.1 Objectif

Le but de ce travail est d'explorer les façons possibles d'adapter les systèmes informatiques à la façon dont les humains traitent l'information. Ce travail s'intéresse particulièrement à l'adaptation des interfaces utilisateurs (UI) aux limites de la mémoire de travail humaine (WM, pour "Working Memory"). La WM est la partie de la cognition humaine responsable du stockage et du traitement des informations verbales et visuelles. Il a été reconnu comme un goulot d'étranglement majeur de la capacité humaine de traitement de l'information. Par conséquent, en termes plus précis, l'objectif de cette thèse est de réfléchir aux possibilités de rendre les systèmes informatiques "conscients" et capables de compenser les limitations de la WM des utilisateurs.

On peut alors définir quatre étapes afin d'effectuer une adaptation appropriée (fondée sur la méthodologie de [3]):

1. *déduire la limitation de capacité de l'utilisateur ;*
2. *identifier l'impact potentiel sur la performance ;*
3. *sélectionner une stratégie compensatoire ;*
4. *appliquer cette stratégie dans le contexte actuel.*

Ces étapes, spécialement les étapes 1 et 2, sont les principaux défis abordés dans ce travail. Étant donné que les limitations cognitives ne sont pas facilement mesurables (autrement qu'en effectuant des tests spécifiques et très contraints), il faut être capable de déduire ces limitations à partir de l'interaction de l'utilisateur avec le système. L'évaluation de l'état de l'utilisateur (qu'il soit cognitif, physique, propre à la personnalité ou affectif) est l'une des fonctionnalités essentielles qui doivent être traitées par des systèmes efficaces de modélisation et d'adaptation de l'utilisateur [3].

Dans ce contexte, l'adaptation nécessite que le système ait une connaissance inhérente de la façon dont une capacité limitée influe sur les performances et comment une action pourrait compenser une telle limitation. Pour ce faire, un système doit aller au-delà des connaissances d'observation liant certains paramètres en entrée aux résultats observés que les techniques traditionnelles d'apprentissage automatique (ML ou 'Machine Learning") peuvent fournir. Les technologies supervisées fondées sur le ML manquent de flexibilité, et elles sont limitées par la quantité de données de l'ensemble d'apprentissage. Logiquement, fournir au système une image complète de tous les différents scénarios possibles qu'il pourrait rencontrer compenserait cette rigidité. Cependant, cela nécessiterait une énorme quantité de données. L'utilisation d'un modèle fondé sur la compréhension de l'utilisateur permet d'aller au-delà des données collectées précédemment et de traiter des cas non encore rencontrés.

Un tel système a besoin d'un modèle des processus causaux sous-jacents impliqués dans la façon dont les limites de capacité affectent la performance. D'une manière générale, les modèles, de divers niveaux d'abstraction, représentent des simplifications d'un système qui servent d'explication limitée à certains des nombreux mécanismes qui le composent et qui fonctionnent ensemble. Les modèles peuvent différer en termes de complexité et de fiabilité, mais ils représentent la compréhension que l'on a d'un phénomène; elle peut prendre la forme d'une théorie ou, dans certains cas, d'une loi. Le fait d'avoir un modèle causal de l'impact des limitations de capacité sur les performances permet de prédire comment la capacité présumée affecte les performances ainsi que de sélectionner des mesures compensatoires adéquates.

L'objectif principal de ce travail est donc d'explorer des méthodes efficaces pour intégrer un système informatique avec un modèle de WM humaine afin de permettre

l'adaptation de l'interface utilisateur aux limitations de la WM des utilisateurs.

1.2 Contributions

Ce travail comprend cinq contributions majeures:

- *Memory Adaptation Through Cognitive Handling Simulation (MATCHS), un nouveau cadre pour la modélisation dynamique, le suivi et l'adaptation des tâches, dont les performances dépendent de la WM, aux limitations perçues de la WM;*

- *An Uscented Hound for Working Memory (AUHWM), une extension de MATCHS, accroissant ses capacités afin de permettre la modélisation en temps réel et le suivi des performances de la WM humaine, qui utilise un modèle déterministe de mémoire de travail et un filtrage de Kalman non linéaire ;*

- *une évaluation expérimentale de la capacité de MATCHS et AUHWM à suivre la capacité WM, en utilisant les données collectées à partir du jeu de mémoire visuelle Match²s que nous avons conçu et implémenté ;*

- *un nouveau cadre fondé sur AUHWM pour l'adaptation automatique des tâches de l'interface utilisateur, en utilisant les paramètres WM suivis comme estimations pour les performances futures, et son évaluation ;*

- *des idées et indications novatrices pour le développement futur des technologies fondées sur AUHWM.*

1.3 Organisation

Ce manuscrit est organisé en 9 chapitres, le chapitre 2 servant d'introduction générale aux principales préoccupations de ce manuscrit.

Le chapitre 3 présente dans sa première section les principes fondamentaux de la WM humaine et de la théorie de la charge cognitive, servant de base au lecteur non familier avec ces concepts ainsi qu'un survol des principales notions abordées dans le reste du document. Dans une deuxième section, ce chapitre décrit les avantages qu'apporte l'adaptation des systèmes informatiques aux limitations de WM, présentant la principale motivation derrière ce travail.

La première section du chapitre 4 présente un aperçu général des différentes méthodes trouvées dans la littérature pour adapter les systèmes informatiques aux

limitations cognitives. La deuxième partie du chapitre 4 est consacrée à la présentation, en profondeur, de deux modèles de calcul de WM qui permettent de simuler l'évolution des informations stockées dans la WM.

Le chapitre 5 présente et décrit MATCHS, qui est notre premier cadre pour la modélisation de la capacité de la WM des utilisateurs ainsi que la stratégie d'adaptation correspondante. Plus loin dans le même chapitre, le lecteur est introduit à notre jeu de mémoire visuelle Match²s, qui est utilisé pour la validation expérimentale de MATCHS, et plus tard également pour AUHWM. Le chapitre se termine par la présentation des résultats obtenus et une discussion sur les performances et les limites du cadre.

Dans le chapitre 6, le lecteur est introduit à AUHWM, une extension des idées de base de MATCHS permettant de modéliser la capacité de la WM de l'utilisateur en temps réel. AUHWM est un cadre pour le suivi de la capacité cognitive de l'utilisateur et utilise un processus de filtrage de Kalman non linéaire comme l'un de ses principaux composants. Étant donné que ce type de filtrage peut ne pas être familier à tous les lecteurs, ce chapitre présente également la base théorique du filtre de Kalman. Le chapitre se termine par une validation expérimentale et une discussion sur les performances de modélisation AUHWM en utilisant les données collectées avec le jeu Match ²s.

Le chapitre 7 va au-delà des capacités de modélisation d'AUHWM et présente un cadre qui, en utilisant AUHWM, est capable d'adapter une interface utilisateur donnée à la capacité cognitive suivie d'un utilisateur. Ce chapitre présente le cadre au lecteur et discute de ses performances en utilisant, encore une fois, les données Match ²s.

Le chapitre 8 est consacré aux conclusions, recommandations et perspectives pour les travaux actuels et futurs.

1.4 Publications

Les travaux discutés dans cette thèse ont donné lieu aux publications suivantes:

- Sguerra, B., Jouvelot, P., and Benveniste, S. Oblivion Tracking: Towards a Probabilistic Working Memory Model for the Adaptation of Systems to Alzheimer Patients. 25th User Modeling, Adaptation and Personnalization Conference Adjunct, Bratislava, Jul. 2017 [4] ;

- Sguerra, B., Benamara, A., Benveniste, S., and Jouvelot, P. Adaptive Human-Computer Interfaces to Working Memory Limitations Using MATCHS. IEEE International Conference on Systems, Man, and Cybernetics (SMC) , Miyazaki, Oct. 2018 [5] ;

- Sguerra, B., and Jouvelot, P. “An Unscented Hound for Working Memory” and the Cognitive Adaptation of User Interfaces. ACM User Modeling, Adaptation and Personnalization Conference (UMAP), Larnaca, Jun. 2019 [6].

Chapter 2

Introduction

"In general, we're least aware of what our minds do best."

Marvin Minsky

In 1967, the American writer John M. Culkin said "We shape our tools and, thereafter, our tools shape us." In 2019, 52 years later, this couldn't be more true. We are currently living a technological revolution where technology advances at an exponential pace. It has never been faster, or easier, to come up with new tools and to put them in the hands of users. These new technological tools are reshaping information channels and human connections, while reshaping jobs, leisure, and consequently the meaning of being human altogether. An overflow of new gadgets, smartphones, smartwatches, applications, chatbots and others, washes over consumers on a daily basis. This tide of new tools eventually calls forth for human adaptation, for a constant struggle to adapt in order not to become obsolete and maintain a place in society. Since never before have so many new tools been created at such a speed; never before has human life changed so quickly.

Most of this revolutionary technological development is blindsided by the goal of being more precise, faster and more efficient. There are a number of reasons that make these metrics worth of being pursued. Still, developers tend to forget to put the human factor in the targeted loss function. However resilient human beings are, being able to adapt to harsh environments and to learn how to deal with complex interfaces, there are intrinsic limitations that are simply unavoidable, limitations such as the human brain speed of processing, available attentional energy, sleep patterns, memory capacity, learning speed and many others. These can be seen as the brain's hardware limitations.

Technologies intended for humans should be designed around humans, which means that one has to consider the functioning of the human mind when designing

a new tool, therefore considering and accounting for human limitations. This is a way of ensuring humans can better adapt to and accept a new piece of technology, a way of smoothing out the learning curve and ultimately rendering tools more efficient. In his 2009 book "Sum: Forty Tales from the Afterlives", the American neuroscientist David Eagleman writes about a fictional race being plagued by questions such as "why are we here?" or "what is the purpose of existence?" to the point where they decided to invest for generations in the development of a supercomputing machine devoted to finding such answers. However, the project failed miserably, for the resulting machine was way too advanced to interact with these beings. When developing a machine more complex and intelligent than you, your ability to understand the machine starts to slip away. The most advanced piece of technology can be underused if the user fails to properly interact with it, for "a tool is only as good as its user", as is often said.

The human mind limitations stand for the breaking point at which technologies are unable to stretch humanity any further. Designing interfaces that flash the maximum amount of information that can be fit on a screen at the highest speed possible is not desirable. And in fact, these are not metrics pursued when developing a new tool. Instinctively, designers, being humans themselves, know that there are cognitive limitations to the amount of information that can be processed by users. Most good interfaces keep it simple; a minimalistic amount of buttons and options displayed helps users focus attention and decide which command they ought to use. However, displaying the minimum amount of information is also not always desirable. For instance, the "minimum" amount of information is subjective. Depending on the user's mastery over the subject concerning the displayed data, he or she can have an easier or harder time pondering over different amounts of data. For example, the game of Go consists of two players putting white and black stones on a board. The standard Go board has a 19×19 grid, and at every intersection a stone can be placed. Since there are 3 possible states for every intersection (empty, black or white), the number of possible configurations of the board is $3^{361} \approx 1.7 \times 10^{172}$ (not every configuration is possible due to the rules of the game; John Tromp determined that there are about 2.08×10^{170} legal configurations [2]). This incredible amount of possible configurations is one of the reasons machines had a hard time beating Go players until Google's AlphaGo showed up in 2016. However, advanced Go players can play a variant of the game called one-color Go, where players use stones of the same color, meaning that the players have to remember which stone is which by remembering the progression of the game as a whole. This impressive feat can appear impossible for a beginner, but it just goes to show how the level of mastery affects the perception of complex information.

Finding the right balance between how demanding and complex a interface is

and how the user perceives such complexity may be the key to extend human and technology performance to the limit. This call for the continuous adaptation of the communication channels between humans and machines. By integrating interfaces with a deep knowledge of how the human mind functions, technology could compensate human limitations, facilitating learning and increasing accessibility. However, not only serving as a crutch for the mind, but by a steady adaptation of systems while accounting for human learning, a new technology, once assimilated, could continuously push human performance by compensating limitations where they appear, yet challenging areas where the user shows proficiency. This means that our tools interfaces should be shaped by the function of the mind. Designing of new tools and their interfaces ought to be centered around these limits and with a deep knowledge of how we "work", as human beings. This could be a key factor in this data-driven era, enabling new technologies to change us, however, not by crashing against users but by guiding them.

2.1 Goal

The goal of this work is to explore the possible ways of adapting computer systems to the way humans process information. It is specially concerned with the adaptation of user interfaces (UI) to the limits of the human Working Memory (WM). WM is the part of human cognition responsible for the storing and processing of verbal and visual information. It has been recognized as a major bottleneck in human processing capability. Therefore, in more precise terms, the goal of this thesis is to ponder over the possibilities of rendering computer systems aware and capable of compensating users' WM limitations.

One can then define four steps in order to perform the appropriate adaptation (based on the methodology of [3]):

1. infer the user's capacity limitation;
2. identify the potential impact on performance;
3. select a compensatory strategy;
4. implement this strategy in the terms of the current context.

These steps, specially steps 1 and 2, are the main challenges addressed in this work. Since cognitive limitations are not easily measurable (other than by performing specific and very constrained tests), one needs to be able to infer these limitations from the user's interaction with a system. The assessment of the user's state (whether cognitive, physical, personality-specific or affective) is one of the core

functionalities that must be addressed by effective user modeling and adaptation systems [3].

In this context, adaptation requires the system to have inherent knowledge of how a limited capacity impacts performance, and how an action could compensate such a limitation. In order to do so, a system needs to go beyond the observational knowledge linking some inputted parameters to the observed output that traditional Machine-Learning (ML) techniques might provide. Supervised ML-based technologies lack flexibility, they are limited by the amount of data in the training set. Logically, providing the system with a comprehensive picture of all the possible different scenarios it might encounter, would compensate for this rigidness. However it would call for a huge amount of data. The use of a model based on understanding allows one to go beyond previous collected data and deal with cases not yet encountered.

Such a system needs a model of the underlying causal processes involved in the limited capacities impacting performance. Models, of various levels of abstraction, stand for simplifications of a system that serve as a limited explanation of some of the many mechanisms it is composed of and that work together. Models can differ in level of complexity and reliability, yet they represent the understanding one has about a phenomenon; it can come in the form of a theory or, in some cases, a law. Having a causal model of how capacity limitations impact performance allows the prediction of how the inferred capacity affects performance as well as the selection of adequate compensatory measures.

The main objective of this work is, therefore, to explore efficient methods to embed a computer system with a model of human WM in order to enable UI adaptation to users' WM limitations.

2.2 Contributions

This work includes five major contributions:

- Memory Adaptation Through Cognitive Handling Simulation (MATCHS), a new framework for the dynamic modeling, tracking and adaption of tasks, whose performance are WM-dependent, to the perceived WM limitations;
- An Unscented Hound for Working Memory (AUHWM), an extension of MATCHS, expanding its capabilities so that to enable real-time modeling and tracking human WM performance, which uses deterministic model of working memory and Unscented Kalman filtering;
- an experimental evaluation of MATCHS' and AUHWM's ability to track WM capacity, using data collected from the visual memory game Match^{2}s that we designed and implemented;

- a new AUHWM-based framework for automatic UI task adaptation, using tracked WM parameters as estimates for future performance, and its evaluation;

- ideas and indications for the future development of AUHWM-based technologies

2.3 Organization

This manuscript is organized in 9 chapters, with Chapter 2 serving as a general introduction to the main concerns of this manuscript.

Chapter 3 presents in its first section the fundamentals of human WM and Cognitive Load Theory, serving as a basis for the reader not familiar with these concepts as well as a highlight of the main notions covered in the rest of the document. In a second section, Chapter 3 describes the main interests behind making computer systems aware to WM limitations, introducing the main motivation behind this work.

The first section of Chapter 4 presents a general survey of the different methods found in the literature for adapting computer systems to cognitive limitations. The second part of Chapter 4 is concerned with the presentation, in depth, of two computational models of WM that allow one to simulate the evolution of the information stored in WM.

Chapter 5 introduces and describe MATCHS, which is our first framework for the modeling of users' WM capacity as well as the corresponding adaptation strategy. Later in the same chapter, the reader is introduced to our visual WM game Match^{2}s; which is used for the experimental validation of MATCHS, and later on also for AUHWM. The chapter ends with the presentation of the obtained results and a discussion about the framework performance and limitations.

In Chapter 6, the reader is introduced to AUHWM, an extension of MATCHS core ideas allowing one to model the user's WM capacity in real time. AUHWM is a framework for tracking the user's cognitive capacity and employs an Unscented Kalman Filtering process as one of its core components. Since this type of filtering may not be familiar to all readers, this chapter also presents the theoretical basis of the Kalman Filter. The chapter ends with an experimental validation and a discussion about AUHWM modeling performance using the data collected with the game Match^{2}s.

Chapter 7 goes beyond AUHWM modeling capabilities and present a framework that, by taking advantage of AUHWM, is capable of adapting a given UI to a user's tracked cognitive capacity. This chapter introduces the framework to the reader and discusses its performance using, again, the Match^{2}s data.

Lastly, Chapter 8 is dedicated to the conclusions, recommendations and perspectives for current and future work.

2.4 Published Work

The work discussed in this thesis resulted in the following publications:

- Sguerra, B., Jouvelot, P., and Benveniste, S. Oblivion Tracking: Towards a Probabilistic Working Memory Model for the Adaptation of Systems to Alzheimer Patients. 25th User Modeling, Adaptation and Personnalization Conference Adjunct, Bratislava, Jul. 2017 [4];
- Sguerra, B., Benamara, A., Benveniste, S., and Jouvelot, P. Adaptive Human-Computer Interfaces to Working Memory Limitations Using MATCHS. IEEE International Conference on Systems, Man, and Cybernetics (SMC) , Miyazaki, Oct. 2018 [5];
- Sguerra, B., and Jouvelot, P. "An Unscented Hound for Working Memory" and the Cognitive Adaptation of User Interfaces. ACM User Modeling, Adaptation and Personnalization Conference (UMAP), Larnaca, Jun. 2019 [6].

Chapter 3

Motivation

Ce chapitre sert de base au lecteur non initié, fournissant les principes fondamentaux de la mémoire de travail (WM) humaine ainsi que la théorie de la charge cognitive associée. La section 3.1 présente la vue théorique de la WM, soulignant également certains des éléments clés qui font de la WM un goulot d'étranglement dans la capacité humaine de traitement d'information. La section 3.2 s'intéresse à la présentation de la théorie de la charge cognitive, qui introduit ce concept de charge cognitive et la manière dont elle est liée à la WM et à ses limites. Le chapitre se termine dans la section 3.3, qui présente trois domaines de systèmes automatisés qui pourraient bénéficier d'une prise de conscience immédiate des ressources WM de l'utilisateur, servant ainsi de motivation à ce travail.

This chapter serves as a basis for the non-initiated reader, providing the fundamentals of human Working Memory (WM) as well as the related Cognitive Load Theory. Section 3.1 presents the theoretical view of WM, also highlighting some of the key elements that make WM a bottleneck in human processing capacity. Section 3.2 is concerned with presenting Cognitive Load Theory, which elaborates upon the concept of cognitive load and how it is linked to WM and its limits. The chapter ends in Section 3.3, which presents three areas of automatized systems that could profit from ready awareness to user's WM resources, therefore serving as the motivation of this work.

3.1 Working Memory

When considering human cognitive limitations, WM is very frequently highlighted. Cognitive psychology theorizes WM as the underlying mechanism responsible for the maintenance of task-related information during the performance of cognitive tasks [7]. At the most fundamental level, WM has been considered the most

significant achievement of human mental evolution [7]. Moreover, WM limitations have been recognized, since the onset of cognitive research [8], as a major bottleneck in human information processing.

WM is the part of the human cognition responsible for conscious short-term storage and recall of information. It is essential to all of us to successfully perform complex cognitive functions, from having conversations, driving in heavy traffic, cooking, calculating change, reading to general problem solving. Acquisition of new information, such as a list-learning task, is also an example of a complex cognitive activity that requires the resources of WM [9]. Consequently, WM deficits have been linked to children with learning complications [10] and difficulties in language acquisition [11] and many other disorders and disabilities.

The term "working memory" was chosen to emphasize the functional role of the system, rather than simply its storage capacity (however, in the body of research concerned with WM, there is not always a clear distinction between WM and short-term memory [7]).

There are a number of metaphors that try to explain WM, such as the "box" metaphor, the "workspace" or "blackboard" metaphor, the "mental energy" or "resources" metaphor, and the "juggling" metaphor [7]. If we view human memory as a huge and dark cavernous library, WM would be the reading light on the desk. It is the workplace were information is pondered; it is where information is considered in the mind.

The body of research in cognitive psychology and neuroscience regarding WM modeling is vast, including for instance works by Atkinson and Shiffrin [12], Baddeley [13], Just and Carpenter [14] and Anderson [15], among others. This WM literature introduces many, and somewhat conflicting, proposals regarding the nature of WM, its functioning and its role in the accomplishment of tasks [7]. The most widely accepted model of human WM was proposed by Baddeley and Hitch [13], outlined below.

3.1.1 The Homunculus

Baddley and Hitch's first proposed model is a multi-component model (see 3.1) that includes two slave information-holding systems: the phonological loop, for verbal information, and the visuospatial sketchpad, for visual and spatial information. A third component, called Central Executive, is described as the most complex part of the WM. All three of these model's components have limited capacity, although the nature of such limitations differ.

- **Phonological loop** The phonological loop is assumed capable of storing speech-based and possibly purely acoustic information in a temporary store.

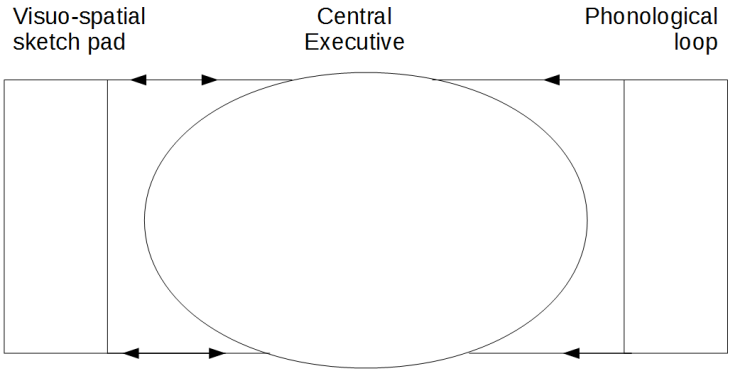


Figure 3.1: Original Baddley and Hitch's WM model.

It comprises not only the phonological storage system, but also a rehearsal mechanism.

- **Visual sketchpad** The visual sketchpad performs a similar function as the phonological loop, however for both visual and spatial information storage.

- **Central executive** The central executive acts as an attentional controller that (1) processes information, (2) focuses, switches and divides attention and (3) links with long-term memory (LTM). Baddeley describes this component as virtually a homunculus, a "little man in the head, capable of doing all the clever things that were outside the competence of the two subsystems"[16].

In this initial model, Baddley and Hitch concentrated on the two first subsystems since they offer a more trackable challenge, leaving the precise nature of the central executive unspecified. Later on, Baddley proposed a revised version of this model [17] and introduces an additional subsystem, the Episodic Buffer. This subsystem is a more general integrated storage system that provides additional memory to manage. It is a buffer that holds episodes, or "chunks", of multidimensional code [16]. Baddley suggests the buffer's capacity to be limited to 4 chunks, agreeing there with Cowan [18]. This system forms an interface between the three working memory subsystems and long-term memory. It serves as a binding mechanism that allows perceptual information and information from other subsystems and from long-term memory to be integrated into a limited number of episodes [19] (hence the "multidimensional code"). Another significant difference in this model is the addition of a connection between a series of "fluid" systems, which require only temporary activation, and long-term memory, representing more permanent crystallized skills and knowledge.

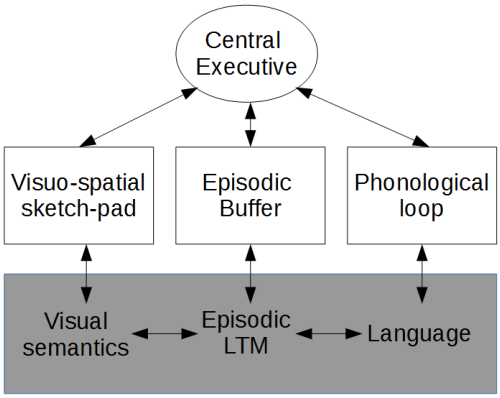


Figure 3.2: Updated Baddley and Hitch's WM model.

3.1.2 Capacity

The capacity of WM is known to be extremely limited and is one of the strongest factors that impact individual differences in cognitive abilities [20], as WM capacity measures can be (and have been) used to predict performance in a series of complex daily cognitive tasks.

There are a number of models of WM capacity. Cowan [18] describes WM as having a limited number of slots (or chunks) where information is stored and, if there is more information than available slots, it will be lost. At first Miller [21] estimated the human capacity as being about 7 ± 2 items at a time, while Cowan [18] estimated that value to be about 4 items. This apparent contradiction could be explained by the nature of the information. When simple information is presented, the limits of WM could go up to around seven items activated in the same time; however, when presented with complex information, the limit tends to be much lower. Therefore one can theorize that WM limitation is closely related to the notion of limited mental resources. Such limited resources would be a commodity that could be distributed between information in order to attain fixation.

There are mostly two classes of WM: item-based and resource-based models [20].

- Item-based models ignore the complex structure of items and their respective parameters and treat each one as a "unit of memory". In this class, the same amount of memory space is allocated to every item, and partial storage does not occur.

- Resource-based models consider the number of properties, called resources or features, of an item. In fact, properties such as the number of parameters

are primary to such models. Complex items may take on more memory space than simple items, and partial storage may occur.

.

Following an item-based approach, Pashler [22] suggested a probabilistic way to estimate a person's WM capacity. In an experiment where n items are presented to the subject, if the person's WM capacity k is such that $k \geq n$, then the person will remember all the items, and the probability D that one item remains stored into the WM is 1. However, if $k < n$, Pashler assumes a uniform distribution, and $D = \frac{k}{n}$. As a whole, the subject stores an item in his WM with probability $D = \min(\frac{k}{n}, 1)$. This is the most common method of estimating working memory capacity and the basis of tests that experimentally assess users' memory span (the number of information one can store in their WM) such as the Corsi test [23], sequence learning tests and many others.

Of course, the behavior of WM varies along various factors. Besides capacity, time is obviously an important factor; the data present in the WM storage areas [24] can be recovered during a finite amount of time with little loss in either quantity or quality [25]. But, after this time, the probability that an item remains in memory declines.

3.1.3 Attention

Another key ingredient is attention [26]. One definition of attention is the selecting of a stimulus in detriment of others [27]. In Baddley and Hitch's initial model, the central executive, the most complex of its components, was responsible for managing and allocating attention between the two slave systems. Therefore, it should not be surprising that the concepts of attention and WM are widely acknowledged as being related [27]. For instance, in Anderson's ACT-R theory [28], attention is seen as a limited resource that "activates" information in declarative memory; this activated information can be seen as an abstraction of WM (the reader is invited to read Section 4.2.1 for a detailed discussion). This close relationship between these two concepts means that attention and WM are known to interact during manipulation and encoding of information, often overlapping each other [27]. If one were to sustain the meaning of attention as the selection of some stimuli in detriment of others, this filtering view of attention clearly separates the concepts of WM and attention. However, if one is to consider the existence of attentional resources, such as in ACT-R, then the division between these two concepts becomes rather blurry [7].

It is not the goal of this work to debate if attention and WM are the same construct or not. What is important here is, for the reader, to be aware that attention is a key factor that can affect WM. Fluctuations in attention can easily

influence WM performance; for instance, if one directs the subject's attention away from the stimuli at the time of their presentation, one clearly observes the impact of such an action on the WM capacity [18]. Therefore, attention could be seen as a key factor in determining the success of encoding information. However, some theories sustain that attention actually plays a role in the post-processing of the perceived stimuli [27]. Although the role of attention during encoding of information may vary, it can be argued that it serves as the gateway for information storage in WM.

3.1.4 Motivation

Motivation is a well known and documented influential factor in WM performance outcome. For instance, in [29], motivation was found to be the biggest cause of daily intrapersonal WM performance variation in a study performed throughout 100 days. Pochon et al. in [30] used fMRI techniques to study the brain activation areas while subjects performed WM tasks with different levels of complexity and monetary reward. They found a correlation between the increase in monetary reward and an increased activation in the brain areas related to WM. Making motivation another key factor regulating WM performance.

3.2 Cognitive Load

Cognitive Load Theory (CLT) [31] is concerned with how mental resources are allocated to (or focused on) different tasks. CLT posits that a person has a finite amount of available cognitive resources, and that different tasks demand different cognitive loads in order to be accomplished. Cognitive load is then considered as a metrics used in the modeling and prediction of human cognition-related performance on different sets of intellectual tasks [32].

3.2.1 Cognitive Load Theory

Back in the early 90's, Sweller used advances in cognitive sciences to explain differences in students' performance on tasks linked to learning and problem solving. Students whose mental resources were burdened with external activities such as problem-solving tasks had less attention to focus on tasks important to learning, such as schema acquisition. The cognitive load of handling information referring to the problem-solving part of the task, such as the relationship between problem-solving operators, could be so demanding that it resulted in little cognitive resources left for learning. This could explain why students could perform

well when solving different problems and yet remain oblivious to the problem's essential structure [31].

CLT was also used to derive techniques for designing how instructions ought to be presented to students, by considering the limited human processing capacity [33]. This can be attained by considering two kinds of observed cognitive loads: extraneous and intrinsic. Extraneous loads are under complete control of the instructors, as they relate to how information is presented and what are the activities students are required to do. Intrinsic load refers to the structure and complexity of new information. In order to adapt instructions to the intrinsic load of a given knowledge, one ought to consider each student's level of mastery when presenting new information. The more familiar a student is with the subject, the easier it is for the student to encode new information related to already acquired knowledge. For example, someone who is proficient in English has little trouble learning a new sentence. However, the same task, when presented to someone who is learning how to read and has to focus on each individual letter at a time, becomes dramatically more difficult.

The notion of cognitive load is closely related to that of WM. CLT is based on several assumptions, one of which is the existence of a limited-processing capacity in human cognition, which closely relates to the limitations of WM. The total cognitive load (extraneous plus intrinsic) of a given task is applied on the storing subsystems that compose WM. If the total cognitive charge corresponds to the student's WM capacity, then the student will find herself under a high cognitive load or even information overload. The difference between a user's WM capacity and the cognitive load of a given task equals the amount of attentional resources that can (we say "can", for those resources are not necessarily used, since learning involves effort and thus depends on the student's motivation) be used for learning [34], defining a third and last form of cognitive load, the germane or effective cognitive load [35]. This load, unlike the others, enhance learning, as it corresponds to the cognitive resources left that are devoted to schema acquisition and automation.

3.2.2 Assessing Cognitive Load

In most of the CLT literature, cognitive load is regarded as a theoretical construct describing an internal processing of information that cannot be observed. However, if one is to use CLT to drive the design of some instructional material (be it computational or not), one has to assess some measure of the imposed cognitive load. In [34], the various presented methods for assessing cognitive load are classified along two dimensions: objectivity and causal relation. The first dimension, objectivity, refers to the method of measurement being based on objective observations such as actual performance or physiological data, or being based on

subjective self-reported data, such as questionnaires. The second dimension, causal relation, corresponds to the link (direct or indirect) between the observation and the attribute of interest. For instance, questionnaires about self-reported difficulty during a task are subjective in the first dimension and direct in the causal relation dimension, for difficulty is directed related to the imposed cognitive charge. Measuring task performance outcome is a indirect-objective method; indeed, while performance is objective, measure-wise, it is an indirect implication of the cognitive load, since performance deterioration can be related to a less than optimal task-related learning process that implies a high cognitive load.

The dual-task paradigm is a direct-objective method for assessing cognitive load. The dual-task paradigm was primarily employed in WM research, aimed at examining Baddley's central executive system [36]. Recall that Baddley's central executive is responsible for attentional control, focusing and switching attention during the performance of a task. The central executive is thought to have limited capacity, therefore, when performing two tasks in close sequence, the performance of the second task is impacted by the load imposed by the first one, at least as long as both tasks require the same type of mental resources. The dual-task paradigm, therefore, consists of a subject performing two tasks in sequence. The variable load applied by an initial task results in more or less available resources for consecutive tasks and, in consequence, different performances arise.

Brünken et al. [34] present the dual-task paradigm as a interesting way of assessing cognitive load during Multimedia Learning. In Multimedia Learning, the extraneous load imposed by the way information is displayed can be measured by having the learners performing secondary tasks; their difference in performance should reflect how taxing the extraneous load was. They show the effectiveness of using secondary tasks, such as a visual-monitoring task, to assess the total cognitive load, as the effects of cognitive overload degrading performance were found in every single participant of their experiments.

3.3 Making Computer Systems Aware of WM Limitations

In the previous sections, the concepts of WM and cognitive load were presented. It should be clear by now that WM limitations represent a major bottleneck for human information processing. Therefore, taking WM limitations in consideration when designing user interfaces (UI) should be nothing less than logical. UIs are the communication channel through which humans and machines communicate. They are everywhere: ATMs, airports, fast food chains, museums, etc. Whenever information is being conveyed through some automated channel, an user interface

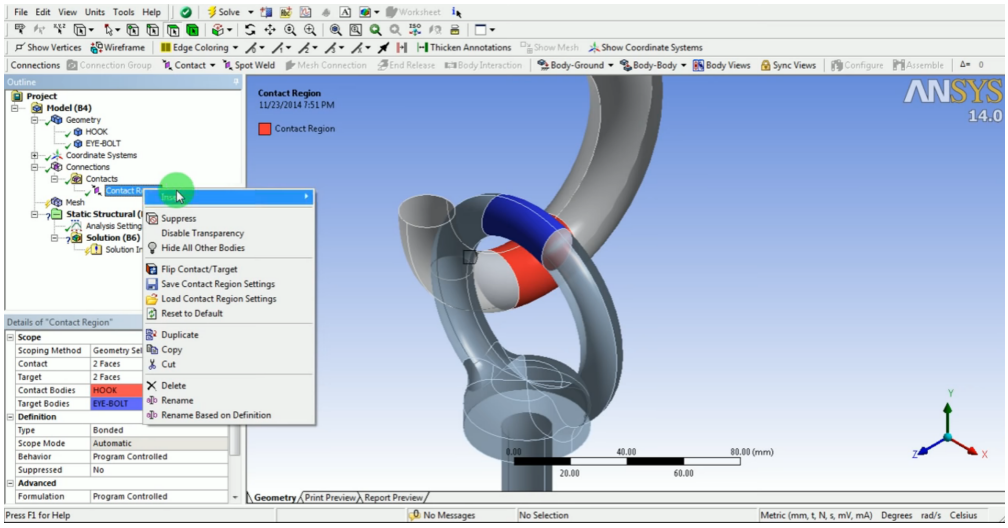


Figure 3.3: Screenshot [1] of Ansys, a software for engineering simulation, showing the great amount of possible commands an user has to consider when modeling a mechanical structure.

arises. The next three subsections highlight the interest, regarding three different areas, to provide computer systems with awareness to the user's WM limitations and the ability to adapt their UIs to compensate such limitations.

3.3.1 Tutoring Systems

Virtually any new software a user wants to be proficient on has a learning curve. Figure 3.3 depicts a screenshot from the Ansys software for engineering simulation. Any novice learner should feel threatened by the amount of windows and command options, which (in some cases) may cause discouragement. A software such as Ansys, which enables the user to perform diverse numeric calculations and model the effects of heat and strength in complex mechanical models by the pressing of a single button, is a technological marvel. Humans have come a long way in developing these software packages in order to serve as tools for building faster and more reliable technologies, and this is a grand accomplishment. However, all these options and commands create an extraneous cognitive information overcharge that can and should be dealt with.

Paper and pencil are tools very much needed to compensate WM limitations. We teach children to add numbers through columnar addition, by writing down carry-overs. Having a symbolic representation on the paper in front of learners removes some of the load in WM. Young children need to consider each number individually when performing addition, meaning the intrinsic load of storing a

double-digit number in WM is pretty elevated. As the level of mastery over number manipulation increases, most adults are able to solve simple additions without external aid. There is no reason why computer systems shouldn't also consider this bottleneck of information. Moreover, making UI sensitive to cognitive loads, both extraneous and intrinsic, would personalize the learning rate in these environments by adjusting the degree of complexity to both the user's individual WM limitations and level of mastery, resulting in more resources dedicated for the germane cognitive load and therefore, a smoother learning curve.

For instance, a framework capable of assessing a user's cognitive load when dealing with a computer application could be used to perform scaffolding in intelligent tutoring systems [37] and adapt the way information is conveyed. Instructional scaffolding are a series of technical method where support is given to students when dealing with new concepts, relieving some of the corresponding cognitive burden. The user's performance, being measured by reaction time or accuracy, could serve as an indirect measure of the cognitive load imposed by the system, but also of the user's mastery of the knowledge being tutored. Following CLT, a new user whose performance is significantly better than the average might be familiar with the concepts being introduced or have a high WM capacity. Therefore, the tutoring system could increase the complexity of the information presented. Conversely, a novice learner, who is struggling with the different concepts, would be presented with a simpler interface, problems or even external support, ensuring the learner has enough available cognitive resources left for the germane load related to learning. Consequently, complex applications such as Photoshop or CAD software, which are known to be complex for novice users, could be personalized by increasing the complexity and amount of information presented according to the user's competence and/or cognitive capacity.

3.3.2 Assistive Technologies

Not only in tutoring systems are limitations in WM a key factor. There is evidence that WM is particularly impaired in people suffering from Alzheimer's disease (AD) [38], with the progression of the disease resulting in a smaller WM capacity. AD is the most common form of dementia, accounting for 50% to 70% of cases (it is estimated that, by 2050, there will be over 100 million people living with AD [39]). Dementiae are chronic neurodegenerative diseases that usually start slowly and worsen with time. They may cause problems with language, disorientation, mood swings, loss of motivation and behavioral issues. As a patient's condition declines, managing common daily tasks becomes increasingly difficult. In the beginning, only the more complex Instrumental Activities of Daily Living (IADL) [40], such as cooking, managing household finances and shopping, are affected. Then, as the disease progresses, core Activities of Daily Living [41] (ADL), such as washing

hands or toileting, are impaired too, placing an even greater burden on caregivers, subjecting them to eventual physical and mental disorders [39].

Due to this dire scenario, Assistive Technologies for dementia represent a very important and rapidly developing field of research: assistive devices can facilitate independent living for patients and reduce caregiver burden, enhancing quality of life for both and helping to restrain care costs that are threatening economies, especially in the most developed countries.

Since AD is a progressive disease in which the symptoms usually worsen over time, a system designed for patient care must be able to adapt to the user's current state. For instance, as the disease progresses, patients may have trouble communicating or even understanding lengthy requests [42]. Therefore, caregivers are required to speak slower or use short sentences with simple words. However, if one designs an assistive system that communicates through verbal cues, a one-for-all solution that uses only slow and short sentences might be seen as boring and unattractive to patients with higher cognitive capabilities. Depending on the progress of the disease, some patients may also need some extra time to process uttered questions before answering a request [42]. Therefore, if an assistive system does not consider the patient-dependent response time and keeps asking the same query over and over again, assuming that the person just did not understand, then any kind of useful interaction with the patient becomes impossible.

Making assistive device technologies sensitive to the patient's cognitive capacities may be a way to render these systems more receptive for people suffering from dementia. Being able to automatically adapt a cognitively demanding task to the available WM capacity is a way to help ensure its accomplishment. Moreover, adjusting UIs to the user's cognitive capacities could be a way to render computer interfaces more accessible to the elderly population suffering from dementia-like diseases. This would facilitate IADLs for patients with up to moderate dementia, restoring some of the patient's capability to live independently as well as relieving the burden of caregivers.

Most importantly here, a system providing this kind of automatic adaptation could be integrated in various assistive technologies to autonomously and continuously tailor the user interfaces of assistive devices, representing a significant contribution in the area of assistive technologies as it has the potential to be of great benefit to individuals suffering from memory deficits. It could also provide data corresponding to the user's evolving cognitive capacities. Beyond its clear relevance in the design of simpler user interfaces for computer-assisted daily-life activities, such information could be also used by caregivers as signals suggesting the possibly setting-in of a neurodegenerative disease.

3.3.3 Systems Sensitive to WM Limitations

One could consider any system where information is to be considered and decisions are to be made to be sensitive to cognitive limitations, as human available cognitive resources might fluctuate even during the time span of a single interaction with a system. In particular, WM becomes specially impaired in stressful situations [43]. Stress and anxiety have been proven to influence performance in processing speed, to reduce WM capacity and selective attention [44] and to be responsible for the deterioration of the quality of decision making. During stressful situations, where individuals are asked to make high-stakes decisions, WM is a crucial element.

There are numerous cases where, in stressful situations, human error was responsible for major accidents. One well-known and documented example is the 1988 railway accident at Gare de Lyon in Paris. What is considered the worst railway accident of the Paris region is due to human factor errors. A train conductor, coming from the French city of Melun, was under pressure to arrive in time at his destination. Due to the stressful situation, the conductor made a series of mistakes that resulted on his train hitting another one that was waiting for departure at Gare de Lyon in Paris, killing 56 people and injuring 57. The stress of the situation made the conductor oblivious to his knowledge of how to stop the train, for instance by forgetting and ignoring the electric brakes.

Another well-known accident is the Air France Flight 447 from Rio de Janeiro to Paris [45]. What started with a trivial aviation problem – frozen Pitot tubes failing to give the proper speed of the aircraft – resulted in 228 people dying. The conversation between the pilots recovered from the airplane's black box shows the series of mistakes and overlooks they made, resulting in the fatal crash. In his book "Smarter Faster Better: The Secrets of Being Productive in Life and Business", Charles Duhigg uses the Air France Flight 447 as an example to discuss what is called cognitive funneling and the importance of mental models. Cognitive funneling, or attentional funneling, stands for a state where attention is focused on one stimulus for more than the optimal time, neglecting other sources of information [46]. The pilots of Flight 447, due to the stress of the situation, focused on what source of information was right in front of them, and where incapable of taking a step back to analyze the situation as a whole.

Accidents such as the ones described above could be prevented by accessing the stress levels of the conductors and simplifying the user interface. The cockpit of the Airbus A330 of Flight 447 was incredibly sophisticated; it had very few screens in order to prevent distractions. However, even in such a minimalist context, attention funneling reduced the pilots capacities for reasoning and processing information to the point where they stood oblivious to what was happening even as the waves of the Atlantic Ocean were rapidly approaching the cockpit window.

Stressful situations and high-stakes decisions are made everyday, not only by

airplane pilots and train conductors, but also by a multitude of drivers. In these situations, an intelligent system, capable of assessing the pilot's stress level or inferring his available cognitive capacity, could add some flexibility to the pilot's actions, eventually taking control over if the pilot is deemed incapable of making reliable decisions (in panic situations, for example). Moreover, knowing how taxing in cognitive capacity a task is would allow the system's UI, in stressful situations, to be simplified, providing the user with only the absolute necessary information to the task at hand. One could think of simplifying the verbal instructions given by GPS systems or selecting which information to display.

Chapter 4

Adaptation and Modeling

Ce chapitre est consacré, dans une première partie (Section 4.1), à une étude des différentes méthodes d'évaluation de la charge de travail cognitive et d'adaptation des systèmes informatiques à celle-ci. La deuxième partie concerne la modélisation informatique de la WM humaine, les sections 4.2.1 et 4.2.2 présentant, en profondeur, deux modèles informatiques différents, à base mathématique, de la WM humaine.

This chapter is dedicated in a first part (Section 4.1) to a survey of the different methods for assessing cognitive workload and adapting computer systems to it. The second part is concerned with the computational modeling of human WM, with Sections 4.2.1 and 4.2.2 presenting, in depth, two different computational models, with a mathematical basis, of human WM.

4.1 Adapting HCI to Cognitive Limitations

A great body of research is concerned with the development of computer systems around human cognitive limitations. Designing computer interfaces around user models gained much attention since the late 70's due to the greater diffusion of computer systems [47]. The area of human-centered design technologies has a multitude of different goals, yet they are based on the same principles: designing technologies while considering the way the human mind functions and its limitations. Among the different solutions encountered, most of them are based on theoretical models, i.e., technologies employing models of (or part of) the human behavior used to drive the functioning of the system. However with the recent "AI renaissance" [48] and its sister discipline of ML, more and more data-based adaptation technologies are being developed by academics and companies around the world.

Different goals justify the inclusion of human models in computational systems. One could consider, for instance, studies such as [49], which cover the **optimization of human learning**, the focus of many researchers and companies that have developed adaptive educational methods for tutoring systems. Corbett and Anderson describe in [49] a Bayesian approach to infer student learning called Bayesian Knowledge tracing (BKT); this is still one of the most popular approaches to access and predict student performance. BKT is based on Anderson's cognitive architecture ACT-R [28] (more on ACT-R later in this chapter) concerning skill-knowledge acquisition and has sprung different solutions for tutoring systems, such as [50] where a dynamic Bayesian network is employed to improve BKT's student model by adding a layer of skill topology, or [51], where once again a cognitive tutor based on Anderson's ACT-R and BKT is used to help students employ self-explanation as a metacognitive strategy for learning.

BKT methods develop models of student learning in order to personalize tutoring systems. Students are presented the material for skill acquisition hierarchically and have their proficiency assessed through performance measurements on problems. The work developed in this document, however, is concerned with more general methods of adapting interfaces and tasks to cognitive limitations, without prior total modeling of the task itself.

Closer to our problematic is the **adaptation of computer systems to human limited cognitive resources**. This research area is mainly focused on CLT and **assessing** rather than **adapting to** cognitive load, as the literature about how systems can react to the assessed cognitive load is by far scarcier than the literature on methods for inferring it [52].

However, in order to adapt, one first needs to infer the user's capacity or the cognitive load a task imposes (the reader is invited to recall the order of the four steps for adaptation, in Section 2.1). Excluding self-reported questionnaires (which are not a great solution for real-time assessments), there are two main classes of objective methods for measuring cognitive load: (1) sensor- or (2) performance-based methods [34].

4.1.1 Sensor-based Cognitive Load Assessment

Cognitive load can be measured through diverse physiological sensors, enabling a system to track cognitive mental workload in real time [52]. These include sensors such as eye trackers [53], which monitor eye movements and also pupil diatation. Pupillary response has long been known to be a reflection of mental effort [54]; pupil dilatation, therefore, can serve as an assessment of brain activity and measure cognitive load [55, 56]. Blink rate can also be employed to infer cognitive workload, as it has been shown to be "one of the most effective measures of mental workload" [57]. Following the classification on objectivity and causal

relationship [34], these are indirect methods, for the measured attributes of interest only reflect the cognitive workload on users.

Brain Computer Interfaces (BCI) are tools that provide the brain with other communication channels than the normal ones, i.e., peripheral nerves and muscles [58]. As the name implies, BCIs are ways for computers to assess brain activity, be in the form of the brain's electrical signals (Electroencephalography, or EEG) [59, 60, 61] or changes in blood oxygenation (Functional Near-Infrared Spectroscopy or fNRI) [62] (this work does not concern itself with invasive or semi-invasive BCIs). Many BCIs are relatively cheap, easy to use and can serve as a reliable method for direct assessment of cognitive load.

In the following paragraphs, we present different studies where the assessment of cognitive load was performed through physiological sensors with the ultimate goal of adapting computer systems, and discuss their relationship with our approach. For example, concerned with changes in available cognitive resources, Tsiakas et al. [60] describe a system that monitors the user's concentration and adapts the task complexity in order to increase engagement and compliance. This is a preliminary work dedicated to data collection and analysis to identify relevant cues for the long-term goal of providing personalized training sessions. Tsiakas et al. employ the commercially available Muse EEG-based headband[1] to measure brain activity and infer concentration. They employed an assistive robot to give subjects feedback during the execution of a sequence-learning task (a WM task that evaluates the subjects' capacity of recalling a list of numbers, words or others) and built a dataset referring to the brain waves obtained through the EEG as well as data related to the task difficulty. Using the collected data they trained a Random Forest to predict user performance. The results show that the selected features are capable (in some measure) to predict user performance. They argue that training the random forest with data corresponding to a single user or a cluster of users could increase the classifier accuracy. This is a limitation that plagues ML solutions, and that we want to avoid: the lack of flexibility, which restricts the good performance to data patterns similar to the ones present in the training set. In a follow-up study, Tsiakas et al. describe in [63] the design of a data-driven Socially Assistive Robot system for personalized robot-assisted training. In this work, interactive reinforcement learning is used to adapt the robot's behavior to users' performance. Data corresponding to EEG signals, performance and engagement is collected and analyzed in order to find clusters. These clusters of users are then used to create simulation models and learn user-specific policies through reinforcement learning, an approach that still suffer from the limitation mentioned above.

Liu et al. in [57] propose a "cognitive pilot-aircraft interface" in order to enable

[1] https://choosemuse.com

single-pilot operations. As has been stressed before, pilots are susceptible to a lot of stress and cognitive load during flights, and this surcharge can result in errors or even fatal accidents. This scenario is aggravated during solo flights, where a pilot has to deal by herself with the many problems and decisions she encounters. In order to compensate for this augmented cognitive load, the proposed interface is capable of adapting itself, by taking over some demanding tasks for instance. The proposed architecture contain 3 main modules: Sensing, Estimation and Reconfiguration. The Sensing module collects data from dedicated sensors such as heart, respiratory and blink rates, pupil dilatation, EEG signals and others, as well as data of the environmental and operational conditions of the flight. The Estimation module classifies the collected data into cognitive load states as well as gives estimations of the cognitive demand of the current mission-task. Finally, the third module, Reconfiguration, takes input from the two previous modules and manages task distribution between the pilot and the automatized system, adapts the UI by regulating the amount of displayed information and gives out alerts. The proposed adaptation system is equiped with parameterized mathematical models for estimating the cognitive state and performance given the sensor data and two decision tables that select the level of automation to compensate overcharged states. The values of the parameters of the mathematical models come from the literature. The study presents initial simulations that show the viability of the proposition. Overall, the system is quite complex, making it difficult to be used without much adaptation in other applications. In our work, we are interested in providing a more general approach to the automatic modeling and personalization of interfaces, without the tedious process of parameter optimization or of modeling of each task individually. Nonetheless the work presented by Liu et al. is promising and the goal a very important one.

Still focused on the mental workload of pilots, the authors in [62] show that, by using Functional Near-Infrared (fNIR) spectroscopy, it is possible to assess the cognitive charge of a task as well as the perceived expertise of the subject. fNIR is a relativity new brain computer interface, where a number of infrared sensors detect changes in the blood oxygenation, giving researchers a picture of cerebral hemodynamic response. The study proposes two experiments, one where subjects are presented to various WM loads and a second one, where subjects control an unmanned air vehicle through a simulator. The reported results show that fNRI can be used to sense cognitive workloads and expertise. This study posits that expertise can be seen as an indirect measure of mental workload, for, as the authors themselves put it, “expertise tends to be associated with overall lower brain activity relative to novices”. Compared to our approach, this work addresses the assessment of a personalized mental workload rather the adaptation of tasks; however, the motivation behind it is to ultimately develop learning environments

able to personalize the training regiment to cognitive charge.

Authors in [64] describe their concept of an architecture for the automatic adaptation of the interface of a ground control station in order to reduce the cognitive load on humans commanding and operating multiple drones, or Unmanned Aerial Vehicles (UAV), at once. In situations where a single pilot is in control of multiple remote-controlled UAVs, the proposed "Cognitive Human-Machine Interfaces and Interactions" (or CHMI^2) framework intends to reduce the workload by switching the level of automation or by defining the level of detail presented in displayed information, as well as emitting alerts. This again is a preliminary work. A sensing module is responsible for collecting data regarding the user's physiological condition, i.e., cardiovascular (heart rate and pressure), eye (gaze, blink, pupillometry), brain signals (EEG and fNIR) and the pilot's control inputs (mouse positions and clicks) as well as external data (environmental and mission conditions). This data is then processed through ML algorithms in a classification module in order to estimate the pilot's cognitive state and the current task cognitive workload demand. Finally the pilot and mission states are inputted into the adaptation module responsible for selecting compensating measures. The presented framework is very similar to the one of Liu et al., described in [57] and discussed above.

Physiological measures are a reliable way that has long been used for inferring cognitive load. In situations such as the ones discussed above (especially in contexts where human error can have fatal consequences), monitoring different bodily responses might be a way of assuring that the user is not under too high a cognitive load and that his/her current capacity for considering information and taking adequate actions is nominal. However, if one's goal is to develop general adaptive methods that can be embedded in many different applications, the use of brain sensing devices isn't optimal. Strapping a number of sensors to users every time they interact with the system is obviously not practical, as it is time consuming, has somewhat invasive impacts and augments the solution's cost. This work will therefore focus on the assessment of WM capacity through non-physiological data.

4.1.2 Performance-based Cognitive Load Assessment

Performance-based methods use the user's performance to infer his/her cognitive load. This method is divided in two classes: primary task measurement, which considers the performance in the current task, and methods employing the dual-task paradigm [35]. In the context of measuring cognitive load, performance is typically inferred through measures of reaction time, accuracy, and error rate. However, it can also include memory retrieval time and correctness, time estimation, rate of physical activity and speech, among others [65]. This family of methods are non-invasive and arguably can be adapted to any computer application. The rest of this section focuses on relevant studies that infer cognitive load

from performance-based data.

Fan et al. [66], for instance, used data related to the performance of a secondary task (using the dual-task paradigm) as an observable signal to learn a hidden Markov model (HMM). This HMM correlates the observed signal to hidden states corresponding to different levels of cognitive load, ranging from "negligible" to "overwhelmed". The learned HMM uses data referring to the user's reaction time, accuracy and error signal in order to infer the hidden cognitive load state, therefore providing non-obtrusive measurements of cognitive load. Once the user's cognitive load is known, the proposed model is used to adapt the collaboration between humans and software agents. This study posits that human-agent collaboration can be improved by providing the agents with cognitive models of how humans function. For instance, by sensing an overload in its human peer, a virtual agent could try to compensate by taking over tasks that are consuming heavy cognitive resources, therefore, allowing the human to focus her attention on tasks where her role is indispensable. However, say a new user whose cognitive style or available resources have not yet been seen in the data collected to learn the HMM, then the agent won't be able to correctly assess the human cognitive load and will malfunction. The motivation behind the paper is excellent. The goal of cognitive models embedded in agents might be of great importance, yet the proposition of using data-based HMM carry the same limitations as other ML techniques, thus motivating our new approach.

Long Short-Term Memory (LSTM) networks are used in [67] to learn different patterns of sequential behavioural data in order to classify dynamically user's behaviour into either (1) under cognitive load or (2) not. The approach is based on data collected from users playing a memory game with or without a secondary task introducing extra cognitive load. Rather than using the performance of the secondary task to assess cognitive workload, the proposition here is to use the sequential data to differentiate both classes. To increase the size of the dataset, a theoretical memory model based on ACT-R [28] was used to generate additional data. The model's parameters were set so that the generated data closely resemble the collected one. The results show Long Short-Term Memory networks outperform a baseline Linear Discriminant Analysis model in predicting the two classes. As above, this work still carries the limitations of being purely based on data, as the authors themselves acknowledge: "the model is still in-progress and these results can be further improved by tuning the hyper-parameters and generating more training data".

Based on the fact that more and more users are interacting with interfaces through vocal instructions and that studies have revealed the influence of cognitive load (and to a lesser degree, time pressure) on speech patterns, researchers in [68] performed an experiment where users had to utter vocal inputs to a system as if

navigating through an airport. The study had users interact with the system under two conditions, while simulating the navigation using a computer interface (which introduces a supplementary cognitive charge) and under time pressure. A selection of six features such as the number of syllables and collected pauses were extracted from the data in order to learn a dynamic Bayesian network for recognizing the effects of time and cognitive load on speech. The results show that the Bayesian network was sensible enough to recognize users under time pressure with good accuracy; however, it did not perform as well when evaluating the condition of the navigating task. The researchers attribute the lack of performance in the navigation task to the fact that it wasn't highly cognitively challenging enough. This work posits that, through the assessment of situation-dependent resource limitations, a system can adapt itself by regulating the way it communicates, for instance by switching to a less demanding style of communication, an approach we directly tackle in our work.

Still concerned with changes in available cognitive resources, Jameson et al. [69] propose an architecture employing a dynamic Bayesian network of a system capable of assessing and adapting itself to a user's changing resources availability (here resources stand for time and WM). Jameson et al. discuss the fact that humans who deal on a daily basis with people suffering from temporary resource constraints, for instance firefighters answering emergency calls, take these restrictions into account by adapting their interaction by selectively minimizing or simplifying what they say. Here too, WM is viewed as a limited capacity the user has in order to perform a task, the system handling situations when the user's whole capacity isn't available (say the user is agitated or performing more than one task at a time). Though no experimental validation is presented, the work highlights the importance of rendering systems aware of resource limitations, and is thus another motivation for our work.

Finally, note that measurements of secondary task performance are sensitive to resources limitations and serve as a reliable technique for assessing them. However, they have rarely been applied in research on CLT [35] for the secondary task can interfere with the primary task, thus limiting the practical impact of this technique.

4.1.3 Adaptation

Most of the work discussed above, although interested in the development of cognitive load-sensitive applications, is concerned with the assessment of the load rather than the corresponding compensation part of the task. Feight et al. lay out four main different mechanisms for adapting system to the user's cognitive needs [70] (three of which can be devised from the previously discussed research):

- modification of interaction;

- modification of task allocation;
- modification of content;
- modification of task scheduling.

"Modification of interaction" relates to simplifying (or complexifying) the communication method, as seen in [57, 68], for instance by changing the interface layout. "Modification of task allocation" means dividing tasks between human and machine; when the human agent becomes overwhelmed, the system compensates his/her cognitive charge by taking over tasks that are less critical; this method is used in [57, 64, 66]. "Modification of content" is the dynamic adaptation of the quantity of presented information, seen in [57, 64]. Finally, "modification of function allocation" stands for the dynamic changes of task scheduling, task priority and duration. Our approach mostly follows the first and third items discussed here.

The authors in [52] present the main challenges and approaches for adapting systems to the perceived mental workload. This work discusses the evaluation of a series of different approaches for adapting workload, mostly based on the modification of function allocation method. Overall the study shows that systems capable of assessing cognitive workload and responding to it result in improvements in user experience, thus providing strong motivation for the general goal of our work. The reader is invited to refer to this article for a more detailed discussion on the evaluation of cognitive adaptive systems.

4.1.4 Discussion

The studies above highlight the importance of assessing cognitive load and adapting interfaces to users' cognitive states. In this section, we put into perspective their main findings with respect to our goals and approach.

The authors in [52] discuss the limitations of the adaptation techniques found in such literature They point out, for instance, that one aspect disregarded by most studies is the change of adaptation quality over time, as most studies only perform a small number of sessions per subject for data collection. Also, they notice that most EEG-based research concentrates on single-user-dependent systems, limiting applications to contexts where a "learning period" is necessary for the system to obtain baseline values for the adaptation features.

Moreover, most of the presented systems (as the approaches discussed in [52]) are data-based ML techniques. In general, researchers perform experiments employing the dual-task paradigm and collect data from a series of sensors such as EEGs and other physiological devices. Later they employ some ML technique for learning a model correlating these measures to cognitive load. This approach

seems efficient; yet it carries some limitations. For example, the neural network learned in [67] might work poorly when presented to a different population, as would the HMM in [66] and the user models of [67]. Deep learning and ML tools work effectively when the data used for learning is comprehensive enough to create a representation of the use case, as they try to fit a function capable of performing meaningful association. This can be problematic when the system is used with different populations that are not represented in the data, say people with cognitive deficits.

If the training database does not represent the full set of possible situations the system can encounter, then the system may have to deal with situations very much different from the ones it was trained on, meaning that it cannot be trusted [71]. Trustworthiness is however a key factor, widely recognized as crucial for the acceptance of "intelligent" systems in various domains [72]. Being able to explain a system's choice of action is crucial for building trust, in particular when dealing with assistive technologies, where the user has to trust the system's decision for it to be effective. This is, for now, critically lacking when dealing with black-box classifiers such as neural networks. And even if adjacent explainable ML models are developed to explain the system's reasoning, they cannot be 100% faithful to the original model, for, as Rudin puts it, "if the explanation was completely faithful to what the original model computes, the explanation would equal the original model, and one would not need the original model in the first place" [71].

In this work, we are interested in flexibility, as it shall focus on systems that can be used with users not yet previously seen. Another key factor we shall consider is interpretability, as we intend to be able to explain to the user why some changes in the UI are being made. Being able to explain the system choices might help users accept and trust the intelligent system. Given these considerations, the approach we intend to take requires well-understood and interpretable components that connect adaptation to the user's perceived cognitive load or capacity. This means one needs an interpretable model of how humans store and deal with presented information, i.e. a model of human WM, a subject we address in the next section.

4.2 Computational Models of WM

In this section, two computational models of WM are discussed; they both will be used in the sequel of this work. These models allow us to leave Badlley's non-computational homunculus behind, and either model user's differences, simulate WM evolutionary dynamics or both. The first model corresponds to WM in the context of Anderson's cognitive architecture, ACT-R, and is described in Section 4.2.1. Section 4.2.2 presents Suchow's Markov Decision Process-approach for WM's dynamics simulation.

4.2.1 WM in ACT-R

Anderson's ACT-R [28] is a cognitive architecture and also a unified theory of cognition. The goal of ACT-R is to serve as an explanatory structure of the brain in order to achieve the function of the mind (by "function of the mind", Anderson means the human cognition in all of its complexity). This fixed architecture can be used to model all cognitive tasks. By modeling a given task in the ACT-R architecture, one obtains a simulation programs that can be used to generate theoretical predictions of the task outcome.

The ACT-R architecture contains eight modules divided in three categories: perceptual, motor and central. The two perceptual modules, Visual and Aural, are responsible for detecting important information in the context of the task. The motor (or response) modules, Manual and Vocal, perform appropriate actions, and the other four modules, Procedural, Declarative, Goal and Imaginal, are the central modules responsible for the coordination of thoughts and actions.

In the context of WM, the most interesting module is the declarative one. It serves as a window to the past where learned information or facts can be accessed, in the same way the perceptual modules can perceive the current environment. The declarative module stores the learned knowledge so that it is readily accessible in order to accomplish a given task.

The procedural module also comes into play when the given task can be divided into smaller sub-tasks. The procedural module is the collection of learned skills. It consists of a set of production rules and actions to be performed in order to achieve a given goal. In ACT-R, the "current goal" (find the sum of 7 and 3, for example) will drive the selection of production rules stored in the procedural module. The actions represent accesses to different modules, as is the retrieval of information in the declarative memory.

Symbolic and subsymbolic components in ACT-R

ACT-R is a hybrid system with two levels of abstraction: symbolic and subsymbolic. The symbolic level is an abstract representation of how the brain stores knowledge. The subsymbolic level is an abstraction of the mechanisms involved in the process of making information available, or how the knowledge encoded in the symbolic representations is to be accessed. Symbolic structures have subsymbolic quantities associated with them, which drive how fast or where knowledge activation will occur.

As said above, the procedure module stores production rules of actions and accesses in order to achieve a given goal. These rules represent how information is to be moved throughout different modules. At the symbolic level, the rules have the form

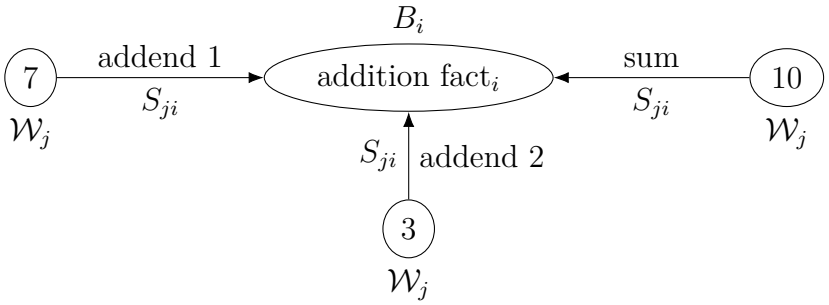


Figure 4.1: Symbolic representation of the declarative knowledge "7 + 3 = 10" with the associated subsymbolic quantities.

IF <condition>, THEN <action>.

The condition specifies a pattern, or the circumstances of the task at hand, and the action the different accesses of information. For instance, if the presented task is to solve the equation $10 - x = 3$, a possible action is to request access to the information referring to the difference between 10 and 3 in the declarative module.

At a subsymbolic level, every rule has an utility value associated to it. When different production rules might correspond to the presented circumstances, the rule with the highest utility value is chosen. The goal acts as a filter in selecting which production rule is selected and then propagates attentional activation to different modules in order to achieve the accomplishment of a given task (see below).

Declarative module

The declarative module gives the system access to its past. Inside ACT-R's declarative module, at the symbolic level, facts are represented as networks of interconnected nodes. These graphs correspond to encoded knowledge and are also called "chunks". For example, the fact that $7+3 = 10$ would be represented as the graph shown in Figure 4.1. The central node, labeled "addition fact$_i$" connected to the elements 7, 3 and 10 represents the memory that "$7 + 3 = 10$". Figure 4.1 also depicts the subsymbolic values associated with that memory. Chunks in declarative memory have activation values associated with them that drive the speed and success of their activation. Every time a new node is created (or information is learned), that node is given some initial activation. In ACT-R, activation is seen as a currency and is the main factor for processing and learning in declarative memory, as the node's activation drives its accessibility. After a chunk is created, its activation decreases over time; however, when its information is reaccessed, it

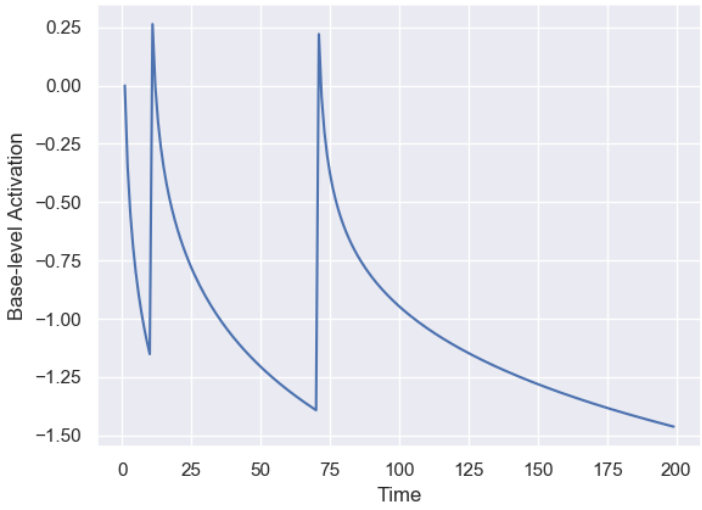


Figure 4.2: Evolution of the base-level activation for a node created at time $t = 0$ and accessed twice, at time $t = 10$ and $t = 70$. Each access increases the node's base-level activation, and is linked to learning.

is "refreshed", meaning that its base-level activation gains a boost. The i-th chunk has a subsymbolic base-level activation B_i, which is given by:

$$B_i = \ln\left(\sum_{k=1}^{n} t_k^{-d}\right),$$

where n corresponds to the number of activations that chunk had and time t_k to the time passed since the k-th access to that memory. The decay parameter d defines the speed of base-level activation decay; in the ACT-R community, a value of 0.5 has surfaced through many applications as the default value for this parameter [28]. Multiple activations to an information result in an augmentation of the base-level activation, which is related to learning. Figure 4.2 depicts the evolution of the base-level activation of a memory that was created at time $t_1 = 0$ and then accessed twice, at time $t_2 = 10$ and $t_3 = 70$. One can see that every access corresponds to a boost in the activation level.

The elements connected to a node influence its activation value. The addition fact node is represented in Figure 4.1 as being linked to three elements, the first and second addends, 7 and 3, and the sum 10, through weighted connections. The associated weights connecting element j to node i are noted S_{ji} and indicate the strength of the relationship between them. The closer two facts are, the higher the associated weight value. The total activation value A_i for chunk i is given by:

$$A_i = B_i + \sum_{j \in C} \mathcal{W}_j S_{ji}, \tag{4.1}$$

where C is the current context and $\mathcal{W}_j$ is the attentional weight given to the j-th element in the context. This means that a chunk receives some added activation from the context elements according to how close those elements are to the chunk (S_{ji}).

In Bayesian terms, B_i corresponds to the prior, i.e., the base-level activation equals to how often an information is used. The term $\mathcal{W}_j S_{ji}$ refers to the likelihood ratio of element j being part of the context, given that chunk i is being required. And A_i refers to the posterior odds that chunk i is needed in the given context. Therefore, Eqn. 4.1 can also be written as:

$$\log(posterior(i|C)) = \log(prior(i)) + \sum_{j \in C} \log(likelihood(j|i)).$$

Attentional source activation

Eqn. 4.1 shows that the activation of some information in declarative memory is influenced by the elements connected to it. The closer the elements are to the knowledge node, the higher the value of S_{ji} and consequently the stronger their influence over the activation. The S_{ji} value is given by:

$$S_{ji} = \ln\left(\frac{P(i|j)}{P(i)}\right);$$

it reflects how likely node i becomes when element j is present in the context. However, A_i also depends on W_j, the attentional weight given to element j. When a task is presented, the Goal module is responsible for allotting source activation to the elements present in the task context; the element source activation is given by:

$$\mathcal{W}_j = \frac{\mathcal{W}}{n}, \tag{4.2}$$

where n is the number of presented elements related to the task and $\mathcal{W}$ is the total amount of attention focused in the current goal. The $\mathcal{W}$ parameter reflects individual differences in memory retrieval performance; it is a key factor for WM (as we will see later) and relates to the amount of attention being focused on the task. When complex tasks are presented, where lots of elements (high n) are necessary for consideration, then the source activation spreading from the Goal module is increased, resulting in a worse retrieval performance.

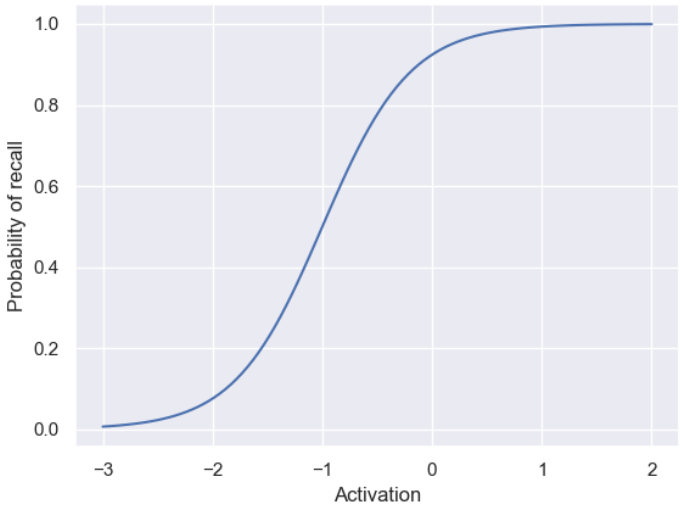


Figure 4.3: Example of the probability of recall over activation values. The parameters τ and s where set to -1 and 0.4 respectively. When the activation value is higher than τ, the probability of recall is higher than 0.5.

The activation value is translated into a retrieval probability through the following equation:

$$P = \frac{1}{1 + e^{-(A_i - \tau)/s}}, \tag{4.3}$$

where τ is the activation threshold below which the odds of the information been retrieved are low. Note that the retrieval is not a binary process; s is a parameter relative to the smoothness of the probability of activation. The parameter s serves to smooth the evolution of the probability of recall over the activation value; if s is very small, then the probability of recall becomes a step function. Figure 4.3 depicts an example of probability of recall versus activation.

WM in the context of ACT-R

In terms of the ACT-R architecture, WM can be seen either as the subset of the declarative memory being activated during the accomplishment of a given task or the propagation of the source activation from the goal node [73]. Following both visions, WM is not seen as a special buffer for information storage, but is defined as the attentional mechanism that selects which information to activate in the context of a given task. In ACT-R, the task goal represents a person's focus of attention, which is then distributed by propagating activation values. Therefore WM limitations in ACT-R can be seen as the amount of attentional

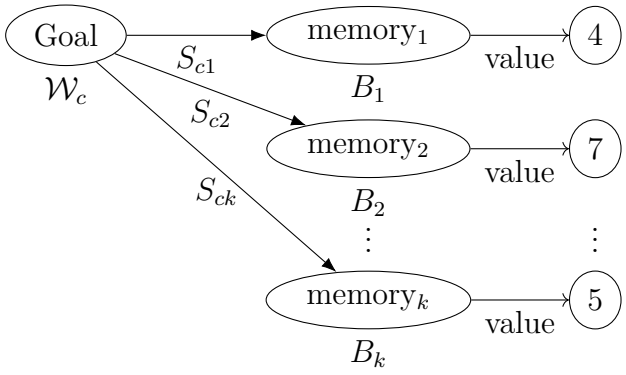


Figure 4.4: Symbolic representation of a memory span task with the corresponding subsymbolic values represented.

source activation $\mathcal{W}$ that is distributed from the goal node. Linking $\mathcal{W}$ to the number of information nodes that can be activated at the same time, this defines the WM limit, capacity-wise. The other WM limitation comes from the time during which an information is retrievable. After a chunk is created, the base-level activation decays over time, resulting in a degradation of the recall probability.

A simple memory span task, where a person has to remember k digits at a time, could be represented in the ACT-R architecture as shown in Figure 4.4. Note that here, only the task of remembering the digits is being modeled, and it is considered that all the information is presented at the same time. In the case where the position of each digit is also requested, then a goal node corresponding to the position has to be added, and the procedural module will be responsible for allocating the attentional activation between the modules (which corresponds to a decrease in the attentional activation available to the single task of remembering the digits). It is also possible to model the mechanism responsible for rehearsal, i.e., the conscious effort of maintaining particular digits in memory, using ACT-R with different and specific production rules.

The Goal node spreads the attentional activation alloted to the current task $\mathcal{W}_c$ between the different chunks $\{i\}_{i=1}^{k}$ representing the encoded memory of each of the k digits. The strength of association S_{ci} can be set by default to reflect the "fan" of an element, i.e., the number of connections with different nodes that element has. The more connected an element is, the less activation it propagates; in this case, one can set $S_{ci} = 1/k$. The attentional source $\mathcal{W}_c$ will then be divided equally between the memory nodes. Table 4.1 sums up the parameters one needs to set in order to simulate the model described here and obtain recall curves.

Once the parameters are set, one can use the equations described above and obtain recall curves representing the evolution of digit retrieval probability as

Table 4.1: WM Simulation parameters for ACT-R WM model.

$\mathcal{W}$	Total amount of source activation
k	Number of information items in WM
τ	Activation threshold
d	Decay parameter
T	Total simulation time
s	Smoothness of the probability of activation

depicted in Figure 4.5. This formulation can serve therefore as a simulator of information decay in WM.

4.2.2 Quantic WM Model

J.W. Suchow proposes in [74, 75] a probabilistic model of the dynamics of human WM. In such a model, the evolutionary dynamics of the information stored in the WM is considered a Moran process [76], a stochastic formalism often used to describe the dynamics of finite populations in biology. In a Moran process, at each instant where the state of the population may change, an individual, chosen at random, dies and another is chosen for reproduction, ensuring a constant yet varying population.

Suchow models the evolutionary dynamics of information in WM as the evolution of a finite population of "memory quanta". When information is presented, a number of quanta is allotted to each information item stored in the WM: the more quanta assigned to an information there is, the better encoded it is, and therefore the easier it is to be retrieved. Although the authors in [75] are noncommittal about what these quanta represent (they could take a number of forms, such as clusters of neurons in the prefrontal cortex, cycles in time-based refreshing processes or other elements), they make it clear that this is a limited commodity the availability of which affects WM performance. Logically, the total number of quanta is positively correlated to the cognitive capacity of an individual; the more available quanta there are, the better the quality and stability of memory.

Following the rules of Moran processes, at each time step, a random quantum assigned to an information "dies" while the so-called WM "maintenance mechanism" selects another quantum to "reproduce". The quantum chosen for reproduction can be related to the same information as the dead one, thus ensuring the persistence in memory of this information. If, however, the quantum selected for reproduction isn't one allotted to the degraded information, but to a different one, the latter is then reinforced in detriment of the former. This dynamics results in

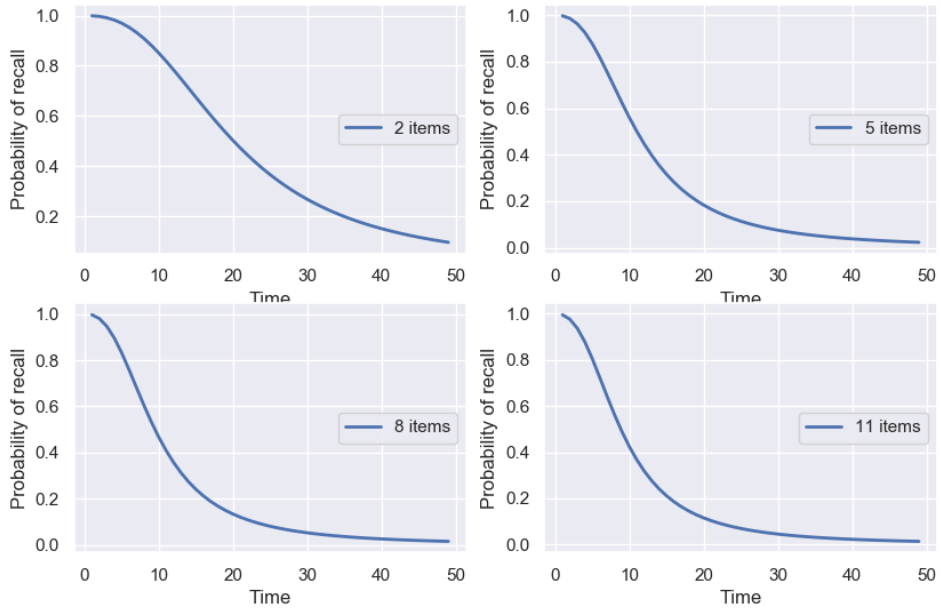


Figure 4.5: Recall curves depicting the decay of encoded digits in WM over time for different number of presented information. The other parameters of Table 4.1 were set to $W = 1$, $\tau = -1$, $d = -0.5$ and $s = 0.2$. These parameters could also be set to values according to some optimal fitting in order to obtain recall curves that more closely resemble the behavior of a specific person or group of people (note, in particular, that the unit of time is not fixed here). As the cognitive charge (k) goes up, the degradation of the encoded information accelerates, due to the decrease of the base activation values.

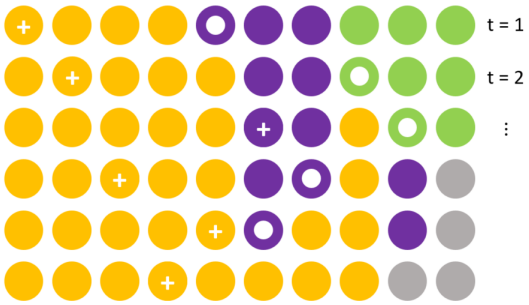


Figure 4.6: Example of execution of a particular maintenance policy for 10 quanta, 3 items and a stability threshold $L = 2$. The circles correspond to the quanta population. The colors orange, green and purple indicate to which information item each quantum is allotted, while a gray-colored circle corresponds to a quantum that is not alloted to any information item. A quantum with a "+" sign is being selected by the maintenance mechanism for reproduction, while a quantum with a white circle inside is the random quantum that degrades. At the end of the execution of the maintenance policy, only the orange information item remains in memory.

a competition for quanta, i.e., for fixation in memory. This model also employs a stability threshold L: any information associated to less than L quanta is considered forgotten and cannot be restored via reproduction. Figure 4.6 depicts an example of the evolutionary dynamics of a population of quanta, following the procedure described above.

MDP formulation

Suchow's WM dynamics is modeled as a Markov decision process (MDP). A MDP is used to model decision making in partially stochastic environments, more precisely on Markov processes. The later are characterized by the fact that the future state evolution is only dependent of the present state, which means that the process is independent of the events that occurred in the past [77]. Formally, a MDP is defined by a state space S, a set of actions A, a probabilistic transition function $\tau : S \times A \times S \to [0, 1]$ that characterizes the probability distribution over the possible next states s' given the present state s and a selected action a and finally, a reward (or cost) function $\rho : S \times A \to R$ that yields the immediate consequence the agent taking an action in a given state gets (in some extended MDP models, the reward also depends on s'). The goal when using a MDP is to find the optimal policy $\Pi^* : S \to A$ that maps a given state s to the optimal action a the agent should take in order to maximize (or minimize) its accumulated reward (or cost).

In Suchow's model, the WM maintenance mechanism acts as the MDP agent. S is the set defined as:

$$S = \{[n_1, \ldots, n_k] \ / \ \textstyle\sum_{i=1}^{k} n_i = Q\},$$

and stands for all of the possible allocations of Q quanta into k information bins. Each action a_j from A represents the selection of a quantum from a specific memory bin, here the j-th, for reproduction. Following Moran's principle, at each system iteration, one quantum decays (i.e., dies) from a bin, say the i-th, randomly selected with probability n_i/Q, while the maintenance mechanism chooses a specific action, say a_j, to have one of the quanta of bin j reproduced; so, if the system is at state $s = [n_1, n_2, \ldots, n_k]$ and the agent selects action a_1, the probability of the agent landing in state $s' = [n_1 + 1, n_2 - 1, \ldots, n_k]$ is given by $\tau(s, a_1, s') = n_2/Q$, which is the probability that one quantum from the second bin was selected to decay.

Regarding the reward function ρ, the behavior of the maintenance mechanism handling the information stored in the WM might vary along the user's goal; information items can be remembered or forgotten intentionally. Thus ρ is clearly task-dependent.

Optimal maintenance policy

As stated before, the space state is all the possible allocations of quanta into information bins. For example, if one has 10 quanta and 4 bins, it makes for 286 possible states; however 40 quanta and 8 items result in 62,891,499 possible states. This increasingly larger space state demands for a generalized policy that can be applied to any configuration. The authors in [74] did so by analyzing the optimal policy obtained in simpler cases in order to propose a generalization. They suggest that the optimal policy of the previously defined MDP can be approximated by a simple strategy known as Luce's choice axiom [78]. This axiom states that when faced with a choice, the decision maker will mostly base his/her decision on the perceived values of the various options at the time of choice, in a "greedy" fashion. Therefore the probability $P(a)$ of selecting action a from a set of alternatives A is given by

$$P(a) = \frac{v(a)^\sigma}{\sum_{x \in A} v(x)^\sigma},$$

where $v(x)$ stands for the strength of the signal generated by action x, and σ is the sensibility of the decision maker[2]. By varying the value of σ for a fixed definition of v, Suchow shows that one obtains different macroscopic behaviors for the WM

[2]Care must be taken to avoid divisions by zero; we don't address these details here.

Table 4.2: MDP simulation parameters for Suchow's WM model.

Q	Number of quanta in WM
k	Number of information items in WM
L	Stability threshold [number of quanta]
δ_t	Time step between actions [ms]
T	Total simulation time [ms]
σ	Sensibility of the decision maker

maintenance mechanism, adapted to different tasks, and draws attention to five specific values of sensibility: $0, 1, -1, +\infty$ and $-\infty$.

Choosing $\sigma = 0$ leads to an unconditional policy, i.e., action choice is independent of the current state and insensitive to the perceived signals. If $\sigma = 1$, the policy will give preference to actions that have the highest perceived value, while the opposite occurs when $\sigma = -1$. Finally, when $\sigma = +\infty$, the maintenance mechanism will always choose the action that has the strongest perceived signal, while when $\sigma = -\infty$, the weakest one will be selected.

Stochastic simulation of WM dynamics

Keeping track of users' WM capacity to model information recall and oblivion relies on the simulation of the MDP defined above; this requires the setting of six parameters, given in Table 4.2, and the definition of the strength function v. One also needs to specify an initial state $s_0 = [n_1, \ldots, n_k]$ representing the default distribution of quanta between information items.

As said before, WM management is task-dependent. The setting of the initial state s_0 and the definition of the signal generated by the possible actions v and the sensibility parameter σ, which characterize the Luce choice axiom underlying the MDP policy, depend thus on the task.

Once the initial state and the simulation parameters are set, one can perform various stochastic simulations of memory degradation, using the optimal policy specified above. For one simulation, at each δ_t, until T is reached, the decision maker, following its policy, will specify which action to perform, i.e., choose which bin will be maintained or see its number of quanta increased (remember that at the very least, the decision maker can only maintain an information in memory, because it has no control on which information is going to be randomly degraded). The ratio of the number of bins with more than L quanta over k is the recall probability of the WM, i.e., the probability of memory retention (or the complement of memory loss). Such a simulation thus yields a recall curve $r(t)$, with time t vary-

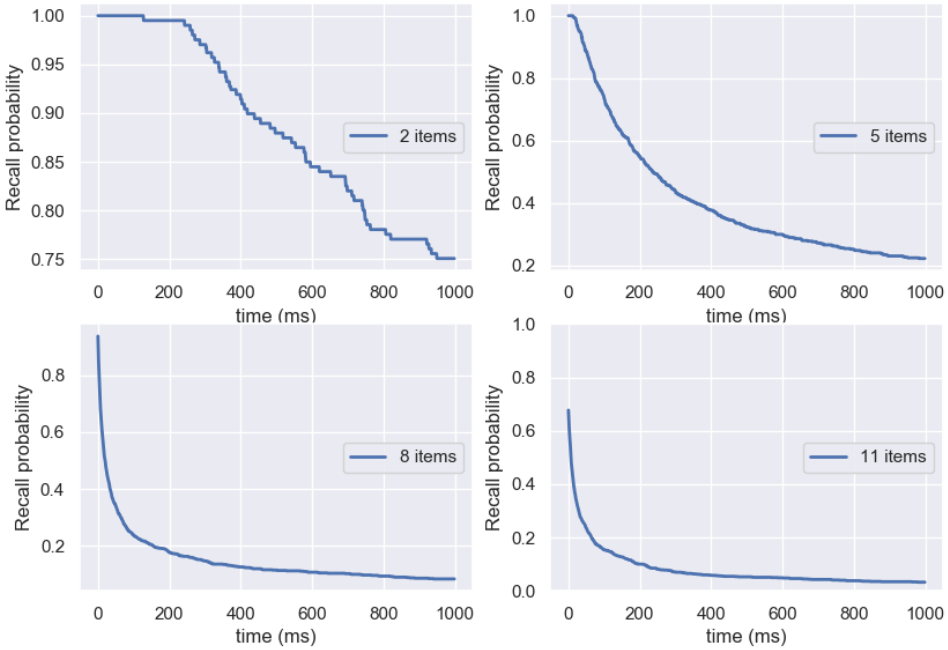


Figure 4.7: Recall probability $r(t)$ for different numbers, k, of items. s_0 was set by distributing the Q quanta in the k bins homogeneously; if Q is too small to fill each bin with at least L quanta, the maximum number of bins are filled with L quanta, and the remaining ones are distributed randomly across bins. Also, we define $v(a_i) = n_i$, i.e., the strength of information fixation in bin b_i, while setting $\sigma = 1$. The other parameters were set as $Q = 60$, $L = 7$, $\delta_t = 10$ and T =1,000. One can see that when more items are presented, less quanta are available, and therefore the oblivion of information is occurs faster.

ing from 0 to T, by increments of δ_t. Given the stochastic nature of the model, a large number of simulations is necessary to average the recall curve. Figure 4.7 presents the average recall curves of 100 simulations for different values of k, given a specific configuration of the parameters of Table 4.2.

Chapter 5

Memory Adaptation Through Cognitive Handling Simulation

Memory Adaptation Through Cognitive Handling Simulation, ou MATCHS, est un nouveau cadre formel capable d'adapter les tâches lorsque la performance de leur bonne exécution dépend des capacités cognitives des utilisateurs. Dans le contexte de cette thèse, il présente une première tentative pour effectuer l'adaptation HCI aux limitations de la WM d'une personne.

Le cadre MATCHS repose sur les mêmes hypothèses que la théorie de la charge cognitive (voir la section 3.2), c'est-à-dire que, si une personne ayant une capacité cognitive plus élevée est capable de stocker et de travailler avec plus d'informations, alors cette personne présentera une performance plus élevée dans les tâches qui dépendent de la WM. Notre cadre s'inspire également du modèle de dynamique de WM de Suchow (section 4.2.2) et de l'idée de quanta de mémoire. En principe, si ces quanta représentent la capacité cognitive de l'utilisateur et si les performances sur une tâche les reflètent, l'augmentation et la diminution d'une estimation du nombre de quanta de l'utilisateur en fonction des performances observées devraient être un moyen d'adapter en continu la capacité de WM estimée de l'utilisateur, permettant à un système de s'adapter à l'augmentation de l'expertise ou à la dégradation continue des capacités cognitives, dans des cas tels que les maladies neurodégénératives, par exemple.

Les sections suivantes sont organisées comme suit. La section 5.1 décrit le cadre en détail. La section 5.2 présente un cas d'utilisation développé pour tester MATCHS. La section 5.2.1 présente une tâche sous la forme d'un jeu, utilisé à des fins de validation. La section 5.2.2 discute de la sélection de certains paramètres de MATCHS en fonction de la tâche choisie. Les performances de MATCHS sont d'abord discutées dans la section 5.2.3, en décrivant les résultats obtenus lors de l'utilisation de MATCHS avec des joueurs simulés, et, plus tard, la section 5.2.4 donne un compte rendu détaillé d'une campagne de test avec des joueurs réels et

présente les résultats obtenus lors du coup d'envoi de MATCHS. Enfin, la section 5.3 discute des performances de modélisation et d'adaptation de MATCHS.

Memory Adaptation Through Cognitive Handling Simulation, or MATCHS, is a new framework capable of adapting tasks when the performance of their proper completion depends on the users' cognitive capacities. In the context of this thesis, it presents a first attempt to perform HCI adaptation to a person's WM limitations.

The MATCHS framework is build upon the same assumptions as Cognitive Load Theory (refer to Section 3.2), i.e., that if a person with higher cognitive capacity is able to store, and work with, more information, then that person will present a higher performance in tasks that are WM-dependent. Our framework also gets some of its inspiration from Suchow's model of WM dynamics (Section 4.2.2) and the idea of memory quanta. In principle, if these quanta represent the user's cognitive capacity and if performance on a task reflects them, then incrementing and decreasing an estimation of the user's number of quanta according to the observed performance should be a way of continuously tailor the user's estimated WM capacity, allowing a system to adapt to increases in expertise, or the continuous decay of cognitive capacity, in cases such as neurodegenerative diseases, for instance.

The next sections are organized as follows. Section 5.1 describes the framework in detail. Section 5.2 presents a use-case developed to test MATCHS. Section 5.2.1 introduces a simple game-like task, used for validation purposes. Section 5.2.2 discusses the selection of some of MATCHS' parameters according to the selected task. MATCHS' performance is discussed first in Section 5.2.3, by describing the results obtained when using MATCHS with simulated players, and later, Section 5.2.4 gives a detailed account of one test campaign with actual players and presents the results obtained when lighting MATCHS up. Finally, Section 5.3 discusses MATCHS modeling and adaptation performance.

5.1 Presentation

MATCHS has a modular architecture with 4 main modules, one of which corresponds to Suchow's model of WM (described in Section 4.2.2) and works here as a simulator for WM dynamics. MATCHS consists of a closed-loop control system capable of tracking the user's estimated cognitive capacity by adjusting it according to the his/her performance. In the field of control theory, traditional techniques are employed to manipulate the input of complex dynamic systems in order to correct or limit the deviation of a measured value from a desired one [79]. This is typically done by measuring the value of the controlled variable and applying a control signal in order to ensure that a certain specification is verified, usually

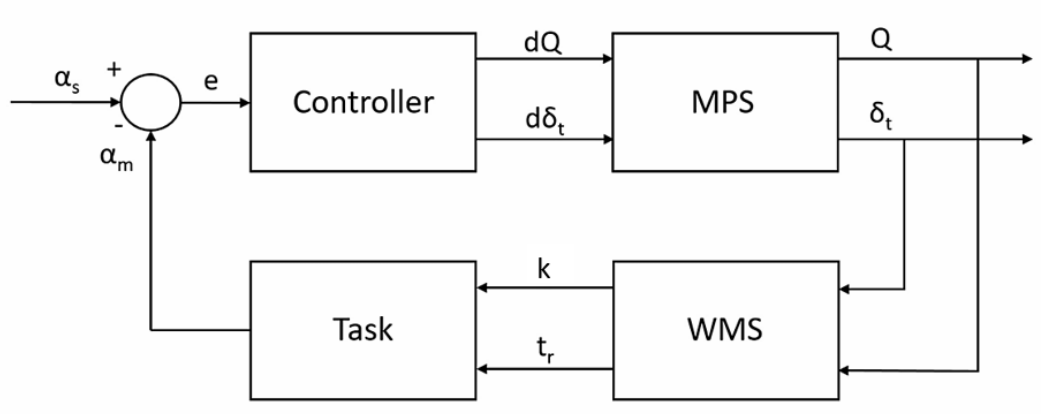


Figure 5.1: MATCHS control loop (see also Table 5.1)

defined in terms of a reference level or reference trajectory that the controlled system's output should track as closely as possible [80].

MATCHS is a closed-loop control system where the controlled variable is the user's WM performance. In control systems, the error value is fed to the controller so as to reduce the error and bring the output of the system to a desired value. In MATCHS, instead of having the error signal driving actuators or valve openings, it drives the estimation of parameters. These parameters are used to construct a model of the user's WM dynamics. The estimated WM capacity is then used to drive the UI adaptation. The user's measured performance when presented with the adapted task serves as a sensor of the difference between the expected performance, derived from the user model, and the measured one. The error signal then will determine how much to change the user model over the subsequent iterations of the task at hand.

5.1.1 MATCHS main loop

Figure 5.1 presents the main components of the MATCHS framework, while Table 5.1 lists all the relevant parameters. This framework is able to adapt a given task (represented by the Task block in Figure 5.1) to an estimation of the cognitive capacity of users. MATCHS is able to simulate the user's WM performance by employing Suchow's MDP together with the estimated user parameters, obtaining recall curves that reflect how information will degrade over time. Therefore, MATCHS can adapt the task by deciding how much information to present, or for how long the user will have to retain that information. The adaptation here corresponds to the modification-of-content method for adaptation of UIs [70], meaning

Table 5.1: MATCHS parameters

α_s	Desired accuracy
α_m	User's accuracy
e	Error value
dQ	Change in quanta value
$d\delta_t$	Change in δ_t value
Q	User's quanta estimation
δ_t	User's time step between policy iterations estimation
k	Number of presented items
t_r	Retention time
MPS	Memory Parameter Space
WMS	Working Memory Simulator

the framework decides the amount of information to present in order to adapt to the user's characteristics.

The main element of the MATCHS framework is the Memory Parameter Space (MPS). In the MPS block, an approximation of the user's WM parameters Q and δ_t, which are some of the WM parameters from Table 4.2 that characterize the WM dynamics, is stored. The MPS is described in detail in Section 5.1.2. The main goal of the MATCHS Controller is thus, depending on the difference between the actual user's performance in a defined Task (α_m), and the target accuracy (α_s), measured by the error e, with

$$e = \alpha_s - \alpha_m,$$

to update its estimation of his/her WM parameters by incrementing (or decrementing) them by dQ and $d\delta_t$. More refined estimations of the user's WM, as s/he interacts with the Task, will ensue:

$$Q = Q' + dQ,$$

$$\delta_t = \delta'_t + d\delta_t,$$

where Q' and δ'_t are the prior estimations. Note that initial estimations for Q and δ_t are needed to start the system.

In order to employ a MATCHS-equipped system, one needs to specify one key application-dependent parameter: a target accuracy $\alpha_s \in [0, 1]$. The parameter α_s is the maximum percentage of information that can be forgotten while ensuring the accomplishment of the task. For instance, in applications where the person

has to retain at least 80% of the information presented when solving a problem, α_s is set to 0.2, while, in applications where all the presented information is strictly necessary, α_s is set to 0.

MATCHS' current estimation of the user's WM parameters are fed into a Working Memory Simulator (WMS), described in Section 5.1.3. The outputted retention time t_r, a measure of how long the user can retain information before forgetting $100 \times \alpha_s$ % of it, can then be used to adapt, in an application-dependent manner, the Task interface. The user interface can also be adapted by modifying k, e.g., by presenting more (or less) information if the user error rate is too high.

The adapted Task must provide, when completed, an estimate of the user's performance α_m, i.e., the measured proportion of forgotten information. The error signal e that drives the Controller indicates how far the outputted simulated oblivion behavior is from the user's actual one.

5.1.2 Memory Parameter Space (MPS)

The core element of MATCHS is the Memory Parameter Space (MPS); it represents the domain of the parameters that characterize users' WM behaviors. There are two clear categories of simulation parameters from Table 4.2: task-dependent and user-dependent. The number of presented items (k) and total simulation time (T) are task-dependent parameters, as they characterize the external task that is presented to the user. However, the number of quanta (Q) and the time step between actions (δ_t) are user-dependent parameters, since they depend on the user's capacity and define the evolutionary dynamics of the degradation of the stored information.

However, the stability threshold (L) and the sensibility of the decision maker (σ) fall in a gray area, for they are neither completely task- nor user-dependent. The stability threshold can be interpreted as the complexity of the stored information: considering a resource-based approach to WM, complex information should be harder to encode. Therefore, for a fixed capacity, if L is larger, more resources (in this case, quanta) are needed to encode each item, resulting in overall less information stored in WM. However, as discussed before, the complexity of information is subjective, as performance in WM tends to increase with practice [81, 82]. This means that the L parameter might change over time. The σ parameter depends on the task, as discussed in Section 4.2.2, but also on the user's strategy, as not always the user will employ the optimal one.

MATCHS' MPS is therefore defined as a 2D space with dimensions Q, the number of available quanta, and δ_t, the time interval (in ms) between policy iterations, which are the user-dependent parameters. Here we posit that adaptation of a task can be performed (at some level) by tracking these two parameters.

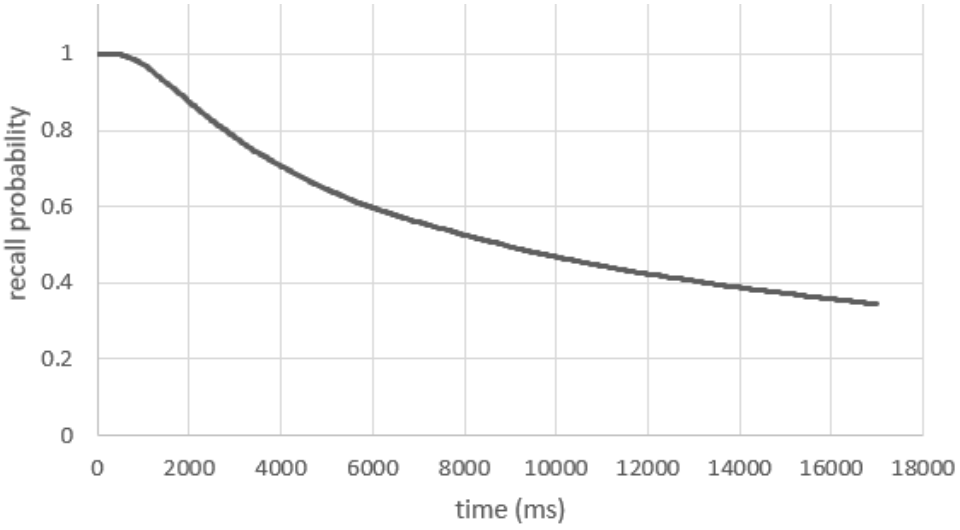


Figure 5.2: Recall curve $r(t)$. The initial state s_0 is set with a homogeneous repartition of quanta between each item and once again, we define the strength function as $v(a_i) = n_i$. The other parameters are set to $Q = 112$, $k = 8$, $\delta_t = 13$ ms, $T = 17$ s, $L = 7$ and $\sigma = 1$.

5.1.3 Working Memory Simulator (WMS)

The Working Memory Simulator (WMS) uses the MDP described in Section 4.2.2, together with a decision maker's policy, to simulate the evolution of someone's WM as described in Section 4.2.2. The user-dependent (Q and δ_t) and task-dependent (k and T) parameters are variable according to the user's performance; however, the other parameters have to be set accordingly. The selection of the parameters stability threshold (L) and the sensibility of the decision maker (σ) are not purely task-dependent, yet here they are set according to the chosen task. This design choice is based on the hypothesis that the "user-dependency" part of these two parameters can be compensated, if needed, through Q. For instance, having a less than optimal strategy will result in an apparent lower cognitive capacity, as would having less proficiency with the kind of information being manipulated (higher value for L). Some of these choices will be discussed in the experimental validation section below.

Once all parameters are set, WMS provides a simulation capability for the WM dynamics. Given its stochastic nature, each simulation is different; therefore a somewhat large number of simulations (here heuristically set to 60) is necessary to obtain a reliable average the recall curve. An example of a typical recall curve $r(t)$ is shown in Figure 5.2; this curve provides the probability that a memory item is in the WM at a given time.

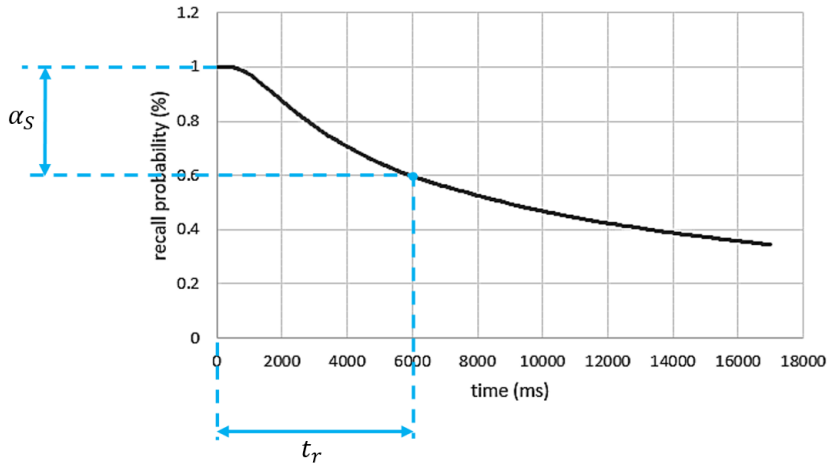


Figure 5.3: Detail of the relation between α_s and t_r in the recall curve, in the case where $\alpha_s = 0.4$ and therefore $t_r = 6,000$ ms.

WMS uses the parameter α_s (desired accuracy) to provide, as output, the corresponding retention time t_r, i.e., the value of t corresponding to $(1 - \alpha_s)$ on the $r(t)$ curve for the number of presented items k:

$$t_r = r^{-1}(1 - \alpha_s).$$

If, for instance, α_s is set to 0.4 (40 % of the information allowed to be forgotten), then the WMS will, for the previously set value of k, search the corresponding recall curve $r(t)$ for the time $t = t_r$ that ensures $r(t) = 1 - \alpha_s = 0.6$ (i.e., 60 % of retained information, i.e., 40% of forgotten information). If the recall curve is the one depicted in Figure 5.2, then t_r will be set to about 6,000 ms, as shown in Figure 5.3. If α_s is instead set to 0.6, then t_r will be around 13,000 ms, as depicted in Figure 5.4.

However, the desired accuracy may not be reachable with the current parameters. In this case, if the minimum attainable recall probability $r(t_{max})$ (after a maximum time t_{max}, which depends upong the task context) is above $1 - \alpha_s$, then WMS will increase the previous value of k by one and perform additional simulations. Increasing k will result in a recall curve that degrades faster, as shown in Figure 4.7; therefore, WMS will keep increasing k until the desired accuracy is attainable withing the maximum time. Similarly, if the maximum reachable accuracy is below $1 - \alpha_s$, then WMS will decrease the number of items by one unit and perform simulations, until the desired accuracy is reached and a proper value for t_r can be found. If none of these task modifications succeed, the user is

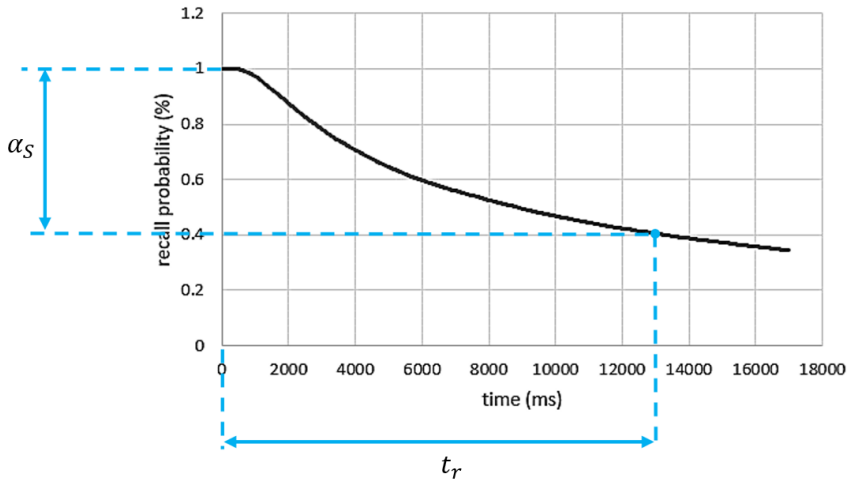


Figure 5.4: Detail of the relation between α_s and t_r in the recall curve, when $\alpha_s = 0.6$ and $t_r = 13,000$ ms

deemed unable to complete the task at hand, calling for external intervention.

5.2 Experimental Validation

In this section is discussed the experimental validation of the MATCHS framework for the adaptation of HCI to user's WM limitations. To validate MATCHS, the game called Match^{2}s (for "match match") was developed, largely based on the study described in [83]. It is a visual game where the player's score is WM-dependent. An initial validation step consisted of a virtual player called Player 2 whose memory dynamics is a direct implementation of Suchow's WM model. Player 2 was used to validate some design choices and analyze in detail the evolution of some of MATCHS' parameters, before testing MATCHS with human users. We then tested the final version of the game with a cohort of 20 players. Those experiments are described below.

5.2.1 Match^{2}s

Match^{2}s consists of eight yellow squares with a "?" sign positioned in a circular fixed order around a white "+" sign, as shown in Figure 5.5. The maximum number of squares (8) was chosen, since we saw that the limits of WM, for simple information, is said to be around 7 ± 2. Since we didn't want the players to be cognitively charged by having to look for the visual cues, we always displayed

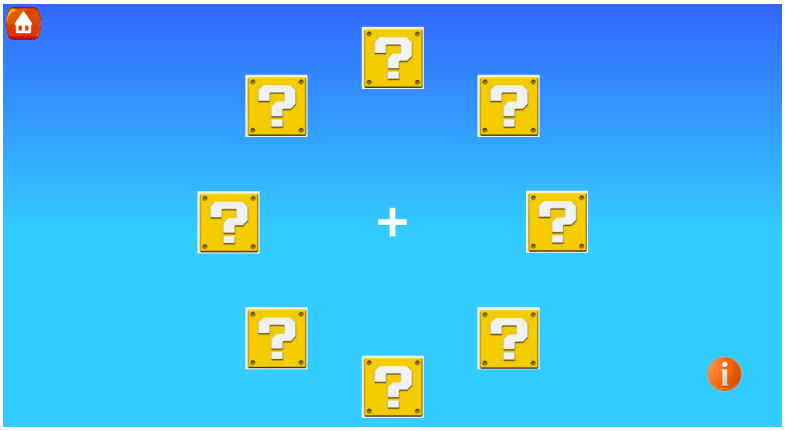


Figure 5.5: Match^{2}s eight yellow boxes are disposed in a fixed order around a circle with a white "+" sign in the middle for fixation. The fixed disposition allows the players to focus on the task of remembering information without having to search for the visual cues.

the squares in a circle around a fixed point, which somewhat corresponds to the standard oval human field of view. Since the position of the squares are fixed, the player doesn't have to search for the visual cues, as she knows where the colors are going to pop up. However, since not all the 8 boxes are always going to present a visual cue, there's still some searching to be done, as, if less than eight colors are to be cued, the presented colors are disposed in a random order around the circle.

Each turn of Match^{2}s consists on having N of the eight "?" randomly colored boxes displayed during 500 ms, as depicted in Figure 5.6, after which the colored squares are hidden in yellow boxes. The choice of using visual information as the foundation of Match^{2}s was made so that we can present all the information at the same time. Suchow's MDP formulation for WM maintenance does not specify how the available quanta are divided between information at an initial time. If we were to work with verbal cues, for instance, the presented information would have to be presented in an 1D array sequence; therefore, we would need to investigate the proper time-wise way to initially distribute the quanta to form a state zero. Moreover, information presented in a sequence would be susceptible to the serial positioning effect [84], with higher primacy and recency probabilities of being remembered. This would require additional work to extend, and then validate, Suchow's model, something we avoid with our visual design.

The presented N colors are a random combination of the eight possible colors without doubles. The colors chosen at first where the eight basic colors (red, orange, yellow, green, blue, purple, brown and black); however after some testing with users, we decided to include white and pink instead of orange and purple,

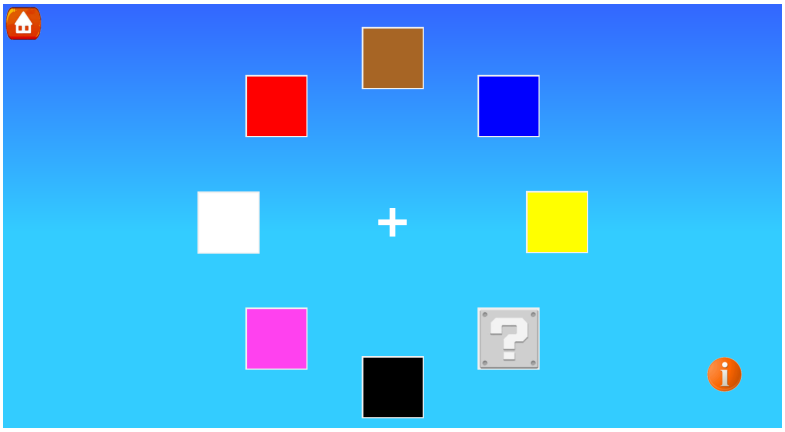


Figure 5.6: Colored cues presented to the player. Here $N = 7$, so only 7 colors are cued; therefore one box remains faded in gray, to signal that it's not being used.

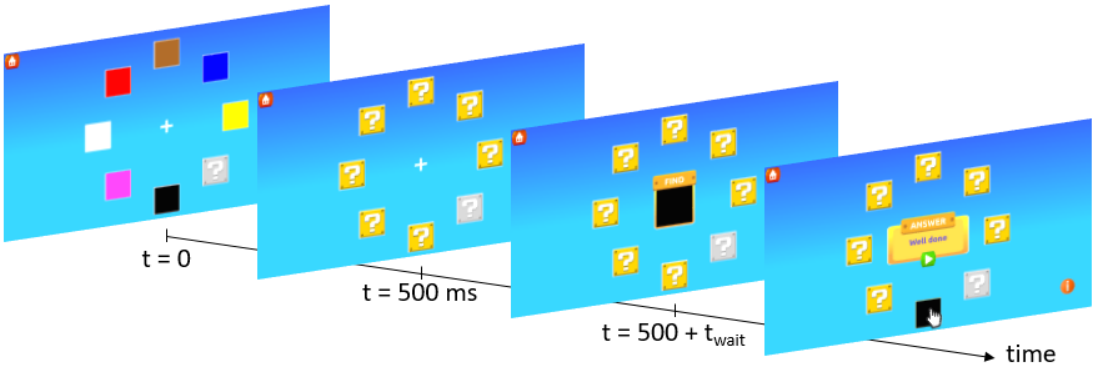


Figure 5.7: Example of a Match^{2}s game turn

as some players were mixing those colors with some of the others. We tested the eight colors with some users before collecting data, and they all seemed comfortable identifying these colors.

After the presentation phase, the colors remain hidden during a variable waiting time t_{wait}, after which a colored box pops up asking the player to click on the "?" box that, s/he thinks, hides the square of the same color. Figure 5.7 depicts the succession of phases of a typical Match^{2}s' turn. It is worth noticing that in the presented turn, only 7 colors are presented; therefore one of the eight squares is left unchanged and colored gray to signal that it is not activated.

Match^{2}s was developed[1] to test MATCHS. The parameters of the game, the number of colors and the hiding time (N and t_{wait}) are the task-dependent pa-

[1]Match^{2}s was coded in python v2.7, with PySide v1.2.4 providing the python binding for the GUI toolkit Qt.

rameters of Suchow's model, as described at the beginning of Section 5.1.2. The goal of Match^{2}s is to "force" a specific score on the players. Recalling the recall curves of Figure 4.7 and knowing how much information is presented to the player, one can, by increasing t_{wait}, make the player forget a bigger portion of the presented information. Also by increasing the number of colored squares cued, the degradation of information is accelerated. By varying these two parameters, one can control the oblivion dynamics of the player's WM and, in consequence, the player's performance, if, that is, the user-dependent parameters are known. This feature can be useful, for instance, in a video game situation, if one wishes to put the player in a state of "flow"[85], where the difficulty of the game matches the player's capacity, immersing her in a state of focus and enjoyment.

5.2.2 Task-Dependent Parameter Setting

Match^{2}s was used as the Task block of Figure 5.1 to test the MATCHS framework. MATCHS will thus perform the adaptation of Match^{2}s' parameters, i.e., the number N of presented squares and the wait time t_{wait}, according to the user's perceived cognitive capacity. This means that, when coupled to MATCHS, $N = k$ and $t_r = t_{wait}$.

As said before, we shall consider here the parameters stability threshold (L) and sensibility of the decision maker (σ) as being solely task-dependent, and therefore, they need to be set according to the task, as is the setting of the initial state s_0. In tasks such as Match^{2}s, players score higher if they are able to retain the maximum number of information items for the longest period of time possible. One can then assume that, on average, players will try to remember as much information as they possibly can, without giving particular preference to a particular stimulus. Recalling from Section 4.2.2 that each state of the MDP formulation corresponds to a partition of all the quanta between k bins b_i, s_0 is then defined as the distribution of the Q quanta in the k bins homogeneously; if Q is too small to fill each bin with at least L quanta, the maximum number of bins are filled with L quanta, and the remaining ones are distributed randomly across bins. Also, we define $v(a_i) = n_i$, i.e., the strength of information fixation in bin b_i, while setting $\sigma = 1$. The probability of choosing action a_i that reinforces bin b_i becomes:

$$P(a_i) = \frac{n_i}{\sum_{j=1}^{k} n_j},$$

and, therefore, $P(a_i)$ will increase proportionally to n_i, i.e., with the number of quanta in b_i. This ensures that our memory maintenance system chooses to maintain the memories that are already better fixated. Eventually, as more and more memory bins degrade, fewer memory bins will remain alive, resulting in a competition for maintenance, and consequently, oblivion.

The α_s parameter that drives the controller has to be set and will correspond to Match^{2}s difficulty, as MATCHS tries to force the user to perform with the specified accuracy $(1-\alpha_s)$. Note though that, since the Match^{2}s gameplay offers the player the possibility of simply guessing where the queried color is hidden among the N squares, this case must be taken into account in the WMS when assessing the WM capacity. In order to do so, a retrieval curve $R(t)$, denoting the probability of finding, at time t, the correct hidden square, is used by the WMS in lieu of the recall $r(t)$:

$$R(t) = \frac{1 - r(t)}{N} + r(t).$$

In order to make Match^{2}s's gameplay more dynamic and prevent player boredom, Match^{2}s is configured to limit the adaptable t_{wait} to a maximum value t_{max}, set at 7 s, thus avoiding situations where good players have to wait tens of seconds before a query is issued. Thus, as soon as t_r is larger than t_{max}, the WMS will increase k by one, resulting in a more difficult game with more squares. A different value for k implies a new retrieval curve and, consequently (to maintain α_s), a new, and smaller, value for t_r and, thus, t_{wait}. Similarly, a minimum time t_{min}, set at 200 ms, is also defined, so that, when $t_r < t_{min}$, the system will decrease k by one unit.

Since a single Match^{2}s turn does not provide enough data to deduce a meaningful value for α_m, a batch process is used. All the query results collected in the last d turns (we use $d = 20$, heuristically, assuming α_m at 0 when the game starts) are used to compute an experimental

$$\alpha_m = \frac{n_f}{d},$$

where n_f is the number of failed queries. The controller and WMS processes are thus executed once at the end of each batch.

All that is left to be specified are the stability threshold L and the Controller. Following [75], we set L to 7. For simplicity reasons, the Controller was set as a simple proportional gain G that regulates how the estimated parameters move around the Q dimension of the MPS. In practice, that means that δ_t is fixed and $\mathrm{d}Q = Ge$. Once again, δ_t was set to 10, in accord with [75]. This choice is discussed and validated in the next section, where we report on the first test of MATCHS, with simulated players.

5.2.3 Player Two

In order to test the MATCHS framework as well as to validate some of our design choices, we implemented "virtual players" called Player Two. These players correspond to simulators of humans assumed to have a WM exactly like Suchow's

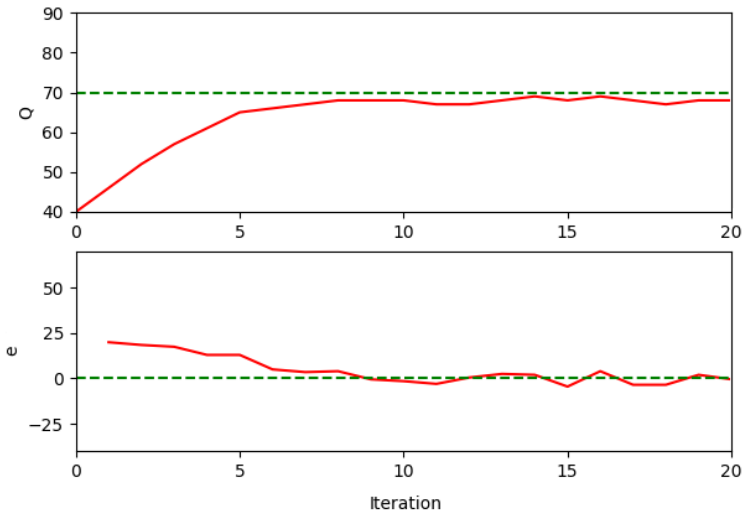


Figure 5.8: Evolution of the quanta estimation and error value during Player Two iterations.

WM model, or our WMS, albeit with fixed values Q_p, δ_{tp} and L_p for, respectively, the number of quanta, the time step between actions and the stability threshold. At each turn of Match^{2}s, Player Two is supposed to behave according to a recall curve $r(t)$ generated using Q_p, δ_{tp} and L_p, together with the number of presented items $k = N$.

When Match^{2}s presents a query at time t_{wait} to Player Two, it can, just like a real player, try to guess where the queried color is hidden. To simulate this, Player Two will compute the retrieval probability $R(t_{wait})$ and infer, based on this value and using a properly set random function, whether the queried square is supposed to be found or not.

Figure 5.8 shows the performance of Player Two "playing" Match^{2}s for 20 system iterations, using $G = 0.25$, $Q_p = 70$, $\delta_{tp} = 11$ ms and $L_p = 7$. The initial guess of MATCHS's WPS was set to $Q = 40$, and the desired accuracy, to $\alpha_s = 0.2$. In the top curve, the evolution of the estimated number of quanta Q is depicted with the continuous line, while Q_p is represented by the constant dashed line. The bottom curve shows the evolution of the error value e. One can see that after 10 system iterations, the error value is already stable near 0 and also that the estimated Q converged to Q_p. This illustrates how the error value drives changes in the quanta estimation, and also how these estimations result in adapted task parameters that, when presented to Player Two, makes the user accuracy converge to the desired one.

Player Two was also used to test some of the design choices previously made.

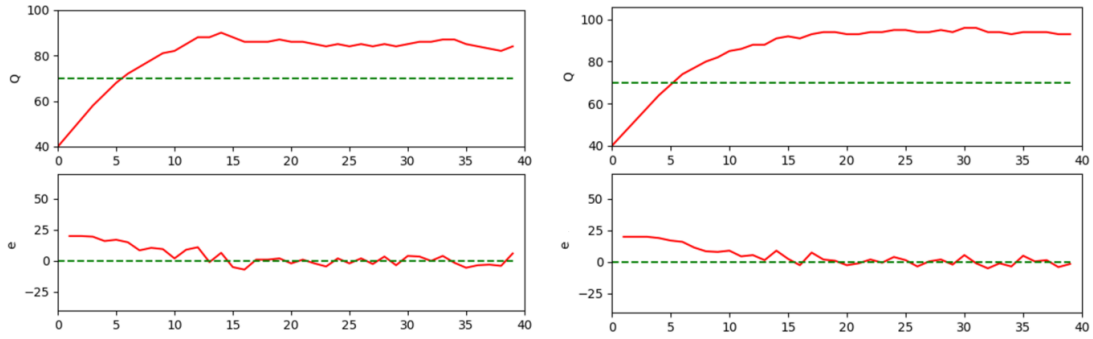


Figure 5.9: Left: $Q_p = 70$, $\delta_{tp} = 20$, $L_p = 7$; Right: $Q_p = 70$, $\delta_{tp} = 10$, $L_p = 3$. These two curves show that even if MATCHS is set with parameters that don't correspond to the user's, by adjusting Q, the framework is able to find the task parameters necessary for making the error value converge to zero.

One can observe that, even when Player Two has values for δ_t and L different from MATCHS's WMS ones, by simply adjusting Q, WMS is able to obtain a retrieval curve quite similar to Player Two's curve. Figure 5.9 shows the evolution of Q (in red, as the top curve) and e (also in red, as the bottom curve) when MATCHS interacts with a Player Two set with different configuration parameters. Here MATCHS's WPS was also set initially with $Q = 40$.

Figure 5.9 shows that a higher value of δ_{tp} can be compensated by a larger Q; since $\delta_{tp} > \delta_t$, Player Two's memory tends to degrade slower, therefore presenting the behavior of a person with high cognitive capacity. The same thing happens when $L_p < L$; a smaller L_p results in a better encoding of information, therefore inducing a behavior similar to having a higher cognitive capacity.

After testing different configurations, the Controller's gain was set heuristically to $G = 0.3$.

5.2.4 Results

We recruited 20 participants (9 females), ranging in age between 18 and 40 (25.47 $\pm$ 4.92), to play Match^{2}s, all with (or at least pursuing) a higher education and without cognitive impairment. We started each Match^{2}s game by presenting 7 squares at once and setting $\alpha_s = 0.3$. Afterwards, WMS will, at the beginning of every new batch, search for the parameters k and t_r in order for the user to recall 70% of the presented information, which corresponds to a quite complex cognitive task.

Each participant played 125 turns of Match^{2}s. The first 5 turns consisted of a training phase where the game's concepts were explained and the players could

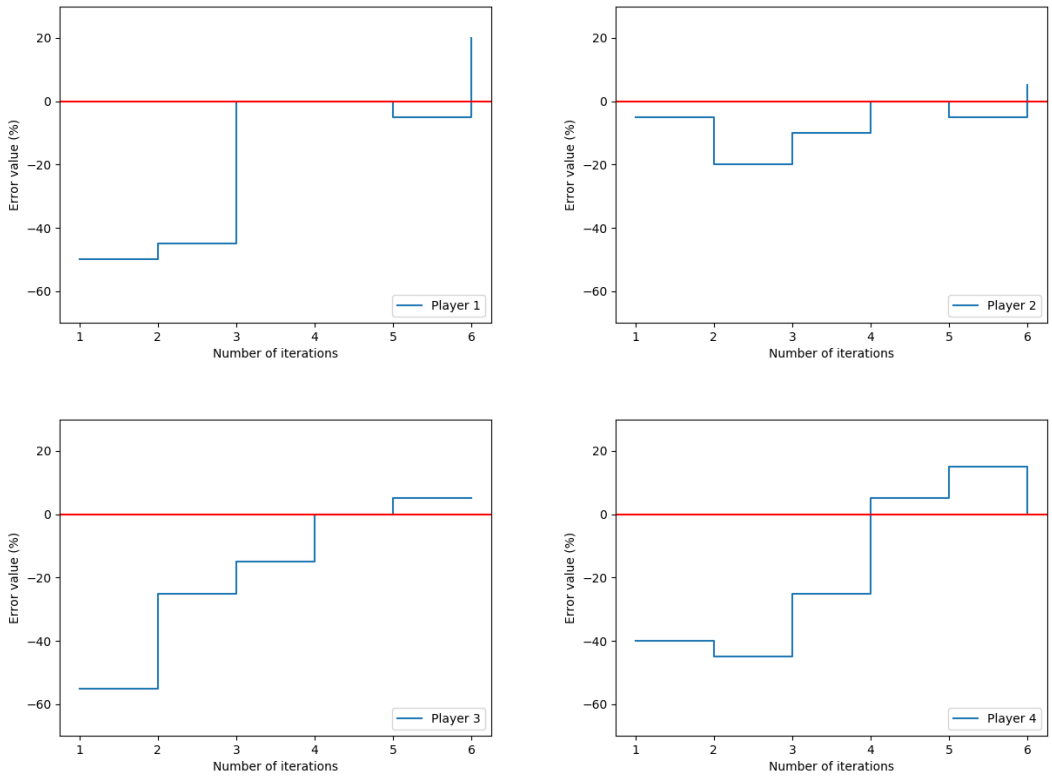


Figure 5.10: Evolution of the error value (in blue) for four of Match^{2}s' 20 players.

familiarize themselves with the game. During the next 120 turns, MATCHS was applied (WPS was set initially to $Q = 68$), controlling the presented information according to the player's performance. The error e was computed at the end of each batch of 20 turns. The collected data can be accessed at http://cri.ensmp.fr/auhwm/. The following figures summarize the main findings regarding MATCHS and its application to Match^{2}s (see next section for a detailed discussion of these data).

Figure 5.10 shows the evolution of the error value e for four of the 20 Match^{2}s players (in blue, with a red line on 0 for reference).

Figure 5.11 depicts the evolution of the quanta estimations for the same four players of Figure 5.10. Here the initial quanta estimation is presented (iteration 0), which is the same for all the players.

Figure 5.12 presents the evolution of the number of quanta for each of the 20 players, over the 6 batches.

Figure 5.13 depicts the distribution of the estimated number of quanta for every player at each iteration (without the initial quanta estimation, which was

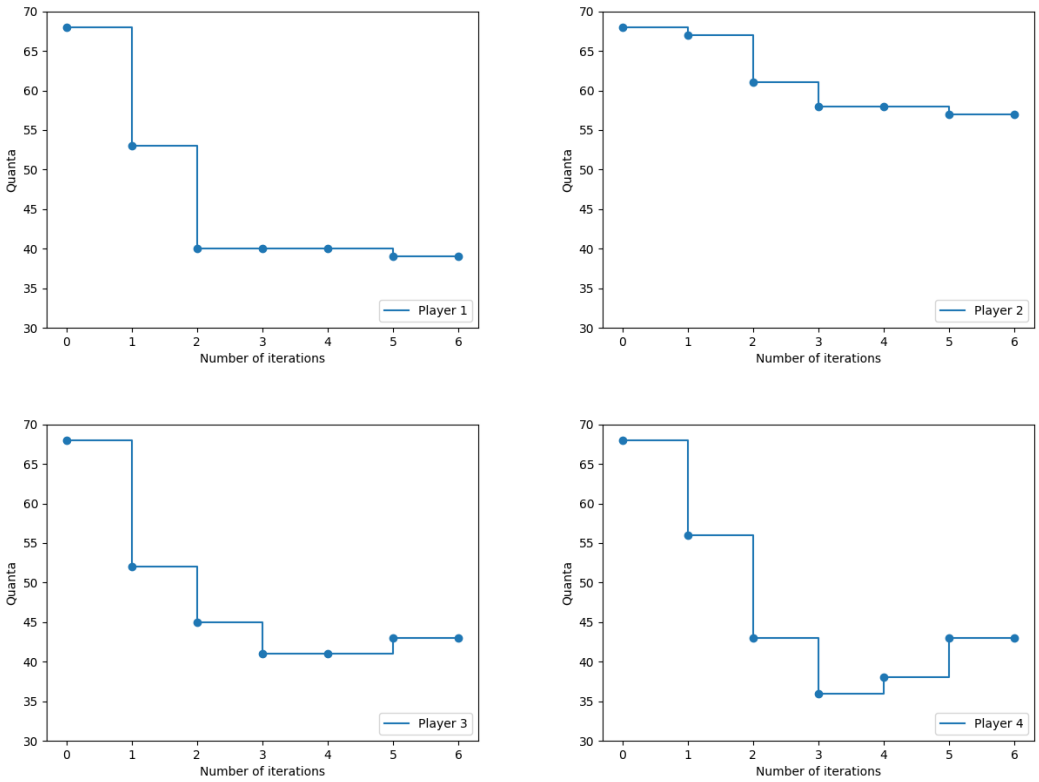


Figure 5.11: Evolution of the quanta estimations for the four players of Figure 5.10.

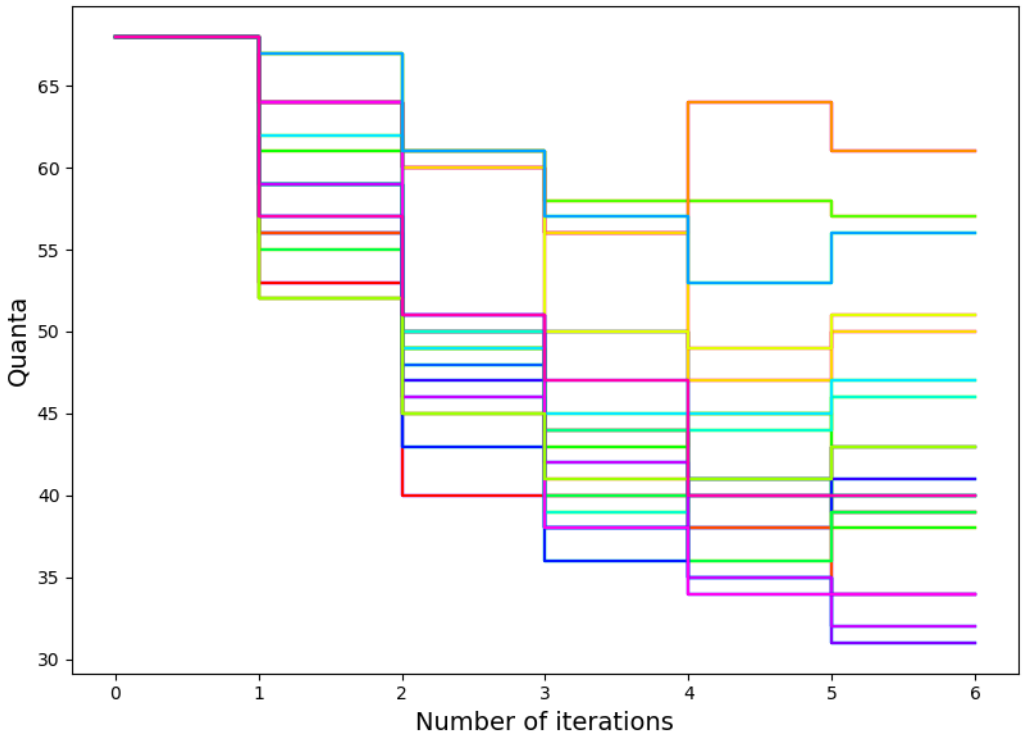


Figure 5.12: Evolution of the estimated quanta values for all the 20 players.

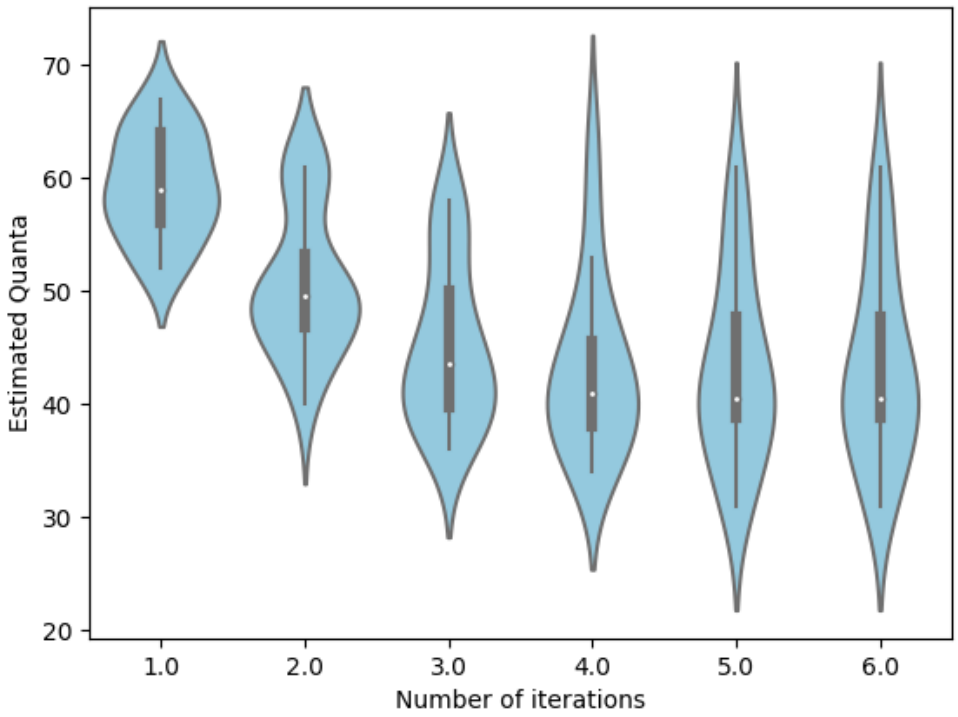


Figure 5.13: Distribution of the estimated quanta values for all the 20 players over the 6 system iterations.

the same for everyone). The median is the white dot; the thicker bar shows the interquartile range, while the thin bar shows the rest of the distribution. Around each line there is also depicted the probability density of the data in blue. The advantage of the used violin plot over the box plot is that it provides information on how the population is distributed in the form of the probability distribution.

The evolution of the mean absolute error value for the population of the 20 players is shown in Figure 5.14.

5.3 Discussion

As shown in Figure 5.10, every player presented a different performance and progression over the 120 turns, as illustrated by the different evolutions of the error values. Therefore, the evolution of the task's adaptation was different for every player. However, in the same figure, one can observe a common trend, namely that the error value gets closer to zero during the first system iterations (except

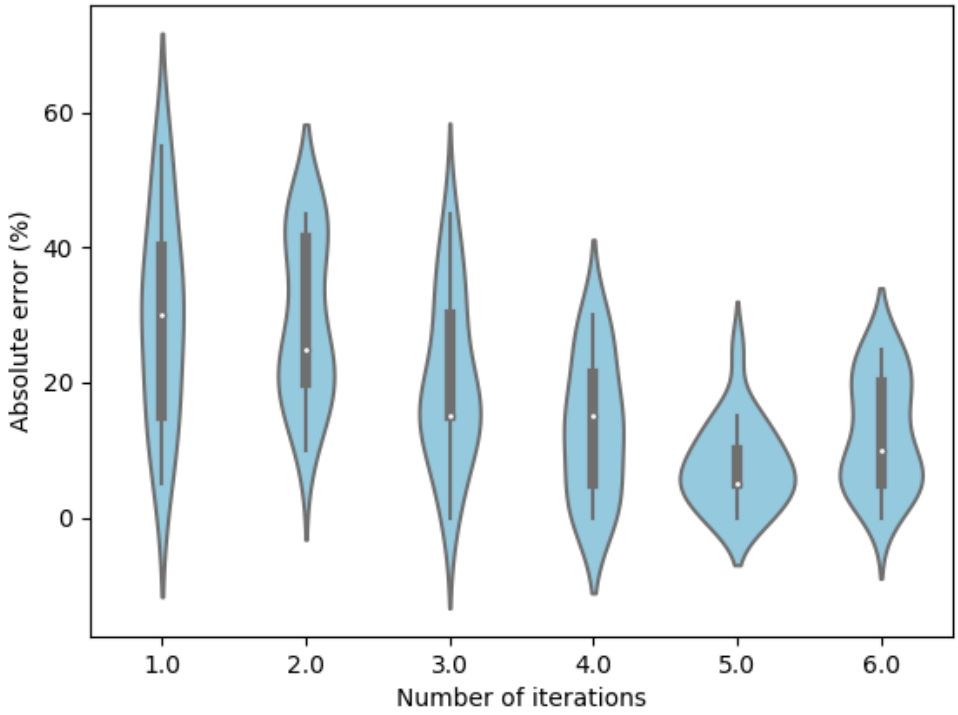


Figure 5.14: Distribution of the absolute error value for all the 20 players while playing Match^{2}s, across iterations.

for player 2, an issue we address below). In the later iterations, some overshooting of the error value is observable, where the error signal exceeds zero. This is a common occurrence in the step response of control systems, which indicates that the settling time has not yet been reached. At first, the larger the absolute error value is, the more significant are the changes of the estimated user parameters inside the MPS, resulting in more visible differences in the presented task parameters. As the error value gets closer to zero, the changes in the task parameters are less evident, resulting in a more refined adapted task. The overshooting of the error value corresponds to a higher observed WM performance, or $\alpha_s > \alpha_m$, which signifies that the player forgot less information than intended, meaning that the task was overly simplified. The positive error value will then result in a more complex task, which is likely to result in a larger α_m.

When compared to the other 3 players mentioned in Figure 5.10, player 2 had a higher cognitive capacity as is seen in the evolution of this player's quanta estimations (Figure 5.11). This player's initial error is considerably small, meaning that the initial quanta estimation ($Q = 68$) was not too far from the player's final one (when compared to the other presented players), which implies that the task parameters of the first batch weren't too far from the ones needed for the player to perform with $\alpha_s = 0.3$. Therefore, player 2's error value was small from the beginning, and its evolution corresponds to an oscillation around 0, as does the others player's error value after the first iterations.

As stressed in Section 3.1.2, WM's capacity is one one of the strongest factors impacting individual differences in cognitive abilities, therefore it is only natural that the players performed differently when playing Match^{2}s. The evolution of the estimated quanta values of every player shown in Figure 5.12 illustrates this fact, as does the augmented spread of the quanta distribution depicted in Figure 5.13. At first, all players started with the same estimated quanta value (68) at iteration 0; later as the players interacted more with the game, MATCHS refined the estimations of the quanta numbers. In consequence, the task could be personalized for each player in accord with his/her characteristics.

Opposed to the behavior of the quanta distribution for all the players, the distribution of the error values, in Figure 5.14, seems to concentrate over the batches. To analyze this behavior, a linear least squares regression was applied over the data corresponding to the absolute error value of all 20 players. The obtained result is shown in Figure 5.15, together with the evolution of the mean absolute error value (in blue). The linear regression strongly suggests that the slope is negative (null hypothesis that the slope is zero and p-value $= 3.11 \times 10^{-11}$), i.e., the error value diminishes over time. This shows that the MATCHS framework is capable of adapting (at some level) the user interface by regulating the number of presented information items as well as the retention time to the user's WM

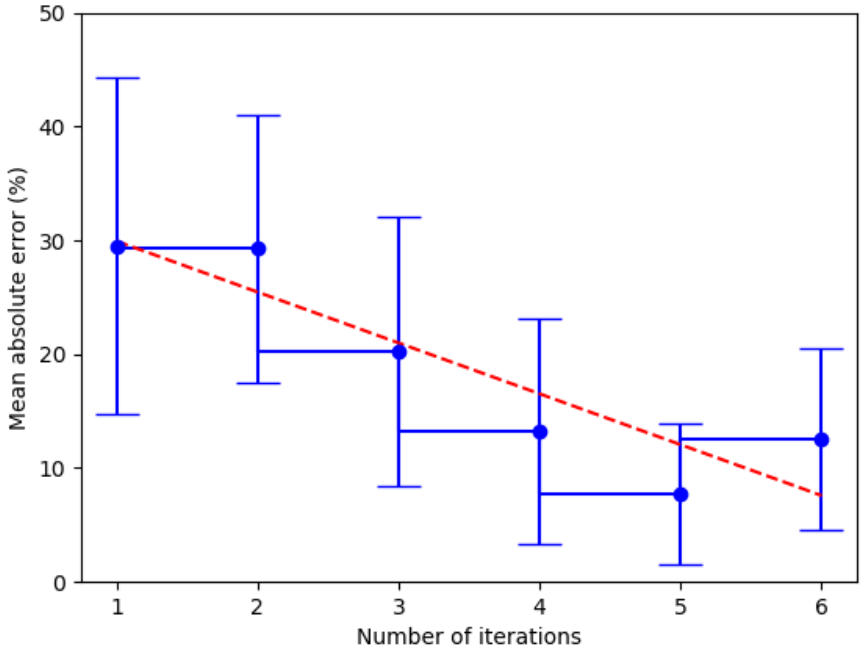


Figure 5.15: Evolution of the mean absolute error, for all players of Match^{2}s, across iterations.

capacity.

However, although the mean absolute error regularly diminishes as the number of MATCHS iterations increases, at the very last one, it slightly augments. This might be an overshoot of the error curve, meaning that 6 iterations are not enough for the error value to reach its settle time. Another possible explanation is that the players were able to familiarize themselves enough with the gameplay to come up with different strategies. A common player remark was: "I was getting better by the last batch", which reflects MATCHS' adaptation, but, in some measure, also the user's adaptation to the game. Of course a better score will result in a harder task to be accomplished next, forcing the player score back down again.

The MATCHS framework is based upon one hypothesis that might contribute to its less than optimal performance. It is assumed that all variations in the user's performance are due to differences in cognitive capacity i.e., differences in quanta population, which means that changes in attention, motivation or fatigue, which are known to be key elements capable of modulating WM performance (as discussed in Section 3), are not taken into account. Therefore, changes of these time-dependent factors lead to frequent updates of the estimated MPS parame-

ters. This can be particularly noticeable when the user interacts with the system in a somewhat intermittent manner, yielding local fluctuations of attention or motivation that have a large impact on the user's estimated capacity. For instance, in the case where a local fluctuation in the player's attention level results in a poor performance, MATCHS will consider that it corresponds to a decrease in the user's cognitive capacity, and new parameter estimations are made. The user's newly estimated parameters may however result in an ill-adapted task. Still considering the example of a poor performance due to an attention fluctuation, the worse performance will produce a simpler task, which, when presented to the user will (likely) result in a higher performance and so on. Therefore a simple local fluctuation in attention will result in the system being destabilized for a while before the estimations settle again.

MATCHS is a simple framework based upon complex core elements. Its modular architecture allows it to be easily improved, for instance by using estimations of attention or/and motivation to regulate the amount of quanta that enter WMS. Therefore, local fluctuations in such factors would not lead to changes in the user's MPS-estimated parameters, creating a more precise long- and short-term adaptation mechanism. We return to this issue later in this work.

MATCHS' modular architecture also allows other, possibly more precise, WM models to be used; as long as a WM model provides a probability of information retention over time, it can be used in MATCHS. The Controller that drives the adaptation could also be replaced by a more refined one. The results presented in this chapter were obtained through a simple proportional gain that made use of the error value to drive the evolution of the estimated quanta. A (possibly ML-derived) more complex control function could be used to also regulate other WM-performance-dependent parameters such as δ_t, resulting in more refined user model.

Going beyond toy use cases such as Match^{2}s, MATCHS could be used to assess the user's capacity for holding information as long as some estimation of how much information was forgotten is provided. As seen in Section 4.1.2, performance on tasks can be used to infer cognitive capacity and cognitive load. Therefore, in cases such as tutoring systems, the error rate or completion time of exercises could serve as an indication of the user's WM capacity. MATCHS could then use this measured performance to adjust the presentation of exercises and the extraneous cognitive load, regulating α_s in a way that the student is not overcharged and has enough available resources left the germane load of learning. However, a big drawback of MATCHS for such as use case, as well as others, is that this framework needs to perform numerous series of simulations in order to obtain the key WM-specific recall curves, which are thus time- and resource-costly components. This issue is addressed in the next chapter.

Chapter 6

An Unscented Hound for Working Memory

"It is sheer folly to take unwilling hounds to the chase."

Titus Maccius Plautus

An Unscented Hound for Working Memory (AUHWM, prononcé "om") est une extension naturelle du cadre MATCHS. Alors que, dans MATCHS, les quanta estimés, ou les ressources cognitives, d'un utilisateur ont été trouvés de manière incrémentielle par le contrôleur entraîné par le signal d'erreur, AUHWM utilise un filtre de Kalman non-linéaire (UKF) pour le suivi en temps réel de la capacité de l'utilisateur.

AUHWM est capable de suivre dynamiquement les capacités cognitives de la WM d'un utilisateur sur des intervalles de temps à court et à long terme. Au contraire des approches typiques de Machine Learning telles que [67, 66, 67], où les modèles de charge cognitive sont dérivés des données, une application fondée sur AUHWM peut être utilisée avec différentes populations, non rencontrées auparavant, sans passer par le processus fastidieux de collecte de données d'entraînement personnalisées. Il le fait parce que AUHWM est fondé sur un modèle bien compris et validé de la WM (par exemple, les modèles présentés dans la section 4.2). De plus, en ayant une compréhension claire de ce que représentent les paramètres de modélisation d'AUHWM, des explications de haut niveau des choix du système peuvent être fournies.

AUHWM consiste en un filtre de Kalman non-linéaire couplé à un modèle déterministe de la dynamique de dégradation de la WM humaine. Il est capable de modéliser la dynamique de la WM d'une personne en suivant des paramètres de personnalisation. Cependant, le but principal de AUHWM est de fournir une adap-

tation de tâches pour les applications qui dépendent de la WM.

Les sections suivantes sont organisées comme suit. Les sections 6.1 et 6.2 présentent au lecteur une revue du filtre de Kalman et du fitre de Kalman étendu. La section 6.3 présente le filtre de Kalman "unscented" (inodore, car neutre) pour les estimations non linéaires. La section 6.4 décrit le cadre de AUHWM pour la modélisation des capacités cognitives, et la section 6.5 discute de la validation des capacités de modélisation d'AUHWM.

An Unscented Hound for Working Memory (AUHWM, pronounced "om") is a natural extension of the MATCHS framework. While in MATCHS, the estimated quanta, or the cognitive resources, of an user were found incrementally by the controller driven by the error signal, AUHWM employs an Unscented Kalman Filter (UKF) to track the user's capacity in real time.

AUHWM is able to dynamically track a user's WM cognitive capabilities over both short- and long-term time intervals. Unlikely typical ML approaches such as [67, 66, 67], where models of cognitive load are derived from data, an AUHWM-based application can be used with different populations, not previously seen, without going through the burdensome process of collecting personalized training data. It does so because AUHWM is based on a well-understood and validated model of WM (for instance, the presented models of Section 4.2). Moreover, by having a clear understanding of what AUHWM's modeling parameters stand for, high-level explanations of the system choices can be provided.

AUHWM consists of an Unscented Kalman Filter coupled with a deterministic model of human WM degradation dynamics. It is able to model a person's WM dynamics by tracking personalization parameters. However, the main purpose of AUHWM is to provide task adaptation for applications that are WM-dependent.

The next sections are organized as follows. Sections 6.1 and 6.2 present the reader with a review of the Kalman Filter and the Extended Kalman filter techniques. Section 6.3 introduces the Unscented Kalman filter for non-linear estimations. Section 6.4 describes AUHWM's framework for modeling cognitive capacities, and Section 6.5 discusses the validation of AUHWM's modeling capabilities.

6.1 Kalman Filter

The Kalman Filter (KF) is an ubiquitous technique for tracking or data-prediction tasks. The KF is useful when compared to other ML techniques because it doesn't require any data or training to be implemented and is a very fast algorithm. The goal of the KF is to derive the best estimates for a discrete-time linear system whose evolution is subject to process noise (or random disturbances). It does so by propagating previous estimations in the form of Gaussian Random Variables

(GRV) through a transition function, generating new estimations, collecting data from sensors and finally updating the system's state belief with the new data [86].

The goal of a KF is provide an estimate $\hat{x}$ of an unknown state vector x. At first, before measurements arrive from the sensors, one needs to make a first estimation of the system's initial GRV state x_0, with known mean μ_0 and covariance P_0. Since no more information is available at this point, the optimal state estimation is given by $(\hat{x}_0, P_0)$, with:

$$\hat{x}_0 = \mu_0 = E[x_0],$$

and the covariance by

$$P_0 = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T].$$

The initial state is thus a GRV where the covariance represents the estimated belief of the estimation.

One also needs to model the systems dynamics by defining a transition function F and an observation function H, assumed in this overview to be independent of time. Both functions have the form of linear vector functions. F describes the linear evolution of the system state, and H defines the relationship between the observation values from the sensors and the system actual state; both linear transforms can be seen as matrices. The transition function models the state evolution as follows:

$$x_t = Fx_{t-1} + w_t, \tag{6.1}$$

where x_t is the current unobserved state vector, x_{t-1} is the previous state and w_t is a random vector representing the process noise (or the uncertainties in the model); w_t is zero-mean and temporally uncorrelated (white noise), i.e., $E[w_t] = 0$. We call W_t the process noise covariance matrix at time t, defined as:

$$W_t = E[w_t w_t^T].$$

The observation function H models the relationship between the states and the measurements:

$$y_t = Hx_t + v_t, \tag{6.2}$$

where y_t is the only observed value from the actual state x_t, and v_t is the random vector of the measurement noise, also zero-mean and temporally uncorrelated ($E[v_t] = 0$). The observation function covariance matrix is then called V_t and defined as:

$$V_t = E[v_t v_t^T].$$

There are two main steps involved in the KF: the model prediction step and the data assimilation step. The model prediction step involves propagating the previous state estimation through the transition function, obtaining a forecast of the state evolution. The data assimilation step uses the information obtained through the sensors to update the forecast state, obtaining the best state estimation possible.

6.1.1 Model Forecast Step

During the model forecast step, the only available information is the previous mean and covariance of the state estimation at time $t-1$. Therefore, using the transition function, one can make some crude estimation of the possible next state $\hat{x}_{t|t-1}$. The mean of the forecasted state is obtained by:

$$\hat{x}_{t|t-1} = F\hat{x}_{t-1},$$

while the covariance of the forecasted state is given by:

$$P_{t|t-1} = FP_{t-1}F^T + W_t.$$

Note the addition to the transformation of the previous state's covariance P_{t-1} of the process noise covariance W_t, corresponding to an increase in the model's uncertainty given by the less-than-optimal modeling of the transition function.

6.1.2 Data Assimilation Step

The data assimilation step is an a posteriori step that uses the information obtained from the sensors to update the forecasted state estimation in order to obtain the optimal estimation $\hat{x}_t$ at current time t. The mean and covariance of the corrected estimation are given by

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(y_t - H\hat{x}_{t|t-1})$$

and

$$P_t = (I - K_tH)P_{t|t-1},$$

where K_t is the so-called Kalman gain, given by

$$K_t = P_{t|t-1}H^T(HP_{t|t-1}H^T + V_t)^{-1}.$$

Note that the evolution of the covariance does not depend on the actual measurements y_t.

When making predictions during the model forecast step, uncertainty is added through the covariance of the transition function W_t, meaning that the Kalman filter "looses" information because of the uncertainties in the model. However, when a new measurement arrives, this new information is added to the forecast. The added information is weighted by the Kalman gain; intuitively, the Kalman gain is the ratio between the confidence of the forecasted prediction and the confidence of the observed value [87].

When K_t is large, more importance is given to the measured signal, meaning that more information is being added by the sensors. Traditionally, one initializes the state covariance matrix P_0 with large diagonal values, meaning that at first we are not sure about the actual state and that K_t is pretty much dominated by $P_{t|t-1}$, so that K_t is giving less importance to V.

Therefore, every time a measurement from the sensor arrives, the state uncertainty is proportionally reduced by $K_t H$, resulting in a smaller state covariance. With a smaller $P_{t|t-1}$, K_t becomes more and more dominated by V_t, that is the uncertainty of the sensors. The Kalman gain reflects then how seriously we should take each measurement into account when updating the predictions.

In the KF formulation, where the estimations are linear Gaussian, when starting with a initial state as a GRV, the filtering process will always produce a GRV, for a linear transformation of a multivariate normal random variable has also a multivariate normal distribution [86]. However, the traditional KF only works for systems with linear dynamics. In order to work with non-linear dynamics, which will be needed given our modeling of the WM dynamics, the KF needs to be modified or extended.

6.2 Extended Kalman Filter

The Extended Kalman Filter (EKF) is a version of the KF that deals with non-linear system dynamics through local linearizations [88]. The EKF employs first-order local linearizations of F and H in the region of $x_t = \mu_t$ of the non-linear system; this is why the EKF is also called a first-order filter. Subsequently, using the traditional linear KF equations, the state distribution, approximated by a GRV, can be propagated analytically through the first-order approximations, obtaining the next estimated distribution.

The EKF assumes the transition and observation models are smooth and well-behaved, meaning that the linear model validity depends on how non-linear the model functions are around the current mean [86]. If F and H are smooth, they can be expanded in Taylor series, and the state distribution GRVs can be prop-

agated through the first-order linearizations analytically. However, there are two drawbacks to the EKF: first, the linearizations can introduce large errors in the true posterior mean and covariance of the GRV [89]; second, the derivation of the Jacobian matrices, needed for the Taylor series expansion, are non trivial in most applications, leading to difficult implementations [90].

6.3 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) is an alternative to the EKF; it is an estimation tool mostly used in non-linear dynamic systems or in probabilistic parameter estimation [89]. Instead of using local linearizations as the EKF does, the UKF works by applying the Unscented Transformation (UT) in order to deal with the non-linear dynamics.

6.3.1 Unscented Transformation

The UT is a method for predicting statistics of random variables undergoing non-linear transformations. It is based on the principle that it is easier to approximate a probability distribution than a non-linear function [90]. The UT relies upon carefully selected "sigma points", i.e., chosen sample points from the prior distribution that captures its characteristics such as the distributions' first two moments (the mean and covariance). These points are then individually propagated through the non-linear transformation, and the transformed set is used to approximate the posterior distribution.

Assume given a random variable x, described by its mean $\hat{x}$ and covariance P_x, with dimension L, and some non-linear function $y = g(x)$. To approximate the statistics of y, we define a set of $2L+1$ weighted points $S = \{(W_i, \mathcal{X}_i)\}_{i=0}^{2L}$, where $\mathcal{X}_i$ are the sigma points and W_i are the corresponding weights. The sigma points $\mathcal{X}_i$ are defined as follows:

$$\begin{aligned}
&\mathcal{X}_0 = \hat{x};\\
&\mathcal{X}_i = \hat{x} + (\sqrt{(L+\lambda)P_x})_i, \qquad i = 1, \dots, L;\\
&\mathcal{X}_i = \hat{x} - (\sqrt{(L+\lambda)P_x})_{i-L}, \qquad i = L+1, \dots, 2L.
\end{aligned} \tag{6.3}$$

where λ is a scaling parameter given by $\lambda = \alpha(L+\kappa) - L$, α corresponds to the spread of the sigma points, κ is a secondary scaling parameter and $(\sqrt{(L+\lambda)P_x})_i$ is the i-th row of the matrix square root. The corresponding weights for the sigma vectors, also coming in pairs $W_i = (W_i^{(m)}, W_i^{(c)})$, are defined as:

$$
\begin{aligned}
W_0^{(m)} &= \frac{\lambda}{(L+\lambda)}; \\
W_0^{(c)} &= \frac{\lambda}{(L+\lambda)} + (1-\alpha^2+\beta); \\
W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, \qquad i = 1, \ldots, 2L.
\end{aligned}
\tag{6.4}
$$

where β is used to incorporate prior knowledge of x's distribution. For Gaussian distributions, the optimal value is $\beta = 2$ [91]. With these sigma points, one can then obtain the transformed points $\mathcal{Y}_i$ by having $\mathcal{X}_i$ go through the non-linear transformation:

$$\mathcal{Y}_i = \sum_{i=0}^{2L} g(\mathcal{X}_i).$$

The mean $\hat{y}$ of the posterior distribution is then approximated as:

$$\hat{y} \approx \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Y}_i, \tag{6.5}$$

and the covariance P_y as:

$$P_y \approx \sum_{i=0}^{2L} W_i^{(c)} [\mathcal{Y}_i - \hat{y}][\mathcal{Y}_i - \hat{y}]^T. \tag{6.6}$$

For Gaussian inputs, UT is accurate to the third order (in Taylor series expansion). For non-Gaussian distributions, the approximated distributions are accurate to the second order; the accuracy of the higher orders will depend on the parameters α and β [89]. Although there are some similarities between the UT and a Monte Carlo method, the UT does not rely upon random sampling, as here only $2L + 1$ deterministically chosen points are required.

6.3.2 UT-based Filtering

The UKF works pretty much as the linear KF described in Section 6.1 does. As in the linear KF, one needs to model the system (a non-linear one) by defining the transition function F, similarly to the one in Eqn. 6.1, and the observation function H, as in Eqn. 6.2 . Also, there is the need of an initial state x_0. However, where in the linear KF the estimated state had the form of a GRV defined only by its mean and covariance $(\hat{x}_t, P_t)$, in the UKF, the random variable (RV) is defined as

the concatenation of the state estimation and noise variables $x_t^a = [x_t^T \quad w_t^T \quad v_t^T]^T$. Therefore, the initial state estimation $\hat{x}_0^a$ is defined as $\hat{x}_0^a = [\hat{x}_0^T \quad 0 \quad 0]^T$, where $\hat{x}_0$ is the mean of the initial state $\hat{x}_0 = E[x_0]$. The covariance of the new augmented RV, P_0^a, is given by:

$$P_0^a = E[(x_0^a - \hat{x}_0^a)(x_0^a - \hat{x}_0^a)^T] = \begin{bmatrix} P_0 & 0 & 0 \\ 0 & W & 0 \\ 0 & 0 & V \end{bmatrix},$$

where P_0 is the covariance of the initial state x_0, W is the process noise covariance and V is the observation function covariance.

Following the linear KF approach, the UKF also has a model forecast step, but instead of having the mean and covariance of the GRV going through the transition function, here sigma points are selected from the previously estimated augmented RV x_{t-1}^a, seen as a sigma matrix $\mathcal{X}_{t-1}^a = [(\mathcal{X}_{t-1}^x)^T \quad (\mathcal{X}_{t-1}^w)^T \quad (\mathcal{X}_{t-1}^v)^T]^T$. These sigma points are calculated following Eqn. 6.3, and they are the ones propagated through the transition model F as follows:

$$\mathcal{X}_{t|t-1}^x = F(\mathcal{X}_{t-1}^x, \mathcal{X}_{t-1}^w).$$

Note the term $\mathcal{X}_{t-1}^w$, corresponding to the process noise. The mean $\hat{x}_{t|t-1}$ and covariance $P_{t|t-1}$ of the predicted state can then be approximated using Eqn. 6.5 and Eqn. 6.6 respectively:

$$\hat{x}_{t|t-1} \approx \sum_{i=0}^{2L} W_i^{(m)} (\mathcal{X}_{t|t-1}^x)_i,$$

where we use $(\mathcal{X}_{t|t-1}^x)_i$ to refer to the i-th propagated sigma point, and

$$P_{t|t-1} \approx \sum_{i=0}^{2L} W_i^{(c)} [(\mathcal{X}_{t|t-1}^x)_i - \hat{x}_{t|t-1}][(\mathcal{X}_{t|t-1}^x)_i - \hat{x}_{t|t-1}]^T.$$

The observation function H is then used to transform the forecasted sigma points $\mathcal{X}_{t|t-1}^x$ into $\mathcal{Y}_{t|t-1}$, which are the projections of the forecasted sigma points into the sensor's plane:

$$\mathcal{Y}_{t|t-1} = H(\mathcal{X}_{t|t-1}^x, \mathcal{X}_{t-1}^v).$$

where, once again, note the term $\mathcal{X}_{t-1}^v$, related to the observation covariance V. Eqn. 6.5 is used again to approximate the forecasted measurement:

$$\hat{y}_{t|t-1} \approx \sum_{i=0}^{2L} W_i^{(m)} (\mathcal{Y}_{t|t-1})_i.$$

Finally, using now the actual observation value y_t from the sensors, the corrected predicted state mean is updated as done in the linear KF:

$$\hat{x}_t = \hat{x}_{t|t-1} + K_t(y_t - \hat{y}_{t|t-1}),$$

and the corrected covariance P_t is given by:

$$P_t = P_{t|t-1} - K_t P_{y_t y_t} K_t^T$$

where K is the Kalman gain, here given by:

$$K_t = P_{x_t y_t} P_{y_t y_t}^{-1},$$

where $P_{y_t y_t}$ and $P_{x_t y_t}$ are helper matrices given by:

$$P_{y_t y_t} = \sum_{i=0}^{2L} W_i^{(c)} [(\mathcal{Y}_{t|t-1})_i - \hat{y}_{t|t-1}][(\mathcal{Y}_{t|t-1})_i - \hat{y}_{t|t-1}]^T;$$

$$P_{x_t y_t} = \sum_{i=0}^{2L} W_i^{(c)} [(\mathcal{X}_{t|t-1})_i - \hat{x}_{t|t-1}][(\mathcal{Y}_{t|t-1})_i - \hat{y}_{t|t-1}]^T.$$

This completes our short presentation of the UKF, which will be the cornerstone of AUHWM, given its ability to handle the non-linear nature of the WM models. The interested reader is invited to look at the seminal papers (e.g., [89]) for more information, but what has been presented here is enough to understand AUHWM.

6.4 An Unscented Hound for Working Memory

AUHWM is our proposed framework capable of tracking in real time a single parameter corresponding to the user's cognitive capacity. AUHWM employs, at its core, the model of a WM dynamics. Concerning the quantic WM model, as seen in Section 5.2.3, there is a strong link between the parameter Q, i.e., the number of quanta, and the other parameters of the model. Also in the ACT-R's context (Section 4.2.1), the W parameter, which stands for the total amount of source activation, is a key factor when modeling individual differences in memory retrieval performance. This means that, in both models, a single parameter, q_t, can serve as an estimation of a person's WM capacity. The parameter q_t corresponds to an individual's cognitive capacity at time t; it could stand for either Suchow's Q or ACT-R's $\mathcal{W}$, both parameters driving the evolution of WM degradation.

When a person is performing a WM-dependent task, her performance is a noisy observation of that person's cognitive capacity. Therefore by employing a UKF for parameter tracking, one can obtain estimations of the user's WM capacity given

noisy observations of WM-dependent performance. Yet, when used for parameter estimation, the UKF requires some slight modeling modifications from what we described above, for state estimation. The estimated state $\hat{x}_t$ becomes the estimated parameter $\hat{q}_t$ to be tracked, modeled as a GRV. Therefore the transition function F for our UKF that defines the evolution of the tracked parameter q_t is thus set as:

$$q_t = q_{t-1} + w_t.$$

In the absence of a more informed time-dependent modeling of users' cognitive capacities, the process noise here correlates to the fluctuations of a person's available cognitive capacity that are bound to happen during the day, given factors such as motivation, attention or fatigue, so that the amount of information that can be stored might increase or decrease. Moreover, a more constant and long-term degradation might also happen, with the onset of neurodegenerative diseases. All those fluctuations on the available cognitive resources are thus driven by the process noise W_t.

The observation function then becomes:

$$y_t = H(q_t, z_t) + v_t, \tag{6.7}$$

where once again v_t is the measurement noise; therefore, it is assumed that y_t is a noisy observation of the parameter q_t, given an application-specific input z_t. The application-dependent input z_t correspond to the task parameters at time t (these will vary according to the user's performance; these changes are linked to the UI adaptation goal that is the main focus of this work). The observation function, therefore, should be able, given the estimated capacity $\hat{q}_t$ and the parameterization of the task z_t, to return an estimation of the performance, $\hat{y}_t$.

Both WM models used here are task-dependent, meaning that some parameters are to be set according to the task (ACT-R's formulation, however, requires the whole symbolic model of the task). From here on, for simplicity reasons we shall consider memory tasks to be purely WM-dependent, as is Match^{2}s, which in an ACT-R's context, can be modeled as shown in Figure 4.4. Therefore, y_t stands for the probability of recall, and using the notation previously introduced for the models presented in Section 4.2, the input z_t corresponds to the task-dependent parameters, i.e., the tuple (k_t, T_t). These parameters are present in both models and are not user-dependent as they relate to the task. For our application, the non-linear observation function $H(q_t, z_t)$, providing an indirect assessment of q_t, is thus the user's recall probability, at time T_t, given an amount of information k_t and available resources q_t; this recall probability is denoted $r_{q_t,k_t}(T_t)$ below.

The recall probability $r_{q_t,k_t}(T_t)$ is derived from the employed WM model. Eqn. 4.3, derived from ACT-R, can be directly used here. However, as described

in Section 5.1.3, given the stochastic nature of Suchow's MDP, a number of simulations are required in order to obtain recall probabilities. Unfortunately, running stochastic simulations is very time consuming; this is not acceptable for our goal of tracking, in real time, the user's cognitive capacity. The selected approach to bypass this problem is described in next section.

6.4.1 Deterministic Simulation of Suchow's WM

To bypass the expensive simulations necessary for the MDP-based quantic model, we propose to implement an approximation of our adaptation of Suchow's model using a gradient-boosting (GB) [92] approach for regression.

Gradient Boosting

Boosting techniques are concerned with ensemble formation through a constructive approach, where new models are added sequentially to the ensemble in order to optimize response accuracy. GB is a ML technique that continuously add, in a sequence, new weak models that maximize correlation with the negative gradient of the loss function [93]. The added models can be chosen from different families of models such as trees, splines and others. The main difference between boosting methods and other ML techniques is that the optimization here is made in the function space.

The goal of supervised learning algorithms is to find an estimation $\hat{f}$ of a function that maps a possibly multi-dimensional input parameter x to its output $y = f(x)$ by minimizing a loss function $\Psi(y, \hat{f})$, given the data $\{x_i, y_i\}_{i=1}^{N}$. The GB function estimate $\hat{f}$ is given by:

$$\hat{f} = \sum_{i=0}^{M} \hat{f}_i(x),$$

where M is the number of iterations used to provide the expected approximation, $\hat{f}_0$ is an initial model and $\{\hat{f}_i\}_{i=1}^{M}$ are the added incremental models, also called "boosts". These models, or "base learners", are parameterized by θ and are noted $h(x, \theta)$. The t-th boost is given by:

$$\hat{f}_t(x) \leftarrow \hat{f}_{t-1}(x) + \rho_t h(x, \theta_t),$$

where ρ_t is a step size. At step t, the parameter θ_t is selected by choosing the parametrization of a predefined base learner $h(x, \theta)$ that produces the increment most parallel to the negative gradient $-g_t$ of the loss function:

$$\theta_t = \arg\min_{\theta} \sum_{i=1}^{N} (-g_t(x_i) - h(x_i, \theta))^2, \tag{6.8}$$

where the gradient g_t is given by:

$$g_t(x) = E_y \left[\frac{\partial \Psi(y, z)}{\partial z} \right]_{z=\hat{f}_{t-1}(x)} .$$

The parametrization is chosen according to the steepest-descent strategy. This is done instead of simply trying to find θ_t that minimizes the loss function

$$\Psi(y, \hat{f}_{t-1}(x) + \rho_t h(x, \theta_t))$$

over the data, because doing so can potentially be very hard [92], and Eqn. 6.8 becomes nothing more than a least-squares function minimization.

Then the last parameter, ρ_t, is found by minimizing the loss function according to the added boost:

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} \Psi(y_i, \hat{f}_{t-1}(x_i) + \rho_t h(x_i, \theta_t)).$$

A GB model of WM dynamics

In order to approximate Suchow's MDP by a GB model for regression, uniform distributions representing each of the key parameters Q, k and T were created. The parameters L and δ_t of Table 4.2 were set in the same fashion as MATCHS' WMS (Section 5.2.2); however σ was set either to 1 and -1 in order to compare two different strategies. If $\sigma = 1$, then the optimal policy (Section 4.2.2) will give preference to information that were better encoded; however, if $\sigma = -1$, then the decision maker will try to maintain the memories with less quanta, meaning that more information will remain in WM for longer. These are two possible strategies a user can employ in games such as Match^{2}s.

With the other parameters set, the recall probability $r_{Q_t,k_t}(T_t)$ has its behavior dictated by the parameters from the distributions. Instead of having the limits of the uniform distributions representing k and T being defined by the possible configurations of the game Match^{2}s (Section 5.2), they were set to broader limits. The maximum number of presented information, k, is set to 10, corresponding to a very challenging cognitive task, and the maximum limit for T is set to $2,500$ ms. The distribution corresponding to Q has a minimum value of 1 and a maximum value of approximately twice the highest final estimated quanta found on MATCHS' results. Thus, using the Python package PyMC3 [94] for probabilistic machine learning and Bayesian statistical modeling, we sampled the following distributions:

$$
\begin{aligned}
Q &\sim \mathcal{U}(0, 120), \\
k &\sim \mathcal{U}(1, 10), \\
T &\sim \mathcal{U}(0, 2500).
\end{aligned}
$$

Using limits for the task parameters that are wider than the ones found in Match^{2}s' possible configurations, together with a wider population of quanta, means that the simulated data corresponds to a broader range of applications as well as user profiles.

From these distributions were sampled $N = 2 \times 10^6$ possible combinations of the parameters $\{Q_i, k_i, T_i\}_{i=1}^N$. These combinations were then used to generate simulation data, using the same principle as the one described in Section 5.1.3, producing two datasets (one for each value of σ) $\{Q_i, k_i, T_i, r_i\}_{i=1}^N$, where r_i is the recall probability outputted by the WMS, and .

Two GB models for regression were learned over the formed datasets, through the GradientBoostingRegressor class in `sklearn` [95], using the default configuration. The resulting gradient-boosting model is an approximate function $\hat{f}$ that maps Q, k and T to an approximation $\hat{f}(Q, k, T)$ of the recall probability $r_{Q,k}(T)$. The accuracy of the model is given by the coefficient of determination R^2. On average, using a 10-fold cross-validation, when $\sigma = 1$, $R^2 = 0.92 \pm 0.00$ and when $\sigma = -1$, $R^2 = 0.93 \pm 0.00$. Since a perfect fit would have yielded a R^2 equal to one, we see that, through this approach, it appears possible to retrieve user- and task-dependent recall performance with good accuracy, without having to go through the large number of expensive stochastic simulations that were used for MATCHS.

Figure 6.1 depicts an example of the evolution of the recall probability over time T obtained using the learned GB models. The left curve depicts the obtained evolution of the retention of $k = 5$ information items presented when $Q = 50$ quanta are available, with the GB model parameterized with $\sigma = 1$, while the right one shows the result obtained with the GB model with $\sigma = -1$. One can observe that when $\sigma = -1$, the policy tends to retain information for a longer period of time, as the negative value ensures that the maintenance mechanism will focus on the least stable information in order to try to keep it in memory as long as possible.

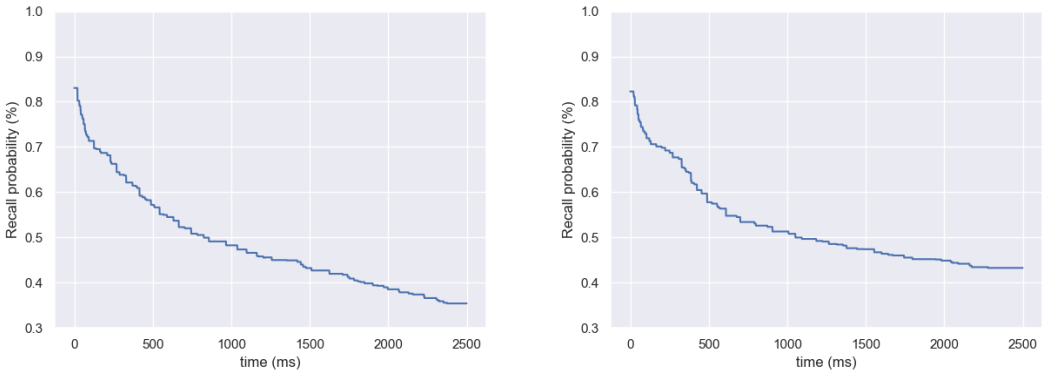


Figure 6.1: Recall probability over time outputted by two learned Gradient Boosting models. Both curves were obtained with the input parameters $Q = 50$ and $k = 5$; however the left one was learned on the data obtained with $\sigma = 1$ and the right one with $\sigma = -1$.

6.4.2 Definition of AUHWM

AUHWM is, at its core, the combination of the UKF[1] and a deterministic model of WM dynamics, as defined by the modeling described at the start of Section 6.4. In this work, one may use the GB approximation of the MDP-based quantic model as the observation model H, Eqn. 6.7 thus becoming:

$$y_t = \hat{f}(q_t, k_t, T_t) + v_t,$$

where y_t is the recall probability observed when the user performs a task parameterized by k_t and T_t. The presence of v_t refers to the fact that the observed probability of recall is a noisy observation of the user's cognitive capacity.

As remarked before, if using an ACT-R-based model instead, one requires a whole symbolic modeling of the task. Remembering that we are interested in Match^{2}s-like applications, then the model depicted in Figure 4.4, and used to generate the recall curves in Figure 4.5, can be employed here. In consequence, the recall probability is thus given by Eqn. 4.3. Therefore, Eqn. 6.7 becomes

$$y_t = \frac{1}{1 + e^{-(A_t - \tau)/s}} + v_t,$$

where one considers there is no difference between the various stored information items, all the information sharing the same recall probability. A_t, the total

[1]The UKF for the estimation of available cognitive resources over time is implemented using the `pykalman` library for Python [96].

activation at time t of one information item, is given by Eqn. 4.1:

$$A_t = B_t + q_t S,$$

where q_t is the tracked parameter representing here the total available source activation. The strength of association S is defined as $S = 1/k_t$ and the base-level activation is given by $B_t = \ln(T_t^{-d})$. The two addends of A_t reflect the division of resources between, respectively, the degradation of each memory after time T_t and the presented information. In this modeling, the parameters τ, s and d were set heuristically to 1, 0.4 and 0.5 respectively.

To summarize, we have introduced here three deterministic models of WM degradation dynamics:

- a GB-based one, with $\sigma = 1$;
- a second GB-based one, with $\sigma = -1$;
- and an ACT-R-based one.

The rest of this chapter is dedicated to the experimental analysis of these various WM models and the evaluation of AUHWM.

6.5 AUHWM Modeling Capabilities

In this Section, we discuss AUHWM modeling performance. In a first time, we shall focus on the framework's performance when employing the GB approximation of the MDP-based quantic model with $\sigma = -1$, as the discussion stands pretty much the same independently of which model is embedded in AUHWM. Then, AUHWM's performance when coupled with the other WM models will be used for comparison and discussion.

6.5.1 GB-based AUHWM Performance

In a first validation step, AUHWM's modeling capability was tested on the data collected using the game Match^{2}s . Remember that during MATCHS' experimental evaluation, every participant played the game for 125 turns, in sequence; the first 5 were used for familiarization with the game's mechanics. For the next 120 turns, for every batch of 20 turns, the number of presented colors k_t as well as the hiding time T_t changed according to the player's performance (the time t stands thus here for the batch number). The player's actual recall probability y_t for each batch is then observationally computed as $n/20$, where n is the number of successful answers for the queried colors. Overall, this resulted in a dataset of six (120/20) data points

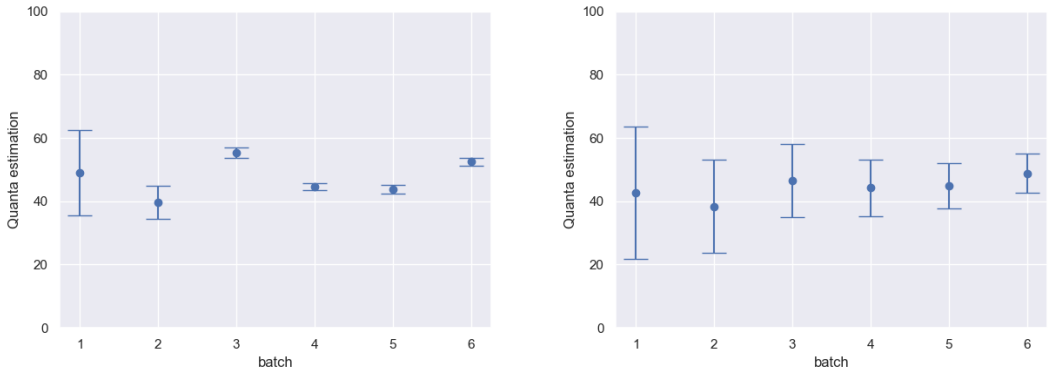


Figure 6.2: Tracked quanta-number values for a typical Match^{2}s player: both estimations were obtained with process noise variance $W = 5$, the left estimations being obtained with observation noise variance $V = 0.001$ and the right one with $V = 0.1$. Note the initial estimation q_0 isn't depicted.

$\{(k_t, T_t), y_t\}_{t=1}^{6}$ for each of the players, which are used to provide an estimate of the WM capacity q_t for each player. Unfortunately, due to technical reasons, the data corresponding to 2 of the 20 players could not be used in this experimental validation.

The UKF requires an initial estimation q_0 (that here represents the number of quanta of Suchow's model) and an initial state covariance matrix P_0 (in AUHWM's case, since the tracked parameter has only one dimension, q_0 and P_0 become respectively the mean and variance of the estimated GRV). The results discussed below were obtained after setting, using an educated guess, q_0 to 40 and P_0 to 1,000.

One also needs to define the process and observation variances matrices, W and V respectively (again, since the tracked variable has only one dimension, W and V become single values). The process variance defines how much uncertainty is added at each iteration, and the observation variance corresponds to how noisy the observations are. AUHWM's outputted estimations will then depend on the definition of these two values.

Figure 6.2 depicts the means and standard deviation bars of UKF-predicted states q_t, as tracked by AUHWM, for one typical Match^{2}s player. These estimations were made with a fixed process noise variance ($W = 5$) and two different values for the observation noise variance ($V = 0.001$ for the left curve, and $V = 0.1$ on the right).

As expected, a less precise sensor (the model with the highest observation noise variance) results in a slower diminution of the initial system's uncertainty P_0. This happens because the Kalman gain is smaller, giving less importance to

Table 6.1: Task parameters used to obtain the recall probabilities in Figure 6.3 .

batch	k	T
1	7	575
2	7	230
3	5	460
4	5	600
5	4	725
6	4	630

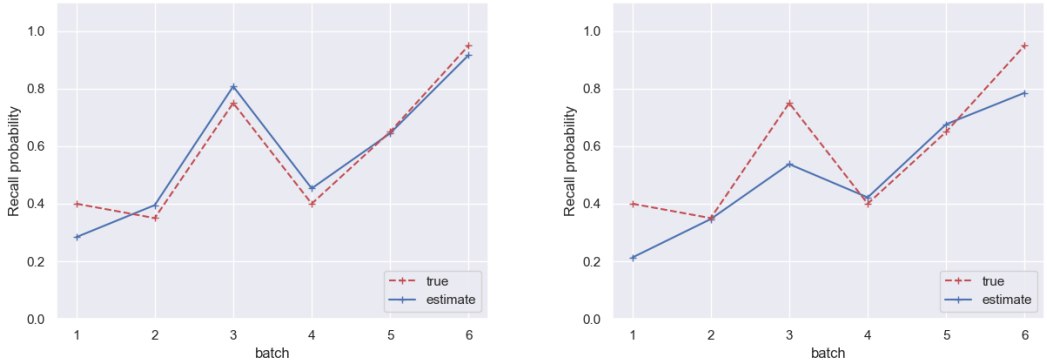


Figure 6.3: Actual vs. estimated recall curves generated by the GB model, using the quanta estimations of Figure 6.2 (left with low V, right with higher V).

the observations value y_t when coming up with the estimation $\hat{q}_t$.

Once equipped with the estimations for each batch $\{\hat{q}_t\}_{t=1}^{6}$ provided by AUHWM, the GB model can be used together with the corresponding task parameters $\{k_t, T_t\}_{t=1}^{6}$ to estimate the recall probabilities each user should present in each batch, remembering that the GB recall probability is given by $\hat{f}(q_t, k_t, T_t)$. Figure 6.3 depicts (dashed line, in red) the evolution of the recall probability the player of Figure 6.2 actually obtained when presented with the 6 different combinations of k_t and T_t in the game as well as the estimated recall probability $\hat{f}(q_t, k_t, T_t)$ (continuous line, in blue) given by the GB model[2]. The 6 task parameters are shown in Table 6.1.

[2]Throughout this work, the recall probabilities are depicted with dashed lines connecting the recall probability of each batch, although connecting the probabilities does not make sense because there is no probability referring to, for instance batch "1.5"; the presence of the connecting lines facilitates comparing the results, though.

Having a smaller observation noise variance correspond to having a more precise sensor. In AUHWM's case, it means that the GB model provides a very good indication of the user's cognitive capacity. As seen in Figure 6.3, the estimations made with the smaller V more closely resemble the actual player recall probability.

The goal of modeling users with AUHWM is to obtain estimations of the user's cognitive capacity at any time. Since no information is available corresponding to the attention level, or any other factor that might influence a person's performance in a WM-dependent task, having a very small observation variance will result in estimations that "overfit" the observed player's performance, meaning that these modulation factors will be encoded in the estimations of the cognitive resources.

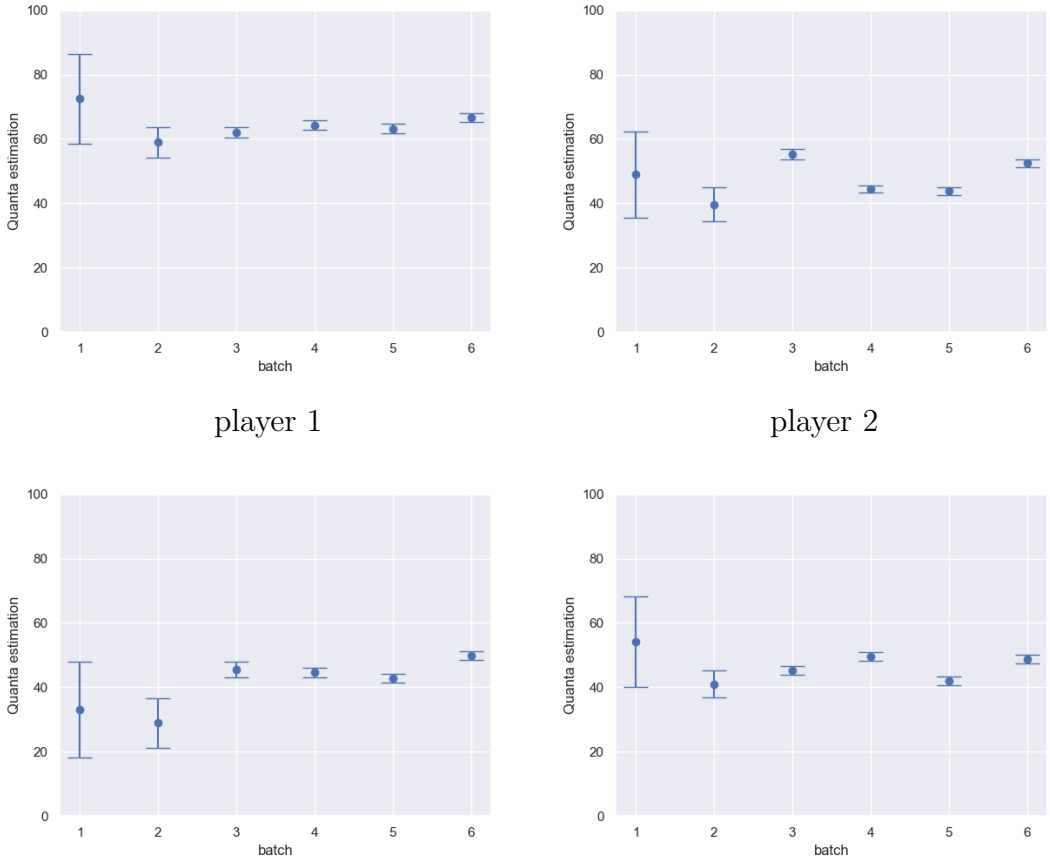
AUHWM's parameters W and V were then set to 5 and 0.001 respectively. The obtained cognitive capacity estimations for the first 4 players are depicted in Figure 6.4. Due to natural individual differences in WM capacity, each player's performance over the batches was different. Therefore, since here q_t is tracking very closely the observed performance, the sets of estimations are different from one another. One can observe that the small value of the observation noise variance results in having the estimation's uncertainty drop after the two first system iterations, indicating the fast convergence induced by the Kalman gain.

Using these estimations, we can derive the estimated recall probability using the GB model, the true (red dashed lines) and the estimated recall probabilities of the same 4 players are shown in Figure 6.5 (the whole collection of all player's estimations and estimated recall probabilities can be accessed at `http://cri.ensmp.fr/auhwm/`). It is clear that after the first two batches, AUHWM's estimations are closer to the true recall values.

To provide a global assessment of AUHWM ability to tract the users' cognitive capacity, Figure 6.6 shows the evolution of the root-mean-square error (RMSE) between the true and estimated recall probabilities of all the 18 players per batch. The last three batch estimations present a mean RMSE error of approximately 4% (the last three estimations were chosen to assure the convergence brought by the Kalman gain), therefore showing that after the initial batches, AUHWM is correctly assessing the number of quanta (when using the GB model based on Suchow's formulation) that corresponds to the player performance. This suggests that AUHWM is tracking reliably the players' cognitive capacity, thus providing additional support for its validity.

6.5.2 Comparison with other WM models

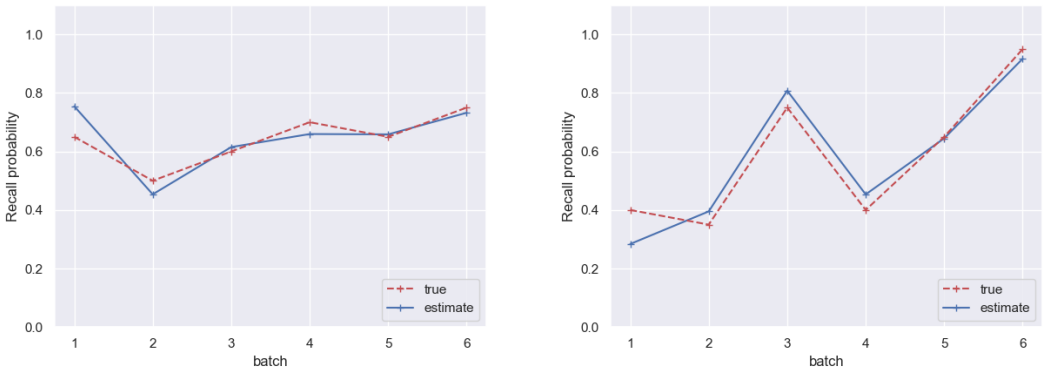
Last section discussed AUHWM's performance when embedded with the GB approximation of Suchow's model when the sensibility of the decision maker σ was set to -1. When embedding AUHWM with the ACT-R-based WM model, one obtains a very similar modeling behavior as the one presented in the previous



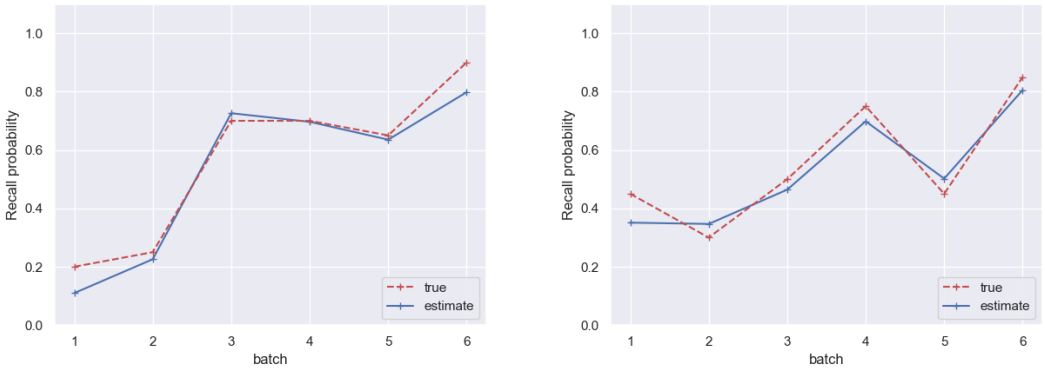
player 1 player 2

player 3 player 4

Figure 6.4: The means and standard deviation bars of AUHWM-predicted states $\hat{q}_t$ of 4 of the 18 Match^{2}s players when employing the GB-based WM model with $\sigma = -1$.



player 1 player 2



player 3 player 4

Figure 6.5: True recall probabilities (in red) and AUHWM-estimated performance (in blue, using the estimated quanta of Figure 6.4) of 4 of the 18 players.

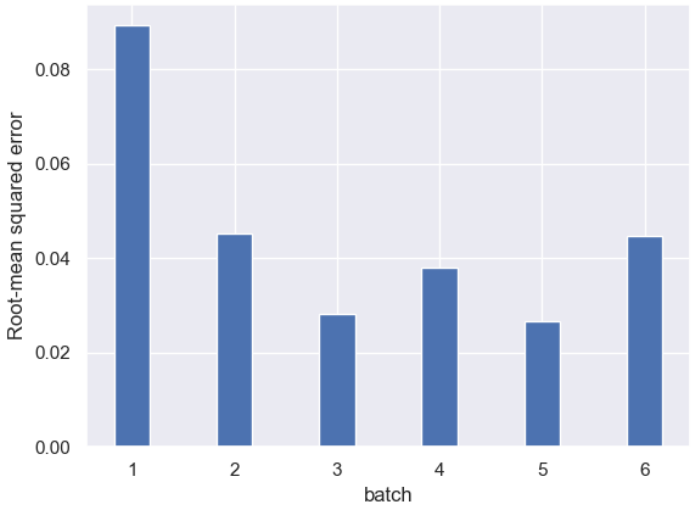


Figure 6.6: Evolution of the RMSE for the recall probability of all the 18 Match[2]s players, per batch.

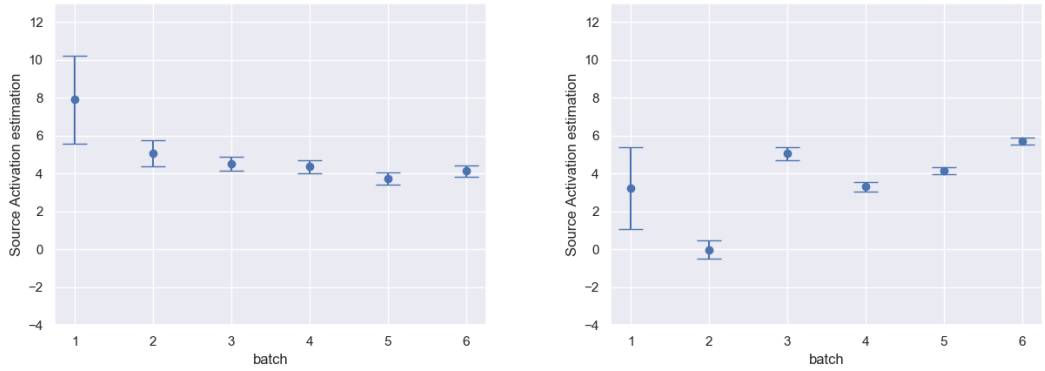
section.

With this embedding, AUHWM was initialized with $q_0 = 1$ and $P_0 = 25$, remembering here that when using the ACT-R's model, q_t stands for the total available resources at time t ($\mathcal{W}$ from Table 4.1[3]). In order to add, at every iteration, the same proportional amount of uncertainty as with the GB-model, we found the factor of proportionality $P_0/W = 0.005$ (for the GB-embedded parameterization), therefore setting $W = 0.125$. The observation noise variance value stayed the same: $V = 0.001$.

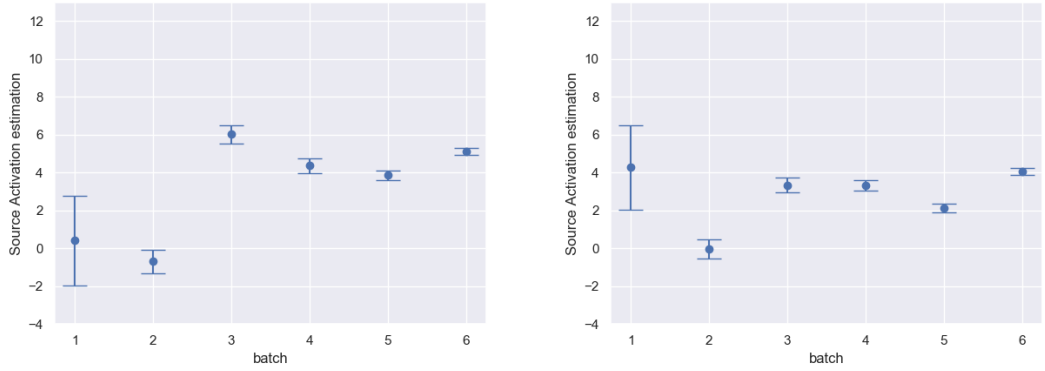
When employing the data of the players depicted in Figure 6.4, one obtains the estimations shown in Figure 6.7. Although the values are different, the presented estimations follow almost the same pattern as the one obtained employing the GB model. Then, using these estimations with Eqn. 4.1 and Eqn. 4.3, one can obtain the information recall probability, shown in Figure 6.8. Once again AUHWM is correctly tracking q_t in order to obtain the right recall probability.

Another similar behavior is found when embedding AUHWM with the GB model with $\sigma = 1$ and initializing it with the same values of process and observation noise variances as in the last section. AUHWM continuously finds the estimated state that, when propagated through the observation model, results in a close approximation of the recall probability. The obtained estimations using the data of the four players and the recall curves obtained from these estimations are presented

[3]This variable should not to be confused with the process noise variance, noted W.



player 1 player 2



player 3 player 4

Figure 6.7: Means and standard deviation bars of AUHWM-predicted states $\hat{q}_t$ of 4 of the 18 Match^{2}s players, when AUHWM is embedded with the ACT-R-based WM model.

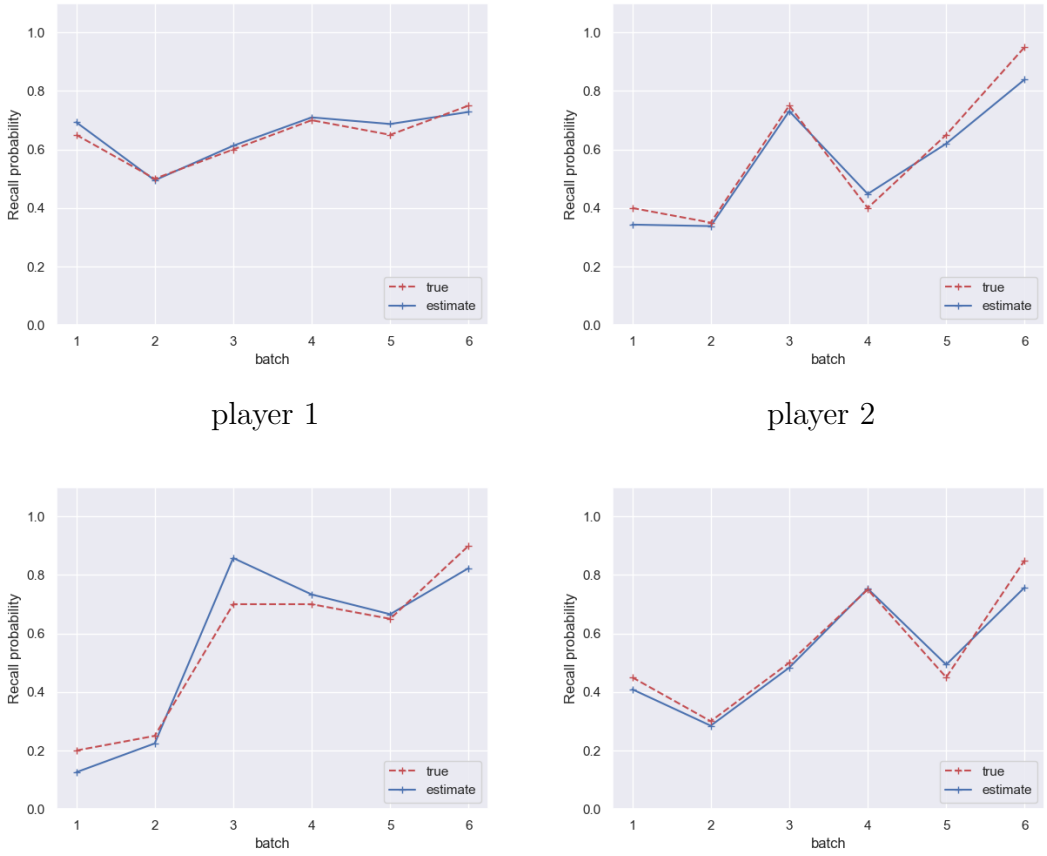


Figure 6.8: True recall probabilities (in red) and AUHWM-estimated performance (in blue) using the source activation estimations of Figure 6.7 of 4 of the 18 players (ACT-R model embedded).

Table 6.2: Mean RMSE of the last three batches for AUHWM embedded with the three WM models.

GB-based, with $\sigma = 1$	0.05
GB-based, with $\sigma = -1$	0.04
ACT-R-based	0.06

in Figure 6.9 and Figure 6.10 respectively.

The evolution of the RMSE with all these three WM models, representing the difference between the true and estimated recall probabilities of all 18 players per batch, is shown in Figure 6.11. One can see that overall, after the first two batches, when employing the GB-based model with $\sigma = -1$, the RMSE is marginally smaller. The mean value for the RMSE during the last three batches for all models is shown in Table 6.2.

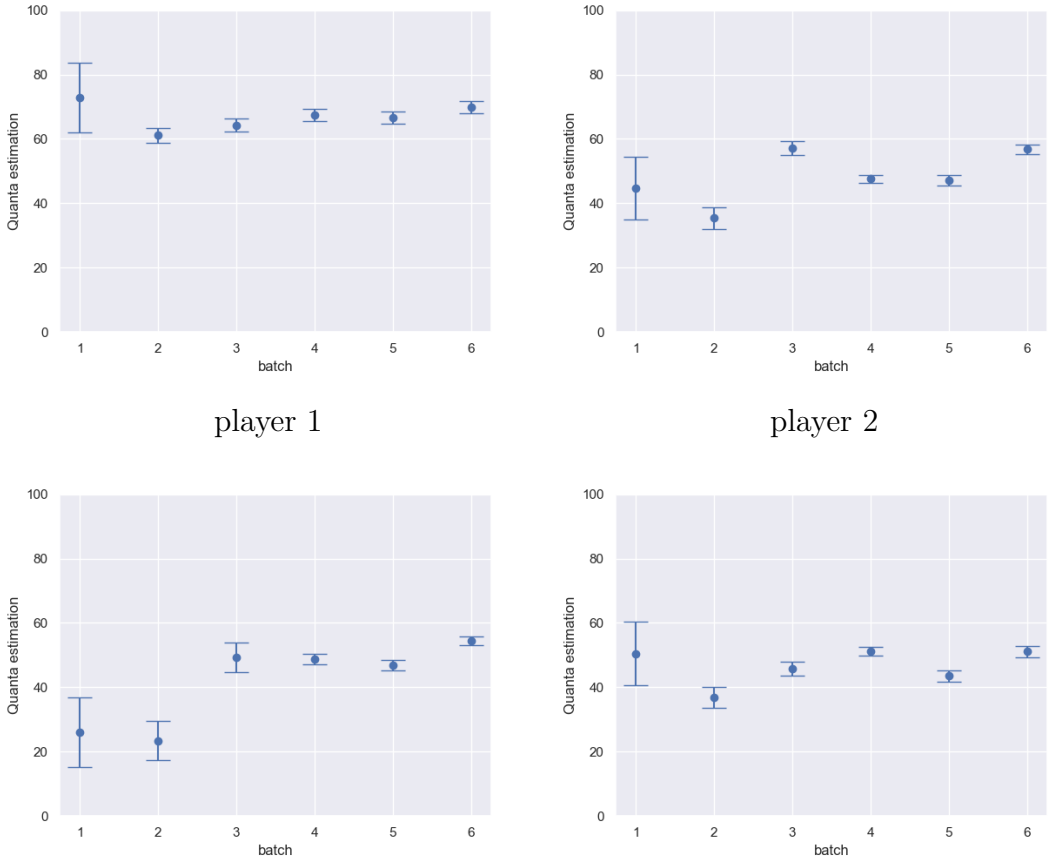
The cognitive capacity estimations and the corresponding estimated recall curve for all the players when embedding AUHWM with the three different WM models can be accessed at `http://cri.ensmp.fr/auhwm/`.

6.6 Discussion

This chapter presented AUHWM, a framework that, by employing the Unscented Kalman filter together with a well-specified model of WM dynamics, is capable of tracking the user's cognitive capacity. The experimental evidence presented in the last section suggests that AUHWM enables the real-time tracking of a person's cognitive capacity when observing his/her performance on a task.

The results obtained when employing three different models show that AUHWM's performance does not depend on the WM model used, as it is able to make estimations of the cognitive capacity (either quanta or source activation) necessary for, given the task parameters, mimicking the observed performance.

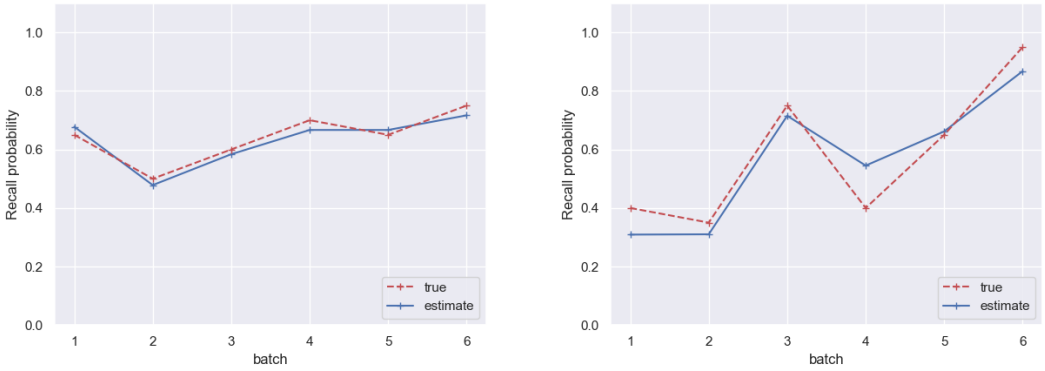
Although during this chapter we have been referring to the tracked parameter q_t as the estimation of a person's cognitive capacity, one can also regard it as a direct assessment of cognitive load. The data used for validation in Section 6.5 was WM-based, meaning that the player's performance is a noisy observation of her capacity. However, when considering the dual task paradigm, if another task is added, resulting in more cognitive charges being applied to the user, then the tracked parameter q_t will reflect the decay of available resources. Therefore, AUHWM can be used for the quantitative determination of the *load* a given task induces on someone's abilities.



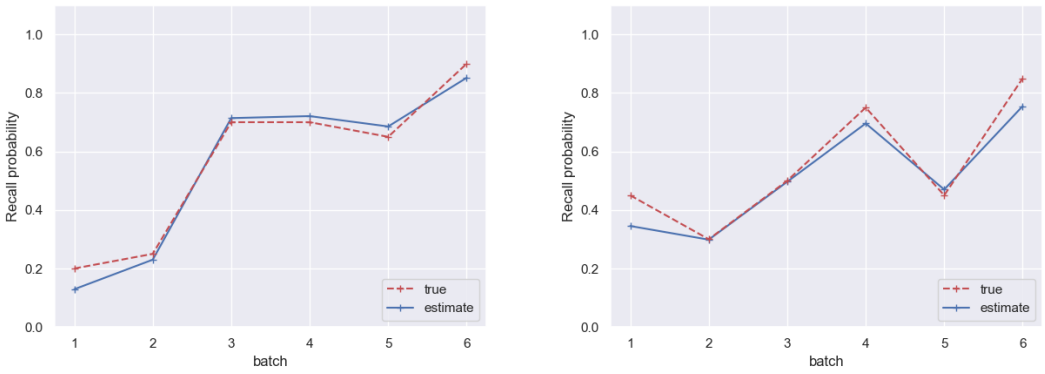
player 1 player 2

player 3 player 4

Figure 6.9: Means and standard deviation bars of AUHWM-predicted states $\hat{q}_t$ of 4 of the 18 Match^{2}s players, when AUHWM is embedded with the GB-based WM model with $\sigma = 1$.



player 1 player 2



player 3 player 4

Figure 6.10: True recall probabilities (in red) and AUHWM-estimated performance (in blue, using the estimated data of Figure 6.9) of 4 of the 18 players. (GB-based model with $\sigma = 1$)

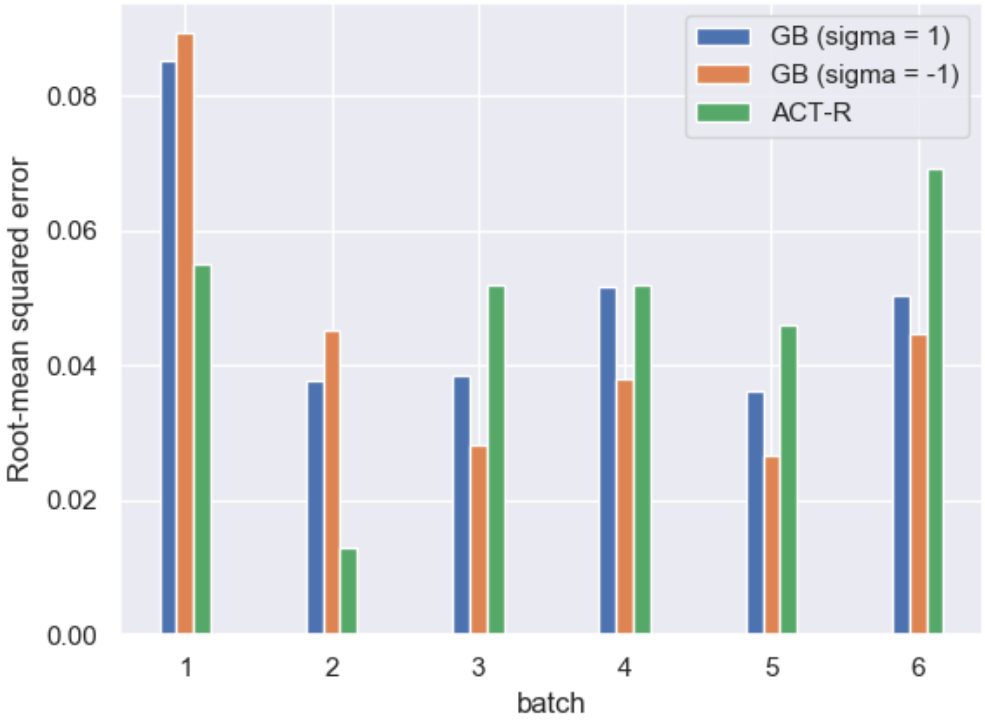


Figure 6.11: Evolution of the RMSE for the recall probability of all the 18 Match^{2}s players, per batch, when employing the 3 different models of WM.

Another direct application of AUHWM's modeling capabilities is the assessment of a user's cognitive decay over time. This can be used for determining a person's cognitive profile as days go by, allowing intelligent systems to consider the user's capacity when adapting task scheduling (which is one of the four mechanisms for adaptation [70]). Applications of such profiles could also be found in the domain of health, for the analysis of demented patients, for instance.

However, as mentioned in the beginning of this chapter, AUHWM's estimations are mostly intended to be used in adapting task-specific UIs to users' cognitive limitations in real time. In this chapter, the modeling task is done a posteriori; AUHWM finds the user capacity after interaction was performed, corresponding to the first step for adaptation defined in Section 2.1. However, if one is interested in adaptation, one has to identify the impact the assessed cognitive capacity has on performance, as well as being able to select a compensatory strategy (step 2 for adaptation). This is where the advantage of embedding our framework with well-specified and validated WM models stands out. The embedded model is supposed to provide the adaptation framework with information necessary to be able to deal with unseen data, allowing it to make predictions on performance a priori, i.e., before the interaction has taken place. Next chapter will discuss a AUHWM-based framework for real-time UI adaptation and evaluate it.

Chapter 7

UI Adaptation using AUHWM

Le chapitre précédent a présenté notre cadre de modélisation et de suivi de la WM, AUHWM, et a discuté de ses capacités. Cependant, comme indiqué précédemment, le principal objectif de AUHWM est d'adapter les interfaces utilisateur aux limitations cognitives. La prise en compte des limitations cognitives à long et/ou court terme lors de l'adaptation des interfaces utilisateur est un moyen d'augmenter les chances qu'une tâche donnée soit exécutée correctement. De plus, même si la tâche a été effectuée sans surcharge cognitive, permettre à l'utilisateur de la réaliser tout en considérant toutes les informations présentées peut entraîner moins d'erreurs humaines.

En se rappelant les quatre étapes principales de l'adaptation vues précédemment, à savoir

1. *constater la limitation de capacité de l'utilisateur,*
2. *identifier l'impact potentiel sur la performance,*
3. *sélectionner une stratégie compensatoire et*
4. *appliquer cette stratégie dans le contexte actuel,*

nous voyons que le chapitre précédent concernait la première étape, fournissant la preuve que AUHWM est capable d'inférer les limites de la capacité cognitive de l'utilisateur à partir des interactions. Ce nouveau chapitre se concentrera donc sur la façon dont ces limitations peuvent être utilisées pour les étapes 2 et 3.

The last chapter introduced our framework for WM modeling and tracking, AUHWM, and discussed its capabilities. However, as said before, the main ultimate goal of AUHWM is to adapt user interfaces to cognitive limitations. Considering long- and/or short- term cognitive limitations when adapting UIs is a way to increase the chances that a given task is going to be performed. In addition, even if

the task was performed without cognitive overload, enabling the user to complete it while considering all the presented information may result in less human errors.

Remembering the four main steps for adaptation seen before, namely

1. the detection of a user's capacity limitation,
2. the identification of its potential impact on performance,
3. the selection of a compensatory strategy,
4. and the implementation of this strategy in the terms of the current context,

we see that the last chapter was concerned with the first step, providing evidence that AUHWM is capable of inferring the limits of the user's cognitive capacity through interaction. This new chapter, thus, will focus on how these limitations can be used for steps 2 and 3.

AUHWM consists on a tracking algorithm (UKF) coupled to any deterministic modeling of human WM, as the Kalman observation function H. The embedded model defines the link between capacity q_t, task parameter z_t and a noisy observation of performance y_t as, recalling Eqn. 6.7:

$$y_t = H(q_t, z_t) + v_t.$$

An AUHWM-based system is able to extract information from the model, obtaining a numerical measure of cognitive capacity from the task parameters and observed performance. However, having a causal model of WM allows a system to go beyond simple tracking. Provided with the tracked cognitive capacity, the model, serving as a link between the three parameters, can, in addition, be used to predict performance, given the task parameters (adaptation step 2). Moreover, this information can be used to find the specific task parameters necessary for attaining a specific performance (step 3). Next section will present the reader a framework capable of doing so.

7.1 An AUHWM-based Framework for Cognitive UI Adaptation

Figure 7.1 depicts a possible AUHWM-based framework capable of adapting an UI in real time to the user's perceived cognitive capacity. There are some clear similarities between the framework depicted above and the MATCHS' one of Figure 5.1.1. Much as in MATCHS, a task-specific parameter π_s must be set by the task manager; this parameter corresponds to the desired performance one wants the user to have when performing the adapted task. However, while in MATCHS

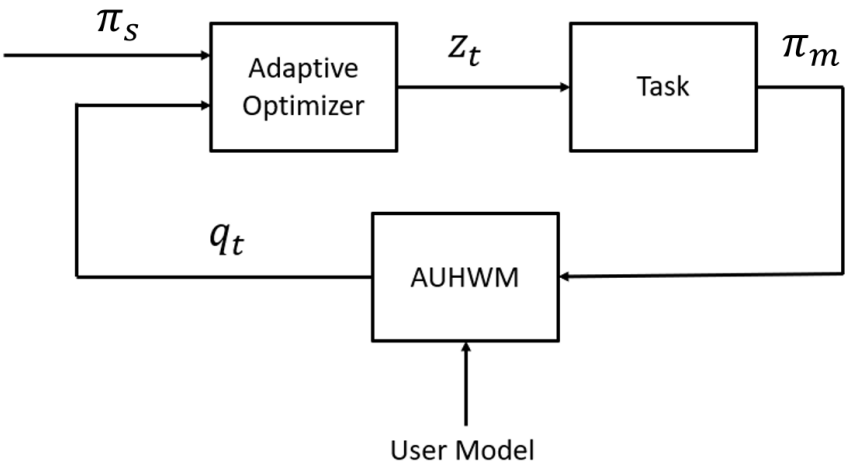


Figure 7.1: AUHWM-based UI adaptation of a task to users' cognitive capabilities (see Table 7.1 for a description of the parameters used).

Table 7.1: Adaptation parameters

π_s	Desired accuracy
π_m	Measured accuracy
z_t	Task parameters
q_t	Estimated capacity

the α_s parameter corresponded to the allowed amount of information to be forgotten, here, when concerning purely WM-dependent tasks, π_s corresponds to how much information has to be remembered: $\pi_s = (1 - \alpha_s)$. For instance, in contexts such as Match^{2}s, π_s corresponds to the probability of recall; if set to 1, one wishes the user to get a perfect score in Match^{2}s; if set to 0.5, the user global performance would show 50% successful answers, on average.

At a given time step t, the Adaptive Optimizer (see Figure 7.1) is responsible for finding the task parameters z_t that will ensure that, on average, the user will perform with performance π_s, given the previous estimation q_{t-1}. The Adaptive Optimizer finds a proper z_t by employing the embedded WM model. Supplied with the previous estimation, the model provides the flexibility to estimate performance even in the presence of previously unseen user's cognitive characteristics. The task parameter could be found, for instance, by searching the task-parameter space for the combination of values that, when propagated through the WM model would

result in the desired performance (or its closest approximation). Once again, in the context of Match^{2}s, z_t would correspond to (k_t, T_t), that is, the number of information items presented as well as the duration of time during which the player has to hold this information in his/her WM. These optimized parameters are therefore the ones that ensure the constraint $H(q_{t-1}, z_t) = \pi_s$. Once given z_t, the task is adapted accordingly and presents its possibly updated UI to the user. The user's measured performance π_m is then used by AUHWM to estimate the next state q_t, corresponding to the updated assessment of the user's cognitive capacity.

Moving beyond the context of Match^{2}s-like games, z_t could correspond to the period of time before the UI refreshes a previously presented information, ensuring the user will be able to perform a task without forgetting more than $(1 - \pi_s)$ of the information content. In the context of decision-making processes, the UI could make sure that the user is solving a problem while considering all the essential information. For assistive technologies, z_t could stand for the number of presented information items; this would enable patients suffering from Alzheimer's disease, or other cognitive deficits, to interact with the adapted UI autonomously, without the help of family or caregivers, restoring some of their lost autonomy.

7.2 Performance Prediction

In order to validate the framework described in the last section, we applied it on the same data used to validate MATCHS and AUHWM's modeling capability. In order to do so, instead of having the Adaptive Optimizer find the optimal task parameters z_t that would result in the performance π_s, we assume that T_t and k_t, i.e., the hiding time and number of squares presented at batch t, are already the optimal parameters. Therefore, if AUHWM works correctly in assessing the user's cognitive capacity, the user's measured performance π_m must be equal to what should have been π_s. This corresponds to having a perfect Adaptive Optimizer that, even when presented with faulty estimations, finds the optimal task parameters. Consequently, the previously estimation output by AUHWM, q_{t-1}, together with the corresponding task parameters T_t and k_t, when ran through our model, should result in π_m. In practice, this means running the same test as the one of Section 6.5, but while "looking ahead", that is, using the state q_{t-1} to predict the recall probability of the next batch, t, by using the embedded model $H(q_{t-1}, z_t)$.

Figure 7.2 shows the histogram representing the configuration of task parameters for all the 18 players, corresponding to the data used here. These parameters where regulated by MATCHS, which means that they variated accordingly to the player's performance. Note that all the players started with more or less the same configuration ($k = 7$ and $T \approx 500ms$).

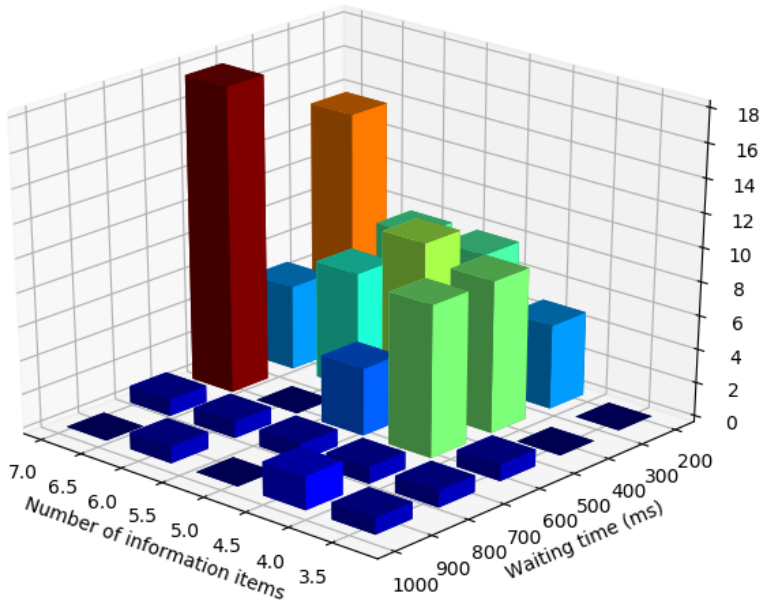


Figure 7.2: Configurations of the parameters in the data collected with the memory game Match^{2}s.

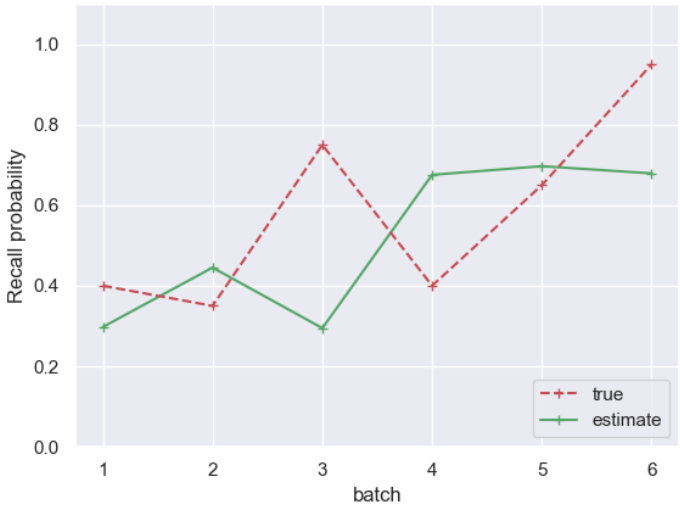


Figure 7.3: Actual vs. estimated recall curves generated by the GB model, using the last estimations q_{t-1} to predict the player performance at batch t. The estimations used here are the ones of the same player as in Figure 6.2

In a first validation step, we will focus on the framework's performance when employing the GB model with $\sigma = -1$, for, once again, the discussion holds up independently of the embedded model. AUHWM was initialized with an initial state $q_0 \sim \mathcal{N}(40, 32)$, and ran to obtain the players' cognitive capacity estimates according to the measured performance π_m. However, parameterizing AUHWM with the same values for the process and observation noise variances as the values used in Section 6.5 results in a less than optimal prediction performance. Figure 7.3 depicts the predicted estimated recall probabilities using the quanta estimations depicted in Figure 6.2 (left estimations) given by $\{H(q_{t-1}, z_t)\}_{t=1}^{6}$, which means that the first estimation is made with the initial quanta guess q_0.

One can see that the estimations get closer to the true value. However, they do not follow the true value as closely as when modeling the user (left curve of Figure 6.3). The performance deterioration becomes clear when we compare the evolution of the RMSE of all the players per batch obtained in the two contexts, modeling and prediction as shown in Figure 7.4. While, when AUHWM was used for modeling the user's cognitive capacity, the last three batch estimations presented a mean RMSE error of approximately 4% (Figure 6.6), when predicting performance, the last three batch estimations present a mean RMSE of 15%.

This degradation is due to the fact that there are a number of factors that result in performance variation other than cognitive capacity. For instance, from one batch to the next, some players lost motivation, due to the task being repetitive,

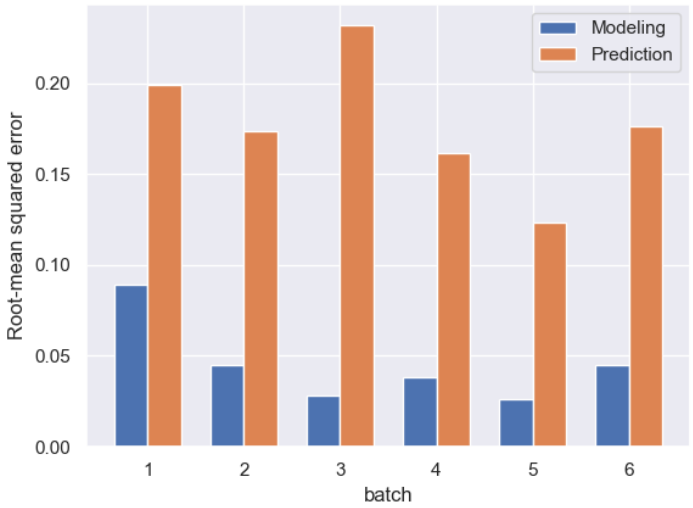


Figure 7.4: Comparison of the evolution of the recall probability RMSE for the 18 Match^{2}s players, per batch t, between modeling and "looking ahead" (prediction based on q_{t-1}).

and therefore became less attentive. Moreover, some players started developing better strategies, or became used to Match^{2}s, resulting in better scores and therefore apparent higher cognitive capacities. And since here the UKF is tracking very closely the observed performance ($V = 0.001$), these local fluctuations result in local changes on the estimated cognitive capacity, meaning that AUHWM's estimations end up encoding these modulating parameters. And since these factors fluctuate from batch to batch, using the previous estimation to predict next batch's performance worsens the system's accuracy. Therefore, the parameterization of AUHWM using the previous batch configuration, which "overfit" the observed player's performance, is not optimal when the objective is to predict next batch's performance.

7.3 AUHWM Prediction Optimization

As seen before, the close tracking of a player's performance results in a less than optimal behavior. With a better parameter fit, the UKF could be expected to filter out the player's performance fluctuations, obtaining smoother quanta estimations that would, on average, result in better predictions.

In order to find AUHWM's optimal parameterization (as before, we focus first on the GB-based WM model with $\sigma = -1$), a technique similar to the "minimizing

the residual prediction error" method described in [97] was employed. However, the method in [97] requires a "highly accurate instrument for measuring either all or a subset of the variables in the state x_t" to compute the prediction error. We, on the other hand, do not have the luxury of an instrument that would give us a precise measure of the user's cognitive capacitye, which means we cannot proceed as in [97] to try to minimize the error between the actual states and the estimated ones. Still, we do have a model that links the cognitive capacity and the recall probability. Therefore, the solution employed was to search the full parameter space for the combination of process and observation noise variances (W, V) that minimizes the RMSE between the predicted and true recall probabilities in the last three batches:

$$(W, V) = \arg\min_{W,V} \frac{1}{N} \sum_{p=1}^{N} (\frac{1}{|B|} \sum_{t \in B} [y_{p,t} - H(q_{p,t-1}, (k_{p,t}, T_{p,t}))]^2)^{1/2}, \tag{7.1}$$

where N is the number of players, and B the set of batches selected for this optimization phase (in our case, the last three batches). H is the (W, V)-parameterized UKF's observation function, linked to the WM model considered. Here only the last three batches are being considered. Note that we do not require that AUHWM gives accurate estimates of its uncertainty, since P_t is not being taken in consideration.

A first uniform discretization of the parameter space $W \times V$ was made. The dimension referring to the process noise variance was quantized in 10 intervals, between 1 and 50, and the dimension corresponding to V was quantized also in 10 intervals between 0.001 and 0.5, therefore creating a parameter space of 100 combinations of values of W and V. Intuitively, the process noise corresponds to the uncertainty being added on the plane of cognitive capacity estimations. Since this section is concerned with the quantic model (as we are embedding AUHWM with the GB model), then the process noise adds uncertainty in the dimension of numbers of quanta.

In order to illustrate the model's sensitivity to the tracked parameter, Figure 7.5 depicts the recall gain for a set task parameter z when q quanta are added to different base values q_{base}; this gain is defined as $H(q_{base} + q, z) - H(q_{base}, z)$. Logically, the smaller the added quanta, the less significant the increase in recall probability. Therefore, a variance of 1 signifies that not much uncertainty is being added from one step to another, since the GB model is pretty much insensible to variations of one quantum. However, a process noise as large as 50 quanta means that AUHWM is highly uncertain about how the user's capacity might evolve from one batch to the next, since a difference of 50 quanta is very expressive in terms of the GB model's output.

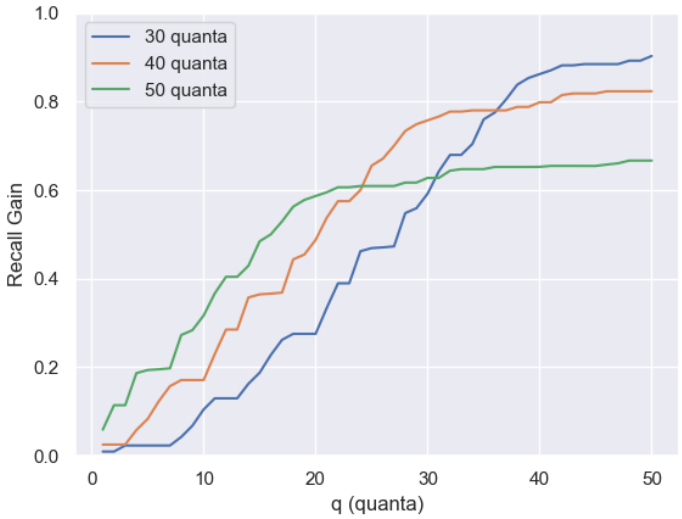


Figure 7.5: Evolution of the gain $H(q_{base}+q, z)-H(q_{base}, z)$ in recall when q quanta are added to different base values q_{base} (30, 40 and 50). The task parameters $z = (k, T)$ was set to $k = 6$ and $T = 1500$.

On the other hand, the role of the observation noise variance V is related to the observation's plane sensor. Since the observations correspond to recall probabilities, when $V = 0.001$, AUHWM is pretty certain the amount of available quanta will result in the observed recall probability y_t, while $V = 0.5$ corresponds to a quite faulty sensor, since the total range of observed values is $[0, 1]$.

Each one of the 100 combinations of W and V was used to configure AUHWM and to track the WM capacity of the 18 players, resulting in 6 quanta estimations (one for every batch) for each of the 18 players, similar to the estimation curves shown in Figure 6.2. The estimated quanta were used, together with the corresponding batch parameters k_t and T_t, to predict estimated recall probability curves with the WM model $H(q_{t-1}, (k_t, T_t))$. Then the RMSE between the predicted recall probabilities and the true values of the last three batches was computed.

Figure 7.6 depicts the mean RMSE of recall probabilites obtained for the different combinations of W and V for all the players. Once the value of the observation noise variance exceeds 0.223, the rooted mean square error becomes constant for a given fixed process noise covariance value. Zooming in the region with the smallest RMSE value, the space was uniformly discretized once again in ten values between 1 and 6.4 for W and 0.001 and 0.056 for V. The obtained RMSE values are depicted in Figure 7.7

The minimum RMSE is found when W was set to 1 and V to 0.025. With such a parameterization, AUHWM obtains the estimations depicted in Figure 7.8 for

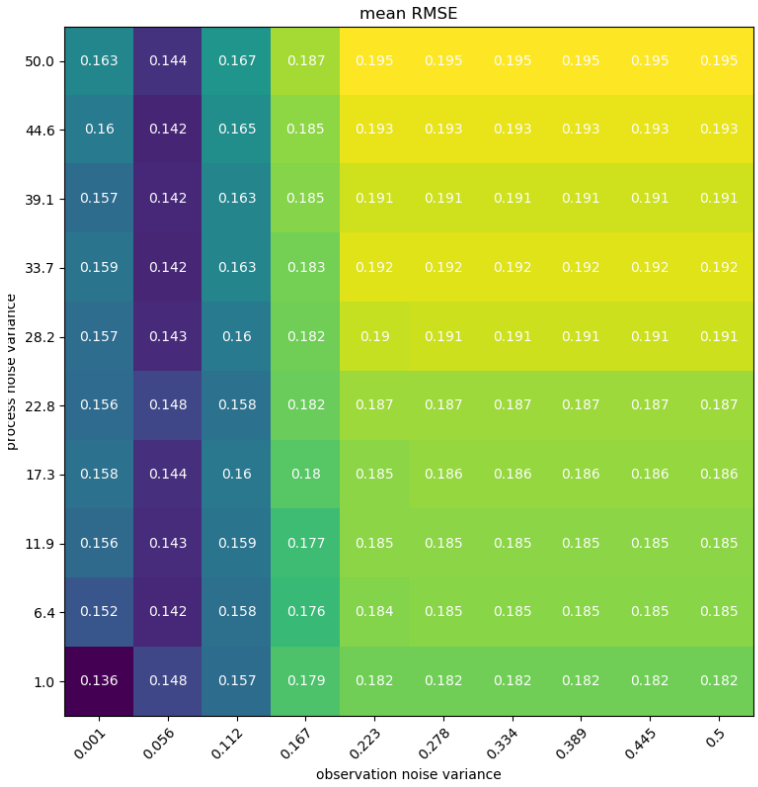


Figure 7.6: Mean RMSE between the actual recall probabilities of the players and AUHWM's predicted recall probabilities, obtained with different configurations of process and observation noise variances.

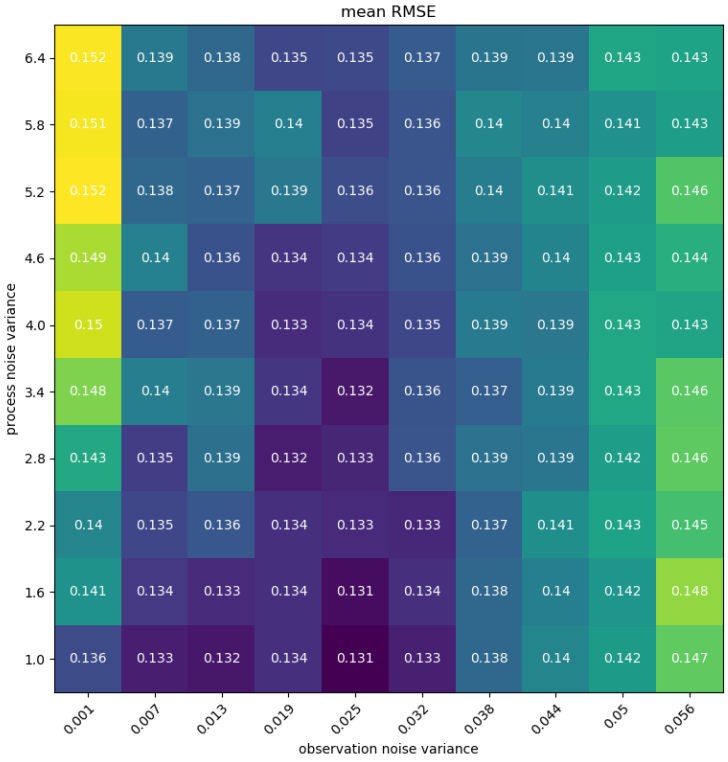


Figure 7.7: Detail of the mean RMSE between the true recall probabilities of the players and AUHWM's predicted ones.

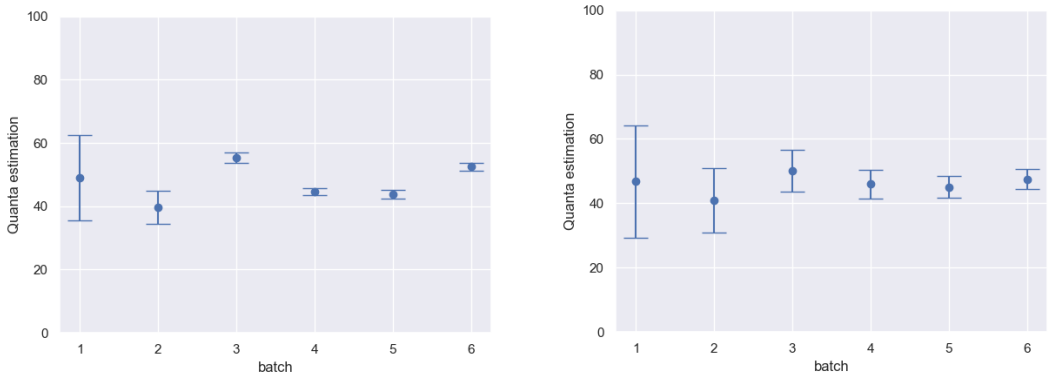


Figure 7.8: Tracked Q values for a typical Match^{2}s players.The left estimations were obtained with process and observation noise variances $W = 5$ and $V = 0.001$ respectively, while the right estimates were obtained with the optimized parameterization $W = 1$ and $V = 0.025$. Note that the right estimations are less precise (wider standard deviation bars) as they do not "overfit" the observed performance as before, due to the higher value for V.

the player of Figure 6.2. When comparing to the estimations made with $W = 5$ and $V = 0.001$ (left of figure), it is clear that, in the "minimum" configuration, the outputted estimations vary less from batch to batch. This is due to the higher observation noise variance, meaning that the UKF is giving less importance to the measurements y_t. Therefore, AUHWM is no longer "overfitting" the observed performance. This results in a smoother evolution of the estimated quanta number q_t; therefore when trying to predict next batch performance, AUHWM will employ an estimation not too far from the general cognitive capacity the player has been displaying up to now, ignoring local fluctuations.

Also one can see that the estimations are less precise than before, as P_t doesn't converge as fast. However, note that Eqn. 7.1 measures the quality of the state q_t estimates outputted by the UKF, and is not concerned with providing precise estimations regarding uncertainty.

Note also that if one were to intersect all the estimations intervals on the right curve, one would obtain a non null intersection. This intersection could possibly correspond to the user's true "base capacity", ignoring fluctuations in performance, which is something that cannot be done on the left curve.

The predicted recall curves for our previous four players among the 18 are presented in Figure 7.9. When compared to the results obtained using AUHWM for modeling the user's WM capacities (Figure 6.5), the performance difference becomes clear. Although Match^{2}s batch-based approach is aimed to filter out local fluctuations of performance, the WM-modulating factors still have their impact by

Table 7.2: Mean RMSE of the prediction of the last three batches for AUHWM embedded with the three WM models.

Model	W	V	RMSE
GB-based with $\sigma = 1$	1	0.001	0.13
GB-based with $\sigma = -1$	1	0.025	0.13
ACT-R-based	3.4	0.026	0.23

restricting AUHWM's predicting capability.

The RMSE per batch then becomes as shown in Figure 7.10. The parameter search resulted in an accuracy increase of roughly 2% for the three last batches, meaning that a mean error of 13% is the limit of AUHWM prediction capability when embedded with the GB-based model with $\sigma = -1$.

7.4 Comparison with Other WM Embeddings

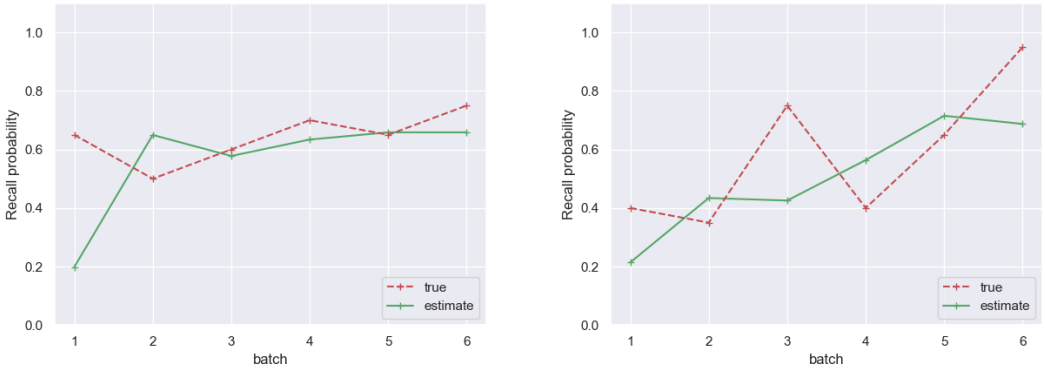
Employing the same procedure as described above for finding the best fit, the optimal parameterization for AUHWM embedded with the other two models was found. The best values for process and observation noise variance, W and V respectively, for each model is shown in Table 7.2, together with the mean RMSE for the last three batches. The comparison of the evolution of the RMSE per model of WM is shown in Figure 7.11.

All data concerning the results obtained here, the cognitive capacity estimations and the corresponding predicted recall curve, for all the players when embedding AUHWM with the three different WM models, can be accessed at http://cri.ensmp.fr/auhwm/.

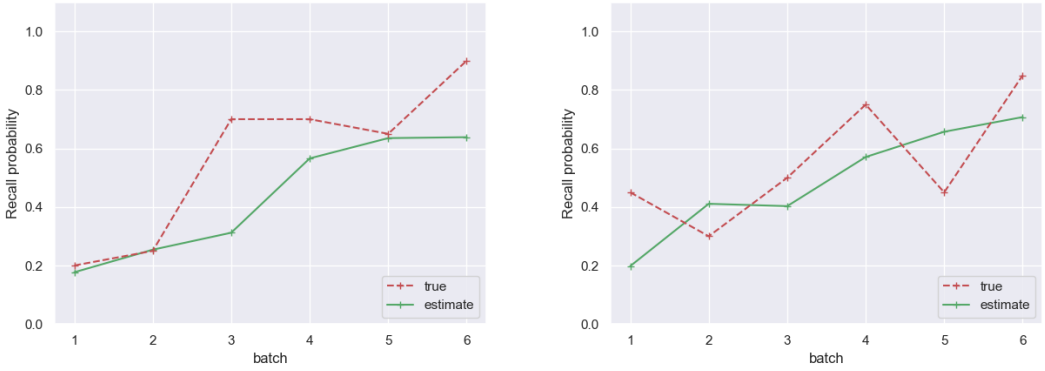
7.5 Discussion

This chapter presented the reader with a framework employing AUHWM to perform UI adaptation. The reader was also presented with validation of the framework's capability to predict user's performance given previous interactions. We do not discuss in this work the implementation of the Adaptive Optimizer block (Figure 7.1) as the validation here corresponds to a case where a hypothetical perfect Adaptive Optimizer is provided, meaning that the performance is exclusively dependent on AUHWM's capacity of inferring cognitive capacity.

The results presented in the last section show that the obtained performance



player 1 player 2



player 3 player 4

Figure 7.9: Predicted recall probabilities of 4 of the 18 players obtained using the GB model together with AUHWM-outputted estimations when parameterized with $W = 1$ and $V = 0.025$.

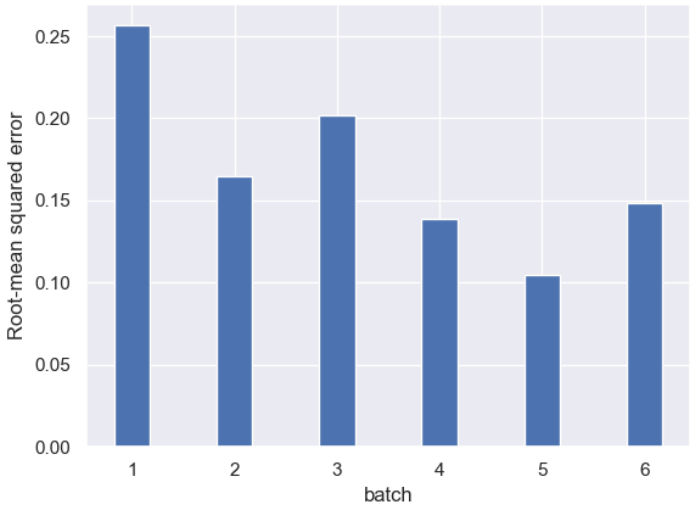


Figure 7.10: Evolution of the RMSE for the 18 players with the optimized parameters (AUHWM embedded with GB-based WM with $\sigma = -1$ model).

when employing the two GB-models show no apparent significant differences. However, when employing the ACT-R-based model, performance becomes considerably worst, as seen in Figure 7.10 . Note that, if the player's reaction time was taken into account (which means that T_t would be hiding time + reaction time, comprehending all the time information was stored) and the the optimization procedure above was performed, then the mean RMSE when employing the ACT-R-based model for the last three batches would decrease by 10%, while for the other models it would stay around the same. Therefore, one can hypothesize that the ACT-R model is more sensible to time than the other two and that a bigger reaction time would result in more memory degradation. Reaction time wasn't taken into account here because when selecting the task parameters, the Adaptive Optimizer has no control over it, moreover we posit that once the player was asked to click on the cued color, if the information is present in WM, he/she will perform it correctly.

However, it is clear that there is a degradation of AUHWM's performance when employing the previous estimations for predicting the future user's behavior, when compared to modeling a posteriori (last chapter). This degradation does not come as a surprise. The modeling results presented in the last chapter shows that in order to "overfit" the recall probabilities, the tracked estimations q_t have to closely follow the observed performance y_t. If performance were purely WM-dependent, and assuming that the user does not has some form of neurodegenerative disease, then the capacity estimations would be "smoother", which is not the case. This

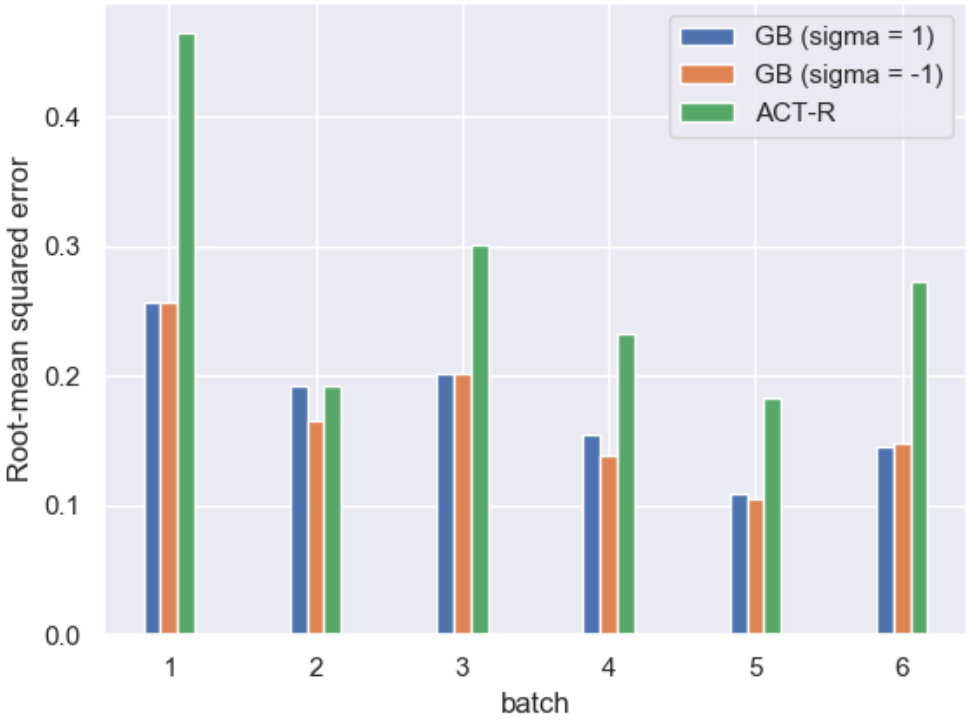


Figure 7.11: Mean RMSE for the prediction of the six batches for AUHWM when embedded with the three WM models.

means that a single parameter, q_t, isn't enough to perfectly reflect the user's WM behavior. As said before, there is no provided information about attention, motivation, or other modulating factors that affect WM performance; therefore, all these factors are encoded in the estimations. The most a AUHWM-based system can do to predict WM performance without being provided with information on these modulating factors is to increase the observation noise variance in order to smooth out the capacity estimations, making predictions around the mean behavior a user is presenting.

However, even without additional information, AUHWM's estimations can serve to predict recall probabilities with a fairly good accuracy (13% RMSE), even when dealing with users who were never seen before. This is possible because of the embedded models of WM degradation, meaning that AUHWM can deal with different populations that are not represented in the data, say people with cognitive deficits, for instance, resulting in valuable operational flexibility and adaptability. Our approach uses an online filtering technique able to adapt smoothly to the specific capabilities of any user, even in the presence of scarce data.

Moreover, given the UKF's capability of improving its estimations through sensor fusion, AUHWM could clearly be enhanced by employing other sensors. For instance, since BCIs such as EEGs can furnish estimations of attention, the user's schedule could be used to infer motivation throughout the day, or even AUHWM could be coupled with theoretical models for learning. The information coming from external sensors or added models would help modulate the estimations according to the users' concentration, tiredness and other key factors that drive WM capacity, therefore improving further the assessment of cognitive capacity and in consequence the automatic adaptation of UIs.

We believe that AUHWM-based frameworks capable of tracking cognitive limitations can be of extreme importance when developing UIs that are aware of the user's reasoning and memorizing abilities. Knowing how taxing in cognitive capacity a task is, would allow it to be simplified when necessary. Therefore, when in stressful situations, a UI could be simplified, providing the user with only the necessary information to the task at hand, compensating cognitive funneling. Aircraft interfaces in crisis situations are a clear application domain of our approach. Assistive technologies would also profit from awareness of the user's cognitive capacity, as it has the potential to be of great benefit to individuals suffering from memory deficits. By adjusting a UI to the user's cognitive capacities, it would render computer interfaces more accessible to the elderly population suffering from dementia-linked diseases. Moreover, by regulating the degree of complexity of the task interface, AUHWM can help immerse this population in a flow-like state, rendering them able to accomplish tasks autonomously, improving self-esteem and

decreasing stress levels. AUHWM can provide as well data specific to the user's evolving cognitive capacities. Beyond its clear relevance in the design of simpler UIs for computer-assisted daily-life activities, such information can be used by caregivers as signals suggesting a possibly setting-in of neurodegenerative diseases. It can also be used to track the gradual temporal decline of cognitive abilities as the disease progresses.

Chapter 8

Conclusion

In this thesis we have reported our work on providing computational systems with the awareness of user's cognitive limitations, notably working memory (WM). In this final chapter, we start by presenting this thesis' main contributions in Section 8.1. Then some general concluding remarks about the ideas developed here are provided in Section 8.2. We close the chapter by discussing the prospects on how this work can be used to develop future research, in Section 8.3.

8.1 Main Contributions

This work introduced two innovative frameworks for modeling user's WM capacity through interaction: MATCHS, which is an incremental batch-based approach for WM modeling, and AUHWM, which tracks in real-time the user's WM capacity. If MATCHS by itself is capable of helping adapt (at some extent) UIs to the estimated cognitive limitations, AUHWM however can be much more easily embedded in other systems to perform UI adaptation. Thus, a possible AUHWM-based UI-adaptation framework was also proposed in this work. To the best our knowledge, no other control system embedded with WM models has ever been developed to perform UI adaptation.

MATCHS

Memory Adaptation Through Cognitive Handling Simulation, or MATCHS, was introduced in Chapter 5. It is the framework where our core ideas used in both adaptation systems were first developed. MATCHS is dedicated to the adaptation of the information presented to a user according to the characteristics of his/her WM, thus helping to ensure the proper completion of a specific task.

MATCHS provides a numerical estimation for a user's WM capacity through

increments proportional to a batch-based assessment of the user's performance, the key idea being that this performance drives estimation inside the parameter space of the user model. MATCHS is embedded with Suchow's model of WM dynamics and uses these estimations to simulate the degradation of the information stored in the user's working memory. Therefore, MATCHS is able to predict how much information can be stored into the WM and for how long. This simulated memory evolution can ultimately be used to adapt the user interface of systems, ensuring the retention of the information necessary for a specific task.

MATCHS was experimentally tested with the data of 20 users, and its performance was analyzed and discussed, providing strong confidence in the validity of our approach.

AUHWM

An Unscented Hound for Working Memory, or AUHWM, was presented in Chapter 6. It is a derivation from MATCHS core ideas where the use of the Unscented Kalman Filter, coupled with a deterministic model of WM dynamics, allows one to leave behind MATCHS' incremental approach to modeling and make WM capacity estimations in real time. AUHWM is thus able to dynamically track a user's WM cognitive capabilities over both short- and long-term time intervals.

AUHWM can be embedded with different models of WM that abstract, via a single integer parameter, a user's memory capacity. Its performance when tracking cognitive profiles of 18 users was tested with three different models (two variants of Suchow's quantic WM model and the ACT-R model).

A proposed AUHM-based framework for automatic UI task adaptation was also presented in Chapter 7. The framework employed the embedded WM model together with AUHWM estimations of WM to go beyond "a posteriori" modeling and start predicting user performance. In cases where performance is degraded to a certain point due to WM limitations, calling for adaptation, the proposed solution is also capable of finding the task parameterization required to compensate such limitations, helping insure the proper completion of the task.

The validity of this framework was experimentally tested with data from 18 users, which showed that AUHWM was able to predict users' performance with an eventual 13% RMSE.

8.2 UI Adaptation to WM limitations

This thesis started with the quote "We shape our tools and, thereafter our tools shape us" from the American writer John M. Culkin. Throughout this work we were concerned with strategies for shaping our tools, which, in the context of this

thesis, stands for shaping computational systems to the human mind's way of storing and processing information.

We focused our work on developing general tools that could be integrated within different applications, providing interfaces with awareness to the user's WM capacity. Since WM is a bottleneck in human information-processing capacity, whenever information is considered and processed, we posit that WM is an important component to be taken into account. Wherever information is being diffused through Intelligent Tutoring Systems, assistive technologies, airplane cockpit displays, or even a multitude of daily-life applications where the limits of information processing are not a key limiting factor, WM is a common (though external) component to all of them, as it is part of the human mind functioning. Therefore, both proposed frameworks for adaptation, MATCHS and AUHWM-based (Figure 5.1.1 and Figures 7.1 respectively), contain a "Task" block corresponding to any task whose performance is directly dependent on WM capacity.

Although validation was concerned with the more tractable problem of adapting tasks whose performance is purely WM-based, we believe that both of these frameworks can be employed with any given task. For instance, we believe that MATCHS, with an appropriate setting of its parameters, can be applied to a wide range of tasks where WM is solicited. Moreover, we posit it should be able to adapt complex cognitive tasks to the user's available cognitive capacity, as long as a reliable estimation of the information to be retained is provided. AUHWM-based frameworks could not only perform adaptation given estimations of forgotten information as in MATCHS (remember that MATCHS' α from Figure 5.1.1 corresponds to the complement of AUHWM's π in Figure 7.1), but the UKF observation function could be enhanced with additional task models. By knowing how taxing in cognitive capacity a task is, an AUHWM-based framework could be used to infer user's performance and, more importantly, perform adaptation by selecting a compensatory action.

We were also interested in developing frameworks that are not restricted to available data modeling a specific task, as one key point considered here was flexibility. Adaptation systems that are purely ML-based can fail to provide proper adaptation when dealing with previously unseen situations. This constrains the system to use cases present in the training dataset, complexifyng the generalization of these solutions. For example, a solution derived from data of one population might not work correctly when faced with different users due to hidden variables that weren't taken in account; this can be a critical flaw when faced with scarce data regarding a specific user group (for example, Alzheimer patients). Both solutions developed here are based of theoretical models of WM. Building modeling and adaptation mechanisms upon well-understood models allows one to go beyond the simple association of input parameters and observed values. These models

provide generalizations of human WM dynamics flexible enough to be employed with users never seen before. Using AUHWM's estimations, for instance, we were able to predict user's performance with an average RMSE of 13% after a couple of interactions. These users didn't have to perform extensive calibration steps before the system could begin to assess their cognitive capacities. Match^{2}s players, when interacting with MATCHS for the first time, had the task parameters adapted to their observed capacity, which is shown by the error value's convergence towards zero throughout the batches.

In the particular case of AUHWM, the coupling of well-understood theoretical models with tracking algorithms such as the Kalman filter is a combination that shows great promise. On one side, there is a multitude of theoretical models in cognitive sciences that can (to some measure) explain the underlying mechanism behind the human mind. One can then select and employ interpretable models of how the mind deals with information, allowing the system decisions to be explained to the user, which ultimately improves the system's acceptance by building trust. Here is a possible example employing AUHWM: when presenting less information to the user by simplifying the interface, the adaptation framework could, if requested, explain to the user that it perceived she is getting tired (or less motivated), which called for simpler tasks.

However, there is an intrinsic fuzziness when dealing with human cognition, and purely deterministic frameworks might fall short when faced with it. The Kalman filter, on the other hand, able to filter out noisy observations and evaluate the uncertainty of its estimates, is a technique that can evaluate uncertain and noisy input data, therefore bypassing this fuzziness. Moreover, the Kalman filter is a powerful sensor fusion algorithm, meaning that AUHWM-based applications could be enhanced through the addition of external sensors. In applications where the precise assessment of cognitive charge is crucial for preventing fatal errors (such as the ones cited in Section 3.3.3) and where the use of physiological sensors does not pose a practical problem, AUHWM can be enhanced with more information about the user's cognitive state, thus providing both monitoring data on how the users capacity is evolving during the continuous interaction, as well as information on how taxing WM an given application is. One can then think of AUHWM's modeling capability as an annotation tool for finding the implication of WM in a given task.

8.3 Future Work

In this section we discuss the perspectives of how the ideas developed in this work can be used to advance further towards the goal of providing cognitive capacity awareness to computer systems.

Future work should focus, at the fundamental level, on improving AUHWM through the addition of models for attention, motivation, learning or other factors that influence WM performance. The results presented here suffer from the fact that a number of hidden factors modulate WM intrapersonal performance. The results presented in Figure 6.7 show that in order to closely track the observed performance, the tracked estimations are not smooth, which also is the cause of the augmentation of the RMSE when predicting performance. This suggests that performance fluctuations are not purely WM-based, capacity-wise, meaning that WM cannot be tracked with a single unidimensional parameter. Future work should, therefore, focus on the addition of sensors or models in order to account for these modulating factors in the modeling equations of the UKF.

For instance, a quantitative estimate of a user's attention to the task at hand would clearly impact positively the assessment of the short-term evolution of her cognitive retention capabilities, and hopefully lead to the removal of some of the fluctuations in the tracking of users' performance. A first approach to such an estimation process could be via the use of dedicated sensors, e.g., brain computer interfaces (BCI) such as the EEG-based headband Muse™, which can be used to provide estimates on brain activity related to vigilance or attention in real time. Even though this doesn't constitute a workable solution in the long term for obvious usability reasons, such a study could nonetheless provide ways to refine AUHWM's process of memory-capacity tracking , and spur further research into finding more pragmatic ways to assess users' attention. Other possible improvements could come through motivation estimations. Since WM performance tends to degrade with lack of motivation, such estimations could be used to decrease the inputted q_t in the observation function H as the user interacts with the UI in a somewhat continuous fashion. Moreover, models of learning (that are very abundant in literature, e.g. BKT) could be used to dynamically change the stability threshold parameter L in Suchow's model, or the base-level activation in ACT-R, in the context of a given task.

In practice, this would result in a multivariate tracking of parameters corresponding to WM capacity, attention, motivation and learning, creating a multidimensional profile of the user's current cognitive state. These parameters would have each a different transition function to be defined, where some variated more than the others during different periods of time. Such a system would also require the definition of a more complex, multidimensional observation function, where all the different dimensions of the user's cognitive state are taken into account to infer task performance. If such a model were developed, AUHWM could assess changes in the modulating factors in real time by the use of other embedded models or sensors; then, instead of having the estimates changing from iteration to iteration, the smoother obtained estimations could be modulated by local changes in con-

centration, fatigue and learning, increasing or decreasing performance accordingly. This would result in a better identification of the user's current cognitive profile's impact on performance as well as selection of adequate compensatory actions, depending on the performance-degradation causes. Needless to say, the models used in the UKF would probably be considerably more complex.

Future work should then be concerned with incremental improvements through the addition of one extra tracked variable at a time. We are currently working on including EEG data corresponding to vigilance and attention, with the goal of filtering out attentional variations in order to obtain more precise predictions of performance; however the results so far have been inconclusive.

Another track for possible continuations is the testing and adaptation of AUHWM to more practical applications. As said before, this work was focused on the tractable problem of adapting task interfaces whose performance was purely WM-dependent. However, we believe that any task whose performance is somewhat dependent on WM can be adapted to some extent by employing AUHWM. Future work should then focus on applying AUHWM to more meaningful UI-adaptation use cases than Match^{2}s. For instance, a framework such as AUHWM could be adapted to perform scaffolding in intelligent tutoring systems. Again, by adding a theoretical transition model of learning and taking into account users' accuracy and reaction times into its observation model, AUHWM could be used to track a second parameter corresponding to the user's mastery of the knowledge being tutored. Therefore, complex applications such as Ansys or CAD software, which are daunting for novice users, could be personalized by varying the level of the UI's complexity and support the computer can offer according to the user's mastery and cognitive state.

8.4 Epilogue

Technologies aimed at humans should be designed around humans. This work tried to shed some light on ways to give computer systems some "understanding" about how human cognitive performance is affected by WM limitations. We believe this is a crucial point in many different areas. For instance, embedding assistive technologies with AUHWM-like technologies would represent a significant contribution in the area and would be of great benefit to individuals suffering from memory deficits. The field of artificial intelligence will continue to develop, and more and more humans will be faced with increasingly complex interfaces and intelligent systems. We sincerely believe that it is our job, when developing these machines, to embed them with a deep understanding of humans and of our limitations, so both human and artificially intelligent systems can advance together. Still, much light has yet to be shed on our mind's function.

Chapter 9

Conclusion

Dans cette thèse, nous avons présenté nos travaux visant à fournir aux systèmes informatiques une prise de conscience des limitations cognitives de l'utilisateur, notamment la mémoire de travail (WM). Dans ce dernier chapitre, nous commençons par présenter les principales contributions de cette thèse dans la section 9.1. Ensuite, quelques remarques générales de conclusion sur les idées développées ici sont fournies dans la section 9.2. Nous terminons le chapitre en discutant des perspectives sur la façon dont ce travail peut être utilisé pour développer de futures recherches, dans la section 9.3.

9.1 Contributions principales

Ce travail a introduit deux cadres innovants pour modéliser la capacité de WM de l'utilisateur par l'interaction : MATCHS, qui est une approche par lots incrémentale pour la modélisation de la WM, et AUHWM, qui suit en temps réel la capacité de WM de l'utilisateur. Si MATCHS par lui-même est capable d'aider à adapter (dans une certaine mesure) les interfaces utilisateur aux limitations cognitives estimées, AUHWM peut cependant être beaucoup plus facilement intégré dans d'autres systèmes pour effectuer l'adaptation de l'interface utilisateur. Ainsi, un cadre d'adaptation de l'interface utilisateur fondé sur AUHWM a également été proposé dans ce travail. À notre connaissance, aucun autre système de contrôle intégré aux modèles de WM n'a jamais été développé pour effectuer l'adaptation de l'interface utilisateur.

MATCHS

"Memory Adaptation Through Cognitive Handling Simulation", ou MATCHS, a été introduit dans le chapitre 5. C'est le cadre dans lequel les idées fondamen-

tales utilisées dans les deux systèmes d'adaptation ont d'abord été développées. MATCHS est dédié à l'adaptation des informations présentées à un utilisateur en fonction des caractéristiques de sa WM, contribuant ainsi à assurer la bonne réalisation d'une tâche spécifique.

MATCHS fournit une estimation numérique de la capacité de la WM d'un utilisateur par incréments proportionnels à une évaluation par lots de la performance de l'utilisateur, l'idée clé étant que ces performances déterminent l'estimation à l'intérieur de l'espace des paramètres du modèle de l'utilisateur. MATCHS contient le modèle dynamique de la WM proposé par Suchow et utilise ces estimations pour simuler la dégradation des informations stockées dans la mémoire de travail de l'utilisateur. Par conséquent, MATCHS est capable de prévoir la quantité d'informations pouvant être stockées dans la WM et pendant combien de temps. Cette évolution de la mémoire simulée peut à terme être utilisée pour adapter l'interface utilisateur des systèmes, assurant ainsi la rétention des informations nécessaires à une tâche spécifique.

MATCHS a été testé expérimentalement avec les données de 20 utilisateurs, et ses performances ont été analysées et discutées, fournissant une forte confiance dans la validité de notre approche.

AUHWM

"An Unscented Hound for Working Memory", ou AUHWM, a été présenté dans le chapitre 6. AUHWM est une dérivation des idées fondamentales de MATCHS où l'utilisation d'un filtre de Kalman non linéaire (Unscented), couplée à un modèle déterministe de la dynamique WM. AUHWM permet d'aller au-delà de l'approche incrémentale de MATCHS pour la modélisation et de faire des estimations de capacité de WM en temps réel. AUHWM est ainsi capable de suivre dynamiquement les capacités cognitives en WM d'un utilisateur sur des intervalles de temps à court et à long terme.

AUHWM peut utiliser différents modèles de WM qui abstraient, via un seul paramètre entier, la capacité de mémoire d'un utilisateur. Ses performances lors du suivi des profils cognitifs de 18 utilisateurs ont été testées avec trois modèles différents (deux variantes du modèle WM quantique et le modèle ACT-R).

Un cadre théorique fondé sur AUHM pour l'adaptation automatique des tâches de l'interface utilisateur a également été présenté dans le chapitre 7. Le cadre a utilisé le modèle WM intégré avec les estimations AUHWM de WM pour aller au-delà de la modélisation "a posteriori" et commencer à prédire les performances des utilisateurs. Dans les cas où les performances sont dégradées à un certain point en raison de limitations de la WM, nécessitant une adaptation, la solution proposée est également capable de trouver le paramétrage de tâche requis pour compenser ces limitations, contribuant ainsi à assurer la bonne exécution de la tâche.

La validité de ce cadre a été testée expérimentalement avec des données de 18 utilisateurs, ce qui a montré que AUHWM était capable de prédire les performances des utilisateurs avec un RMSE de 13%.

9.2 Adaptation des UIs aux limitations de WM

Cette thèse a commencé par la citation "Nous façonnons nos outils et, par la suite, nos outils nous façonnent" de l'écrivain américain John M. Culkin. Tout au long de ce travail, nous nous sommes intéressés aux stratégies pour façonner nos outils, qui, dans le contexte de cette thèse, représentent l'adaptation des systèmes informatiques à la manière dont l'esprit humain stocke et traite les informations.

Nous avons concentré notre travail sur le développement d'outils généraux qui pourraient être intégrés dans différentes applications, fournissant à des interfaces une estimation de la capacité de la WM de l'utilisateur. Étant donné que la WM est un goulot d'étranglement dans la capacité de traitement de l'information humaine, chaque fois qu'une information est prise en compte et traitée, nous croyons que la WM est un élément important à prendre en compte. Partout où des informations sont diffusées via des systèmes de tutorat intelligents, des technologies d'assistance, des écrans de poste de pilotage d'avion ou même une multitude d'applications de la vie quotidienne où les limites du traitement de l'information ne sont pas un facteur limitant clé, la WM est un composant commun (bien qu'externe) de ces systèmes, car elle fait partie du fonctionnement de l'esprit humain. Par conséquent, les deux cadres proposés pour l'adaptation, fondés sur MATCHS et AUHWM (Figure 5.1.1 et Figures 7.1 respectivement), contiennent un bloc "Task" correspondant à toute tâche dont les performances dépendent directement de la capacité de WM.

Bien que la validation concernât le problème plus facile à appréhender de l'adaptation de tâches dont les performances sont purement fondées sur la WM, nous pensons que ces deux cadres peuvent être utilisés avec n'importe quelle tâche donnée. Par exemple, nous pensons que MATCHS, avec un réglage approprié de ses paramètres, peut être appliqué à un large éventail de tâches où la WM est sollicitée. De plus, nous supposons qu'il devrait être capable d'adapter des tâches cognitives complexes à la capacité cognitive disponible de l'utilisateur, tant qu'une estimation fiable des informations oubliées est fournie. Les systèmes fondés sur AUHWM pourraient non seulement effectuer l'adaptation étant donné les estimations d'informations oubliées comme dans MATCHS (rappelez-vous que l'α de MATCHS de la figure 5.1.1 correspond au complément de π d'AUHWM dans la figure 7.1), mais la fonction d'observation de l'UKF pourrait être améliorée avec des modèles de tâches supplémentaires. En sachant à quel point la tâche est exigeante en termes de capacité cognitive, un système fondé sur AUHWM pourrait être utilisé pour déduire les performances de l'utilisateur et, plus important encore,

effectuer l'adaptation en sélectionnant une action compensatoire.

Nous étions également intéressés à développer des cadres théoriques qui ne se limitent pas aux données disponibles modélisant une tâche spécifique, car un point clé considéré ici était la flexibilité. Les systèmes d'adaptation qui sont purement fondés sur le ML peuvent ne pas fournir une adaptation appropriée lorsqu'ils font face à des situations inédites. Cela contraint le système à utiliser les cas présents dans le jeu de données d'apprentissage, complexifiant la généralisation de ces solutions. Par exemple, une solution dérivée des données d'une population peut ne pas fonctionner correctement face à différents utilisateurs en raison de variables cachées qui n'ont pas été prises en compte ; cela peut être un défaut critique face à des données rares concernant un groupe d'utilisateurs spécifique (par exemple, les patients Alzheimer). Les deux solutions développées ici sont fondées sur des modèles théoriques de la WM. Construire des mécanismes de modélisation et d'adaptation sur des modèles bien compris permet d'aller au-delà de la simple association des paramètres d'entrée et des valeurs observées. Ces modèles fournissent des généralisations de la dynamique de la WM humaine suffisamment flexibles pour être utilisées avec des utilisateurs jamais vus auparavant. En utilisant les estimations d'AUHWM, par exemple, nous avons pu prédire les performances de l'utilisateur avec un RMSE moyen de 13% après quelques interactions. Ces utilisateurs n'ont pas eu à effectuer des étapes d'étalonnage approfondies avant que le système puisse commencer à évaluer leurs capacités cognitives. Les joueurs de Match^{2}s, lors de leur première interaction avec MATCHS, utilisaient les paramètres de tâche adaptés à leur capacité observée, ce qui est illustré par la convergence de la valeur d'erreur vers zéro tout au long des lots.

Dans le cas particulier d'AUHWM, le couplage de modèles théoriques bien compris avec des algorithmes de suivi tels que le filtre de Kalman est une combinaison très prometteuse. D'un côté, il existe une multitude de modèles théoriques en sciences cognitives qui peuvent (dans une certaine mesure) expliquer le mécanisme sous-jacent derrière l'esprit humain. On peut ensuite sélectionner et utiliser des modèles interprétables de la façon dont l'esprit traite les informations, permettant aux décisions du système d'être expliquées à l'utilisateur, ce qui améliore finalement l'acceptation du système en instaurant la confiance. Voici un exemple possible utilisant AUHWM : lors de la présentation de moins d'informations à l'utilisateur en simplifiant l'interface, le cadre d'adaptation pourrait, si demandé, expliquer à l'utilisateur qu'il avait l'impression que l'utilisateur se fatiguait (ou avait moins de motivation), ce qui nécessitait des tâches plus simples.

Cependant, il y a un flou intrinsèque lorsqu'il s'agit de la cognition humaine, et les cadres purement déterministes peuvent échouer face à elle. Le filtre de Kalman, d'autre part, capable de filtrer les observations bruitées et d'évaluer l'incertitude de ses estimations, est une technique qui peut évaluer des données d'entrée incertaines

et bruitées, contournant ainsi ce flou. De plus, le filtre de Kalman est un puissant algorithme de fusion de capteurs, ce qui signifie que les applications fondées sur AUHWM pourraient être améliorées grâce à l'ajout de capteurs externes. Dans les applications où l'évaluation précise de la charge cognitive est cruciale pour prévenir les erreurs fatales (telles que celles citées dans la section 3.3.3) et où l'utilisation de capteurs physiologiques ne pose pas de problème pratique, AUHWM peut être amélioré avec plus d'informations sur l'état cognitif de l'utilisateur, fournissant ainsi à la fois des données de surveillance sur la façon dont la capacité des utilisateurs évolue au cours de l'interaction continue, ainsi que des informations sur la façon dont la WM est impactée par une application donnée. On peut alors penser à la capacité de modélisation d'AUHWM comme un outil d'annotation pour trouver l'implication de la WM dans une tâche donnée.

9.3 Travaux futurs

Dans cette section, nous discutons des perspectives sur la façon dont les idées développées dans ce travail peuvent être utilisées pour progresser davantage vers l'objectif de mieux prendre en compte les capacités cognitives par des systèmes informatiques.

Les travaux futurs devraient se concentrer, au niveau fondamental, sur l'amélioration d'AUHWM par l'ajout de modèles d'attention, de motivation, d'apprentissage ou d'autres facteurs qui influencent les performances de la WM. Les résultats présentés ici souffrent du fait qu'un certain nombre de facteurs cachés modulent les performances intrapersonnelles de la WM. Les résultats présentés dans la figure 6.7 montrent que, pour suivre de près les performances observées, les estimations suivies ne sont pas lisses, ce qui est également la cause de l'augmentation du RMSE lors de la prévision des performances. Cela suggère que les fluctuations de performance ne sont pas uniquement liées à la WM ; en termes de capacité, ce qui signifie que la WM ne peut pas être suivie avec un seul paramètre unidimensionnel. Les travaux futurs devraient donc se concentrer sur l'ajout de capteurs ou de modèles afin de tenir compte de ces facteurs de modulation dans les équations de modélisation du filtre UKF.

Par exemple, une estimation quantitative de l'attention d'un utilisateur à la tâche à accomplir aurait clairement un impact positif sur l'évaluation de l'évolution à court terme de ses capacités de rétention cognitive et, espérons-le, entraînerait la suppression de certaines des fluctuations du suivi de performance des utilisateurs. Une première approche d'un tel processus d'estimation pourrait être via l'utilisation de capteurs dédiés, par exemple, les interfaces cerveau-ordinateur (BCI) telles que l'EEG Muse™, qui peut être utilisé pour fournir des estimations sur l'activité cérébrale liée à la vigilance ou attention en temps réel. Même si cela ne constitue

pas une solution viable à long terme pour des raisons évidentes d'utilisation, une telle étude pourrait néanmoins fournir des moyens d'affiner le processus de suivi de la capacité de la mémoire par AUHWM, et stimuler la recherche pour trouver des moyens plus pragmatiques d'évaluer l'attention des utilisateurs. D'autres améliorations possibles pourraient venir des estimations de la motivation. Étant donné que les performances de WM ont tendance à se dégrader avec un manque de motivation, de telles estimations pourraient être utilisées pour diminuer le q_t entré dans la fonction d'observation H lorsque l'utilisateur interagit avec l'UI de manière peu continue. De plus, des modèles d'apprentissage (très abondants dans la littérature, par exemple BKT) pourraient être utilisés pour modifier dynamiquement le paramètre de seuil de stabilité L dans le modèle de Suchow, ou l'activation de base dans ACT-R, dans le contexte d'une donnée tâche.

En pratique, cela entraînerait un suivi multivarié des paramètres correspondant à la capacité, à l'attention, à la motivation et à l'apprentissage de la WM, créant un profil multidimensionnel de l'état cognitif actuel de l'utilisateur. Ces paramètres auraient chacun une fonction de transition différente, à définir, certains variant plus que d'autres au cours de différentes périodes. Un tel système nécessiterait également la définition d'une fonction d'observation multidimensionnelle plus complexe, où toutes les différentes dimensions de l'état cognitif de l'utilisateur sont prises en compte pour déduire les performances de la tâche. Si un tel modèle était développé, AUHWM pourrait évaluer les changements des facteurs de modulation en temps réel en utilisant d'autres modèles ou capteurs intégrés; puis, au lieu de faire basculer les estimations d'itération en itération, des estimations obtenues plus lisses pourraient être modulées par des changements locaux de concentration, de fatigue et d'apprentissage, augmentant ou diminuant les performances en conséquence. Il en résulterait une meilleure identification de l'impact du profil cognitif actuel de l'utilisateur sur les performances ainsi qu'une sélection d'actions compensatoires adéquates, en fonction des causes de dégradation des performances. Il va sans dire que les modèles utilisés dans le filtrage UKF seraient probablement beaucoup plus complexes.

Les travaux futurs devraient sans doute porter sur des améliorations progressives grâce à l'ajout d'une variable de suivi supplémentaire à la fois. Nous travaillons actuellement sur l'inclusion de données EEG correspondant à la vigilance et à l'attention, dans le but de filtrer les variations attentionnelles afin d'obtenir des prédictions plus précises des performances ; cependant, les résultats n'ont jusqu'à présent pas été concluants.

Une autre piste pour les suites possibles est le test et l'adaptation d'AUHWM à des applications plus pratiques. Comme dit précédemment, ce travail s'est concentré sur le problème traitable de l'adaptation des interfaces de tâches dont les performances étaient purement dépendantes de la WM. Cependant, nous pensons que

toute tâche dont les performances dépendent quelque peu de la WM peut être adaptée dans une certaine mesure en utilisant AUHWM. Les travaux futurs devraient alors se concentrer sur l'application d'AUHWM à des cas d'utilisation d'adaptation de l'interface utilisateur plus significatifs que Match²s. Par exemple, un cadre comme AUHWM pourrait être adapté pour effectuer des échafaudages ("scaffolding") dans des systèmes de tutorat intelligents. Encore une fois, en ajoutant un modèle de transition théorique de l'apprentissage et en tenant compte de la précision et des temps de réaction des utilisateurs dans son modèle d'observation, AUHWM pourrait être utilisé pour suivre un deuxième paramètre correspondant à la maîtrise par l'utilisateur des connaissances enseignées. Par conséquent, des applications complexes telles que Ansys ou un logiciel de CAD, qui sont intimidantes pour les utilisateurs novices, pourraient être personnalisées en variant le niveau de complexité de l'interface utilisateur et le soutien que l'ordinateur peut offrir en fonction de la maîtrise de l'utilisateur et de son état cognitif.

9.4 Épilogue

Les technologies destinées aux humains doivent être conçues autour de l'homme. Ce travail a tenté de faire la lumière sur les moyens de donner aux systèmes informatiques une certaine "compréhension" de la façon dont les performances cognitives humaines sont affectées par les limitations de la WM. Nous pensons que c'est un point crucial dans de nombreux domaines différents. Par exemple, l'intégration de technologies d'assistance aux technologies de type AUHWM représenterait une contribution importante dans le domaine et serait très bénéfique pour les personnes souffrant de déficits de mémoire. Le domaine de l'intelligence artificielle continuera de se développer et de plus en plus d'humains seront confrontés à des interfaces et des systèmes intelligents de plus en plus complexes. Nous croyons sincèrement qu'il est de notre devoir, lors du développement de ces machines, de les intégrer à une compréhension profonde des humains et de nos limites, afin que les systèmes humains et artificiellement intelligents puissent progresser ensemble. Pourtant, nous sommes encore loin d'avoir fait toute la lumière sur le fonctionnement de l'esprit humain.

Bibliography

[1] C. Worldwide, "Ansys workbench tutorial video | beginner/expert | crain hook contact non linear fe analysis | grs |." [Online] https://youtu.be/YEI62Mgb5P4, 2014.

[2] J. Tromp, "The number of legal go positions," in *International Conference on Computers and Games*. Springer, 2016, pp. 183–190.

[3] E. Hudlicka and M. D. McNeese, "Assessment of user affective and belief states for interface adaptation: Application to an Air Force pilot task," *User Modeling and User-Adapted Interaction*, vol. 12, no. 1, pp. 1–47, 2002.

[4] B. Massoni Sguerra, P. Jouvelot, and S. Benveniste, "Oblivion tracking: Towards a probabilistic working memory model for the adaptation of systems to alzheimer patients," in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 2017, pp. 253–256.

[5] B. M. Sguerra, A. Benamara, S. Benveniste, and P. Jouvelot, "Adapting human-computer interfaces to working memory limitations using matchs," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 1309–1314.

[6] B. Massoni Sguerra and P. Jouvelot, "" an unscented hound for working memory" and the cognitive adaptation of user interfaces," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. ACM, 2019, pp. 78–85.

[7] A. Miyake and P. Shah, *Models of working memory: Mechanisms of active maintenance and executive control*. Cambridge University Press, 1999.

[8] B. R. Huguenard, F. J. Lerch, B. W. Junker, R. J. Patz, and R. E. Kass, "Working-memory failure in phone-based interaction," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 4, no. 2, pp. 67–102, 1997.

[9] C. Germano and G. J. Kinsella, "Working memory and learning in early Alzheimer's disease," *Neuropsychology Review*, vol. 15, no. 1, pp. 1–10, 2005.

[10] K. Schuchardt, C. Maehler, and M. Hasselhorn, "Working memory deficits in children with specific learning disorders," *Journal of Learning Disabilities*, vol. 41, no. 6, pp. 514–523, 2008.

[11] A. Baddeley, "Working memory and language: An overview," *Journal of communication disorders*, vol. 36, no. 3, pp. 189–208, 2003.

[12] R. C. Atkinson and R. M. Shiffrin, "Human memory: A proposed system and its control processes," in *Psychology of learning and motivation*. Elsevier, 1968, vol. 2, pp. 89–195.

[13] A. Baddeley, "Working Memory Alan Baddeley," *Science*, vol. 255, no. 5044, pp. 556–559, 1992.

[14] M. A. Just and P. A. Carpenter, "A capacity theory of comprehension: individual differences in working memory." *Psy. Rev.*, vol. 99, no. 1, p. 122, 1992.

[15] J. R. Anderson, L. M. Reder, and C. Lebiere, "Working memory: Activation limitations on retrieval," *Cognitive psychology*, vol. 30, no. 3, pp. 221–256, 1996.

[16] A. Baddeley, "Working memory: theories, models, and controversies." *Annual review of psychology*, vol. 63, pp. 1–29, 2012.

[17] ——, "The episodic buffer: A new component of working memory?" pp. 417–423, 2000.

[18] N. Cowan, "The magical number 4 in short term memory. A reconsideration of storage capacity," *Behavioral and Brain Sciences*, vol. 24, no. 4, pp. 87–186, 2001.

[19] A. Baddeley, *Working memory, thought, and action*. OUP Oxford, 2007, vol. 45.

[20] K. Oberauer and H.-y. Lin, "An Interference Model of Visual Working Memory." *Psychological Review*, vol. 124, no. 1, pp. 1–39, 2016.

[21] G. A. Miller, "the Magical Number 7, Plus or Minus 2 - Some Limits on Our Capacity for Processing Information," *Psychological Review-New York-*, vol. 63, no. 2, pp. 81–97, 1956.

[22] H. Pashler, "Familiarity and visual change detection." *Perception & psychophysics*, vol. 44, no. 4, pp. 369–378, 1988.

[23] R. P. Kessels, M. J. Van Zandvoort, A. Postma, L. J. Kappelle, and E. H. De Haan, "The corsi block-tapping task: standardization and normative data," *Applied neuropsychology*, vol. 7, no. 4, pp. 252–258, 2000.

[24] A. Baddeley, "Working memory: looking back and looking forward," *Nature Reviews Neuroscience*, vol. 4, no. 10, pp. 829–839, 2003.

[25] W. Zhang and S. J. Luck, "Sudden death and gradual decay in visual working memory." *Psychological science*, vol. 20, no. 4, pp. 423–8, 2009.

[26] K. Oberauer, "Working memory and attention–a conceptual analysis and review," *Journal of cognition*, vol. 2, no. 1, 2019.

[27] D. Fougnie, "The relationship between attention and working memory," *New research on short-term memory*, vol. 1, p. 45, 2008.

[28] J. R. Anderson, *How can the human mind occur in the physical universe?* Oxford University Press, 2009, vol. 3.

[29] A. Brose, F. Schmiedek, M. Lövdén, and U. Lindenberger, "Daily variability in working memory is coupled with negative affect: the role of attention and motivation." *Emotion*, vol. 12 3, pp. 605–17, 2012.

[30] J. Pochon, R. Levy, P. Fossati, S. Lehericy, J. Poline, B. Pillon, D. Le Bihan, and B. Dubois, "The neural system that bridges reward and cognition in humans: an fmri study," *Proceedings of the National Academy of Sciences*, vol. 99, no. 8, pp. 5669–5674, 2002.

[31] P. Chandler and J. Sweller, "Cognitive load theory and the format of instruction," *Cognition and instruction*, vol. 8, no. 4, pp. 293–332, 1991.

[32] J. Gwizdka, "Distribution of cognitive load in web search," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 11, pp. 2167–2187, 2010.

[33] J. Sweller, "Cognitive load theory, learning difficulty, and instructional design," *Learning and instruction*, vol. 4, no. 4, pp. 295–312, 1994.

[34] R. Brunken, J. L. Plass, and D. Leutner, "Direct measurement of cognitive load in multimedia learning," *Educational psychologist*, vol. 38, no. 1, pp. 53–61, 2003.

[35] F. Paas, J. E. Tuovinen, H. Tabbers, and P. W. Van Gerven, "Cognitive load measurement as a means to advance cognitive load theory," *Educational psychologist*, vol. 38, no. 1, pp. 63–71, 2003.

[36] M. D'Esposito, K. Onishi, H. Thompson, K. Robinson, C. Armstrong, and M. Grossman, "Working memory impairments in multiple sclerosis: Evidence from a dual-task paradigm." *Neuropsychology*, vol. 10, no. 1, p. 51, 1996.

[37] J. R. Anderson, C. F. Boyle, and B. J. Reiser, "Intelligent tutoring systems," *Science*, vol. 228, no. 4698, pp. 456–462, 1985.

[38] J. D. Huntley and R. J. Howard, "Working memory in early Alzheimer's disease: A neuropsychological review," pp. 121–132, 2010.

[39] W. H. Organization and A. D. International, "Dementia: a public health priority." [Online] http://www.who.int/mental_health\protect\discretionary{\char\hyphenchar\font}{}{}/publications/dementia_report_2012/en/, 2012.

[40] C. Graf, "The Lawton instrumental activities of daily living scale," *The American Journal of Nursing*, vol. 108, no. 4, pp. 52–53, 2008.

[41] J. A. Opara, "Activities of daily living and quality of life in Alzheimer disease." *Journal of Medicine & Life*, vol. 5, no. 2, pp. 162–167, 2012.

[42] A. Association, "Communication and alzheimer." 2017.

[43] S. L. Beilock and M. S. DeCaro, "From poor performance to success under stress: working memory, strategy selection, and mathematical problem solving under pressure." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 33, no. 6, p. 983, 2007.

[44] C. D. Wickens, A. Stokes, B. Barnett, and F. Hyman, "The effects of stress on pilot judgment in a midis simulator," in *Time pressure and stress in human judgment and decision making*. Springer, 1993, pp. 271–292.

[45] S. Conversy, S. Chatty, H. Gaspard-Boulinc, and J.-L. Vinot, "The accident of flight af447 rio-paris: a case study for hci research," in *Proceedings of the 26th Conference on l'Interaction Homme-Machine*. ACM, 2014, pp. 60–69.

[46] C. D. Wickens, "Attentional tunneling and task management," in *2005 International Symposium on Aviation Psychology*, 2005, p. 812.

[47] P. Biswas and P. Robinson, "A brief survey on user modelling in hci," in *Proc. of the International Conference on Intelligent Human Computer Interaction (IHCI)*, vol. 2010, 2010.

[48] K.-H. Tan and B. P. Lim, "The artificial intelligence renaissance: deep learning and the road to human-level machine intelligence," *APSIPA Transactions on Signal and Information Processing*, vol. 7, 2018.

[49] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User modeling and user-adapted interaction*, vol. 4, no. 4, pp. 253–278, 1994.

[50] T. Käser, S. Klingler, A. G. Schwing, and M. Gross, "Beyond knowledge tracing: Modeling skill topologies with bayesian networks," in *International Conference on Intelligent Tutoring Systems*. Springer, 2014, pp. 188–198.

[51] V. A. Aleven and K. R. Koedinger, "An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor," *Cognitive science*, vol. 26, no. 2, pp. 147–179, 2002.

[52] F. Putze and T. Schultz, "Adaptive cognitive technical systems," *Journal of neuroscience methods*, vol. 234, pp. 108–115, 2014.

[53] J. Zagermann, U. Pfeil, and H. Reiterer, "Measuring cognitive load using eye tracking technology in visual computing," in *Proceedings of the sixth workshop on beyond time and errors on novel evaluation methods for visualization*. ACM, 2016, pp. 78–85.

[54] J. Beatty, "Task-evoked pupillary responses, processing load, and the structure of processing resources." *Psychological bulletin*, vol. 91, no. 2, p. 276, 1982.

[55] D. Toker and C. Conati, "Leveraging pupil dilation measures for understanding users' cognitive load during visualization processing," in *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 2017, pp. 267–270.

[56] O. Palinko, A. L. Kun, A. Shyrokov, and P. Heeman, "Estimating cognitive load using remote eye tracking in a driving simulator," in *Proceedings of the 2010 symposium on eye-tracking research & applications*. ACM, 2010, pp. 141–144.

[57] J. Liu, A. Gardi, S. Ramasamy, Y. Lim, and R. Sabatini, "Cognitive pilot-aircraft interface for single-pilot operations," *Knowledge-Based Systems*, vol. 112, pp. 37–53, 2016.

[58] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, and T. M. Vaughan, "Brain-computer interface technology: a review of the first international meeting," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 2, pp. 164–173, 2000.

[59] A. Gevins, M. E. Smith, H. Leong, L. McEvoy, S. Whitfield, R. Du, and G. Rush, "Monitoring working memory load during computer-based tasks with eeg pattern recognition methods," *Human factors*, vol. 40, no. 1, pp. 79–91, 1998.

[60] K. Tsiakas, C. Abellanoza, M. Abujelala, M. Papakostas, T. Makada, and F. Makedon, "Towards Designing a Socially Assistive Robot for Adaptive and Personalized Cognitive Training," in *International Conference on Human Robot Interaction (HRI 2017)*, 2017.

[61] P. Zarjam, J. Epps, and F. Chen, "Spectral eeg featuresfor evaluating cognitive load," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 3841–3844.

[62] H. Ayaz, P. A. Shewokis, S. Bunce, K. Izzetoglu, B. Willems, and B. Onaral, "Optical brain monitoring for operator training and mental workload assessment," *Neuroimage*, vol. 59, no. 1, pp. 36–47, 2012.

[63] K. Tsiakas, M. Abujelala, and F. Makedon, "Task engagement as personalization feedback for socially-assistive robots and cognitive training," *Technologies*, vol. 6, no. 2, p. 49, 2018.

[64] Y. Lim, T. Samreeloy, C. Chantaraviwat, N. Ezer, A. Gardi, R. Sabatini *et al.*, "Cognitive human-machine interfaces and interactions for multi-uav operations," in *AIAC18: 18th Australian International Aerospace Congress (2019): HUMS-11th Defence Science and Technology (DST) International Conference on Health and Usage Monitoring (HUMS 2019): ISSFD-27th International Symposium on Space Flight Dynamics (ISSFD)*. Engineers Australia, Royal Aeronautical Society., 2019, p. 40.

[65] S. Oviatt, "Human-centered design meets cognitive load theory: designing interfaces that help people think," in *Proceedings of the 14th ACM international conference on Multimedia*. ACM, 2006, pp. 871–880.

[66] X. Fan, P.-C. Chen, and J. Yen, "Learning hmm-based cognitive load models for supporting human-agent teamwork," *Cognitive Systems Research*, vol. 11, no. 1, pp. 108–119, 2010.

[67] F. Putze, M. Salous, and T. Schultz, "Detecting memory-based interaction obstacles with a recurrent neural model of user behavior," in *23rd International Conference on Intelligent User Interfaces*. ACM, 2018, pp. 205–209.

[68] C. Müller, B. Großmann-Hutter, A. Jameson, R. Rummer, and F. Wittig, "Recognizing time pressure and cognitive load on the basis of speech: An experimental study," in *International Conference on User Modeling*. Springer, 2001, pp. 24–33.

[69] A. Jameson, R. Schäfer, T. Weis, A. Berthold, and T. Weyrath, "Making systems sensitive to the user's changing resource limitations," *Knowledge-Based Systems*, vol. 12, no. 8, pp. 413–425, 1999.

[70] K. M. Feigh, M. C. Dorneich, and C. C. Hayes, "Toward a characterization of adaptive systems: A framework for researchers and system designers," *Human Factors*, vol. 54, no. 6, pp. 1008–1024, 2012.

[71] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, p. 206, 2019.

[72] M. Siebers and U. Schmid, "Please delete that! why should i?" *KI-Künstliche Intelligenz*, pp. 1–10, 2018.

[73] M. C. Lovett and L. M. Reder, "Modeling working memory in a unified," *Models of working memory: Mechanisms of active maintenance and executive control*, p. 135, 1999.

[74] J. W. Suchow and T. L. Griffiths, "Deciding to remember: memory maintenance as a markov decision process," in *Proceedings of the 38th Annual Conference of the Cognitive Science Society*, 2016.

[75] J. W. Suchow, B. Allen, M. A. Nowak, and G. A. Alvarez, "Evolutionary dynamics of visual memory," *J.of Vision*, vol. 13, no. 9, pp. 20–20, 2013.

[76] P. A. P. Moran, "Random processes in genetics," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 54, no. 1, pp. 60–71, 1958.

[77] E. Zanini, "Markov decision processes," 2014.

[78] R. D. Luce, *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

[79] K. Ogata, *Modern control engineering*, vol. 4.

[80] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19–22, 1992.

[81] K. C. Adam and E. K. Vogel, "Improvements to visual working memory performance with practice and feedback," *PloS one*, vol. 13, no. 8, p. e0203279, 2018.

[82] T. Klingberg, "Training and plasticity of working memory," *Trends in cognitive sciences*, vol. 14, no. 7, pp. 317–324, 2010.

[83] J. N. Rouder, R. D. Morey, N. Cowan, C. E. Zwilling, C. C. Morey, and M. S. Pratte, "An assessment of fixed-capacity models of visual working memory," *Proceedings of the National Academy of Sciences*, vol. 105, no. 16, pp. 5975–9, 2008.

[84] B. B. Murdock, "The serial position effect of free recall." *Journal of Experimental Psychology*, vol. 64, no. 5, pp. 482–488, 1962.

[85] J. Nakamura and M. Csikszentmihalyi, "The concept of flow," in *Flow and the foundations of positive psychology*. Springer, 2014, pp. 239–263.

[86] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[87] C. A. Pruneau, *Data analysis techniques for physical scientists*. Cambridge University Press, 2017.

[88] G. A. Terejanu *et al.*, "Extended kalman filter tutorial," 2003.

[89] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. Ieee, 2000, pp. 153–158.

[90] S. J. Julier, "The scaled unscented transformation," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 6. IEEE, 2002, pp. 4555–4559.

[91] S. Haykin, *Kalman filtering and neural networks*. John Wiley & Sons, 2004, vol. 47.

[92] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[93] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in neurorobotics*, vol. 7, p. 21, 2013.

[94] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, "Probabilistic programming in python using pymc3," *PeerJ Computer Science*, vol. 2, p. e55, 2016.

[95] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[96] pykalman, "pykalman 0.9.2 documentation," [Online] https://pykalman.github.io/, 2012.

[97] P. Abbeel, A. Coates, M. Montemerlo, A. Y. Ng, and S. Thrun, "Discriminative training of kalman filters." in *Robotics: Science and systems*, vol. 2, 2005, p. 1.

RÉSUMÉ

La mémoire de travail est la partie de la cognition humaine responsable du stockage et du traitement de l'information à court terme. Elle constitue également un goulot d'étranglement majeur dans le traitement de l'information. Dans ce travail, nous présentons deux cadres d'adaptation cognitive des interfaces utilisateur liées à des tâches. Le premier, appelé MATCHS (Memory Adaptation Through Cognitive Handling Simulations), est un système de contrôle en boucle fermée capable de suivre la capacité cognitive estimée de l'utilisateur en l'ajustant en fonction de ses performances. Le deuxième, AUHWM (An Unscented Hound for Working Memory), est développé sur les idées de MATCHS, en utilisant un filtre de Kalman "Unscented" pour le suivi, en temps réel, de la capacité cognitive humaine. Nos testons et validons les deux approches. Enfin, nous exposons les perspectives futures que les idées développées dans ce travail laissent entrevoir pour fournir une évaluation et une adaptation meilleures et plus générales des capacités cognitives humaines.

MOTS CLÉS

Adaptation de l'interface utilisateur, mémoire de travail, adaptation cognitive, filtre de Kalman.

ABSTRACT

Working Memory (WM) is the part of human cognition responsible for the short-term storage and processing of information; it is also a bottleneck in information processing. In this work, we develop strategies for providing computer systems with awareness of the user's WM limitations. We introduce two frameworks for the cognitive adaptation of task-related user interfaces. The first one, MATCHS (Memory Adaptation Through Cognitive Handling Simulations), is a closed-loop control system capable of tracking the user's estimated cognitive capacity by adjusting its value according to performance. The second one, AUHWM (An Unscented Hound for Working Memory), is developed upon MATCHS's ideas, employing an Unscented Kalman filter for the real time tracking of human cognitive capacity. We test and discuss the performance of both frameworks. Lastly, we lay out prospects of how the ideas developed here can be extended to provide better and more general assessment and adaptation to human cognitive capacities.

KEYWORDS

UI adaptation, Working Memory, Cognitive adaptation, Unscented Kalman filter.