

Influence of interaction techniques on vims in virtual environments : estimation et prédiction

Mohammad Ali Mirzaei

▶ **To cite this version:**

Mohammad Ali Mirzaei. Influence of interaction techniques on vims in virtual environments : estimation et prédiction. Traitement du signal et de l'image [eess.SP]. École Nationale Supérieure des Arts et Métiers, 2014. Français. ⟨NNT : 2014ENAM0056⟩. ⟨tel-01131426⟩

HAL Id: tel-01131426

https://pastel.hal.science/tel-01131426v1

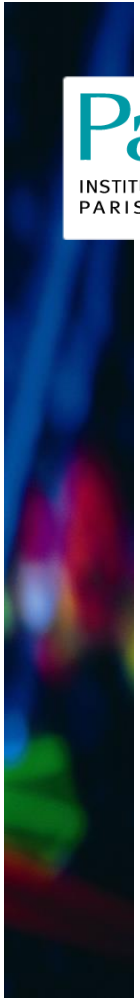
Submitted on 13 Mar 2015

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

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



HAL Authorization



2014-ENAM-0056

École doctorale n° 432 : Sciences des Métiers de l'Ingénieur

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure d'Arts et Métiers

Spécialité " Informatique-traitement du signal "

présentée et soutenue publiquement par

Mohammad Ali MIRZAEI

le 16 décembre 2014

Influence of interaction techniques on VIMS in virtual environments : estimation and prediction

Directeur de thèse : **Frédéric MERIENNE**
Co-encadrement de la thèse : **Jean-Rémy CHARDONNET**

Jury

M. Luis PASTOR, Directeur de recherche, CRVM, Université de la Méditerranée	Président
M. Daniel MESTRE, Directeur de recherche, CRVM, Université de la Méditerranée	Rapporteur
M. Guillaume MOREAU, Professeur des universités, CERMA, Ecole Centrale de Nantes	Rapporteur
M. Frédéric MERIENNE, Professeur des universités, Le2i, Arts et Métiers ParisTech	Examinateur
M. Jean-Rémy CHARDONNET, Maître de conférences, Le2i, Arts et Métiers ParisTech	Examinateur
M. Trino BELTRAN, Directeur R&D Innovation, Bouygues Bâtiment International	Invité
M. Christian PERE	Invité

T H È S E

Arts et Métiers ParisTech - Centre de Cluny
Le2i – UMR CNRS 6306

Influence of interaction techniques on VIMS in virtual environments: estimation and prediction

a dissertation presented
by
Mohammad Ali Mirzaei
to
The Department of Ecole doctorale n. 432 : Sciences des Métiers de l'Ingénieur

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computer science-signal processing

Ecole Nationale Supérieure d'Arts et Métiers
Centre de Cluny, France
December 2014

©2014 – Mohammad Ali Mirzaei
all rights reserved.

Thesis advisor: Professor Frédéric Mérienne and Jean-Rémy Chardonnet Mohammad Ali Mirzaei

Influence of interaction techniques on VIMS in virtual environments: estimation and prediction

Abstract

Understanding oculo-vestibular dynamics during sensory rearrangement in the central nervous system plays an extremely important role in better understanding human perception, and improves the technology in many engineering fields. Besides, the sensory conflict that occurs between ocular, vestibular and proprioception during sensory rearrangement at certain occasions might adversely affect the user performance in a wide variety of domains including flight/car simulators, scale-one 3D systems, large-scale displays, serious games, and so on. Therefore, knowing the condition in which the sensory conflict happens has a great deal of importance. This study aims at understanding the nature of sensory conflict by modeling and subjective studies, and the conditions in which it takes place in a synthetic environment. The results then will be used to design better navigation and manipulation interfaces in immersive and interactive Virtual Environments. A set of novel features including the area/shape of the user's COG post-exposure postural sway, the difference between LF and HF components of the sway in a frequency space, and the time of exposure will be proposed as indicators of this conflict in real-time processes. Finally, the proposed method will be used to evaluate a set of navigation interfaces. The interfaces include device-based, walking in place, speech processing, iDevice-based, and finally sensor fusion. It will be shown that naturally inspired interfaces create less conflict comparing to artificial ones. Moreover, user trajectories and inappropriate settings of navigation parameters can lead to higher conflict. In summary, to avoid user's general discomfort, the parameters of navigation need to be set in a certain range.

Contents

0 Introduction 1

1 Problem definition 3
- 1.1 Context of the thesis . . . 3
- 1.2 Objective of the thesis and research challenge . . . 9
- 1.3 Overall organization of the thesis . . . 15

2 Oculo-Vestibular dynamics and VIMS modeling 18
- 2.1 Bayesian model and stochastic process . . . 19
- 2.2 State estimation theory using Kalman Filter . . . 22
- 2.3 Neural network and cognitive process modeling . . . 28
- 2.4 Oculomotor control system: theory and modeling . . . 36
- 2.5 Binocular vision model . . . 52
- 2.6 Visual cortex anatomy and biological mechanism of Visually Induced Motion Sickness (VIMS) . . . 66
- 2.7 VIMS modeling using the Bayesian theory . . . 69
- 2.8 Conclusion . . . 72

3 Interaction techniques and implementation 73
- 3.1 Introduction . . . 73
- 3.2 Interaction in scale-one 3D VEs used for modeling and simulation . . . 74
- 3.3 Navigation principles and interfaces . . . 76
- 3.4 Manipulation principles and interfaces . . . 84
- 3.5 Navigation/manipulation interface mathematical definition . . . 93
- 3.6 Interface design . . . 101
- 3.7 Software platform and hardware architecture . . . 123
- 3.8 Conclusion . . . 129

4 Signal measurement and processing techniques 131
- 4.1 Introduction . . . 131
- 4.2 Psychological measurement . . . 132
- 4.3 Subjective measurements and statistical data analysis . . . 134
- 4.4 Signal processing approaches . . . 145
- 4.5 Conclusion . . . 161

5 Usability of manipulation interfaces in a real vs virtual environment 162
5.1 Apparatus and test setup . . . 163
5.2 Comparison between virtual and real object manipulations . . . 164
5.3 Conclusion . . . 178

6 VIMS estimation to guide navigation techniques 180
6.1 Features for VIMS prediction and estimation . . . 181
6.2 Experimental validation . . . 190
6.3 Prediction of VIMS in a real-time process . . . 217
6.4 Effect of the navigation parameters on VIMS . . . 217
6.5 Conclusion . . . 228

7 Conclusion 231
7.1 Conclusion . . . 231
7.2 Future work . . . 232

Appendix A Q1: Examinee general information (pre-exposure) 234
A.1 English . . . 234
A.2 French . . . 235

Appendix B Kennedy's Simulator Sickness Questionnaire (SSQ) 237
B.1 Questionnaire and report example . . . 237
B.2 Calculation instruction . . . 237

Appendix C List of publications 241
C.1 Journal papers . . . 241
C.2 Conference papers . . . 241

References 259

Listing of figures

1.1 Callisto immersive room during the final construction steps. . . . 4
1.2 Immersive room in operation. . . . 5
1.3 Proposed work flow. . . . 10
1.4 Final objective of the thesis and design-evaluation work flow. . . . 12
1.5 Hardware/software design and verification work-flow for navigation. . . . 13
1.6 Hardware/software design and verification work-flow for manipulation. . . . 14
1.7 Evaluation and validation process of navigation/interaction interfaces. . . . 15

2.1 Semi-circular canal and otolith End-Organs in the middle ear. . . . 23
2.2 Borah's model for the vestibular system. . . . 24
2.3 Linear system model in KF scheme. . . . 25
2.4 Kalman filter including the estimation part. . . . 26
2.5 Nervous cell and its basic elements. . . . 29
2.6 Example of Artificial Neural Network (ANN). . . . 29
2.7 Kohonen Neural Network. . . . 30
2.8 Typical neural network with input/output and hidden layers. . . . 30
2.9 Saccadic eye movement. . . . 37
2.10 Model of the saccade mechanism proposed by Young et al. . . . 38
2.11 Model of the saccade mechanism proposed by Robinson. . . . 38
2.12 Foveal eccentricity definition. . . . 39
2.13 Characteristic of the main sequence adapted from[5]. . . . 41
2.14 Neural based architecture for saccade modeling. . . . 42
2.15 Comparison of model velocity profiles to experimental data. . . . 44
2.16 Young's model of smooth pursuit. . . . 45
2.17 Robinson's model of smooth pursuit eye movement. . . . 45
2.18 Smooth pursuit mechanism of the eye movement control system. . . . 47
2.19 Step response to a sudden movement of the target object. . . . 48
2.20 Eye movement due to Vestibular Ocular Reflex. . . . 48
2.21 VOR simplified model along with neuron pathway. . . . 49
2.22 Complete VOR model provided for simulation purpose. . . . 49
2.23 (a) Small and large vergence, (b) vergence biological model. . . . 50
2.24 Saccade-vergence control mechanism. . . . 51
2.25 Modified control mechanism based on vergence. . . . 51

2.26 Neural path of the oculomotor mechanism. . . . 52
2.27 Complete monocular vision mechanism. . . . 54
2.28 Simplified version of the eye movement mechanism. . . . 56
2.29 Neural pathway of the binocular vision system. . . . 57
2.30 Complete model for binocular vision simulation with Flocculus comportment. . . . 58
2.31 Simplified binocular vision neural model. . . . 59
2.32 Simplified binocular vision model. . . . 60
2.33 Eye movement control system for a single eye. . . . 61
2.34 Difference between the L/R eyes and the object position at time $t = 0^+$. . . . 62
2.35 Output of the Flocculus network without movement. . . . 63
2.36 a) Original pattern $\omega = 0$ °/s, b) pattern in motion for $\omega = 20$ °/s c) 60 °/s. . . . 64
2.37 Output of the Flocculus network for a pattern in motion. . . . 65
2.38 (a) Different sections of a brain, (b) detail of sub-areas associated with the visual cortex. . 66
2.39 Hypothalamus position in the brain. . . . 67
2.40 Connection between the hypothalamus and the visual cortex. . . . 67
2.41 Vagus nerve activation and biological mechanism of nausea. . . . 69
2.42 a) Modified MS model proposed by Oman for the sensory conflict theory, b) dynamic model of the body, sensory organs and their signaling with the CNS. . . . 71

3.1 Example of laser based tracking system. . . . 80
3.2 Examples of navigation interfaces based on physical locomotion. . . . 81
3.3 a) Task definition of the steering technique, b) navigation in VEs by gesture-based steering technique. . . . 81
3.4 IR image analysis for skeleton detection. . . . 81
3.5 Example of body gesture analysis for five actions. . . . 82
3.6 Output of the gesture detection processing unit. . . . 82
3.7 WIP interface designed by swing wires for sensory incorporation. . . . 83
3.8 Cyber-walk on an omnidirectional treadmill. . . . 84
3.9 Hand tracking by skeleton detection. . . . 86
3.10 Hand gesture analysis and coding. . . . 86
3.11 The Go-Go method. . . . 88
3.12 a), b) Object selection with a virtual pointer, c) ray-casting technique. . . . 89
3.13 Egocentric object manipulation using the Head Crusher technique. . . . 91
3.14 Virtual image plane. . . . 91
3.15 a) Aperture technique and b) selection-orientation planes. . . . 92
3.16 Voodoo Dolls object manipulation. . . . 92
3.17 The object position is defined as distance d and directions α, β in the user-centered coordinate system. . . . 94
3.18 Rotation of θ degrees around vector $\vec{u}$ in 3D space. . . . 97
3.19 Rotation angle in Cartesian coordinates with Euler representation. . . . 98
3.20 Reorientation during navigation. . . . 100
3.21 Translation and rotation in 3D space. . . . 101
3.22 Different parameters of a navigation device. . . . 103

3.23 Schematic representation of the depth-disparity relation. . . . 107
3.24 Example of a human walking signal with a) constant, b) variable steps and distance from the sensor. . . . 108
3.25 IR signal generation using body skeleton. . . . 109
3.26 Scale variation during walking in a real-scale 3D model. . . . 110
3.27 Mechanical restriction for the elimination of involuntary body movement. . . . 110
3.28 Interpretation of a gait signal for navigation command generation. . . . 111
3.29 Block diagram of a navigation system by walking gait analysis. . . . 113
3.30 Scheme for microphone-array sound capturing and speech processing in a noisy acoustical environment. . . . 114
3.31 Different components of speech processing with a Kinect. . . . 116
3.32 Hardware infra-structure for speech processing and data transmission. . . . 117
3.33 Left) vision configuration, right) indicators in an E-shape arrangement. . . . 119
3.34 Microphone-array configuration. . . . 119
3.35 a) Head tracking system and b) sensor fusion configuration. . . . 120
3.36 Sensor fusion scheme for combining audio and vision sensor data. . . . 120
3.37 Magic bracelet interface for manipulating and interacting with a 3D object in VEs. . . . 122
3.38 Navigation/manipulation with iDevice. . . . 122
3.39 Overall scheme for the system development and software platform. . . . 124
3.40 Extended software platform with the graphic engine. . . . 125
3.41 Development platform. . . . 125
3.42 Simulation with two screens. . . . 127
3.43 Simulation with four screens on a single screen display. . . . 128
3.44 Example of Javascript code for a basic navigation. . . . 129
3.45 Our CAVE system. . . . 130

4.1 Schematic diagram of iiVR platform data logging during a navigation/manipulation task. 134
4.2 Different psychophysiological measurements. . . . 139
4.3 (a) Electroencephalography, (b) Techno-concept center of gravity board, (c) Sense-Wear Pro2 GSR sensor worn by a test participant. . . . 141
4.4 (a) Example of metallic sensors, (b) wireless heart rate sensor, (c) blood pressure sensor, (d) typical ECG signal for a healthy person. . . . 142
4.5 Sensor network for blood pressure measurement. . . . 143
4.6 (a) Minimum components of a digital pill package, (b) bio-component of a digital pill, (c) digital pill iPhone application and example of output of the transmitter. . . . 144
4.7 Single-channel EEG signal. . . . 146
4.8 Single-channel EEG signal in the frequency domain. . . . 147
4.9 EEG probe location on the scalp for amplitude map extraction. . . . 148
4.10 Example of a multi-channel EEG signal. . . . 149
4.11 Sub-categories of an EEG signal. . . . 150
4.12 Few examples of wave patterns in an EEG signal. . . . 150
4.13 Multi-resolution analysis with wavelet. . . . 154
4.14 Time-frequency EEG signal analysis. . . . 157

4.15 Example of an EMG signal. . . . 157
4.16 Time-frequency representation of the EMG signal shown in Fig. 4.15. . . . 158
4.17 (a) Brain EEG amplitude map, (b) 3D representation of a multi-channel EEG signal, (c) equivalent dipole source locations and mean, and individual scalp maps for five IC clusters, 3D dipole source locations and their projections onto average. . . . 160

5.1 Visualization hardware architecture and software platform designed for the experiments. 163
5.2 iiVR and JVM joint platform for real-time synchronous experiments. . . . 164
5.3 Object manipulation in (a) a virtual and (b) a real environments. . . . 165
5.4 Wireless EMG sensor, measuring and data logging stand. . . . 167
5.5 Precise measurements setup for sensor positioning. . . . 167
5.6 Three arm positions (left), data recorded by the BIOPACK sensor in the time and frequency domains (right). . . . 168
5.7 The biceps muscle is used for EMG data logging (a), EMG sensor position for measurement (b). . . . 169
5.8 EMG signals recorded in the real/virtual environments. . . . 170
5.9 Calculation of the length for a given curve. . . . 170
5.10 Spatial representation of an object displacement in the virtual and real environments. . . . 172
5.11 Length of the movement trajectory in the virtual environment for the three cubes with the order shown in Fig. 5.10. . . . 173
5.12 Amount of rotation for each cube in the real and virtual environments. . . . 173
5.13 Time of task completion in the virtual and real environments. . . . 175
5.14 Time-frequency representation for object manipulation in the real and virtual environments. . . . 175
5.15 Time-frequency representation of the spectrum in the real and virtual environments and the difference. . . . 176
5.16 Time-frequency components of the hand movement with/without an object. . . . 176
5.17 Four cases during interaction with an object in the real (a) and virtual environments (b). . . . 177
5.18 Level of satisfaction in both environments. . . . 177
5.19 Sub-score of object manipulation for the virtual environment. . . . 178
5.20 Level of involvement and distraction of object manipulation in the virtual environment. 179

6.1 Dynamic model of the body sway during a quite upright stance. . . . 182
6.2 Inverted pendulum model and closed-loop control scheme of a quiet stance. . . . 183
6.3 (A) Forward/backward and left/right signals for a marginally stable state, (B) forward/backward and left/right signals for a stable state, (C) COG area created by projecting the forward/backward and left/right corresponding values onto XY plane. . . . 186
6.4 Body F/B postural sway signal for the a) marginally stable, b) unstable, c) stable states in the time domain and associated spectrum in the frequency domain. . . . 187
6.5 a) Experiment inside the CAVE, b) path indicator and environment pattern. . . . 192
6.6 Example of F/B and L/R sways signals for pre and post exposures. . . . 193
6.7 Variation of the COG area and the shape during the experiment (1.d), pre-exposure (1.a), at the sickness onset (1.b), post-exposure (1.c), and correlation between the pre-exposure (2), post-exposure (3), at the sickness onset (4) areas and the SSQ total score. . . . 196

6.8 Correlation between pre_sick_area and SSQ1. . . . 197
6.9 Correlation between pre_post_area and SSQ2. . . . 198
6.10 Correlation between sick_post_area and SSQ3. . . . 198
6.11 Relation between the area dilation and the VIMS score. . . . 199
6.12 Shape variation of the COG area before and after the sickness onset increase. . . . 199
6.13 Correlation between the COG shape and the VIMS score. . . . 200
6.14 Frequency components of the F/B sway signals for the pre, post-exposures and before the sickness onset instances. . . . 202
6.15 VIMS grows when the difference between the HF and LF components of a postural sway increases. . . . 202
6.16 Correlation between Δf_1 and SSQ1. . . . 203
6.17 Correlation between Δf_2 and SSQ2. . . . 203
6.18 Correlation between Δf_3 and SSQ3. . . . 204
6.19 (a) Real test setup for the experiment and (b) calibration pattern for the calibration of the SMI system. . . . 205
6.20 SMI eye tracking sensor. . . . 206
6.21 (a) Eye position and pupil counter estimation, (b) gaze position. . . . 206
6.22 (a) Path planning for the experimental test, (b) stimulation pattern. . . . 208
6.23 (a) Left/right eye position and difference between the two eyes during translational movement, (b) difference between left/right movements immediately after rotation of the scene. 209
6.24 Difference between the left and right eyes during navigation with respect to the reference lines, (1-2) during translation and right before rotation, (3-6) immediately after rotation. . 210
6.25 (a) Right and (b) left eyes position, (c) difference between the two eyes and (d) rotation command versus time. . . . 212
6.26 Eye-blink and residual signals. . . . 213
6.27 Sub-score average across the subjects. . . . 214
6.28 Correlation between the SSQ and the difference during rotation. . . . 214
6.29 Analysis of the difference between the two eyes during saccade and fixation. . . . 215
6.30 Time constant of VIMS and comparison with the time constant of MS for 9 subjects who participated in the experiment of feature 2. . . . 216
6.31 VIMS prediction in a real-time application. . . . 217
6.32 Implantation of a VIMS predictor in a real-time process. . . . 218
6.33 a) Mechanism of the speed controller and sequence of its operation, b) Fly-stick and navigation sub-task assigned to each button. . . . 219
6.34 Navigation based on natural gesture. . . . 220
6.35 (a) Test parameters, path planning for navigation inside a 3D VE, (b) real path setup inside a scale-one 3D display. . . . 222
6.36 CNS model for explaining the reason of VIMS by the theory of sensory conflict. . . . 223
6.37 Experimental design for studying the effect of the distance on VIMS. . . . 224
6.38 Frequency components analysis of the COG signal. . . . 225
6.39 Post-exposure sickness score associated with paths 1 and 4. . . . 226
6.40 Spectrum analysis for NGBNI. . . . 228
6.41 Spectrum analysis for DBNI. . . . 229

6.42 (a) VIMS-rotational velocity, (b) VIMS-translational velocity, (c) VIMS-distance from the virtual barrier. 230

0 Introduction

The current thesis supervised by Prof. Frédéric Mérienne, Dr. Jean-Rémy Chardonnet and Christian Père started on mid-November 2011 and was conducted under the FUI Callisto-SARI project. Institut Image (Arts et Métiers ParisTech/Le2i) and Cité des Sciences et de l'Industrie (Paris) were two main centers who hosted this thesis. The final implementation of the software platform, hardware architecture and HMI interfaces were demonstrated in the immersive room located in the Cité des Sciences et de l'Industrie for Bouygues Construction, the Callisto-SARI project team, and the industrial partners in November 2013. Different development steps, related technologies, in-house provided technologies with their engineering principles were presented to the project team in August 2013.

The final goal of the thesis was to find a practical solution for the well-known sensory conflict problem in a virtual environment. Then, based on this solution, a set of efficient interaction (navigation / manipulation) interfaces was proposed employing different technologies. As it will be seen, the sensory conflict in the oculo-vestibular dynamics has been a very fundamental research question and affected several domains including engineering, aviation, emerging technologies, car industries and so on, and considered as a serious industrial challenge. Reliable research results will influence directly technologies such as aviation (flight simulators, drone land base control, unmanned vehicle control and navigation), car industry (car simulator, manufacturing, assembling and disabling of comportment), display systems, robotics and

training. In addition, it can improve the quality of cyber products such as games, HCI and automation industries.

This problem has been inquired by different teams and research groups all across the world. Researchers have studied the problem from various points of view including psychology, psychophysiology, neuro-science, computer vision, Man Machine Interface (MMI), Human Computer Interaction (HCI), user study, biology, robotics and telecommunication, and so on. We will study the sensory conflict problem from modeling, signal processing and computational neuro-science perspectives, however the main focus will be on signal processing. Simply, the modeling and experimental data will be recorded and analyzed to validate our theory and hypotheses. However the proposed modeling solution should be approved by practical experiments. Therefore, one chapter will be dedicated to the modeling and simulation, and a huge amount of mathematical equations regarding computational neuro-science models were studied.

A nearly entire display system was developed on Windows platform and NVidia Quadroplex GPUs. To simplify the development procedure for newcomer developers and future researchers, all the GPU Kernels, C++ code, MATLAB engine, wireless network telecommunication and interfacing toolboxes were wrapped under Javascript in Institut Image's software platform which makes the development very fast and easy. Important efforts in debugging, software tests were made for building such a user friendly and handy platform. Nearly two third of the thesis was spent on practical development preparing above-mentioned aspects. Practical aspects and development challenges will not be deeply explained in the manuscript. As a result, this manuscript provides a solution for the proposed research question and scientific challenges.

Last, the development, related technologies and design aspects were presented in the form of posters, technical notes, conferences and journal papers as well as presentations. A list of publications is included at the end of the manuscript.

The sole aim of this manuscript is to present the sensory rearrangement of the oculo-vestibular system, the sensory conflict condition and the benefits from the research achievements to design better interaction interfaces. Hence a major part of this manuscript is dedicated to sensory conflict and the so-called visually induced motion sickness. A summary of the development and related algorithms are explained in one dedicated chapter.

1 Problem definition

1.1 Context of the thesis

This thesis is part of the Callisto-SARI project: a collaborative research and development project led by a French company (Bouygues Bâtiment International) and bringing together eight partners* (industrial and research laboratories) from the fields of scale-one 3D visualization and virtual reality, architecture, and other scientific and engineering research purposes. The project aims at constructing a novel immersive room and developing different software tools for engineering and research uses. The project is focused on 3D building modeling and environmental simulation as the first objective and other industrial and scientific usages later as the second purpose. For that reason, a 3D model of a building will be used in the entire experiments and evaluation processes throughout this thesis.

The construction domain is currently undergoing many changes: environmental (new regulations, energy constraints...) and industrial (better processes). The challenges require moving from the use of 2D

*The projet consortium comprises: one large group, Bouygues Bâtiment International (BBI) (Callisto leader), two small companies, Immersion S.A., Art Graphique et Patrimoine (AGP), two half-public institutions, Universcience (Cité des Sciences et de l'Industrie), le Centre des Sciences et Techniques du Bâtiment (CSTB), and three research laboratories, Arts et Métiers ParisTech (AMPT - Le2i), Ecole Centrale Paris (ECP), le Laboratory des Usages en Technologies d'Information Numérique - Paris 8 (LUTIN).

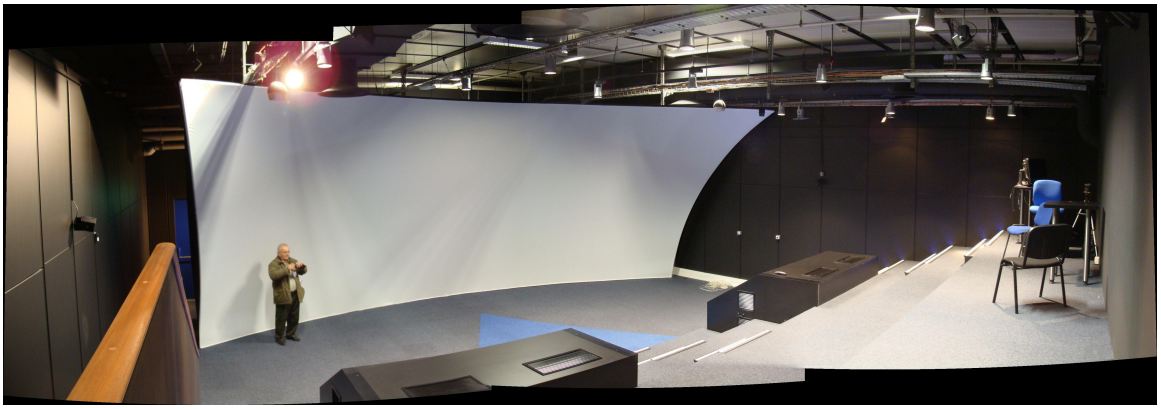


Figure 1.1: Callisto immersive room during the final construction steps.

plans to CAD (Computer Aided Design), 3D Building Information Model (BIM). The BIM includes semantics for the construction process (as an example: structure, air conditioning / ventilation, mechanical, electrical, piping, etc.) and data for simulations (materials/structure resistance, energy consumption, thermal calculations, lighting, acoustic simulations...). These issues must be addressed throughout the construction project but mainly at the beginning during the design phase to fulfill the client's requirements, during construction to anticipate technical constraints on the construction site and during the maintenance phase to control the building.

Introducing 3D models in the construction process is a main way to do the following checks before entering into the construction phase of the project:

1. Test the 3D mockup virtually and correct a construction project before the realization;
2. Reduce the production costs by the construction of a real mockup;
3. Avoid mistakes on site by a better project management which in turn leads to less material wastes.

1.1.1 Objectives of the Callisto project

The Callisto-SARI project came about for two main reasons: 1) the need to communicate or share the efficiency of using 3D visualizations in the building production and 2) the need to aware constructors and general public of 3D CAD model usages in the pre-inspection of a construction process and make use of it in cost estimation and optimization. A list of functional requirements was suggested by the client of the project to be considered during the research and engineering design.

The main objectives of the Callisto-SARI include:

1. To allow people to visit interior/exterior spaces of full-scale buildings with the physical properties, acoustic and light rendering in real-time and in an innovative immersive system;



Figure 1.2: Immersive room in operation.

2. To give the users, visual and acoustic sensations close to a real-life visit, and then to enable them to take decisions concerning the construction project (and to avoid building a scale-one mockup);
3. To navigate inside the building, interact with the elements, modify them during the visit, and immediately see the modifications results;
4. To provide a procedure in order to facilitate the visualization of scientific contents in 3D, open up the system for scientific research and give access to university professors and partners laboratories;

During three years of the project development (2010-2013), the following research outcomes and engineering developments have been achieved:

1. A lighting model for real-time applications has been derived and simulated (for natural and artificial light sources);
2. An acoustic model for real-time applications has been developed and simulated (processing spectral and spatial 3D sound field informations);
3. The visual spectrum of the building material was extracted and simulated;

4. Natural and artificial lighting simulations for a dynamic visit of heritage sites were done;
5. Progress in 3D BIM-IFC models toward building detailed modeling and realistic rendering;
6. Combination of laser-based and photo-based 3D digitalization and 3D digital building modeling;
7. Real-time, full-scale, constant geometrical visualization of a 3D building model;
8. New interaction modalities with 3D models adapted for architecture and construction usages.

The main cutting edge achievements of the project include:

1. The Callisto-SARI room which is well equipped with an innovative, interactive, visually and acoustically immersive system;
2. Software applications for:
    - Optimum light and/or sound-process integration
    - Link between spectral simulation applications (light, material, and sound) and real-time applications
    - Link between digital CAD model and hyper-realistic model rendering (engineers' 3D model $\rightarrow$ promotion using computer graphics)
    - Adapted 3D navigation/interaction based on gesture
3. A lower affordable VR (Virtual Reality) mobile system to allow a wider use of these techniques in the construction trade.

1.1.2 Operational immersive room

A room in the Cité des Sciences et de l'Industrie, a science museum located in the north of Paris, previously dedicated to 3D films, has been renovated to install the immersive system. At the end of the project, the public will have access to this room (50 people guided by a demonstrator) and, outside public opening hours, industrialists or scientific researchers will be allowed to use it (five people in the center of the immersive room for a project review use, one person for immersive/interaction use). The room has been studied considering these three possible types of uses.

The immersive system includes:

1. A large hemispherical screen (10 m wide, 4 m high) which helps immersive sensations avoiding angles (as in CAVE systems) and with good visualization of height for full-scale, building interior visits (see Fig. 1.1);

2. A GPU-based cluster projection system composed of eight HD video projectors, active stereo, and a graphic server (see Fig. 1.2);

3. An acoustic ambisonic type system, allowing advanced sound propagation with speakers installed all around the room, driven by a sound PC server synchronized on the graphic server. The screen is acoustically transparent.

1.1.3 Requirements of the Callisto-SARI project

The current PhD thesis was proposed by Bouygues to fulfill the 3rd objective of the project, as explained in section 1.1.1. As mentioned in the third objective of the project, the immersive room must provide comfortable, user friendly and real-time navigation/interaction interfaces for the visitors. In this regard, the management team provided a demonstration video along with a document to hand in the operational requirements. Table 1.1 shows the short list of the operational requirements provided by the client (Cité des Sciences et de l'Industrie museum).

The interaction system should be based on gesture (for example based on Kinect or other technologies) but the system should allow the connection of other possible devices. However interaction and navigation in 3D models in an immersive environment brings out technical challenges: how to implement user friendly interaction systems, how to deal with well-known problems of cyber-sickness such as headache, nausea and general discomfort when working in an immersive environment? In other words, how does one clearly identify the origin of discomfort and sickness, and how to avoid them by a proper interaction system design, technology selection, and parameter settings?

Different challenges have been discussed in the literature such as usability, precision, delay between the command and the visual feedback, cyber-sickness, and so on. Among the challenges, cyber-sickness, or visually induced motion sickness (VIMS), is more important than the others, because most of the visitors have generally some ideas about smartphone applications and computer games but never experienced immersion before, which results in severe sickness sometimes. Sometimes, a high rank manager can be among the visitors and has never been immersed. The first immersion may adversely affect this type of visitors which in turn may change their decision in financing a specific project. Therefore, VIMS or in general cyber-sickness minimization was set as a target parameter in all the navigation/interaction interface design by the project team.

This requirement will play a key role in the interface design and development technology selection throughout the thesis.

Function	Requirements
Navigation	1. Traveling inside VEs (start, translation/rotation, stop) 2. Continuous (natural walking)/discontinuous (consecutive by a command from navigation devices) and path planning 3. Path indication (start/stop definition, placing bench markers and user guides) 4. Staircase climbing up/downward 5. Multimodal navigation devices with selection possibility
Interaction	• Manipulation of an object 1. Object manipulation menu (initiate, set and reset) 2. Place a new object in the scene 3. Select, move and replace an object that already exists 4. Rotate an object in place 5. Allocate an action to the selected object • Interaction with the environmental properties 1. Display menu (object, multimedia and lighting manipulation) 2. Scene, furniture and interior decoration display, placement, and modification 3. Multimedia (display, move, play, stop, pause, next and previous in the sound data base)

Table 1.1: Operational requirements provided by the project team.

1.2 Objective of the thesis and research challenge

The preliminary study on different technologies, the scientific challenges in the field (immersive VR) and different subjective and objective evaluation approaches were reviewed in the first phase of the project, knowing the requirements of the industrial partner (Bouygues). Based on this review, the final objective of the thesis is to find distinct answers for the following questions:

1. How visually induced motion sickness (VIMS) can be described in terms of navigation/interaction parameters in real-time?
2. Can we use VIMS as a criterion to evaluate interaction/navigation interfaces? How can we validate our findings with psychophysiological measurements analyses?
3. What is the main reason of VIMS in a synthetic environment?
4. How can the display parameters affect VIMS?
5. How does the achievement of the VIMS study influence the computational methods in other research domains?

All these questions were combined in a main question called "research challenge" in this thesis. In fact, the above-mentioned questions will be well addressed by answering this main question. The proposed answer as well as the strategy and the methodology selected to verify this answer describe our "contribution" (as shown in the next section).

The two main research questions, one for navigation and one for manipulation, are:

1. How can we navigate inside a real-scale virtual environment of a building? $\Rightarrow$ Visually Induced Motion Sickness
2. How to manipulate VR objects in a real-scale virtual environment of a building? $\Rightarrow$ Manipulation in the Virtual Environment (VE) versus the Real Environment (RE)

We sum up these questions by this main research question that will be the research challenge:

> *"Why does inappropriate selection of the navigation/interaction interface or parameter settings lead to VIMS and divergence between real and virtual environment?"*

1.2.1 Contribution

Definite findings during nearly a century of research were summarized in a theory proposed by two neuroscientists, Reason and Brand. The theory is known as "sensory conflict" in the literature and scientific

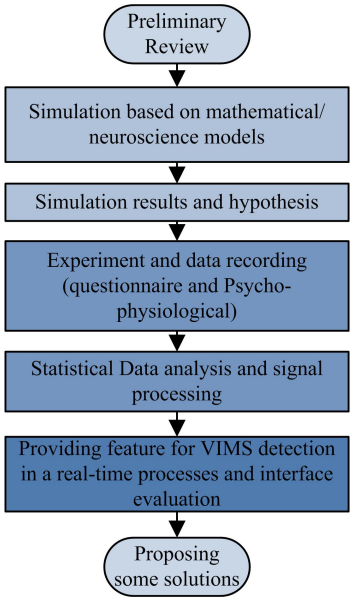


Figure 1.3: Proposed work flow.

documents. It is known as the "sensory rearrangement theory" alternatively in some literatures too. It states that "*Motion sickness is a self-inflicted maladaptation phenomenon which occurs at the onset and cessation of conditions of sensory rearrangement when the pattern of inputs from the vestibular system, other proprioceptors and ocular system is at variance with the stored patterns derived from recent transactions with the spatial environment*" [158].

We will see that VIMS is defined as the difference between the estimation coming from the nervous system and the measurements reported by the sensory organs. This difference in mathematics is called residual and in neuroscience is called motion sickness (MS) or visually induced motion sickness. Therefore, the reduction of VIMS diverse effects is equal to the minimization of the residual signal. To this end, first, the theoretical model will be used to show what the meaning of MS/VIMS is and how it emerges by simulation and mathematical/computational neuroscience models. Then, different experiments will be established to validate the theoretical achievements. Using the theoretical achievements helps us to validate our findings with even few participants. For that, the following work flow (Fig. 1.3) will be followed to find an appropriate answer for the research challenge and validate our answer by both objective and subjective studies. The work flow has three distinctive parts: theory and simulation, experiments, and data analysis using statistical methods and signal processing approaches.

Simulation and theory: human perception will be modeled using the Bayesian theory and Kalman model. Then, the human vision system as the most important organ in the VIMS study will be simu-

lated and different aspects will be considered. Since VIMS involves different sensory-motor functions, the body equilibrium will be selected and analyzed in the time and frequency domains. Some features for the VIMS detection will be proposed based on simulation.

Experiments: several experiments will be set up to see how much effective are the selected features in the real process. Two types of data, self-reported questionnaires as well as psychophysiological signals are recorded during the experiments.

Data analysis: Data is analyzed using signal processing approaches and statistical methods, and correlated to find the most effective feature for practical applications.

The main contribution of this thesis includes the modeling and analysis of the binocular vision from a computational neuroscience perspective, and providing a solid theory for a better understanding of VIMS using some objective features such as the Center of Gravity (COG) area and its shape, the difference between low and high frequency components of the body sway, and the difference between two eye movements, which make the evaluation process much more reliable than subjective studies. Unfortunately, due to the fact that subjective studies are partly based on self-report questionnaires, it can be easily biased which makes the research results totally unreliable if the number of participants is not enough. We hope our proposed method can give higher reliability to the results by multiple checking.

1.2.2 Hardware/software design and verification work-flow

A design/verification work-flow is proposed to facilitate easier and faster progress through different stages of design and development with respect to the schedule of the project time-line. The primary aim of the thesis is to fulfill the project requirements by designing a novel set of interfaces for a large scale graphic control. Then, the effectiveness of the interfaces must be proven by conducting a subjective surveillance. Besides, exposing a subject into a large scale video display is firmly connected to the oculo-vestibular research. Therefore, the secondary aim is to study the effect of different parameters of navigation/manipulation interfaces on the level of VIMS and presence. To this end, first, different technologies are selected to develop different navigation/interaction interfaces for VR applications, as shown in Fig. 1.4. Then, a technological approach is selected for interface design regarding three main requirements: operational requirements provided by the project team, user requirements as explained in section 1.1.3 and provided by the partner research labs and oculo-vestibular constraints.

Interfaces are split into two groups: 1- navigation interfaces, 2- manipulation interfaces which will be detailed below.

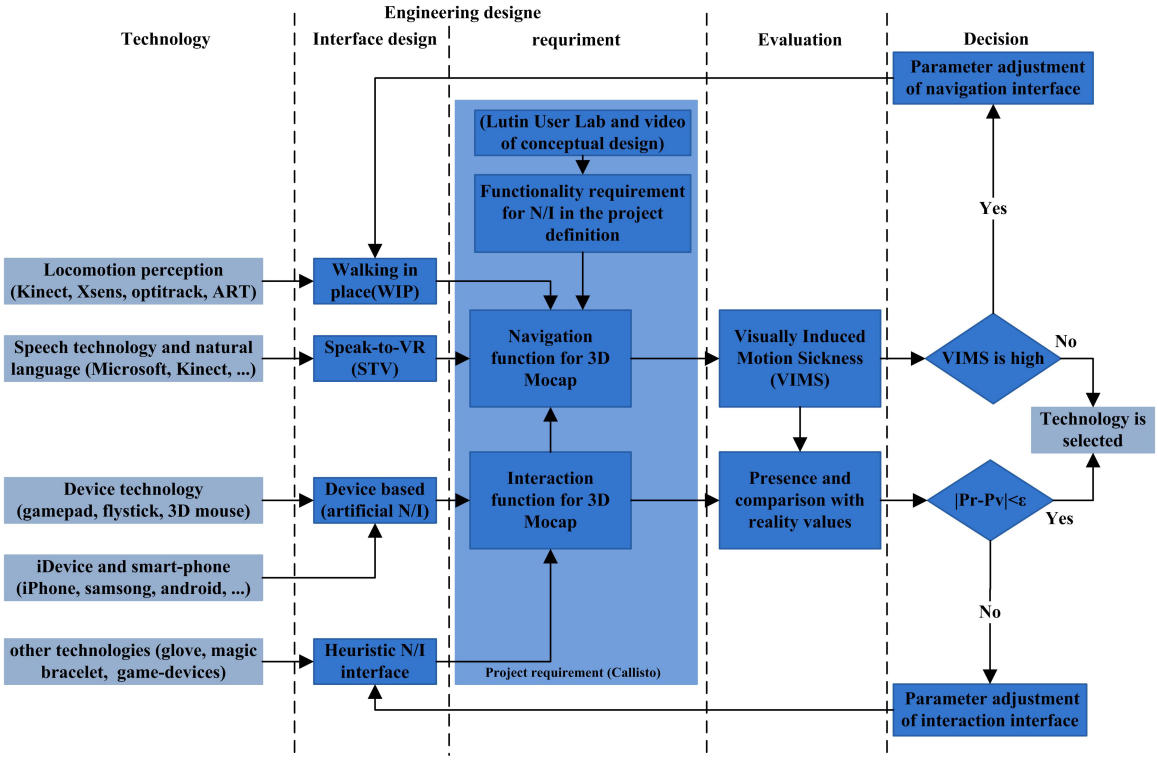


Figure 1.4: Final objective of the thesis and design-evaluation work flow.

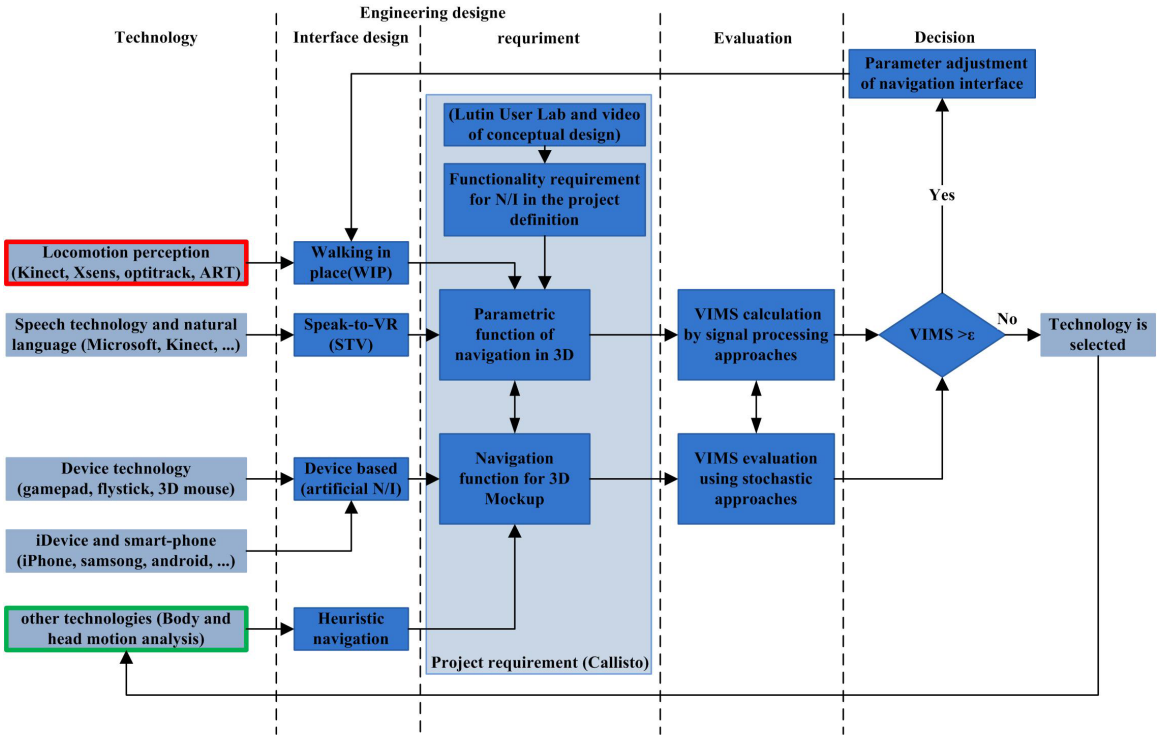


Figure 1.5: Hardware/software design and verification work-flow for navigation.

Navigation interface

Navigation interfaces are again divided into two main categories. The first category is called gesture-based interfaces (natural) which are designed by the analysis and the classification of the user's body locomotion. The second category is called device-based interfaces (artificial). A set of necessary GPU kernels and drivers are developed and test points are created after designing an interface. However, to evaluate the real performance of the user and to verify the design approach, we need a test-bench. For that reason, a test-bench is established in a scale-one 3D display system to investigate the user performance during a navigation/interaction task inside a given 3D model, and further analysis is made to better understand the cons and pros of the design and pave the road for the study of the oculo-vestibular dynamics and state estimation mechanism. The results of this investigation then are validated by the analysis and interpretation of the recorded signals from the process and the user. If the result of the evaluation proves that the level of VIMS is low then the interface is selected otherwise, the parameter is readjusted and the test is repeated again in a second round or more. Sometimes, input devices are combined under a sensor fusion scheme to improve the performance while decreasing the level of VIMS. A summary of this explanation is illustrated in Fig. 1.5.

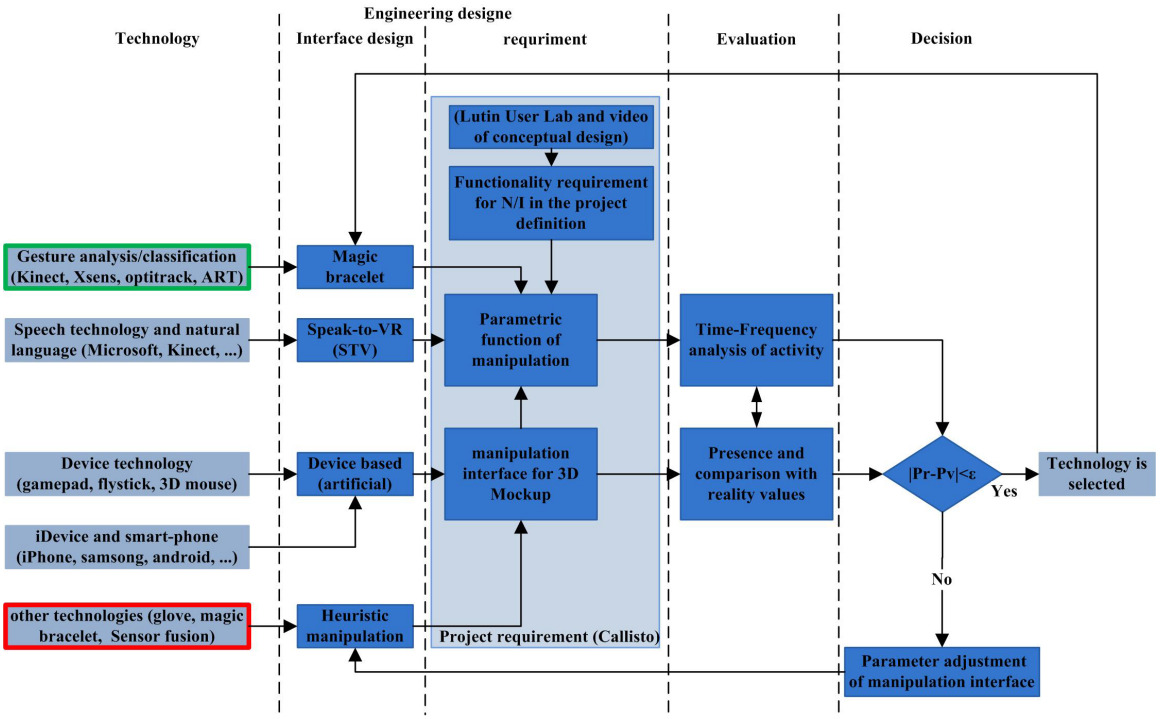


Figure 1.6: Hardware/software design and verification work-flow for manipulation.

Manipulation interfaces

Two categories of manipulation interfaces are selected, exactly the same as navigation interfaces, however the task definition and the design of the interfaces are different. The same test-bench as for navigation interfaces is re-established to investigate the manipulation parameters, however with different oculo-vestibular criteria as will be detailed later in chapter 3. Similarly, the design will be verified, evaluated and finally modified to fulfill the user and design constraints. A design and verification work-flow is proposed for manipulation interfaces as illustrated in Fig. 1.6.

1.2.3 Interface validation process

When the psychological measurements are collected and the statistical analysis is made (first method), two other auxiliary methods, i.e., theory/simulation (second method) and a stochastic signal processing approach (third method), are used to validate the accuracy and correctness of the results (see Fig. 1.7, where, F_M, F_{PS}, and F_{ST} represent the results from simulation, psychophysiological, and psychological data analysis, respectively). The signal processing approach as the second method includes simple feature extraction, classification and deterministic/stochastic signal processing techniques. The first and second meth-

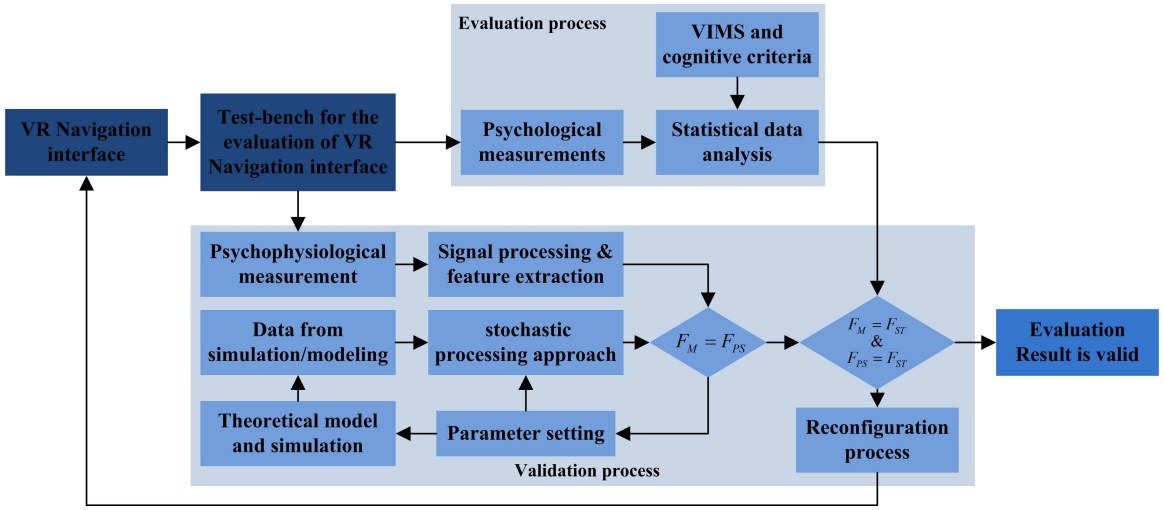


Figure 1.7: Evaluation and validation process of navigation/interaction interfaces.

ods are highly linked in the validation process as seen in Fig. 1.4. Usually, two types of data are analyzed: synthetic data which comes from simulation and real data from the experiment. The synthetic data is generated by mathematical models in simulation and the settings of initial conditions close to the real experiment and similar assumptions. The synthetic and the recorded signal are processed by the same processing approaches. The outcome of the modeling and simulation is used to evaluate the processing results of the recorded signal. If the results from the theory, signal processing and statistical analysis are in accordance, it means the results are valid and reliable. Finally, it will be shown that the results from self-reported questionnaire analysis and the signal processing results are highly correlated. This process will provide alternative quantitative criteria (a set of features) which can be used instead of self-report measurements.

1.3 Overall organization of the thesis

The thesis is organized in six chapters as follows to explain the details of development and the scientific study:

Introduction: The introduction is a one page to explain the overall structure of the thesis and how the thesis has been completed, what platform has been developed during the thesis, and the achievements of the thesis including technological and scientific knowledge as well as publications.

Chapter one: A short summary of the Callisto-SARI project is introduced. In this short introduction, the basic development idea and the final product are briefly explained. The project requirements, research objective and exact scientific question are discussed in this chapter. The scientific contribution of the thesis

is detailed and the outcome of thesis is discussed. Then, a work scope including the design and development plan, a review over concurrent work, and the prospective research plan to find an appropriate answer to the scientific question and solve the industrial challenge are proposed. Visually induced motion sickness during navigation inside the virtual model is the scientific challenge.

Chapter two: The neural model of the oculo-vestibular dynamics based on neural network and Bayesian formulation will be explained briefly. Since theoretically previous research in VIMS conducted based on Bayesian and Kalman models, the principle of their modeling technique will be summarized. Binocular vision will be modeled by splitting different eye movements into five categories and deriving a deterministic model for each category. The final binocular model will be made by combining five sub-models and taking into account ganglion nuclei in the model. Finally, a neural model will be integrated in the model of the binocular vision to make any analysis possible. The study of the theoretical background, modeling and simulation will be used to construct a solid background for the validation process.

Chapter three: First, a review over different navigation and interaction interfaces will be presented. A software platform was developed to code any navigation and interaction application in Virtual Environments (VEs) easily and fast. Different components, the hardware and data communication infrastructure, the configuration and display system will be explained, and their connection with the software platform will be shown. The system has two operation modes: simulation and scale-one display. The configuration for each mode will be detailed. The development language and its instructions will be summarized. There are different versions of real-scale 3D displays. We did the navigation/interaction experiments in a CAVE system which is widely used in VR studies because at the time of development, the Callisto immersive room was under construction. The hardware compartments of our CAVE system will be briefly introduced. Finally, five different technologies will be introduced which will be used in the development of interaction/navigation interfaces. We will use these interfaces both to complete our studies and assess the performance of the interfaces subjectively and objectively later in chapters 5 and 6.

Chapter four: Self-report questionnaires and psychological measurements will be explained at the beginning of the chapter. We will briefly list different valid questionnaires and explain how to calculate scores from the questionnaires. Later, these scores will be used to rate and compare different navigation/interaction interfaces using statistical analysis. Few approaches for statistical data analysis will be explained. Different psychophysiological signals along with their measurement tools and sensors will be introduced after statistical analysis. Some signal processing approaches will be used in the following chapters. These approaches with a couple of examples will be summarized and explained in more detail mathematically. The final goal is to see how the features from psychophysiological signals are correlated with

the variation of the scores (calculated by the questionnaires) to be able to use these features alternatively as detection criteria in a real-time process.

Chapter five: Manipulation (selection, movement and placements) with virtual and real objects will be compared by a presence questionnaire and Electromyography (EMG). A standard method will be presented to compare the user performance and assess the usability of interaction interfaces via only measurements. The final achievement of this part is the standard time-frequency patterns for the comparison and evaluation of different object manipulation mechanisms.

Chapter six: Three features including the COG shape, its area and the difference between frequency components can be used to detect VIMS in a real-time process. The analysis of the features and their effects on VIMS will be studied in more detail. Kennedy's SSQ (Simulator Sickness Questionnaire)[98] as psychological and the COG (Center Of Gravity) as psychophysiological measurements are used in this study. The difference between the two eyes during navigation is another useful feature for real-time processes.

The practical experiments and the corresponding results will be discussed in this chapter. The effect of the translational and the rotational velocities on the level of VIMS will be studied relaying only on the binocular disparity and Kennedy's SSQ. Moreover, translational and rotational movements will be compared using the difference of left and right eye movements. The effect of the distance from a virtual barrier and the translational velocity of navigation on VIMS will be discussed in a second experiment. Natural versus device-based navigation interfaces will be compared using the distance between frequency components of a COG signal.

Conclusion: The results and achievement of the thesis will be summarized in this chapter, we will show how the question asked in chapter one is answered by the achievements of this thesis. Some future works and new challenges will be proposed for further research.

2

Oculo-Vestibular dynamics and VIMS modeling

Mathematical models for three dimensional human dynamics as a part of the brain functions have been a hot research topic during the past four decades. Neuroscientists, cognitive scientists, mathematicians, robotics and computer vision specialists, electronics and control engineers were largely involved in spatial orientation modeling by developing, applying and modifying different algorithms and solutions to make the models operate similar to the human dynamically. Consequently, the models have found a wide variety of applications not only in scientific laboratories, but also in engineering fields such as electronics engineering, aircraft avionic system design, accident investigation, car and flight simulator, robotics and vision, motion system design, and in better understanding of disorientation among astronauts.

Researchers have followed two different approaches to explain the human Central Nervous System (CNS) mathematically: 1) dynamic modeling using classic control rules and stochastic signal processing techniques, 2) artificial intelligence relying on soft computing methods. The proposed models are based on biological and neuroscience studies and researches have been done since 1960. However, some aspects of these researches have been unknown to the best of our knowledge and further investigations and surveillance are required. For instance, the real biological mechanism and cognitive process behind nausea and

vomiting and the way this mechanism is connected to vagal nuclei are not yet clear. Usually, a model is made to facilitate the explanation of a physical phenomenon or a cognitive process via mathematical descriptions. The outcome of the process and phenomenon will be called observation throughout this thesis. The final objective of this chapter is to review these two approaches and then merge the second approach into the first one to make a stronger model in order to explain more observations. Besides, we believe that our fundamental research will provide a better solution for the current engineering problems and extend the border of technology. The theory of Bayesian[173] (with emphasis on Oman[144] / Kalman[190] model) and neural network as two bases of state estimation and perception will be introduced. Then, neural network, as an example of naturally inspired solution for prediction, will be embedded in the Bayesian model to construct an approximate model for explaining a cognitive process more quantitatively.

Another important part of this chapter is about the human vision modeling. Different components of an eye movement will be characterized and a mathematical model will be derived. Then these components are merged to construct a complete model for monocular vision and finally for binocular vision. Due to the complexity of this model a simplified version will be introduced to explain visually induced motion sickness caused by binocular disparity. At the end of this chapter, the biological mechanism behind visually induced motion sickness will be explained and Oman's model as a complete model of descripting this process will be presented.

2.1 Bayesian model and stochastic process

Many problems in engineering and science need an estimation of a state of a system that changes over time using a sequence of noisy measurements recorded on the output of the system. We estimate the state because it is hidden and we do not have access directly. For instance, each cognitive parameter can be considered as a state in the human brain. The parameter can be estimated by psychophysiological measurements and symptoms. The Bayesian theory provides a very generic conceptual solution based on noisy measurements for the state estimation problem. To make ease of further reference, the principle of Bayesian model[173] will be briefly reviewed in this section. To define the problem, consider the evolution of the state sequence $\{x_k, k \in \mathbb{N}\}$ of a target system given by

$$x_k = f_k\left(x_{k-1}, v_{k-1}\right) \tag{2.1}$$

where, $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{n_z}$ in (2.1) denotes a possibly nonlinear function of the state x_{k-1} and v_{k-1}, process noise distribution. $\{v_{k-1}, k \in \mathbb{N}\}$, n_x, n_v, n_z represent the process noise sequence, the dimensions of the state and process noise vectors, and measurement noise vectors, respectively. The

objective of tracking, for instance, is to recursively estimate x_k from the measurements

$$z_k = h_k\left(x_k, n_k\right) \tag{2.2}$$

where, $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \rightarrow \mathbb{R}^{n_z}$ in (2.2) similarly denotes another possibly nonlinear function, $\{n_k, k \in \mathbb{N}\}$ is a measurement noise sequence, and n_n demonstrates the dimension of the measurements. In particular, we are looking for filtered estimates of x_k based on the set of all available measurements $z_{1:k} = \{z_i, i = 1 \ldots k\}$ up to time k.

For a given event space with two members, A and B, the Bayesian theorem gives the relationship between the probabilities of A and B, $p(A)$ and $p(B)$, and the conditional probabilities of A given B and B given A, $p(A|B)$ and $p(B|A)$. Formula (2.3) denotes the most common form of the theorem.

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{2.3}$$

Often, for some partition $\{A_j\}$ of the event space, the event space is given or conceptualized in terms of $p\left(A_j\right)$ and $p\left(B|A_j\right)$. It is then useful to compute $p(B)$ using the law of total probability:

$$p(B) = \sum_j p\left(B|A_j\right) p\left(A_j\right) \tag{2.4}$$

Then conditional probability of A_j given B is calculated by (2.5).

$$p\left(A_j|B\right) = \frac{p\left(B|A_j\right) p\left(A_j\right)}{\sum_j p\left(B|A_j\right) p\left(A_j\right)} \tag{2.5}$$

From a Bayesian perspective[173], the tracking problem is simplified to recursively calculate some degrees of belief in the state x_k at time k, given measurement values up to time k, $z_{1:k}$.Thus, it is required to construct the probability distribution function (pdf) $p\left(x_k|z_{1:k}\right)$. It is assumed that the initial pdf $p\left(x_0|z_0\right) \equiv p\left(x_0\right)$ of the state vector, which is also known as a prior, is available. There is a set of no measurements at time $k = 0$, $z_0 = 0$ always.

Then, applying the Bayesian theorem, the pdf $p\left(x_k|z_{1:k}\right)$ may be obtained in two stages recursively: prediction and correction. Suppose the required pdf $p\left(x_{k-1}|z_{1:k-1}\right)$ at time $k-1$ is available. The prediction stage involves using the system model (2.1) to obtain a prior pdf of the state at time k via the Chapman–Kolmogorov equation (2.6).

$$p\left(x_k|z_{1:k-1}\right) = \int p\left(x_k|x_{k-1}\right) p\left(x_{k-1}|z_{1:k-1}\right) \mathrm{d}x_{k-1} \tag{2.6}$$

Note that in (2.6), the fact has been used that $p\left(x_k|x_{k-1}, z_{1:k-1}\right) = p\left(x_k|x_{k-1}\right)$, as (2.1) describes a

Markov process of order one. It means the distribution of a state at k knowing the state at $k-1$ and the measurements from the beginning to $k-1$ is equal to its distribution knowing only the state at $k-1$ because the distribution has been corrected already. The probabilistic model of the state evolution $p\left(x_k|x_{k-1}\right)$ is defined by the system equation (2.1) and the known statistics of v_{k-1}. At time k, a measurement z_k becomes available, and this may be used to correct the prior (correction stage) via the Bayesian theorem (2.7)

$$p\left(x_k|z_{1:k}\right) = \frac{p\left(z_k|x_k\right) p\left(x_k|z_{1:k-1}\right)}{p\left(z_k|z_{1:k-1}\right)} \tag{2.7}$$

where, the normalizing constant (2.8) is calculated using the law of total probability (2.4).

$$p\left(z_k|z_{1:k-1}\right) = \int p\left(z_k|x_k\right) p\left(x_k|z_{1:k-1}\right) \mathrm{d}x_k \tag{2.8}$$

It depends on $p\left(z_k|x_k\right)$, the likelihood, and n_k, the dimension of the measurement vector, defined by the measurement model (2.2). The measurement at time k, z_k, is used to modify a prior density to obtain the required posterior density of the current state in the correction stage (2.7).

The recurrence relations (2.6) and (2.7) form the basis for the optimal Bayesian solution. This recursive calculation of the posterior density is only a conceptual solution and cannot be determined analytically in general. Solutions only do exist in a restrictive set of cases (e.g., items 1-4). The Bayesian model provides a very generic solution for the state estimation problem. Assumptions and solutions for four most common cases of Bayesian model will be reviewed below[173]:

1. *Kalman Filter (KF)*: The Kalman filter assumes that the posterior density at every time step is Gaussian and, hence, parametrized by a mean and covariance. These assumption and optimized solution are only applied to linear systems. This case will be discussed in more detail later in section 2.2.

2. *Extended Kalman Filter (EKF)*: If (2.1) and (2.2) cannot be rewritten in the form of KF because the functions are nonlinear, then a local linearization of the equations may be a sufficient description of the nonlinearity.

3. *Approximate Grid-Based Methods (AGBM)*: If the state space is continuous but can be decomposed into N_s subspaces ("cells"), $\left\{x_k^i : i = 1, \ldots, N_s\right\}$, then (2.9) can be used to approximate the posterior pdf at time k.

$$p\left(x_{k-1}|z_{1:k-1}\right) \approx \sum_{i=1}^{N_s} w_{k-1|k-1}^i \delta\left(x_{k-1} - x_{k-1}^i\right) \tag{2.9}$$

   Then, (2.10) and (2.11) can be used recursively in prediction and correction stages.

$$p\left(x_k|z_{1:k-1}\right) \approx \sum_{i=1}^{N_s} w_{k|k-1}^{i} \delta\left(x_k - x_k^i\right) \tag{2.10}$$

$$p\left(x_k|z_{1:k}\right) \approx \sum_{i=1}^{N_s} w_{k|k}^{i} \delta\left(x_k - x_k^i\right) \tag{2.11}$$

4. *Particle filter (PF)*: Monte Carlo (MC) filtering based on the Sequential Importance Sampling (SIS) algorithm has constructed the basis for most sequential MC filters developed over the past decades. This sequential MC (SMC) approach is widely known as bootstrap filtering, particle filtering, interacting particle approximations and so on. It is an approach for implementing a recursive Bayesian filter by MC simulations. The very basic idea behind is to represent the required posterior density function by a set of random samples with associated weights and to estimate the target states by combining these samples and weights. As the number of samples becomes very large, this MC characterization becomes an equivalent representation to the usual functional description of the posterior pdf, and the SIS filter approaches the optimal Bayesian estimate.

As it will be shown in section 2.7, Oman[144] uses a Kalman filter scheme to explain VIMS. Therefore, the first case of Bayesian filters, KF, will be discussed in more detail in section 2.2 to provide a clear insight into the Kalman gain, a prior and posterior pdf calculation for both scalar and vector states and how Kalman Filter works in practice. Cases 2 through 4 will be skipped because we will not refer to them anymore. To this end, the biological background of Kalman filter will be reviewed in section 2.2.1.

2.2 State estimation theory using Kalman Filter

2.2.1 Biological basis of Kalman filter

The Kalman model was inspired by the brain functions in combining different sensory organs such as vestibular and ocular for state estimation. In the Kalman approach, dynamic models are typically defined for individual sense organs. Then the human CNS and brain cortex resolve the inherent ambiguities in incoming sensory information by applying various systems identification and estimation methods. Recently, there has been renewed interest in the observer theory and different types of stochastic filtering (Bayesian filtering, Kalman filtering, Unscented Kalman filtering, and Particle filtering) because both pose that the sensory input is compared at the level of a second order neuron with internal model derived predictions (Cullen[34]). The value of different types of Bayesian models is that they reproduce - and also successfully predict - a wide variety of 3D responses using a relatively small number of assumptions and parameters.

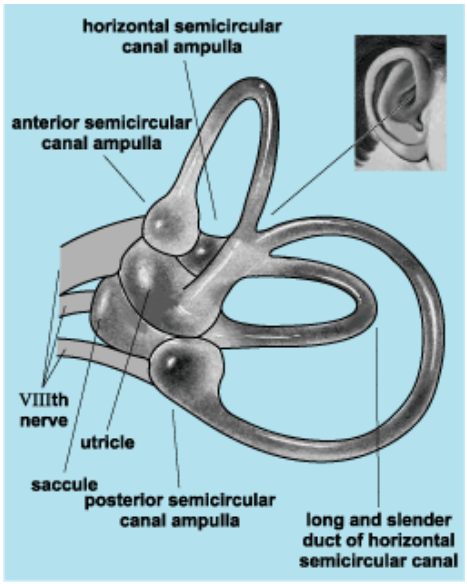


Figure 2.1: Semi-circular canal and otolith End-Organs (utricle and saccule) in the middle ear [182].

Two sensory organs are located in the inner ear: the vestibule and the cochlea. The vestibule is the organ of equilibrium and the cochlea the organ of hearing. The Semi-Circular Canal (SCC) and the otolith End-Organ Dynamics (EOD) are two organs of the vestibule as shown in Fig. 2.1. This arrangement is called the vestibular system in the scientific literature and explains part of the human orientation perception. Since the world is three-dimensional, the vestibular system contains three semi-circular canals in each labyrinth. They are approximately orthogonal (right angles) to each other. Human spatial perception not only relies on vestibular organs but the vision system (ocular) as well. The earliest mathematical models for human orientation perception [122] attributed the attenuation of sensations during prolonged rotation around the vertical axis of the Earth entirely to the SCC and EOD. However, by Fernandez and Goldberg's studies [44,51], it became clear that the time course of perception was not entirely determined by the SCC and EOD. Animal vestibulo-ocular reflex (VOR) data indicated that the mechanisms in the CNS extend the bandwidth of motion perception and increase the VOR time constant and were not primarily due to the otolith. Dynamically equivalent models of the Vestibule-Ocular system were proposed by Robinson [161] and Raphan et al. [156] to explain the perception mechanism. Mayne [122] proposed a 3D orientation model where the CNS estimated "down" and linear accelerations respectively via complementary low and high pass filtering of gravireceptor cues.

After these progress in system recognition and mathematical modeling, aerospace engineers further developed general methods for estimating the orientation and position of a vehicle, in real-time, based on data from a relatively small set of sensor measurements [71]. The vehicle trajectory is continuously estimated and

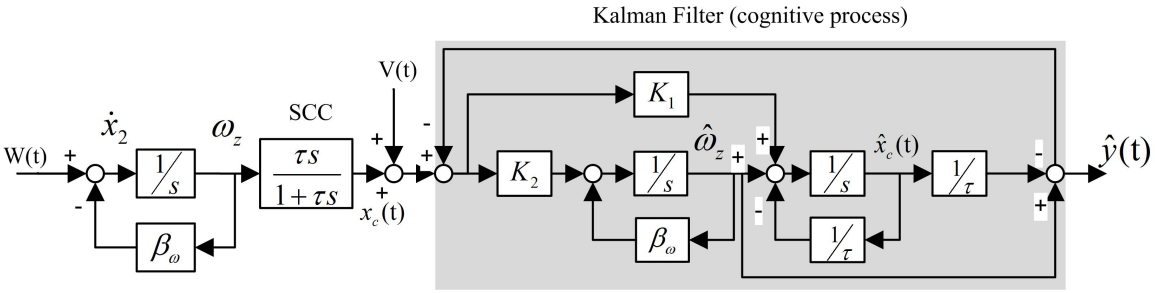


Figure 2.2: Borah's model for the vestibular system.

corrected by using internal models and the measurements. The difference between predicted and actual sensor measurements is then appropriately weighted and used to correct the estimated trajectory (Luenberger Observer[113]). For the most common case in engineering, where the entire system is linear, and where "noise" disturbances to both the vehicle and sensor measurements can be well characterized, Kalman[91] demonstrated how to calculate optimal residual weighting coefficients that minimize the stochastic error in the trajectory estimation, as seen in section 2.2.2. Such state estimator is referred to as a steady state Kalman Filter (KF). Kalman Filter is a special case of the Bayesian Filter which was reviewed briefly in section 2.1. Borah et al.[12,13] applied KF to a human riding passively in a vehicle (Fig. 2.2). Borah et al.'s orientation estimator incorporated dynamic models for the SCC and EOD, as well as visual angular and linear velocity cues. Other parameters such as the bandwidth and noise were regarded as noted by Mac Neilage et al.[116]. They applied their solution to the vehicle dynamics study.

Many engineering and cognitive paradigms involve an input variation (real or perceived) so large that the system cannot be regarded as a linear system. Therefore, a linear KF model is not appropriate for such an application. Merfeld et al.[128,127] therefore proposed an "Observer" model to solve this problem. Merfeld et al.'s model has been put into practice, validated and extended by Haslwanter et al.[58] and Vingerhoets et al.[214]. Although model inputs and outputs are usually physical quantities in the proposed models, most CNS mechanisms remain physiologically unknown. For instance, Borah et al.'s KF model explains how vestibulo-ocular cues are combined to describe the human orientation perception. However, Borah et al. does not incorporate any information from proprioception. Weighted residual corrections were applied in an ad-hoc fashion in the previous KF models, and linear system disturbances as well as sensor noise were considered Gaussian noise. Besides, measurement transfer function in the architecture of the estimator and process is assumed identical. This is not true when we are applying KF to the human orientation perception because the state estimation comes from the CNS which is not necessary similar to physical measurements.

It is widely accepted that simulator sickness, cyber sickness, and visually induced motion sickness occur partly due to the difference between real and estimated measurements (perceived measurement)[157]. This

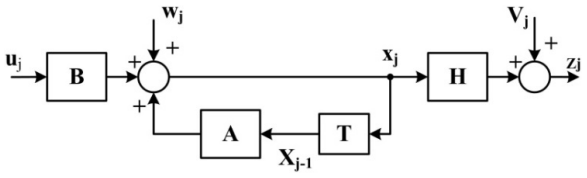


Figure 2.3: Linear system model in KF scheme.

difference is called residual vector or conflict vector in the Kalman theory of state estimation. As we will see later in section 2.7, Oman[144] will use this vector to explain motion sickness. Therefore we need to study the Kalman filter and understand how it works.

2.2.2 Kalman filter

In 1960, R. E. Kalman[91] published his famous paper describing a recursive solution to the discrete-data linear filtering problem. The Kalman filter is a set of equations that provides an efficient computational mean to estimate and correct the state of a process by a set of noisy measurements recursively, in a way that it minimizes the mean of the squared error. The Kalman filter assumes that the posterior density at every time step is Gaussian and, hence, parametrized by a mean and covariance. Equations (2.12) and (2.13) are two main equations that link the input and output of a linear system to the hidden target state, x_j, at time t.

A discrete time system with process noise w and measurement noise v is defined by:

$$x_j = Ax_{j-1} + Bu_j + w_j \tag{2.12}$$

$$z_j = Hx_j + v_j \tag{2.13}$$

The corresponding block diagram is shown in Fig. 2.3, where, distribution functions w and v indicate the process and measurement noise with Q and R covariance matrices accordingly. Variables x_j, u_j and z_j represent the hidden state, the input and output (measurement) at time t and matrices A, B, and H define the associated transfer function for a linear system and measurement devices. The block diagram including the estimation part is given by Fig. 2.4. Usually, the parameters of the estimator and the main system are the same in Fig. 2.4. But, in a stochastic process like the human brain, the estimator is being replaced by a cognitive process. The cognitive process models the cortex operation or the human perception.

Fig. 2.4 models the prediction and correction stages for the cognitive process given by (2.14) and (2.15), where, matrix K_j refers to as Kalman Gain in the literature.

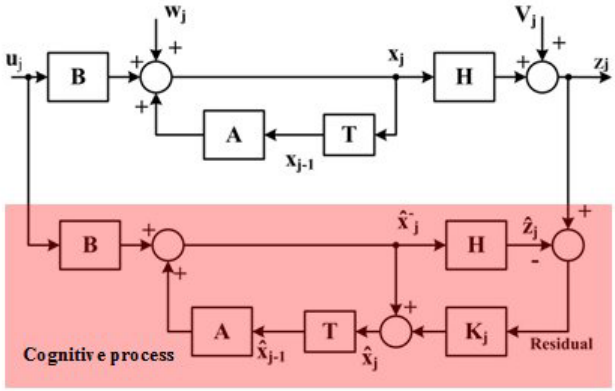


Figure 2.4: Kalman filter including the estimation part.

$$\hat{x}_j^- = A\hat{x}_{j-1} + Bu_j \tag{2.14}$$

$$\hat{x}_j = \hat{x}_j^- + K_j\left(z_j - H\hat{x}_j^-\right) \tag{2.15}$$

We want to use our knowledge of the measured value z to generate and estimate $\hat{x}$ of the unknown state x, in general. Thus we want to find the value of x that minimizes the weighted sum of squares of the vector $[z - H\hat{x}]$. If we let matrix W be a general weighting matrix, then the aim is to find the vector $\hat{x}$ that minimizes the scalar cost J:

$$J = \frac{1}{2}\left[z - H\hat{x}\right]W\left[z - H\hat{x}\right]^T \tag{2.16}$$

Where the superscript T denotes the matrix transpose. Note that if $W = I$, this is standard least squares, with

$$J = \frac{1}{2}\left[z - H\hat{x}\right]\left[z - H\hat{x}\right]^T = \frac{1}{2}\sum_{i=1}^{m}\left[z_i - H\hat{x}_i\right]^2 \tag{2.17}$$

If W is a diagonal matrix with diagonal terms $w_1, w_2, \ldots, w_m$, then

$$J = \frac{1}{2}\sum_{i=1}^{m} w_i\left[z_i - H\hat{x}_i\right]^2 \tag{2.18}$$

This minimization is accomplished by the values $\hat{x}_{wls}$, where the subscript denotes weighted least

squares, if

$$\left.\frac{\partial J}{\partial \hat{x}}\right|_{\hat{x}=\hat{x}_{wls}} \triangleq \left.\left[\frac{\partial J}{\partial \hat{x}_1}\frac{\partial J}{\partial \hat{x}_2}\cdots\frac{\partial J}{\partial \hat{x}_n}\right]\right|_{\hat{x}=\hat{x}_{wls}} = 0 \tag{2.19}$$

and

$$\left.\frac{\partial^2 J}{\partial \hat{x}^2}\right|_{\hat{x}=\hat{x}_{wls}} \geq 0 \tag{2.20}$$

However, often, we apply equation (2.23) to KF gain calculations. The priori (P_j^-) and posteriori (P_j) covariance matrices are given by (2.21) and (2.22):

$$P_j^- = E\left\{e_j^-\left(e_j^-\right)^T\right\} = E\left\{\left(x_j-\hat{x}_j^-\right)\left(x_j-\hat{x}_j^-\right)^T\right\} \tag{2.21}$$

$$P_j = E\left\{e_j e_j^T\right\} = E\left\{\left(x_j-\hat{x}_j\right)\left(x_j-\hat{x}_j\right)^T\right\} \tag{2.22}$$

To find the best value for the filter gain, K_j, we differentiate the posteriori covariance and set it to zero:

$$\frac{\partial P_j}{\partial K_j} = \frac{\partial E\left\{\left(x_j-\hat{x}_j\right)\left(x_j-\hat{x}_j\right)^T\right\}}{\partial K_j} = 0 \tag{2.23}$$

In fact, our brain tries to solve this equation while the cognitive process is estimating the body state. In terms of mathematical modeling, the following equations propose a good solution, however the underlying cognitive process of the brain in reality is not very well understood and still the mental mechanism is unknown. Besides most of the problems we face in reality, and in virtual reality as a subspace of a global environment, is rooted in (2.22). Moreover, sensory conflicts[157] somehow are related to (2.23) which in turn is hiddenly linked with (2.23). If the engineering design in aviation, flight and car simulator, or virtual reality does not work and creates lots of cognition problems for the end-user, it means the effect of (2.23) on the other part of the design has not been considered appropriately.

The Kalman filter gain, the recursive form of the priori, and posteriori covariance are obtained after much algebra and are given by the following equations (2.24) through (2.26):

$$K_j = P_j H^T\left(HP_jH^T+R\right)^{-1} \tag{2.24}$$

$$P_j^- = AP_{j-1}A^T+Q \tag{2.25}$$

$$P_j = \left(I-K_jH\right)P_j^- \tag{2.26}$$

2.3 Neural network and cognitive process modeling

We have mentioned two simulation approaches for the cognitive process or Central Nervous System (CNS) modeling at the beginning of the chapter, i.e., stochastic and artificial intelligence. In the second approach of the human CNS modeling, computational neuroscientists[126] proposed different models such as Artificial Neural Networks (ANN)[74], Fuzzy Logic[151], and Neuro-Fuzzy[22] to simulate the human brain and create a similar system for engineering applications. ANNs are trainable models inspired by animal central nervous systems (in particular the human brain) and capable of performing machine learning and pattern recognition algorithms in parallel. Two examples of ANN are shown in Fig. 2.6 and Fig. 2.7. Due to the similarity of ANN to the human CNS, it attracts significant attention in cognitive modeling and stochastic signal processing.

McCulloch and Pitts[126] (1943) created a mathematical model for neural networks referred to as "threshold logic" in the literature. Rosenblatt[165] (1958) created the perceptron ANN and a learning algorithm based on a two-layer network. Some mathematical computation (e.g., exclusive-or) in Rosenblatt's model could not be processed until after the back-propagation algorithm proposed by Werbos[224]. Unfortunately, the back-propagation of error algorithm is not very biologically plausible. Signals have never been seen to flow backward across synapses in an actual neural tissue contrary to the manner necessary for the back-propagation algorithm to be implemented. O'Reilly expanded on McClelland et al.[124] to implement a biologically plausible version of back-propagation of error. This is called the generalized recirculation algorithm (GRA)[146]. McClelland et al.[124] interpreted the back-propagation error signal as the difference between the expected outcome and the perceived outcome. Under this interpretation, these algorithms are quite general in practical implementations.

2.3.1 Neural network architecture

The basic computational unit in the Central Nervous System (CNS) in general and the brain especially is the nerve cell, or neuron. A neuron has been made up of three basic elements: Dendrites (inputs), Cell body, Axon (output), as shown in Fig. 2.5. The mathematical model of a single neuron or a network of neurons has the same architecture. In this architecture, any ANN has an input layer, an output layer and a network body. The diversity and complexity of different ANNs is related to the network body and its application. For example, a multi-layer perceptron (MLP) has a fairly simple architecture. Its body consists of different hidden layers with different number of neurons and transfer functions. Now, let's look at another network with a fairly complex architecture which is inspired by brain functions. The cerebral cortex is arguably the most fascinating structure in all of human physiology. Although vastly complex on a microscopic level, the cortex reveals a consistently uniform structure on a macroscopic scale, from one brain to another. Centers for such diverse activities as thought, speech, vision, hearing, and motor

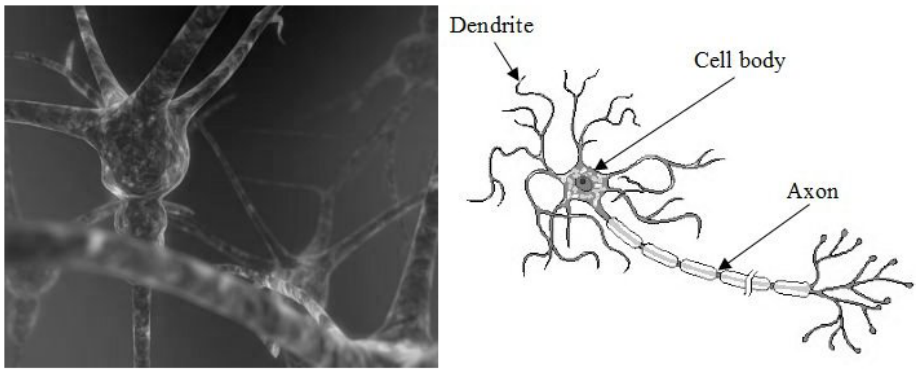


Figure 2.5: Nervous cell and its basic elements.

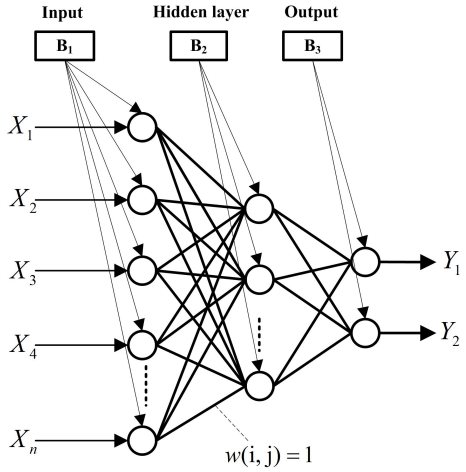


Figure 2.6: Example of Artificial Neural Network (ANN).

functions lie in specific areas of the cortex, and these areas are located consistently relative to one another. Moreover, individual areas exhibit a logical ordering of their functionality. Auditory and vision regions and motor nerves can be referred to as ordered feature maps. Kohonen[104] proposed an ANN architecture based on this biological knowledge called self-organizing map (SOM). A SOM or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map. As seen, a Kohonen neural network[104] has a very complicated architecture in comparison with MLP. Its architecture consists of different layers connected in a 2D configuration.

Fig. 2.6 and Fig. 2.7 illustrate two examples of neural networks for two different purposes. Sometimes, more than one type of ANN is used in a configuration to fulfill the requirement of an application. Two computational neurobiologists, McCulloch and Pitts[126], have constructed a very elaborate numerical

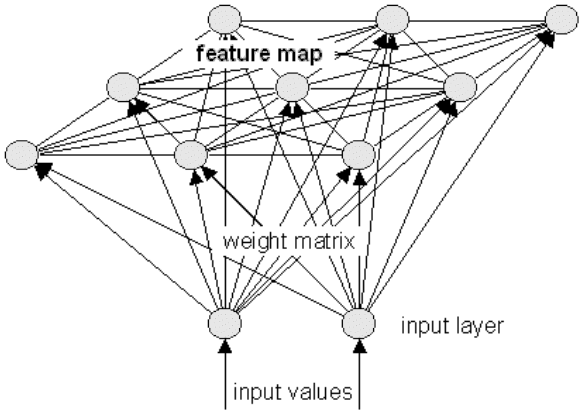


Figure 2.7: Kohonen Neural Network.

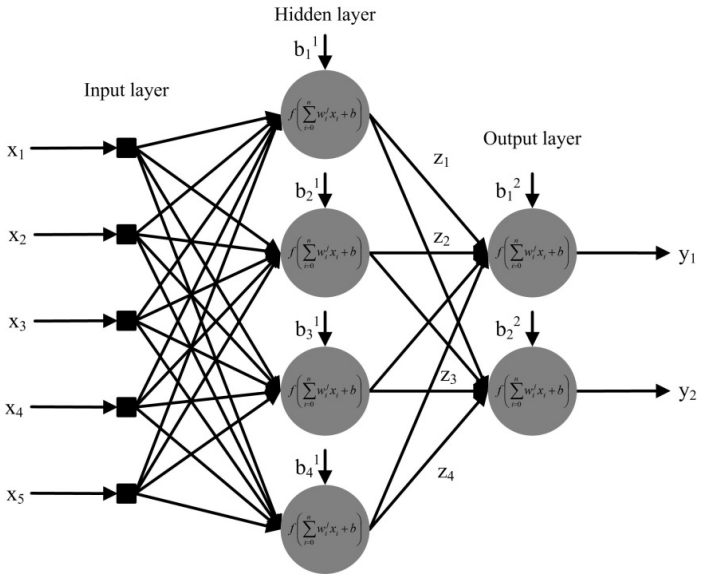


Figure 2.8: Typical neural network with input/output and hidden layers.

model of neurons in order to run detailed simulations of particular circuits in the human brain (Fig. 2.8). Then, they made an artificial neural network (ANN) based on a set of neurons to explain a mental process. They proposed a two-layer perceptron neural network to validate their idea however, their model was very limited. For example, they could not implement the logic gate XOR. Finally, it turned out that the McCulloch and Pitts's model could not learn anything requiring more than two layers. Chauvin and Rumelhart [23] found a generic solution for this problem with two insights. First, they implemented a non-linear sigmoid function (neuronal threshold). Second, they developed an algorithm called "back-propagation", which allows the output layer to propagate its error back (by updating the bias and weighting values in each iteration) across all the layers such that the error can be corrected in a distributed fashion. In section 2.3.2 we will discuss this training algorithm in more detail. Chauvin and Rumelhart used this new learning algorithm to explore how cognition might be implemented in a parallel and distributed fashion in neuron-like units.

Mathematical equations corresponding with an MLP ANN will be discussed below to show how we can build a network in a simulation environment such as MATLAB Simulink. The same formulation can be repeated for any type of ANN to draw a set of equations which connects the output to the input variables. As shown in Fig. 2.8, a typical ANN is built up of layers of neurons (minimum three layers: input, hidden and output). Each neuron can either accept a vector or a scalar input ($\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$) and gives a scalar or a vector output ($\mathbf{z} = [z_1, z_2, z_3, z_4]$). The inputs are weighted by $\mathbf{w} = \left[w_i^j\right]_{m\times n}$ (index i and j referred to as the selected input and layer index respectively) and given a bias vector ($\mathbf{b} = \left[b_1^1, b_2^1, b_3^1, b_4^1\right]$). For instance, w_2^1 represents the weighting for input 2 in the input layer). The first output of the hidden layer, z_1, can be expressed by (2.27):

$$z_1 = f\left(\sum_{i=0}^{5} w_i^1 x_i + b_i^1\right) \tag{2.27}$$

The output vector of the hidden layer then will be (2.28)

$$\mathbf{z} = f\left(\mathbf{w_h x} + \mathbf{b_1}\right) \tag{2.28}$$

We originally configured our neural network to use neurons with a linear (purelin, $f(x) = x$) or non-linear (logsig, $f(x) = \frac{1}{1+e^{-x}}$), tansig, $f(x) = \frac{1-e^{-x}}{1+e^{-x}}$) transfer function in (2.28). Index h in (2.28) demonstrates the weighting matrix of the hidden layer. The non-linear neuron transfer function (2.29) operates on this value to generate the final scalar output or output vector in the last layer ($\mathbf{y} = [y_1, y_2]$).

$$\mathbf{y} = f\left(\mathbf{w_0 z} + \mathbf{b_2}\right) \tag{2.29}$$

The final transformation between the input and output vectors will be achieved by substituting $\mathbf{z}$ from

(2.28) into (2.29), as shown in (2.30)

$$\mathbf{y} = f\left(\mathbf{w_0} f\left(\mathbf{w_h}\mathbf{x} + \mathbf{b_1}\right) + \mathbf{b_2}\right)$$

or

$$\mathbf{y} = \mathbf{M}(\mathbf{w}, \mathbf{b})\mathbf{x} \tag{2.30}$$

A unique weighting and bias matrix can be calculated by a training algorithm, such as back-propagation [117] and a set of training data. It means we can define a transformation between each sensory organ such as visual and auditorial, and require sensory features in the cognition process.

2.3.2 Training algorithm

Back propagation

Once the network weights and biases are initialized, the network is ready for training. The multilayer feedforward network can be trained for function approximation (nonlinear regression) or pattern recognition. The training process requires a set of examples of proper network behaviors - network inputs p and target outputs t. The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance, as defined by the network performance function net. The default performance function for feedforward networks is the mean square error (MSE) - the average squared error between the networks outputs a and the target outputs t. It is defined by (2.31) as follows:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \tag{2.31}$$

(Individual squared errors can also be weighted) There are two different ways in which training can be implemented: the incremental mode and the batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode, all the inputs in the training set are applied to the network before the weights are updated. This chapter describes the batch mode training with the train command. Batch training is significantly faster and produces smaller errors than incremental training.

For training multilayer feedforward networks, any standard numerical optimization algorithm can be used to optimize the performance function, but there are a few key ones that have shown excellent performance for neural network training. These optimization methods use either the gradient of the network performance with respect to the network weights, or the Jacobian of the network errors with respect to

the weights. The gradient and the Jacobian are calculated using a technique called the back propagation algorithm, which involves performing computations backward through the network[117].

Deep Learning

What is deep learning? An intelligent system such as ANN needs to learn as a human being learns. Although up to day back propagation is widely used for training purpose, a new strong training algorithm which is called Deep Leering has been shown to work effectively in practice as well. Deep learning is a set of algorithms in machine learning that attempt to model high-level abstractions by using architectures composed of multiple non-linear transformations[6]. Deep learning architectures, specifically those built from artificial neural networks (ANN), date back at least to 1980[46]. ANNs themselves date back even further. Yann LeCun et al.[105] were able to apply the standard back propagation algorithm to a deep neural network with the purpose of recognizing handwritten zip codes on mail. Despite the success of applying the algorithm, the time to train the network on this dataset was approximately 3 days, making it impractical for general use[105]. Many factors contribute to the slow speed, one being due to the so-called vanishing gradient problem (see Hochreiter et al.[68,69]). In combination with speed issues, ANNs fell out of favour in practical machine learning and simpler models such as support vector machines (SVMs) became the popular choice of the field in the 1990s and 2000s.

The term "deep learning" gained attraction in the mid-2000s after a publication by Geoffrey Hinton showed how a many-layered feedforward neural network could be effectively pre-trained one layer at a time, treating each layer in turn as an unsupervised restricted Boltzmann machine, then using supervised back propagation for fine-tuning[66]. In 1992, Schmidhuber had already implemented a very similar idea for the more general case of unsupervised deep hierarchies of recurrent neural networks, and has also experimentally shown its benefits for speeding up supervised learning[175,176].

Depth The computations involved in producing an output from an input can be represented by a flow graph: a flow graph is a graph representing a computation, in which each node represents an elementary computation and a value (the result of the computation, applied to the values at the children of that node). Consider the set of computations allowed in each node and the possible graph structures and this defines a family of functions. Input nodes have no children. Output nodes have no parents. The flow graph for the expression $\tan\left(a^2 + ab\right)$ could be represented by a graph with two input nodes a and b, one node for the division ab taking a and b as input (i.e., as children), one node for the square (taking only a as input), one node for the addition (whose value would be $\left(a^2 + ab\right)$ and taking as input the nodes a^2 and ab, and finally one output node computing the tangent, and with a single input coming from the addition node.

A particular property of such flow graphs is depth: the length of the longest path from an input to an

output. Traditional feedforward neural networks can be considered to have a depth equal to the number of layers (i.e., the number of hidden layers plus 2, for the input and output layers). Support Vector Machines (SVMs) have depth 2 (one for the kernel outputs or for the feature space, and one for the linear combination producing the output).

Motivations for deep architectures The main motivations for studying learning algorithms for deep architectures are the following:

- The brain has a deep architecture;
- Cognitive processes seem deep;
- Insufficient depth can hurt.

Since the resurgence of deep learning, it has shown to be part of many state-of-the-art systems in different disciplines, particularly that of computer vision and automatic speech recognition (ASR). Results on commonly used evaluation sets such as TIMIT (ASR) and MNIST (image classification) are constantly being improved with new applications of deep learning. Currently, it has been shown that deep learning architectures in the form of convolution neural networks have been the best performing; however, these are more widely used in computer vision than in ASR.

Advances in hardware have also been an important enabling factor for the renewed interest of deep learning. In particular, powerful graphics processing units (GPUs) are highly suited for the kind of number crunching, matrix/vector maths involved in machine learning. GPUs have been shown to speed up training algorithms by orders of magnitude, bringing running times of weeks back to days and days to hours[32,155].

The brain has a deep architecture The brain has a deep architecture for the reason using deep learning is always better than back propagation. For example, the visual cortex is well-studied and shows a sequence of areas each of which containing a representation of the input, and signals flow from one to the next (there are also skip connections and at some level parallel paths, so the picture is more complex). Each level of this feature hierarchy represents the input at a different level of abstraction, with more abstract features further up in the hierarchy, defined in terms of the lower-level ones.

Note that representations in the brain are in between dense distributed and purely local; they are sparse: about 1% of neurons are active simultaneously in the brain. Given the huge number of neurons, this is still a very efficient (exponentially efficient) representation.

Cognitive processes seem deep

- Humans organize their ideas and concepts hierarchically;
- Humans first learn simpler concepts and then compose them to represent more abstract ones;
- Engineers break-up solutions into multiple levels of abstraction and processing.

It would be nice to learn/discover these concepts (knowledge engineering failed because of poor introspection). Introspection of linguistically expressible concepts also suggests a sparse representation: only a small fraction of all possible words/concepts are applicable to a particular input (say a visual scene).

Insufficient depth can hurt Depth 2 is enough in many cases (e.g., logical gates, formal neurons, sigmoid-neurons, Radial Basis Function (RBF) units like in SVMs) to represent any function with a given target accuracy. But this may come with a price that the required number of nodes in the graph (i.e., computations, and also the number of parameters, when we try to learn the function) may grow very large. Theoretical results showed that there exist families of functions for which in fact the required number of nodes may grow exponentially with the input size. This has been shown for logical gates, formal neurons, and RBF units. In the latter case, Hastad[59] has shown families of functions which can be efficiently (compactly) represented with $\mathcal{O}(n)$ nodes (for n inputs) when depth is d, but for which an exponential number ($\mathcal{O}\left(2^n\right)$) of nodes is needed if depth is restricted to $d-1$.

One can see a deep architecture as a kind of factorization. Most randomly chosen functions cannot be represented efficiently, whether with a deep or a shallow architecture. But many that can be represented efficiently with a deep architecture cannot be represented efficiently with a shallow one (see the polynomials example in the Bengio survey paper[6]). The existence of a compact and deep representation indicates that some kind of structure exists in the underlying function to be represented. If there was no structure whatsoever, it would not be possible to generalize well.

Deep neural networks training A deep neural network (DNN) is defined to be an artificial neural network with at least one hidden layer of units between the input and output layers; it is also a universal approximator[193]. Similar to shallow ANNs, it can model complex non-linear relationships. The extra layers give it added levels of abstraction, thus increasing its modeling capability. DNNs are typically designed as feedforward networks, but recent research has successfully applied the deep learning architecture to recurrent neural networks for applications such as language modeling[130]. Convolutional deep neural networks (CNNs) are used in computer vision where their success is well-documented[106]. More recently, CNNs have been applied to acoustic modeling for automatic speech recognition (ASR), where they have shown success over previous models[171]. For simplicity, a look at training DNNs is given here.

A DNN can be discriminatively trained with the standard back propagation algorithm. The weight updates can be done via a stochastic gradient descent using the following equation:

$$\Delta\omega_{ij}(t+1) = \Delta\omega_{ij}(t) + \eta\frac{\partial C}{\partial\omega_{ij}} \tag{2.32}$$

Here, η is the learning rate, and C is the cost function. The choice of the cost function depends on factors such as the learning type (supervised, unsupervised, reinforcement, etc.) and the activation function. For example, when performing supervised learning on a multiclass classification problem, common choices for the activation function and cost function are the softmax function and cross entropy function, respectively. The softmax function is defined as

$$p_j = \frac{e^{x_j}}{\sum_k e^{x_k}} \tag{2.33}$$

where, p_j represents the class probability and x_j and x_k represent the total input to units j and k respectively. Cross-entropy is defined as

$$\mathrm{C} = -\sum_j d_j \log\left(p_j\right) \tag{2.34}$$

where, d_j represents the target probability for output unit j and p_j is the probability output for j after applying the activation function[65].

2.4 Oculomotor control system: theory and modeling

Oculomotor nerve originates from motor neurons in the middle brain and is responsible for eyeball and eyelid movement. Numerous mathematical models have already been proposed to explain eye movement. However, none of them presented a complete model. Robinson et al.[162,163] discussed the mechanism involved in only saccadic and pursuit eye movements using the control system theory. Saeb et al.[170] found an analytical solution to explain various aspects of saccadic movement relying on a mathematical formulation and solving partial differential equations governing eye movement. The analytical results of this study and the results from psychological studies are identical. Mergenthaler and Engbert[129] proposed a model for eye fixation and retinal slip. Zhang and Wakamatsu[238] classified eye movements into five categories: saccade, smooth pursuit, optokinetic reflex, vestibulo-ocular reflex (VOR), and vergence. Each category of eye movement is tried to be explained in more detail in the following subsections. This introduction is necessary here because these models will be referred to during the following chapters.

Since saccadic movement, smooth pursuit and retinal slip have great deal of importance in understanding the other types of eye movement, more time will be spent on these three categories to make them

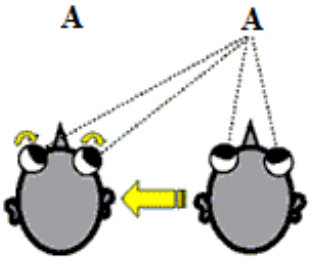


Figure 2.9: Saccadic eye movement.

completely clear and understandable in terms of theoretical background.

2.4.1 Saccade

Saccade is the ability of the eye(s) to move quickly from one point of interest to the next after an appropriate time of fixation (100 to 300 ms[170]) as shown in Fig. 2.9. Saccadic movement is characterized by the duration, the peak velocity and the amplitude. The relationship between the duration, the peak velocity and the amplitude of saccadic eye movements is known as the "main sequence". Mathematically, the saccade control system is modeled by an optimal control. The optimized control signals achieved by this method are compatible with "neuronal firing patterns" observed in the "Medial Superior Temporal" (MST) area of the cortex and the oculomotor "Neural Integrator" (NI) neurons of the brainstem[203,199].

In 1968, Young et al.[232] developed a sampled-data and non-predictive model (Fig. 2.10) of the eye movement system. Measurement devices, electronics and computer interfacing were not very well developed and so little was known about the brain stem organization at the time. For that reason, the elements of that model were not intended to resemble to the actual brain structure as it was rather influenced by the theory of control systems. Besides, the model was not able to predict the correct experimental responses when the target moved in a ramp, a step-ramp and in two steps in rapid succession.

In Fig. 2.10, angles θ_t and θ represent the target and the eye position respectively if we consider only horizontal movements. The eye plant (eye muscles and eyeball) is explained by a second order linear system where $\omega_n = 240$ rad/s and $\xi = 0.7$. When the error between the target and the eye position, $e = \theta_t - \theta$ exceeds the threshold e_t, the pulse generator (PG) is triggered which causes a sample to be taken. At the same time, INHBT Dead Zone (DZ) blocks for 0.2 s. INHBT is a simple timer in the circuit to create a delay for a certain period and if the signal meets the conditions then after, that period will be directed to the next element. Consequently, when $|e| \geq e_t$, samples will occur every 0.2 s, and when $|e| < e_t$, the samples will stop.

Robinson[162] modified the model by integrating the cognitive components, Neural Integrator (NI) and Medial Longitudinal Fasciculus (MLF), in the model (Fig. 2.11). These components were discovered by

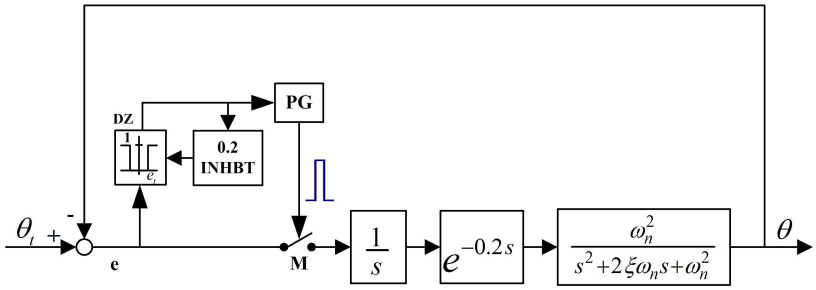


Figure 2.10: Model of the saccade mechanism proposed by Young et al.[232].

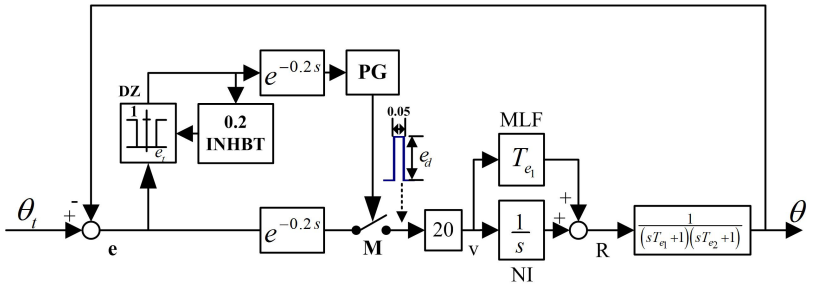


Figure 2.11: Model of the saccade mechanism proposed by Robinson[162].

neuroscientists when they were studying the vision of monkeys[7]. Till now, these components have been constant elements of any saccade mechanism. The second change has been made in the transfer function of the eye. As seen in Fig. 2.10, the denominator of the transfer function was replaced by two poles with time constant T_{e_1} and T_{e_2}. Zuber[240] stimulated cat's eyes in 1968 and discovered this relation between oculomotor nerves and eye movement.

Much effort has been made by previous researchers in order to identify the optimality principles which underlay the kinematic characteristics of saccades. The final objective of all these efforts was to propose a functionally optimal and biologically plausible cost function. To this end, several studies have suggested different cost functions. Enderle and Wolfe[41] proposed a cost function to minimize the time to reach the target (saccade trajectories optimization). Their assumption leads to a bang-bang control solution[189], however the resulting velocity profile was not biologically plausible[53]. In control theory, a bang-bang controller known as an on-off controller is a feedback controller that switches abruptly between two states. Since Enderle and Wolfe's model and its cost function were biological implausible, his successors relied more on Harris solution[54,55].

Harris and Wolpert[54] simplified the problem to find the optimal trajectory that minimizes the total cost associated with moving the eye to a target imaged on the retina at a specified position as foveal eccentricity. They approximated the total cost of an eye movement integrated over the movement as being composed of two components: 1) the cost associated with retinal slip (movement cost), 2) the fixation cost (after the

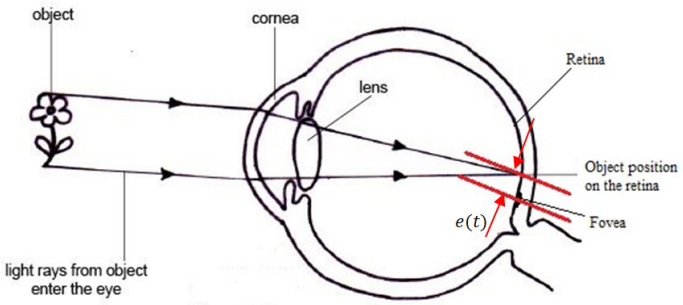


Figure 2.12: Foveal eccentricity definition.

movement ends). Equation (2.35) shows this cost function.

$$\text{cost} = \underbrace{\int_0^T \alpha \mathrm{d}t}_{\text{Movement cost}} + \underbrace{\int_T^{T+F} \beta e(t)^2 \mathrm{d}t}_{\text{Fixation cost}} \tag{2.35}$$

where, $e(t)$ is the foveal eccentricity of the target at time t (the difference between the fovea center and the current position of the object image on the retina, as shown in Fig. 2.12, and β is a constant value. Once the saccade has begun and vision is lost, the cost is a constant α (per unit time) until the saccade ends. T and F represent movement and post-movement periods.

To explain briefly Harris and Wolpert's solution, we denote the eye position at time t for a movement made of desired amplitude P, as $x_{P(t)}$. We assume that the aggregate neural command, $u(t)$, is perturbed by a zero-mean additive white noise process with instantaneous standard deviation (2.36) proportional to the mean.

$$\sigma_u(t) = k\,|u(t)| \tag{2.36}$$

We can rewrite $x_p(t)$ as (2.37) by the linearity of the system, where, g ($g = \frac{P}{A}$, A is the movement resolution: the distance between two consecutive steps and P is the amplitude of the saccade) is a random variable with mean, $\bar{g}$ and variance, σ_g^2.

$$x_p(t) = g x_A(t) \tag{2.37}$$

The cost function can be defined by (2.38),

$$J = E(\text{cost}) = \alpha T + \beta' \int_T^{T+F} \sigma_x^2(t)\mathrm{d}t + \gamma \tag{2.38}$$

where, $\sigma_x^2(t)$ is the variance of the eye position in the fixation period,

$$\beta' = \left(\bar{g}^2 + \sigma_g^2\right)\beta \tag{2.39}$$

and

$$\gamma = F\beta\left[(A - A\bar{g})^2 + \sigma_g^2 A^2\right] \tag{2.40}$$

We denote the eye position as $x(t)$ which is the output of a linear pole-only ocular motor plant, with an impulse response $p(t)$, whose input is a single scalar aggregate mean neural command $u(t)$. By definition, the mean and variance of a random variable $x(t)$ is defined by (2.41) and (2.42).

$$E\left[x(t)\right] = \int_0^t ku(\tau)p(t-\tau)\mathrm{d}\tau \tag{2.41}$$

$$\sigma_x^2(t) = \mathrm{Var}\left(x(t)\right) = \int_0^t k^2u^2(\tau)p^2(t-\tau)\mathrm{d}\tau \tag{2.42}$$

For more detail on (2.42), see[55]. Based on this assumption, the variance of the eye response increases when one tries to decrease the saccadic duration by recruiting larger command signals. As a result, a trade-off needs to be made between the speed and the accuracy of saccades. The optimal solution to this trade-off is a trajectory that is biologically plausible. If (2.42) is substituted in (2.38), the final cost function will be achieved by (2.43).

$$J = \alpha T + \beta'\int_T^{T+F}\sigma_x^2(t)\mathrm{d}t + \gamma = \gamma + \alpha T + \beta'\int_T^{T+F}\left[\int_0^t k^2u^2(\tau)p^2(t-\tau)\mathrm{d}\tau\right]\mathrm{d}t \tag{2.43}$$

We can split this function into movement ($0 \leq t \leq T$) and fixation ($T \leq t < T+F$) as (2.44):

$$J = \gamma + \alpha T + \beta' k^2\int_T^{T+F}\left[\int_0^T u^2(\tau)p^2(t-\tau)\mathrm{d}\tau + \int_T^t u^2(\tau)p^2(t-\tau)\mathrm{d}\tau\right]\mathrm{d}t \tag{2.44}$$

and finally, (2.44) can be expressed by (2.45)

$$J = \gamma + \alpha T + \beta' k^2\int_T^{T+F}u^2(\tau)q_T(\tau)\mathrm{d}t \tag{2.45}$$

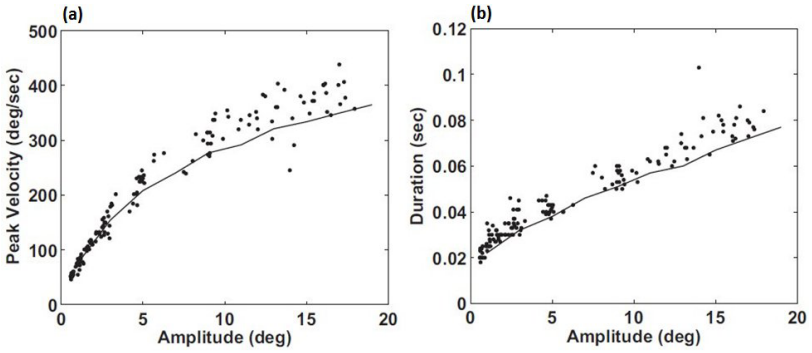


Figure 2.13: Characteristic of the main sequence adapted from [5]: solid line demonstrates data from simulation and analytical approaches, and scatter points shows the data from experiments.

where

$$q_T(\tau) = \int_T^{T+F} p^2(t-\tau)\mathrm{d}\tau \tag{2.46}$$

Then, by assuming the eye plant as (2.47), which is an order n state space model, the optimization result will be as shown in (2.48).

$$\dot{x} = Ax(t) + Bu(t) \tag{2.47}$$

From Pontryagin's minimum principle the optimal control signal is given by

$$u(t) = \sum_{r=1}^{n} C_r \frac{e^{\gamma_r t}}{q_T(t)} \; (0 \leq t \leq T) \tag{2.48}$$

where, n and γ_r are the order of the system and a damping constant respectively. Coefficient C_r is determined by the boundary and initial conditions. Later Harris and Wolpert's model was extended by introducing an internal feedback which consists of two state estimators [27]. One predicts the state of the eye, and the other estimates the state of the target. This feedback is used to generate the neural control signal when the input is the target position. The feed-forward controller is optimized in a similar way as explained above. It requires re-optimization for every saccade. Kardamakis and Moschovakis [92] proposed another cost function based on the squared sum of the eye torque signals integrated over the movement velocity profile. This approach is compatible with the "main sequence" characteristics. The main sequence is stereotyped: The duration increases linearly with the saccadic amplitude, and the peak velocity increases linearly for low amplitudes and then undergoes a soft saturation for larger amplitudes [5] as shown in Fig. 2.13.

Although the approaches proposed by Harris and Wolpert [54] have been successful in explaining the kinematics of saccades in terms of satisfying some optimality criteria and the result was biologically plausible,

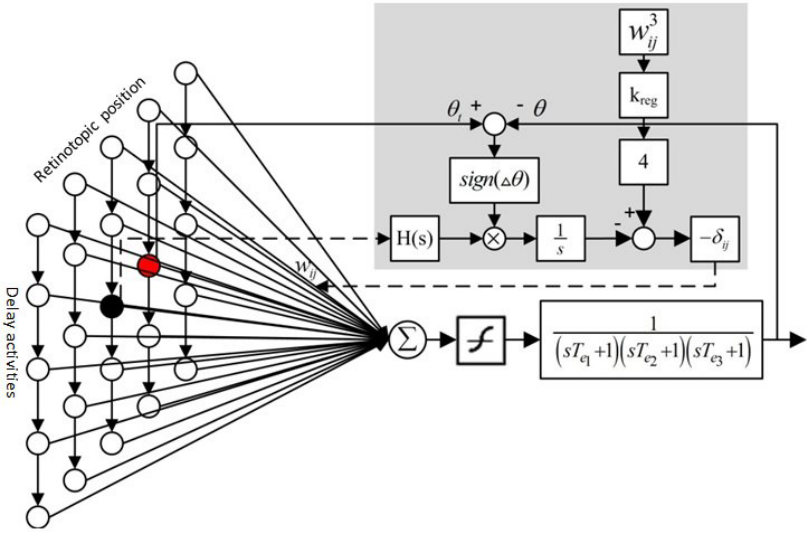


Figure 2.14: Neural based architecture for saccade modeling.

they do not propose any neural implementation for the optimization procedure they use. This is because they followed a pure mathematical procedure to solve the optimization problem. Optimization procedures used by these approaches are based on Pontryagin's extremum principle, which requires boundaries, initial and final conditions to calculate the special solution, otherwise the solution will be a global solution.

Saeb et al.[170] proposed a neural based optimization procedure which can be implemented relying on a neural function. Saeb et al.'s open-loop architecture (Fig. 2.14) provides local and biologically plausible solutions on one side and on the other side minimizes the cost function. The model consists of two parts: "open-loop control" and "adaptation". The model calculates the object position with respect to the retina coordinates and expressed as a visual error ($\Delta\theta$). When a target object appears, depending on the initial value of the visual error, the corresponding delay line is activated (Fig. 2.14). The saccade is initiated when the first neuron in a delay line starts firing. The next neurons in the delay line start firing with a Gaussian pattern (2.49) in a successive manner.

$$s_{ij}(t) = Ae^{-\frac{\left(i-\frac{t}{\Delta t}\right)^2}{2\sigma^2}} \tag{2.49}$$

where, s_{ij} represents the firing rate of neuron i in line j and Δt is the sampling rate. The neural command then can be calculated by (2.50)

$$u(t) = \sum_{j=1}^{N}\sum_{i=1}^{M} w_{ij}s_{ij}(t) \tag{2.50}$$

Symbol w_{ij} represents the weighted connection between neuron i in line j and N is the total number of delay lines and M is the number of neurons in a delay line. Then, the cost function is defined as (2.51)

$$J = \int_0^T |\Delta\theta| \mathrm{d}t + k_{\text{reg}} \sum_{j=1}^{N} \sum_{i=1}^{M} |w_{ij}|^n \tag{2.51}$$

Period T has a sufficiently large value so that the integral covers the whole movements and $n = 4$. A gradient decent method was used for minimizing the cost function. To this end, the derivative of the cost function (2.53) is calculated and the unknown parameters are replaced from the transfer function and the neural signal.

$$\frac{\partial J}{\partial w_{ij}} = \int_0^T \frac{\partial}{\partial w_{ij}} |\Delta\theta| \mathrm{d}t + k_{\text{reg}} \frac{\partial}{\partial w_{ij}} \sum_{j=1}^{N} \sum_{i=1}^{M} |w_{ij}|^4 \tag{2.52}$$

$$= -\int_0^T \text{sign}(\Delta\theta) \frac{\partial\theta}{\partial w_{ij}} \mathrm{d}t + 4k_{\text{reg}} w_{ij}^3 \tag{2.53}$$

where,

$$\text{sign}(x) = \begin{cases} -1 & \text{if} \quad x < 0 \\ 0 & \text{if} \quad x = 0 \\ +1 & \text{if} \quad x > 0 \end{cases} \tag{2.54}$$

The eye movement at time t can be calculated by convolving the neural signal in an impulse response (2.55), and by substituting $\frac{\partial\theta}{\partial w_{ij}}$ from (2.53), we will get (2.56). The detail of calculating the impulse response $h(t)$ is given in[5]. In this model, we use a linear 3-pole oculomotor plant with time constants $T_{e_1} = 232$ ms, $T_{e_2} = 14$ ms, $T_{e_3} = 4$ ms and DC gain $k = 0.217$. $H(s)$ and the oculomotor plant are considered the same function in Fig. 2.14 as mentioned in[5].

$$\theta(t) = \int_0^t u(\tau) h(t-\tau) \mathrm{d}\tau \tag{2.55}$$

$$\frac{\partial J}{\partial w_{ij}} = -\int_0^T \text{sign}(\Delta\theta) \left(\int_0^t s_{ij}(\tau) h(t-\tau) \mathrm{d}\tau \right) \mathrm{d}t + 4k_{\text{reg}} w_{ij}^3 \tag{2.56}$$

Now that we obtained the gradients, the weight adaptation rule can be defined based on the gradient

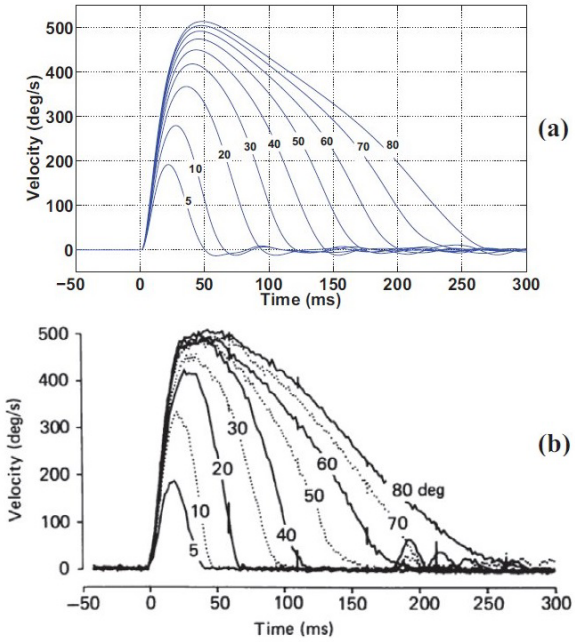


Figure 2.15: Comparison of model velocity profiles to experimental data: (a) adapted velocity profiles for target positions from $5°$ to $80°$, (b) experimental data taken from [57]. In both plots, the symmetry degrades gradually as the saccadic amplitude increases.

descent method as (2.57).

$$\Delta w_{ij} = -\delta_{ij} \frac{\partial J}{\partial w_{ij}} \tag{2.57}$$

The result from this model is consistent with the practical experiment as shown in Fig. 2.15.

2.4.2 Smooth pursuit eye movement

Smooth pursuit eye movements allow the eyes to closely track a moving object in the visual environment. It is one of the two ways that the vision system in primates and humans can voluntarily shift gaze, the other one is saccadic eye movement. Large errors are eliminated by saccade and the role of the pursuit is subsequently to match the eye velocity to the target velocity. Most people are unable to initiate pursuit without a moving visual signal. The pursuit of targets moving with velocities greater than a certain value tends to require another tracking mechanism. Smooth pursuit eye movements are controlled by visual feedback and thus the delays present in the visual system influence their characteristics (see Fig. 2.16). When the target object velocity varies rapidly and unpredictably (faster than 30°/s), these delays cause the accumulation of retinal error. Under such a condition, the strategy used by primates to track moving objects is to

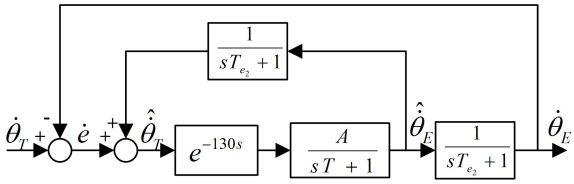


Figure 2.16: Young's model of smooth pursuit.

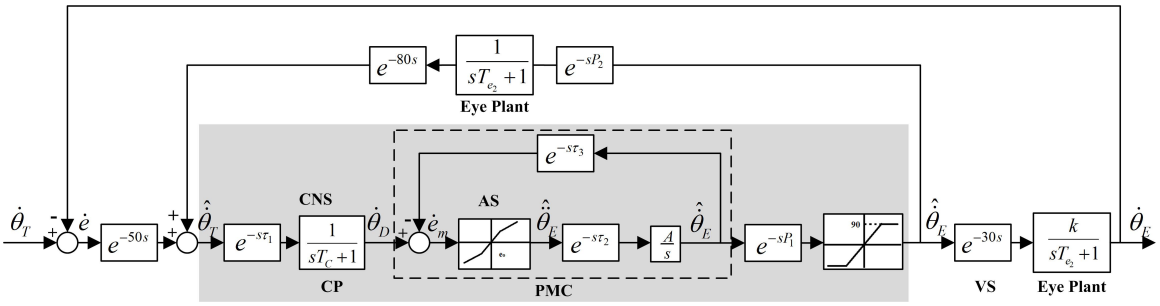


Figure 2.17: Robinson's model of smooth pursuit eye movement.

combine smooth eye movements with catch-up saccades that are rapid eye movements executed without visual feedback[232].

Young et al.[232] proposed the first model for a pursuit system in 1968 (Fig. 2.16). Lisberger et al.[109] discovered the saturation effects on eye acceleration and Optican et al.[145] demonstrated the motor learning in the pursuit system in 1981 and 1985 respectively. Meanwhile, the model remained unchanged and only more data has been added to the model. Robinson reviewed all the previous works from 1965 to 1985 and studied several parameters and proposed his final model (Fig. 2.17). This model answered nearly all the unknown previous questions and very few components have changed after him. He started his study from Young's model with few parameters as shown in Fig. 2.16.

The signals $\dot{\theta}_T$ and $\dot{\theta}_E$ represent the velocity of the target and the eye movement, respectively. The retinal slip signal, $\dot{e}$, is proportional to the eye acceleration. It is a velocity error and is a command to change the eye velocity. The model must be modified subsequent to $\hat{\dot{\theta}}_T$ to provide a signal proportional to the eye acceleration. The estimated target velocity, $\hat{\dot{\theta}}_T$, is declared by the central processing (CP) to be the desired eye velocity, $\dot{\theta}_D$. This process exhibits the activity of the Central Nervous System (CNS) with time constant T_C. The Pre-Motor Circuitry (PMC) provides the command to the oculomotor plants (eyeball and muscle) and contains an Acceleration Saturation (AS) with an integrator inside. The pre-motor circuit compares the current eye velocity $\hat{\dot{\theta}}_E$ to $\dot{\theta}_D$ to create a motor error signal, $\dot{e}_m$, that is amplified to drive $\hat{\dot{\theta}}_E$ closer to $\dot{\theta}_D$. This sub-system incorporates the findings of Lisberger et al.[109] and Optican et al.[145] in the model. The values associated with the time delays and gains are shown in Table 2.1.

The modified version of Robinson's model (Fig. 2.17) later was proposed by Barnes and Asselman in

	parameters and constants									
	τ_1	τ_2	τ_3	T_C	T_{e_2}	$\dot{e}_o$	P_1	P_2	k	A
values	15 ms	35 ms	30 ms	7 ms	15 ms	4 °/s	0.95	1.0	1.0	1.1

Table 2.1: Constant values in Robinson's model.

1991[4]. Barnes and Asselman added two new components to Robinson's model, prediction and memory, and kept the rest of his model. The schematic of Barnes and Asselman's model is illustrated in Fig. 2.18 in more detail. They believed that prediction plays an important role in the smooth pursuit mechanism. In addition, they added the head movement acceleration to the model which will be shown it is a very key factor in the activation of smooth pursuit rather than the vestibulo-ocular reflex mechanism.

There is not a huge difference between Robinson and Barne and Asselman's models, however the configuration and details are sometimes different. In Barnes and Asselman's model, the visual feedback from the retina, $\dot{\theta}_E$, and the target velocity, $\dot{\theta}_T$, (head velocity $\dot{\theta}_H = 0$) form the error signal. The mechanism has one external feedback which calculates the error signal, $\dot{e}$, and an internal feedback which controls the smooth pursuit movement. The internal feedback is composed of two basic components, a direct feedback of the retinal velocity error and a secondary pathway incorporating a predictive velocity estimator (PVE). The PVE is a sample and a hold mechanism that takes its input from an afferent copy of the gaze velocity. In the nervous system, "*afferent neurons*" carry nerve impulses from receptors or sense organs toward the central nervous system, while "*efferent neurons*" are doing reverse. So, when we say afferent copy, it means a copy of the sensory signals. The output from the PVE is controlled by a periodicity estimator (PE) that derives its control from the retinal velocity error signal and thereby estimates the periodicity of the most frequently changing component of the stimulus. Any kind of estimator such as neural based and stochastic based can be used in this circuitry. However, the neural one is strongly recommended since it is closer to the brain function. The PVE may obtain estimates of the required eye velocity either from direct sampling of the afferent copy, through the predictor, or from information stored in memory (MEM) based on the previous half-cycle of the response, as shown in Fig. 2.18 (gray area).

The output from PVE is partially integrated by a low pass filter $P(s)$, which has a time constant T_p of approximately $0 - 15$ s, and is summated with the retinal velocity error signal before passing through a band-pass filter $F(s)$ with a time constant T of approximately $0.5 - 1$ s. $F(s)$ and $P(s)$ have dynamic characteristics as defined in the diagram, where s is the Laplace operator. Constant K_v represents the visual feedback gain and τ the delay in visual feedback, approximately 0.1 s.

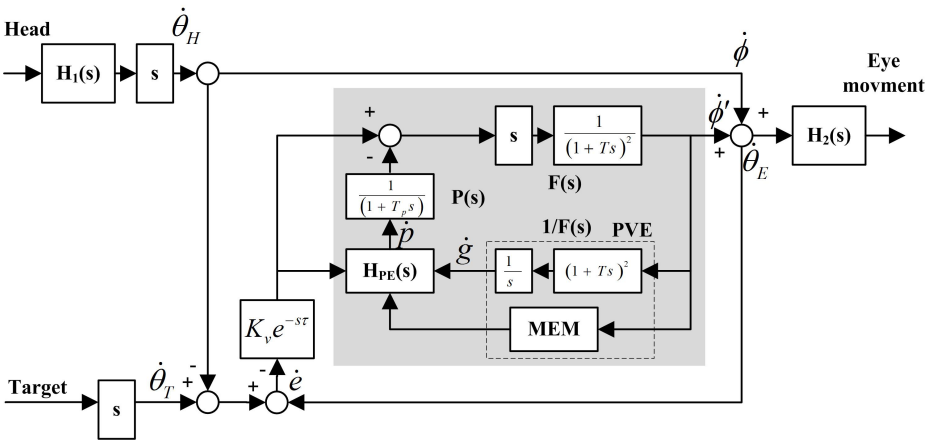


Figure 2.18: Smooth pursuit mechanism of the eye movement control system.

2.4.3 Optokinetic Reflex

The optokinetic reflex (OKR) is a combination of a saccade and smooth pursuit eye movements. Therefore, a perfect understanding of saccade and smooth pursuits means a complete understanding of the OKR. The OKR appears when an individual follows a moving object and finally the target object moves out of the field of view and the eyes move back to the position it was in, where the object has been seen at the beginning. The OKR happens when an observer watches a rotation drum with a stripped pattern (optokinetic drum). The head does not move in the OKR, and if it moves then vestibular ocular reflex happens, which will be discussed in the next section. The OKR has two components: slow and fast components. As mentioned above, the OKR is composed of saccadic and smooth pursuit eye movements. If the target object moves periodically, nystagmus can appear. Optokinetic nystagmus is caused either by the retinal slip velocity signal (fast phase) or by the retinal slip signal itself (slow phase). An example of a step response to a moving target can be simulated by the model shown in Fig. 2.17 and Fig. 2.18. The response (OKR) and its associated target movements are shown in Fig. 2.19.

2.4.4 Vestibulo Ocular Reflex (VOR)

The vestibulo-ocular reflex occurs only during head movement and serves to stabilize gaze on a stationary or moving object (Fig. 2.20). For instance, when an object is located outside the field of view and it needs to be seen due to any reason, the eye movement mechanism in the sensory-motor cortex uses the VOR. Another case of VOR is when the object is leaving the field of view and the eye needs to track down the object. One of the best models of the VOR has been proposed by Minor and Goldberg[133] in 1991. Their study was very vital for a better understanding of the eye movement mechanism.

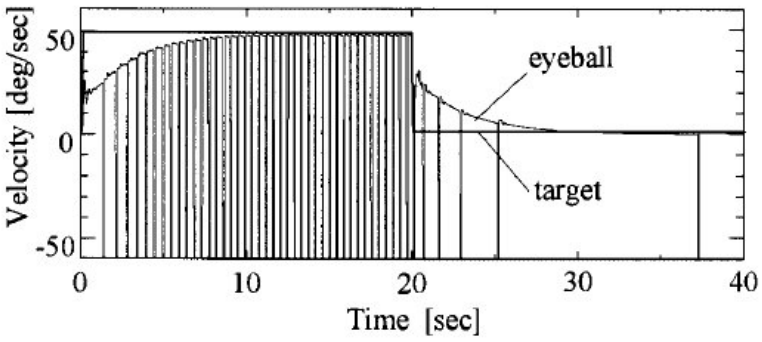


Figure 2.19: Step response to a sudden movement of the target object (optokinetic reflex against moving target).

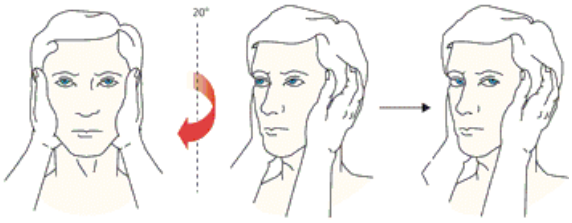


Figure 2.20: Eye movement due to Vestibular Ocular Reflex[40].

Semicircular-canal afferents, SCC, ($H_C = s\tau_{c_1}\frac{1+s\tau_v}{(1+s\tau_{c_1})(1+s\tau_{c_2})}$) provide inputs to the vestibular nuclei (VN) as shown in Fig. 2.21. Central pathways (H_B) link the VN to the oculomotor nuclei (OMN), and drives the oculomotor plant ($H_E = s\frac{1+s\tau_{E_4}}{(1+s\tau_{E_1})(1+s\tau_{E_2})(1+s\tau_{E_3})}$). In the VOR simplified model, Fig. 2.21, H_B is composed of three parallel pathways, i.e., an eye-position ($H_2 = \frac{g_2}{s}$), a head-velocity ($H_1 = g_1$), and a filtered eye-velocity ($H_3 = \frac{g_3}{s\tau_B+1}$). This arrangement is called a PID controller in the classic control theory. Usually, H_B refers to a much more complex mechanism than the one illustrated in Fig. 2.21. This is because different mechanisms serve in the substrate of an eye movement control system. As seen in Fig. 2.22, this component is replaced by another transfer function. For instance, in Fig. 2.17, this model is the smooth pursuit mechanism. Two delays exist in the direct pathway from the SSC to the eye: one between head rotations and the OMN (T_1) and the other (T_2) between the OMN and the eye movements, totally $T = T_1 + T_2 = 14$ ms. The rest of the parameters and their associated values are shown in Table 2.2. The value τ_v depends on the discharge regularity of the assumed afferent input.

2.4.5 Vergence

Vergence is the last item in the list of eye movements that we had mentioned above (Fig. 2.23). Vergence eye movements are commonly treated as distinct subclasses of eye movements, with largely separate anatomic

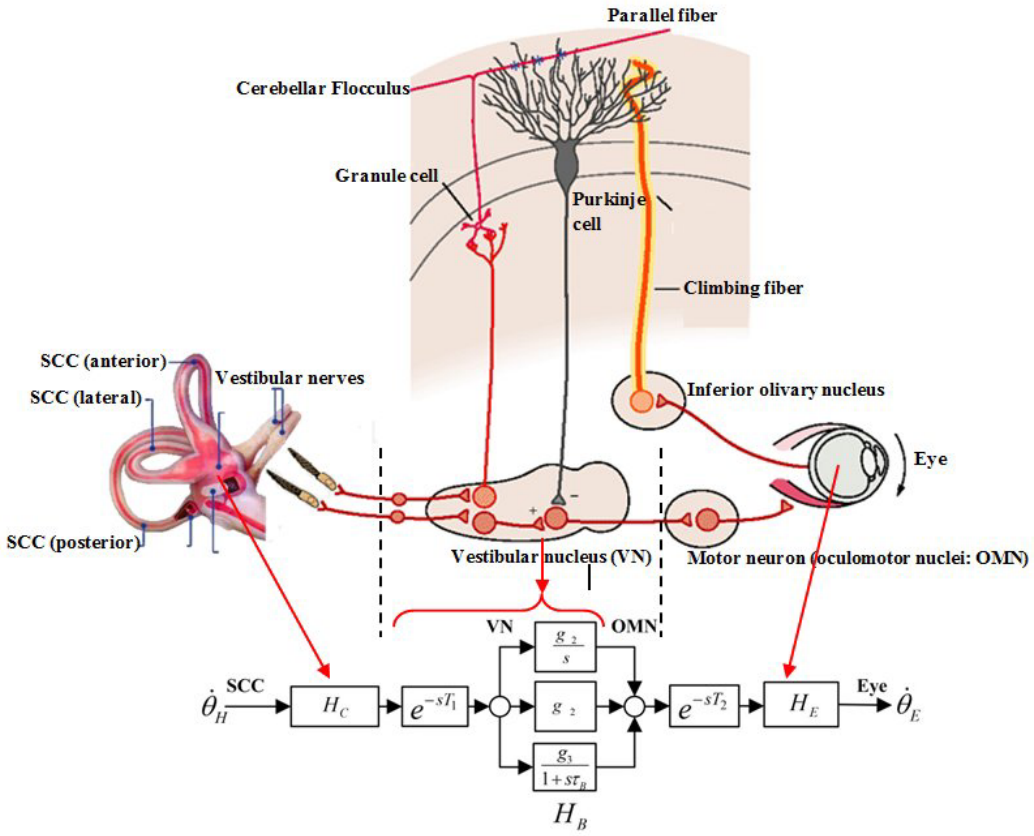


Figure 2.21: VOR simplified model along with neuron pathway.

	parameters					
values	τ_{C_1}	τ_{C_2}	τ_{E_1}	τ_{E_2}	τ_{E_3}	τ_{E_4}
	6 s	3 ms	0.28 s	37 ms	3 ms	0.14 s

Table 2.2: Typical parameters setting of the VOR model shown in Fig. 2.22.

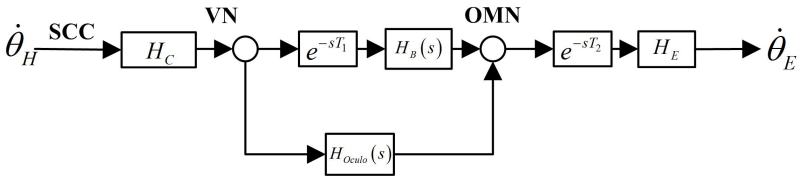


Figure 2.22: Complete VOR model provided for simulation purpose.

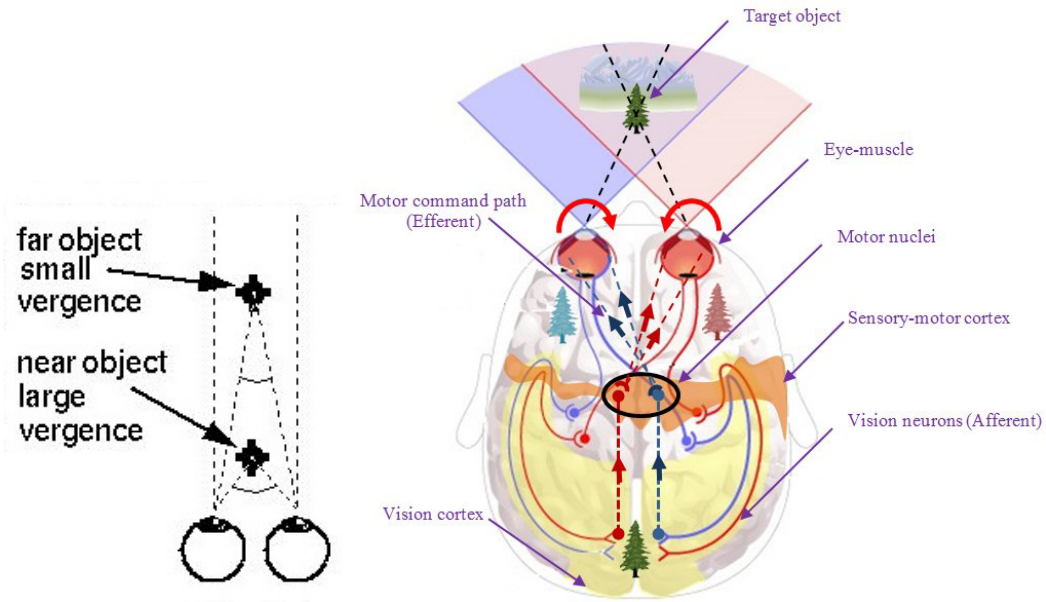


Figure 2.23: (a) Small and large vergence, (b) vergence biological model.

and physiological substrates, and control systems characteristics. A vergence is defined as the simultaneous movement in both eyes in opposite directions (as shown in Fig. 2.23.b in red) to obtain a single image by binocular vision. We can consider vergence as the last step in the sequence of eye movement which always exists in any type of eye movement, no matter the target object moves or the head. Vision neurons bring images from the left and the right eyes into the visual cortex to give a sensation of an object and analyze the object for further motor command generation. As seen, not only the vision signal (afferent) but also the motor commands (efferent) cross over each other, the former in the ganglion nuclei (GN) and the later in the motor nuclei (MN).

Zee et al. [234] established a solid base for research on vergence by reviewing previous works and proposing an analytical model (Figs. 2.24 and 2.25). Although this model explains how the saccade mechanism works in conjunction with vergence, there are models [121] that explains how other eye movement can be completed by vergence.

As shown in Fig. 2.23.b, the image from the left and right eyes should be mixed up to make a unique image. If the brain detects a little error between these two images due to the misalignment of the visual, it will automatically initiate the vergence mechanism. The error signal, (ΔC), is a desired change in the conjugate position (see Fig. 2.24). The signal is compared with an afferent copy of the change ($\Delta C'$) to produce a instantaneous Conjugate Motor Error (CME) which drives the Saccade Burst Neurons (SBN) to produce a Conjugate Velocity Command (CVC) by accessing to the nonlinearity innately existing in the neuron model. The CVC also feeds back by a Resettable Integrator (CRI) and a time delay. Omnidirec-

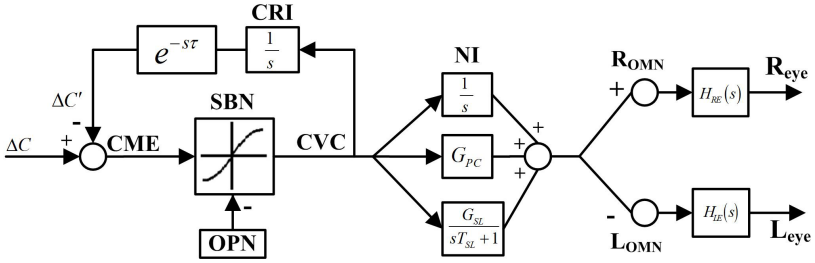


Figure 2.24: Saccade-vergence control mechanism.

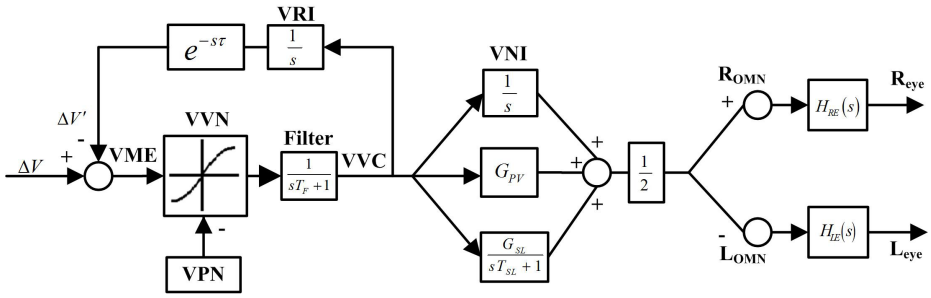


Figure 2.25: Modified control mechanism based on vergence.

tional Pulse Neurons (OPN) are inhibited to initiate saccades or kept inhibited till saccade is active. The velocity signal directly is sent to oculomotor neurons after being integrated by NI, amplified by gain G_{PC} and filtered by a slide filter with gain G_{SL} and 3 dB frequency $f = \frac{1}{T_{SL}}$, as shown in Fig. 2.24.

The error signal is the velocity difference between the left and the right eyes (ΔV), a desired change in Fig. 2.25 is based on angular velocity. As shown in Fig. 2.25, the signal is compared with an afferent copy of the change in the vergence velocity ($\Delta V'$) to produce an instantaneous Vergence Motor Error (VME). This signal drives the vergence velocity neurons (VVN) to produce a Vergence Velocity Command (VVC) according to the nonlinearity as characterized in[123]. The VVC signal is filtered by a low-pass filter with a time constant (T_F) of 0.01 s and 0.05 s for convergence and divergence respectively. The rest of the process is similar to that of Fig. 2.24 except the oculomotor neuron command is multiplied by 0.5 before entering to the eye plants to give equal velocity to both eyes during vergence.

So far, saccadic eye movement, smooth pursuit, OKR, OVR and vergence as five fundamental components of an eye movement have been discussed in section 2.4 and the biological mechanisms were analyzed. Besides, a computational model for each component were derived from the literature or existing models were extended. Section 2.4 has already explained the brain cortex modeling using Artificial Neural Network (ANN). Now, these five eye movement components and ANN will be used to build the complete eye movement mechanism. The mechanism will explain the vision neural path for one eye (monocular). Then, the model will be replicated and ganglion nuclei as well as motor nuclei will be added to explain the

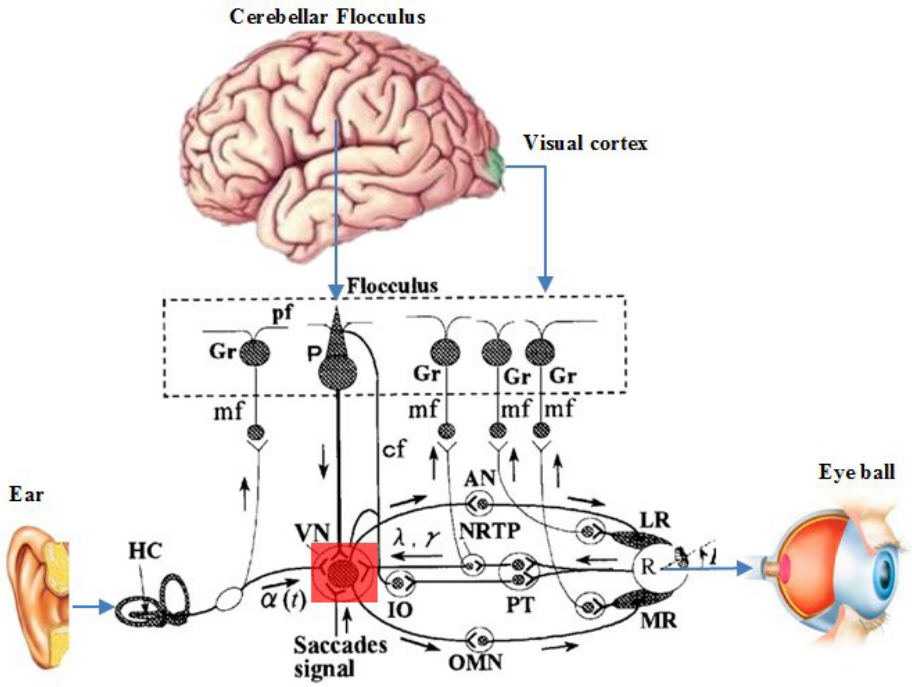


Figure 2.26: Neural path of the oculomotor mechanism.

complete binocular vision system. Later, the binocular vision model will be employed in order to explain the sensory conflict and deduce the condition in which visually induced motion sickness takes place.

2.5 Binocular vision model

Vergence is the sole component which connects the monocular vision mechanism to binocular vision and explains the relation between the two eyes. As mentioned, all of the previous models (see section 2.4.1 to 2.4.5) will be integrated into a unique oculomotor model to build a complete monocular vision model. The vision neural path and its connection with the cortex will be explained in the following section. These explanations will help to understand how ANN should be embedded in the model.

2.5.1 Oculomotor neural path

The neural path of the oculomotor mechanism based on a previous study[77] is illustrated in Fig. 2.26.

The retinal image and the head movement are regarded as inputs, horizontal rotation angle E, as the output of the oculomotor mechanism, and the eyeball and ocular muscles are the eye movement plant. The head horizontal/rotational movement is detected by the horizontal canal (HC) of the otolith and

considered as the vestibular signal (Fig. 2.26). The vestibular signal uses two paths to initiate the control plant:

1. Vestibular Nucleus (VN), Oculomotor Nucleus (OMN), Medial Rectus muscles (MR);
2. VN, Abducent Nucleus (AN), Lateral Rectus Muscles (LR).

The retina signal is extracted from the retina image and can pass through two paths:

1. Retina (R), pretectum (PT), nucleus reticular tegmental pontis (NRTP), VN, OMN, MR;
2. VN, AN, LR.

The saccadic signals inputs from the Flocculus (a small lobe part of the VOR system helping stabilize gaze during head rotation) are superior to VN.

The structure of the Flocculus has been described in[77] in detail and will be considered as a black box here. The input signals are transferred through the mossy fibers (mf). The mf stretch the receptors of the eye muscles (LR, MR) that are considered as paths to transfer the signals of the eye rotational angle as shown in Fig. 2.26. The VN plays an important role in the oculomotor mechanism and is thought as the reason of sensory conflict. Reason and Brand[158] suggested a theory based on this conflict to explain motion sickness, which is known as the "sensory conflict theory". This theory was completed and detailed by Bos et al.[17] for visually induced motion sickness in 2008 which will be discussed in more detail in section 2.7.

2.5.2 Monocular vision mechanism

The mathematical model of the eye movement mechanism (Fig. 2.27) is derived to simulate the oculomotor neuron path equivalent to the one that has already been explained in section 2.5.1 (see Fig. 2.26). This model explains the neuron path of a single eye movement only. In Fig. 2.27, only the rotational movement of the head (θ_H) and the target (θ_o), the retinal slip and the retinal slip velocity represented by $\epsilon(s)$ and $o_v(s)$, respectively, will be considered to simplify the final interpretation of the binocular vision model (see section 2.5.4). The angle of the optical axis, E, always moves in the opposite direction by the VOR mechanism (Figs. 2.20 and 2.21) during head rotation. Therefore, an inverted head angle is used in the input of the model. The Flocculus is modeled by a simple multi-layer perceptron Artificial Neural Network with a training stage, as shown at the top of Fig. 2.27.

To simplify the discussion, the total eye plant (muscles and eyeball) was modeled by a first-order system, where, T_e is the time constant of the eyeball, g is a amplification gain. The neural integrator between the VN and the OMN is expressed as the sum of an imperfect integrator and a direct path, where T_n is the time constant of the integrator, and g_e is the gain of the direct path[237]. The symbols $\alpha(t)$, γ and λ represent the synaptic transmission gains of the neural fibers that transfer the head velocity signal, the retinal slip

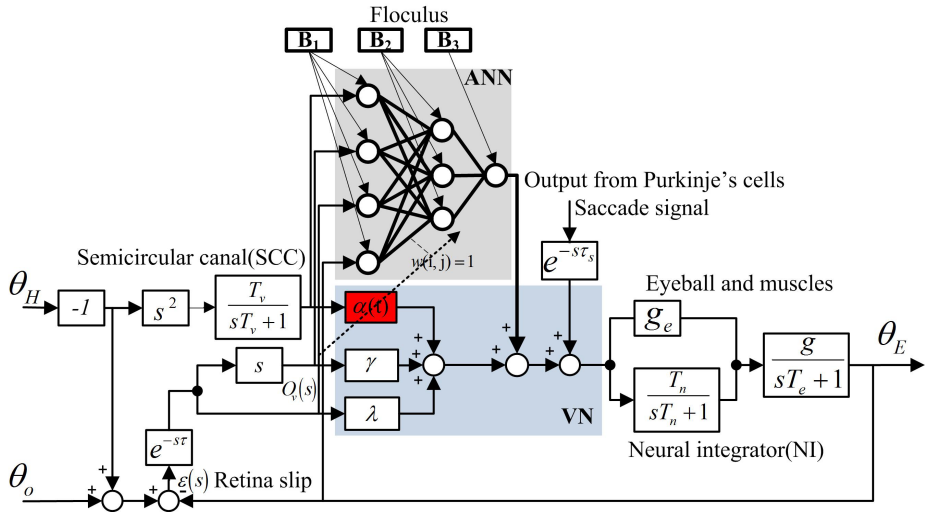


Figure 2.27: Complete monocular vision mechanism.

velocity signal, and the retinal slip signal. We need a training process to extract the weights and the bias values of the neural network and $\alpha(t)$.

The training process for the Flocculus is very similar to ANN. Plenty of training algorithms can be found in the literature but the one that has been explained in [100] describes the training algorithm which is inspired by Biological Neural Network (BNN).

Adaptive performance is not only governed by the Flocculus, but also by the vestibular nucleus [100]. Since the neurons in the VN do not have a complex network structure as the Flocculus, the VN can be considered as a simple network with a training system that does not have enough ability to make an accurate learning like the Flocculus, but can learn quickly due to its simple structure [100]. The eye movement control mechanism suggests that the performance of the whole training system is improved by the combination of two learning systems: 1) one ANN (Flocculus) with slow and precise learning which will be feasible by setting the learning constant small, 2) another ANN (VN) with a simple structure and learning system with quick convergence. It can be thought that a simple learning system not only improves the total learning speed, but also makes the main learning system avoid local minimums. Thus, the following algorithm is used to simulate the learning performance of the vestibular nuclei. The training algorithm of the second network is adapted from (2.58).

$$\alpha\left(T(k+1)\right) = \alpha(kT) + \delta\left(\rho - \alpha(kT)\right) + \epsilon\xi(kT+\tau)H_v(kT+\tau) \tag{2.58}$$

where, δ is a forgetting constant, ρ is the standard value of $\alpha(t)$, ξ is the learning constant, k is the learning time, and T is the cycle time. Different steps of the training algorithm were clearly explained

in [237]. In spite of previous simplifications, the model shown in Fig. 2.27 is still complicated for explaining binocular vision. So, we will go one step ahead to make a simpler model for studying binocular vision.

2.5.3 Monocular vision simplified model

We assume $T_n \gg g_e$ and $T_e = g_e$ to simplify the equation and the transfer function, and consequently the monocular vision model in Fig. 2.27. Now, we can rewrite the neural integration (NI) as (2.59)

$$\mathrm{NI}(s) = g_e + \frac{T_n}{sT_n + 1} = \frac{sg_eT_n + (T_n + g_e)}{sT_n + 1} \tag{2.59}$$

Using the first assumption, (2.59) is written in the form of (2.60)

$$\mathrm{NI}(s) = \frac{T_n\,(sg_e + 1)}{sT_n + 1} \tag{2.60}$$

By imposing a second condition and accepting an eye movement pole cancellation which has been proven and accepted by previous research [20] and studied by biological experiment, the eyeball transfer function, $(\frac{g}{sT_e+1})$ can be expressed by (2.61)

$$H_E(s) = \frac{T_n\,(sg_e + 1)}{sT_n + 1} \times \frac{g}{sT_e + 1} \tag{2.61}$$

Gain $g = 1$, therefore, the final transfer function of the eyeball and the NI is simplified to (2.62)

$$H_E(s) = \frac{T_n}{sT_n + 1} \tag{2.62}$$

Fig. 2.28 shows the simplified version of Fig. 2.27 which will be used later in binocular vision. This model will help to understand the eye movement mechanism better. For instance, (2.63), (2.64), and (2.66) explain the VOR, the OKR, the saccadic movement respectively. Only the effect of the Purkinje cell [237] from the Flocculus is neglected in this model for more simplicity. The Purkinje effect is well described in [237] which will be added to the final binocular model as an ANN in the next section. However, for deeper consideration in a specific application, it needs to be considered similar to that shown in Fig. 2.27.

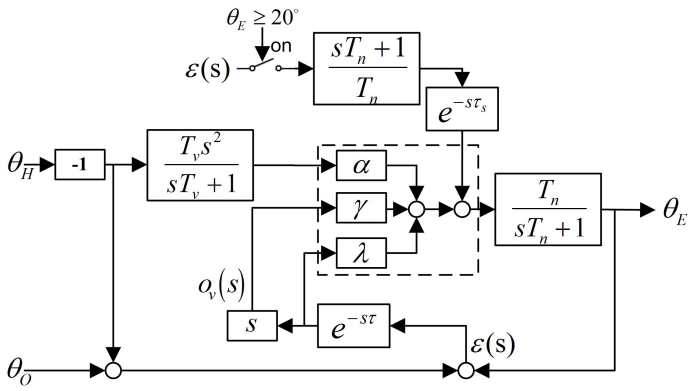


Figure 2.28: Simplified version of the eye movement mechanism.

$$H_{\text{VOR}}(s) = \frac{\alpha T_v T_n s^2 + T_n(\gamma s + \lambda)\,(T_n s + 1)\,e^{-s\tau}}{(T_n s + 1)\,[(1 + \gamma e^{-s\tau})\,T_n s + \lambda T_n e^{-s\tau} + 1]} \tag{2.63}$$

$$H_{\text{OKR}}(s) = \frac{T_n(\gamma s + \lambda)e^{-s\tau}}{(T_n s + 1) + T_n(\gamma s + \lambda)e^{-s\tau}} \tag{2.64}$$

$$\epsilon(s) = \theta_E(s) - \theta_o(s), \theta_H(s) = 0 \tag{2.65}$$

$$H_s(s) = \frac{(sT_n + 1)\,e^{-s\tau_s} + (\gamma s + \lambda)e^{-s\tau}}{sT_n + 1} \tag{2.66}$$

In the next section, this model will be repeated for the left and right eyes and the ganglion nuclei will be added to join the two eyes and complete the binocular vision model.

2.5.4 Binocular vision

A simple sketch of binocular vision of the human being is demonstrated in Fig. 2.29. The optic nerves carry visual information from each eye to the brain via ganglion cells. These nerves travel from the back of the eye ball (retina), entering the brain through the orbit at a small "hole" (the optic canal) in the skull bone. The optic nerves of the left and right eyes meet and cross each other to form the optic chiasm (since the visual cortex does not receive signals from each eye unilaterally, the right eye vision is partially perceived on the left side of the brain and vice versa). Half of each optical field is directed to the opposite part of the brain. This occurs when the bundled fibers of the optic nerves meet and cross at the optic chiasm (cross road), located just a few centimeters inside the brain. The true vision information is formed and processed in the visual cortex of the brain. The eye movement mechanism in the sensory motor cortex of the middle brain processes the visual and vestibular data to generate an ocular motor command. Different elements of the eye movement mechanism were introduced in section 2.4.

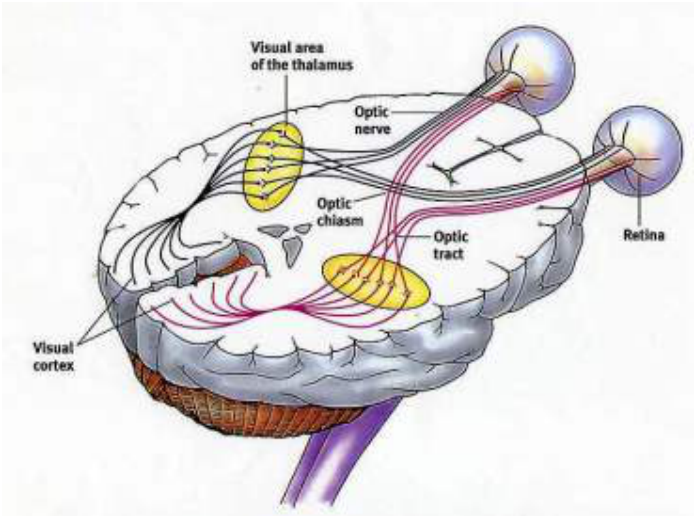


Figure 2.29: Neural pathway of the binocular vision system.

A complete model of binocular vision is achieved by combining all the previous components and embedding the optical never cross-over effect (Fig. 2.30). A simplified version of binocular vision was derived by Zhang and Wakamatsu [237]. His model forms the back bone of this research while the saccadic eye movement mechanism is integrated and the Flocculus network is embedded into the ANN for simplicity. The model is modified to be more consistent with its biological equivalent as shown in Fig. 2.29, and the previous model of the monocular vision mechanism as introduced in Fig. 2.27 and Fig. 2.28 in section 2.5.3. The symbol of the constant values were changed slightly in this model compared to that of Fig. 2.28 because there are too constants in the model. In general, α and β, γ are used to explain the VN and the OMN constants respectively. Besides, the rule of the ganglion nuclei (cross-over effect) is incorporated in the model, the blue area in Fig. 2.31. As seen, not only the incoming vision information is crossed over but the outgoing motor command for each eye is affected by the other eye too.

The proposed model is complete for studying the effect of different parameters involved in an eye movement and better understanding of the mechanism of binocular vision, although this model is a little bit complicated in general. As a result, this complexity motivates us to think of a simple representation such as the one shown in Fig. 2.32, which can be applied to the qualitative study conducted later only to have a general idea. Fig. 2.32 introduces a block diagram-based of the model which makes the final study easier and the model more comprehensive. In section 2.5.5, it will be explained how the model was derived and what each element means. Later, the model will explain some visual conflicts during sensory rearrangement.

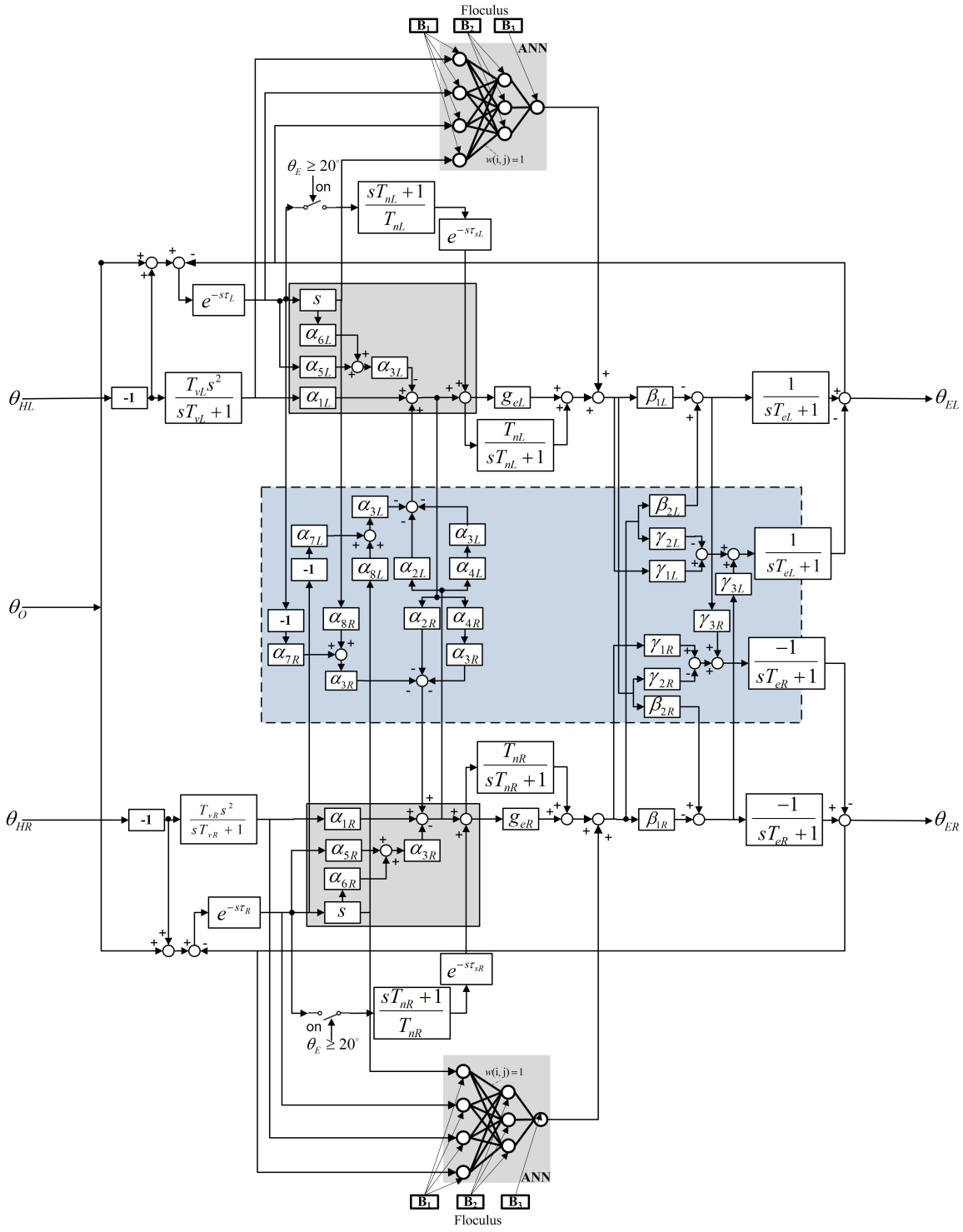


Figure 2.30: Complete model for binocular vision simulation with Flocculus comportment.

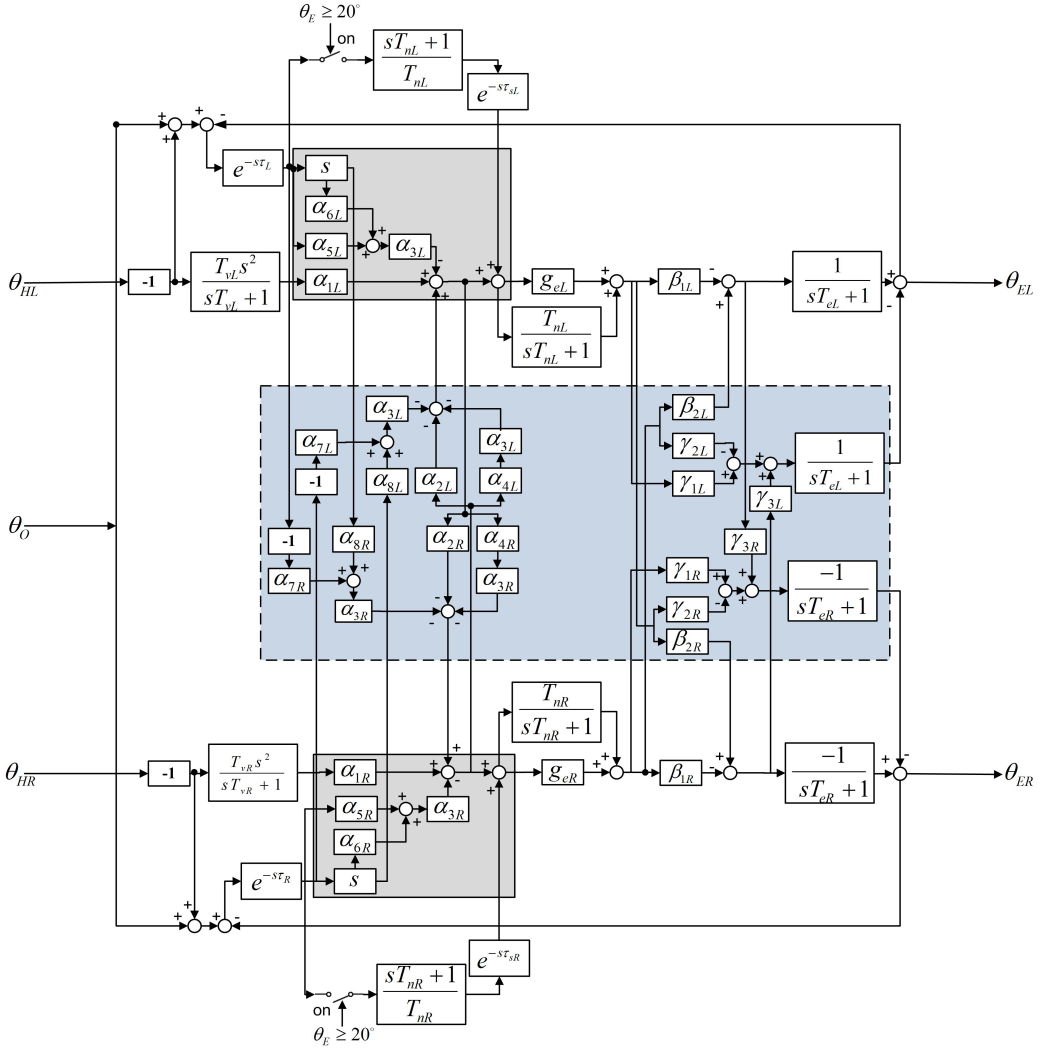


Figure 2.31: Simplified binocular vision neural model.

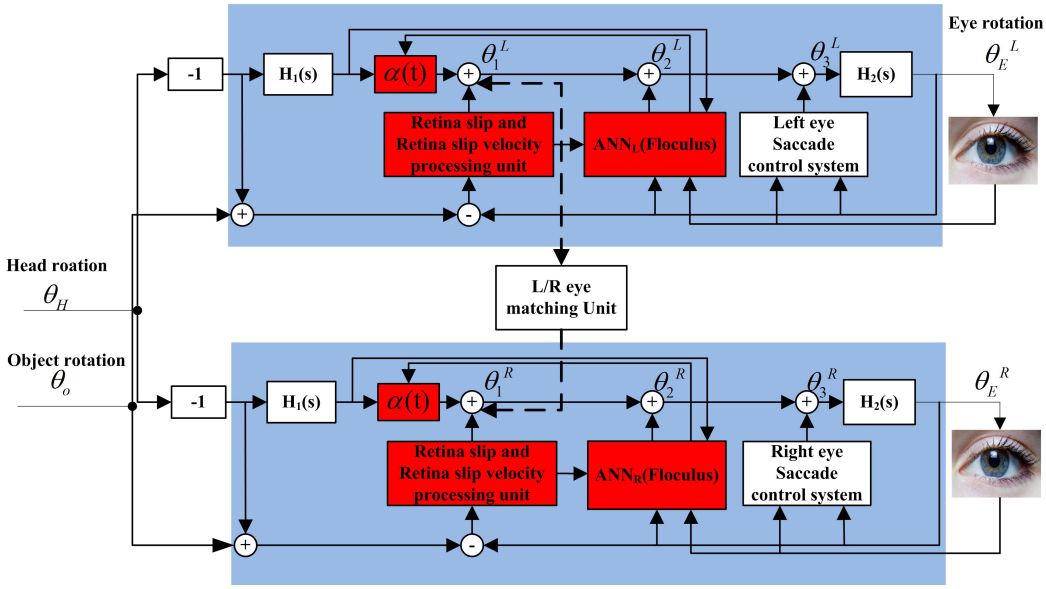


Figure 2.32: Simplified binocular vision model.

2.5.5 Simplified control mechanism of monocular vision

Zhang and Wakamatsu[238] derived an eye movement control system from human vision for robotic applications. The model is capable of explaining more visual effects and phenomena occuring during eye movements. However, the model will be used to interpret binocular disparity[211] in detail and with more precision later. However, some elements such as saccade still need to be integrated in the model comparing the model shown in Fig. 2.27. We incorporate the saccade control mechanism (Robinson[162]) into the model and summarize the other elements in block A, B, and C to make the process more understandable (see Mergenthaler and Engbert[129] for more detail). The detail of the model, only for one eye, is explained below to make each element clear and understandable (Fig. 2.33). Then the model is repeated for the left and right eyes in conjunction to construct a model for binocular vision similar to what shown in Fig. 2.32.

Input/output: The eye moves due to either head or object motion or both. The head, the target object, the system inputs, and the eye movement (degree), the system output, are denoted by θ_H, θ_o, and θ_E respectively in Fig. 2.33. Object motion is detected by the visual cortex, V5 region of the visual cortex (Born[15]), and head movement by the semicircular canal and cochlea (Oman[144]). The head rotation angle is converted to a nerves signal by a "semicircular canal transfer function", $H_1(s)$. The transfer function is a function that convert the signal from one type to another. Since, the optical axis always moves in the opposite direction of the head rotation which is multiplied by -1 before entering into the semicircular transfer function as stated above.

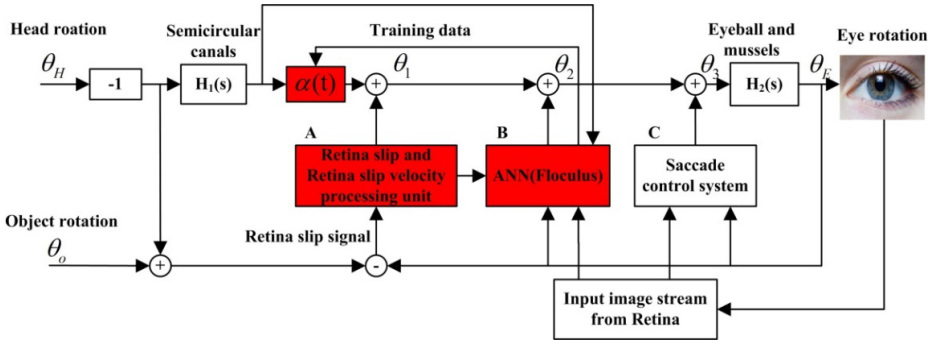


Figure 2.33: Eye movement control system for a single eye.

Blocks: Block $\alpha(t)$ represents the synaptic transmission gains of the neural fibers that transfer the head velocity signal. The synaptic gains are modeled by a set of neurons (a simple perceptron ANN with a training algorithm explained in section 2.5.2) and is trained in real-time by the data coming from the Flocculus (block B in Figure 2.33). It acts as a transfer function of the vestibular nucleus where the ocular and vestibular signals are mixed up. When a movement is detected, the oculomotor control system is triggered to perform different kinds of eye movements according to the position, movements of the target object, and the head. The final motor command is converted to a rotation angle by an "Eyeball and muscle transfer function", $H_2(s)$, before entering into the eye muscles (see section 2.4.4 for more detail). The retina slip processing unit (block A in Fig. 2.33, in red) and its parameters were well defined and discussed by Zhang and Wakamatsu [237]. The saccade control system (block C) is adapted from Robinson [162] and Saeb et al. [170] works. Most eye movements are composed of several types, which come one after another and with co-ordination. As a result, more than one parameter is usually involved in a single movement analysis.

Parameters: Each component of the oculomotor can be mathematically modeled and described by equations. The classic control theory can describe the relation between the input (retina image) and the output (vertical and horizontal amount of eye rotation) and the characteristics of eye movement. Although several components are involved in eye movement, only the effect of saccade (θ_3), the Flocculus (θ_2) and the retina slip (θ_1) will be described to give an insight into binocular disparity. For that, the oculomotor model for binocular vision will be discussed and then the effect of these three parameters on binocular disparity will be discussed.

2.5.6 Simplified oculomotor mechanism of binocular vision

A simplified block diagram of the eye movement control system for binocular vision is shown in Fig. 2.32. To the best of our knowledge, this model is the most completed binocular model for human vision. Exactly the same control system as Fig. 2.33 is used for each eye except the "L/R eye matching unit" is added to the

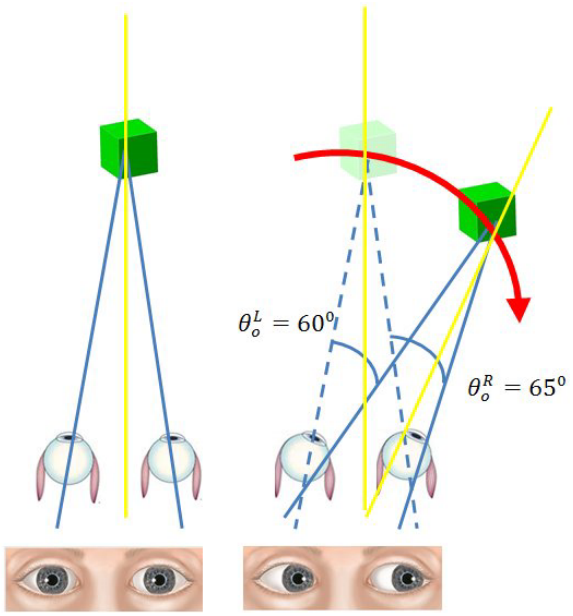


Figure 2.34: Difference between the L/R eyes and the object position at time $t = 0^+$ (immediately after object movement).

binocular control system just to model the Lateral Geniculate Nucleus (LGN) which models the cross over effect of the optical nerves. An analytical definition of the LGN function and its components were perfectly detailed by Zhang and Wakamatsu[237] and intensive research has been conducted to study the effects of different parameters on binocular vision. Now, we are ready to analyze the effect of blocks A, B, and C in Fig. 2.33 to show how binocular disparity happens and is intensified by rotational motion.

2.5.7 The effect of the Flocculus, saccade and the retina slip velocity on binocular disparity

Retina slip and retinal slip velocity effects (θ_1)

Considering only rotational movement, the retina slip signal (ϵ) at time t is defined as the angular difference between the current position of the object with respect to the head and the optical axes, $\epsilon = \theta_E - (\theta_O - \theta_H)$, as shown in Fig. 2.28.

If we assume the head is fixed, there is a natural difference between the left and right eyes at time $t = 0^+$, immediately after object movement, as shown in Fig. 2.34. Since this difference (for instance for values shown in Fig. 2.34) is multiplied by an internal coefficient (velocity: $\gamma = 0.5$, angle: $\lambda = 0.01$ as determined by Zhang and Wakamatsu[237]), the first difference between the left and right eyes appears (2.67). The same calculation with a similar assumption is possible for the retina slip velocity. The velocity effect is stronger because the coefficient is bigger ($\gamma > \lambda$).

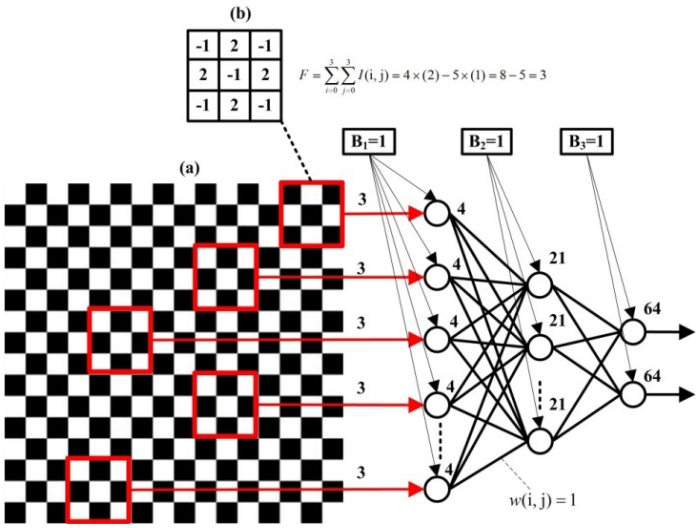


Figure 2.35: Output of the Flocculus network without movement.

$$\theta_1^R = \left[\theta_E^R - \left(\theta_O^R - \theta_H^R\right)\right]\lambda = 65 \times 0.01 = 0.65, \theta_1^L = 60 \times 0.01 = 0.6$$
$$\theta_1 = \theta_1^R - \theta_1^L = 0.65 - 0.6 = 0.05 \tag{2.67}$$

Faster rotation (bigger rotational velocity) leads to a bigger difference which in turn creates larger disparity between the two eyes. However, the control system decreases the difference during the transition to the new point by its natural feedback (Fig. 2.33).

Flocculus effect (the effect of Purkinge cells, θ_2)

The second difference appears due to a mismatching between the input pattern and the pattern existing in Purkinge cells in the Flocculus during rotational movement. A simple 3-layers perceptron neural network with a "purelin transfer function", 5, 3 and 2 neurons in the input, hidden and output layers respectively (Fig. 2.35), is selected to study the effect of the difference in the Flocculus due to a pattern mismatching on binocular disparity. We will calculate the network output for two conditions:

Flocculus output for an original pattern We assume the Flocculus neural network has been trained by the pattern and the feature as shown in Fig. 2.35.a. Therefore, the bias and weighting values for each layer are known. Again for simplicity, we assume all the bias and weighting values equal to "1", as shown in Fig. 2.35. 5 rectangular features are selected and the feature value (F) is calculated based on

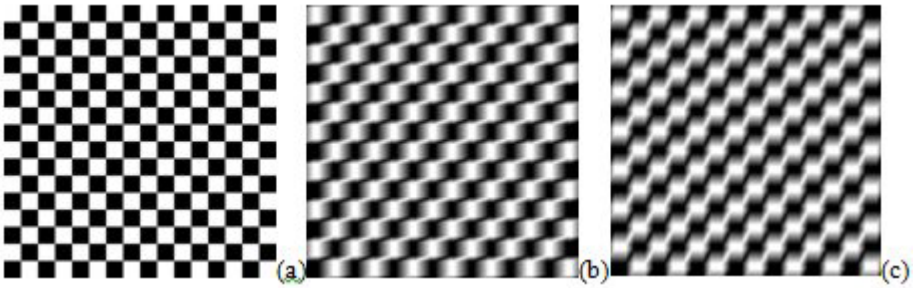


Figure 2.36: a) Original pattern $\omega = 0$ °/s, b) pattern in motion for $\omega = 20$ °/s, c) 60 °/s.

the pixel values shown in Fig. 2.35.b. Then the output of each layer in the Flocculus network is calculated using (2.68).

$$o_j = \sum_{i=1}^{n_1} (w(i,j)I_i + B_j) \tag{2.68}$$

where, n_i is the number of neurons in the previous layer, w and B are the weightings and bias values and I_i, the input values. If the calculation is repeated 3 times, $O_O = 64$ is yield for the original pattern (Fig. 2.36.a). For a given original pattern, the effect of the speed on the pattern during the movement can be modeled by applying simple "motion blurring filters". The parameters of the filter for different speeds must be set as described by Kim and Ko[101]. Fig. 2.36 shows the original pattern (a), $\omega = 0$ °/s, and the pattern in motion (b, c), $\omega = 20, 60$ °/s.

FLOCCULUS OUTPUT FOR A PATTERN IN MOTION We accept the training values from the original pattern and will replace the input pattern with a pattern in motion (Fig. 2.36.b, 20 °/s) in Fig. 2.35. The arrangement of the Flocculus will be as shown in Fig. 2.37.

The output of the Flocculus network in Fig. 2.37 is calculated using (2.68), $O_M = 25.8$, there is a big error between the output of the network to the original pattern and the pattern in motion, $\theta_L = 64 - 28.5 = 38.2$, and this error in turn leads to a wrong estimation of $\alpha(t)$ and more disparity between the left and right eyes. The bias and the weight of the left and right Flocculus networks are not really equal naturally. It means either the bias or weight is different in Fig. 2.37, $B_j^R \neq B_j^L$ or $w_{ij}^L \neq w_{ij}^R$. We can assume the same values and see the error due to the difference in the training data. This difference also leads to more disparity between the two eyes because it is accumulated with the motor command directly. The second source of disparity due to a pattern mismatching in the Flocculus network is described by (2.69).

$$\theta_2 = \theta_2^L - \theta_2^R \tag{2.69}$$

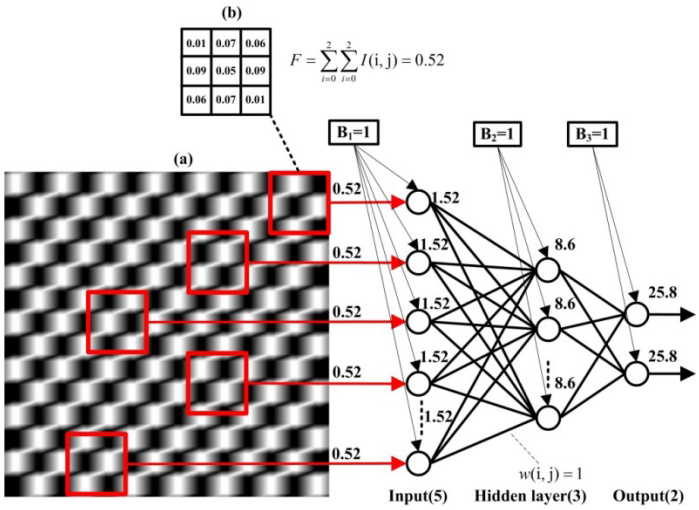


Figure 2.37: Output of the Flocculus network for a pattern in motion.

Saccadic effect (θ_3)

The saccade control system as described by Saeb et al.[170] is a neural network and can be modeled in a similar way as the Flocculus, however it is slightly different in detail. The network is trained by a pattern (a) in the real environment while a pattern (b) is captured by the retina and processed during rotational movement. Therefore, it creates an extra error which enters into the muscle transfer function and gives rise to erroneous rotation consequently. The saccade control system is activated only when the position of the eye and the object is different as described by Kim and Ko[101]. As a result, this initial error is accumulated with the two previous effects (θ_1, θ_2) and exacerbates binocular disparity (2.70).

$$\theta_3 = \theta_3^L - \theta_3^R \tag{2.70}$$

The retina slip velocity and saccade are dependent systems therefore, we cannot say which one is the main cause of the difference between the two eyes. However, we can say, both of them contributes in the disparity. Finally we can say binocular disparity results from the accumulation of three factors (2.71):

$$\Delta\theta_{L-R} = \theta_1 + \theta_2 + \theta_3 \tag{2.71}$$

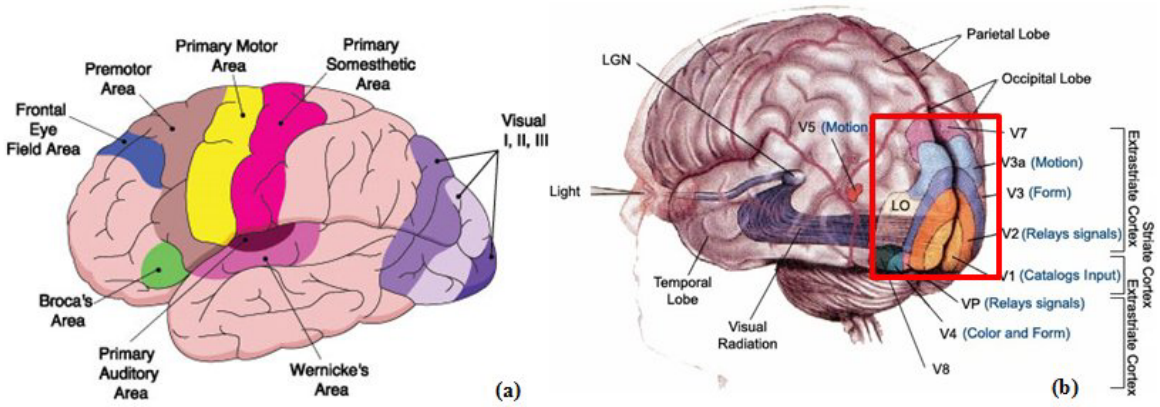


Figure 2.38: (a) Different sections of a brain [Igou], (b) detail of sub-areas associated with the visual cortex [Wikipedia].

2.6 Visual cortex anatomy and biological mechanism of Visually Induced Motion Sickness (VIMS)

2.6.1 Brain anatomy

The brain is the most complicated element in the human Central Neural System (CNS) and plays a vital role in different mechanisms of the human body. It is composed of different areas and each area is responsible for controlling and analyzing a unique organ of the body independently or dependently to other areas. The detailed map of the brain areas is shown in Fig. 2.38.a [Igou]. The most important areas from an ocular-vestibular perspective are the primary motor, the premotor, the visual and primary auditory areas.

2.6.2 Visual cortex

The visual cortex in turn consists of several sub-areas as specified in Fig. 2.38.b [Wikipedia]. Fig. 2.38.b shows the position and the function of each sub-area of the vision cortex in the occipital lob of the brain. The visual cortex has several sub-areas named from V1 to V8 and VP. These sub-areas are responsible for motion detection and analysis (V5 and V3a), form and color recognition (V3 and V4), and relaying the processing results to the other parts of the brain (V2). The V5, V3a, and V4 are more important than the others in this research because they are responsible for motion and color. Later, it will be shown that visual stimulation is one of the major sources of cyber sickness. This kind of sickness is normally induced by inharmonious moving scenes and comes along with dizziness.

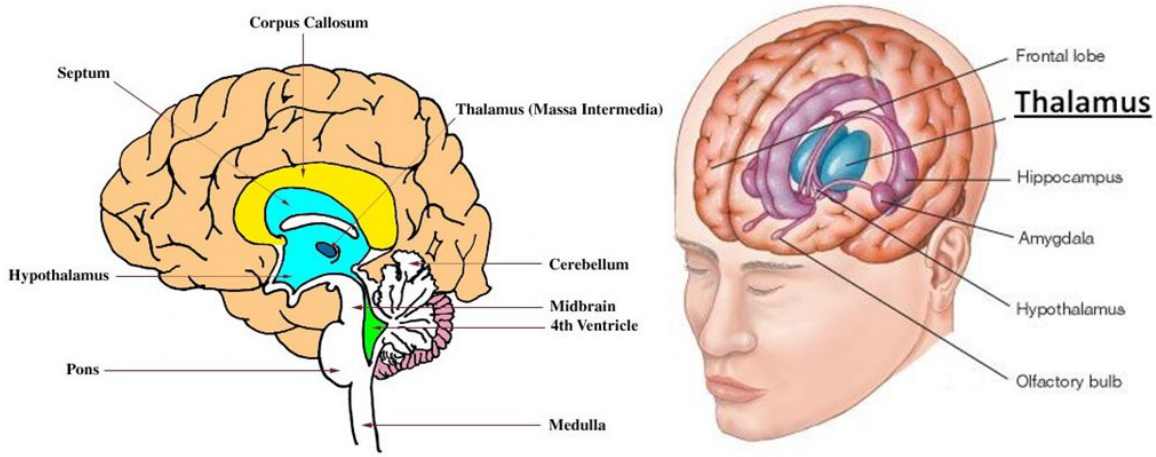


Figure 2.39: Hypothalamus position in the brain [NCI].

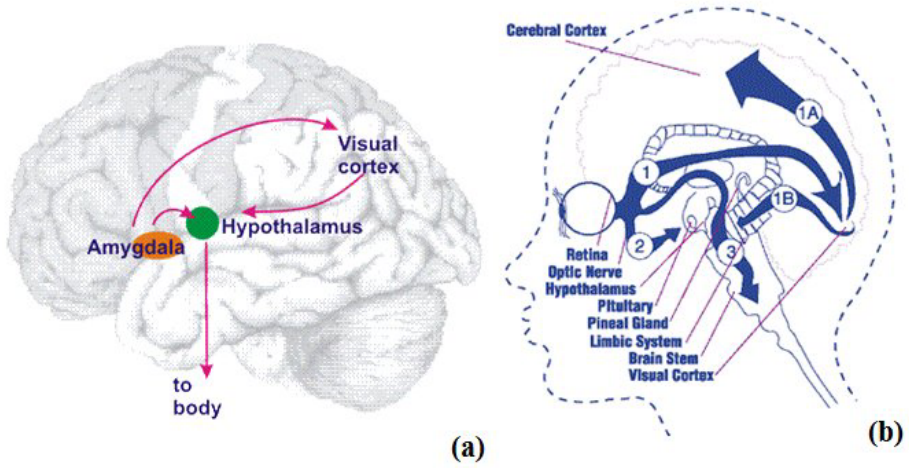


Figure 2.40: Connection between the hypothalamus and the visual cortex [142].

2.6.3 Hypothalamus and its function

The hypothalamus is a portion of the brain that contains a number of small nuclei with a variety of functions. One of the most important functions of the hypothalamus is to link the nervous system to the endocrine system via the pituitary gland. The hypothalamus is responsible for certain metabolic processes and other activities of the autonomic nervous system. It synthesizes and secretes certain neuro-hormones, often called releasing hormones or hypothalamic hormones, and these in turn stimulate or inhibit the secretion of pituitary hormones. The hypothalamus controls the body temperature, hunger, important aspects of parenting and attachment behaviors, thirst, fatigue, sleep, and circadian rhythms[NCI]. A 2D view of the brain from a side perspective shows the size and the position of the thalamus and hypothalamus (Fig. 2.39). It is located approximately behind the ear.

2.6.4 Connection between the hypothalamus and the visual cortex

The connection between the hypothalamus and the visual cortex is interesting for VIMS studies. As shown in Fig. 2.40, visual motion is analyzed by the visual cortex in region V5, V3a, and V4, and the result of the analysis is relayed into the hypothalamus[142]. Among the brain structures, the hypothalamus and the vision cortex play an important role in VIMS studies because the vision cortex provides an input to the hypothalamus (see Fig. 2.40) and the hypothalamus serves as a medium to transfer the signal from the visual cortex to the vagal nuclei.

2.6.5 Biological mechanism of nausea

Now the question is "why is the hypothalamus important in the current research?" This is because, the Area Postrema (AP)[142], which is a medullary structure in the brain and controls nausea and vomiting, gets an input from the hypothalamus as seen in Fig. 2.41. In addition, the vagus nerve, which is responsible for sending an excitation nervous signal to the abdominal muscles, is directly triggered by the hypothalamus (see Fig. 2.41). The hypothalamus is directly connected to the thalamus. Any mismatching between the stored pattern in the brain and the incoming image during motion can trigger the thalamus and lead to VIMS. Consequently it may lead to headache, changes in saliva, stomach awareness, and finally burping in the stomach. Because of these symptoms, in the experiments (see chapters 5 and 6), Kennedy's questionnaire[98] will be selected to study the effect of the velocity of the visual flow on the level of VIMS induced by presence in a virtual environment. All these signalings in the brain are managed by an organ called the thalamus. The thalamus is a midline symmetrical structure of two halves, within the vertebrate brain, situated between the cerebral cortex and the midbrain. Some of its functions are the relaying of sensory and motor signals to the cerebral cortex and the regulation of consciousness, sleep, and alertness. The thalamus

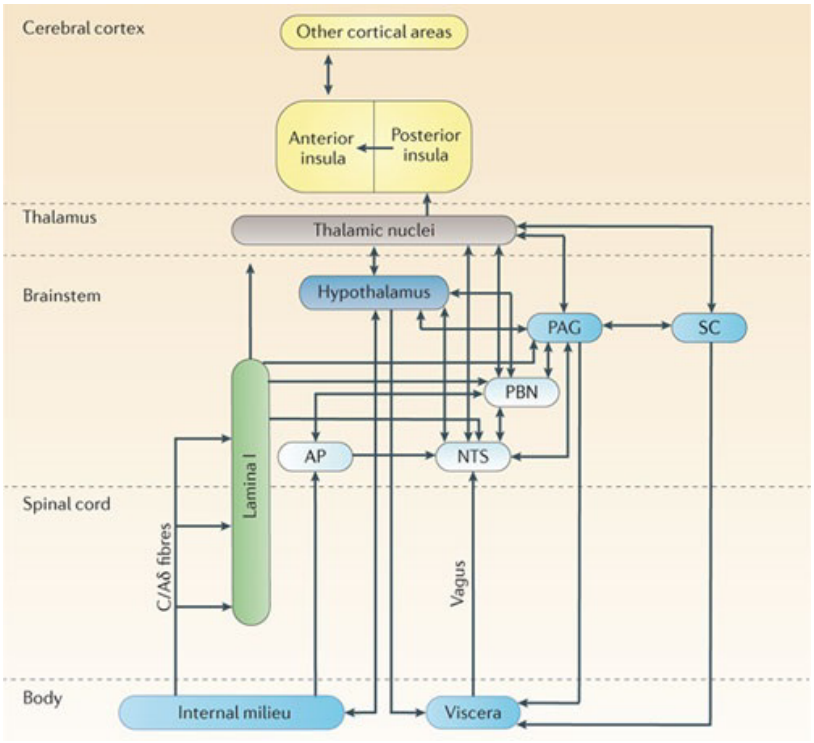


Figure 2.41: Vagus nerve activation and biological mechanism of nausea [142].

is better demonstrated with the surrounding structures in Fig. 2.39. It is surrounded by the hypothalamus and the hippocampus in the bottom and top respectively. A 2D view of the brain from a side perspective shows the size and the position of the thalamus (Fig. 2.39 left). It is located approximately behind the ear.

2.7 VIMS modeling using the Bayesian theory

2.7.1 Motion sickness and visually induced motion sickness

When some species (mammals and especially humans) are exposed to motion, they may get sick. The motion may concern physical self-motion, the motion of a visual scene alone, and their combination. Typically, only synthetic conditions, like being on a moving platform (e.g., in a vehicle, train, boat) or viewing a visual motion (as stationary observers) may induce sickness. The former is called motion sickness (MS) and the later visually induced motion sickness (VIMS) in different literatures[144,17]. Sickness is characterized by physical signs of malaise such as cold sweating, belching, retching, pallor, and decreased gastric tonus, and typical subjective symptoms such as headache, stomach discomfort, feeling of warmth, nausea, and eventually vomiting.

Human experience with MS chronically dates back to the time when US Admiral Nelson reported

his soldiers are suffering from boat sickness[134]. He found out by experience that adaptation and repeated exposure can minimize these adverse effects. Additionally, a human encountering with motion and virtual environments as well as video games demonstrates the general rule that MS adversely affects susceptible individuals[157]. A definite finding during nearly a century concerns the fact that people without functioning organs of balance in the inner ears, so called labyrinthine defectives, never get sick from motion. This has been studied and reported several times so far[76,81,97]. Interestingly, these patients do also not suffer from visual motion, even in the absence of physical self-motion[28,29,86]. Another observation concerns the fact that passive passengers get sick more than people who do control their motion themselves (like pilots)[164,194]. Based on such observations and as mentioned earlier, Reason and Brand suggested a theory on motion sickness, the "sensory conflict theory"[158]. Although the theory was very successful in explaining a lot of observations, it also had some flaws. That is the reason why Bos et al.[17] further detailed this theory focusing on the observation that "*people only get sick when there is an (apparent) change of gravity with respect to their head*". Bos et al. also discussed the most prevalent cause of VIMS in more detail.

A set of symptoms might appear due to a susceptibility to VIMS during or after exposure to certain dynamic visual displays. VIMS[60] can be measured by psychological and physiological methods. The simulator sickness questionnaire (SSQ) is a well-known psychological method for measuring the extent of motion sickness[98]. Although MS and VIMS are different in some characteristics, there are some fundamental similarities. For that reason, the SSQ is used herein for verifying the occurrence of VIMS. The following parameters of an autonomic nervous activity are appropriate for physiological methods: heart rate variability, blood pressure[70], electrogastrography[62], galvanic skin reaction[231], human postural sway[202], and so on (for more detail see chapter 4).

The human postural sway during upright standing is an interesting feature among the above-mentioned features which can provide a good feedback from the user during immersion into a synthetic environment. The human upright standing is preserved by an involuntary physiological adjustment mechanism called the "righting reflex" (body balance function[143]). Sensory signals such as visual, auditory and vestibular as well as proprioceptive signals from the skin, the muscles, and the joints trigger the function[89]. The study of this function helps a better understanding of equilibrium disorders such as cerebellar degenerations and Parkinson disease[143] as well as VIMS characteristics. Different parameters are involved in VIMS, such as circular and linear vection, rod and frame effects, pseudo Coriolis and Purkinje effects, the time delays, foreground/background motion, the field of view, the image velocity, the frame rates, and so on.

2.7.2 VIMS modeling using the Bayesian theory

Reason and Brand[158]'s "sensory conflict" or "sensory rearrangement" theory is the most cited theory on motion sickness. They accordingly stated that "*motion sickness is a self-inflicted maladaptation phenomenon*

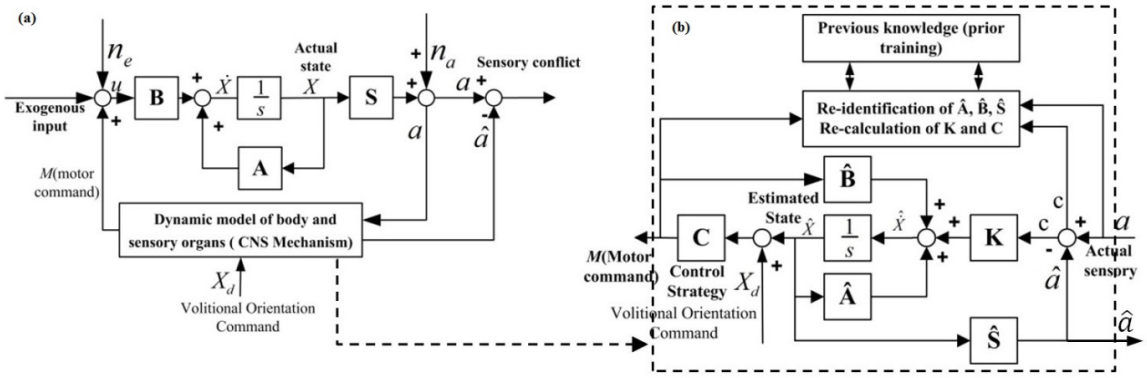


Figure 2.42: a) Modified MS model proposed by Oman [144] for the sensory conflict theory, b) dynamic model of the body, sensory organs and their signaling with the CNS.

which occurs at the onset and cessation of conditions of sensory rearrangement when the pattern of inputs from the vestibular system, other proprioceptors and vision is at variance with the stored patterns derived from recent transactions with the spatial environment".

The conceptual Bayesian model of this cognitive process is shown in Fig. 2.42.a. An "exogenous" (externally generated), versus "indigenous" (self-generated), motion stimulus enters the model and a sensory conflict is created as an output. The coefficients of the state equations ($\mathbf{A}$, $\hat{\mathbf{A}}$) for the body ($\mathbf{B}$, $\hat{\mathbf{B}}$) and the sense organ dynamics ($\mathbf{S}$, $\hat{\mathbf{S}}$) are embodied in this model. Fig. 2.42.b details the central nervous system (CNS) mechanism. Hatted variables of the state estimator with its matrices correspond to neural stores of Reason and Brand's more qualitative model [158]. The observer continuously estimates and predicts the body orientation and corrects the posterior estimation with a Kalman Gain ($\mathbf{K}$) [91]. Then, the CNS re-identifies the observer coefficients ($\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{S}}$), re-calculates $\mathbf{K}$ and updates the control strategy ($\mathbf{C}$) for the next iteration (the CNS repeats this chain infinitely and non-stop). The sensory conflict vector $\mathbf{c}$ (Fig. 2.42.b), divergence vector, is obtained by subtracting the actual sensory from the expected input $\hat{\mathbf{S}}\hat{\mathbf{X}}$. When the image velocity goes above certain limit, the conflict vector grows because vestibular data is missing. Consequently, the resulting vector $\mathbf{Kc}$ grows which in turn leads to more sickness.

Sensory conflict can occur due to a difference between the vestibular-ocular, intra-ocular or intra-vestibular and finally ocular-vestibular with a proprioceptive pattern. The proprioceptive pattern is defined as a pattern derived from a recent transaction with a real environment. The term "non-vestibular" is alternatively used instead of "proprioceptive pattern" in scientific publications [144]. The proprioceptive pattern is adjusted and the cortex is trained in the real environment by observing events, feeling the physical properties of objects, and interacting with the real environment. The objects and events in a virtual environment are quite artificial with only few physical properties. Due to the lack of physical tangible properties such as gravity, force feedback, temperature, and so on, the difference between the real and virtual environments

is unavoidable even though the virtual environment is processed and visualized by very fast and high-end graphic processors. The visual- non-vestibular difference perceived by immersion into the virtual environment increases the amplitude of the divergence vector. A larger divergence vector means a larger sensory conflict vector consequently, which leads to strong VIMS. VIMS in a synthetic environment mainly emerges due to oculo-vestibular and oculo- non-vestibular conflicts. However, there are some cases that VIMS appears due to intra-sensory conflicts[179].

2.8 Conclusion

In this chapter, we reviewed and proposed mathematical models for the vestibular dynamics as well as human vision. These models allow us to better understand how the brain works when humans move and see. We used these models to explain motion sickness or visually induced motion sickness that occurs when moving in a virtual environment. In the next chapter, we will discuss the technical part of the thesis, that is, the technical tools used to set up our virtual reality platform and prepare the experiments for answering our research questions.

3

Interaction techniques and implementation

3.1 Introduction

This chapter is organized to give an insight into the development steps as follows: a review over different navigation and manipulation interfaces will be presented in section 3.2. Sections 3.3 and 3.4 will be dedicated to a review on different interaction interfaces. We will use this interfaces both to complete our studies and assess the performance of the interfaces subjectively and objectively later in chapters 5 and 6. Their design and mathematical definition will be explained in section 3.5 and 3.6. We have developed a software platform to code any navigation and interaction in the Virtual Environments (VEs) easily and fast. Different components, the hardware and data communication infrastructure, the configuration and the display system will be explained in section 3.7. The system has two operation modes: simulation and real scale display. The configuration for each mode will be detailed and the development language and its instructions will be summarized in this section. There are different versions of real-scale 3D displays. We did the navigation/manipulation experiments in a CAVE system which is widely used in VR studies. The hardware compartments of our CAVE system will be briefly introduced finally.

3.2 Interaction in scale-one 3D VEs used for modeling and simulation

A virtual environment (VE) is an interactive computer image that employs a wide variety of display systems to convey information to the users. The users of a VE are immersed in a synthetic space of virtual images enhanced by special processing and nonvisual display modalities[137]. Three-dimensional virtual environments are used in fields as diverse as manufacturing[35,206], medicine[207,178], construction[225,160,112], psychotherapy[30], design[147], and education[217,216]. They also play an important role in the investigation of spatial processes, such as examining directional knowledge[82] or assessing spatial abilities[24], allowing researchers to design realistic experimental settings and flexibly record user's behaviors[88].

Although the diversity of VE applications makes it necessary to design VE interfaces that support domain-dependent needs[37], some tasks, such as navigation, are common to all VE applications and are essential even when they are not the main objective of a user in a VE. Navigation can be defined as the process whereby people determine where they are, where everything else is and how to get to particular objects or places[49]. Navigation is the aggregate task of way-finding and motion. Way-finding is the cognitive element of navigation. It does involve not only movement of any kind but the tactical and strategic parts that guide movement as well[167]. Navigation of users in VEs has been investigated to a large degree[166,181,213]. To navigate successfully, people must plan their movements using spatial knowledge, such as visual cues, the position of the object and their color. Users gain spatial knowledge about the environment and store them as a mental map. However, accurate spatial knowledge of VEs typically is developed very slowly after long periods of navigation or study, and users may not always be willing to spend this time[208]. Thus, if the navigation support provided by VEs' user interfaces is insufficient or inappropriate, people become disoriented and get lost.

Navigation problems are even more serious in large-scale VEs, where there is no vantage point from which the entire world can be seen in detail, and the amount of detail that can be seen (e.g., from a bird's eye view) is drastically reduced by occlusion. To learn the structure of an environment, users are thus forced to navigate extensively and to integrate information derived from different points of view. In scaled desktop VEs, where a mouse and keyboard are usually the main input devices and the virtual world is experienced through a computer screen, navigation is further complicated by the absence of many sensorial stimuli (e.g., vestibular and proprioceptive) that are commonly exploited by users in the physical world.

A large number of recent research focuses on how to face navigation issues in VEs. In particular, a lot of effort has been aimed at developing navigation aids that help the user to explore and learn the environment around, preventing disorientation and simplifying navigation. However, only limited attention has been devoted to compare different navigation aids, and the relations between the effectiveness of navigation aids and different levels of user experience in navigating VEs have been left largely unexplored.

In a recent study on the effect of age on the use of VEs, Sayers[174] took into consideration the impact of

computer experience, Internet experience, and 3D-application experience in tasks where users were asked to search for specific objects in a 3D store, with or without an overview map of the VE. The study showed that, when the overview map was used, previous Internet experience did matter with respect to the time spent to perform the tasks, but not with respect to the number of interaction steps to move around in the VE. The experience in using VEs, instead, was not found to have an effect on performance.

In general, having an effort, such as walking, talking, and actively involving in some physical movement, during immersion in VEs helps better spatial perception and improves the navigation and manipulation parameters. For instance, Giordano et al.[50] showed that exploring a VE in a walking condition allowed better spatial learning of the environment with respect to a fly condition, particularly for complex 3D VEs. Nevertheless, with practice, the performance in the fly condition improved whereas the performance in the walk condition remained at its initial maximum. We can benefit all those domains where it is important to provide simple yet effective indications to support user navigation in a VE, regardless of its scale, while limiting visual obstruction and keeping as much as possible the feeling of immersion which is typical of VEs. Examples include training, videogames but also new car or pedestrian navigation systems based on 3D environments.

For example, 3D arrows have been recently employed in an augmented reality environment to inform drivers about dangerous situations around their cars[208]. At the same time, studying experienced versus inexperienced users makes it possible to determine the most appropriate solution for different situations, e.g., allowing navigation support to adapt a user's navigation skill improvement.

To effectively navigate in an environment, users rely on three distinct types of spatial knowledge: landmark, route, and survey[33,169]. In unfamiliar environments, people first learn about landmarks, which are distinctive environmental features (such as specific buildings, supermarkets, city squares, bridges, etc.) functioning as reference points during navigation[33,67]. Landmarks act as visual anchors that identify different regions of space[215] or provide an organizational structure that facilitates the location of points that are nearby[135]. Route knowledge is usually developed from a first person perspective and makes it possible to connect different landmarks in a sequence, creating paths between locations in the environment[33,154,67]. It allows a user to travel to destinations through known routes, but does not allow the user to take alternative unfamiliar routes. Finally, survey knowledge is developed from a third person perspective (e.g., through maps) or by extensive traveling in an environment and describes the relationships among locations allowing users to assess where certain objects are located with respect to others in the environment and to recognize alternative routes[33,154].

It is now generally accepted that people simultaneously develop landmark, route and survey knowledge[168,218], and form mental images of the places, known as cognitive maps. Research also suggests that vestibular and proprioceptive cues are important in developing spatial knowledge[229], although some researchers found little effect of proprioceptive information on spatial learning of environments[174]. This

contributes to the difficulties in navigating in VEs, as they are often limited to the motor-environment interaction afforded by conventional keyboards and/or joysticks.

Navigation difficulties in VEs originate from different factors. Obvious sources of navigational problems are represented by not knowing the structure and layout of a particular VE as well as a lack of familiarity with VEs in general[213]. Lack of intuitiveness of traditional navigation methods, such as navigating by mouse movement, a joystick or a keyboard causes additional difficulties[30,183].

Additional factors such as a lack of support for speed control, navigation mode (e.g., walking vs. flying), lack of landmarks and restricted field of view create navigation problems[83]. In general, there are two main ways to provide navigation support in VEs: building VEs that follow navigability guidelines and providing navigation aids to help the user.

Approaches based on structuring a VE to facilitate navigation often derive from other fields which have already faced the problem in the physical world, e.g., Lynch's work on the classification of navigational contents of cities[114]. Some authors[135] derive the requirements for spatial design in VEs from architectural theories, while others[167] discuss methods to organize the space for navigability, inspired by previous research in fields such as urban planning. Extensive work is available on the design and placement of landmarks[215]. The important role played by landmarks when acquiring route knowledge in a network of paths is investigated by Jansen-Osmann and Wiedenbauer[83], who found that landmarks do indeed aid orientation in way-finding, and a route with landmarks is learned faster than one without landmarks.

3.3 Navigation principles and interfaces

3.3.1 Navigation principle

Navigation devices and approaches are application dependent, however its principle and theoretical fundament in VEs are approximately similar. In many domains, inappropriate solutions for a beginner might be very efficient and comfortable for an experienced and vice versa. For instance, the graphical interface provided by Microsoft in the Windows (XP and 7) platform is very comfortable and user-friendly for any user; however software engineers prefer to work with Linux which is not really easy in general. As another famous example, it is well-known that WIMP (Windows, Icons, Menus, and Pointers) interfaces are suitable for novices but experienced users may prefer quicker alternatives such as keyboard shortcuts or even command line interfaces. Similarly, navigation aids which are appropriate for experienced users may not provide a suitable level of support for inexperienced users. Contrarily, the solutions that may improve the navigation performance of inexperienced users may not benefit experienced users beyond a certain degree. Despite the diversity of application, some aspects of navigation interfaces are common in all the applications. For example, all the navigation can be defined as a set of continuous movements. Each movement has a starting and stop point. We can move from the starting points off with acceleration and continue

Principle components of a navigation in an environment			
Steps	*Description of the sub-task*	*Detail of the sub-task*	
1	Start		
2	Indicate target position	position	Discrete target
			One-time route
			Continuous
3	Indicate orientation		
4	Move	velocity	
		acceleration	
5	Stop		

Table 3.1: Basic components of a navigation.

Direction/target point parameter		
Choice	*Methods*	*Description*
1	Gaze-direct steering	
2	Pointing by gesture (including props)	
3	Discrete	List (e.g., menu)
		Environmental or directional targets (objects in the virtual world)
4	2D pointing	

Table 3.2: Direction/target selection methods.

our path with a constant velocity. The description of the navigation components and their associated sub-tasks are detailed in Table 3.1. As shown in Table 3.1, each navigation task begins with "start" and ends to "stop" commands. The rest consists of a movement toward a target point along a pre-specified direction.

The direction of the movement, the target point and the velocity/acceleration are the key parameters of navigation during traveling from a given point A to destination B. Each of these tasks can be selected in different ways. For example, only the target and direction can be selected by four methods as shown in Table 3.2. Usually, it is very preferable to select the target when walking in virtual reality and change it during navigation. But in some applications, the user navigates to manipulate an object. The target point must be specified in advance if the user is not close enough to the target object.

It is very important to know how we can approach to the target point. The instant variation of some

Velocity/acceleration parameter		
Choice	*Methods*	*Description*
1	Constant velocity and acceleration	
2	Gesture-based (including props)	
3	Explicitly assigned	Discrete (1 or N)
		Continuous range
4	User of environment scaling	
5	Automatic or adaptive approaches	

Table 3.3: Different methods of acceleration selection.

parameters such as the velocity, the acceleration, and the direction influence the user adversely which leads to general discomfort and nausea if the effect is quite severe. For example, very fast motion or sudden variation in the velocity gives rise to VIMS as detailed in chapter 6 (see section 6.4). Table 3.3 shows different ways of acceleration and velocity selection. Velocity/acceleration can be set constant or adjusted by gesture such as walking and head movement.

Not only the target and traveling parameters but also the navigation device is important for natural and appropriate interaction. For example, stop and start buttons are used in some devices like the joystick, while in gesture-based applications, only the human gesture controls the movement, albeit more constrains can be imposed to the second traveling task. The movement can start and stop by pushing a button from the navigation device and in between the motion continues constantly and non-stop. It can be started and stopped automatically by analyzing a signal and checking some criteria. If the condition and criteria are met, the movement will start or stop. This technique is used more in gesture-based interfaces. Traveling sub-tasks need to be defined first in order to use navigation interfaces effectively everywhere in VEs (Table 3.4). Let's assume traveling along a random path from point A to F while passing through randomly located points B, C and D between these two points. The very basic navigation task is to go from point A to B (short travel). If this task is defined appropriately, the travel from A to F will be completed by repeating the short travel four times. Moreover, this navigation strategy can be employed everywhere with any kind of navigation tools. That is why defining these details has a great deal of importance and plays a critical role in designing any navigation interface. Table 3.1 shows that each travel can be split into five steps. The points located along the path, between the start and stop points, can be assigned in different ways as shown in Table 3.4.

The user position needs to be calculated or measured during navigation because the display system uses this information to update the display area by projecting a new part of the 3D scene (3D mock-up). Different techniques have been proposed for user tracking in VEs in the literature. Since these techniques

Path planning by indicator position along the path		
	Methods	*Description*
1	Discrete target	Marker selection from the environment
		Position 3D cursor
		Automatic position dependent selection
2	One-time route	Set series of markers
		Specify by curvature and distance
		Itinerary planning
3	Continuous	Gaze-directed position
		Physical steering props
		2D pointing via a virtual control interface

Table 3.4: Different strategies for path planning and point selection.

are more application dependent it is not easy to collect all of them under limited categories. The most common form of the tracking system is the ray-based tracking as shown in Fig. 3.5.

The most predominant tracking system is an infrared-based system which uses a triangulation method to calculate the user's position (e.g., AR-Tracker system) and is arranged in a configuration of three or four Z-depth IR cameras in combination with laser projectors. The position (x, y, z) and the orientation (p_0, p_1, p_2, p_3) are calculated by applying the interpolation and triangulation algorithm to the coordinates of a set of laser-sensitive balls attached to 3D glasses. The results are reported in Cartesian coordinates and the quaternion system. A setup of a tracker system with four infrared cameras is shown in Fig. 3.1.

3.3.2 Examples of navigation interfaces

Navigation interfaces in VEs are a great issue for a long time in computer science, VR technology and HCI[183] and quite a number of solutions have been proposed. These solutions can be divided into two main categories:

1. Physical locomotion (ray based tracking, walking, "walking in place"[183], "redirected walking" (RW)[135], device actuated by walking and cycling, etc.) as shown in Fig. 3.2;

2. Steering techniques (gaze-directed steering, pointing, torso-directed steering, and camera in hand, virtual motion controller and semi-automated steering, etc.) as shown in Fig. 3.3.a. They are referred to as human posture and gesture based navigation interface (the best example of this type of navigation interface is the Kinect interface) (Fig. 3.3.b). These techniques apply human gesture

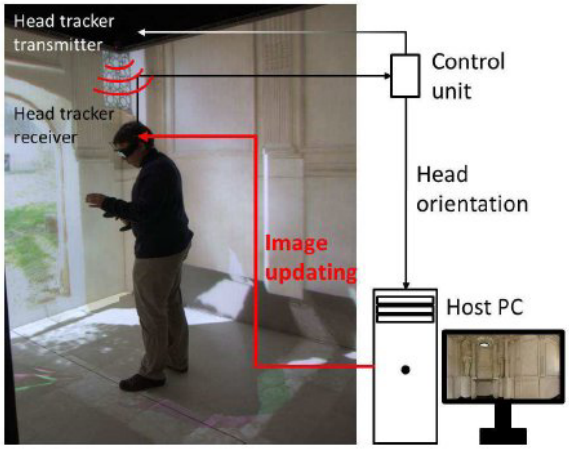


Figure 3.1: Example of laser based tracking system.

and posture to provide a corresponding command to navigate in VEs. In fact all the commands that one needs to inform the control system to update the 3D scene are extracted from the user's body motion. See an example of these body motion analysis in Fig. 3.4. As shown, the body image grabbed by IR cameras is processed by a segmentation algorithm and then cleaned up by morphological operators. The morphological centroid is calculated for the resulting image and the position of the maxima on the silhouette contour are calculated. The centroid and the maximum points are connected to make the skeleton of the body out of it. The extracted skeleton, for instance using a Kinect sensor, is analyzed by pattern matching and a target gesture is assigned to the gesture in the training phase, as seen in Fig. 3.5. In the evaluation phase, for an incoming IR image containing a gesture, the gesture is detected as shown in Fig. 3.6.

An overview of these technologies can be found in [88]. Several heuristic solutions have been developed to enable omnidirectional walking through scale-one VEs, however all the possible omnidirectional walking developments require strict constrains and dealing with security concerns [139,191]. On the other hand, a user really needs some constrain-free omnidirectional navigation interfaces with lower cognitive issues. The most natural way to navigate inside the real world is by foot. However, it is not possible to walk inside a scale-one 3D VE in the same way as in reality because: 1) the user is limited to the projection space extended up to maximum few meters in length; 2) only part of the 3D VE is visible, when a very huge model is displayed. Therefore, either we need to develop an omnidirectional treadmill to use the same idea [50] or the walking concept needs to be changed. The first approach has some physical constrains and need lots of security issues to be taken care. Most of the papers followed the second approach.

Our natural walking in the real environment is a relative movement. It is the best starting point because immediately an idea strikes to the mind that: "*instead of going towards a target point, the target point can*



Figure 3.2: Examples of navigation interfaces based on physical locomotion (Sarcos)[78].

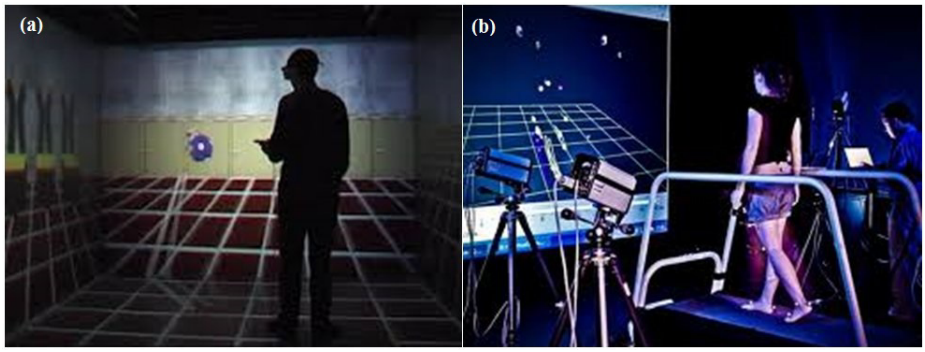


Figure 3.3: a) Task definition of the steering technique, b) navigation in VEs by gesture-based steering technique.

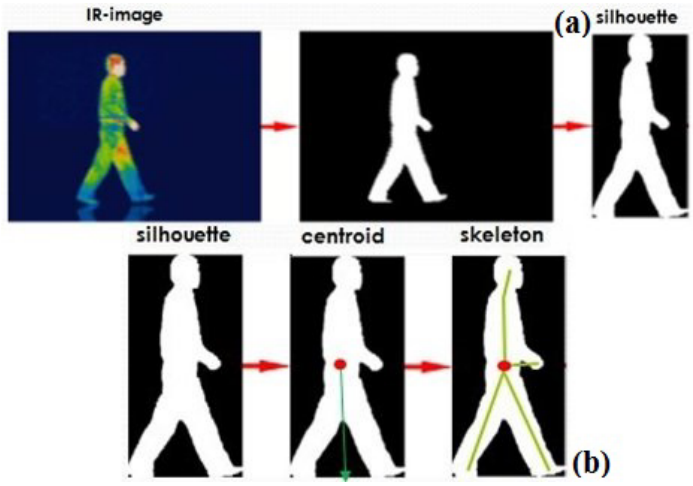


Figure 3.4: IR image analysis for skeleton detection: a) silhouette detection, b) skeleton definition.

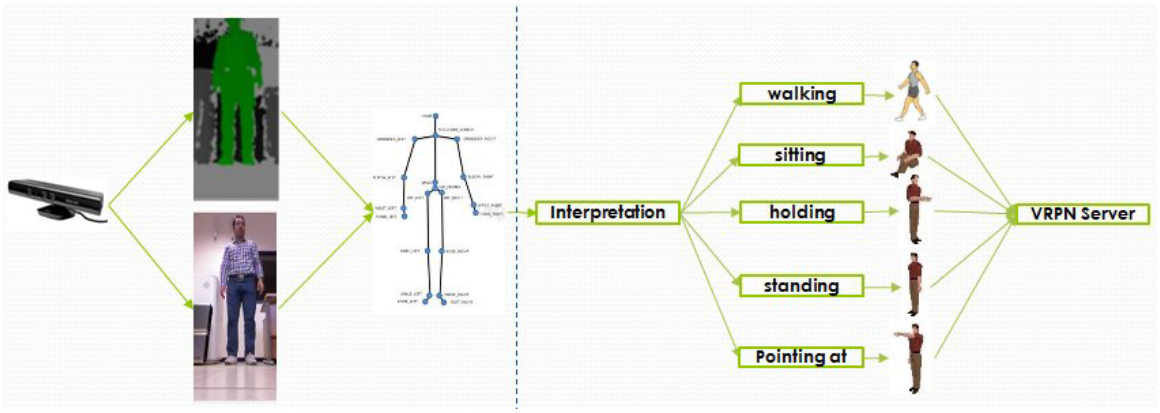


Figure 3.5: Example of body gesture analysis for five actions (walking, sitting, holding, standing, pointing at).

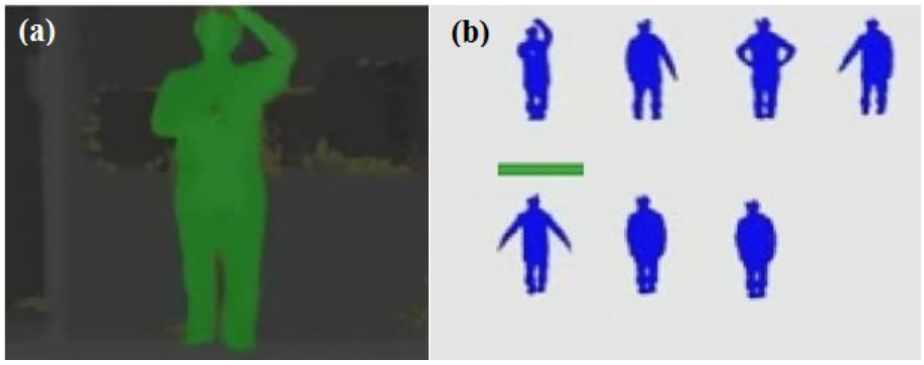


Figure 3.6: Output of the gesture detection processing unit: a) input IR image, b) detected gesture.

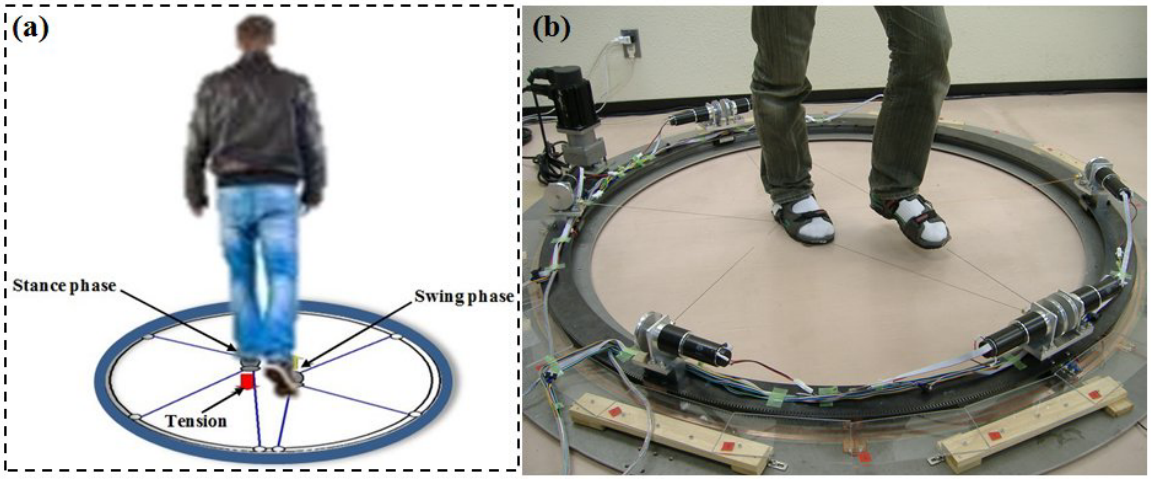


Figure 3.7: WIP interface designed by swing wires for sensory incorporation: (a) concept, (b) real WIP interface [79].

be drawn to us (by pulling the virtual scene)". This will motivate us to construct a navigation interface which moves the scene and all the objects attached to the scene back and forth by walking in place, moving the left and right feet in place consecutively. This navigation interface is called "Walking in Place" (WIP) in the literature. In this way, the user feels that he is going ahead inside VEs in the selected direction very smoothly.

Research on VEs has provided converging evidence that being placed in a virtual environment can lead to motion sickness [96], as explained in chapter 2. For example, Kennedy et al. [96] have shown that following exposure to virtual environments and simulators, up to 30% of exposures result in reported nausea, up to 40% of exposures result in reported eyestrain, as well as the experience of numerous other symptoms of motion sickness including drowsiness, salivation, sweating, headache, dizziness/vertigo and loss of postural stability, although the prevalence of motion sickness in virtual environments appears greater than in flight simulators [94,95,38].

Recent work mainly tries to incorporate more sensory information into the design of WIP interfaces. This is because, missing sensory will convey an incomplete pattern to the cortex which in turn, leads to a sensory conflict and sever VIMS in practice (divergence between afferents and proprioception). Different versions of the WIP interface have been proposed in the literature [50,36] regarding this fact. Fig. 3.7 shows one of the WIP interfaces designed based on incorporating more sensory information. This incorporation does have a profound vestibular and rather hypothetical basis. Nevertheless, in our view, a lack of sensory information largely contributes in a VIMS symptoms provocation in a VE. Furthermore, different aspects involved in VIMS, such as circular and linear vection, rod and frame effects, Purkinje effects, time delays, foreground/ background motion, field of view, image velocity, frame rates, and the display parameters must appropriately be considered in the interface design. The more sensory is incorporated in the interface



Figure 3.8: Cyber-walk on an omnidirectional treadmill [50].

design, the lower is VIMS because a more complete pattern will be relayed to the cortex. One of the most completed WIP interface was proposed by Giordano et al.[50] as shown in Fig. 3.8.

However, walking is quite interesting for the researchers; there are some other devices that can be used to handle a navigation task. Recent work proposed to use technologies provided by game manufacturers such as the so-called Fly-stick, Gamepad, Wii board, and Wiimote or combined with other methods [239,228]. Others designed special shoes including actuators and sensors and providing haptic and audio-haptic feedback[141]. Although these technologies seem promising for navigating in a scale-one 3D model, they can be intrusive for the users, thus they do not allow long navigation tasks.

3.4 Manipulation principles and interfaces

3.4.1 Manipulation principles

Object selection and positioning are among the most fundamental interactions between humans and environments, whether it is a "desktop" of 2D direct manipulation interfaces, a 3D virtual environment, or the physical world[18,219]. In the literature, a manipulation refers to as any act of handling physical/virtual objects with one or two hands or interaction devices. Spatial rigid manipulation is basically a manipulation that preserves the shape of the object. Manipulation consists of three fundamental sub-tasks as shown in Table 3.5: object selection, main manipulation sub-task, and object positioning. Target object selection is the first sub-task performed by the user in each manipulation. The main manipulation sub-task includes

Basic manipulation sub-task		
	Sub-task	*Definition*
1	Selection	Identifying an object
2	Main manipulation sub-task	Moving, rotating and changing the property
3	Positioning	Placing an object in the target position

Table 3.5: Sub-tasks associated with object manipulation.

a wide range of actions such as rotation, translation, deformation, scaling, and so on and so forth. The last sub-task is positioning in which the user leaves the object in the target position after applying different types of manipulation.

An object can be manipulated either by user gesture or interaction devices. If interaction devices are used for object manipulation, some characteristics of the input device are important such as:

1. Degrees of Freedom (DOF), the more the better;
2. Returning feedback (force, pressure, temperature) from the target object;
3. Device placement and form factor (shape of the input devices and the place we put the devices, e.g., finger, hand, ...).

An example of interaction method based on hand gesture is shown in Figs. 3.9 and 3.10. As shown in Fig. 3.10, the first step of gesture analysis is the skeleton detection. Then, the hand gesture will be assigned by the finger configuration. For instance, if the finger configuration is as the first image (Fig. 3.9 top) then code 001 is generated. At the same time, the position of the hand and of each finger is coded as analog values. The latter is useful when two hands are involved in the object manipulation. Different steps of a hand skeleton detection algorithm is shown in Fig. 3.9. The algorithm simply uses RGB and IR images of the hand to segment the hand against the background. Small regions of segmentation can be eliminated using one step of a morphological operation such as "Opening" or "Closing". The points with a maximum distance from the centroid in the resulting image are the finger tips. Having the finger tips and the hand centroid, the skeleton can be drawn.

Prior research on manipulation in VEs relates primarily to the assessment of user performance as a function of input and display devices and their properties. For example, a pioneering study by Ware [219] demonstrated the applicability and ease of use of a 3D input device for a six degree of freedom (6DOF) placement task. A study by Zhai and Milgram [236] compares isometric versus isotonic input devices in various conditions of spatial manipulation. Studies of stereoscopic versus monoscopic display devices suggest that stereoscopy improves user manipulation performance [192,14]. The effects of the system parameters (such as lag and frame rate) in the manipulation task on user performance have also been extensively studied [221,115].

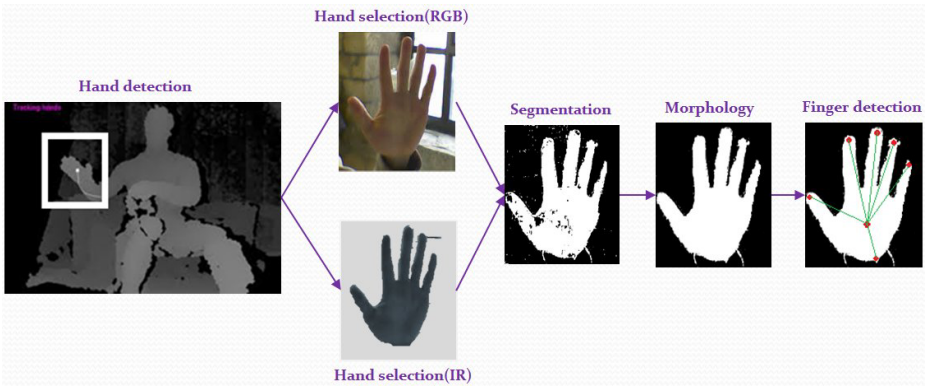


Figure 3.9: Hand tracking by skeleton detection.

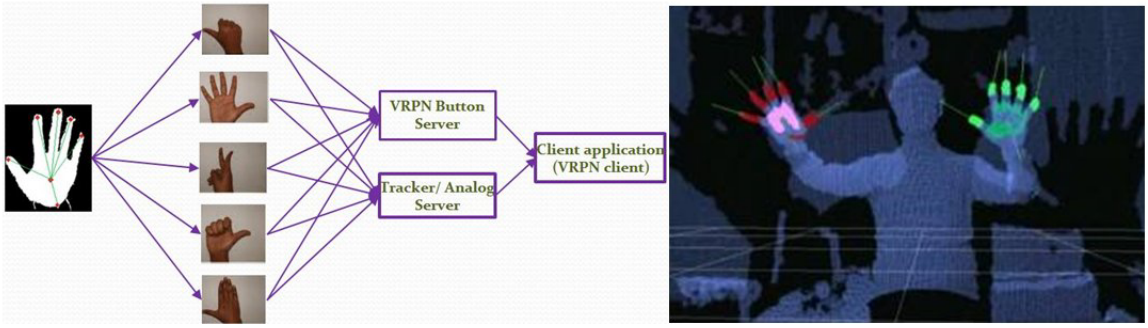


Figure 3.10: Hand gesture analysis and coding.

Investigation of the human factors related to input and output devices has a considerable value; however, the lack of systematic research on interaction techniques, which map the user's actions captured by input devices into resulting actions in the VE[131], may significantly limit their appropriate use in VE design. Interaction techniques essentially define the "look and feel" of VEs. A wide variety of techniques can be implemented using the same input devices, and quite a few techniques for spatial manipulation have been demonstrated. Still, there have been few attempts to formally evaluate them, to assess and compare their functional capabilities under various circumstances.

A number of surveys have summarized and classified various approaches for designing techniques for spatial inputs and identified problems and possible solutions[131,63]. Zhai et al.[235] evaluated the application of volumetric semi-transparent cursors ("silk" cursors) in a 3D target acquisition task, and reported a user performance improvement as compared with traditional cursors. A study by Hinckley et al.[64] evaluated and compared several spatial rotational techniques. In a study by Mine et al.[132], automated world scaling techniques were evaluated. More relevant here is the pioneering usability study reported by Bowman and Hodges[18] that evaluated several VE techniques for manipulation at a distance. Although no quantitative data were collected, this study provided useful preliminary observations of the techniques.

Starting with early techniques that simply mapped the position and orientation of the user's hand onto the position and orientation of manipulated objects[220,11], the field has been expanding with more sophisticated techniques such as flashlight[107], aperture[45], Go-Go[18,153] (Fig. 3.11), World-In-Miniature[196], image plane[149], scaled-world grab[220] and many others. We will present some of them later. This variety of techniques, however, is also a source of difficulty. How do all these techniques relate to each other? Which interaction techniques should be chosen for particular task conditions? Which among the parameters of interaction techniques, tasks, and environments should be considered to design efficient VE interfaces? These questions persist and merit careful scrutiny by researchers and practitioners.

A straightforward evaluation and comparison of manipulation techniques is difficult. There are a multitude of different techniques; even for the same technique, the performance varies depending on particular implementation; and studies of a particular technique implementation may not be readily generalized to other implementations of the same technique, thus limiting their external validity.

On the other hand, many techniques apparently relate to each other and share many common properties. For example, there are more similarities between ray-casting and flashlight techniques than between ray-casting (see Fig. 3.12.c) and techniques that use non-linear mappings (as in Go-Go[153]). While an evaluation of ray-casting might provide insight into techniques similar to ray-casting, such as flashlight, it probably would not help in understanding techniques like Go-Go. The taxonomy of techniques, classifying them according to their common properties, can be instrumental in understanding the relations between the techniques and directing their design and experimental evaluation.

The analysis of current VE manipulation techniques suggests that most of them are based on a few



Figure 3.11: The Go-Go method [153]: while the real hand (Rr) is within the distance D ($Rr < D$), the mapping is linear and the movements of the virtual hand correspond to the movements of the real one. When the user extends the hand further than D ($Rr > D$), the mapping becomes non-linear and the virtual arm "grows".

interaction metaphors. Each of these basic metaphors forms the fundamental mental model of a technique - a perceptual manifestation of what users can do, how they can do it (affordances), and what they cannot do (constraints) when using the technique [11,42].

Particular techniques are essentially implementations of basic metaphors, often extending them in order to overcome some of the metaphor's shortcomings and constraints. For example, the flashlight technique enhances ray-casting by using a spotlight to ease selection of small objects [107] (see below). These improvements often result in new constraints; for example, with the flashlight technique, an ambiguity might occur if several small objects fall into the spotlight.

As presented in Table 3.5, each object manipulation task has three sub-tasks, i.e., "selection", "positioning" and "rotation". Table 3.6 explains in brief only object "selection" techniques. The effect of the distance and object size on user performance has been extensively studied, prior studies relate either to real world target acquisition [135] or non-immersive object manipulation with 2D and 3D user interfaces [236,14]. In both cases, manipulation occurs only within the natural reaching distance, as opposed to immersive VEs in which users often need to access and manipulate objects located both nearby and far outside the normal area of reach [132].

3.4.2 Examples of object manipulation interfaces

The task of selecting and manipulating objects located far away is often referred as "action at-a-distance" [63]. The development of effective means of object manipulation across a wide range of distances has been rec-

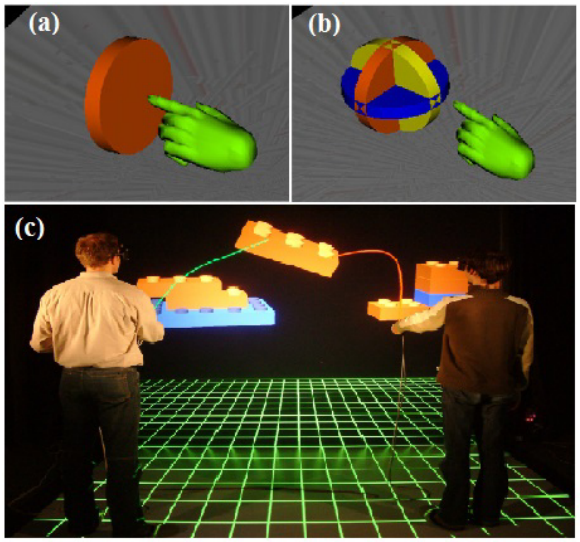


Figure 3.12: a), b) Object selection with a virtual pointer [152], c) ray-casting technique.

	Techniques	*Description*	*Detail*
1	Indication of an object	Occlusion	List
		Pointing	Voice selection
		Touching of an object →	Automatic
		Indirect selection	Iconic object
2	Confirmation of selection	Event	
		Gesture	
		Voice command	
		Not explicit command	
3	Feedback	Text, symbolic	
		Aural	
		Visual	
		Force, tactile	

Table 3.6: Decomposition of an object selection sub-task.

ognized as an important problem in virtual interface research and development [226]. The whole variety was divided into exocentric (Fig. 3.13) and egocentric techniques. Originating in studies of cockpit displays [80], these terms are used now to distinguish between two fundamental frames of reference for user interaction with VEs. With the exocentric interaction, also known as the God's eye viewpoint, users interact with VEs from the outside (the outside-in world referenced display [80]). An example is the World-In-Miniature technique, which allows manipulation of objects by interacting with their representation in a miniature model of the environment held by the user [196]. Another example is the automatic scaling technique [220], which scales down the world so the user can access the objects located at a distance.

With the egocentric interaction, which is the most common for immersive VEs, the user is interacting from inside the environment - i.e., the VE embeds the user [80]. Currently there are few basic metaphors for egocentric manipulation that will be introduced below [63,219,149].

Another technique is called the virtual hand. Users can grab and position objects by "touching" and "picking" them with a virtual representation of their real hand. The major design factor that defines a particular technique is the choice of input devices and mappings between the real hand's position and orientation and the virtual hand's position and orientation. For example a "classical" virtual hand technique provides a one-to-one mapping between the real and virtual hands.

The Go-Go technique is another virtual hand technique that uses a non-linear mapping function to translate the measured distance to the real hand into the controlled distance to the virtual one [153] (Fig. 3.11). This allows for significant expansion of the user's area of reach.

The virtual pointer is an object manipulation interface with which the user selects and manipulates objects by pointing at them. When the vector emanating from the virtual pointer intersects with an object, it can be picked and manipulated [63]. The major design aspects that distinguish techniques based on this interface are the definition of the virtual pointer direction, the shape of the pointer (selection volume), and the methods of disambiguating the object the user wants to select. In the simplest case, the direction of the virtual pointer is defined by the orientation of the virtual hand, the pointer is a "laser ray," and no disambiguation is provided [152] (Fig. 3.12.a,b). Some techniques define the direction of the virtual pointer by using two points: the position of the user's dominant eye and the location of the tracker manipulated by the user [45,149]. Volumetric pointers are also used to ease selection of objects [45].

The direction of the virtual pointer is defined by the position and orientation of the virtual hand (Fig. 3.12.a). The working volume of the technique is an invisible infinite ray emanating from the user's hand; a short segment of the ray is attached to the hand to indicate the direction of pointing. To select an object, the user points at it and presses a button on a button device. When visual feedback is applied, the color of an object changes when the ray intersects with it. Repositioning objects has been reported as difficult using virtual pointer techniques [219,220]. The classical virtual pointer implementation does not permit any change in the ray length; therefore, moving the object from a close to a far distance or vice versa can

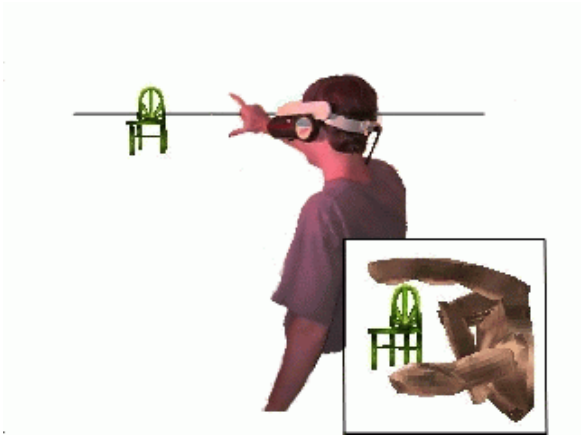


Figure 3.13: Egocentric object manipulation using the Head Crusher technique [149].

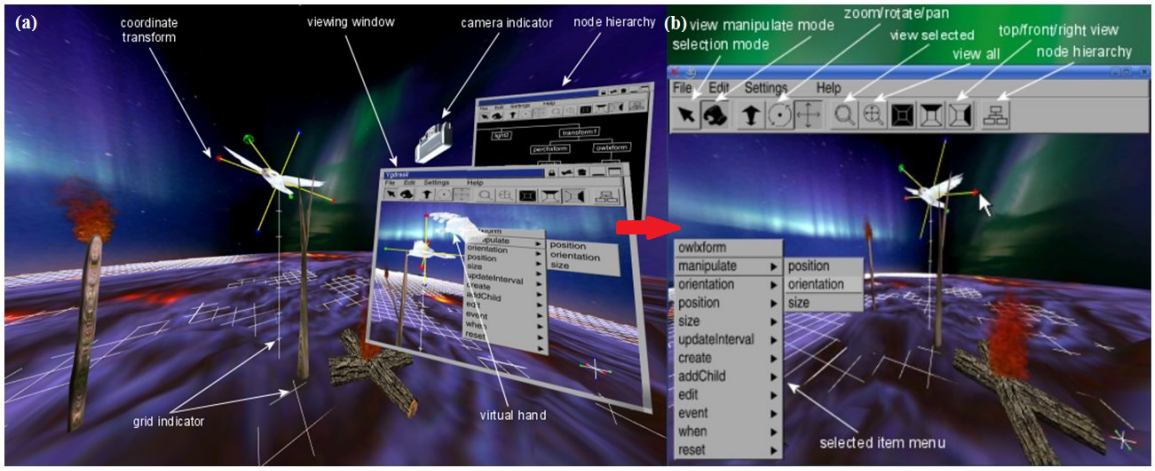


Figure 3.14: Virtual image plane: a) inside a virtual environment, b) detail of the virtual image plane [61].

be accomplished only through iterative object picking, moving, releasing, re-picking and so on [220]. This method is obviously very inefficient.

Another useful manipulation interface is image plane. The image-plane family of techniques simplifies the object selection task by requiring the user to control only 2 DOF. With this technique, the user selects and manipulates 3D objects by touching and manipulating their 2D projections on a virtual image plane located in front of the user (Figs. 3.13 and 3.14).

The spotlight and flashlight techniques were developed to provide a "soft" selection that does not require precision and accuracy of pointing to virtual objects with a ray. The techniques imitate pointing at objects with a flashlight, which can illuminate an object even when the user does not point precisely at it. The user can interactively control the spread of the angle of the selection volume simply by bringing the

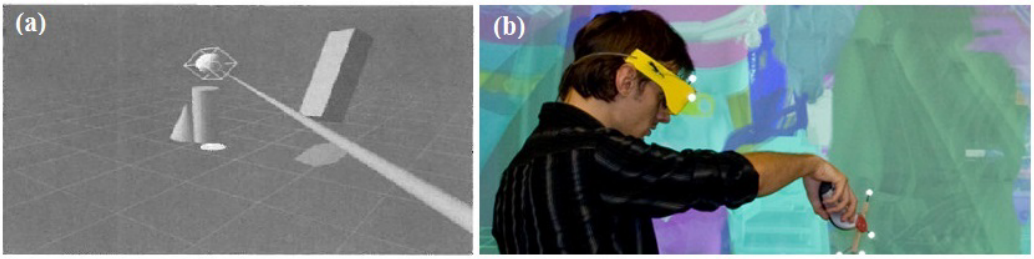


Figure 3.15: a) Aperture technique and b) selection-orientation planes.

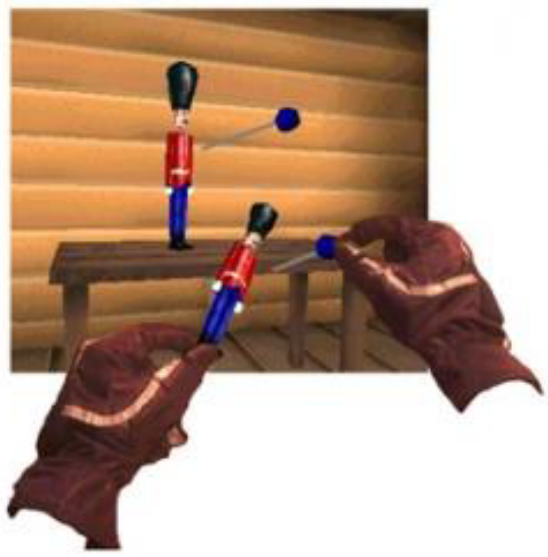


Figure 3.16: Voodoo Dolls object manipulation [150].

hand sensor closer or moving it farther away. The aperture technique further simplifies the selection of virtual objects by using the orientation of the pointer around a central axis as an additional disambiguation metric (Fig. 3.15).

The Voodoo Dolls (Fig. 3.16)[150] interface is based on several key ideas. First, it proposes to manipulate virtual objects indirectly, using temporary, miniature, handheld copies of objects called dolls. Similar to the word in miniature (WIM), the user manipulates these dolls instead of the virtual objects, which creates the dolls representing the target objects and places them in the user's hand.

So far we have suggested a taxonomy that identifies only the basics, which can be further subdivided to reflect particular aspects of each technique. Also, techniques based on different metaphors can be combined together to form new manipulation techniques. For example, techniques that combined virtual pointer and virtual hand metaphors have been reported[18].

Positioning could be performed using iterative movements, i.e., the subjects could pick, move, and re-

lease the object several times. Each time the object is released, the error of positioning is calculated. The researcher can control the required accuracy of positioning by specifying maximal vertical and horizontal displacements for task conditions. For example, a 0% displacement means that the target object must be aligned on top of the terminal object without any positional error. When the error of positioning falls below a specified threshold, the trial is completed and both objects disappear, cueing the subject that the task is successfully accomplished.

3.4.3 Independent variables of navigation and manipulation

The primary goal of an interface assessment and evaluation is to understand the usability characteristics of different object manipulation and navigation interfaces. Focusing on basic sub-tasks allows us to limit the number of studied techniques and to generalize results beyond their specific implementations so as the results of the evaluation can be applied to all techniques based on their sub-task. A lot of parameters can be measured in tracking, navigation, and interaction inside a VE. Tables 3.7 and 3.8 show two short lists of these parameters for navigation and manipulation interfaces. The main independent variables of interest for the selection task are the distance to the object, the object size, the interaction techniques, and visual feedback. However, as shown in Fig. 3.17, different parameters can be selected from the list for the object manipulation evaluation. The position of an object in a VE is defined as the length d and orientation α, β of the vector pointing from the user's chest to the object (Fig. 3.17)[79].

The size of the stimulus is defined as its non-occluded visual size: the vertical (φ), and horizontal (ϕ) angles the object occupies in the user's field of view (Fig. 3.17). Visual angles are also user-centered units. The geometrical size of test objects is recalculated before each trial depending on the current position of the user, in order to maintain the objects' visual size as specified a priori by the experimenter. The benefit of visual angles is the separation of influence of the distance and the object size on user's performance: when an object's size is defined in terms of visual angles, it has the same visual size at different distances. Visual angles also allow for an easy generalization of the results beyond the particular test VE.

The main independent variables of interest for manipulation tasks are the initial distance to the object, the distance to the final position, the required accuracy of positioning, and the interaction technique.

3.5 Navigation/manipulation interface mathematical definition

As explained before (section 3.3.1), each navigation can be split up to several shorter navigations. In addition, a short navigation then can be decomposed into two types of movement: translational and rotational.

Navigation and tracking parameters		
	Parameter	*Description*
1	Motion platform size	
2	Position tracking error	
3	Resolution	3D display system
		Movement of the scene
		Producing a command by a navigation device
4	Scene content	
5	Time lag	
6	Update rate	
7	Calibration	
8	Color	
9	Contrast	
10	Field of view, degree of control	
11	Flicker	
12	Velocity and acceleration	Rotational
		Translational

Table 3.7: Independent variables for the evaluation of a navigation interface.

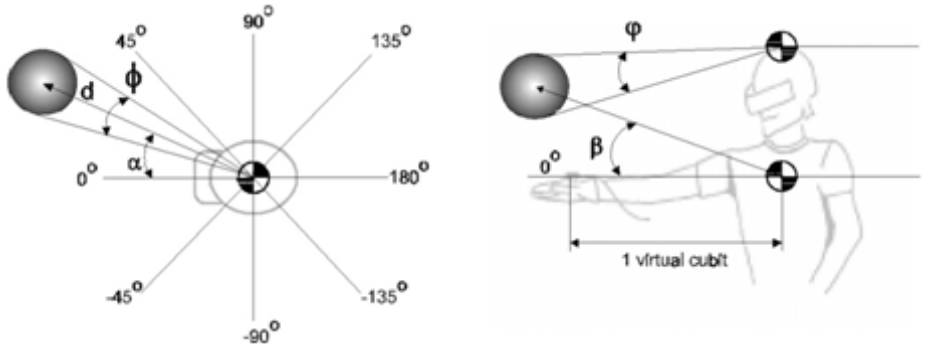


Figure 3.17: The object position is defined as distance d and directions α, β in the user-centered coordinate system. The object size is defined in terms of vertical (φ) and horizontal (ϕ) angles of the visual field subtended by the object.

Canonical object manipulation parameters		
	Parameter	*Description*
1	Selection	Distance
		Direction
		Size
		Density of the objects around the target object
		Number of the targeted objects
		Target occlusion
2	Manipulation (rotation for example)	Distance
		Initial an final orientation
		Precision of the rotation
3	Position	Direction and orientation to the initial position
		Distance and orientation to target position
		Translation distance
		Required precision of positioning

Table 3.8: Independent variables for the evaluation of a manipulation interface.

3.5.1 Translational movement

The real-time user's position $(x_p(t), y_p(t), z_p(t))$ and orientation $(p_0(t), p_1(t), p_2(t), p_3(t))$ are measured by the head tracker system with respect to the immersive room (here a CAVE) origin ($x = 1.5$, $y = 1.5$, $z = 0$ for a CAVE with size $3 \times 3 \times 3$ m) and expressed by (3.1).

$$\vec{P}(t) = (x_p(t), y_p(t), z_p(t)) = x_p(t)\vec{i} + y_p(t)\vec{j} + z_p(t)\vec{k} \tag{3.1}$$

where, x_p, y_p, and z_p are the Cartesian coordinates and i, j and k are the unique vectors along x, y, and z axes respectively. The position vector at time $t + \Delta t$ is represented by $\vec{P}(t + \Delta t)$ and expressed by (3.2).

$$\vec{P}(t + \Delta t) = \vec{P}(t) + \Delta\vec{P} \tag{3.2}$$

where, vector $\Delta\vec{P}$ is called the translation vector and is calculated based on a constant or variable speed. The translation vector depends on the linear velocity, v, and the parameters of the navigation device as shown in Fig. 3.22 for instance.

The displacement value is calculated by (3.3) if the mapping function (f) between the real-virtual worlds

is considered linear.

$$d = f\left(a(t)\right) * v = k_t a(t) v \tag{3.3}$$

The translation matrix is calculated by (3.4) if the movement occurs along x axis.

$$\begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

If the movement is along another direction we need to redirect the displacement vector via the rotation matrix (3.5) to the right direction then apply the translation operator.

$$\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \tag{3.5}$$

If we know the exact direction of the movement we can use (3.6) to calculate the new position, where indices A and B point at the initial and target positions, respectively.

$$\begin{bmatrix} x_B \\ y_B \\ z_B \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ z_A \\ 1 \end{bmatrix} = \begin{bmatrix} x_A + x_B \\ y_A + y_B \\ z_A + z_B \\ 1 \end{bmatrix} \tag{3.6}$$

As we will see later, redirection is calculated by quaternion algebra.

3.5.2 Rotational movement and path correction using the orientation vector

Spatial rotations in 3D can be parametrized using both Euler angles and unit quaternions. A quaternion is an extension of Euler's formula. A given 2D point q with coordinate (r, θ) in the complex plane and (x, y) in Cartesian coordinates can be represented either by a vector (3.7) or Euler representation (3.8).

$$q = x\hat{a}_x + y\hat{a}_y = \hat{i}x + \hat{j}y \tag{3.7}$$

$$x + jy = re^{j\theta} = r\left(\cos(\theta) + j\sin(\theta)\right) \tag{3.8}$$

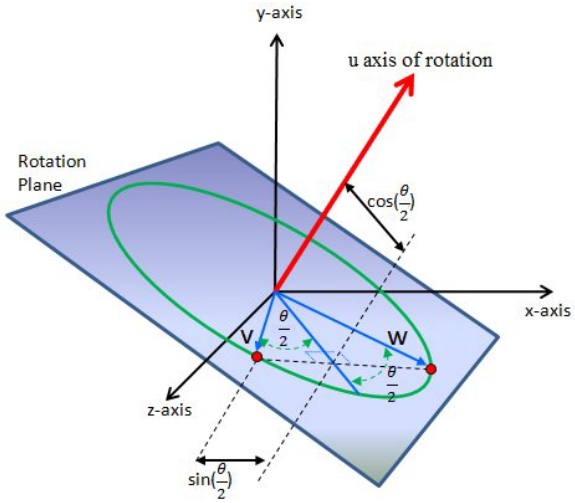


Figure 3.18: Rotation of θ degrees around vector $\vec{u}$ in 3D space.

where, $r = \sqrt{x^2 + y^2}$ and $\theta = \tan^{-1}\left(\frac{y}{x}\right)$ and $\hat{a}_x = \hat{i}$, $\hat{a}_y = \hat{j}$ are unit vectors along x and y axes. Equation (3.5) can be interpreted as a rotation of θ degrees around z axis. In 3D space, point q with (x, y, z) in 3D Cartesian coordinates, as shown in Fig. 3.18, is represented by $x\hat{i} + y\hat{j} + z\hat{k}$ vector. The rotation angle of θ degrees around $\vec{u} = (u_x, u_y, u_z)$ is defined as (3.9) based on quaternion geometry.

$$q = e^{\frac{1}{2}\theta(u_x i + u_y j + u_z k)} = \cos\left(\frac{1}{2}\theta\right) + (u_x i + u_y j + u_z k)\sin\left(\frac{1}{2}\theta\right) \tag{3.9}$$

Now, using equation (3.9), a unit quaternion can be described as (3.10):

$$q = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T \tag{3.10}$$

where, $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$ and the unique quaternion $\vec{q}$ is represented by vector $\vec{q} = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$. Quaternion q can be written with respect to Euler rotation angle by (3.11) as shown in Fig. 3.19.

$$q = \begin{bmatrix} \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right)\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\psi}{2}\right) + \sin\left(\frac{\phi}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{3.11}$$

Equation (3.11) links Euler to quaternion and defines the conversion from Euler to quaternion and vice

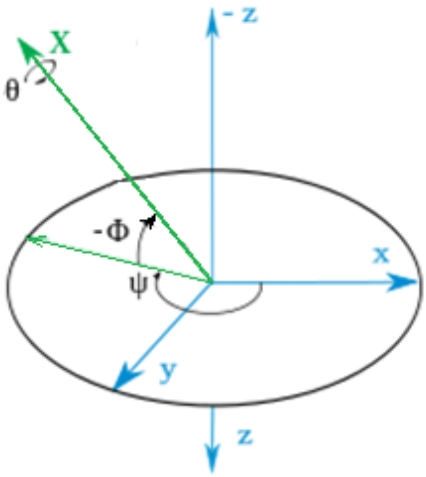


Figure 3.19: Rotation angle in Cartesian coordinates with Euler representation.

versa. As a result, the rotation angles are calculated by (3.12).

$$\begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(q_0q_1+q_2q_3)}{1-2\left(q_1^2+q_2^2\right)}\right) \\ \arcsin\left(2\left(q_0q_2-q_3q_1\right)\right) \\ \arctan\left(\frac{2(q_0q_3+q_1q_2)}{1-2\left(q_2^2+q_3^2\right)}\right) \end{bmatrix} \tag{3.12}$$

$\arctan$ and $\arcsin$ have a result between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. When the rotation angles lay between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ it is not possible to identify in which quarter we have been located in, and some orientations might be miscalculated. Therefore, we need to replace $\arctan$ by $\mathrm{atan2}$ in (3.12) to generate all the orientations as restated in (3.13).

$$\begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} \mathrm{atan2}\left(2\left(q_0q_1+q_2q_3\right),1-2\left(q_1^2+q_2^2\right)\right) \\ \arcsin\left(2\left(q_0q_2-q_3q_1\right)\right) \\ \mathrm{atan2}\left(2\left(q_0q_3+q_1q_2\right),1-2\left(q_2^2+q_3^2\right)\right) \end{bmatrix} \tag{3.13}$$

A rotation with angle θ, around a unit vector $\vec{u}$, the rotation axis, in a 3D space for Euler representation

is defined by (3.14). Throughout this thesis, equation (3.14) will be used for rotation.

$$R = \begin{bmatrix} \cos(\theta) + u_x^2 (1 - \cos(\theta)) & u_x u_y (1 - \cos(\theta)) - u_z \sin(\theta) & u_x u_z (1 - \cos(\theta)) + u_y \sin(\theta) \\ u_y u_x (1 - \cos(\theta)) + u_z \sin(\theta) & \cos(\theta) + u_y^2 (1 - \cos(\theta)) & u_y u_z (1 - \cos(\theta)) - u_x \sin(\theta) \\ u_z u_x (1 - \cos(\theta)) - u_y \sin(\theta) & u_z u_y (1 - \cos(\theta)) + u_x \sin(\theta) & \cos(\theta) + u_z^2 (1 - \cos(\theta)) \end{bmatrix} \tag{3.14}$$

Usually, $\cos(\theta)$ and $\sin(\theta)$ are replaced with C and S respectively to make the representation of the rotation matrix simpler as seen in (3.15).

$$R = \begin{bmatrix} C + u_x^2 (1 - C) & u_x u_y (1 - C) - u_z S & u_x u_z (1 - C) + u_y S \\ u_y u_x (1 - C) + u_z S & C + u_y^2 (1 - C) & u_y u_z (1 - C) - u_x S \\ u_z u_x (1 - C) - u_y S & u_z u_y (1 - C) + u_x S & C + u_z^2 (1 - C) \end{bmatrix} \tag{3.15}$$

If the rotation takes place around z axis in the plane $z = z_0$, the rotation axis is defined by $\vec{u} = (0, 0, 1)$ then the rotation matrix is rewritten as (3.16) which is the famous rotation matrix we know from analytical geometry.

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

It can be shown that this rotation can be applied to an ordinary vector $\vec{p} = (p_x, p_y, p_z) = p_x\vec{i} + p_y\vec{j} + p_z\vec{k}$ in a 3D space. Vector $\vec{p}$ can be considered as a quaternion with a real coordinate equal to zero and the rotation operation can be expressed by (3.17) (well known as the Hamilton product), where, $\left(p'_x, p'_y, p'_z\right)$ in (3.17) is the new vector after rotation. If p and q are quaternions representing two continuous rotations, then (3.18) is the result of the final rotation which is the same as rotating by q and then by p. The power n of a quaternion rotation q^n is a rotation by n times the angle around the same axis as q.

$$p' = qpq^* \tag{3.17}$$

$$pq\vec{v}(pq)^* = pq\vec{v}q^*p^* = p\left(q\vec{v}q^*\right)p^* \tag{3.18}$$

where, q^* is called the conjugated quaternion and is defined by $q^* = \begin{bmatrix} a & -b & -c & -d \end{bmatrix}^T$ if $q = \begin{bmatrix} a & b & c & d \end{bmatrix}^T$. The conjugation definition in quaternion algebra is the generalized form of a complex number. The conjugation of a complex number $z = x + jy$ is defined by $z^* = x - jy$. As seen, only j is converted into $-j$ to calculate the conjugated complex number. The same operation can be done here,

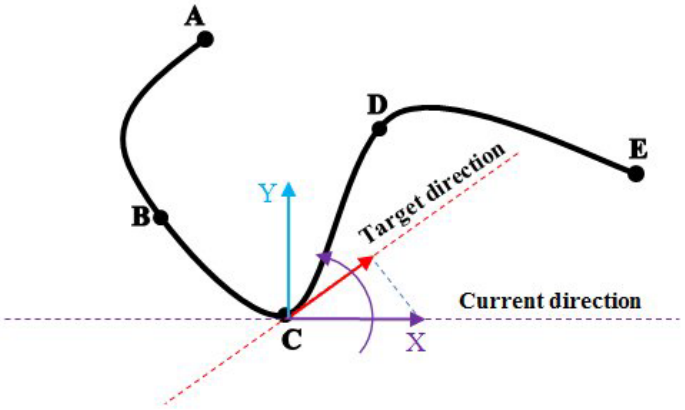


Figure 3.20: Reorientation during navigation.

$q = a + bi + cj + dk$ is conjugated by converting i, j, k into $-i, -j, -k$, $q = a - bi - cj - dk$.

Sometimes the rotation takes place around z axis. In this case, it is much easier to project the navigation path on xy plane to make the calculation simpler and of course faster. In 2D, on $z = 0$ plane, the 2D rotation matrix (3.19) describes the kinematics of the user rotation. The current and target orientations are required to calculate the rotation matrix and make rotation in VEs possible (Fig. 3.20). Common descriptions of a rotation include Euler angles and matrix products as explained above (3.14).

$$M\left(\theta(t)\right) = \begin{bmatrix} \cos\left(\theta(t)\right) & -\sin\left(\theta(t)\right) \\ \sin\left(\theta(t)\right) & \cos\left(\theta(t)\right) \end{bmatrix} \tag{3.19}$$

The simplified rotation matrix (3.19) is used in angular rotation which means vector $\vec{P}(t)$ is redirected to $\vec{P}(t + \Delta t)$ by (3.20) as shown in Fig. 3.20.

$$\vec{P}(t + \Delta t) = M\left(\theta(\Delta t)\right)\vec{P}(t) \tag{3.20}$$

where, $\theta(\Delta t)$ expresses the angular difference between the current and target orientations. The translational movement of the scene is along x and y directions. However, if the movement takes place in 3D space (walking on the stairs, Fig. 3.21) the general form of the rotation and translation matrices (3.6) and (3.14) will be used.

3.5.3 Acceleration/velocity of the movement

The movement will be accelerated/decelerated at the beginning/end of the navigation. Aside from this, the acceleration varies with the movement velocity alternation during natural walking which in turn changes the user head movement. If the head moves rapidly the velocity will increase. If the user tends to slow

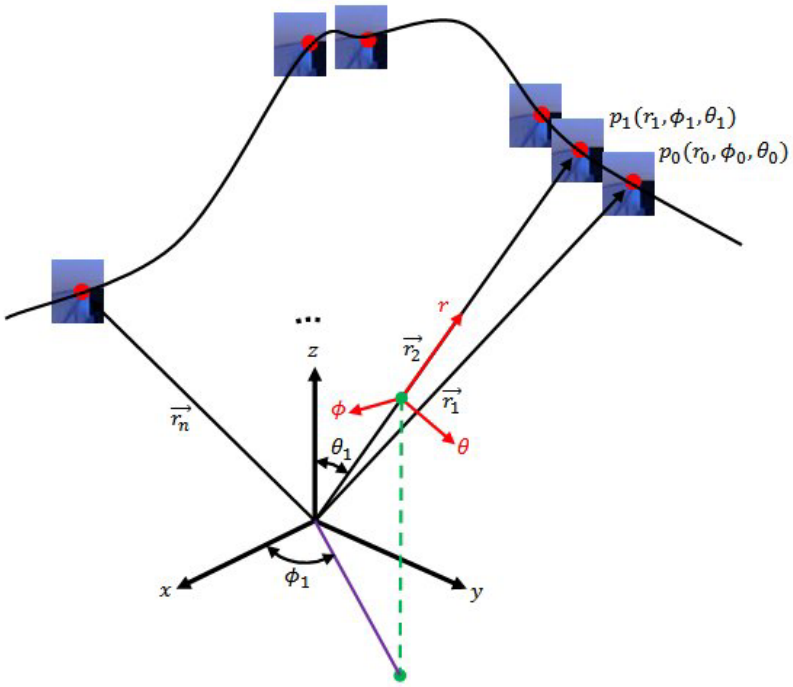


Figure 3.21: Translation and rotation in 3D space.

down, the velocity will decrease, otherwise it will remain constant. The head movement not only represents the velocity and the acceleration but helps the user to experience navigation close to natural walking. The body can adjust the speed autonomously. This automatic adjustment of the velocity is called automatic adaption. There should be a mechanism to control the velocity and the acceleration during navigation by any type of interface. In the interfaces designed based on body motion, the mechanism comes from gesture analysis while in device-based interfaces, either a speed/acceleration curve is used for this purpose or they are kept unchanged.

3.6 Interface design

In this section, the development details of six navigation/interaction interfaces are explained:

1. Device-based interface;
2. Walking in place based on gesture analysis and body motion classification;
3. Walking in place by head movement analysis;
4. Navigation with sound command (speech processing);

5. Bracelet and sensor fusion;

6. Navigation using iDevice such as iPhone, iPad.

The second, third and fourth interfaces are referred to as natural navigation/interaction because at least one element of navigation/interaction in the real environment was incorporated in the design while the other three are artificial ways of navigation. In fact the final aim of the thesis is to compare these two types of interfaces and adjust the parameters of the second group of interfaces to act close to natural. For that reason, the navigation parameters and traveling components will be discussed mathematically before entering to the design details.

3.6.1 Interface one: device-based interface

In device-based interfaces, a command from a navigation device is the starting point of any movement in a virtual reality platform. Therefore, the first set of parameters of navigation and interaction corresponds to navigation devices. The important parameters of a navigation device are:

1. Capability to generate an appropriate command (can generate digital, analog, and tracker data type);

2. The resolution of an analog signal can be easily adjusted;

3. Smoothness of the generated command (generates and transmits commands without delay).

Here, the so-called Fly-stick will be used to introduce a device-based interface. Comparing to the previous generation of navigation devices (mouse, keyboard, etc.), it seems the Fly-stick is capable to generate different commands and has enough digital buttons, analog values and a tracker with position and orientation which makes it a very suitable device for navigation/interaction in VEs. For example, different buttons on a Fly-stick can be allocated to different tasks as shown in Fig. 3.22. Different buttons can be combined to generate more commands if it is required. It can be reconfigured for a new task as well.

As mentioned in section 3.5.2, several parameters are necessary for 3D navigation/interaction devices:

1. Orientation (vertical: θ, horizontal: φ);

2. Resolution ($\Delta\theta, \Delta\varphi, \Delta\phi, \Delta x, \Delta z$);

3. Backward/forward along the orientation and up/down movement ($x \pm \Delta x, z \pm \Delta z$) with respect to the main coordinates (usually the display coordinates);

4. Turning left/right ($\phi \pm \Delta\phi$) in $\alpha\beta$ coordinates;

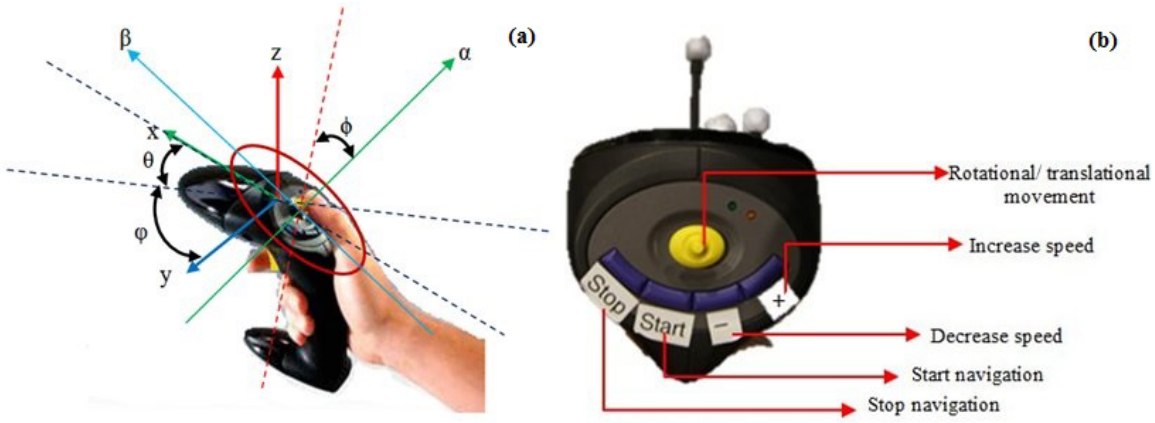


Figure 3.22: Different parameters of a navigation device: a) position and orientation, b) navigation/interaction task definition.

5. Current location of the device (x, y, z);

6. Button for task definition.

Some parameters such as the movement resolution are very important. But, if the device has not this capability it is possible to code it in the client application. Besides, the Fly-stick has a mechanical resolution however, it is recommended to deactivate this capability and activate the software resolution. More accurate movement can be designed by the software resolution. Navigation paths are usually predefined and marked up in the 3D scene to help the subjects follow them exclusively from departure to the destination during the performance evaluation. In such cases, the tracker position is recorded to calculate the efficiency of the navigation device. Any kind of complex movements can be built by combining the elements which have been introduced (items 1-6). Thus, these parameters are the minimum requirements for any 3D navigation device. These requirements need to be incorporated in the navigation devices and not gesture-based interfaces which will be explained later.

A transformation function usually maps analog values (angles) to the navigation parameters. For example, the rotation angle (ϕ) of the joystick (yellow button in Fig. 3.22.b) is used to rotate the 3D scene to the left/right during navigation. It is mapped to the real rotation angle by (3.21).

$$\theta_r(t) = f\left(\phi(t)\right) \tag{3.21}$$

where, function f is any analytical function in $\mathbb{R}$ and is defined in the form of a constant value (3.22) or a polynomial function (3.23) most of the time.

$$f(x) = k_r \tag{3.22}$$

$$f(x) = a_{n-1}x^n + a_{n-2}x^{n-1} + \cdots + a_1 x + a_0 \tag{3.23}$$

For the simplicity of the analysis, equation (3.22) is selected for this study. Therefore, the actual rotation angle is defined by (3.24) and consequently the rotation matrix is calculated by (3.16) or (3.25) if the scene rotates around z axis. If the rotation axis is different, equation (3.14) will be applied.

$$\theta_r(t) = k_r\phi(t) \tag{3.24}$$

$$R = \begin{bmatrix} \cos(k_r\phi(t)) & -\sin(k_r\phi(t)) & 0 \\ \sin(k_r\phi(t)) & \cos(k_r\phi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

Each 3D point associated with a given scene, with coordinate $A = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix}$, will be in position $B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}$ after rotation by applying the rotation matrix.

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} \cos(k_r\phi(t)) & -\sin(k_r\phi(t)) & 0 \\ \sin(k_r\phi(t)) & \cos(k_r\phi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} \tag{3.26}$$

However, because of the simplicity of the calculation, quaternion equations are applied in practice. In that way, the first rotation vector (3.27) in Cartesian coordinates is calculated. In (3.27), we assume that the scene rotates around z axis.

$$\begin{bmatrix} \theta_r \\ \phi_r \\ \psi_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ k_r\phi(t) \end{bmatrix} \tag{3.27}$$

Then, quaternion q is calculated by (3.13) and yields (3.28).

$$q(t) = \begin{bmatrix} \cos\left(k_r \frac{\phi(t)}{2}\right) \\ 0 \\ 0 \\ \sin\left(k_r \frac{\phi(t)}{2}\right) \end{bmatrix} \tag{3.28}$$

Finally, the rotation of the scene (3.29) is achieved by applying the Hamilton product (3.17).

$$\begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = q(t) \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} q^*(t) = \begin{bmatrix} \cos\left(k_r \frac{\phi(t)}{2}\right) \\ 0 \\ 0 \\ \sin\left(k_r \frac{\phi(t)}{2}\right) \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} \begin{bmatrix} \cos\left(k_r \frac{\phi(t)}{2}\right) \\ 0 \\ 0 \\ -\sin\left(k_r \frac{\phi(t)}{2}\right) \end{bmatrix} \tag{3.29}$$

As seen, the calculation is fairly time consuming in a real-time process especially when millions of points are rotating with a simple command. Therefore, it is not easy to select a transfer function with more parameters like (3.23) to map the analog value to the real rotation angle, the more parameters the more delay. The translation movement is calculated along a given motion vector in the same way. For instance, if the second analog value is selected for translation (a joystick generates two analog values), it will be converted to real displacement values by (3.30) by a transfer function similar to that of rotation, however the motion vector along x axis is calculated by (3.31).

$$\Delta d = k_t d(t) \tag{3.30}$$

$$\overrightarrow{\Delta d} = \begin{bmatrix} k_t d(t) \\ 0 \\ 0 \end{bmatrix} \tag{3.31}$$

If the target direction is along x axis, then simply the translation matrix is calculated by (3.32) and the scale factor is set to 1 because the environment is real-scale. The future position of given points A and B, is calculated by applying the translation matrix to A as shown in (3.33).

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.32}$$

$$\begin{bmatrix} x_B \\ y_B \\ z_B \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_A \\ y_A \\ z_A \\ 1 \end{bmatrix} = \begin{bmatrix} \Delta x + x_A \\ \Delta y + y_A \\ \Delta z + z_A \\ 1 \end{bmatrix} \tag{3.33}$$

If the target direction is different, we need to rotate the scene to reorient along the desired direction first. In both examples we assumed the velocity is constant. We can assume a accelerated movement by adding one more parameter to the equations. In a accelerated/constant-speed movement, $\theta_r(t)$ is defined by (3.34) and (3.35), respectively.

$$\theta_r(t) = k_r(\omega t) \tag{3.34}$$

$$\theta_r(t) = k_r\left(\omega t + at^2\right) \tag{3.35}$$

The same assumption can be made for translational movement. Here, the selection of a simple transfer function becomes clear again. We will repeat the same strategy to explain the next navigation/manipulation interfaces.

3.6.2 Interface two: walking in place by Kinect (WIP)

The first step for understanding the walking in place interface using a Kinect is to know whether the Z-depth camera is working. For that, the principle of the infra-red camera in the depth ranging is explained first.

Depth signal and object coordinates calculation using an infrared camera

The relation between the distance of an object (k) to the sensor relative to a reference plane and the measured disparity (d) is shown in Fig. 3.23. A depth coordinate system with its origin at the focal point of the infrared camera will be established to express the 3D coordinates of points on an object. The Z axis is orthogonal to the image plane towards the object, the X axis perpendicular to the Z axis, and the Y axis orthogonal to X and Z making a right handed coordinate system. Assume that an object is located in the

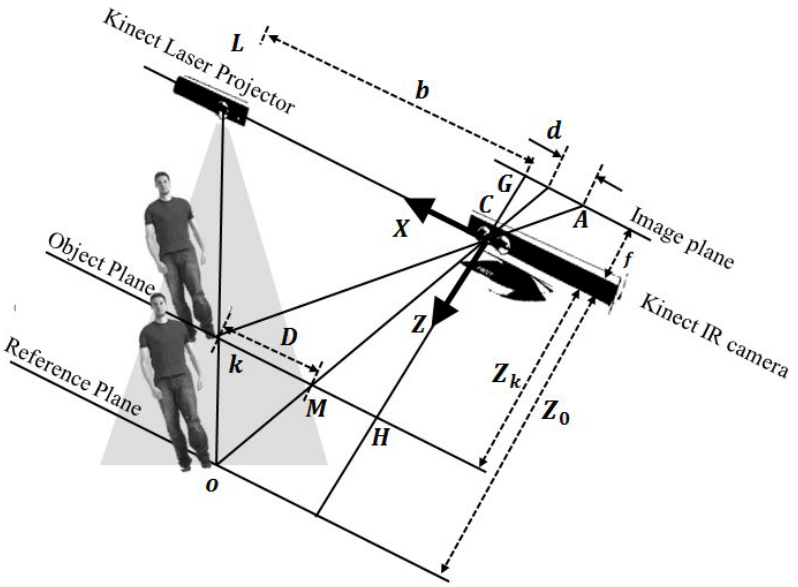


Figure 3.23: Schematic representation of the depth-disparity relation.

reference plane at a distance Z_0 and a point on the object is projected on the image plane of the infrared camera. If the object is shifted closer to the camera, the new location of the point on the image plane will be displaced along X direction (disparity d).

Substituting D from $\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0}$ (from the similarity of ΔOCL and ΔOMK triangles) into $\frac{d}{f} = \frac{D}{Z_k}$ (from the similarity of ΔHKC and ΔCGA triangles) and expressing Z_k in terms of other variables yields (3.36)

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{fb} d} \tag{3.36}$$

where Z_k denotes the distance (depth) of point k in the object space, b is the base length between the camera and the projector, f is the focal length of the infrared camera, D is the displacement of point k in the object space, and d is the observed disparity on the image plane. The constant parameters Z_0, f, and b can be determined by the camera calibration. The Z coordinate of a point together with f defines the imaging scale for that point. The coordinates of each point on the object can then be calculated from its image coordinates and the scale by (3.37) and (3.38).

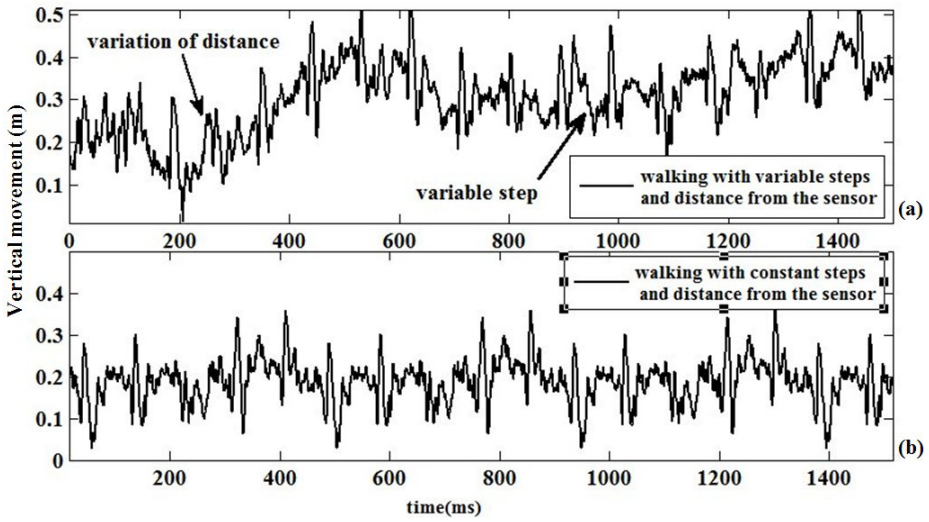


Figure 3.24: Example of a human walking signal with a) constant, b) variable steps and distance from the sensor.

$$X_k = \frac{Z_k}{f}(x_k - x_0 + \delta x) \tag{3.37}$$

$$Y_k = \frac{Z_k}{f}(y_k - y_0 + \delta y) \tag{3.38}$$

where x_k and y_k are the image coordinates of the point, x_0 and y_0 are the coordinates of the principal point, and δx and δy are corrections for lens distortion.

Walking signal definition

The movement of a user is limited to a small area in a multi-projector real-scale 3D system. Therefore, the user is not free to walk more than few meters. Throughout this thesis when we are talking about walking, it means the user will stay in a specific coordinate and will move his feet up and down while his body is not going forward or backward. In different literature, this interface is known as walking-in-place (WIP) interface, as explained in section 3.3.2.

When the user is walking in place, the left and right ankles are moving up and down one after another (Fig. 3.25.a). If only the motion of the point attached to the right ankle (based on the skeleton definition shown in Fig. 3.25.c) is captured by IR-cameras (Fig. 3.25.b) and the X, Y, and Z coordinates of the point are calculated by equations (3.36)-(3.38), the variation signal along Y axis (the Y axis is in accordance with the coordinates defined in Fig. 3.23) can be depicted as Fig. 3.24b. The signal is called "walking with a constant step length". The difference between two picks shows the step length in Fig. 3.24.b. The user

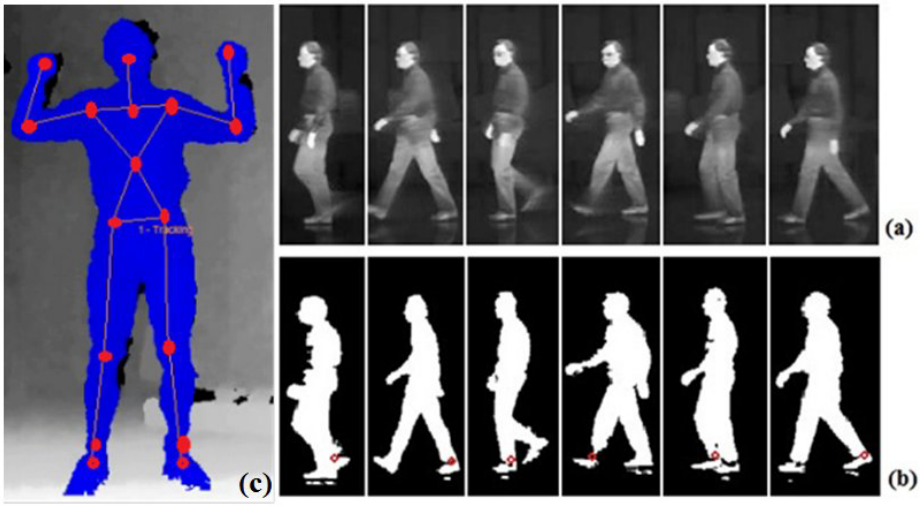


Figure 3.25: IR signal generation using body skeleton: a) IR image, b) IR image after process and c) body skeleton.

might move few centimeters along Z axis (forward/backward) due to involuntary movement of the body. A movement in Z direction changes the amplitude of the signal. When the user is moving involuntarily along Z axis, the scale of the body in the images changes and it makes the same movement of the ankle looks smaller or bigger. The distance between the ankles and the ground is highlighted for four scales on Fig. 3.26 by a small white line near the user's ankle.

That is the reason why the amplitude of the signal is changing by the movements along Z axis (indicated in Fig. 3.24.a). This can be deducted from (3.36) as well. An example of the actual signal is shown in Fig. 3.24.a. As involuntary movement of the body is controlled by the autonomous part of the Central Nervous System (CNS), it is very hard to control this movement, especially when the user focuses on a navigation task (see chapter 2).

The movement along X axis has the same effect. Unlike Fig. 3.24.b, the steps have different lengths in Fig. 3.24.a. Moreover, when the ankle comes up more, the amplitude of the signal is bigger and the length of the step grows. For instance, around sample 1000, a longer step has been taken (Fig. 3.24.a). When the amplitude and the step length change, the analysis becomes harder and a more complicated algorithm needs to be used which creates delay in the calculations and makes the problem more complicated. To solve the problem in a simple way, we used a mechanical restriction as shown in Fig. 3.27. When this problem is solved, the navigation task can be assigned to a set of specific features of the walking gait signal. However, another method based on wavelet has been proposed to solve the amplitude alternation problem automatically.

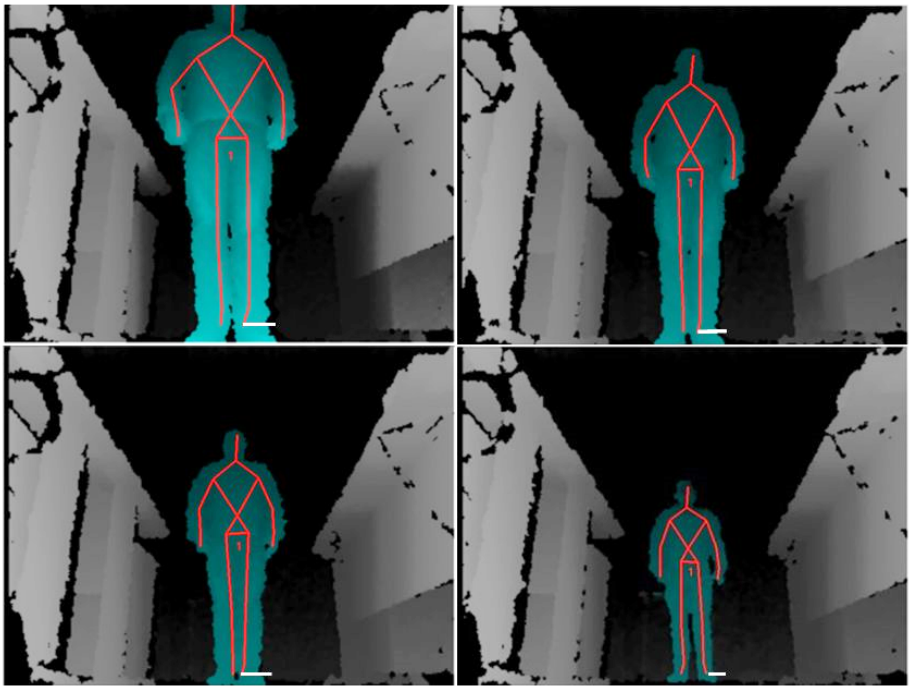


Figure 3.26: Scale variation (movement in Z direction) during walking in a real-scale 3D model.

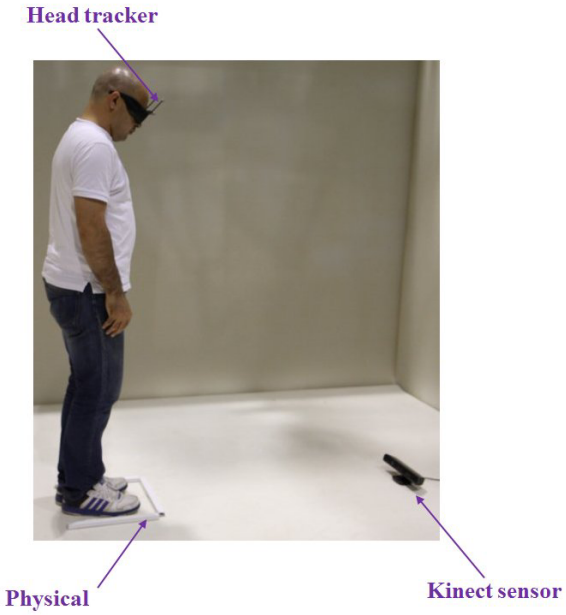


Figure 3.27: Mechanical restriction for the elimination of involuntary body movement.

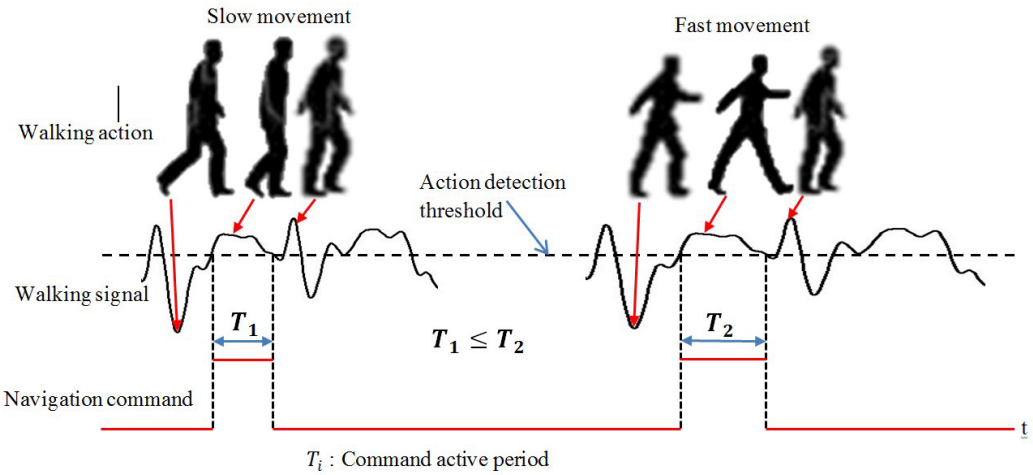


Figure 3.28: Interpretation of a gait signal for navigation command generation.

Navigation command generation based on walking signal analysis

The navigation task is constructed either by forward/backward movements and turning to the left/right or by the combination of these movements and the itinerary as explained before. It means, to go from point A to C by the way of point B, we go from point A to B while using only translational (forward/backward) and rotational (left/right) movements, and then continue from B to C in the same way. Thus, the very basic navigation task is to go from point A to B by combining translational and rotational movements. If this task works perfectly, then it will be repeated in a for-loop to facilitate movement continuously and real-time for different direction and destinations.

In this approach, user's gesture is interpreted and the result is coded into a button type. Then, the different values of the variable are used to initiate or terminate different tasks. A simple interpretation of the gesture will be introduced here just to make the methodology clear. The current interpretation uses the threshold as an evaluation criterion. The analysis applies the criteria to the walking signal to generate a command pulse.

The user's walking signals (solid black line) are shown in Fig. 3.28. The dot-line presents an activation threshold (depicted on the walking signal, at the top of Fig. 3.28). The forward function will be activated if the position of the ankle reaches to the threshold level and will remain active till the ankle is above the threshold. During a short period of time, a pulse will be activated. The length of the pulse is equal to the step length. The pulse activates and deactivates forward/backward movements. The acceleration of the scene movement is adjusted to the average value of the walking signal. In this interpretation, if the user's step is bigger, the scene will move more because the pulse is longer.

In this interface the amount of displacement is proportional to the pulse duration and is calculated by

(3.39).

$$d = kT_i(t) \tag{3.39}$$

The rest of the calculation is similar to what has already been explained in (3.30) through (3.33). The rotation signal is generated by hand movement. When the right hand reaches 10 cm above the shoulder, the pulse becomes active and remains active till the time the hand is in this position. When it comes down, the pulse is set to zero. The rotation angle is calculated by (3.40) in the same way as displacement. The rest is similar to the Fly-stick mode.

$$\theta_r = k_r T_{ir}(t) \tag{3.40}$$

3.6.3 Interface three: walking in place using head movement analysis

As seen in the first interface, a button is allocated to each subtask (translation and rotation). By pushing each button the associated function is activated and the task begins. The function will run until the termination condition is met either by pressing another button or by checking a variable status which refers to the result of measurement analysis. In the second approach, user's gesture is analyzed and interpreted as an action and the result is coded into a button. Then, these values are used to initiate or terminate different navigation tasks of the client application. If the user starts walking in place, the head will swing to the left and the right. This movement can be used to initiate or terminate a navigation function such as translation or rotation.

Fig. 3.29 shows the system architecture for walking in a virtual environment. The system has six main parts including: measurement devices, movement detection, infrared head tracking system, path correction and command generation, speed adjustment and visual display.

Translational/rotation by head movement analysis

The real-time user's position (x, y, z) and orientation (p_0, p_1, p_2, p_3) are measured by ART head tracker. As in section 3.5.1, the position vector is represented by $\vec{P}(t)$, (3.41).

$$\vec{P}(t) = (x_p(t), y_p(t), z_p(t)) = x_p(t)\vec{i} + y_p(t)\vec{j} + z_p(t)\vec{k} \tag{3.41}$$

The position vector at $t + \Delta t$ is represented by $\vec{P}(t + \Delta t)$ and the difference vector with respect to time t is expressed by (3.42).

$$\Delta\vec{P} = \vec{P}(t + \Delta t) - \vec{P}(t) \tag{3.42}$$

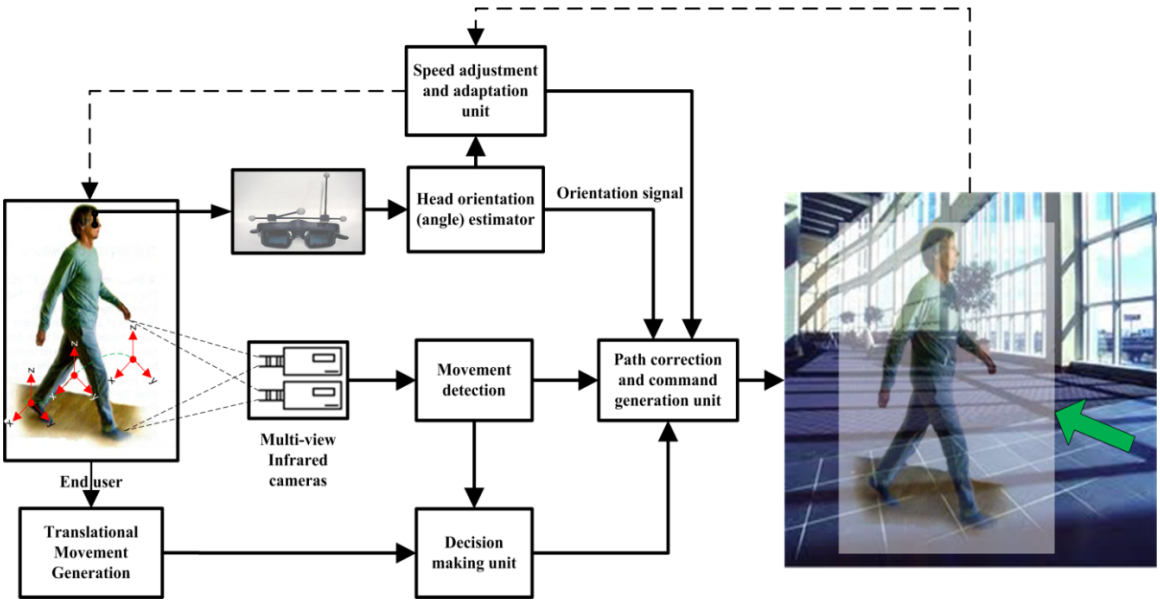


Figure 3.29: Block diagram of a navigation system by walking gait analysis.

Now, the detection function is defined as (3.43).

$$D = \begin{cases} 1 & \text{if } \left|\Delta\vec{P}\right| \geq \epsilon \\ 0 & \text{if } \left|\Delta\vec{P}\right| < \epsilon \end{cases} \tag{3.43}$$

The detection threshold, ϵ, is $\epsilon = 0$ in theory. However it is non-zero practically (due to involuntary movement of the body which creates non-zero movement even if the user stands still). In fact, if the position of the user changes from t to $t + \Delta t$ or $D = 1$, the translational movement will start and moves the scene in $+y$ or $+x$ directions (z upward). This movement will stop when the user stops walking, $D = 0$. The rotation is done based on the head orientation instantaneously. When the head orientation is changed, the rotation matrix as explained in (3.19) changes consequently.

3.6.4 Interface four: fusion of speech command with WIP

Signal processing approach for two types of sensors

Two sensor configurations will be introduced in this study: 1) microphone-array (voice) and 2) infrared depth (IR-D) sensor (video). Each sensor employs its own unique configuration and processing approaches to capture and analyze the associated signal.

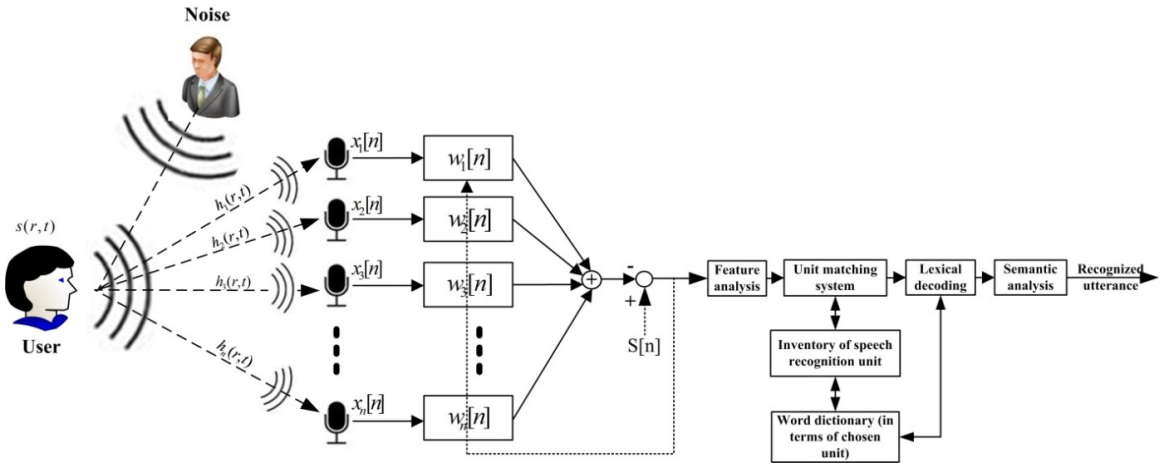


Figure 3.30: Scheme for microphone-array sound capturing and speech processing in a noisy acoustical environment.

Audio sensor (microphone-array)

A robust real-time speech processing approach with the presence of noise while creating less delay will be addressed here. We consider a sound capture situation, noise, echoes and reverberation as sketched in Fig. 3.30. The channel impulse responses $h_i(r,t)$ describe the sound propagation from the source to the individual microphones. The discrete-time beamformer is modeled by an FFT overlap-add filter bank. The Minimum Variance Distortionless Response (MVDR) Beamformer algorithm in the frequency domain is used to analyze this multi-channel system. An MVDR Beamformer optimizes the power of the output signal under the constraint that the signals from the desired direction are maintained.

Optimization constraints (3.44) can be solved using Lagrange's method (3.45):

$$w_0 = \arg\min_w w^H S_{xx} w \text{ with } w^H h = 1 \tag{3.44}$$

$$\nabla_w \left[w^H S_{xx} w + \lambda \left(w^H h - 1 \right) \right] = S_{vv} w + \lambda h = 0 \tag{3.45}$$

where, S_{xx}, w, ∇_w are the spatio-spectral correlation matrix, beamformer weights and the gradient with respect to the weight vector, respectively. Superscript H and h denote the conjugate transpose and the channel transfer function vector. Combining the constraint equation from (3.44) with (3.45) leads to the well-known solution for the optimum weight vector (3.46)

$$w_0 = \frac{S_{vv}^{-1} h}{h^H S_{vv}^{-1} h} \tag{3.46}$$

If the noise is assumed as an homogeneous diffuse noise and if we estimate S_{vv} for each signal frame

with index m by

$$S_{vv}\left(e^{j\theta},m\right)=\alpha S_{vv}\left(e^{j\theta},m-1\right)+(1-\alpha)v\left(e^{j\theta},m\right)v^{H}\left(e^{j\theta},m\right) \tag{3.47}$$

where $\theta = 2\pi\frac{f}{f_s}$ is the frequency variable and $\alpha \approx 0.8$ (v is the vector of the noise spectra), the optimum weight vector can be found iteratively with a steepest descent algorithm expressed by (3.48).

$$w_{k+1}=w_k-\mu\nabla_w\left[w_k^H S_{xx}w_k+\lambda\left(w_k^H h-1\right)\right]=w_k-\mu\left(S_{vv}w_k+\lambda h\right) \tag{3.48}$$

Lagrange multiplier, λ, is calculated by substituting the second constraint of (3.44) in (3.45). By eliminating λ from (3.48), we finally get the update equation (3.49).

$$w_{k+1}=w_k-\mu\underbrace{\left(I-\frac{hh^H}{||h||^2}\right)S_{vv}w_k}_{g_k} \tag{3.49}$$

In (3.49), the weight vector is updated by using S_{vv} estimated from (3.47) and iterated in each frame. Furthermore, the convergence speed is improved by computing an optimum step size factor μ. We choose the step size so that it minimizes the noise power at the beamformer output in each iteration.

$$\frac{\partial\left(w_{k+1}^H S_{vv}w_{k+1}\right)}{\partial\mu^*}=0 \tag{3.50}$$

Combining (3.49) and (3.50) results in (3.51).

$$\mu_k=\frac{g_k^H S_{vv}w_k}{g_k^H S_{vv}g_k} \tag{3.51}$$

Speech processing

Fig. 3.31 shows the speech recognition approach and the system components. The key signal processing components include a Feature analysis, a Unit matching system, a Lexical coding and a Syntactic analysis. The speech processing algorithm is performed in the frequency and time domains to extract observation vectors which can be used to train the Hidden Markov Model (HMM). The HMM is a strong algorithm to characterize various speech sounds.

First, a choice of speech recognition must be made by a unit matching system. Possibilities include linguistically based sub-word units as well as derivative units. For a specific application (current application), it is both reasonable and practical to consider the word as a basic speech unit. We will consider such systems exclusively in this section. For that, we use only the words included in Table 3.9.

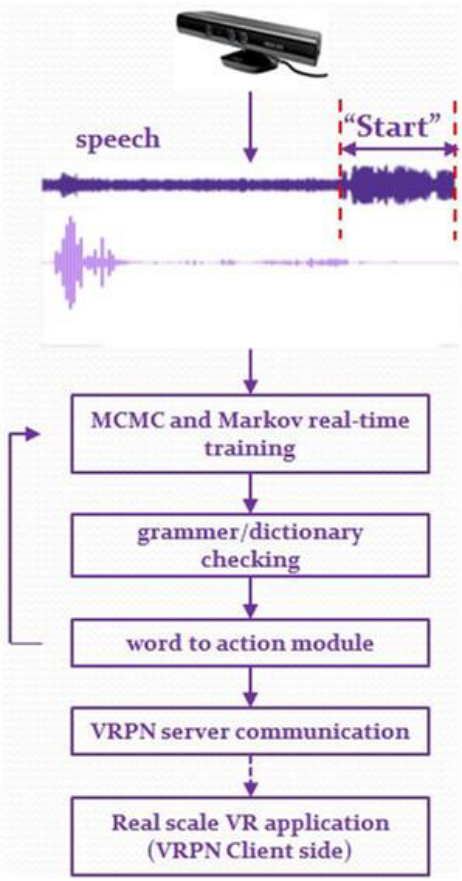


Figure 3.31: Different components of speech processing with a Kinect.

	Vocabulary					
	start	stop	forward	backward	turn right	turn left
Code	000001	000010	000100	001000	010000	100000

Table 3.9: Vocabulary listed in the dictionary and used speech processing system (lexicon).

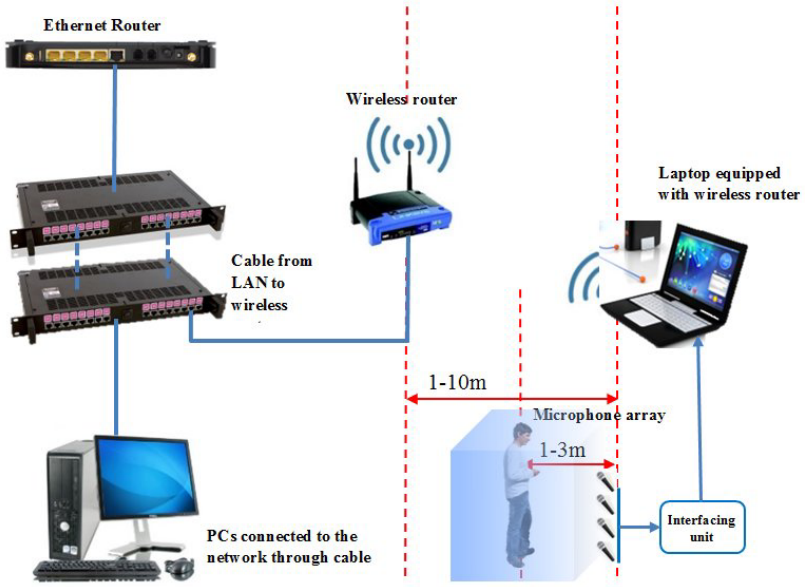


Figure 3.32: Hardware infra-structure for speech processing and data transmission.

The lexical coding process places the constraints on the unit matching system so that the paths investigated are corresponding to the sequences of speech units which are stated in a word dictionary (lexicon). This procedure implies that the speech recognition result must be specified in terms of basic units chosen for recognition. Syntactic analysis adds further constraints to the set of recognition search paths. One way in which semantic constraints are utilized is via a dynamic model of the state of the recognizer. When the vocabulary is recognized by the speech processing system, an associated code (second row in Table 3.9) will be selected (Fig. 3.32), an appropriate command is generated and sent to the navigation application via a wifi network.

IR-D (vision sensor) and laser beamer configuration

An IR-D camera and a laser beamer (Fig. 3.33.a) were arranged in a precise configuration to localize and calculate the position of the user head. The setup is a real-time optical tracking system including four IR-D cameras. The system uses the TOF (Time of Flight) theory (corresponding to interface two), image pro-

cessing based on morphology and triangulation to estimate the position of the head. Five balls (indicators in Fig. 3.33.a, b) were attached to an E-shape marker and mounted on 3D stereoscopic glasses. An example of this type of indicator arrangement is shown in Fig. 3.33.b.

IR image processing for head tracking

The optic tracking system uses a non-linear/non-Gaussian Bayesian algorithm to track the markers (Fig. 3.33.b). From a Bayesian perspective, the tracking problem is to recursively calculate some degrees of belief in the state of x_k at time k, taking different values, given the data $z_{1:k}$ up to time k. Thus it is required to construct the pdf (probability distribution function) $p\left(x_k|z_{1:k}\right)$. The pdf $p\left(x_k|z_{1:k}\right)$ may be obtained, recursively, in two stages: prediction and update. Suppose pdf $p\left(x_{k-1}|z_{1:k-1}\right)$ at time $k-1$ is available. The prediction stage involves using (3.52) (process model) to obtain the prior pdf (prediction stage) of the state at time k via (3.53).

$$x_k = f_k\left(x_{k-1}, v_{k-1}\right) \tag{3.52}$$
$$z_k = h_k\left(x_{k-1}, n_k\right)$$
$$p\left(x_k|z_{1:k-1}\right) = \int p\left(x_k|x_{k-1}\right) p\left(x_{k-1}|z_{1:k-1}\right) \mathrm{d}x_{k-1} \tag{3.53}$$

At time step k, a measurement z_k becomes available and this may be used to update the prior (update stage) via Bayes' rule (3.54).

$$p\left(x_k|z_{1:k}\right) = \frac{p\left(z_k|x_k\right) p\left(x_k|z_{1:k-1}\right)}{p\left(z_k|z_{1:k-1}\right)} \tag{3.54}$$

where, the normalizing constant is calculated by (3.55)

$$p\left(z_k|z_{1:k-1}\right) = \int p\left(z_k|x_k\right) p\left(x_{k-1}|z_{1:k-1}\right) \mathrm{d}x_k \tag{3.55}$$

Now by having the position and the orientation of the head, we can implement the navigation task (rotation by head orientation). Any rotation in a 3D space can be represented as a combination of an axis vector and an angle of rotation. Quaternions give a simple way to encode this axis-angle representation in four numbers and apply the corresponding rotation to a position vector representing a point relative to the origin in $\mathbb{R}^3$.

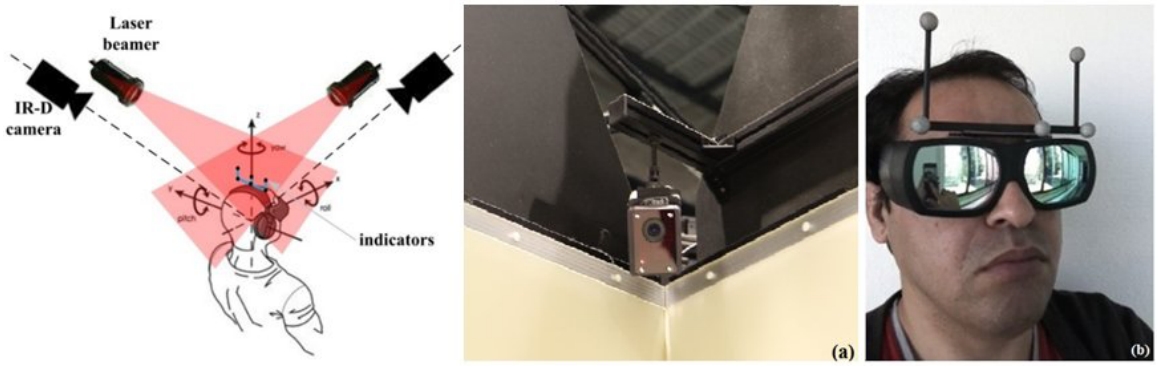


Figure 3.33: Left) vision configuration (IR-D camera-laser beamer), right) indicators in an E-shape arrangement.



Figure 3.34: Microphone-array configuration.

Translation/rotation command generation

When the word "start" is said, the translational movement begins and by saying "stop" the movement is terminated. The velocity and acceleration can be automatically/manually adjusted by the navigation function. The rotation angle around Yaw axis (Fig. 3.33.a) is used for rotation and reorientation of the movement. Figs. 3.34 and 3.35 show the audio, video and sensor fusion configuration. The user was standing 1.5 meter away from the Kinect microphone-array (Fig. 3.34) and commanding the system.

The rotation command comes from ART head-tracking with a configuration shown in Fig. 3.35.a.

Two sensors in the previous setups were combined in sensor fusion (Fig. 3.35.b). The translational movement can be initiated and terminated by both a Fly-stick and the user head movement (the person imitates walking in place and the algorithm, as designed and detailed above, see section 3.6.2, captures and analyzes the movement).

Sound and vision sensor fusion

A sensor fusion scheme for combining video and audio sensor data is shown in Fig. 3.36. A complete navigation task in a scale-one 3D display or simulator consists of translational and rotational movements. Audio and video data are analyzed and processed to provide a command for translational and rotational movements respectively.

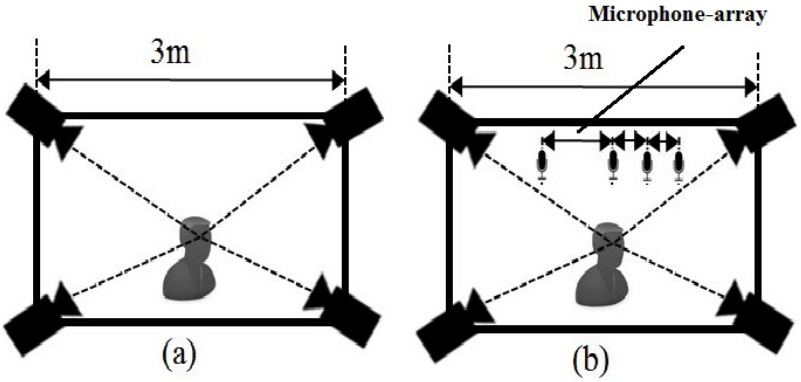


Figure 3.35: a) Head tracking system and b) sensor fusion configuration.

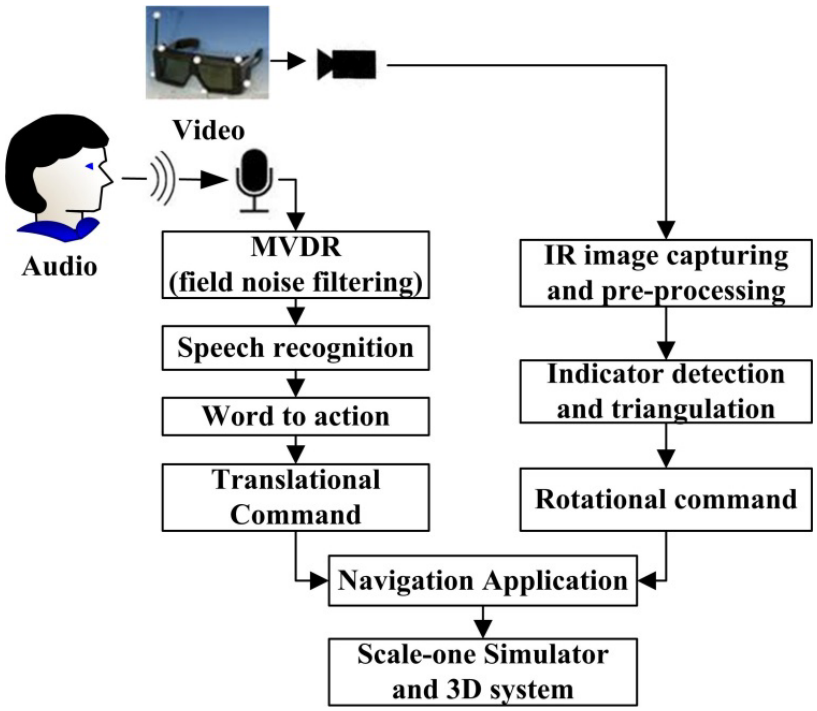


Figure 3.36: Sensor fusion scheme for combining audio and vision sensor data.

3.6.5 Interface five: magic bracelet

An interface was designed based on a pair of bracelets in order to manipulate an object in 3D. In fact, in this interface, both hands are identified and tracked by a high precision AR-tracker and the position/orientation of the hand are used as input in different applications (Fig. 3.37. For instance, the hand position and the orientation are used to select/manipulate and place an object in a scale-one VE. Some of the characteristics of the magic bracelet are as follows:

1. Used mostly for object manipulation and menu selection;
2. Provides position, orientation and analog values both for the left and right hands which can be used for hand gesture analysis;
3. Because there is no button on the bracelet so far, it is being used along with the Fly-stick. Now, we are using this bracelet only for tracking purpose;
4. AR-tracker is used to detect and localize the position of the hands and orientation;
5. It provides a very high precision data (with good calibration), however the calibration takes some time and more data needs to be recorded for training.

3.6.6 Interface six: iDevice (iPhone 4)

Nowadays, several sensors such as GPS, gyrometer, the electromagnetic field, temperature are embedded in different smart phones. Different Apple devices we will call iDevice also are equipped with such sensors which can provide a rich source of navigation/interaction interfaces. In addition, the screen and the keypad of the iDevice can provide another source for interface design. For instance, the sensor output can be treated by signal processing approaches and coded as buttons as it has already been seen in section 3.6.2. Recently, a lot of X-code examples have been released for speech processing and gesture detection. The finger movement on the touch screen of any iDevice can be interpreted as a gesture. The gesture consequently can be streamed as a button using wifi or Bluetooth connections.

In this interface, we are using the keypad buttons as values to navigate in a scale-one VE. For that, three virtual buttons like the mouse buttons are coded on an iPhone as shown in Fig. 3.38.a. Then, the right/left buttons are used for rotational movement and the center button performs translational movement with a constant speed. Different versions of the iDevice-based interface were designed and tested using the similar idea that has already been introduced in the five previous interfaces (Fig. 3.38.b).

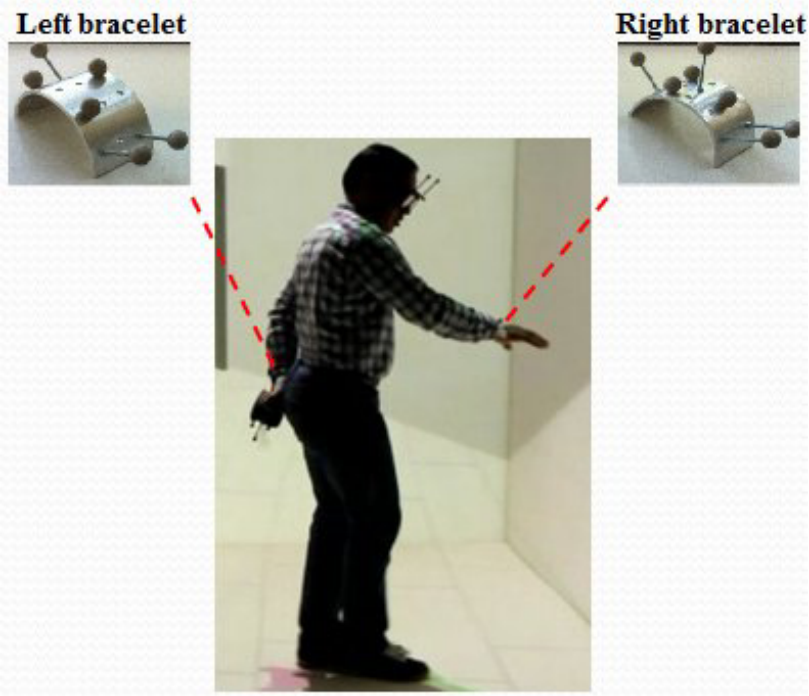


Figure 3.37: Magic bracelet interface for manipulating and interacting with a 3D object in VEs.

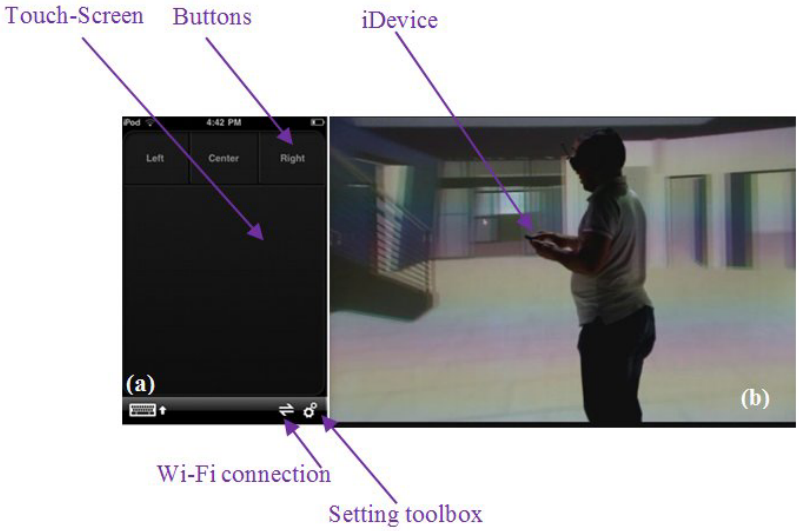


Figure 3.38: Navigation/manipulation with iDevice: a) iDevice interface of VRPN server, b) interface in application.

3.7 Software platform and hardware architecture

3.7.1 System architecture

Fig. 3.39 shows the overall scheme of the software platform and the hardware components with their interconnections. The measurement and calculation unit (MCU) was incorporated in the system to update the appropriate settings for the selected device and dynamically renew the parameters of interfacing units, and provide natural and comfortable navigation for the user. As mentioned above, putting inappropriate settings for a navigation interface creates some problems for the users. This approach can solve the interfacing problem by a proper setting of navigation/interaction parameters. To evaluate the user performance, different strategies are required to be implemented. This requirement was fulfilled by adding a matching system, the device function matching unit (DFMU). A group of participants will be selected to navigate with the interface and their data will be recorded during the test procedure.

This online data is referred to as psychophysiological signal (bio-feedback) throughout this thesis (for instance, EEG represents a bio-feedback in Fig. 3.39). Analyzing these data and having some criteria which will be defined later will help to find out which device fits to a specific application. Signal recording and evaluation process will be discussed in detail in chapter 4. The dedicated arrangement of the software and hardware set-up is called a test-bed, or test-bench, in the literature. The test-bench should both collect information of the user, the navigation device and the 3D system.

Fig. 3.40 demonstrates the complete platform with some extra modules. DT-entity graphic engine was added to make the development of an immersive virtual environment easier and decrease the development time to less than a week. The graphic engine allows adding some objects during the real-time rendering of a 3D model and environmental properties. Some properties like spawning a new object in the scene are interesting in a lot of applications. Moreover, we need to change the properties of the scene such as the color, the lighting, the texture and so on. Sometimes, the VR system requires importing an external 3D model from co-designed or joint platforms (any standard 3D model) into the platform. The engine provides a handy interface to incorporate this type of 3D models easily in the environment.

Additionally, the platform benefited from features such as hardware configuration which makes the parameters setting for a specific hardware and display system very easy. The modified software platform is an extension of the previous architecture shown in Fig. 3.39. The platform was set up on a CAVE system for the implementation and the test.

The platform, called iiVR, uses OpenSceneGraph on the top of OpenGL to render the 3D model. A set of functions for moving forward, backward, up, down, rotation to the left and right have been developed for navigation purpose. The velocity of the straight and rotational movements can be controlled by the user via a pair of buttons on the device or by gesture analysis. One button is allocated for rotational velocity adjustment and the other for straight movement. However, there are different ways to adjust the

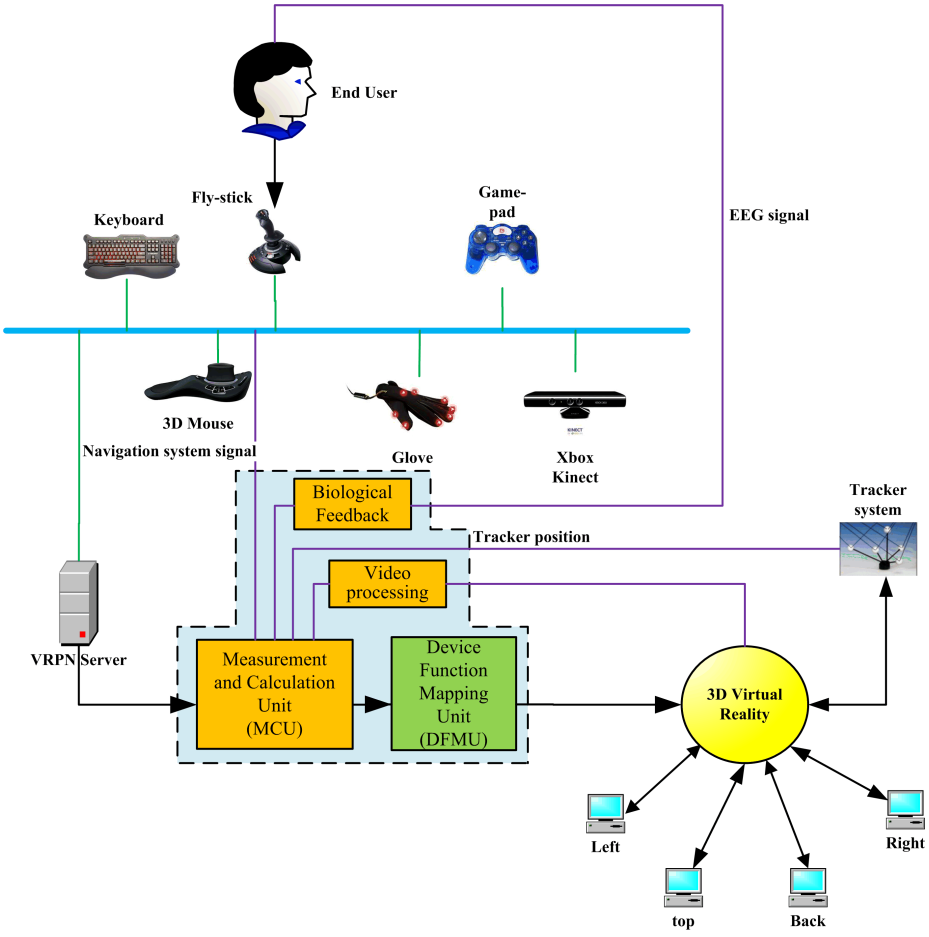


Figure 3.39: Overall scheme for the system development and software platform.

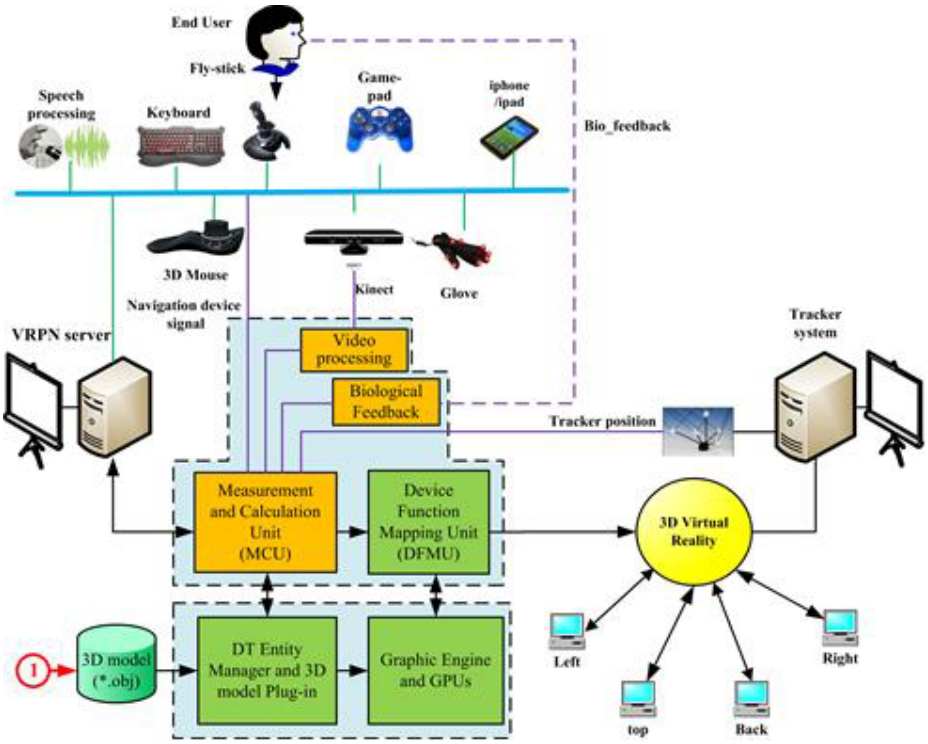


Figure 3.40: Extended software platform with the graphic engine.

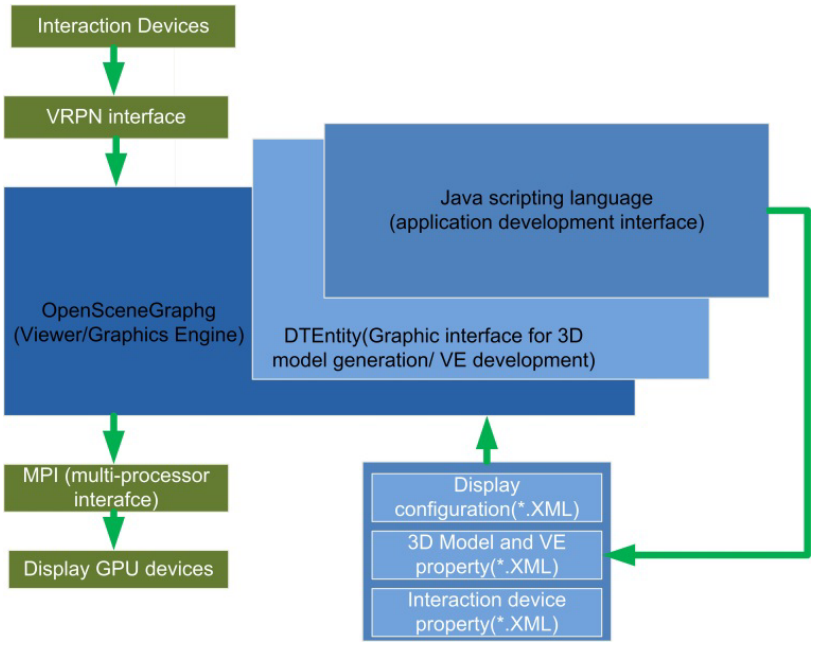


Figure 3.41: Development platform.

speed and acceleration of the movement which has already been discussed. The resolution of the velocity variation (increased and decreased steps) can be either constant or variable. When the step is set to be a constant value, each step is 0.05 unit by default and needs to be replaced with the desired value (it can be adjusted by an internal function non-linearly as well). The device is connected via VRPN to the VR system. The software architecture is shown in Fig. 3.41 in more detail which shows the connection between input/output devices as well as the software component interconnection; four main configuration files manage the device, the display system, the VE and Javascript coding. More details can be found in the iiVR user guide.

VRPN (Virtual-Reality Peripheral Network) was proposed and implemented by Russell M. Taylor et al.[204]. It has been very fruitful to think of VRPN not as providing drivers for a set of devices, but rather as providing interfaces to a set of functions. Particular devices are of one or more canonical device types. VRPN also provides an abstraction layer that makes all devices of the same base class look the same; however, it does not mean that all trackers produce the same report. For example, all tracking devices are known by the type `vrpn_Tracker`. The current VRPN types are Analog, Button, Dial, ForceDevice, Sound, Text and Tracker.

The most used device types in our software platform are listed below:

- *Tracker:* position and orientation;
- *Button:* press and release events for one or more buttons;
- *Analog:* one or more analog values.

Other device types are provided in the source code of VRPN which exist in the VRPN website; new types can be created. Each of these abstracts is a set of semantics for a certain type of device. Like any other type of network, VRPN has two important parts:

1. VRPN server: it is known by a unique address in the network, for instance "`Vrpn_server@localhost`";
2. VRPN client: there are one or more servers for each type of device, and a client-side class to read values from the device and control its operation.

All the parameters of VRPN, the display system and the VE properties (3D model) are configured in three different configuration files which can be found with `*.jconf` (or `*.xml`), i.e., `Display_Configuration.xml`, `3D_model_property.xml`, `Interaction_device_property.xml`, in the "project asset" folder (see iiVR user guide). The Javascript function associated with the navigation/manipulation strategy is embedded in the xml model file because it explains the behavior of the model during navigation/manipulation. It is possible for an application that requires access to specialized features of a certain tracking device (for example, telling a certain type of tracker how often to generate reports), to derive a class that communicates

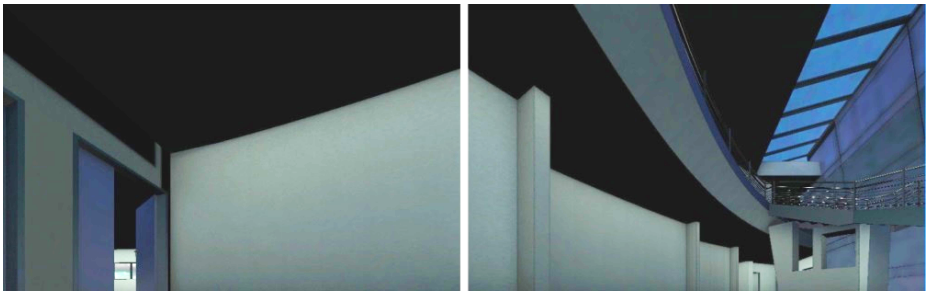


Figure 3.42: Simulation with two screens, left and right views.

with this type of tracker. If this specialized class were used with a tracker that did not understand how to set its update rate, the specialized commands would be ignored by that tracker. Mapping a set of devices into one canonical type requires mapping the different capabilities of each device onto one interface.

3.7.2 Configuration for real experiments and simulation mode

An end-user is connected to a virtual environment via navigation devices, VRPN network and the hardware resources of the operating system (OS). Navigation devices such as a gamepad, a Fly-stick, a mouse, a keyboard and so on, must be connected to the OS (32-bit system in Windows OS, for instance) to be able to propagate their output data into the network. VRPN server uses these devices to collect the tracker position, analog, digital data and send it to the target client application. The server address, device type and driver (`*.dll`) are defined in a VRPN configuration file. The display system, which is usually single or multi-screen, is configured in the display configuration file.

A single screen display (laptop or PC) is used for simulation purpose. Functional simulation is carried out before launching an application on the real-scale system only to test and debug the application code. In the simulation mode, multiple windows appear on the single screen to show different views. An example of this simulation with two (left, right) and four (left, right, top, back) views are shown in Figs. 3.42 and 3.43. The user is located at the point $(-7, -1, -3)$ in the real system coordinates while looking into the interior space.

The display, the device and the VRPN connection are set up in an immersive configuration file. The shadow, the lighting, the texture, the color, the material along with collision estimation and detection are accommodated with the 3D model to build the unified 3D VE. The Javascript code calls the associated libraries and the immersive configuration files to get information from the input devices. The 3D model, its attributes and the script code are called by MPI. The number of processors (depending on the hardware) are allocated by MPI. All these information is sent to the scene viewer for display. Most of the time, an executable `*.bat` file is built to run the application easier. After running the application, the scene appears and we can navigate inside the scene. The command signal is started from the user and ends to the display

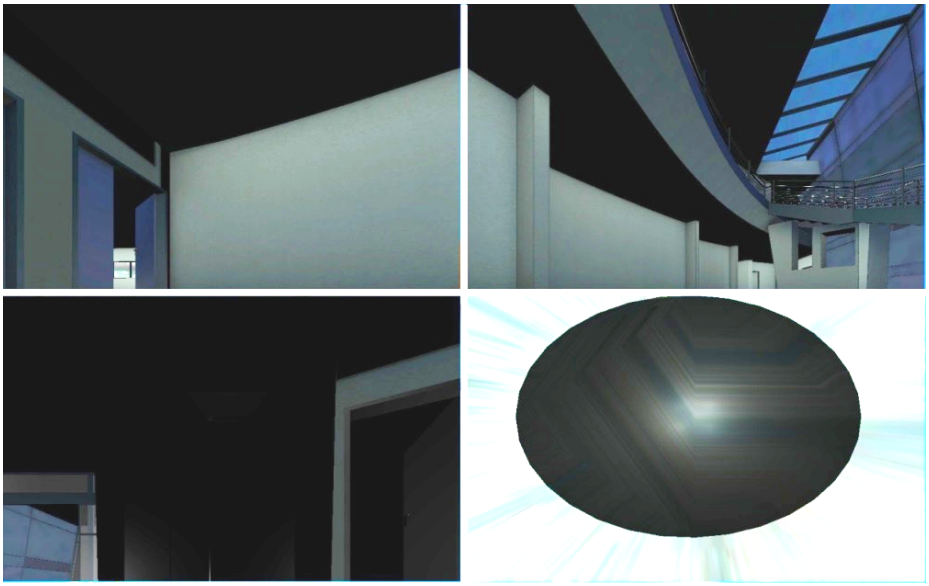


Figure 3.43: Simulation with four screens (views: left, right, top and back) on a single screen display.

system.

3.7.3 Development language

The main development language in the platform is Javascript. Javascript was selected because it is simple, handy and the instructions are easy to remember. All the C++ functions were wrapped under a Javascript code. An example of a Javascripting code is shown in Fig. 3.44. A set of functions have been developed for navigation inside a 3D model. The velocity and the acceleration of translational and rotational movements can be controlled by the head movement, a pair of buttons, speech and the body movement in general. Ascending and descending steps of the acceleration/speed variation can be defined in different ways. For instance, if a Fly-stick is used for navigation, then a pair of buttons can be allocated to control the step resolution. If a motion capture is used for navigation then a natural acceleration/speed of the body movement can be used to adjust the speed.

Since a Javascript function explains the behavior of the model during navigation and manipulation inside the VE, the script is incorporated in the model properties the same way as other properties such as the lighting, the texture, the sound and so on. Any application has two important parts in the Javascript file: an initialization and a body function.

```
function listenMovement()
{
    var rotateValue = -targetComponentFlystickImmersive.Analogs[0] * actualSpeed * 10;
    var roomRotate = [];
    osg.Quat.makeRotate(degToRad(rotateValue),0,0,1,roomRotate);
    osg.Quat.mult(actualAttitude,roomRotate,actualAttitude);

    var displacement = [] ;
    var d = [0,actualSpeed * targetComponentFlystickImmersive.Analogs[1],0];
    var q = [];
    q = targetComponentFlystickImmersive.Attitude ;
    osg.Quat.mult(q,actualAttitude,q);
    osg.Quat.rotate(q,d,displacement);
    osg.Vec3.add(actualPosition,displacement,actualPosition);

    targetComponentRoomImmersive.Position = actualPosition ;
    targetComponentRoomImmersive.Attitude = actualAttitude ;
}
```

Figure 3.44: Example of Javascript code for a basic navigation.

3.7.4 Hardware apparatus and devices

The main compartment of the hardware infra-structure consists in a display system, a head tracking system, and graphic processors. The main display system is multi-projector CAVE system. Fig. 3.45 shows the CAVE system we used for practical experiment during the evaluation process both for navigation and manipulation.

Our CAVE system has four walls (left, right, back and bottom - there is no ceiling and front walls), two projectors per wall (to project images for the left and right eyes). An infrared based head-mounted tracking system (AR-tracker) is used to localize the user position. Devices come along with the system for navigation and control purposes. The model is projected into the display area by MPI and four NVidia Quadroplex 7000 GPUs.

3.8 Conclusion

We presented a review of navigation and manipulation interfaces as well as metaphors. We showed several interfaces we designed for interacting as naturally as possible, as well as technical implementation. Some of these interfaces will be used for conducting experiments and evaluate VIMS.

In the next chapter, we will present different methods for measuring physiological parameters that will allow us to detect and evaluate VIMS.



Figure 3.45: Our CAVE system (multi-screen display system) (3×3 meters across).

4

Signal measurement and processing techniques

4.1 Introduction

Motion sickness (MS) and Visually Induced Motion Sickness (VIMS) are a common experience of numerous people and has motivated extensive engineering, physiological, neurophysiological and psychophysiological research. The focus of early MS and VIMS studies was on the physiological changes emerging during various conditions. For instance, electrogastrography (EGG) signals[26,223] have been employed to detect the symptoms of motion sickness, for instance vomiting, and galvanic skin responses (GSR) have been used to detect sweating[125]. Holmes and Griffin[70] observed increased heart rate variability (HRV) during nausea, indicating the modulation of the automatic nervous system (ANS) in MS. Rapid advances in measurement technology have enabled the neural correlates of MS and VIMS to be examined.

The aim of the current chapter is to introduce a set of psychophysiological signals and the corresponding sensors as well as the processing techniques used to treat these signals. However, only a small set of signals are accessible for measurement and the rest is identified by symptoms. Symptoms are quantified as scores or levels and calculated by psychological self-report questionnaires. Different statistical analysis techniques

will be explained in brief in this chapter.

This chapter is organized in three main sections: section 4.2 explains self-report questionnaires and psychological measurements. Data acquisition from Man-Machine System (MMS) will be explained in an example. We will briefly list different valid questionnaires and explain how to calculate scores from the questionnaires. Later, these scores will be used to rate and compare different navigation/manipulation interfaces using statistical analysis. Few approaches for statistical data analysis will be explained in this section. Different psychophysiological signals along with their measurement tool and sensors will be introduced in section 4.3. Some of these signal processing approaches will be used in this chapter and the following chapters 5 and 6. These approaches will be summarized in more detail mathematically in section 4.4. The final goal is to show how the features from psychophysiological signals are correlated with the variation of the scores (calculated from the questionnaires) to be able to use these features alternatively as indicators of MS/VIMS in a real-time process.

4.2 Psychological measurement

A biofeedback or a psychophysiological signal is a process of gaining greater awareness and real-time monitoring of many physiological functions primarily using instruments that provide information on the activity of the user. It is applied to a system with the goal of evaluating the general condition of a user or returning an abnormal condition back to normal in case sudden changes are detected. Some of the processes that can be controlled include brainwaves, muscle tone, skin resistance, heart rate, pain perception and so on. A real-time psychophysiological signal is used as an input to the control system when it is monitoring the user condition. Biofeedback may also be used to improve health, performance and the physiological changes which often occur in conjunction with changes to thoughts, emotions and behavior. Eventually, these changes may be maintained without the use of extra equipments, even though no equipment is necessarily required to practice biofeedback actually.

As it has been explained in chapter 3 (section 3.7), a typical VE (Virtual Environment) has three parts: 1) a user, 2) navigation devices and 3) real-time 3D rendering and display systems. It is necessary to collect real-time data from these three parts to have a full control over the entire events that are happening during operation. Fig. 4.1 illustrates how data is collected from the user, the navigation devices and the 3D system.

4.2.1 Data acquisition system

Data acquisition will be detailed for three above-mentioned parts with respect to Fig. 4.1 as follows:

3D rendering and display systems

All the virtual reality platforms provided some tools in order to record system side data such as the frame rate, the output images, the rendering system, different signals of Graphic Processing Units (GPUs), General Purpose Processor (GPP) of the host workspace, projectors and external signals from the trackers. However, the number of system side signals should not be long because it makes the visualization process slow and consequently induces some cognitive problems. The parameters of a 3D system are not collected necessarily from a 3D rendering/visualization system, rather it might be recorded by an external recorder system and then transferred into the data acquisition system via a networking interface.

Interaction devices

Data acquisition of interaction devices are highly tied up with the 3D rendering system. However, it can operate sometimes totally independent from the 3D system. Usually, device manufacturers produce their device regarding into the requirements of available 3D display systems. As an example, University of Michigan 3D Lab has been using the Emotive EPOC device[21] for navigation in a virtual reality CAVE. Standard tools that come with the Emotive EPOC SDK and the rendering engine, VR juggler[8], are capable to provide some basic navigation through a 3D scene and data logging during operation. Throughout this thesis, an in-house developed platform called iiVR (see chapter 3, section 3.7) will be used for interaction (navigation/manipulation) in a 3D VR environment. The platform provides a rich source of navigation/manipulation interfaces and data logging tools through a VRPN server for VR system developers. Besides, the tool is connected to MATLAB for data analysis.

In Fig. 4.1, the blocks in gray color show the access points from which the system side measurements are recorded. For instance, this data acquisition set-up was established to measure the delay between the navigation command and the display system operation. This was a great first step into figuring out how reliable the device is and whether it can be used with immersive Virtual Reality.

End-user

Data acquisition from the user side is a very wide area and requires an exhaustive discussion and review over the related literature. Avoiding further extra explanation herein, it is insisted that only a little portion of data acquisition techniques will be covered in this chapter. To make our explanation comprehensive and coherent, first, the physiological function of the brain, cerebral and non-cerebral biological mechanism behind MS/VIMS will be explained in brief before getting into different questionnaires and statistical data analysis. Physiological aspects will be explained here once and will be referred to elsewhere in this thesis. Following this short introduction about the brain anatomy opens the way to better understand the

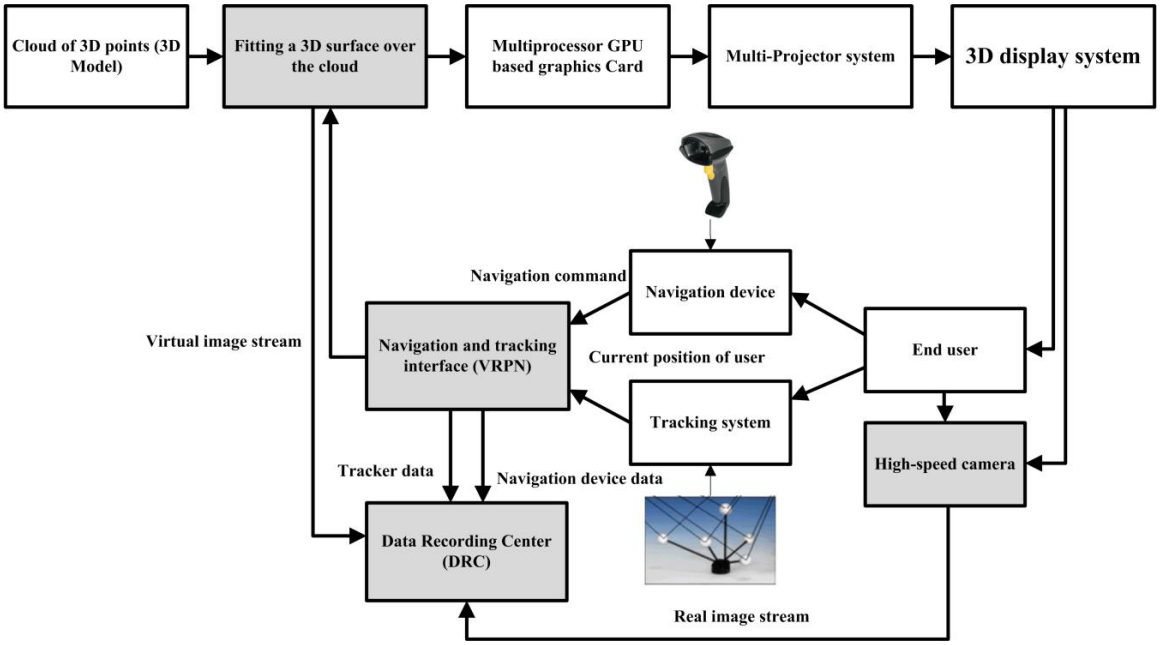


Figure 4.1: Schematic diagram of iiVR platform data logging during a navigation/manipulation task.

mechanism behind VIMS and MS. As explained before, the concentration of the current research is the sickness resulting from motion due to presence in virtual environments. Thus only the parts in charge of VIMS will be focused in this thesis.

4.3 Subjective measurements and statistical data analysis

4.3.1 Self-report questionnaires

Different self-report questionnaires have been created and used for psychological measurements. Kennedy's simulator sickness questionnaire (SSQ)[98] (see appendix B), the presence questionnaire[230], Pensacola Motion Sickness Questionnaire (MSQ)[93] and the NASA task load index[56] were widely used to evaluate VIMS and presence in virtual environments. The subjects experience numerous symptoms of motion sickness including drowsiness, salivation, sweating, headache, dizziness/vertigo and loss of postural stability during immersion or navigation in VEs which are a lot in common with simulator sickness. That is one of the reasons why we will use SSQ.

Measurements of motion sickness by pre and post self-report questionnaires are significantly biased by demand characteristics. Comparative studies of motion sickness in virtual environments should employ experimental designs not subject to such biases or at least take measures to balance the biases. Alternatively, more objective physiological measures may be used with awareness that these measures too can be

subject to experimental demand characteristics as noted in the introduction. Early worries that motion sickness could significantly limit that application of the virtual environment technologies[180] clearly have been shown by experience to have some basis, but motion sickness in a fast, low-latency, well-calibrated system for manipulative simulation does not appear to be a major problem.

Consequently, all the subjects should be screened for absence of illnesses. To reduce the possibility that the screening could be suggestive of motion sickness effects, it should be done in the context of admission to a general subject pool rather than as a requirement for a specific experiment. This type of generic pre-screening may also be used to inform the subjects of the overall possibility of motion sickness without specifically suggesting that a particular experiment might provoke it. Finally, it should be noted that though physiological measurement techniques can provide a potentially more objective motion sickness measure, these also need to be compared to norms since they can be influenced by suggestion[158].

4.3.2 Statistical data analysis

Data is needed to be analyzed across the subjects after collecting data from the subjects to see which parameters are changing significantly or are in correlation with other parameters. For that reason, different statistical tests are used as follows.

T-test

The t-test is used when comparing sample means, when only the sample standard deviation is known. The t-test, and any statistical test of this sort, consists of three steps.

1. Define the null ($H_o : \mu = \mu_o$) and alternate hypotheses ($H_A : \mu > \mu_o$),

2. Calculate the t-statistic for a data,

3. Compare t_{calc} to the tabulated t-value, for the appropriate significance level and degree of freedom.

$$t_{\text{calc}} = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \tag{4.1}$$

If $t_{\text{calc}} > t_{\text{tab}}$ we will reject the null hypothesis and accept the alternate hypothesis, otherwise, we accept the null hypothesis. The t-test can be used to compare a sample mean to an accepted value (a population

mean), or it can be used to compare the mean of two sample sets (4.2).

$$t_{\text{calc}} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{4.2}$$

F-test

The F-test is used to compare two variances or standard deviations. When using the F-test, we again require a hypothesis, but this time, it is to compare standard deviations. That is, we will test the null hypothesis $H_o : \sigma_1^2 = \sigma_2^2$ against an appropriate alternate hypothesis. We calculate the F-value as the ratio of the two variances where, $s_1^2 \geq s_2^2$, so that $F \geq 1$. The degrees of freedom for the numerator and denominator are $n_1 - 1$ and $n_2 - 1$, respectively. As with the t-test, we compare F_{calc} to a tabulated value F_{tab}, to see if we should accept or reject the null hypothesis.

ANOVA

Analysis of variance (ANOVA) is a collection of statistical models used to analyze the differences between group means and their associated procedures (such as "variation" among and between groups). The following formulas are used to calculate an F value for the validation of the hypothesis. Sums of squares are calculated by (4.3) through (4.5) where, p and n_j represents the number of groups and the number of cases in each group respectively.

$$SS_{\text{total}} = \sum_{j=1}^{p} \sum_{i=1}^{n_j} (x_{ij} - \bar{x})^2 \tag{4.3}$$

$$SS_{\text{between}} = \sum_{j=1}^{p} n_j \, (\bar{x}_j - \bar{x})^2 \tag{4.4}$$

$$SS_{\text{within}} = \sum_{j=1}^{p} \sum_{i=1}^{n_j} (x_{ij} - \bar{x}_j)^2 \tag{4.5}$$

Mean squares are calculated by (4.6) and (4.8).

$$MS_{\text{between}} = \frac{SS_{\text{between}}}{p-1} \tag{4.6}$$

$$N = \sum_{j=1}^{p} n_j \tag{4.7}$$

$$MS_{\text{within}} = \frac{SS_{\text{within}}}{\sum_{j=1}^{p} n_j - p - 1} \tag{4.8}$$

Finally F is calculated by (4.9) as follows

$$F_{\text{calc}} = \frac{MS_{\text{between}}}{MS_{\text{within}}} \tag{4.9}$$

The value of F_{crit} for such an analysis is $F_{\alpha,p-1,N-p}$ and can be found in a standard F-table. If $F_{\text{calc}} > F_{\text{crit}}$ the null hypothesis will be proved, otherwise the alternative hypothesis will be proved, similar to the F-test. This test is called one-way ANOVA in a statistical toolbox. The test can be extended to a two-way ANOVA by little modification in the formula.

Correlation

Correlation is defined as any of a broad class of statistical relationships involving a dependence between two variables. Pearson's product moment correlation coefficient (r) is given as a measure of a linear association between the two variables, x_i, y_i (4.10).

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{4.10}$$

If the two variables are correlated, then it is possible to fit a line to the data by $y_i = b_0 + b_1 x_i$ where b_0 and b_1 are calculated by (4.11) and (4.12).

$$b_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{4.11}$$

$$b_0 = \bar{y} - b_0 \bar{x} \tag{4.12}$$

4.3.3 Psychophysiological measurement and data acquisition

Some studies[119] are underway to determine whether advanced psychophysiological biofeedback such as EMG, EOG, EEG, heart rate, muscle contraction and so on[148] are helpful or not. A group of researchers

worked to base the device on NASA's "artificial vision" technology that allows pilots to see a virtual view of what should be outside the cockpit no matter what the actual visibility conditions are. In this project, the sensors attached to the fingers measure the volume of blood pumping through the fingers with each pulse and the temperature of the fingers. A computer translates this data into a 3-dimensional representation of red and blue blood vessels that change the size and shape as blood vessels are manipulated.

In the previous section, some psychological measurements were introduced and data logging from the process was briefly explained. This section will introduce some of the cutting edge sensors and psychophysiological measurements in addition to those that have already been introduced. A measurement device is selected to record some signals from the participants to be able to find common features across the subjects and generalize the result of a subjective study to similar cases. Later, measurement and questionnaire results will be linked to theoretical results via modeling and mathematical descriptions in order to prove our hypothesis and validate the results.

Electromyography (EMG)

Electromyography (EMG) is a technique for evaluating and recording the electrical activity produced by skeletal muscles. EMG (Fig. 4.2.a) is performed using an instrument called an electromyograph, to produce a record called an electromyogram. An electromyograph detects the electrical potential generated by muscle cells when these cells are electrically or neurologically activated. The signals can be analyzed to detect medical abnormalities, the activation level and the recruitment order or to analyze the biomechanics of human or animal movement.

Electro-Oculography (EOG)

Eye movements (Fig. 4.2.b) are measured from the bioelectrical potential appearing between two cutaneous electrodes located on each side of the cornea-retinal axis. This is a very useful measurement because it has been proven that the speed of the eye movement from left to right or up and down can strongly be correlated with VIMS[25], if we analyze this signal in the frequency domain. We should carefully place the sensors in the right position and select an appropriate position for the reference electrode. Any mistake in this step will yield nothing. Recently, most of the smartphones and modern media use this signal as a commanding media. For instance, the Google glass in the X-project uses this feature to command a Skype operation.

- EOG uses the same recording equipment as visual electrophysiology examinations;
- Both eyes can be recorded simultaneously;
- It is limited to horizontal and vertical movements.

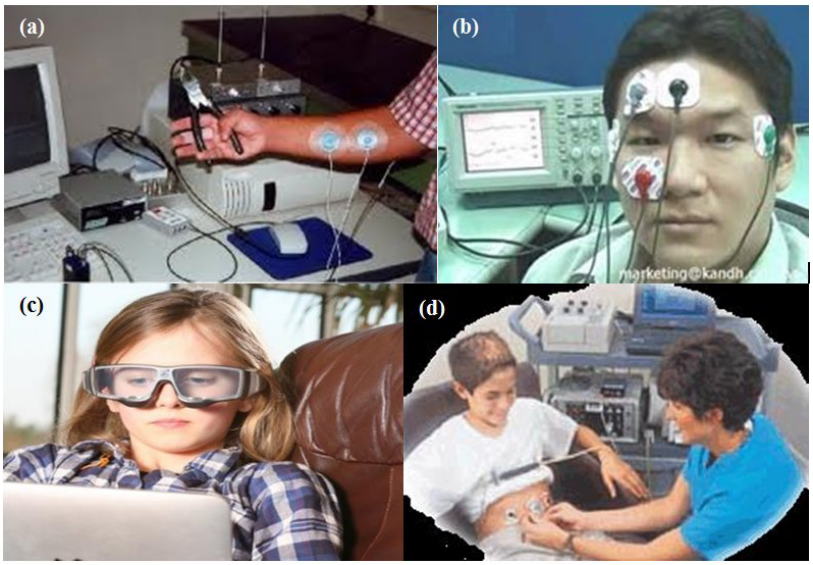


Figure 4.2: Different psychophysiological measurements: (a) EMG [UJI], (b) EOG [233], (c) MET or VOG [SMI] and (d) EGG [1].

Video-Oculography (VOG) and Mobile Eye Tracking System (METS)

The SMI Eye Tracking Glasses [SMI] is a tool for just about any mobile applications, ranging from brand awareness at the point of sale, user experience on mobile devices, man machine interaction or even professional training in sports. There are no adjustments in lifestyle that the user needs to make. This system has an array of 6 IR cameras and an IR emitter LED embedded in the glasses to record the reflecting laser beam from the eye. The precision of the system in the pupil diameter measurement is very high. The software application removes all the eye movement and blinking artifact from the signal. This is an amazing tool to record human reactions during an activity which needs high concentration and visual attention. The gaze direction is measured from the distance between the corneal reflex and the pupil center (Fig. 4.2.c). This technique provides measurements which are:

- Absolute (no drift) and easily quantified;
- Reliable in all gaze directions (horizontal, vertical and oblique);
- Independent from head movements.

Electrogastrogram (EGG)

An electrogastrogram (EGG) is a graph and signal produced by an electrogastrograph, which records the electrical signals that travel through the stomach muscles (Fig. 4.2.d) and control the muscles' contractions.

An electrogastroenterogram (or gastroenterogram) is a similar procedure, which writes down electric signals not only from the stomach, but also from the intestines. Recently some atlas has been proposed by researchers for an appropriate placement of the recording electrodes. An electrogastrogram and a gastroenterogram are similar in principle to an electrocardiogram (ECG) in that the sensors on the skin detect electrical signals indicative of the muscular activity within. Where the electrocardiogram detects the muscular activity in various regions of the heart, the electrogastrogram detects the wave-like contractions of the stomach (peristalsis).

Electroencephalography (EEG)

Electroencephalography (EEG) is the recording of the electrical activity along the scalp (Fig. 4.3.a). EEG measures voltage fluctuations resulting from ionic current flows within the neurons of the brain. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a short period of time, usually 20-40 minutes, as recorded from multiple electrodes placed on the scalp. Diagnostic applications generally focus on the spectral content of EEG, i.e., the type of neural oscillations that can be observed in EEG signals. In neurology, the main diagnostic application of EEG is in the case of epilepsy, as an epileptic activity can create clear abnormalities on a standard EEG study. A secondary clinical use of EEG is in the diagnosis of coma, encephalopathy and brain death. A third clinical use of EEG is for studies of sleep and sleep disorders where recordings are typically done for one full night, sometimes more. EEG used to be a first-line method for the diagnosis of tumors, stroke and other focal brain disorders, but this use has decreased with the advent of anatomical imaging techniques with high (< 1 mm) spatial resolution such as MRI and CT. Despite a limited spatial resolution, EEG continues to be a valuable tool for research and diagnosis, especially when a millisecond-range temporal resolution (not possible with CT or MRI) is required.

Center of gravity (COG)

The center of mass is the balance point of an object's mass. If a pivot was placed at this point, the object would remain in place and be balanced (Fig. 4.3.b). The center of mass of a system is not always at the geometric center of the system. For example, a car's center of mass is closer to the ground rather than in the geometric center of the car so that the car is better balanced. Another example of this is the technique of a high jumper. A high jumper bends his body in a certain way so that the center of mass does not clear the bar, but the body does.

When a system is balanced around its center of mass, it is said to be in a state of equilibrium. The center of mass can be referred to as a pivot point around which the system can revolve. The system revolves due to the rotational equivalents of force, known as torques, which rotate the system either clockwise or counter-

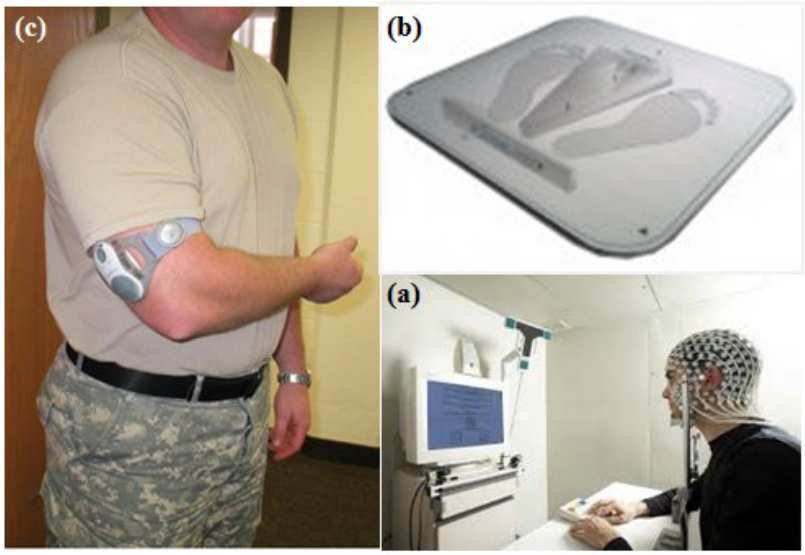


Figure 4.3: (a) Electroencephalography [Unige], (b) Techno-concept center of gravity board [205], (c) Sense-Wear Pro2 GSR sensor worn by a test participant [2].

clockwise. Placing a pivot at the center of mass of a system results in that system being in equilibrium and having a net torque of zero. On each end of a long, rigid body, the torque on one end is equal in magnitude but opposite in direction to the torque on the other end, resulting in a net torque of zero. The center of gravity (COG) of the human body is a hypothetical point around which the force of gravity appears to act. It is the point at which the combined mass of the body appears to be concentrated. Because it is a hypothetical point, the COG needs not to lie within the physical bounds of an object or person. One subjective way (there are objective measures) to approximate the COG of an object is to visualize it balancing on one finger.

Galvanic Skin Response (GSR)

Galvanic skin response is another psychophysiological signal which can be recorded during subjective experiments (Fig. 4.3.c). The GSR signal can be recorded by different sensors. One of the most known is a small, lightweight, unobtrusive body monitor called the Sense-Wear Pro2 armband manufactured by BodyMedia, Inc. (comparable products are available which measure similar autonomic functions). The sensor is worn on the back of the upper arm, which enables continuous physiological data collection (Fig. 4.3.c). Using metallic sensors close to the skin (Fig. 4.4.a), the sensor collects biorhythmic data in real time, with a configurable sample rate, and gathers raw physiological data such as the movement, the heat flow, the skin temperature, the ambient temperature and the galvanic skin response. A data time stamp feature allows the researcher to mark specific events in the data to facilitate later data analysis. The



Figure 4.4: (a) Example of metallic sensors [2], (b) wireless heart rate sensor [Garmin], (c) blood pressure sensor [140], (d) typical ECG signal for a healthy person.

device is designed to provide auditory and tactile feedback during certain events; however, the feature can be altered (through firmware modification) for specific research, so this feedback does not interfere with the experimentation.

Heart rate

Heart Rate Monitors (HRM) have been an inherent part and an important step of diagnosis in the medical field for ages (Fig. 4.4.c). The current generation of portable heart rate sensors has extended the usability of these devices out to the average person for use during day to day activities such as exercise and sports. These devices are typically either mechanical-based (converting vessel ripples using piezoelectric signals) or utilize pulse rate electrical signal detection methods, which require the placement of two or more electrodes on the skin for electric impulse reception. An example of a heart rate signal (associated to an ECG signal) is shown in Fig. 4.4.d. Modern heart rate monitors (Fig. 4.4.c) usually comprise two elements: a chest strap transmitter and a wrist receiver or mobile phone (which usually doubles as a watch or phone). In early plastic straps, water or a liquid was required to get good performance. Later units have used conductive smart fabric with built-in microprocessors which analyze the ECG signal to determine the heart rate. Strapless heart rate monitors now allow users to just touch two sensors on a wristwatch display for a few seconds to view their heart rate.

These are popular for their comfort and ease of use though they do not give as much detail as monitors

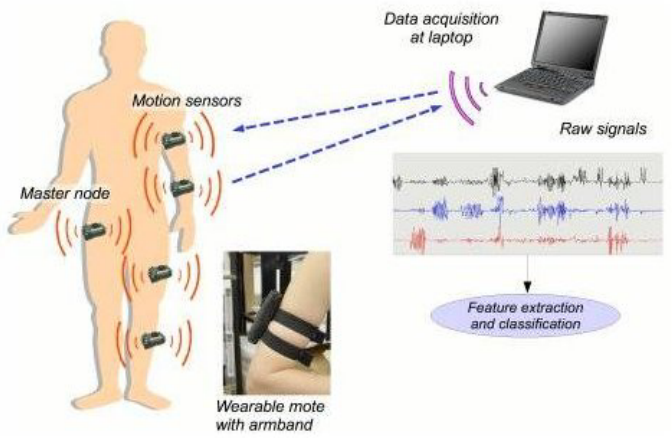


Figure 4.5: Sensor network for blood pressure measurement [140].

which use a chest strap. More advanced models will offer measurements of the heart rate variability, activity and the breathing rate to assess the parameters relating to a subject's fitness. Sensor fusion algorithms allow these monitors to detect the core temperature and dehydration. Another style of heart rate monitors replaces the plastic around the chest strap with fabric sensors - the most common of these is a sports-bra for women which include sensors in the fabric.

Blood pressure

Blood pressure also can be used as a good feedback from the user side. Different sensors for blood pressure are used nowadays but the wireless one is more interesting because of the simple usage of that. An example of this sensor is shown in Fig. 4.4.b. The sensor shown has a measurement module which helps to get accurate values of blood pressure. The sensors can be attached to different organs of the body and synchronized by a measuring center which in turn enables better monitoring and measurement. Fig. 4.5 shows a network configuration for network based measurements. Data from different sensors is transmitted to the workstation PC or laptop and is recorded by the user application simultaneously and synchronously. Most of the softwares and attached SDK provide support for more development and design for specific applications which makes the sensor usability very flexible and efficient.

Digital pill

The Food and Drug Administration in the United States has approved a request by Proteus Digital Health to allow the inclusion of tiny digestible microchips into medicines to assist health care workers in monitoring intake of medicines by patients. This new technology can help to get psychophysiological signals

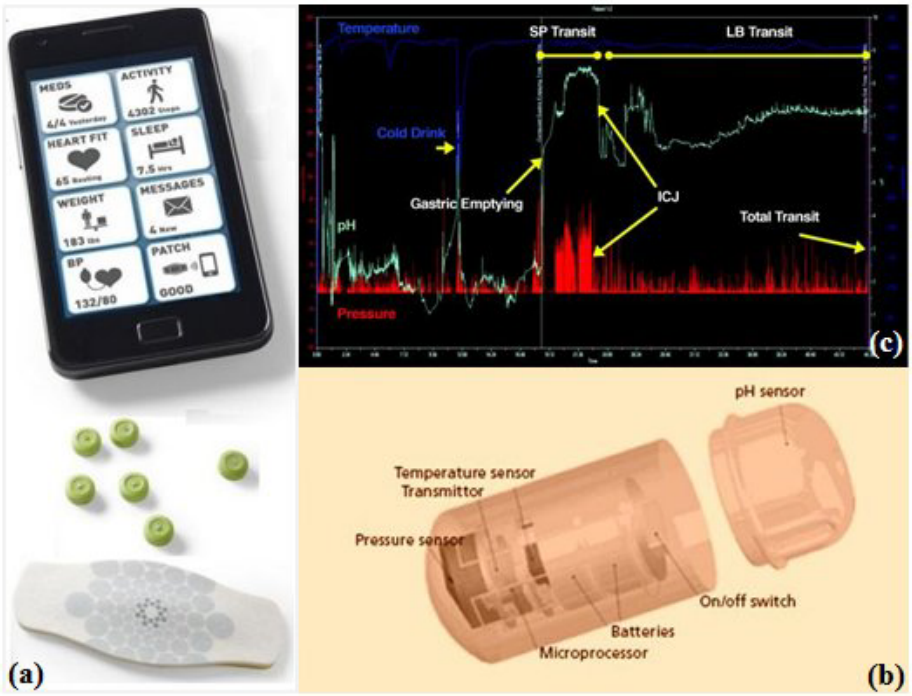


Figure 4.6: (a) Minimum components of a digital pill package, (b) bio-component of a digital pill, (c) digital pill iPhone application and example of output of the transmitter [102].

directly from the stomach, the place which is very important for cyber sickness. The microchip, which is described as about the size of a grain of sand, is made of copper, magnesium and silicon, reacts with the stomach juices when swallowed along with a pill (Fig. 4.6.b). Upon reaction, it sends a signal to a patch the users have applied to their skin where it is relayed to a smartphone (Fig. 4.6.a). The smartphone then relays the information to the researcher or doctor office, allowing the researcher to track how well a subject adheres to instructions on when and how sickness or any other phenomena will emerge (Fig. 4.6.c).

Once the microchip has done its job, it dissolves and passes out of the body along with other digested food. Just as a review over the human aspect of this technology, it is enough to say new technology is being pushed forward by recent reports that have found just half of all patients take their medicines the way they are supposed to, which of course can reduce their effectiveness. An example of digital pills is shown in Fig. 4.6.a (green pills)[102].

Proteus believes some patients can benefit more than others from the microchips, such as those that take medicines toward off tuberculosis, diabetics and elderly patients who have difficulty remembering to take their pills at the right time or if they have already taken them. Making things even more difficult, some have a whole list of pills with different schedules for each. The next logical step would of course be to allow the patient access to the data or to have it routed to a device set near where they keep their

medicine, which would both alert them when it is time to take their medicines and to let them know, if they cannot remember, if they took them or not. Some suggest this move by the Food and Drug Administration (FDA) is the first of many likely to come over the next few years, as other technologies are waiting in the wings. Coming soon may be swallowed or implanted devices that dose us automatically, sensors that report on other bodily activities, or devices that swim around in our bloodstream monitoring conditions and cleaning out plaque deposits. The output of smart pills will be similar to Fig. 4.6.c.

If the permission can be issued by the FDA, these devices can help a lot to the progress of cutting edge researches. Because, not only a vagus signal is send to the stomach muscles when VIMS occurs but enzymes are secreted by the abdominal glands too, which is easy to detect by a digital pill. A digital pill is capable to report the temperature, the pressure, the pH of the stomach.

4.4 Signal processing approaches

Very few unmanned 3D systems exist. Almost in all the available systems, the human being as an end-user is in one side and the machine performs specific tasks in the other side. The end-user convenience is one of the big concerns of a system designer in such a system. Although collecting appropriate feedbacks from a machine (alternatively we call it a process) is less complex than the user side, new technological developments, especially in EEG (Electroencephalography), MRI (Magnetic Resonance Imaging), EM (Eye Movement), Infrared (contactless body temperature measurement), EMG and others as mentioned above, make recording feedback from the end-user feasible. We need multiple feedbacks from both the process and the end-user simultaneously and synchronously in order to involve all the possible parameters in our study and analysis.

Multiple feedbacks from the process can be acquired by different sensors and measurement devices. These feedbacks can be analyzed consequently to understand the characteristics of the process and to study not only the effects of different parameters on the target variables but on user performance as well. These feedbacks are known as psychophysiological measurements as explained above, which provide quantitative values for the analysis, assessment and comparison. In addition, these analyses provide objective criteria for the amendment and modification of the system which in turn makes the design of better systems feasible.

This section will introduce different representations of measurement signals, stochastic and deterministic approaches for signal processing as well as mathematical tools that we will use to analyze and treat signals.

4.4.1 Signal representation in the time and frequency domains

We will use an EEG signal as an example to explain the signal representation in the time and frequency domains. An EEG signal is the spontaneous cortical electrical activity recorded at the scalp. The brain

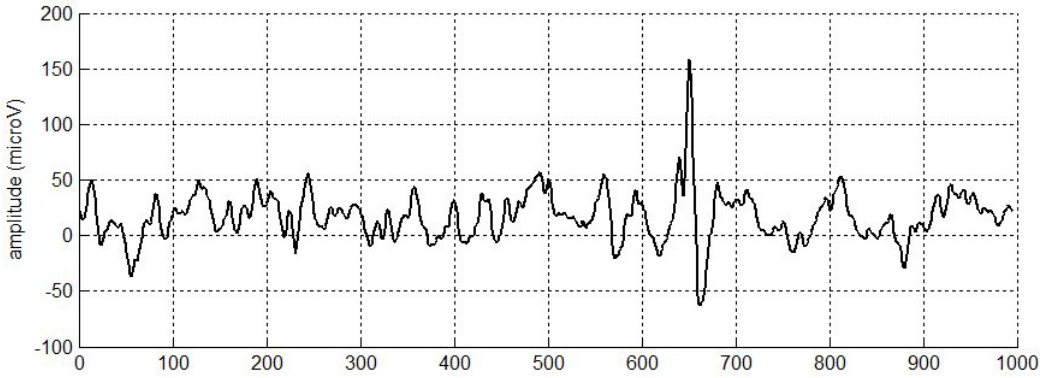


Figure 4.7: Single-channel EEG signal.

EEG was discovered by Berger in the 1800s using a primitive galvanometer with a surface electrode placed on his son's scalp and recorded a rhythmic pattern of electrical oscillations. This signal was the instantaneous electrophysiological response of cortical brain cells. Understanding an EEG signal is very important because later the same principle can be applied to EMG, EOG, and ECG but with different sensors.

Within depth inquiries during the last two decades, it is understood that the electrical potentials recorded as EEG are produced by electrical dipoles in the pyramidal cell layer. Many pyramidal cells and their dendrites are arranged vertically. This arrangement sets up a dendro-somatic "dipole" or potential which oscillates with the arrival of excitatory or inhibitory postsynaptic potentials. An EEG signal practically can easily be recorded by electrodes of some conductive metal (gold, silver/silver chloride and tin are popular choices) attached to the scalp in a bed of conductive gel.

A time domain representation of a signal can be continuous or discrete. We will deal with a discrete (sampled) signal in entire this thesis; the equations are extracted from continuous approaches with little modification. A single-channel EEG signal is shown in the time domain in Fig. 4.7. This signal is a time series with 1000 samples and the amplitude varies between -62.9 and 158.6 μV. Typical values of EEG is in the range of $20 - 100$ μV, however lesser values can be recorded in averaged evoked potentials: nearly 10 μV. Larger values are recorded in epilepsy and other disorders.

One of the most common ways to convert a signal from the time domain to the frequency domain is the Fourier transform. The Fourier transform of a given function $f(t)$, with limited integral, is called $F(\omega)$ and calculated by (4.14).

$$\int_{-\infty}^{+\infty} |f(t)|\,\mathrm{d}t < \infty \tag{4.13}$$

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega nt}\mathrm{d}t \tag{4.14}$$

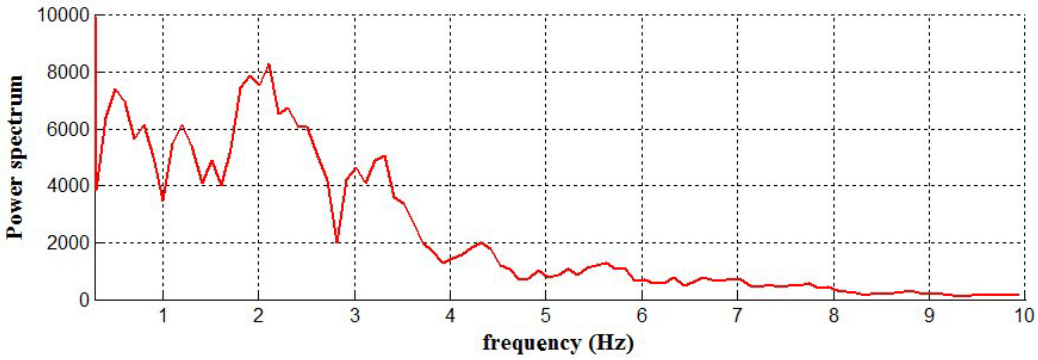


Figure 4.8: Single-channel EEG signal in the frequency domain.

For a discrete signal with a sequence of N samples, $x_0, x_1, x_2, \cdots, x_{N-1}$, the Discrete Fourier Transform (DFT) is defined by (4.15)

$$X_k(\omega) = \sum_{i=0}^{N-1} x_i e^{-j2\pi \frac{kn}{N}} \tag{4.15}$$

Notice that this new sequence X_k repeats after N terms which means the Fourier transform of a sample signal is periodic. However, we use the Fast Fourier Transform (FFT) to calculate the DFT of a signal in practice. The EEG signal of Fig. 4.7 is represented by a spectrum as shown in Fig. 4.8.

When the objective is to study a certain part of the brain, usually we use multi-channel EEG. The electrical activity of the brain behaves like any other material system changing in membrane polarization; inhibitory and excitatory post synaptic potentials, action potentials, etc., impress voltages are conducted through the surrounding medium of the CSF (Central Spinal Fluid), the meninges, the skull and the scalp. Usually, more than one electrode becomes active during brain activity. That is the reason why a multi-channel configuration of sensors is employed to record the brain activity in certain regions of the brain. An example of a high resolution EEG sensor arrangement, the spontaneous measurement of multiple electrodes, is shown in Fig. 4.9[172]. The electrical voltages conduct up through the brain tissue, enter the membranes surrounding the brain, continue on up through the skull to appear at the scalp. At this point they are reduced from mV range (of the membrane potential gradients and action potentials) to a few μV. A typical multi-channel EEG is shown in Fig. 4.10. As seen, certain parts of the brain under these electrodes became active during a specific period.

4.4.2 Instrumentation consideration during data acquisition

These signals can be processed and meaningful information can be taken out of them. The amplitude of a measurement signal is small compared to the amplitude of common artefacts (muscle, mains power



Figure 4.9: EEG probe location on the scalp for amplitude map extraction [172].

frequency radiation). The quality of the recorded signals depends on the scalp/electrode impedance, differential amplifiers, and filtering. Low impedance, a better amplifier and filtering devices and techniques will help to get cleaner signals. The position of the amplifiers is very important. The amplifiers are in the measuring box most of the time. In this case, the signals are affected by different disturbances such as power supply noise, the electrostatic charge of the body and so on. An electrically shielded recording laboratory can also help in this respect. However, in recently developed technologies, the amplifiers are mounted on the electrodes. These systems are highly secured against noise and any other ambient contamination.

The EEG signal is typically recorded by an "instrumentation" amplifier which uses a third "common" electrode to remove noise. The signals on this common channel are subtracted from each of the signal and reference channels. This removes the signals which are present on both the signal and reference electrodes (the common mode signal). A good amplifier has a high common mode rejection ratio - the ability to subtract out these common signals. The common mode rejection rate is measured in dB and should be more than 90 dB. Additional factors in the amplifier design include high input impedance. This factor determines the amount of current that can be driven. High input impedance leads to a low current and, consequently, even a high resistance scalp-electrode interface will be quite immune to picking up the environmental electrical noise which surrounds us. A kind of interfacing gel is used most of the time to reduce this impedance and increase the amplitude of the signal.



Figure 4.10: Example of a multi-channel EEG signal [172].

4.4.3 Classification of different events in a signal

One strong signal processing technique is the classification of an event in the signal based on the period of the event, statistical features such as the mean and standard deviation, the amplitude, the frequency spectrum or other features in the frequency domain. For instance, most EEG waves range from $0.5 - 500$ Hz. Alpha, beta, theta and delta waves are observed more than other waves and are called Essential Brain Activity Waveforms (EBAW). Fig. 4.11 shows these waves in the time domain (a) and their spectrum (b) in the frequency domain. Alpha and beta signals are differentiated by their amplitude and the period, while the same thing is difficult to apply to theta and delta signals. In this case, it is better to diagnose one from another by their spectrum in the frequency domain as shown in Fig. 4.11.b. However, other physiological characteristics as shown in Tables 4.1 and 4.2 can be used to identify each signal.

The most important features of these waves, EBAW, such as the bandwidth, the age dependency, the variation speed (slow or fast) and so on are detailed in Table 4.1.

Some phenomena can be recognized by their EEG wave forms. In fact, only by looking at the waveform we can say approximately what kind of event is happening. These waves are recognized by their shape and form and secondarily by their frequency. They include waves that may be normal in some settings and abnormal in others. Table 4.2 shows a summary of these waves and their characteristics (see Fig. 4.12).

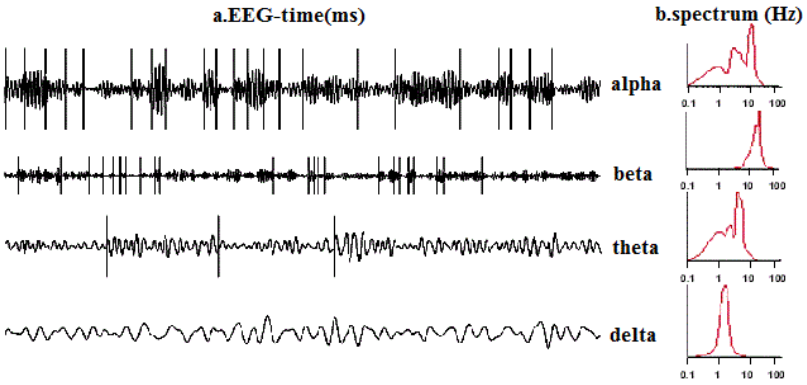


Figure 4.11: Sub-categories of an EEG signal.

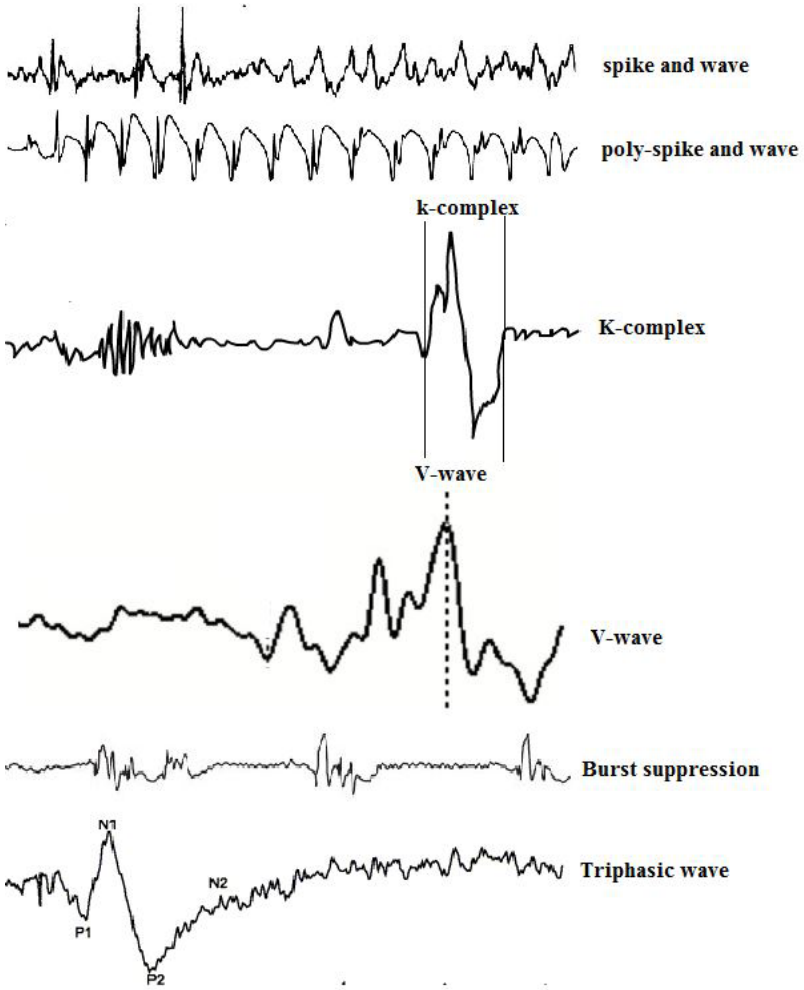


Figure 4.12: Few examples of wave patterns in an EEG signal.

Wave name	*Range (Hz)*	*Characteristics*
Alpha	$8.0 - 13.0$	adults
		closed eyes and with relaxation
		disappears with attention (mental arithmetic, stress, opening eyes)
Beta	$f \geq 13.0$	all age groups
		open eyes and anxieties
		small in amplitude, symmetric, and fast
		accentuated by sedative-hypnotic drugs (ex. benzodiazepines)
Theta	$3.5 - 7.5$	awake adults and children up to 13 years
		slow
Delta	$f \leq 3.0$	infants up to one year and in stages 3 and 4 of sleep (abnormal in an awake adult)
		high amplitude and slow (the slowest among the four waves)

Table 4.1: Four essential brain activity waveforms.

EEG signals are analyzed in different ways: Fourier time-frequency, 3D representation of time-frequency-amplitude, wavelet analysis and brain map. The best way of analyzing an EEG signal is multi-spectrum analysis. In this method, different resolutions of time and frequencies are applied to a signal with a pair of filters, a scale and a wavelet filter. An example of this multi-resolution analysis is shown in Fig. 4.13.

4.4.4 Wavelet multi-spectrum analysis

Since wavelet multi spectrum analysis is very important in signal processing, this part will provide a brief explanation of wavelet analysis. More information is available in the related literature (see Mallat[118] and Antonini et al.[3] publications).

Mathematically, a signal $x(t) \in L^2(\mathbb{R})$ can be decomposed into a linear combination of a set of n base functions $\{\phi_0, \phi_1, \cdots, \phi_n\}$ if the signal is in the space spanned by the basis. The linear combination of function $x(t)$ is,

$$x(t) = \sum_k a_k \phi_k(t), k \prec n (k \prec \infty), k \in \mathbb{Z} \tag{4.16}$$

where, k is an integer index of the finite or infinite sum and a_k, $\phi_k(t)$ are expansion coefficients and expansion functions respectively. Fourier ($\phi_k(t) = e^{\frac{2fktj}{T}}$) and wavelet ($\phi_{j,k}(t)$) are two well known

Wave name	*Range (Hz)*	*Feature*
Spike and wave	3 and 6	children
		large amplitude, slow wave (delta), generated in the thalamus
Poly spike and wave	$3.5 - 4.5$	with multiple spikes, slow, generated in the myoclonus
Lambda and posts		occurs in the second stage of sleep and awaken patient when the eyes stare at a blank surface
		triangular shape
K complex	$f \geq 13.0$	sleep when aroused (2 stages of sleep), sleep again or the awake state
		similar to beta wave
V-wave		seen in the second stage of sleep
		occurs in the parasagittal and bi-parietal
MU activity	$(1/2) * f$ of fast activity	shape of a wicket fence with sharp tips
		phase reversal between two channels
Psychomotor variant		rare rhythm
		higher in amplitude than the surrounding waves
		asymmetrical (14 & 6 rhythm)
14 & 6 rhythm	$6 - 14$	children and adolescents
		opposite/same direction with the wave
		in sleep and mono-polar recording
Periodic Lateralized Epileptic Discharges		acute brain injury or damage associated with severe focal signs and much illness
Tri-phasic wave		the wave was found with hepatic encephalopathy, subsequently been associated with many other metabolic encephalopathies
Burst suppression		slow and mixed waves often of high amplitude
		seen after a severe brain injury such as post-ischemia or post-anoxia

Wave name	*Range (Hz)*	*Feature*
Artifact		produced by technical or other disturbances
		solution between electrodes shorting them out
EKG and pulse artifact		recognized by their periodicity
		much larger electrical signal than EEG
		caused by an electrode moving
Pop		patient movement artifacts
		large in amplitude and last a long time by EEG standards
Electrode and other movement artifact		
Movement and 50 Hz artifact	50 or 60	periodic and small in amplitude and easy to recognize
		causes spikes at 50 or 60 per second
		60 Hz is seen with poor electrode contacts

Table 4.2: Morphology of different EEG waves.

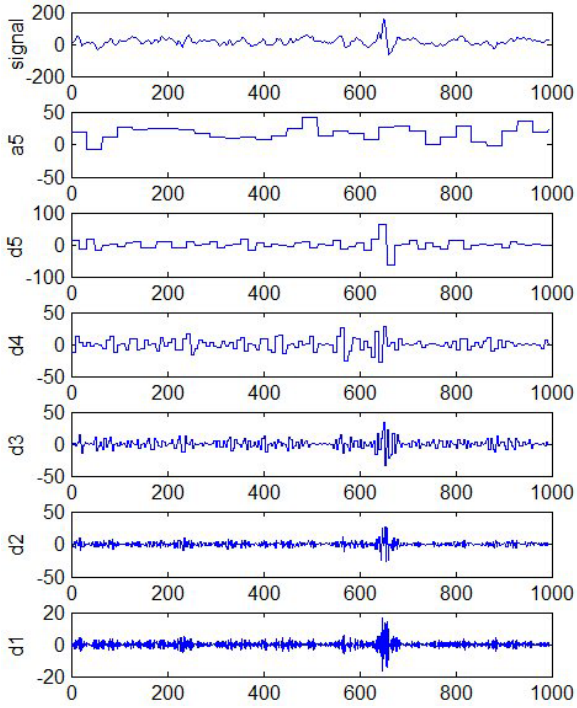


Figure 4.13: Multi-resolution analysis with wavelet.

examples of decomposition into primitive or fundamental constituents of their spaces. In fact, a Fourier series decomposes a signal into a set of sine and cosine functions. By a Discrete Wavelet Transform (DWT) in multi-resolution analysis, however, a signal is represented by a sum of a more flexible function called a wavelet that is localized both in time and frequency. In the DWT where the decompositions are based on a rate of power 2, which has been mostly used in practice base functions, $\phi_{j,k}(t)$ can be expressed by $2^{\frac{j}{2}}\varphi\left(2^j t-k\right), j \succ 0, k \in \mathbb{Z}$. In this decomposition, the signal $x(t)$ can be written as a linear combination of wavelet functions as follows:

$$x(t)=\sum_{j,k} a_{j,k}\phi_{j,k}(t) \tag{4.17}$$

$\phi_{j,k}(t)$ is called the mother wavelet or scale function (dual function), and the set $\{\phi_{j,k}(t), k \in \mathbb{Z}\}$ is an orthonormal basis for V_j, which can be expressed as: $V_j = \text{span}\{\phi_{j,k}(t), k \in \mathbb{Z}\}$. In definition, the subspace $V_j (j \in \mathbb{Z})$ should satisfy the following properties:

- $\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_{+1} \subset V_{+2} \subset \cdots$;
- $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}, \overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$;
- $f \in V_j$ if and only if $f(2\cdot) \in V_{j+2}$;

- $f \in V_0$ implies $f(\cdot - k) \in V_0$ for all $k \in \mathbb{Z}$.

If the properties above are met, there exists a set of functions $\psi_{j,k}(t)$, which is called the father wavelet, and the set $\{\psi_{j,k}(t), k \in \mathbb{Z}\}$ is an orthonormal basis for W_j.

The inner product is

$$\langle \psi_{j,k}(t), \phi_{j,k}(t) \rangle = \int \psi_{j,k}(t) \phi^*_{j,k}(t) \mathrm{d}t = \delta_{i,j} \tag{4.18}$$

With this orthonormal property, we can find the coefficients by

$$\begin{aligned} \langle x(t), \psi_{j,k}(t) \rangle &= \int x(t) \psi^*_{j,k}(t) \mathrm{d}t = \int \left(\sum_{k'} a_{j,k'} \phi'_{j,k'}(t) \right) \psi^*_{j,k}(t) \mathrm{d}t \\ &= \sum_{k'} a_{j,k'} \left(\int \phi'_{j,k'}(t) \psi^*_{j,k}(t) \mathrm{d}t \right) = \sum_{k'} a_{j,k'} \delta_{k'k} = a_{j,k} \end{aligned} \tag{4.19}$$

rewritten as follows:

$$a_{j,k} = \langle x(t), \psi_{j,k}(t) \rangle \int x(t) \psi^*_{j,k}(t) \mathrm{d}t \tag{4.20}$$

The satisfying result comes because of the orthonormal property of the basis. Note that W_j is the orthogonal complement of spaces V_j and V_{j+1}, which can be expressed as:

$$V_{j+1} = V_j \oplus W_j \tag{4.21}$$

In general,

$$V_j = V_{j_0} \oplus W_{j_0} \oplus \cdots \oplus W_{j-1}, j \succ j_0 \tag{4.22}$$

and

$$L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus W_3 \cdots \tag{4.23}$$

It is important to choose an appropriate set of basis and its dual for the signal decomposition. For the signal we want to deal with, if a particular basis satisfying the orthonormal property is applied, it will be easy to find the expansion coefficients a_k. Fortunately, the coefficients concentrate on some critical values, while others are close to zero. We can drop the small coefficients and record the important values. Various sources of noise usually inject various types of contamination into a signal and disturb the recording pro-

cess. Therefore, the polluted signal needs to be cleaned up by a de-noising filter. Some information of the signal appearing in high frequencies can be mixed up with noise.

4.4.5 Time-frequency analysis

Fourier time-frequency analysis has been a powerful tool to extract features from a signal or see some hidden characteristics of a signal. A simultaneous demonstration of time-frequency of a signal can illustrate the real capability of this analysis technique.

Given the width and the definition of the window function $w(t)$, we initially require the area of the window function to be scaled so that

$$\int_{-\infty}^{+\infty} w(\tau)\mathrm{d}\tau = 1 \tag{4.24}$$

It easily follows that

$$\int_{-\infty}^{+\infty} w(t-\tau)\mathrm{d}\tau = 1 \ \forall t \tag{4.25}$$

and

$$x(t) = x(t)\int_{-\infty}^{+\infty} w(t-\tau)\mathrm{d}\tau = \int_{-\infty}^{+\infty} x(t)w(t-\tau)\mathrm{d}\tau \tag{4.26}$$

The continuous Fourier Transform is

$$X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t}\mathrm{d}t \tag{4.27}$$

Substituting $x(t)$ from above

$$X(\omega) = \int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty} x(t)w(t-\tau)\mathrm{d}\tau\right]e^{-j\omega t}\mathrm{d}t = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-j\omega t}\mathrm{d}\tau\mathrm{d}t \tag{4.28}$$

Swapping the order of integration

$$\begin{aligned} X(\omega) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-j\omega t}\mathrm{d}t\mathrm{d}\tau = \int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-j\omega t}\mathrm{d}t\right]\mathrm{d}\tau \\ &= \int_{-\infty}^{+\infty} X(\tau,\omega)\mathrm{d}\tau \end{aligned} \tag{4.29}$$

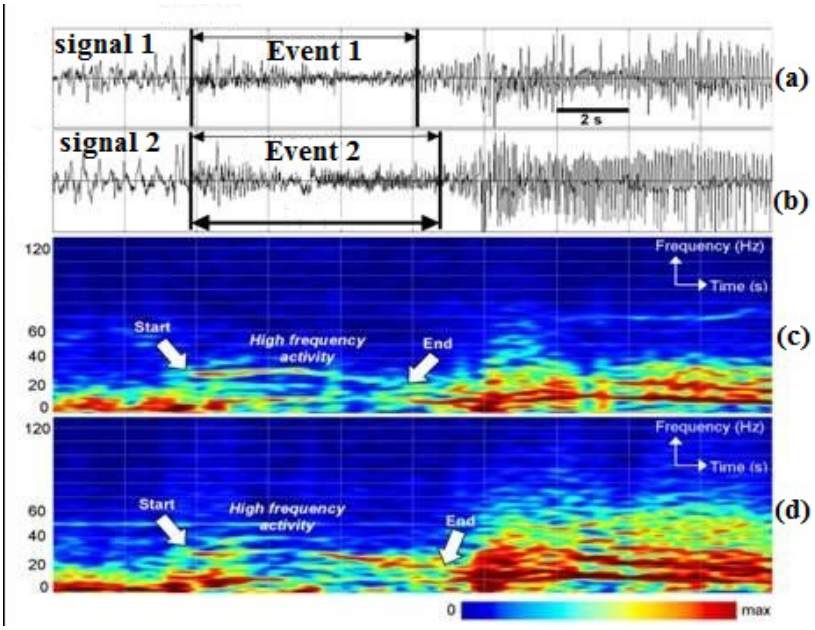


Figure 4.14: Time-frequency EEG signal analysis.

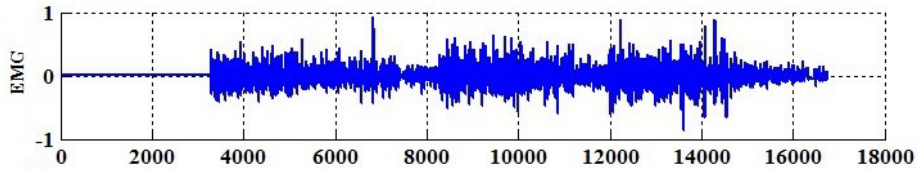


Figure 4.15: Example of an EMG signal.

The time-frequency representation can be calculated by

$$X(\tau, \omega) = \int_{-\infty}^{+\infty} x(t) w(t - \tau) e^{-j\omega t} \mathrm{d}t \tag{4.30}$$

Fig. 4.14 shows two different signals (a, b) in the time domain which contain two events (Event 1, Event 2) with different lengths and their corresponding time-frequency representations (c, d). Only the length of the event can be well seen from the time domain representation, not the detail. The time-frequency representation shows more detail.

For example, in general, signal 1 contains frequency components up to 60 Hz while the frequency components of signal 2 approach beyond 60 Hz and reaches to nearly 90 Hz. Besides high frequency components were active approximately from the beginning till the end of the measurements.

The starting point and the stop point of both events were marked up in Fig. 4.14.c and d.

Fig. 4.16 shows another example of time-frequency analysis. This time, an EMG signal and its time-

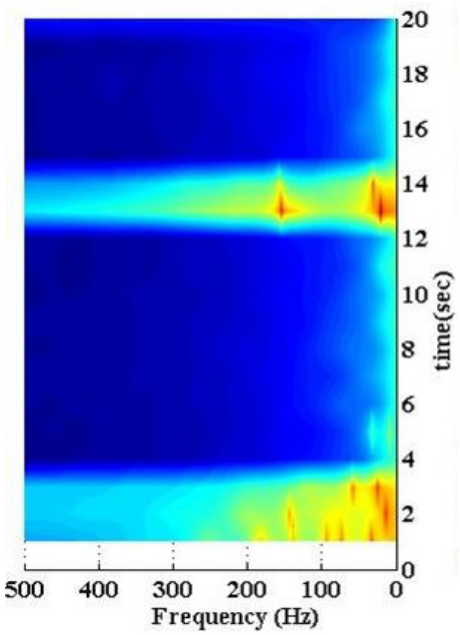


Figure 4.16: Time-frequency representation of the EMG signal shown in Fig. 4.15.

frequency will be discussed. Fig. 4.15 shows that three events occurred with the same length and approximately the same amplitude. There are high and low components in the signal that hardly can be differentiated. On the other hand, time-frequency shows a very clear image of the signal, shown in Fig. 4.16. Three events occurred during this EMG activity and three events are approximately identical. This is the only information that can be understood from the time domain, however more detail needs to be extracted from this signal to better identify each event from the other. The time-frequency representation perfectly does this and separates the content of each event into components with long (14 s) and short ($1 - 3$ s) durations. Besides, the long period component has three low frequency components, 15 Hz, 20 Hz and 150 Hz. Maximum 300 Hz appears in the signal.

In general, the time and frequency domains provide very useful information about the characteristics of a signal. The time-frequency representation is extracted only when the time and frequency domains cannot provide enough information.

Equation (4.29) is one of the ways to generate a time-frequency representation. However, as seen in section 4.4.4, a wavelet can easily provide us a time-frequency analysis with (4.31) as well if the mother wavelet function is appropriately chosen.

$$C(t,s) = \frac{1}{\sqrt{s}} \int x(\tau) w\left(\frac{t-\tau}{s}\right) \mathrm{d}\tau \tag{4.31}$$

If the function is selected, the so-called Morlet mother wavelet (4.31) then (4.32) will be frequency dependent and the result will be a time-frequency representation.

$$w(t) = \frac{1}{k\sigma} e^{-(\sigma t)^2} \cos(2\pi f t) \tag{4.32}$$

4.4.6 3D representation of time-frequency data

Two types of signals, the real signal and noise are really difficult to separate even with very sophisticated adaptive filters. To avoid bad election, it is better to apply a multi resolution analysis to be able to recognize the real signal values from the noise as explained in section 4.4.4. A 3D representation of a signal can effectively help us to see this difference and select the right signal and suppress the noise.

Fig. 4.17.b shows the 3D representation of a signal. As seen the signal has two distinct components. The first part looks quite regular and periodic and the second part has less regularity but still keeps its periodicity. Although this representation provides a very useful insight into the signal, it can help to do better analyses only in case we do not have any idea about the content of the signal. Most of the time, a 3D representation is counted as a preprocessing step for time-frequency analysis and processing in the frequency domain. If we are dealing with multiple channel data acquisition, for instance high resolution EEG, it is better to use another representation called mapping. Mapping is transforming from one space to another space.

4.4.7 Mapping into a new space

One of the best ways of multiple channel signal analysis, for instance EEG, in the case of a clean measurement (noise free) is the brain map. This representation is shown in Fig. 4.17.a. As seen, it is very easy to see which part of the brain is active at a certain moment. This representation is a mapping of sensors data similar to that of Fig. 4.17 at time t into a 2D space. The red and blue colors illustrate the regions with severe brain activities and the inactive regions respectively. For instance, it has been understood from the brain map shown in Fig. 4.17.a that the frontal lobe of the brain has experienced severe activities from $t = 0.78$ s to $t = 1.14$ s while at the same time the occipital lobe was totally inactive. In turn, it can be deduced that the participant's body was in motion during this measurement.

4.4.8 Signal fusion

Electroencephalography (EEG) is one of the best methods for brain dynamics monitoring because of its high temporal resolution and portability. However, using only an EEG signal is not enough to track down the activity of a specific region and associate a function with the region. Usually, an extra sensor data such

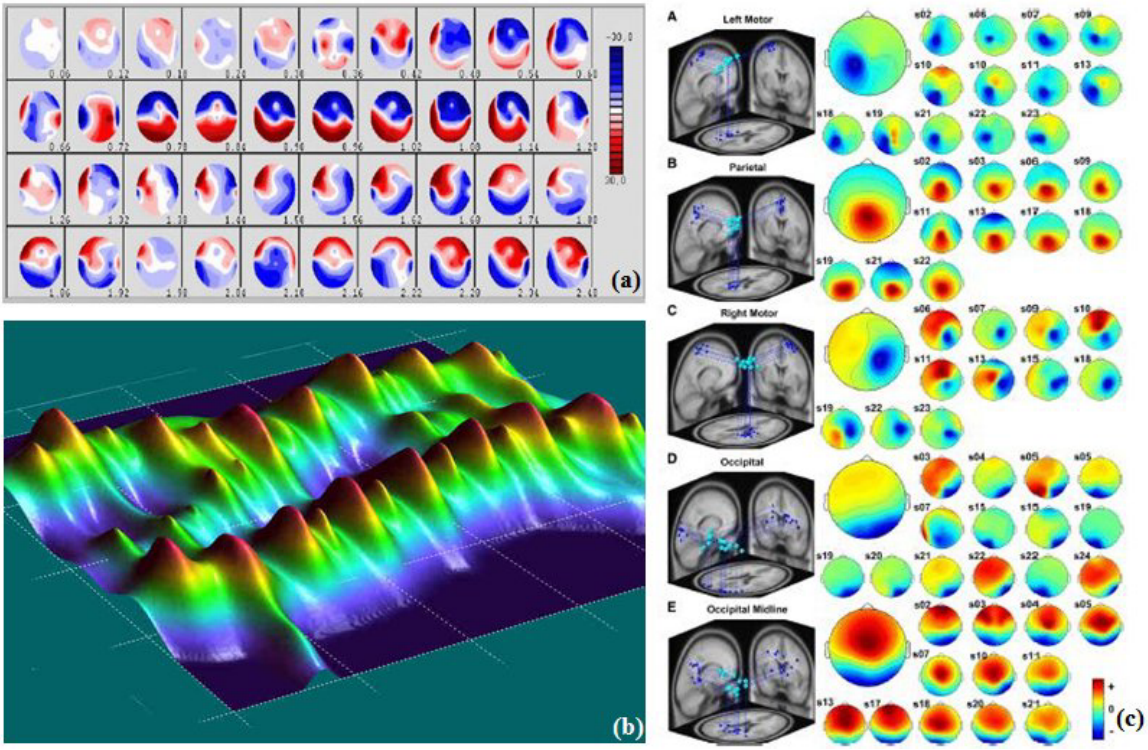


Figure 4.17: (a) Brain EEG amplitude map, (b) 3D representation of a multi-channel EEG signal, (c) equivalent dipole source locations and mean, and individual scalp maps for five IC clusters, 3D dipole source locations and their projections onto average[26].

as MRI is required to do the analysis more accurately. In such a case, we need to fuse and combine the data from two different sensors to get more precise results and find closer answers to our question. For instance, Chen et al.[26] proposed a method based on simultaneous vestibular and ocular signal measurements for motion sickness evaluation. The analysis has been done by ICA (see below) and time-frequency analysis. The result of these analyses is shown in Fig. 4.17.c.

The right side shows the 3D representation of the brain and the left side is the brain map. A transformation is used to transfer data from 2D to 3D. This combined representation will help better understand where the source of activity is. However, finding a mapping transformation is not always easy. Fig. 4.17.c is an example of an MS study on a car motion platform (car simulator). Fig. 4.17.c compares the mean component power spectra of the IC clusters under different motion sickness levels and various road conditions. Evidently, the alpha powers of the right, left motors and the parietal components were suppressed from a straight-road driving to a winding-road driving as the car swayed from side to side. This finding suggests that these brain networks may be highly responsive to the movements of the platform and hence to somatic sensation. Further, the occipital midline IC cluster exhibited significant alpha power suppression.

4.4.9 Independent component analysis (ICA)

When the statistical independence assumption is correct, a blind ICA separation of a mixed signal gives very good results[43]. It is also used for signals that are not supposed to be generated by a mixing for analysis purposes. A simple application of ICA is the "cocktail party problem", where the underlying speech signals are separated from a sample data consisting of people talking simultaneously in a room. Usually the problem is simplified by assuming no time delays or echoes.

4.5 Conclusion

In this chapter, we have reviewed different psycho-physiological methods to measure biofeedback from a user. Some of the methods described require specific tools and materials that can be very expensive and hard to use. We will see in the following chapters how we will use some of these methods to measure VIMS in an interaction (navigation/manipulation) task.

5

Usability of manipulation interfaces in a real vs virtual environment

Different navigation and manipulation interfaces were introduced in chapter 3. These interfaces facilitate the navigation and movement in a virtual environment. This chapter will provide a set of heuristic manipulation interfaces to interact and modify different elements of a large-scale 3D scene. An end user is fully immersed in a virtual environment during manipulation; that is the reason why the proposed manipulation interfaces proposed in this chapter are important. The feedback is taken from the system side, including graphic processors, visualization and other devices. In addition, psychophysiological data are recorded from the user during immersion, as detailed in chapter 4.

The effects of the manipulation parameters including the interface modality, the effect of presence on user performance in the absence of some sensory information will be studied in this chapter. Besides, manipulation in a real and virtual environments will be compared.

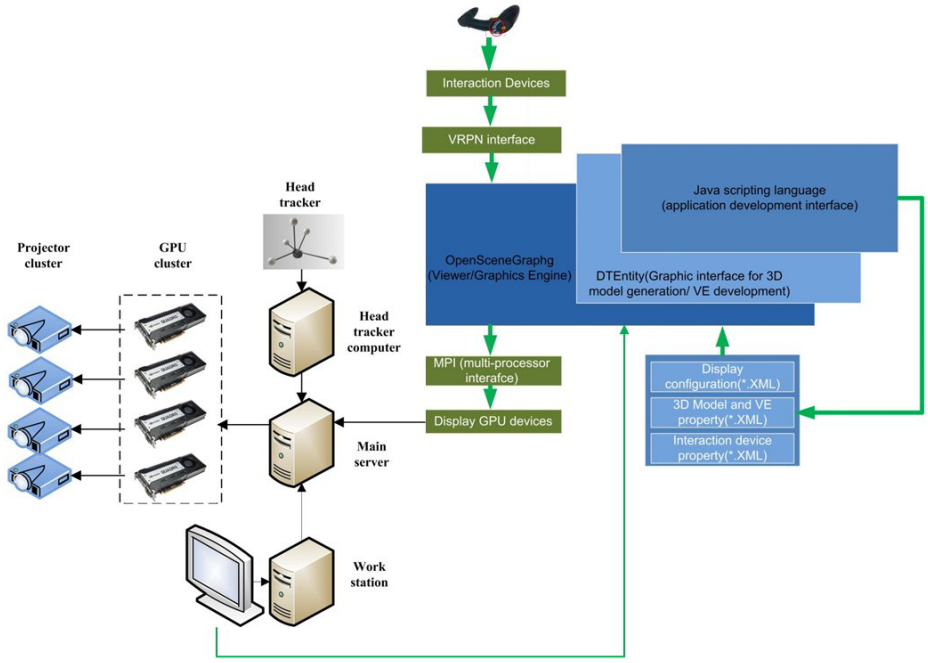


Figure 5.1: Visualization hardware architecture and software platform designed for the experiments.

5.1 Apparatus and test setup

A test setup was established to perform the experiments, as shown in Fig. 5.1. The test-bench consists of a multi-screen scale-one 3D display system, a navigation device and its interfacing system based on a VRPN server, a LAN network, a wireless and Ethernet router, a LAN Hub and a VE workstation computer along with its accessories. Different versions of scale-one 3D display systems were proposed and experimented in the literature[17]. A CAVE system is used to perform the experiments and the subjective evaluation test setups. The CAVE system was already presented in chapter 3. Two images are generated for each wall and are sent to a GPU cluster via Multi-Processor Interface (MPI) for display and projection. Based on the software platform presented in chapter 3, the visualization hardware architecture and software platform designed for the experiment is shown in Fig. 5.1.

A second joint platform was developed on a Java Virtual Machine (JVM) platform using MATLAB engine C++ library. The aim of the platform was to import test data from the experiments via VRPN and easier and faster using MATLAB scripting language, Simulink and processing toolboxes. All the sensors used in the experiments are connected to the JVM platform as psychophysiological measurement as shown in Fig. 5.2 (right). The iiVR and JVM platforms are able to communicate with each other through wireless/LAN network and TCP/IP protocol (Fig. 5.2 (left)). The JVM platform is capable to analyze measurements in real-time synchronous with the display system and send the command to the motion

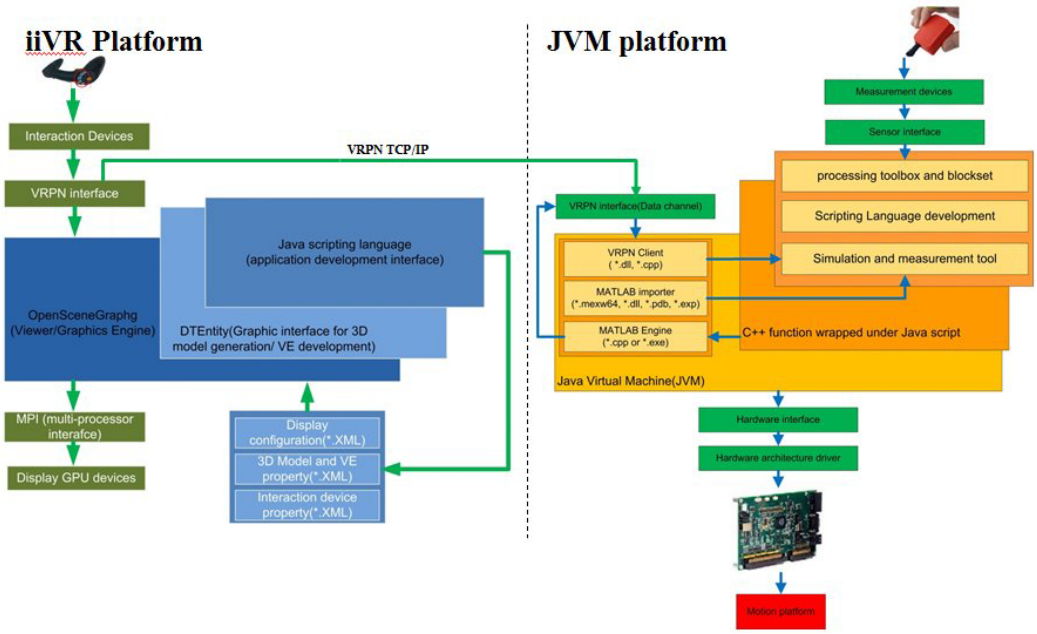


Figure 5.2: iiVR and JVM joint platform for real-time synchronous experiments.

platform.

The participants are capable of moving forward/backward, rotating to the left/right up to 360°, if it is required, using the Fly-stick as navigation/manipulation device in the validation experiment. It is possible to fly along a specified direction or along x, y, and z axes which means complete unconstrained movements in 3D space. The Fly-stick was presented in chapter 3.

5.2 Comparison between virtual and real object manipulations

Object manipulation has attracted special attention among other interaction tasks especially in virtual environments and mixed real-virtual environments. Obviously, manipulation with an object depends on the environmental situation such as the lighting, the presence of other objects, the location of the objects in the scene and so on. There is a huge difference between object manipulation in the virtual and real environments because of missing sensory feedbacks. For instance, sensory feedback such as gravity, roughness, the pressure, the temperature and the sensation associated with materials are missing during the interaction with virtual objects which in turn leads to this difference.

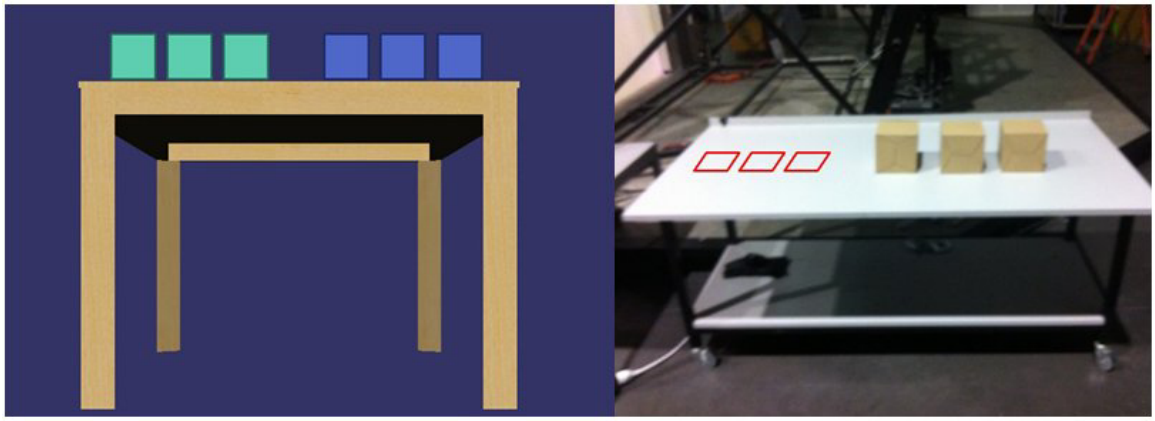


Figure 5.3: Object manipulation in (a) a virtual and (b) a real environments.

5.2.1 Experiment design

Two test-benches were set up in both real and virtual environments to study the effect of these above-mentioned missing feedbacks on the human perception during an object manipulation task in scale-one immersion (Fig. 5.3). As shown in Fig. 5.3.a, the object manipulation task consists of replacing three blue cubes into three green cubes using a Fly-stick in the virtual environment. The object manipulation task has three sub-tasks, i.e., selection, displacement and positioning. Selection is performed by pointing the Fly-stick at the center of the target cubes and pushing button 1 (see Fig 3.22 of chapter 3 for Fly-stick detail) when the Fly-stick is enough close to a target cube ($d \leq 20$ cm, d represents the distance between the Fly-stick and the cube). If $d \leq 20$ cm, then by pushing the Fly-stick button the selected cube will be attached to the Fly-stick. During translation, the participant will keep pushing the button if he aims at relocating the selected cube. The blue cube will be carefully placed inside the marker cube (three left cubes) during the positing sub-task. The right blue cubes should be placed inside the marker cubes (the first cube inside the first marker) in order. The same task is performed by hand in the real environment (Fig. 5.3.b). The participant grabs the first cube ($15 \times 15 \times 15$ cm^3 and an approximate weight of 1 kg), holds it up and places it inside the first red square marked up on the left side of the table (Fig. 5.3.b), the same task is repeated for each cube. The test procedure is detailed below step by step. The presence questionnaire [230] and electromyography [10] (EMG) are used as psychological and psychophysiological measurements respectively in this study.

5.2.2 Test procedure

1. Three Bio-Pack sensors are placed on the participant's right arm (see section 5.2.3, sensor positioning).

2. The participants were asked to lift off three cubes one by one from the right side of a white table

(Fig. 5.3.b), move in a semi-circular path and place them inside three red squares on the left side of the table.

3. The participants use the red marker to adjust each cube in the exact position.
4. An EMG signal is recorded during the experiment (Fig. 5.4).
5. The same setup is established in the virtual environment by placing a virtual table with exactly the same size in the CAVE and a set of cubes and markers are assigned for object manipulation (blue and green). The blue cubes represent the target objects which will be manipulated and the green cubes will be used as markers to help the user to place each cube in the exact position.
6. The position of the object will be tracked by an AR-tracker system consisting of four infra-red high precision cameras.
7. An EMG signal is recorded in the virtual environment during the experiment too.
8. The time of the task completion is recorded in both environments.

5.2.3 EMG Sensor positioning

A three-lead wireless Biopack EMG sensor[10] is used in this experiment (+, -, GR). The sensor is connected to an amplifier and a A/D stand (Fig. 5.4) via wireless and from there, data logging becomes feasible by a PC via a network connection. The Wireless EMG BioNomadix (Fig. 5.4)[9] consists of a matched receiver module. The module is specifically designed to measure EMG data and provide a fully-wireless recording connection from the computer to the subject, with all the benefits of a wired system. EMG signal data is transmitted at a rate of 2, 000 Hz, providing an extremely high resolution wireless EMG waveform at the receiver's output. Raw data from the sensor is band-limited from 5.0 Hz to 500 Hz, to provide a very high quality recording.

The position of the sensors was selected so that we can have a stable connection during the experiment. Different muscles of the arm were selected and tested using MSO9200 double-probe scope of LinkInstrument[LinkInstruments] to find the most stable muscles for the sensor position and to record an EMG signal with higher amplitude. A precise measurement test setup was established as shown in Fig. 5.5. The data is recorded for biceps muscles as shown in Fig. 5.7. The arm keeps opened (a), semi-opened (b) and closed (c) as seen in Fig. 5.6 (left). The signals associated with these three positions were recorded and analyzed in real-time both in the frequency and time domains as shown in Fig. 5.6 (right). As shown, in the frequency domain, the difference apparently is not noticeable, while in the time domain, the difference between opened and closed arm is noticeable.

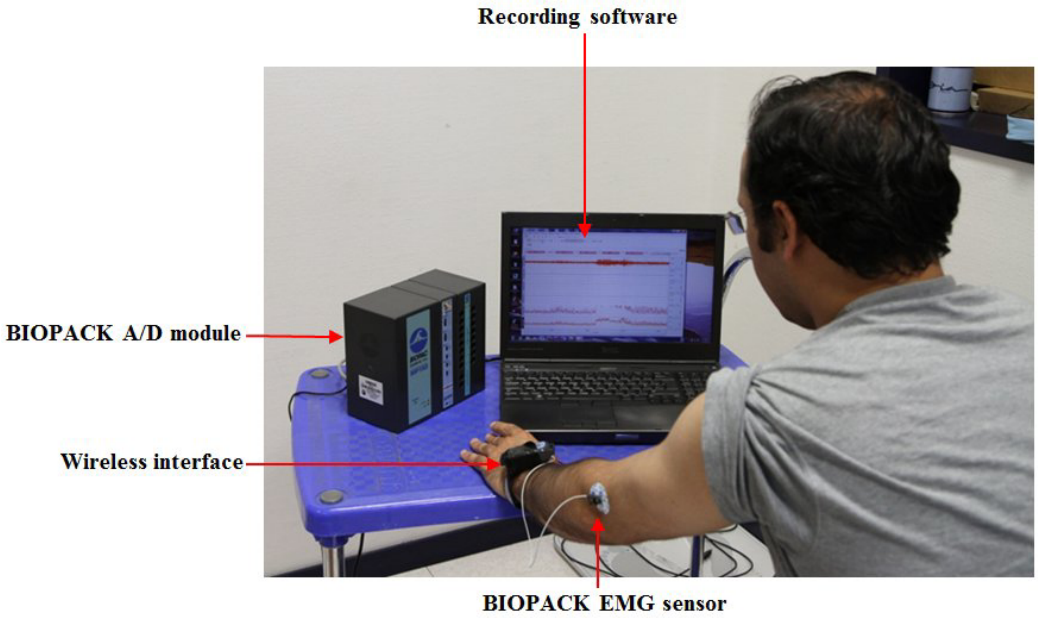


Figure 5.4: Wireless EMG sensor, measuring and data logging stand.

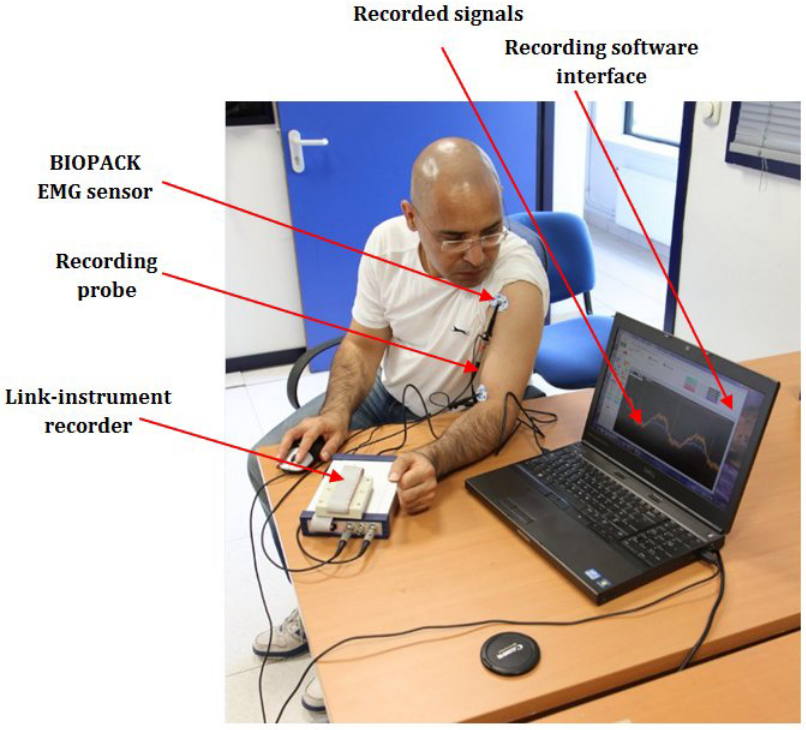


Figure 5.5: Precise measurements setup for sensor positioning.

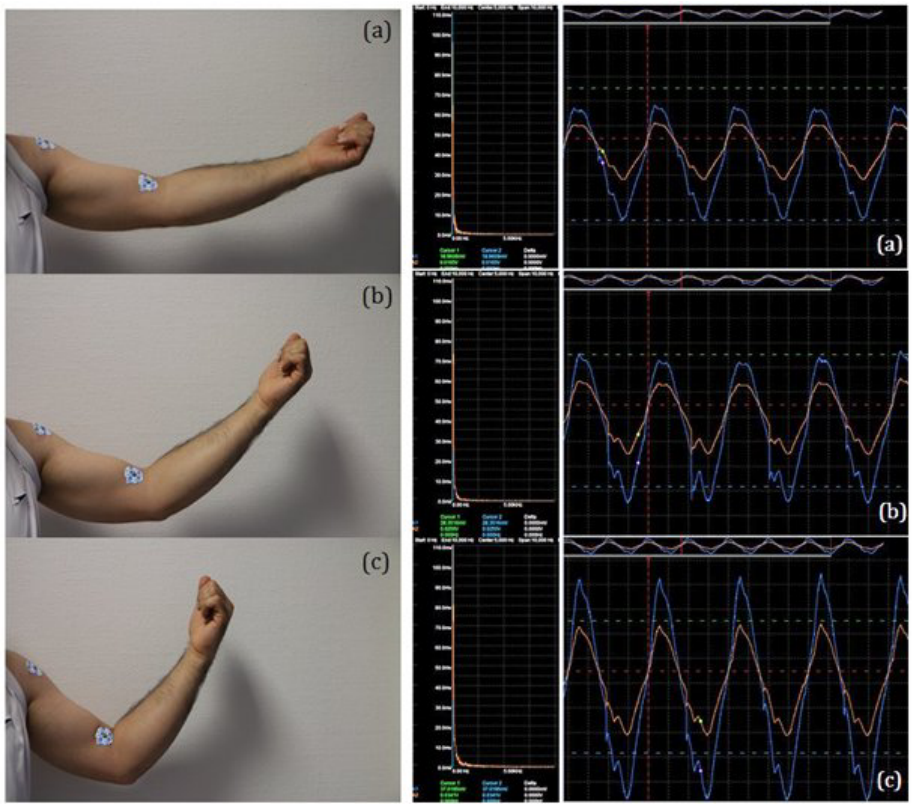


Figure 5.6: Three arm positions (left), data recorded by the BIOPACK sensor in the time and frequency domains (right).

Not only the signal variation should be noticeable but also practically the sensor position during the measurement regarding the body temperature variation and movement tension needs to be completely stable and solid. Finally by taking into consideration all these criteria, the biceps muscle (Fig. 5.7.a) was identified as an appropriate place for EMG signal logging. As seen in Fig. 5.7.a, the positive (+) and negative (-) leads were placed at the origin and end of the biceps muscle respectively and the ground lead (GR) right on the elbow. We have noticed by experiment that biceps muscles are more involved in object manipulation too. It means, five subjects did the trial run experiments and the data was analyzed. The results have shown that this sensor placement can give us robust and meaningful results.

5.2.4 EMG sensor data analysis

An example of a logged signal is shown in Fig. 5.8. The muscle activity during the placement of each cube can be easily recognized in the real environment while it is not the same for the virtual environment. This is because in the real environment complete feedbacks are provided by the sensory organs for the CNS, consequently the human brain can recall the appropriate pattern and generate more accurate motor

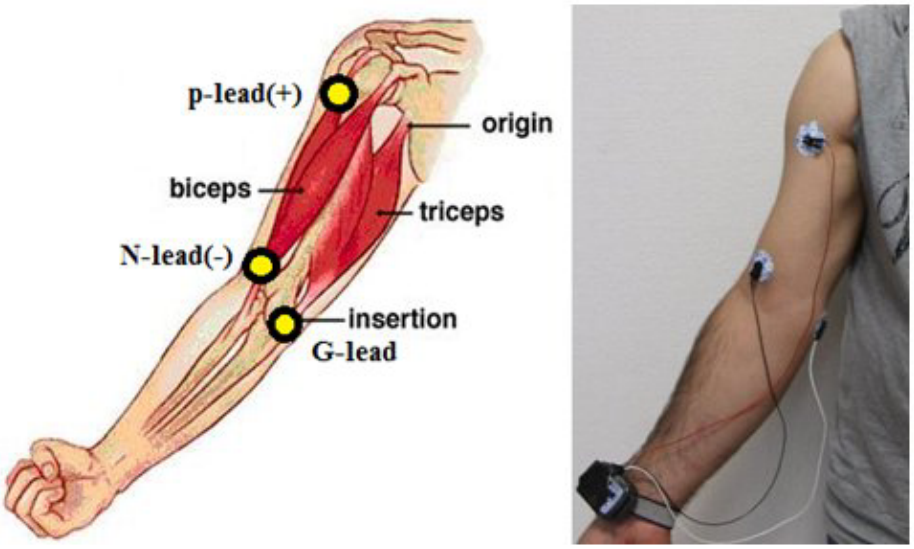


Figure 5.7: The biceps muscle is used for EMG data logging (a), EMG sensor position for measurement (b).

commands for the muscles. In the virtual environment, some sensory information is missing; as a result the brain cannot extract the appropriate pattern and therefore, generate not very accurate motor commands for the muscles. Later, it will be shown that this pattern can be easily seen in the time-frequency responses. As shown in Fig. 5.8 (first signal), three activities can be easily distinguished in the time space (reality) while in the virtual environment the activities (Fig. 5.8, third signal) are not easily distinguishable, although the tasks in both environments are approximately the same.

5.2.5 Object manipulation evaluation criteria

We will propose three criteria based on the evaluation parameters that have already been reviewed in chapter 3 (see section 4.2.1). The length of the movement, φ_1, the amount of rotation, φ_2, and the completion time, t, are considered as time space features for the comparison between the real and virtual environments (5.1).

$$\text{criteria} = \{\varphi_1, \varphi_2, t\} \tag{5.1}$$

Criteria 1: the length of the object movement trajectory (φ_1)

Fig. 5.10 shows the average path of the movement for each cube in the real environment (red dot-line) and the virtual environment (solid-line). The first feature that will be discussed here is the length of the movement. The movement in the real environment is considered as a semi-circular path with an average

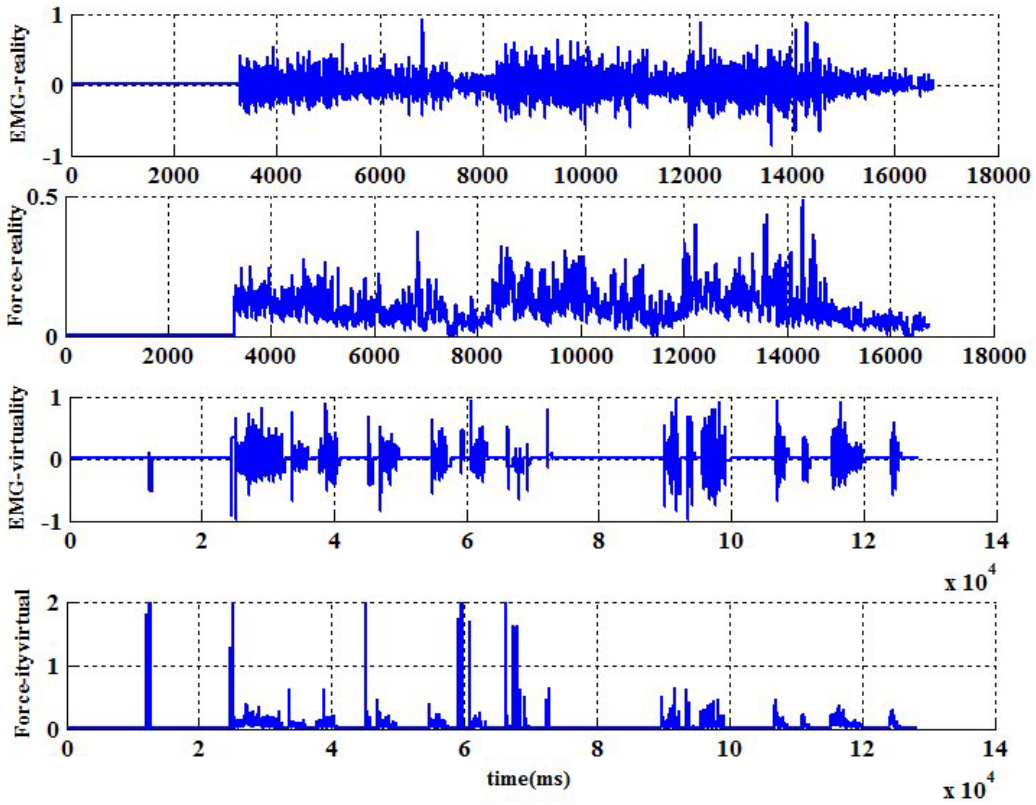


Figure 5.8: EMG signals recorded in the real/virtual environments.

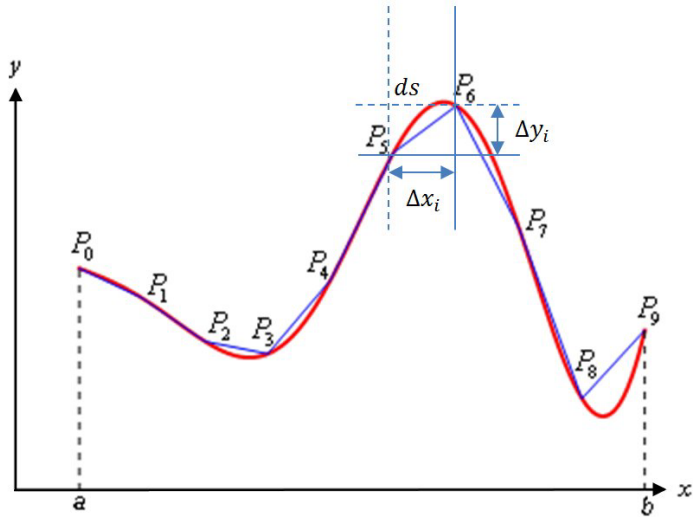


Figure 5.9: Calculation of the length for a given curve.

radius ($\bar{r}$) and an average length $\pi\bar{r}$. Theoretically, the length of a given continuous curve similar to the one shown in Fig. 5.9 is calculated by (5.2) in 2D and (5.3) in 3D space.

$$L = \int_a^b \mathrm{d}s = \int_a^b \sqrt{\mathrm{d}x^2 + \mathrm{d}y^2} = \int_a^b \left(\sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2} \right) \mathrm{d}t \tag{5.2}$$

$$L = \int_a^b \left(\sqrt{\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}y}{\mathrm{d}t}\right)^2 + \left(\frac{\mathrm{d}z}{\mathrm{d}t}\right)^2} \right) \mathrm{d}t \tag{5.3}$$

In 3D space, (5.3) can be easily substituted by (5.4) for a curve with sampled data with $\Delta t = 1$.

$$L = \sum_{i=1}^{n-1} \left((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2 \right)^{\frac{1}{2}} \tag{5.4}$$

where, n indicates the number of points along the path.

Criteria 2: the amount of rotation (φ_2)

The amount of rotation along the circular path in YZ plane is defined by (5.5) as specified in [201].

$$\nabla \times F = \oint_0^{2\pi} \vec{F}(x, y, z)\mathrm{d}\vec{r} \tag{5.5}$$

Since the movement is semi-circular, (5.6) and (5.7) are used for the rotation calculation for a continuous and a discrete functions respectively.

$$\nabla \times F = \int_0^{\pi} \vec{F}(x, y, z)\mathrm{d}\vec{r} \tag{5.6}$$

$$\nabla \times F = \sum_{i=1}^{n-1} \left(\frac{-(x_{i+1} - x_i)\cos(\theta)\cos(\phi) - (y_{i+1} - y_i)\cos(\theta)\sin(\phi) + (z_{i+1} - z_i)\sin(\theta)}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}} \right) \tag{5.7}$$

where, θ and ϕ represent the parameters of the sphere coordinates and F is the movement trajectory. Triple x_i, y_i, z_i demonstrates the coordinate of the ith point, P, on curve F. We assume, the reference semi-circular path lies on YZ plane, $x = 0$, therefore $\phi = \frac{\pi}{2}$. Moreover, the resolution of an angle θ is chosen $\frac{\pi}{n}$ for simplicity. As a result, (5.7) can be rewritten as (5.8). Using (5.8) the total rotation along the

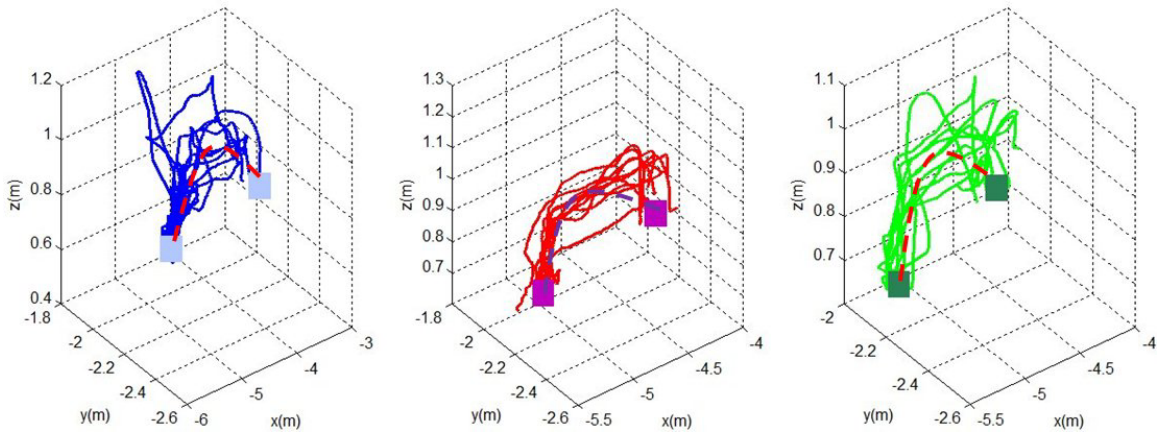


Figure 5.10: Spatial representation of an object displacement in the virtual and real environments.

path can be calculated.

$$\nabla \times F = \sum_{i=1}^{n-1} \left(\frac{-\left(y_{i+1} - y_i\right) \cos\left(i\frac{\pi}{n}\right) + \left(z_{i+1} - z_i\right) \sin\left(i\frac{\pi}{n}\right)}{\sqrt{\left(x_{i+1} - x_i\right)^2 + \left(y_{i+1} - y_i\right)^2 + \left(z_{i+1} - z_i\right)^2}} \right) \tag{5.8}$$

5.2.6 Data analysis in the time domain

Recorded sensor data will be analyzed in the time and frequency domains. Recorded data will be analyzed in the time domain first and then in the frequency domain.

Length of the movement trajectory

By applying (5.4) the length of the movement can be calculated for each cube. The results of this calculation are shown in Fig. 5.11 for each participant. The length of the movement in an environment for a given task (selection, movement and placement of the first cube) is a criterion to assess the task in different environments. Besides, it can be used to compare a manipulation mechanism or to compare the same task in two different environments. As a result, the longer the length of the movement, the harder the task is.

We found a significant difference between the length of the average movement in the virtual ($M = 2.48$, $SD = 0.53$) and the real ($M = 1.58$, $SD = 0.13$) environments, $p < 0.001$ ($p = 0.039$) and $t(9) = 5.41$ as shown in Fig. 5.11. An ANOVA test shows that neither in the real nor in the virtual environment the difference of the movement among the three cubes is significant, $p = 0.141$, $p = 0.243$. We did an ANOVA test to be sure that the difference between each cube displacement with the next one is not significant in both environments.

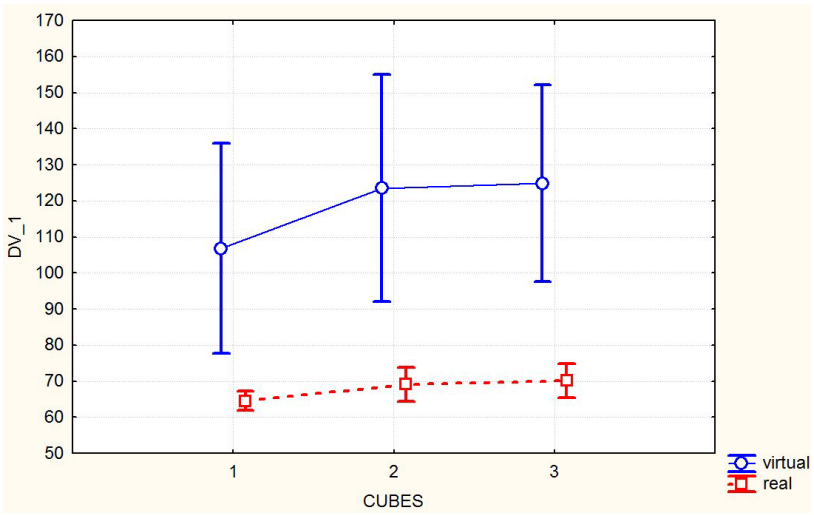


Figure 5.11: Length of the movement trajectory in the virtual environment (blue) for the three cubes with the order shown in Fig. 5.10.

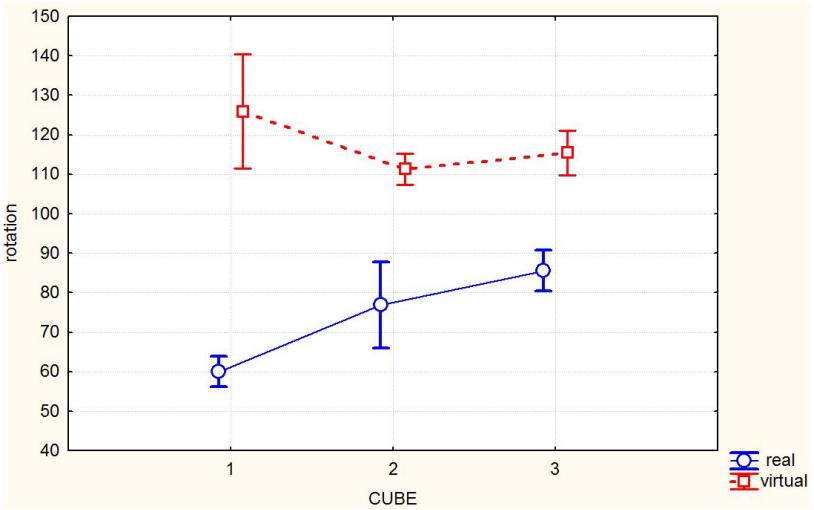


Figure 5.12: Amount of rotation for each cube in the real and virtual environments.

Amount of rotation

The amount of total rotation can be calculated easily by (5.8). Constant n in (5.8) represents the number of points on F. The amount of total rotation during the experiment for each subject is shown in Fig. 5.12. A comparison of the amount of rotation during the manipulation of the cubes in the two different environments shows that in the virtual environment this value is significantly $p < 0.001$ ($p = 0.0035$) higher than in reality.

Time of completion

Fig. 5.13 shows the time taken to complete the manipulation task both in the virtual (black) and the real (gray) environments. We found a significant difference, $p < 0.005$ ($p = 0.0026$, $t(9) = 3.62$), between the time of completion in the virtual ($M = 79.78$ s, $SD = 43.20$ s) and the real ($M = 24.16$ s, $SD = 5.095$ s) environments. Therefore, the participants spent more time in the virtual environment ($M_v = 79.78 > M_r = 24.16$) to complete the manipulation task which shows object manipulation in the virtual environment is harder than ($\cong 3$ times) in the real environment. Moreover, the variation in the virtual environment is much bigger ($SD_v = 43.20 > SD_r = 5.095$) than in reality. This huge difference, 9 times, is because of different training experiences. All the participants did not have any object manipulation experience in the virtual environment nearly. However, surely they had experienced similar object manipulation experience in the real environment. Since the participants did not have the same experience in the virtual environment, they recalled their experience from the real environment. Due to the fact that there is a big difference in terms of sensory input to the cortex and expectation, the calculated motor command either is overestimated or underestimated which in turn creates a big deviation from the mean value in the virtual environment.

5.2.7 Data analysis in the frequency domain

The recorded EMG signal is converted to the time-frequency space with the equation which has already been presented in chapter 4 (see section 4.4.5). The time-frequency space is a 3D representation of EMG data, Fig. 5.14. A comparison between the two environments in this space is not easy because many details appear. We transform this representation to an image representation by putting the amplitude as a pixel value. Now, the representation is much clearer and the difference can be seen much easier as shown in Fig. 5.15. In addition, we have recorded the signal for a complete set of sub-tasks involved in object manipulation as shown in Fig. 5.16. These sub-task time-frequency responses will help a lot to understand the nature of the manipulation in the two environments. For instance, as shown in Fig. 5.17.a, case 4, three activities can be easily distinguished in the real environment, for picking an object, carrying and placing it

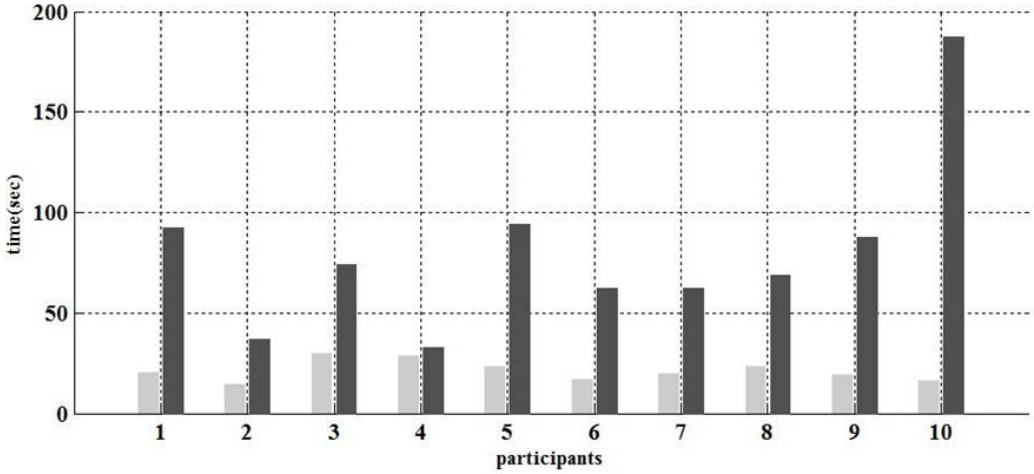


Figure 5.13: Time of task completion in the virtual and real environments.

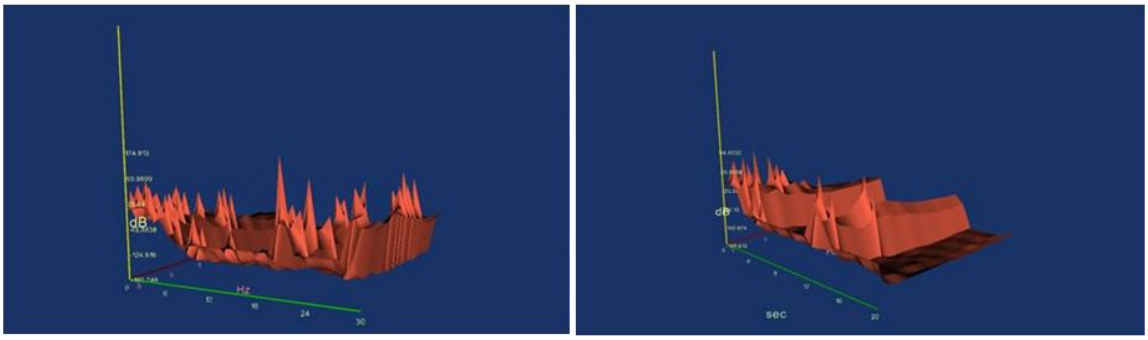


Figure 5.14: Time-frequency representation for object manipulation in the real and virtual environments.

in a given position. While as seen in the virtual environment (Fig. 5.17.b), these activities cannot be easily distinguished and some are short while the others are quite long.

5.2.8 Questionnaire data analysis

Data acquired from the participants were analyzed by the method explained in [230]. Here we will analyze some of the questions and answers from the questionnaires. We asked the participants to rate the level of their satisfaction in the real and virtual environments. As seen, the highest rate has been given to the real environment, in average 5.9 (see Fig. 5.18). We found the difference between the real ($M = 5.9$, $SD = 1.17$) and the virtual ($M = 4.5$, $SD = 1.19$) environments significant, $p < 0.05$ ($p = 0.0013$, $t(9) = 3.45$). The control (CF), sensory (SF) and distraction (DF) factors were calculated as three main factors with the correction factor proposed by Witmer and Singer [230], the result of the calculation is shown in Fig. 5.19. In average, the score associated with CF and SF is higher than DF.

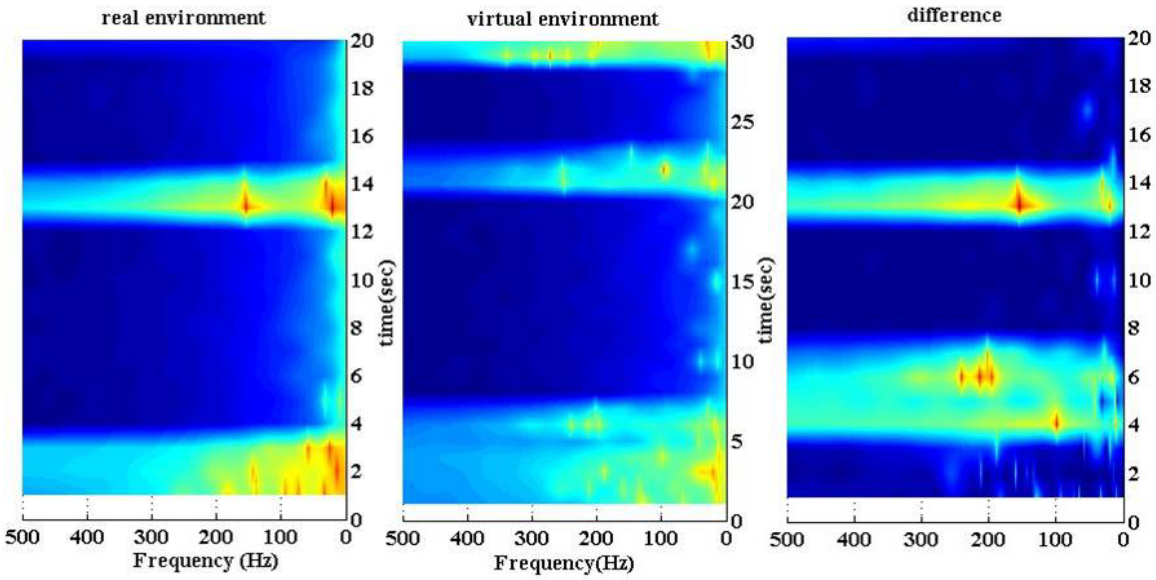


Figure 5.15: Time-frequency representation of the spectrum in the real and virtual environments and the difference.

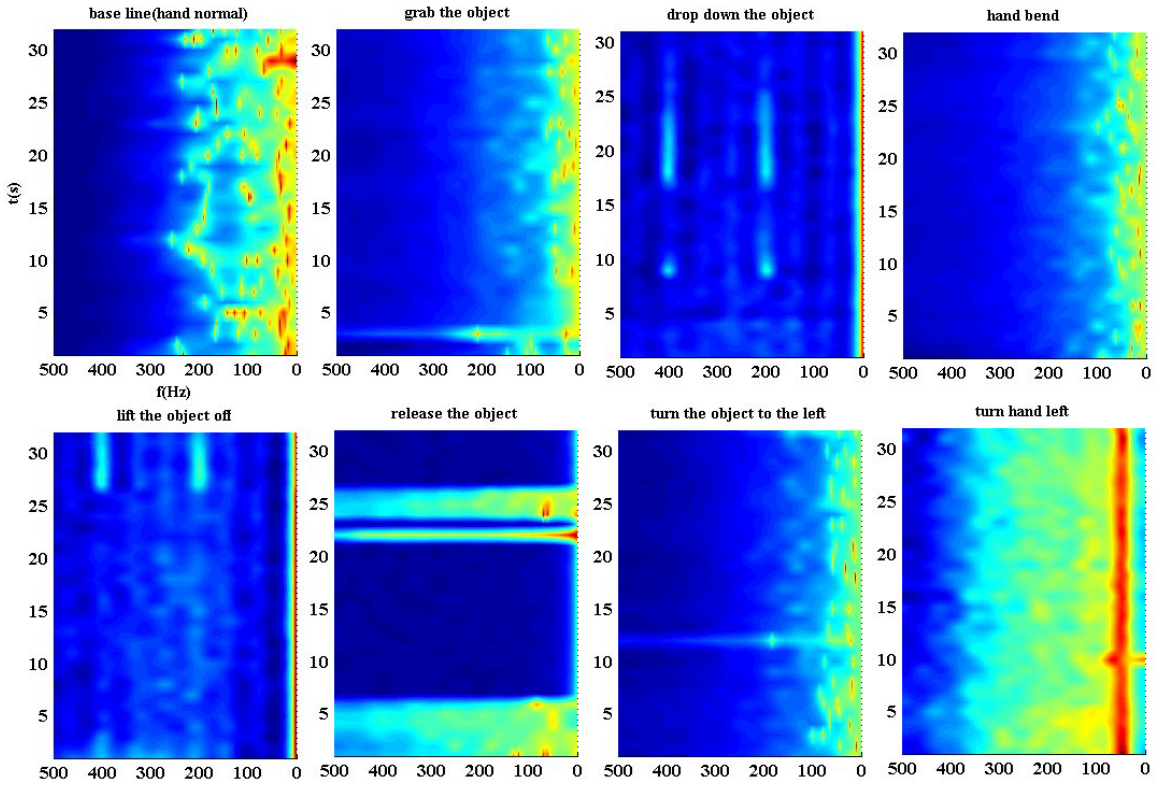


Figure 5.16: Time-frequency components of the hand movement with/without an object.

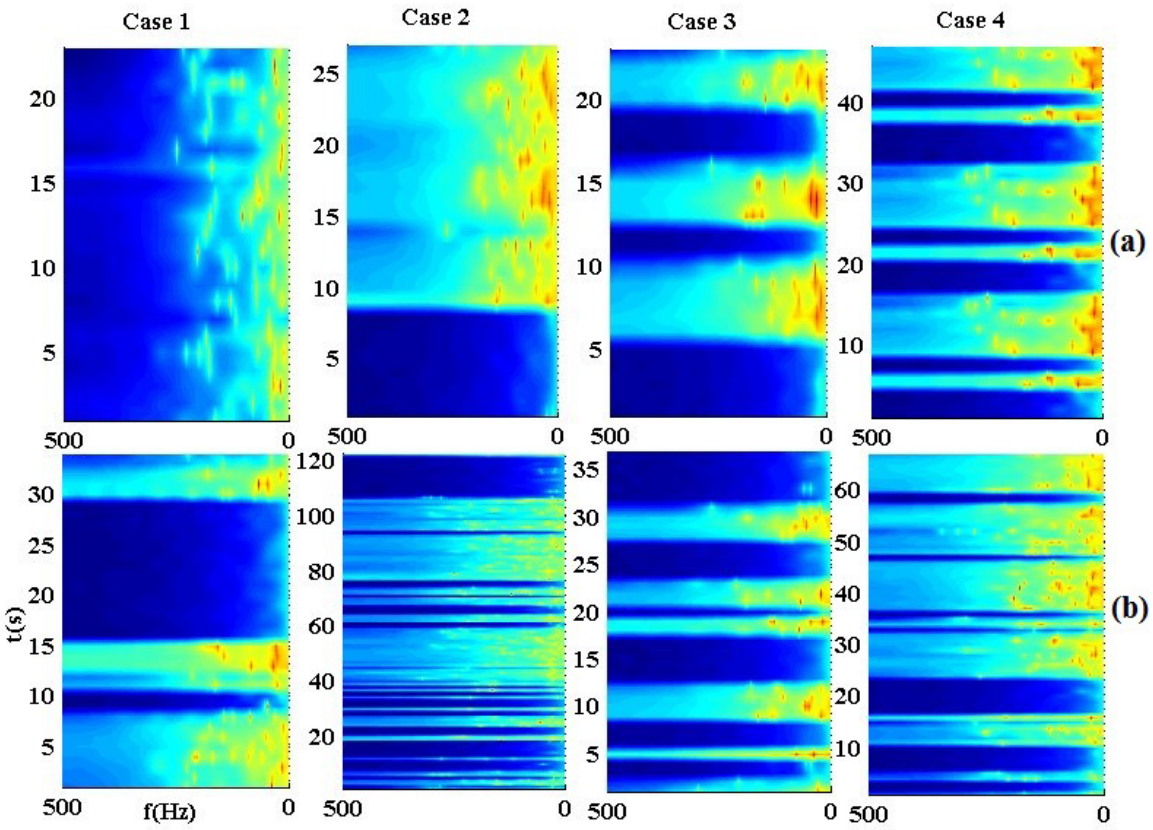


Figure 5.17: Four cases during interaction with an object in the real (a) and virtual environments (b).

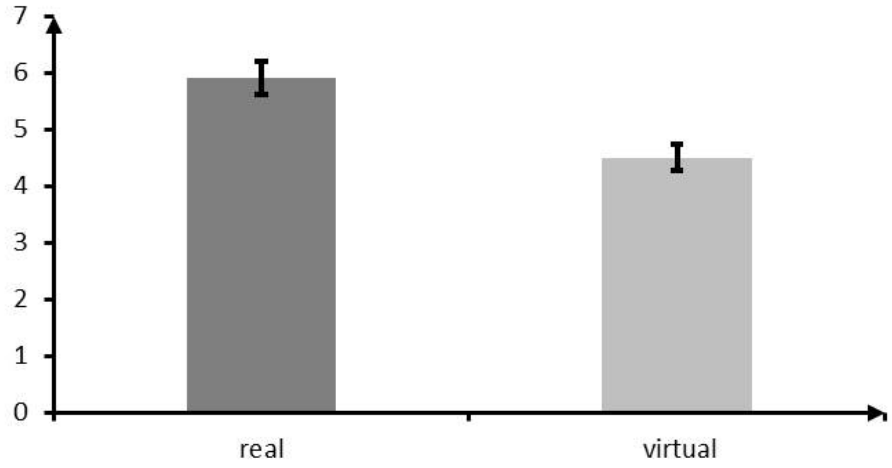


Figure 5.18: Level of satisfaction in both environments.

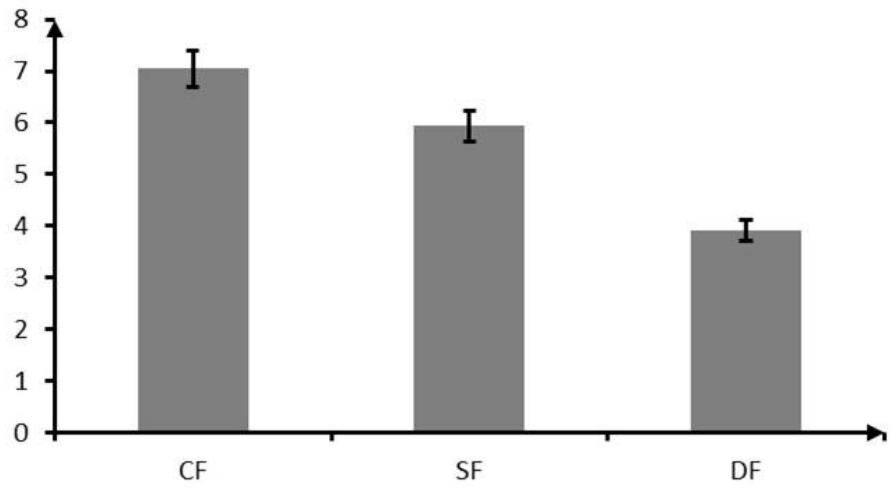


Figure 5.19: Sub-score of object manipulation for the virtual environment.

CF ($M = 7.049$, $SD = 1.32$) and SF ($M = 5.93$, $SD = 1.37$) are significantly higher, $p_1 < 0.001$ ($p_1 = 2.71 \times 10^{-6}$, $t(9) = 4.32$) and $p_2 < 0.001$ ($p_2 = 0.0009$, $t(9) = 3.75$) than DF ($M = 3.9$, $SD = 0.79$). In turn, it means the user interface was very well designed for an object manipulation task and was less distractive. It was capable to involve a lot of sensory inputs. To verify this claim, we have calculated DF against the involvement (INV). We found INV ($M = 3.9$, $SD = 0.785$) is significantly, $p < 0.005$ ($p = 0.0038$, $t(9) = 3.95$), higher than DF ($M = 5.75$, $SD = 1.64$) as shown in Fig. 5.20.

5.3 Conclusion

We evaluated a simple manipulation task using EMG and presence questionnaires and compared the user performance as well as presence between a real and a virtual environment situations. We showed that the task is much difficult to complete in the virtual environment than in the real environment. This is because of missing sensory proprioceptors.

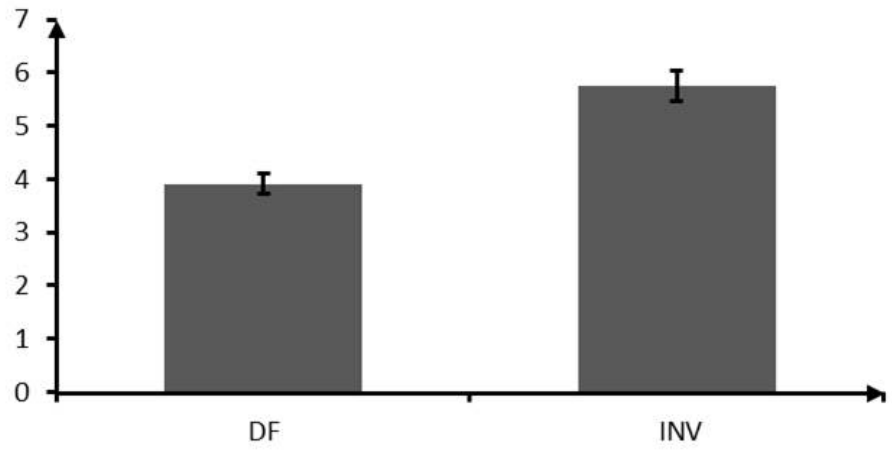


Figure 5.20: Level of involvement and distraction of object manipulation in the virtual environment.

6

VIMS estimation to guide navigation techniques

In this chapter, several features will be presented to describe VIMS mathematically. In fact, the entire biological model which has already been presented in chapter 2 will be studied by experiment. Since the first place that sickness can affect is the sensory motor area, we expect to see some changes in any aspect of locomotion and body movement in the time and frequency domains. For instance, in section 6.1, the effect of cognitive parameters will be simulated on the upright postural stand. This simulation is a connection between previous research and the analytical study of VIMS. In sections 6.1 and 6.2, four features including the area, the shape and the frequency components of the postural sway, and binocular disparity during vergence will be presented. These features will be used to predict and anticipate VIMS occurrences. We will present the application of the features in a real-time monitoring will be presented in section 6.3. In section 6.4, three experiments will be presented to show the effectiveness of the features across the subjects.

6.1 Features for VIMS prediction and estimation

6.1.1 Body postural sway modeling

The human upright equilibrium is controlled by a function called the "righting reflex". The righting reflex in humans malfunctions when exposed to 3D visual stimuli[202]. It has been observed that VIMS is caused by the disagreement between vergence and visual accommodation in a virtual environment[90]. Thus, virtual environments have been devised to reduce this disagreement[90]. Several parameters such as the "area of sway", the "total locus length" and the "locus length per unit area" have been widely proposed in clinical studies to quantize the instability involved in the standing posture[143]. It has been reported that a wide stance significantly increases the total locus length of the center of gravity (COG) of individuals with high SSQ scores, while the length in those of individuals with low scores is less affected by such a stance[177]. Takada et al.[202] observed that a high density of the center of pressure (COP) decreases during exposure to stereoscopic images. They notice that the Sparse Density (SPD), the dilation of the COG area, would be a useful index of a COG signal to measure VIMS.

A projection of a body's center of gravity (COG) on an XY plane of a detection balance board is measured as an average of the center of pressure (COP) of both feet. The analysis of a COG signal gives a good insight into the control theory of two-legged robots and the human stance[200], helps medical diagnosis and treatment for preventing falls in elderly people[197]. Recent studies suggest that animals experience sickness symptoms in circumstances where they have not acquired strategies to maintain their balance[159,197]. Riccio and Stoffregen argued that MS is caused by postural instability[159]. Stoffregen et al.[197] reported that the onset of motion sickness may be preceded by significant increases in postural sway.

The human quite upright stance is inherently unstable since it requires a large body consisting of multiple flexible segments to be kept in an upright posture with the center of gravity (COG) located high above a relatively small base of support (Fig. 6.1). The complexity of this system and its ability to maintain stable, against various perturbations, have attracted attention of researchers in various fields such as robotics, vision, control engineering, Human Machine Interface (HMI) and Human Computer Interaction (HCI). It has inspired various theories that try to explain the control mechanism of the human quiet stance. However, the true nature of this control mechanism is still an object of discussion and controversy. The ankle joint torque needed to control the body during a quiet stance can be evoked actively. Since the COG is located in front of the ankle joint, a plantar flexing torque[185,136,111,120] and an additional active torque[136,111], regulated by the CNS and produced by the plantar flexors[136,120], are continuously required to prevent the body from falling forward. Gatev et al.[48] reported a significant statistical correlation between a lateral gastrocnemius muscle activity and the position of a spontaneous body sway, which was measured as the COG position. The gastrocnemius muscle is a very powerful superficial bipennate muscle located in the back part of the lower leg. This finding supports the notion that the active torque is provided by lateral

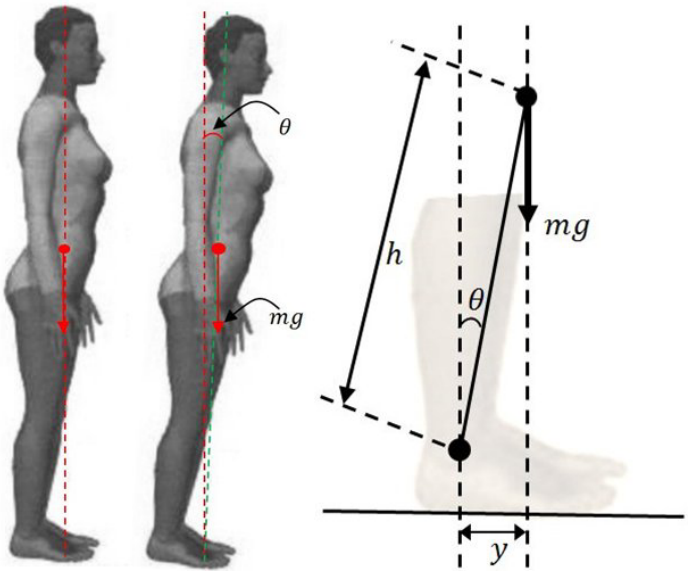


Figure 6.1: Dynamic model of the body sway during a quite upright stance.

gastrocnemius muscle contractions in response to body sway.

Fig. 6.2 illustrates the schematic of the model in which the body is regulated by a simple Proportional Differential (PD) controller. The model was adapted from[120], the components of the neurological time delay (closed-loop system with a time delay) were chosen according to recent findings available in the literature as discussed in[120].

The body dynamics and kinematics during a quiet stance were described using an inverted pendulum (Fig. 6.1). The input to the body model was the total torque exerted about the ankle joint. The PD controller was used to simulate the regulation of the balance performed by the CNS as defined by the proportional and derivative gains, K_p and K_d, respectively. A low-pass filter after the proportional gain is used to smooth the controller output signal. The time delay, τ_F, represents the latency recorded from the instant the sensory stimulation is provided to the foot, to the instant the sensory evoked potential is recorded in the somatosensory area I of the brain as explained in chapter 2 (see section 2.6).

The electromechanical response time, τ_E, represents the time difference between the moment a muscle activity signal is generated in the sensorymotor and the moment the muscle reacts. The motor command time delay, τ_M, represents the cumulative time loss due to the sensory-motor information process in the CNS and the neural transmission from the CNS to the plantar flexors. These three time constants, τ_E, τ_M and τ_F, are referred to as the cognitive parameters of the model. Two remaining components represent the body model which will be explained below in detail. The motor command (M_c) is calculated using

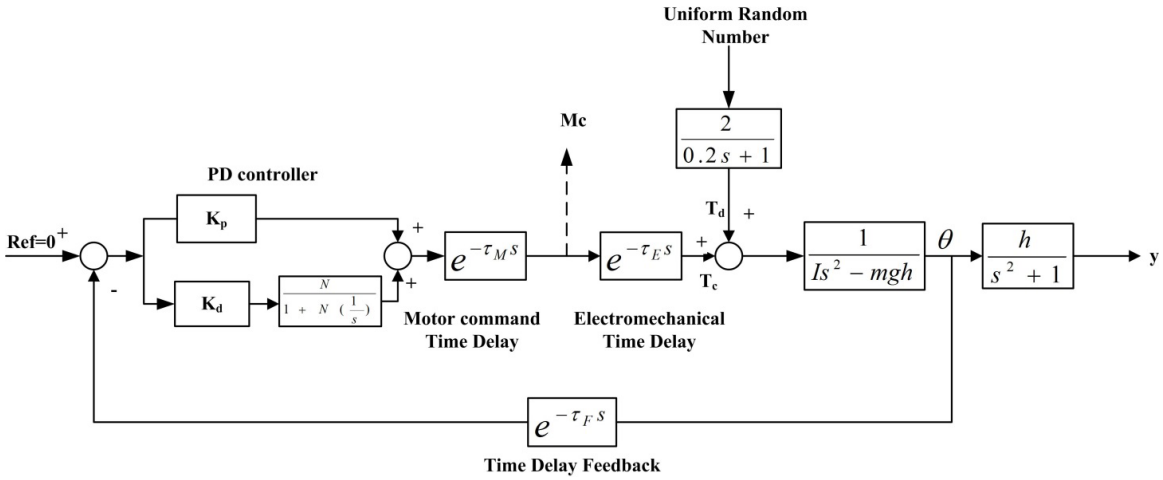


Figure 6.2: Inverted pendulum model and closed-loop control scheme of a quiet stance.

the COG position and velocity information according to the following equations, (6.1)-(6.3):

$$M_c(t) = -K_p\theta\left(t - \tau_F - \tau_M\right) - K_d\dot{\theta}\left(t - \tau_F - \tau_M\right) \tag{6.1}$$

Equation (6.2) is achieved after applying the Laplace transform to (6.1)

$$M_c(s) = \left[-K_p - K_d s\right]\theta(s)e^{-(\tau_F+\tau_M)} \tag{6.2}$$

To show the effect of indigenous (internal from the CNS) and exogenous (external physical) forces, equation (6.2) is written as (6.3)

$$M_c(s) = \left[K_p + K_d s\right]\theta(s)e^{-\tau_M} \times \left(0 - e^{-\tau_F}\right) \tag{6.3}$$

We have introduced θ to describe the angular deviation from the upright position, and assume that we can affect the body by the total torque $T = T_c + T_d$ at its base. As illustrated in Fig. 6.1, gravity forces the body to get away from its upright equilibrium. The motion of the human body during the stance posture is rotational rather than translational, so we will have to use the following variation of Newton's law

$$M = I\ddot{\theta}(t) \tag{6.4}$$

Coefficient I in (6.4) represents the moment of inertia, and M denotes the total momentum acting on the body:

$$M = T(t) + mgl\sin\left(\theta(t)\right) - b\dot{\theta}(t) \tag{6.5}$$

The first two terms in (6.5) represent the force applied at the base (time-varying) and the influence of gravity, while the last term describes a damping proportional to the angular velocity. We can now combine (6.4) and (6.5) to construct (6.6).

$$I\ddot{\theta}(t) = T(t) + mgl\sin\left(\theta(t)\right) - b\dot{\theta}(t) \tag{6.6}$$

The solution $\theta(t)$ to this ordinary differential equation predicts the motion of the body around the z axis. We assume the human body as a system without loss and damping, just for the simplicity of the simulation and analysis. Therefore, term $b\dot{\theta(t)}$ is eliminated from (6.6), see[120] for more detail. As it is often hard to find analytical expressions and solutions for nonlinear differential equations, we will assume that the deviation from the upright position is small, so that $\sin\left(\theta(t)\right) \approx \theta(t)$, and consider the linearized model as (6.7).

$$I\ddot{\theta}(t) = T(t) + mgl\theta(t) \tag{6.7}$$

After applying the Laplace transform to (6.7) and assuming all the initial condition zero, the final equation (6.8) will be achieved

$$T(s) = Is^2\theta(s) - mgl\theta(s) \Rightarrow H(s) = \frac{\theta(s)}{T(s)} = \frac{1}{Is^2 - mgh} \tag{6.8}$$

Equations (6.1) through (6.8) allow simulating human postural sway by substituting the system parameters with appropriate numerical values. The body model, as shown in Fig. 6.2, was simulated in MATLAB-Simulink. The PD controller is obtained by setting integrator gain $I = 0$ in the Simulink standard PID controller definition (6.9).

$$P + I\frac{1}{s} + D\frac{N}{1 + N\frac{1}{s}} \tag{6.9}$$

Human upright postural modeling and simulation

The model of body sway during upright standing including cognitive parameters was derived. The model can successfully describe body movements under different cognitive parameters reconfigurations. The area associated with a scatter plot of Left/Right (L/R) and Forward/Backward (F/B) signals in XY plane is defined as an ellipse or a circle which holds 90% of the scatter points. Both L/R and F/B signals are needed to depict the COG movement on XY coordinates, however the model can provide either F/B or L/R movement. To overcome this problem, the F/B and the L/R movements were simulated with the same model however different values were set up for the cognitive parameters (PD gains and time delays).

	Parameters					
States	τ_F (ms)	τ_E (ms)	τ_M (ms)	k_p	k_d	N
Selection range	$35.1 - 40.1$	$10.54 - 11.5$	$25 - 215$	$50 - 3000$	$50 - 2000$	$5 - 2000$
Stable L/R	40	11	135	750	350	20
Stable F/B	40	11	135	680	320	15
Marginally stable L/R	40	11	175	750	392	27
Marginally stable F/B	40	11	175	750	380	25

Table 6.1: Simulation parameters for stable and marginally stable studies of a stance posture.

In the simulation, we kept the time delay unchanged and modified PD parameters to generate F/B and L/R signals (see table 6.1).

Based on the classic control theory[99] for any close-loop control process, there exist three states: stable, marginally stable and unstable. Stability during a stance posture indicates the situation that the body stays upright with very tiny movements. On the contrary, the body tends to fall down when it is unstable. During marginally stable, the body tends to become unstable while the CNS tries to keep it upright by sending commands to different joints using the righting reflex function. Consequently, the body physically fluctuates around z axis and the CNS alternates between stable and unstable states. An example of F/B and L/R movements for marginally stable and stable states are shown in Figs. 6.3.A and 6.3.B respectively. As shown in Figs. 6.3.A and 6.3.B, the amplitude of the signal is higher for the marginally stable state comparing to the stable state which in turn means the course of movement in marginally stable is higher. Besides, the amplitude of the two signals for marginally stable at the beginning is low and increases exponentially by time, while the external input (see input values, Ref $= 0$, in Fig. 6.2) always is zero during the entire simulation time.

The controller and the cognitive (CNS) parameters are set to values shown in table 6.1. Different constants of the body transfer function are calculated for a male adult as found by[120] ($m = 76$ kg, $I = 66$ kg.m^2 and $h = 0.87$ m). The same constants will be used later for frequency component analysis.

The simulation data will be analyzed in the time and frequency domains separately as follows.

Area and shape of the COG (time domain analysis)

First, the simulation data (Figs. 6.3.A and 6.3.B) will be analyzed by extracting features in the time domain without converting the signal into another space such as frequency, time-frequency and multi-resolution. The F/B and L/R signals at time t (or k^{th} sample) determine x and y coordinates in Cartesian coordinates. These two signals are projected into XY plane by $(x(t), y(t))$ couple, a COG scatter plot, as shown in

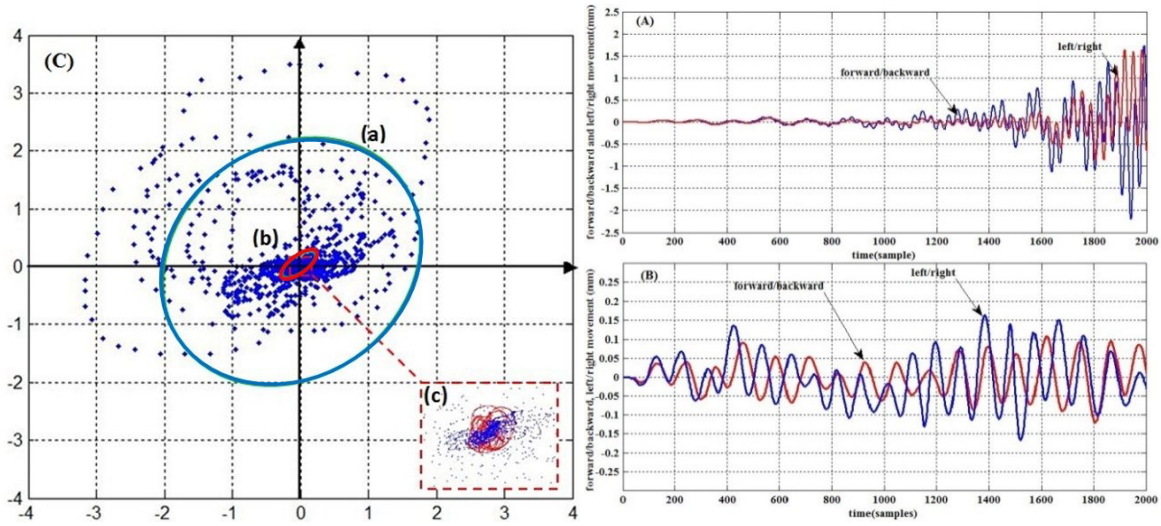


Figure 6.3: (A) Forward/backward and left/right signals for a marginally stable state, (B) forward/backward and left/right signals for a stable state ,(C) COG area created by projecting the forward/backward and left/right corresponding values onto XY plane: a) points associated with the stable state, b) points associated with the marginally stable state, c) zoom in to show detail of the stable state which is hidden under the marginally stable points.

Fig. 6.3.C. The area and the shape are extracted from the COG scatter plot as features. The features will be used as criteria to compare body sway during the stable and marginally stable states. As mentioned above, by definition [205], the COG area is defined as an optimum ellipse (circle) surrounded by 90% of the points. The confidential interval is calculated from the scatter center not from the origin of the coordinates.

The marginally stable and stable signals as shown in Figs. 6.3.A and 6.3.B, blue and red curves, are projected on XY plane. Since the blue points cover the red points, we zoomed in a small region of Fig. 6.3.C to show the points associated with a stable state, red points. Circles "b" and "a" (Fig. 6.3.C) demonstrate the COG area for the stable and marginally stable states of the body. As seen, the body does not move widely during the stable state, hence the COG area (Fig. 6.3.C.b) is quite dense and smaller than in the marginally stable state in which the body movement is larger and the corresponding area dilates significantly (Fig. 6.3.C.a). As shown, the area associated with the stable state has an elliptical shape while marginally stability is circular.

Difference between LF and HF components (frequency domain analysis)

The L/R and F/B sway signals are generated using the proposed model for three different states: marginally stable (Fig. 6.4.a), unstable (Fig. 6.4.b) and stable (Fig. 6.4.c). First, the F/B and L/R signals are converted to a spectrum-frequency representation using the Fast Fourier Transform (FFT) in order to analyze and interpret the signals in the frequency domain and disclose their hidden aspects.

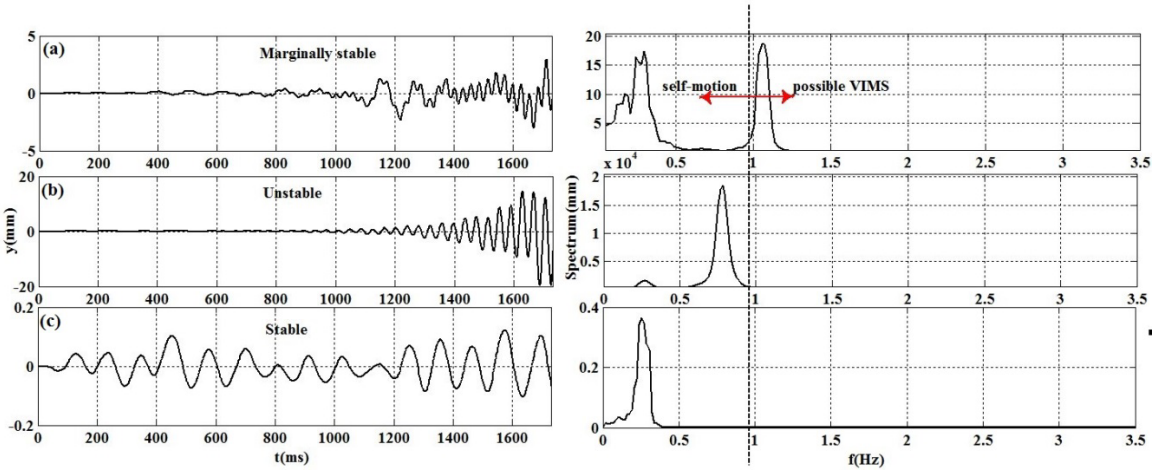


Figure 6.4: Body F/B postural sway signal for the a) marginally stable, b) unstable, c) stable states in the time domain (left column) and associated spectrum in the frequency domain (right column).

	Parameters					
States	τ_F (ms)	τ_E (ms)	τ_M (ms)	k_p	k_d	N
Possible choice	$35.1 - 40.1$	$10.54 - 11.5$	$25 - 215$	$50 - 3000$	$50 - 2000$	$5 - 2000$
Stable	40	11	135	750	350	20
Marginally stable	40	11	175	750	380	9.5
Unstable	40	11	190	850	590	10

Table 6.2: Simulation parameters for different states of the stance posture.

Since the F/B and L/R signals are similar both in characteristics and in shape, only the F/B signal will be analyzed in this section. The high and low frequency components will be extracted as features to study the upright postural sway. The time delay associated with feedback (τ_f) and feed forward (τ_e), (τ_m) delays were set to 40 ms, 11 ms and 135 ms respectively in this simulation (table 6.2). However, the exact value of the motor command time delay, i.e., the time needed for the sensorymotor information to be processed in the CNS, is unknown. Since the upright postural sway is important in this study, only the stable and marginally stable states will be considered.

The body can fall into marginally stable under different situations. As shown in Fig. 6.4.a, the spectrum of the marginally stable F/B sway has two components: high (HF) and low frequency (LF), regardless of the stimulus causing the sway, external or internal. The LF and HF components are associated with voluntary and involuntary movements. The HF component stretches beyond 1 Hz (see Fig. 6.4.a), while the LF component remains below 1 Hz. A previous study by Bos[16] showed that 1 Hz is an important frequency because it is a criterion to measure the body motion. As shown in Fig. 6.4, under a completely stable state,

the body stays upright with minimum sway and only voluntary movement can occur. Voluntary movement is slow thus its spectrum has frequency components close to zero ($f \downarrow = \frac{1}{T\uparrow}$). The bandwidth of voluntary movement (stable state) is limited to 0 Hz-0.5 Hz while the bandwidth for unstable movement has two components 0.3 Hz-0.4 Hz and 0.5 Hz-1.0 Hz (Fig. 6.4.b). As seen in Fig. 6.4.a, the second frequency component (HF) falls beyond 1.0 Hz in the marginally stable state. It means, more high frequency components due to a malfunctioning of the CNS or an inaccurate estimation of the body state appear in the sway signal.

6.1.2 Binocular disparity

Since nearly a century the theory of sensory conflict has been the subject of many studies[17,72,84,85,19,25,210]. There is no doubt in the efficiency of VR technology for commercial usage such as manufacturing, design, construction, exhibitions and industrial public shows. However, the main issue of VIMS still remains unsolved and more research is felt more than any time in this field, especially with the rapid growth of video games, VR and display technologies. The severity of VIMS in some applications such as virtual reality was so severe that the International Organization for Standardization (ISO) has held several symposiums and workshops in dynamic image safety and reported that more research is needed to determine the causal factors and indexes for VIMS[210]. About 30% of the population are susceptible to motion sickness[188,187]. In this regards, different statistics have been published so far to show the importance of VIMS susceptibility. For instance, one study showed that about 30% of the Chinese population is susceptible to motion sickness[186].

Researchers have been measured VIMS in virtual environments (VEs) for two purposes. First, they thought of VIMS as an effective criterion to evaluate different navigation/manipulation interfaces and select the interface with less sickness. This group of researchers used SSQ as psychological measurements in their surveillances. Second, VIMS is measured and analyzed to see under what condition it will emerge and how it progresses by time with different configurations of navigation parameters. The detection of first symptoms of VIMS helps to inform the user of imminent sick in a couple of minutes and consecutive cares. This section will focus on the second aspect of VIMS.

When watching a moving visual stimulus, our eyes will voluntarily follow the moving stimulus. Eye movements are classified in saccade, smooth pursuit, optokinetic reflex (OKR), vestibulo-ocular reflex (VOR) and vergence[237], as seen in chapter 2. Different efforts have been made to characterize, model and study each component of eye movement. For example, Juhola[87] established a solid theoretical background to detect different kinds of OKR by applying a recursive filter under an ideal situation (without noise). This type of eye movement is characterized by two phases: 1) slow phase eye movement (smooth pursuit) following the stimulus, 2) fast saccadic eye movement to reset the eye positions. Depending on

the velocity of the stimulus, the slow phase OKR velocity can match about 30% to 70% of the velocity of the stimulus[103].

In 1994, Ebenholtz et al.[39] proposed a theory to link VIMS with eye movements. This theory is consistent with the findings that the suppression of eye movements by fixation can significantly reduce the level of VIMS[195,222]. Ebenholtz et al. hypothesized that the afferent signals in the ocular mechanism will trigger the vagal nuclei, resulting in a range of sickness symptoms associated with the autonomous nervous systems (so called parasympathetic nerves system), and the nystagmus theory. Because eye movements follow a foveal stimulation and vection follows a peripheral stimulation[19], the nystagmus theory indicates that in the presence of foveal stimulation, sickness will correlate with eye movements but not necessarily with vection. Since then, there have been competing studies reporting the decoupling between vection and VIMS[222] as well as the coupling between vection and VIMS[25,110]. However, the finding of neuroscientists[25] shows a coupling between vection and VIMS, at least based on the modeling of binocular vision.

During watching a moving stimulus, eye fixation will result in the suppression of eye movement and an increase in the retinal slip velocity. The retinal slip signal is defined as the difference between the current position of the eye and the target object position. The derivative of the retinal slip signal is called the retinal slip velocity. Both retinal slip and its velocity signals provide feedback for the eye movement mechanism[237] to correct the image position on the fovea. As a result, the significant reduction of VIMS during eye fixation can be due to either the suppression of optokinetic nystagmus (OKN), or an increase in the retinal slip velocity, or both. Because changes in the retinal slip velocity have been shown to affect the level of VIMS and vection[73], if the effects of eye fixation are proven to be due to an increase in the retinal slip velocity, it will provide much support to the sensory conflict theory which predicts VIMS associated with vection. Hu et al.[73] reported that when watching striped rotation patterns and changing the rotating velocity once from 60 dps (degree per second) to 90 dps (increasing), and once from 60 dps to 15 dps (decreasing) maximum, VIMS will appear at 60 dps. On the other hand, reducing the rotating velocity of the moving pattern from 60 dps to 15 dps also reduces the level of VIMS. Maximum VIMS occurs when the moving striped patterns rotate at 60 dps.

The effects of OKN and vection can be separated by modifying the rotating pattern[85]. Ji et al.[85] has shown that VIMS severity increases as the pattern velocity increases from 34 dps to 60 dps in the absence of vection. The same effect has been observed when OKN is suppressed. The profile of the summed result well matched with the profile reported by Hu et al.[73]. Guo et al.[52] studied the use of measurable optokinetic after nystagmus (OKAN) parameters to predict a susceptibility to VIMS. They suggested this measure as an objective indicator of VIMS to ISO organization.

VIMS can appear under different conditions during navigation in virtual environments (VEs). Basically, any travel in VEs consists of translation and rotation or both. It has been shown that the main cause of VIMS during translation movement is vection[25]. This finding was justified in several studies[31]. The

main cause of VIMS during rotational movement still is the subject of various studies as seen above. Lo and So[110] compared the level of VIMS resulting from watching a visual scene with oscillations around different rotational axes in a VE. A stereoscopic immersive virtual environment with a multi-mural display system (for instance a CAVE) uses 3D glasses to convey the sensation of full immersion to the user. It seems the projection of two images for each eye with two different colors disturbs vergence[237] and leads to binocular disparity. The same problem and corresponding effects on headache were reported by Ukai[211]. These findings can be strongly supported not only by subjective studies but also by the binocular vision modeling proposed by Zhang and Wakamatsu[237]. Binocular disparity can be exacerbated during rotational movements in VEs and leads to severe headache or VIMS.

6.2 Experimental validation

Based on Reason's theory[157], VIMS is expressed by (6.10)

$$f\left(I_O, I_V, I_{\text{pro}}\right) = \sum_{i=1}^{3} \alpha_i \left|\Delta I_{VN}^{i}\right| \tag{6.10}$$

where, $\left|\Delta I_{VN}^{i}\right|$ in (6.10) is defined by $\left|I_{\text{pro}} - I_O\right|$, $\left|I_{\text{pro}} - I_V\right|$, $\left|I_O - I_V\right|$ and indexes O, V and pro refers to ocular, vestibular and proprioceptive inputs, respectively. Bos et al.[17] extended the model (6.10) by proposing a new index as a strong cause of VIMS, "subjective-vertical". He states "*people only get sick when their head changes (apparent) suddenly against gravity*". Earth gravity is aligned with r axis of the spherical coordinates, in each point on the Earth. As a result, the new component of VIMS can be interpreted as the variation of the head along r axis with respect to time which will be denoted by $\frac{\partial^2 I_V}{\partial r \partial t}$. Consequently, equation (6.10) is replaced by (6.11) and the model is capable to explain more observations related to VIMS and MS.

$$f\left(I_O, I_V, I_{\text{pro}}\right) = \sum \alpha_i \left|\Delta I_{VN}^{i}\right| + \beta \frac{\partial^2 I_V}{\partial r \partial t} \tag{6.11}$$

One of the main focus of this thesis is to show the inter-ocular difference during rotational movement can significantly contribute in VIMS. Accordingly, model (6.11) can be completed by adding another component, ΔI_O, as shown in (6.12).

$$f\left(I_O, I_V, I_{\text{pro}}\right) = \sum \alpha_i \left|\Delta I_{VN}^{i}\right| + \beta \frac{\partial^2 I_V}{\partial r \partial t} + \gamma \Delta I_O \tag{6.12}$$

Above, four features were introduced which can be used for VIMS prediction. These features include:

1. The difference between LF (Low Frequency) and HF (High Frequency) components of post-exposure postural sway (the more the difference the higher the sickness);

2. The dilation of the post-exposure COG (Center of Gravity) area and the COG shape variation;

3. Binocular disparity at the onset of eye-movement (the difference between the two eyes immediately after stimulation);

4. The expected time of VIMS onset.

Below, we will study these four features in more detail practically and will show how they can help to predict a VIMS occurrence. The efficiency of the proposed features will be validated by practical experiments. The validation procedure is quite simple: a group of participants is immersed in a virtual environment (VE), different psychophysiological measurements (such as the COG and eye movement) are recorded and self-report questionnaires (Kennedy's SSQ[98]) is filled out by each participant. The extracted features from measurements and the questionnaire scores should be highly correlated. Three experiments were designed for this purpose which will be presented below.

6.2.1 Experiment design

The test setup used in the following experiments is the same as the one described in section 5.1 of chapter 5.

Participants and questionnaire

Seventeen subjects (13 males and 4 females: 31.58 ± 12.69 years, 74.65 ± 15.22 kg) participated in the experiment. There was a briefing to give enough information about the test procedure and possible risks before each experiment individually. All the subjects participated voluntarily in the experiments. A pre-exposure questionnaire (Q1) was obtained from each subject to know their backgrounds and to evaluate their health condition (see appendix A). The result of the questionnaire showed that there was no test subject whose participation in the experiment would be unsuitable due to health issues. Kennedy's SSQ questionnaire (see appendix B) was used with the same scoring indexes adapted to collect a self-report questionnaire and calculate the sickness total score.

Experiment procedure and data acquisition

The procedure was designed as follows:

1. A pre-exposure questionnaire was asked to be filled out (Q1) by each participant.

2. Since the participants needed to know how to navigate with devices we had a training period (2-3 min) before exposing participants to visual stimuli (base line).

3. The COG of each participant was recorded for $t_1 = 30$ s (pre-exposure measurement).

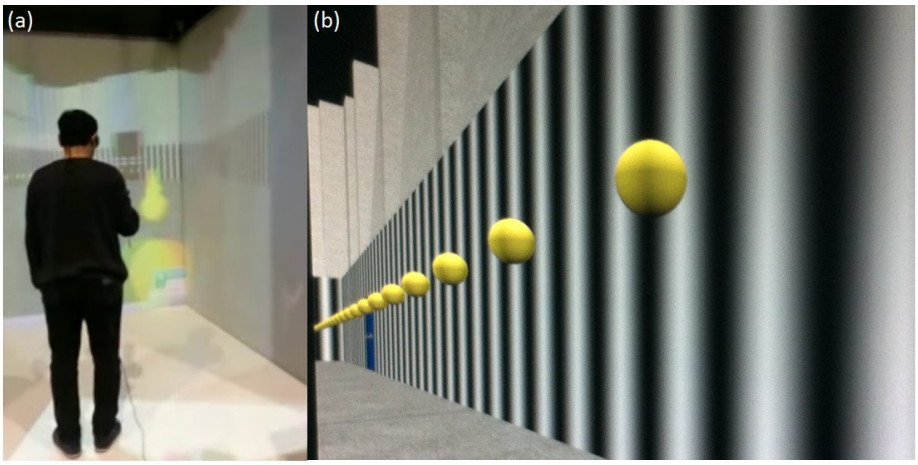


Figure 6.5: a) Experiment inside the CAVE, b) path indicator and environment pattern.

4. The participants navigated along a path (detailed in Fig. 6.5.a) for t_2 s.
5. The COG signal was recorded immediately after navigation for $t_3 = 30$ s (post exposure measurement).
6. An SSQ (Q2) was filled by the participants.
7. Steps 3 to 6 were repeated 7 times.

In the following section, the data will be analyzed both in the time and frequency domains to validate the results and compare them with the simulation results. In general, a feature explains certain characteristics of a given signal, measurement, and time series. The mean, median, variance, maximum and minimum of a signal are the most known features, however an efficient feature has a more complicated definition in practice. The efficiency of three features, i.e., the difference between LF and HF components, the area and the shape of the COG, and binocular disparity will be studied using experimental data.

6.2.2 Feature 1: COG area and shape

The experiment has been carried out for a period of $t = 50 \pm 20$ min based on the procedure explained in section 6.2.1. The F/B and R/L signals are recorded directly by the COG sensor (here a Techno-concept sensor) and then the parameters are calculated by the interfacing software (see the Techno-concept manual for further detail [205]). Fig. 6.6 shows an example of a recorded F/B (red) and L/R (blue) signal for pre and post-exposures. As seen, it is fairly difficult to compare these two signals in the time representation without any feature extraction. Feature extraction facilitates the comparison between these two raw signals.

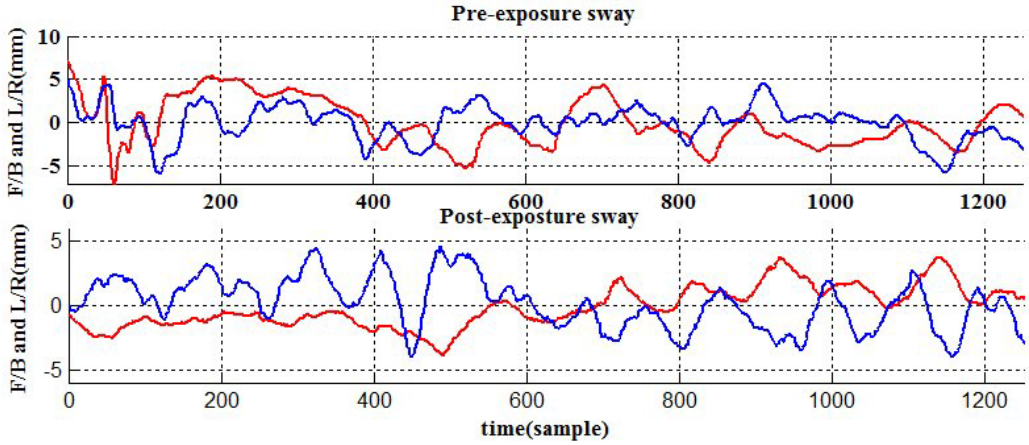


Figure 6.6: Example of F/B and L/R sways signals for pre and post exposures.

Moreover, the analysis of a huge amount of data by feature extraction is limited to the study and comparison of a few set of features. 13 features (see the first column in table 6.3) are calculated using the pre and post-exposures measurements as shown in table 6.3. The pre and post-exposures values refer to *min* (pre-exposure) and *max* (post-exposure) values among the participants. Now, more efficient parameters can be identified only by looking at the variation (fourth column in table 6.3). The column associated to "*Variation*" shows the difference between pre and post-exposures measurements (columns 2 and 3).

Statistical analysis for $n = 17$ participants shows that 6 out of 13 COG features including Area ($F(1,16) = 198.6, p < 0.005$), Length ($F(1,16) = 154.9, p < 0.005$), Lng. L/R ($F(1,16) = 105.2, p < 0.001$), Lng. Fr./Bk. ($F(1,16) = 149.5, p < 0.001$), Slope ($F(1,16) = 91.23, p < 0.01$), S Var. ($F(1,16) = 98.23, p < 0.001$) have experienced significant variations. Other features either do not follow a specific trend or did not undergo significant variations. The term "*no trend*" means the pre and post-exposures difference is not significant and the variation across the subjects is random. On the other hand, the term "*with trend*" means there is a meaningful difference either positive or negative (some remains unchanged) but it is not significant. L/R Avg. ($F(1,16) = 1.23, p = 0.46$, no trend), Fr./Bk. Avg. ($F(1,16) = 2.19, p = 0.58$, no trend), ANo2 L/R. ($F(1,16) = 3.94, p = 0.17$, with trend), ANo2 Fr./Bk. ($F(1,16) = 2.98, p = 0.96$, with trend), LFS ($F(1,16) = 1.73, p = 0.64$, no trend), SV Fr./Bk. ($F(1,16) = 1.05, p = 0.31$), S Avg. ($F(1,16) = 2.87, p = 0.08$, no trend) have not changed dramatically. Therefore they have not been considered in this study. As seen in section 6.1.1 (postural sway study), some researches were focused on the area as well. Furthermore, the aim is to see if the simulation results can be proven by experiments or not. Among all the features that experienced a significant difference, the "*area*" and the "*shape of the area*" are selected to be analyzed.

The COG area is calculated at 8 points during the test (see the test procedure in section 6.2.1). However

	COG SENSOR DATA				
PARAMETERS	abbreviation	pre-exposure	post-exposure	variation	p
Forward/Backward Average	Fr./Bk. Avg.	-26.2	3	29.2	0.46
Left/Right Average	L/R Avg.	-8.8	59.8	68.6	0.58
Area	Area	29	553.4	524.4	0.0035
Length	Length	163	489.3	326.3	0.0023
Length of Left/Right	Lng. L/R	59.9	195.5	135.6	0.0001
Length of Forward/Backward	Lng. Fr./Bk.	127.5	436.9	309.4	0.00054
Slope	Slope	43	129.3	86.3	0.0075
Normal Amplitude at 0.2 Hz Left/Right	ANo2 L/R	1.1	26.4	25.3	0.17
Normal Amplitude at 0.2 Hz Forward/Backward	ANo2 Fr./Bk	2.1	29.5	27.4	0.96
Length function of the surface	LFS	0.4	0.9	0.5	0.64
Sway Velocity Forward/Backward	SV Fr./Bk.	-17.2	8.2	25.4	0.31
Speed Variance	S Var.	10.7	188.1	177.4	0.00017
Speed Average	S Avg.	5.3	15.6	10.3	0.08

Table 6.3: Sensor features extracted from pre and post-exposures measurements of the COG sensor.

only three instances are more important in this study: base-line (pre-exposure), at the sickness onset (before sickness) and at the end of the experiment (post-exposure). The time of the sickness onset is determined by the examiner based on the participants' self-reports. The examiner simply asks the participant "*Does he/she feel any vomiting, stomach awareness, saliva increase and nausea?*" at the end of each step. Moreover, two indexes, i.e., "*Susceptibility (S)*" and "*Average Sickness (AS)*", are defined to double check the self-report. The S and AS indexes are calculated by $S = \left|\frac{\text{SSQ}_{n-1} - \text{SSQ}_n}{t}\right|$ and $AS = \frac{\sum_{i=1}^{m} \text{SSQ}_i}{m}$, respectively using SSQ reports, where SSQ_{n-1} and SSQ_n represent the total sickness score associated with the previous and the current steps, SSQ_i is the score calculated at the end of the ith step, t is the time the participants spend to complete the current step, and m is the number of the steps that have already been completed. For example, suppose $\text{SSQ}_1 = 125$, $\text{SSQ}_2 = 150$, $\text{SSQ}_3 = 300$, $t_1 = 3$, $t_2 = 2.5$ min are reported in an experiment, then $S_1 = \left|\frac{150-125}{3}\right| = 8.3$, $S_2 = \left|\frac{300-150}{2.5}\right| = 60$, $AS_1 = \frac{275}{2} = 137.5$, $AS_2 = \frac{575}{3} = 191.6$. Now, it can be clearly seen that some event is happening in step 2 because the two indexes increased suddenly. Based on the experiment, a rapid change in these two indexes happened approximately at the same step reported by the participant. Another way to double check the self-report is to plot the SSQ score versus the steps and find the first step where the variation happens. To the best of our knowledge very little is known about the time of the sickness onset (Oman's work in chapter 2).

The COG area associated with these three instances, for one of the participants, is depicted in Fig. 6.7. The COG areas corresponding to pre-exposure (Fig. 6.7.1.a), S_{pre}, at the sickness onset (Fig. 6.7.1.b), S_{SO}, and post-exposure, S_{post}, (Fig. 6.7.1.c) are shown in Fig. 6.7 (top-right in red, green and blue colors). Then, these areas are superimposed as a scatter plot in Fig. 6.7.1.d, one at the top of the other to highlight the difference between the areas. As seen, post-exposure distributed sparsely comparing to the pre-exposure area and the shape is almost circular. The shape of the COG area remains elliptical (red and green areas in Fig. 6.7, left side) till the onset of sickness and after that it tends to be circular (blue circle). The COG area is defined as the geometrical shape (ellipse or circle) which contains 90% of the points of a scatter plot. The area, shape and sickness score are changing by time simultaneously due to exposure to the virtual environment. Now, it must be proven that the variation in the area and shape is correlated with the variation in the sickness score across the subjects.

The difference between the three areas, i.e., pre_sick_area ($\Delta S_1 = S_{\text{SO}} - S_{\text{pre}}$), pre_post_area ($\Delta S_2 = S_{\text{post}} - S_{\text{pre}}$), and sick_post_area ($\Delta S_3 = S_{\text{post}} - S_{\text{SO}}$) are calculated for each participant.

Then, equivalently the difference between the total SSQ score at these three important instances as mentioned above, i.e., SSQ1 ($\text{SSQ}_{\text{SO}} - \text{SSQ}_{\text{pre}}$), SSQ2 ($\text{SSQ}_{\text{post}} - \text{SSQ}_{\text{pre}}$), and SSQ3 ($\text{SSQ}_{\text{post}} - \text{SSQ}_{\text{SO}}$) are derived assuming that the total sickness scores for pre, at the sickness onset, and post-exposures are represented by SSQ_{pre}, SSQ_{SO}, and SSQ_{post}. Fig. 6.7.2 shows the correlation between SSQ1 and pre_sick_area which means the dilation of the area is highly correlated with the increase in the total calculated SSQ score. Figs. 6.7.3 and 6.7.4 present similar results for sick_post_area and pre_post_area respectively. STATIS-

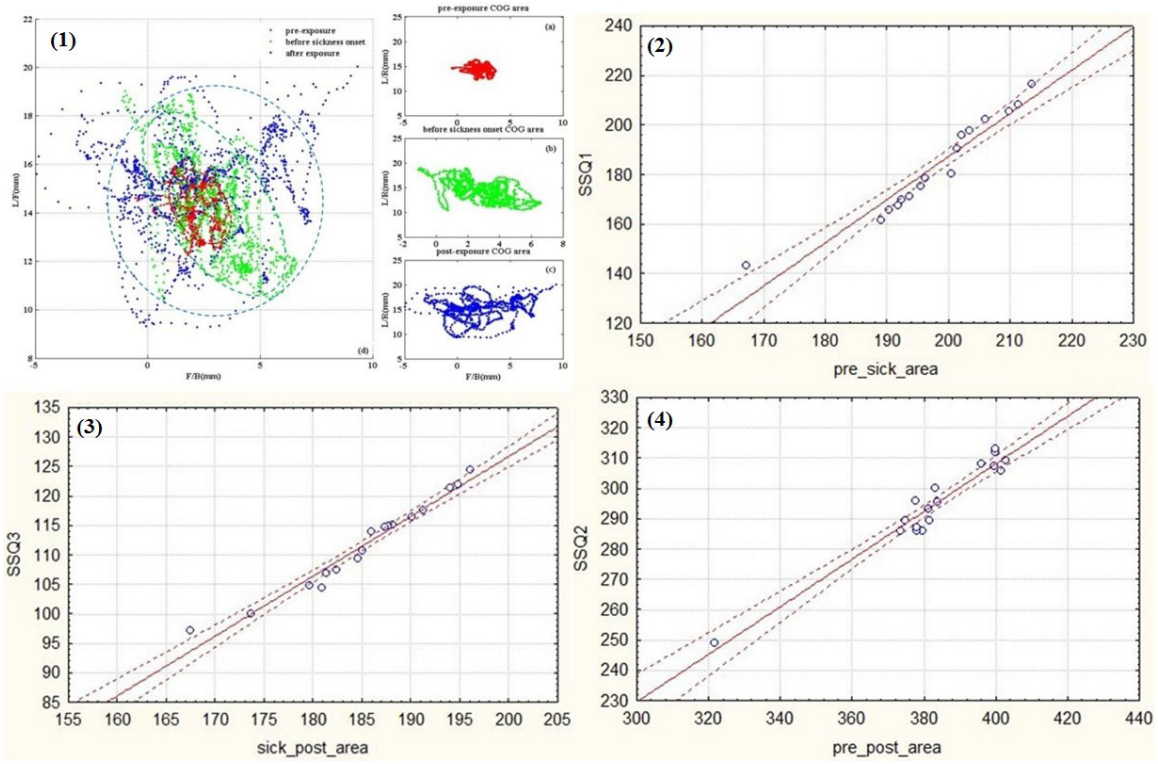


Figure 6.7: Variation of the COG area and the shape during the experiment (1.d), pre-exposure (1.a), at the sickness onset (1.b), post-exposure (1.c), and correlation between the pre-exposure (2), post-exposure (3), at the sickness onset (4) areas and the SSQ total score.

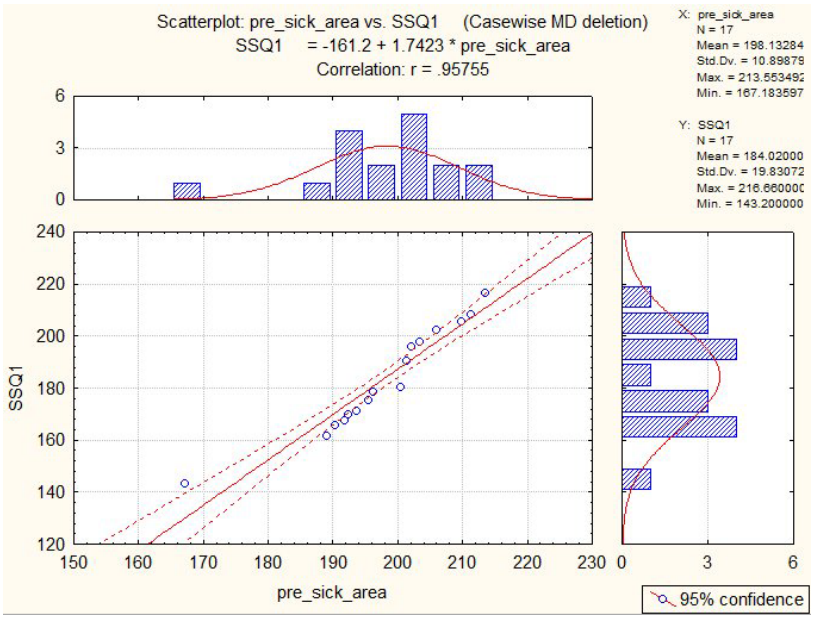


Figure 6.8: Correlation between pre_sick_area and SSQ1.

TICA 8.0 and MATLAB 2013.b statistic toolboxes were used for data analysis. The result of the data analysis shown in Fig. 6.7.2-4 is presented in more detail in Fig. 6.8 through Fig. 6.10. As seen, pre_sick_area ($M = 198.13$, $SD = 10.898$) and SSQ_1 ($M = 184.02$, $SD = 19.83$), pre_post_area ($M = 383.19$, $SD = 18.97$) and SSQ_2 ($M = 294.89$, $SD = 15.3$), and sick_post_area ($M = 185.35$, $SD = 7.44$) and SSQ_3 ($M = 111.8$, $SD = 7.7$) are highly correlated, $r = 0.96$, $r = 0.97$ and $r = 0.98$ respectively.

The mean of the dilated area (sick_post_area$= 185.35$ and pre_sick_area$= 198.13$, pre_post_area$=$ 383.19, as mentioned above) and the corresponding SSQ scores ($SSQ_1 = 184.02$, $SSQ_2 = 296.42$, $SSQ_3 =$ 111.34) are calculated across the subjects. The SSQ score versus the COG area dilation is depicted in Fig. 6.11 for three instances. As seen, the COG area becomes larger with an increase in the sickness score simultaneously, but not with the same rate necessarily, which means the COG area is a good feature for VIMS detection.

The COG area is plotted at 8 points for two example participants to explain how the geometric shape of the COG area varies with VIMS (Fig. 6.12). In the first example, the area starts with an elliptical shape (Area$= 234.38$ mm^2) before sickness, the shape remains an ellipse (Area$= 423.13$ mm^2) during the 7th step and the shape switches from an ellipse to a circle (Area$= 553.41$ mm^2) immediately after the sickness onset. In the second example, the ellipse to circle alternation occurs in the 5th step, however the dilation of the area carries on till the end of the experiment.

An ellipse can be recognized from a circle using $R = \frac{r_2}{r_1}$ fraction, where r_1 and r_2 represent the short and long radius. If $\frac{r_2}{r_1} = 1$ the shape is a circle, otherwise ($\frac{r_2}{r_1} \neq 1$, $r_2 \neq r_1$) it is an ellipse. The

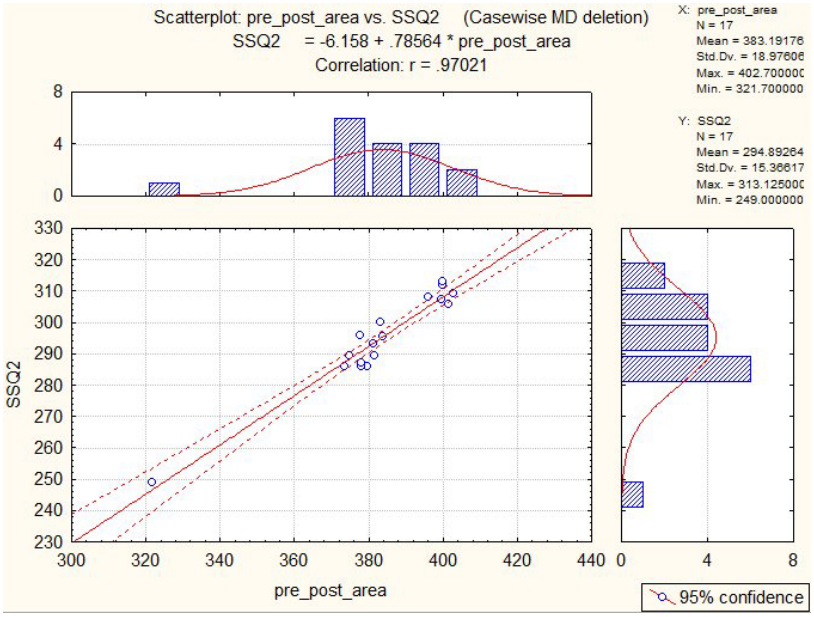


Figure 6.9: Correlation between pre_post_area and SSQ2.

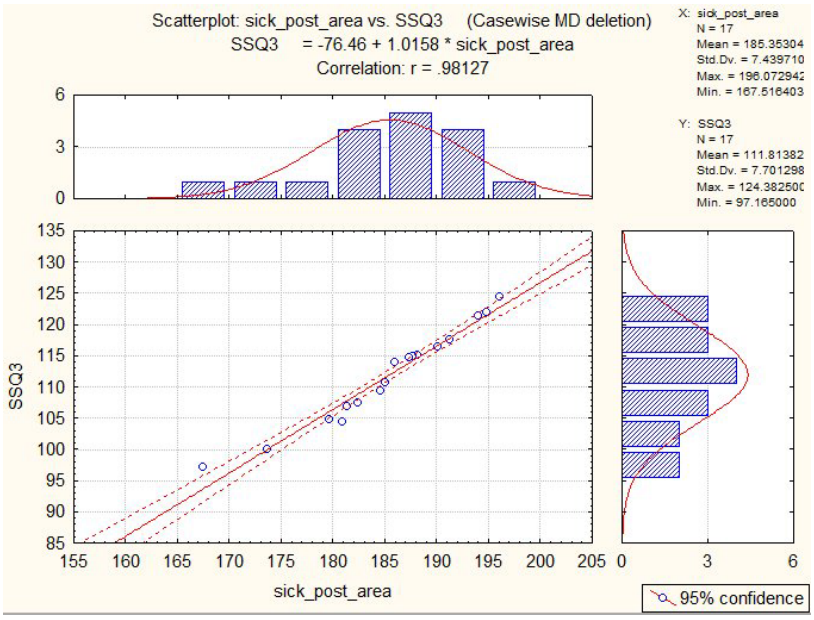


Figure 6.10: Correlation between sick_post_area and SSQ3.

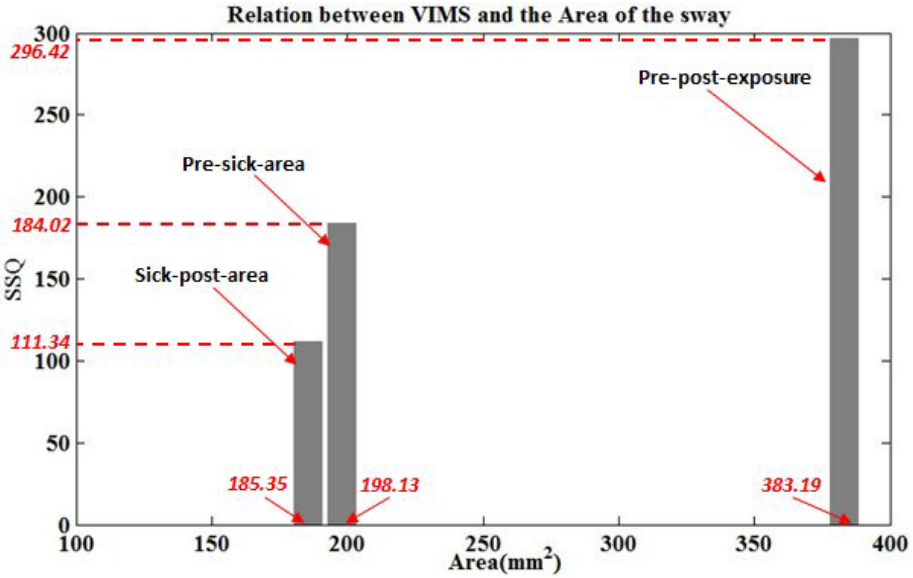


Figure 6.11: Relation between the area dilation and the VIMS score.

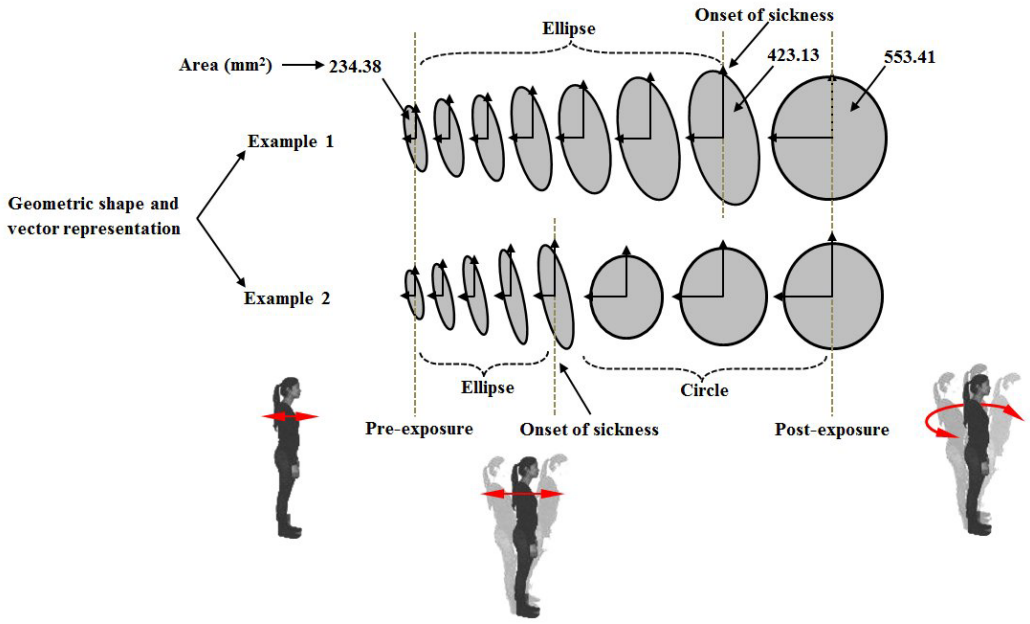


Figure 6.12: Shape variation of the COG area before and after the sickness onset increase.

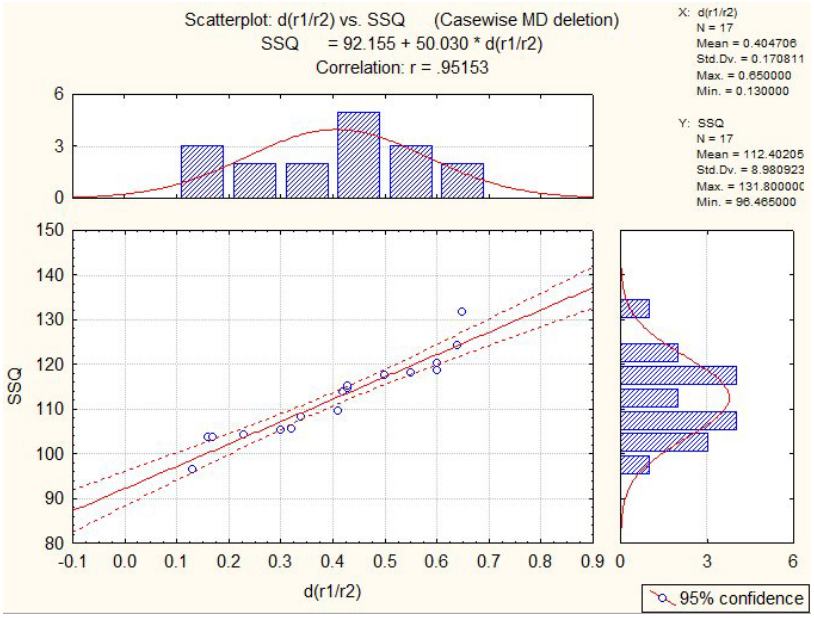


Figure 6.13: Correlation between the COG shape and the VIMS score.

fraction and the SSQ total score are calculated for the sickness onset (SO) and one step after (SO+1). Then, the difference between two fractions ($\Delta\left(\frac{r_2}{r_1}\right) = R_{\text{SO+1}} - R_{\text{SO}}$) and two SSQ scores ($\text{SSQ} = \text{SSQ}_{\text{SO+1}} - \text{SSQ}_{\text{SO}}$) are extracted for each participant. Statistical analysis across the subjects shows that the fraction difference ($M = 0.41$, $SD = 0.17$), $\Delta\left(\frac{r_2}{r_1}\right)$ and SSQ ($M = 112.4$, $SD = 8.98$) are highly correlated ($r = 0.95$) as shown in Fig. 6.13. As a result, the COG shape is an efficient indicator of VIMS.

Fraction R is one of the shape descriptor which can be used when r_1 and r_2 are available in the measurement and can be recalculated from measurement. Otherwise, the shape can be alternatively described by a new descriptor $R = \sqrt{\frac{S}{2\pi}}$.

6.2.3 Feature 2: difference between LF and HF components

The F/B sway signals of the experiment are transformed into the frequency space using the Fast Fourier Transform (FFT). The frequency components of the F/B signal corresponding to the one of the participants at three instances, i.e., pre-exposure, at the sickness onset, and post-exposure, are illustrated in Fig. 6.14.1. A curve is fitted over each signal (samples) to make them visually recognizable and distinguishable from one to another. The vertical red dotted line indicates the first frequency component of voluntary movements and the green and blue dotted lines show the first components of involuntary movements. As a matter of fact, the most important criterion is the difference between the frequency components associ-

ated with voluntary and involuntary movements.

In general, the frequency components of the body movement can extend up to 20 Hz however here only $0-3$ Hz has been depicted. This is because the amplitude of the spectrum is almost close to zero for $f \geq 6$ Hz. The pre-exposure F/B signal (in red color) has two frequency components: $0.0-0.36$ Hz, $0.43-0.93$ Hz. Comparing these frequency components with the one that has already been presented in Fig. 6.4 (see section 6.1.1, frequency components in the stable state: $0-0.5$ Hz) makes this point clear that these components are related to voluntary movements because they are smaller than 1 Hz. The F/B signals associated with the sickness onset (in green color) has two frequency components for voluntary movements, $0.0-0.36$ Hz, $0.5-1.2$ Hz, and one component for involuntary movements, $1.18-2.63$ Hz. Referring to the previous discussion, the later components will be responsible for VIMS as illustrated in Fig. 6.4 and discussed in section 6.1.1. The post-exposure F/B signal (in blue color) has three components for voluntary movements, $0.0-0.4$ Hz, $0.45-0.79$ Hz and $0.85-1.2$ Hz, and one component for involuntary movements, $1.75-2.85$ Hz. The frequency components associated with voluntary ($f \leq 1$ Hz) and involuntary ($f > 1$ Hz) movements will be referred to as Low Frequency (LF) and High Frequency (HF) components.

Though, the first components of involuntary movements, $0.0-0.36$ Hz, in the pre-exposure signal are selected as the LF component (LF_{pre}). However, the average of the first components in the three signals ($0.0-0.36$ Hz, $0.0-0.36$ Hz and $0.0-0.4$ Hz) also can be selected as LF which does not make really a big difference (their first spectrums expand approximately on the same interval). The last components of the sickness onset (HF_{SO}), $1.18-2.63$ Hz, and the post-exposure signal (HF_{post}), $1.75-2.85$ Hz, are selected as the HF components.

It needs to be shown that the difference between the LF and HF components is significant across the subjects. It means always the difference between the LF and HF components remains significantly large. The difference between LF_{pre} and HF_{SO}, $\Delta f_1 = 1.80$ Hz, is significant ($F(1,16) = 85.71, p = 0.0048$). Statistical data analysis using an F-test shows that the difference between HF_{post} and LF_{pre} is quite significant, $\Delta f_2 = 2.13$ Hz, ($F(1,16) = 98.21, p = 0.0032$) and the same for the difference between HF_{post} and LF_{SO} ($F(1,16) = 94.03, p = 0.012$). The difference between each two components is calculated, Df1 ($\Delta f_1 = \text{HF}_{\text{SO}} - \text{HF}_{\text{pre}}$), Df2 ($\Delta f_2 = \text{HF}_{\text{post}} - \text{HF}_{\text{pre}}$), Df3 ($\Delta f_3 = \text{HF}_{\text{post}} - \text{HF}_{\text{SO}}$), for each participant. The difference between the HF and LF components is important for this study, however the difference between the HF components can be interesting since the HF components of the sickness onset is lower than the HF components of post-exposure. Now, it has to be proven that the distance between the LF and HF components is correlated with the SSQ total score.

SSQ1, SSQ2 and SSQ3 have the same definition as explained in section 6.2.2 and are calculated by the same formulation for each participant.

Statistical analysis shows that Df1, Df2 and Df3 are highly correlated with SSQ1 ($r = 0.96$), SSQ2

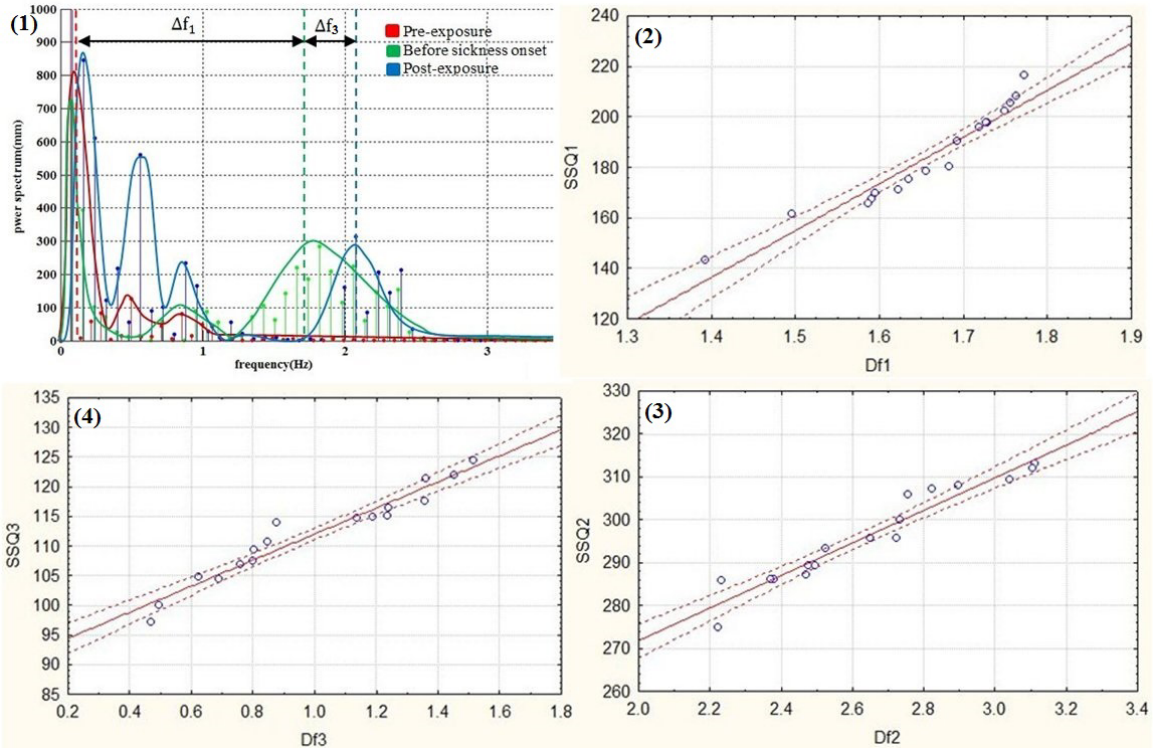


Figure 6.14: Frequency components of the F/B sway signals for the pre, post-exposures and before the sickness onset instances.

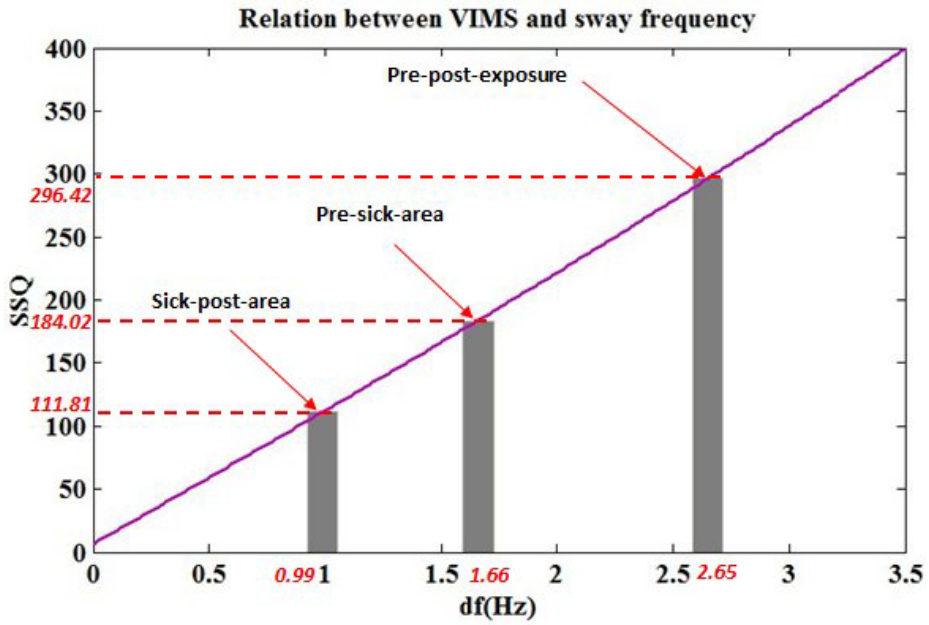


Figure 6.15: VIMS grows when the difference between the HF and LF components of a postural sway increases.

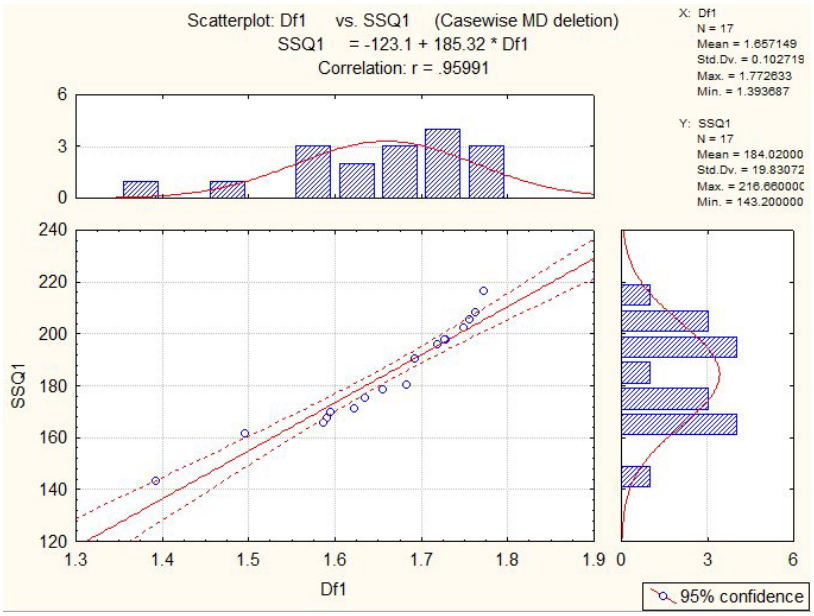


Figure 6.16: Correlation between Δf_1 and SSQ1.

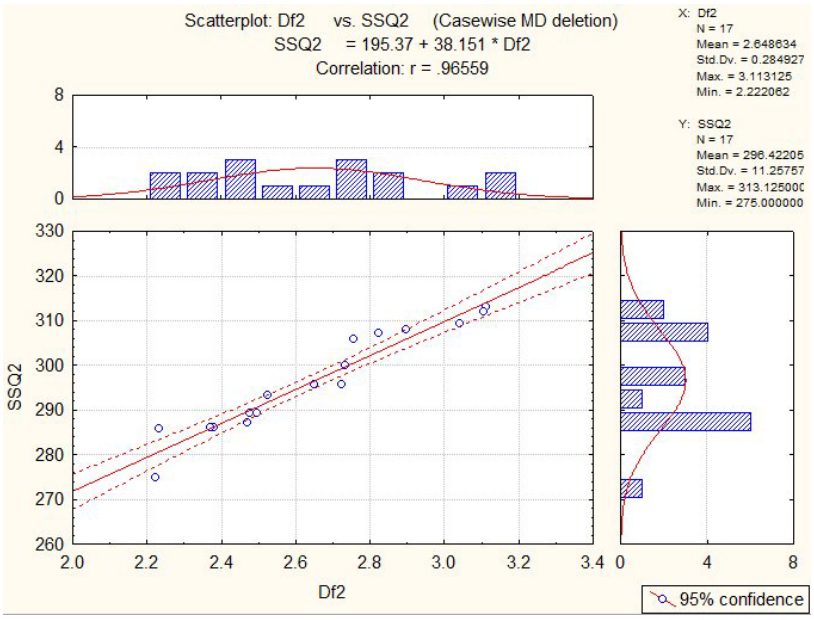


Figure 6.17: Correlation between Δf_2 and SSQ2.

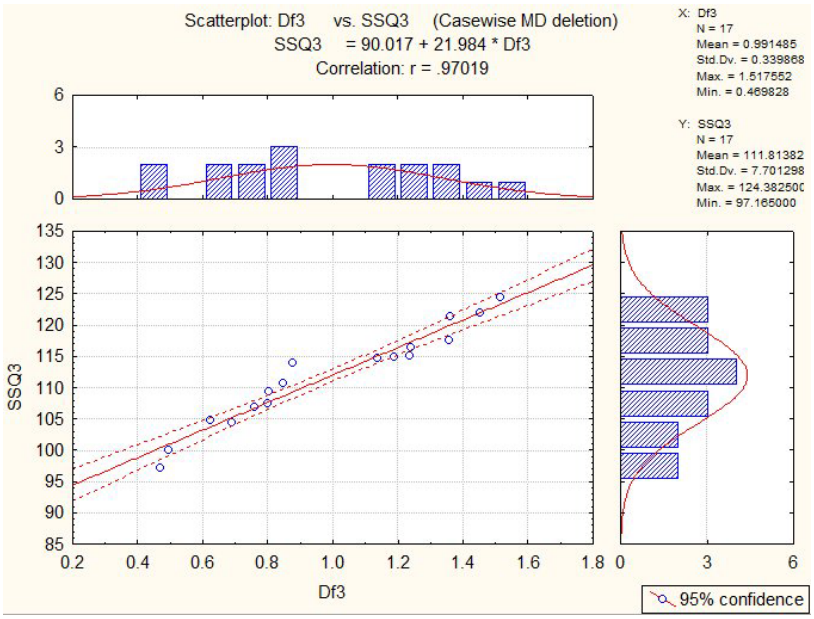


Figure 6.18: Correlation between Δf_3 and SSQ3.

($r = 0.97$) and SSQ3 ($r = 0.97$) respectively. The correlation between the SSQ score and the HF-LF difference is shown in Fig. 6.14.2 through Fig. 6.14.4 respectively. Fig. 6.16 through Fig. 6.18 detail the results of the data analysis. As seen when the difference between HF and LF increases, the SSQ score grows simultaneously. The results of the analysis are summarized in Fig. 6.15, which shows, when the difference between the LF and HF components of a postural sway increases (0.99 Hz$\rightarrow$ 2.63 Hz), the VIMS score grows ($111.81 \rightarrow 296.42$) which in turn means the difference between the LF and HF components is a very effective feature for detecting a VIMS occurrence and estimating the amount of sickness.

In fact, three different features, i.e., the area, the shape and the frequency components of a postural sway explain different aspects of the same signal for that reason it is logical to think they are all correlated with SSQ score.

6.2.4 Feature 3: binocular disparity

The experiment setup and test procedure are modified to demonstrate that VIMS can be detected and estimated quite reliably and effectively by binocular disparity. In this experiment, the visualization system still is the CAVE, with the compartments that has been detailed in section 5.1 of chapter 5. The AR-tracker calculates the user position inside the projection space and an ART Fly-stick2 is used as the navigation device (six buttons, one joy-stick, a tracker and a position sensor).

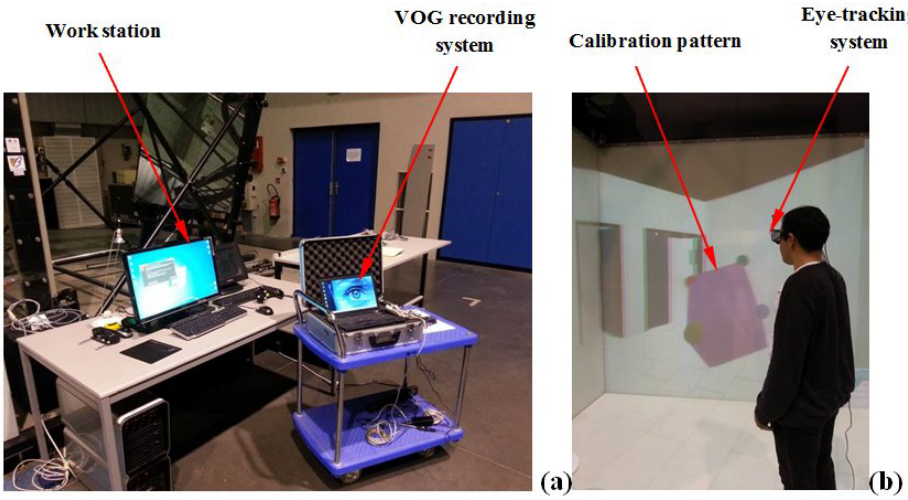


Figure 6.19: (a) Real test setup for the experiment and (b) calibration pattern for the calibration of the SMI system.

Measurement sensor

SMI glasses (designed by SMI Vision Inc. [SMI] and referred to as Video-Oculography, VOG, in this manuscript) records left/right eye movements and the 3D scene simultaneously. The glasses provide a set of noise free psychophysiological measurements for biological, psychological and cognitive studies. The SMI sensor (Fig. 6.20) is glasses consisting of two interior cameras mounted on the inner side of the frame, capturing images from the left/right eyes, and one exterior camera right in the middle of the glasses, looking outward and recording images from the scene. Sample data is transferred to a laptop computer optimized for SMI glasses and equipped with image/signal processing toolboxes.

The end-user wears the VOG glasses with a tracker mounted on top of it (Fig. 6.19.b). A calibration procedure is conducted to obtain the best possible calibration for the interior and exterior cameras of the VOG glasses. A 3D cubic pattern, as demonstrated in Fig. 6.19.b, is projected into the CAVE and the participant is asked to look at the red, green and blue balls at the top corners of the cube during calibration (3 points calibration). The accuracy of the calibration is double checked by asking the user to look at a yellow ball at the left bottom corner of the pattern (Fig. 6.19.b).

Several parameters including the pupil diameter, the center and the position of the eye in the eye image plane (interior camera, Fig. 6.21.a), the gaze position in the scene image plane (exterior camera, Fig. 6.21.b), the calibration detail and three types of eye movement, i.e., “saccade”, “fixation” and “eye blink”, are extracted from the images. The human eyes can reflect and absorb the emitted light, similar to a mirror or any optic devices. This is the principle of the pupil counter detection used in the SMI sensor. As seen in Fig. 6.20, six LEDs were installed in a precise configuration inside the SMI glasses. The eye reflects the emitted light from the LEDs into the vision sensor (interior camera), as seen in Fig. 6.20. The pupil



Figure 6.20: SMI eye tracking sensor.

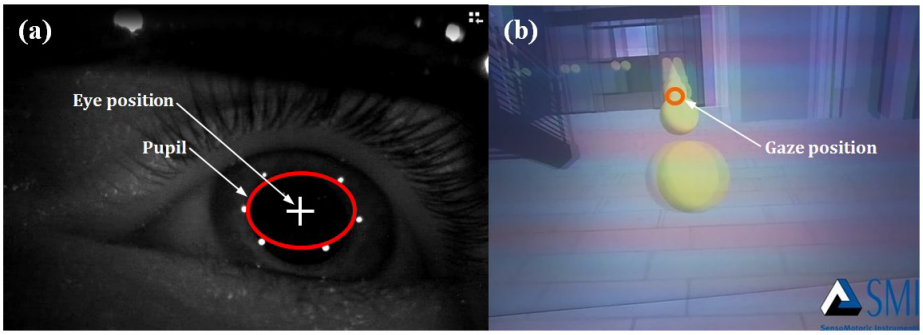


Figure 6.21: (a) Eye position and pupil counter estimation (internal camera), (b) gaze position (external camera).

counter is extracted using the position of these LEDs on the image plane by a precise image processing algorithm. The gaze position is calculated using the eye position and the camera calibration parameters, and superimposed on a scene image captured by an external camera as shown in Fig. 6.21.b. The gaze shows the place and the direction where the participant looks. The two interior cameras and the exterior camera are recording images synchronously with a simple rate of 30 Hz as reported in the data acquisition sheets. The final calculation result sheet is reported in Excel which makes further process and data analysis extremely easy. The Excel sheet can be easily exported to other processing toolboxes such as MATLAB, SPSS, STATISTICA and so on.

Objectives and hypotheses

Based on the theory and relative work on VIMS, introduced in sections 2.6 and 2.7 of chapter 2, the objective of this study is to test whether the effect of the velocity of a visual flow on VIMS during rotational movements inside a real-scale VE is primarily higher than translation movement. Besides, we would like to see if binocular disparity is a good criterion to detect VIMS. To this end, the following hypotheses are studied:

1. The participants will report a significantly higher sickness score during navigation in a VE (H1a), the score associated with oculomotor is expected to be higher than disorientation and nausea (H1b).

2. The difference between the left and right eyes during rotational movement is significantly higher than translational movement (H2), the difference between the two eyes during rotation is highly correlated with the total sickness score reported in the SSQ.

Experiment design and test procedure

The experiment setup permits the participants to navigate inside a building model (Fig. 6.22.a). The projection room (the place the CAVE is located in) is completely darkened during the exposure and all lights are turned off except that from the display projectors. During navigation, standing participants are asked to follow the test path illustrated in Fig. 6.22.a using a Fly-stick within 5 minutes maximum. Similar to the previous experiments presented in section 6.2.1, the path indicators (yellow balls in Fig. 6.22.b) with the same size and color are set to 1.25 m above the ground. Navigation is designed to be an exhaustive combination of rotational (with a rotating velocity of 60 degrees per second, dps) and translational (with a linear velocity of 2.5 meter per second (mps)) movements. A 60 dps rate is chosen because the highest level of VIMS was reported in this speed (see Hu et al.[73]). The translational velocity is set to 2.5 mps to lower sickness. The 3D scene is updated in the projection space based on the new position of the user instantaneously with a high frame rate (nearly 60 fps for the selected model). This is important as any

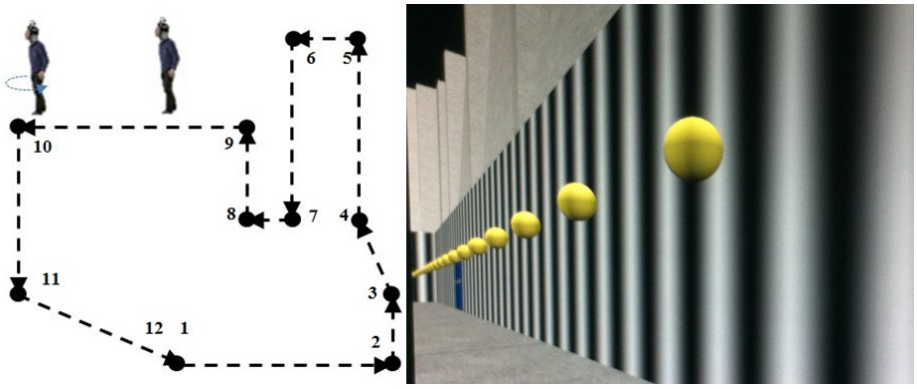


Figure 6.22: (a) Path planning for the experimental test (maximum rotation angle: 90°, minimum: 30°, speed of rotation: 60 dps), (b) stimulation pattern (the yellow balls indicate the path).

response delays commonly found in a head-tracked virtual reality (VR) system can significantly affect user performance and questionnaire scores taken subsequently.

Participants

11 participants aged from 27 to 38 (6 male and 5 female) took part in the experiment. They were either with normal eye sight or corrected eye sight. They all attained a 10/10 visual acuity. Before the experiment, they were informed of the general aim of the study but not the specific objectives and hypotheses. They needed to complete a pre-exposure SSQ (Kennedy et al.[98]) after the VOG calibration and before a 5 minutes exposure. Should a participant report a total pre-exposure SSQ total score of 10 or more, he or she would be asked to rest for 5 minutes and fill in another pre-exposure SSQ. If the total score is still greater than 10, he or she will be asked to come back on another date. The post-exposure SSQ is filled at the end of the experiments (point 12 in Fig. 6.22.a).

Data analysis and feature study

Raw sensor measurements include the left/right eye images, in contrast image, and camera calibration data. The rotational/translational commands of the navigation device are recorded synchronous with the SMI glasses using time stamps. The left and the right images were analyzed and selected during translation (Fig. 6.24.1-2) and rotation (Fig. 6.24.3-6) using the time stamps. The two images are put side-by-side using image processing techniques to display the eye situation at each time stamp. The images are superimposed by two reference lines which show the difference between the two eyes move evidently.

As shown in Fig. 6.24, the difference between the left and right eyes with respect to the reference lines is not very noticeable (Fig. 6.24.1-2). The first two images in Fig. 6.24 belong to the end of the translational

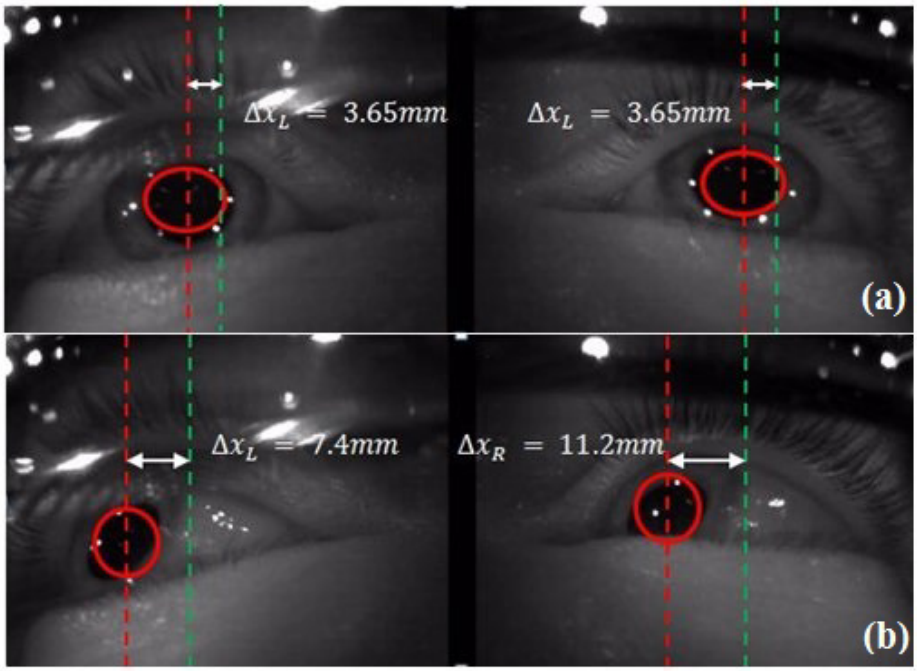


Figure 6.23: (a) Left/right eye position and difference between the two eyes during translational movement, (b) difference between left/right movements immediately (200 ms) after rotation of the scene (the right eye moved 3.8 mm more than the left eye).

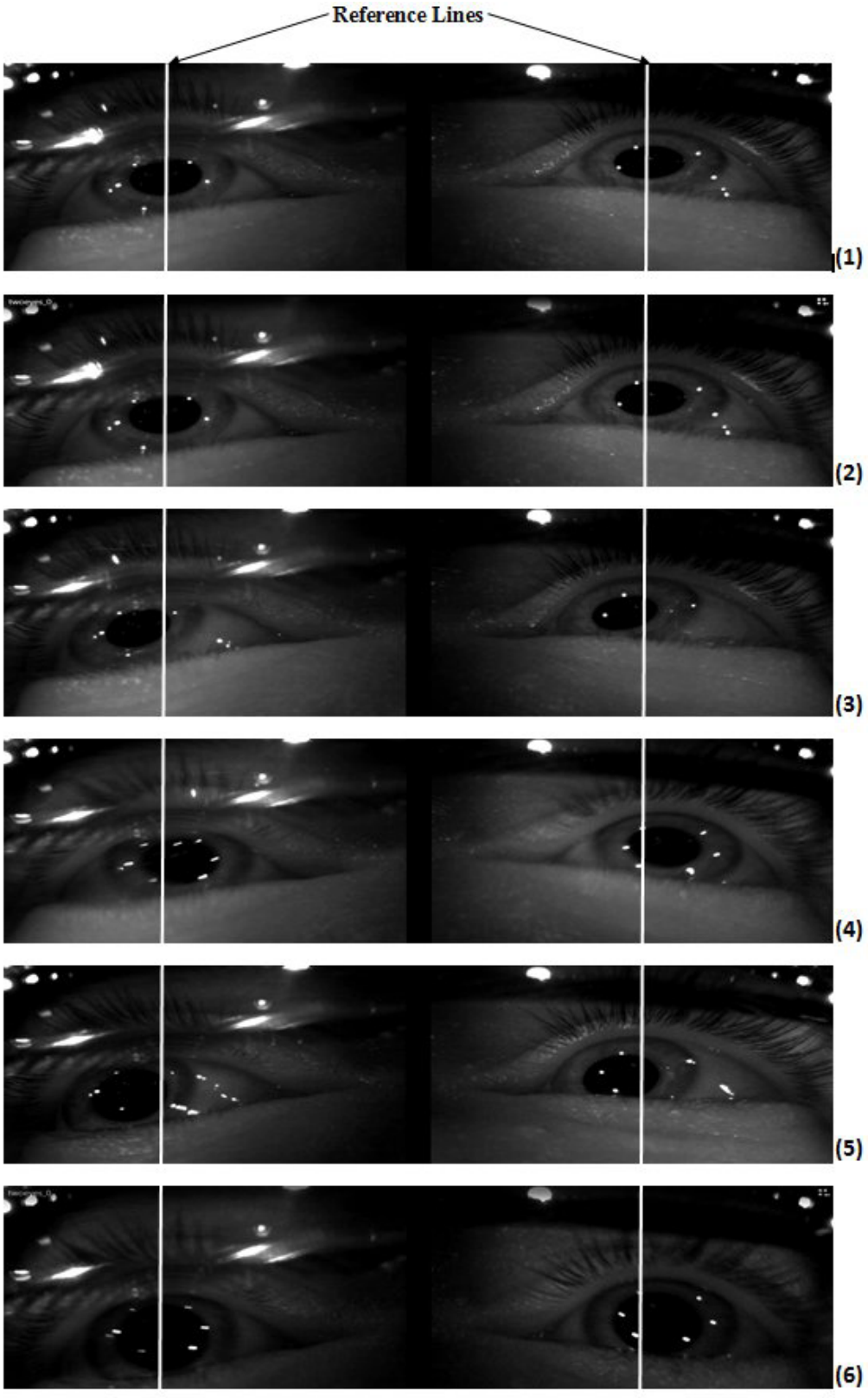


Figure 6.24: Difference between the left and right eyes during navigation with respect to the reference lines, (1-2) during translation and right before rotation, (3-6) immediately after rotation.

movement, right before the activation of the rotational command. This selection is done on purpose and the reason will be explained later. On the contrary, the difference between the two eyes is significantly noticeable during rotation as shown in Fig. 6.24.3-6. The difference is calculated for two samples during translation (Fig. 6.23.a) and rotation (Fig. 6.23.b) to give a numerical estimation concerning the amount of the difference mentioned above. The eye position is defined as the center of the pupil on the eye image and is measured in pixel in the image coordinates (see the SMI Vision manual[SMI]). The pupil is located within the area enclosed by the LEDs (highlighted in red in Fig. 6.23.a, b). The position of the left and right eyes before (during translation) and during rotation is shown in Fig. 6.23.a and Fig. 6.23.b, respectively. The red dot-line shows the position of the eyes at the current time stamp and the green dot-line illustrates the position of the eye in the previous time stamp (the time difference between two stamps is $\Delta t = 30$ ms). Nearly, there is no difference between the left/right eyes during translation, however this difference is 3.8 mm for rotation.

Regarding the previous discussion in chapter 2 (see section 2.5.7), during navigation in virtual environments, binocular disparity is deteriorated and leads to VIMS. However, previous observations shows that this disparity during rotation is higher than during translation. Therefore, it is logical to hypothesis that the induced sickness is more due to the rotation than the translation.

The calculated parameters of the SMI glasses (Excel sheets) is used to validate this hypothesis. In fact, the position of each is extracted from the test sheets and depicted versus time for each participant. Fig. 6.25.a, b and c illustrate left and right eyes movements and the difference between the two eyes (residual signal) during navigation. Fig. 6.25.d shows the rotation points (12 turning points for the given example), where a participant commands the rotational movement, and the duration of the command at each point. For the given example, intervals $[t_n\ t_{n+1}]$ and $[t_{n+1}\ t_{n+2}]$ are the duration of the translation and rotation commands at point $n = 3$, respectively (Fig. 6.25.d). Most of the rotation points are in accordance with the navigation path as shown in Fig. 6.22.a. However, the user is not restricted to 11 rotation points along the path. If he/she makes a mistake and selects a wrong path, he/she can correct his/her path by extra rotational commands.

The pre (at the beginning) and post-exposure SSQ scores are calculated for each participant. The value associated with rotational (stripped red in Fig. 6.25.c) and translation (non-stripped) movements is calculated and accumulated for rotation and translation separately (two values) for each participant. The primary result of the statistical analysis using a T-test shows the difference between the two eyes during rotation ($M = 140.7$, $SD = 95.6$) is significantly higher ($t(10) = 4.25$, $p < 0.05$, $p = 0.0152$) than the difference during translation ($M = 97.8$, $SD = 53.8$).

To be sure that the significant difference between the two eyes is due to the navigation mode and not an eye-blink artifact, eye-blinks are extracted from sensor data and plotted together with the difference (Fig. 6.26). As seen, none of the eye blink happened at the same time with the peaks. This double-check has

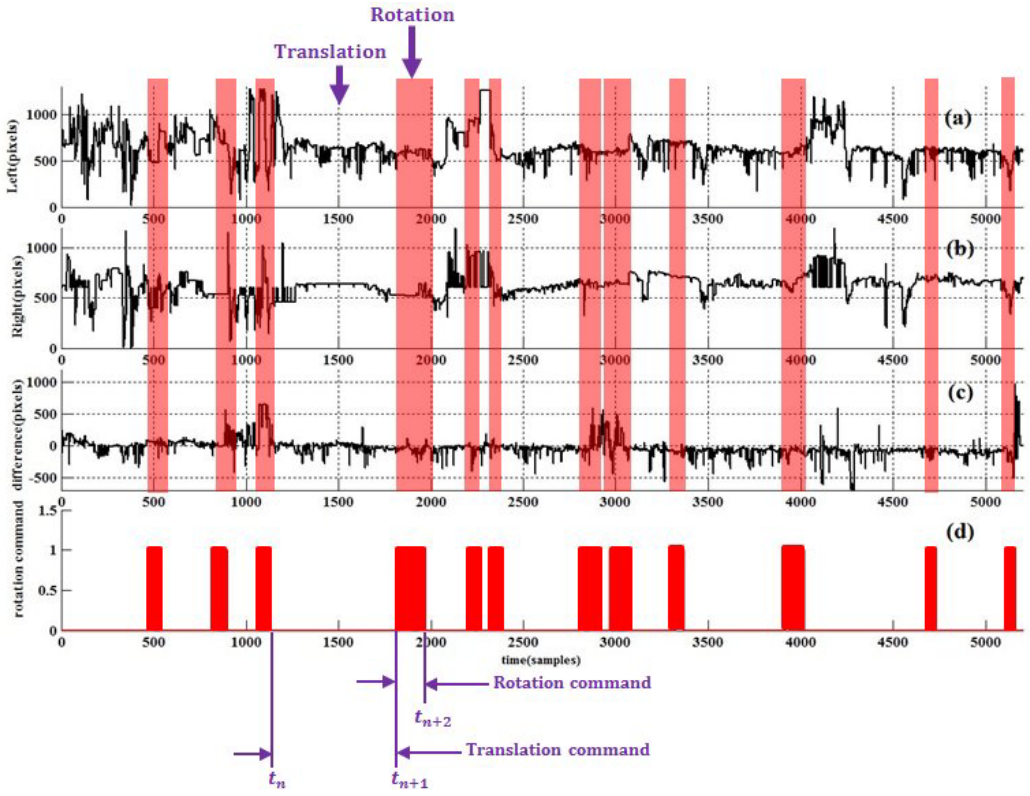


Figure 6.25: (a) Right and (b) left eyes position, (c) difference between the two eyes and (d) rotation command versus time.

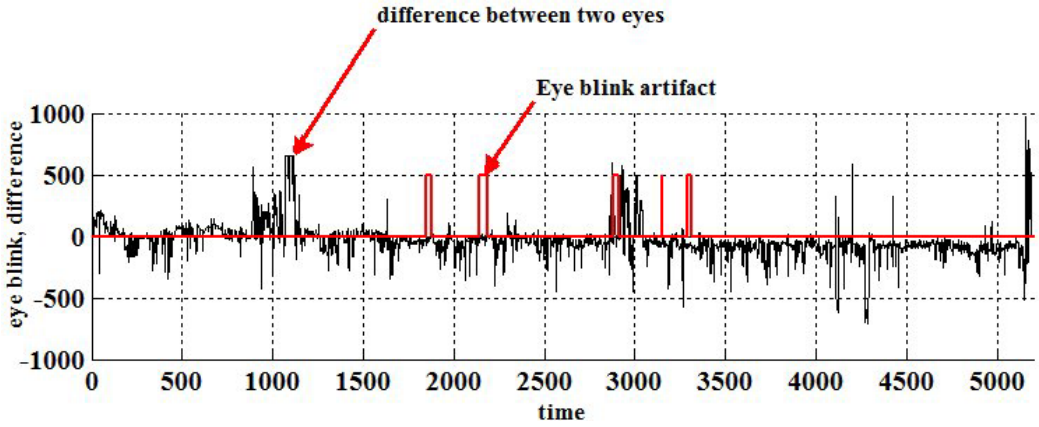


Figure 6.26: Eye-blink and residual signals.

been carried out across the subjects (for all the participants) and the same results were observed. However, in case an eye-blink and a peak appear at the same place during translation/rotation, the corresponding value is removed and replaced with the mean of the neighboring values. Nevertheless, few eye-blinks do not affect the statistical analysis noticeably.

As mentioned above, two types of measurement were recorded for evaluating the level of the sickness induced by a visual stimulus, i.e., the difference between the two eyes and SSQ. The total VIMS score for each subject is extracted from participant's SSQ reports. Besides, the Oculomotor, Nausea and Disorientation sub-scores are calculated separately using the same SSQ reports by applying a correction factor proposed by Kennedy et al.[98]. Based on the calculation results for each individual, the total VIMS score varies from 31.04 to 512 ($M = 251.62$, $SD = 159.83$). Statistical data analysis using an ANOVA test clearly shows that oculomotor is significantly higher than nausea and disorientation ($F(2, 20) = 10.939$, $p < 0.001$, $p = .00062$, Fig. 6.27). The summary of the ANOVA test is shown in tables 6.4 and 6.5. As seen, there is a significant difference between oculomotor, nausea ($p < 0.01$, $F = 12.296$) and disorientation ($p < 0.005$, $F = 14.55$) but there is not such a significant difference between nausea and disorientation ($p = 0.302$, $F = 1.851$).

In addition, the total sickness scores were calculated by applying the correction factor of Kennedy et al. The correlation between the average residual during rotation ($M = 92.01$, $SD = 52.48$) and the total SSQ level ($M = 251.6$, $SD = 159.8$) is calculated and depicted in Fig. 6.28.

The test result shows that the SSQ score is highly correlated ($r = 0.98$) with the mean of the difference between the two eyes and can be replaced with the SSQ report as a feature to evaluate the level of sickness. The difference of eye movement during saccade and fixation is explored too. The result of this analysis is shown in Fig. 6.29.

The difference between fixation during rotation and translation is not significant ($p = 0.149$, $F =$

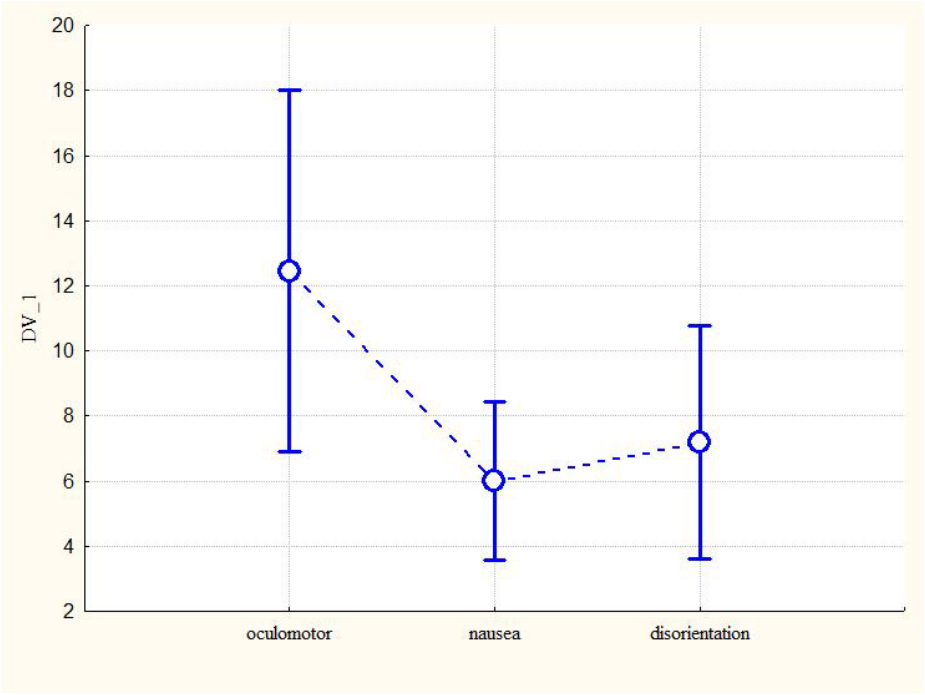


Figure 6.27: Sub-score average across the subjects.

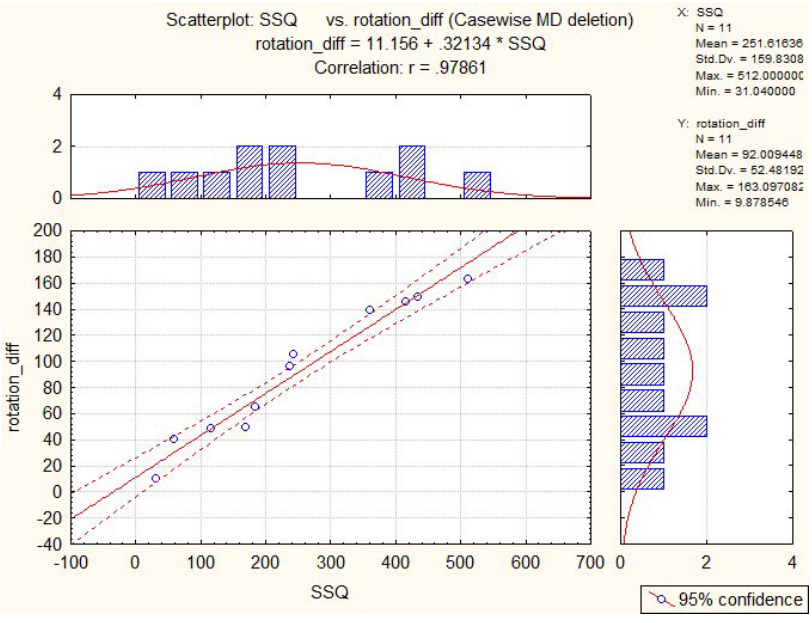


Figure 6.28: Correlation between the SSQ and the difference during rotation.

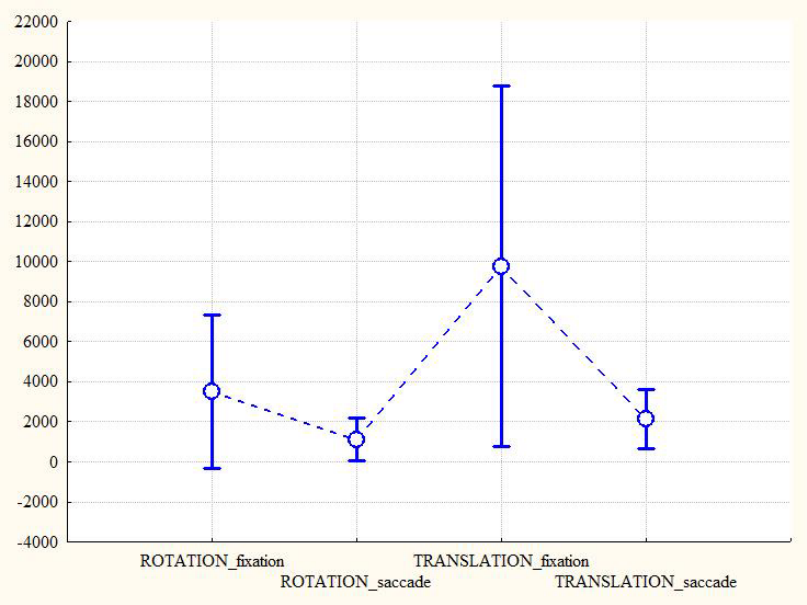


Figure 6.29: Analysis of the difference between the two eyes during saccade and fixation.

Factors	*Mean*	*SD*	*-95%*	*+95%*
oculomotor	12.45455	2.487655	6.911704	17.99739
nausea	6.00000	1.095445	3.559196	8.44080
disorientation	7.18182	1.605775	3.603929	10.75971

Table 6.4: Statistical information of each sub-score in the SSQ.

Factors	*oculomotor*	*nausea*	*disorientation*
oculomotor	-	$p = 0.006/F = 12.296$	$p = 0.0034/F = 14.550$
nausea	$p = 0.006/F = 12.296$	-	$p = 0.302/F = 1.851$
disorientation	$p = 0.0034/F = 14.550$	$p = 0.302/F = 1.851$	-

Table 6.5: p and F values associated with each two pairs of parameters in table 6.4.

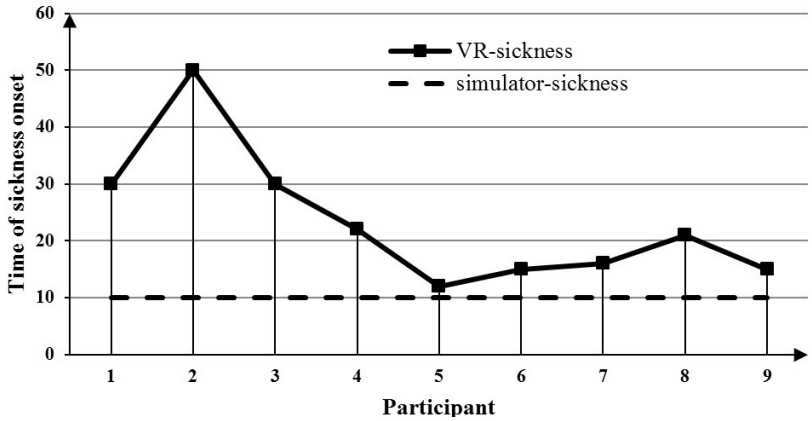


Figure 6.30: Time constant of VIMS and comparison with the time constant of MS for 9 subjects who participated in the experiment of feature 2.

2.625). Similarly, the difference between saccadic eye movement is not significant ($p = 0.0787$, $F = 3.93$) either. However, as seen the difference between rotation and translation due to saccadic eye movement is very close to significant.

6.2.5 Feature 4: expected time of the VIMS onset

Based on previous research[144], MS can appear via two different processes with two different time constants: T_s (short: less than 30 s), T_L (long: more than 10 minutes). Usually the later process with a longer time constant appears because of accumulation effect[144]. From this perspective, MS appears after passing a period of time. During the experiment, we have noticed that VIMS has a long time constant during translational movement, even longer than MS. For that reason, not only the SSQ and relevant features (features 1-3) were measured during the experiments but also the time of the experiments were carefully monitored. For instance, Fig. 6.30 shows the time constants corresponding to the 9 first participants who completed the experiment of feature 2 (solid-line). The measurement shows the average time constant of $T_L = 24$ min ($M = 23.89$, $SD = 11.54$) for translational movement. The total time of the task completion during the study of feature 3 was set to < 5 min. The time constant during rotational movement is $T_s \leq 5$ min ($M = 2.34$, $SD = 1.75$) which means probably the short process ($T_s = 30$ s) of motion sickness appeared during rotational movement too. This is because the experiment is not long enough for the activation of the long process. That is the reason why the participants might get sick immediately during rotational movement.

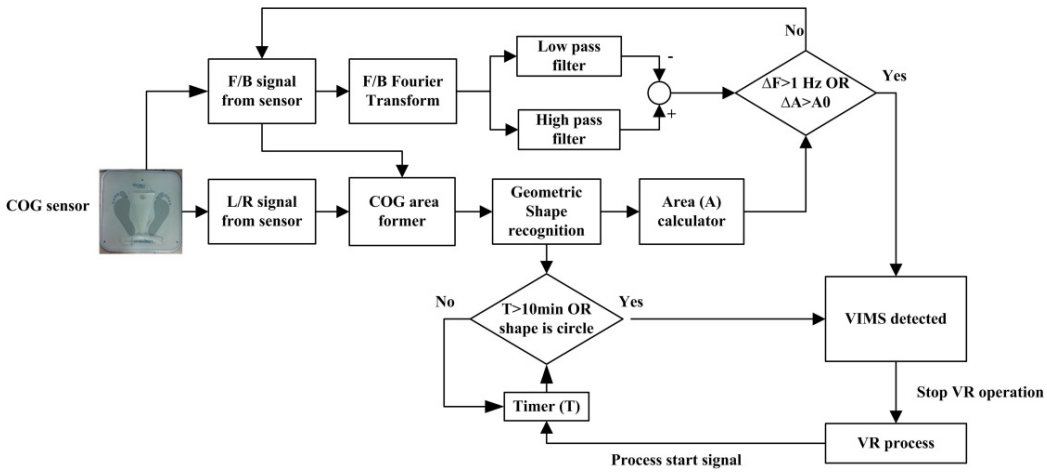


Figure 6.31: VIMS prediction in a real-time application.

6.3 Prediction of VIMS in a real-time process

We had two main achievements in the feature studies and later during the practical experiments: 1) VIMS is time dependent, 2) the amount of the COG area dilation, the difference between the HF-LF components of F/B sway and the COG shape as well as the difference between the two eyes (four features) indicate the level of sickness. Now we can use these two results to predict the onset of VIMS and track the level of sickness during navigation and presence in a virtual environment. When the level of sickness goes above a certain threshold we can inform the end user or stop the process. Fig. 6.31 shows a prediction system which uses only the shape, the area, the frequency components and the time features to predict an occurrence of VIMS.

From an implementation perspective, we can develop an extra module and attach it to the JVM and iiVR platforms. The connection and operation cycle are shown in Fig. 6.32.

6.4 Effect of the navigation parameters on VIMS

Four features including the shape and the area, the difference between the LF and HF components of the F/B sway, the difference between the left/right eyes and the time of the sickness onset were introduced as a set of efficient features to predict and estimate VIMS during navigation in a VE. Sickness emerges due to an inappropriate adjustment of the navigation parameters, the navigation interface, an incorrect selection of the scene contents and so on. The following section studies only the effects of the navigation parameters such as the velocity and the distance as well as the navigation interface on the level of sickness. Six different navigation interfaces were proposed in chapter 3 using various technologies. They can be divided into two main groups: natural and artificial. Now, we will select two navigation interfaces of each group and

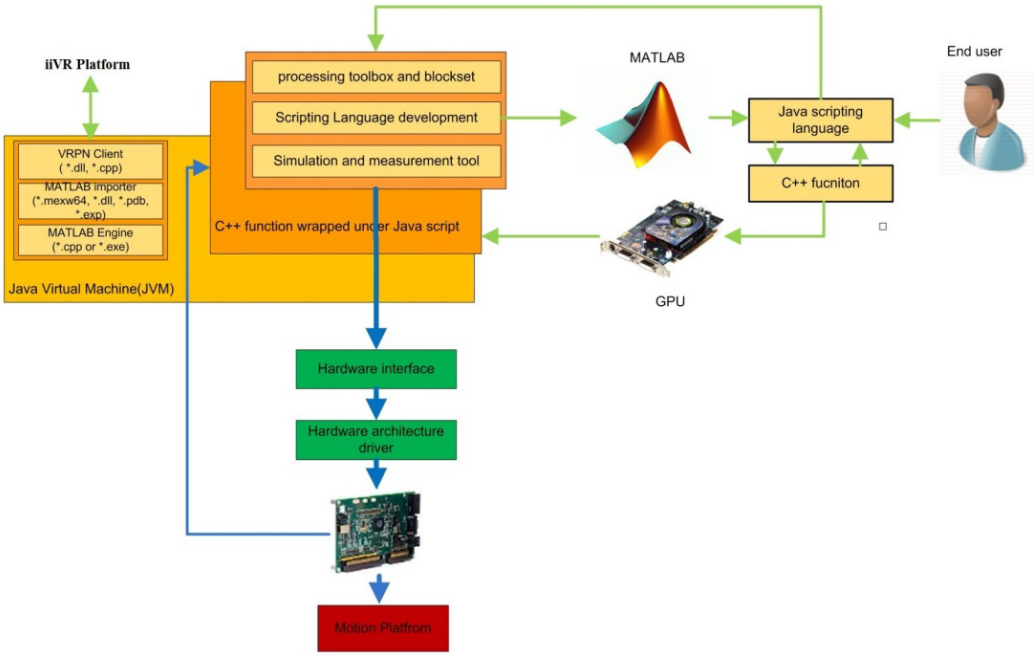


Figure 6.32: Implantation of a VIMS predictor in a real-time process.

compare user performance during navigation with the selected interface.

6.4.1 Navigation interface

Device-Based Navigation Interface (DBNI)

This navigation interface is designed based on the Fly-stick2 (Fig. 6.33) and the navigation principles which have already been explained in detail in chapter 3 (section 3.6.1). The participants are capable of moving forward/backward and turn to the left/right up to $\pm 15°$, if it is required. The Fly-stick has five buttons, one joy-stick handle, a laser-based optic tracker (position and orientation) and a trigger button. Fig. 6.33.b demonstrates the upper view of a Fly-stick with a label on each button. The movement can be accelerated/decelerated by two buttons allocated for this purpose on the Fly-stick (blue buttons 1, 2 with +/- label). The speed variation steps are identical for positive and negative directions (increase/decrease), 0.45 per unit. This value is adapted from trial run experiments and is calculated by $\Delta V = \frac{V_{\max} - V_{\min}}{N}$, where $V_{\max}$, $V_{\min}$, and N represent the maximum and minimum speeds and the number of steps. If 1.8 m/s and 0 are selected for the speed limits and 4 for the number of steps, then the step resolution is $\Delta V = 0.45$. However, all these parameters are adjustable by the user depending on the experiments.

A speed controller was designed to keep the user speed within a certain speed limit. If the user crosses the upper or lower limit then the speed controller will return the speed to the operation range according

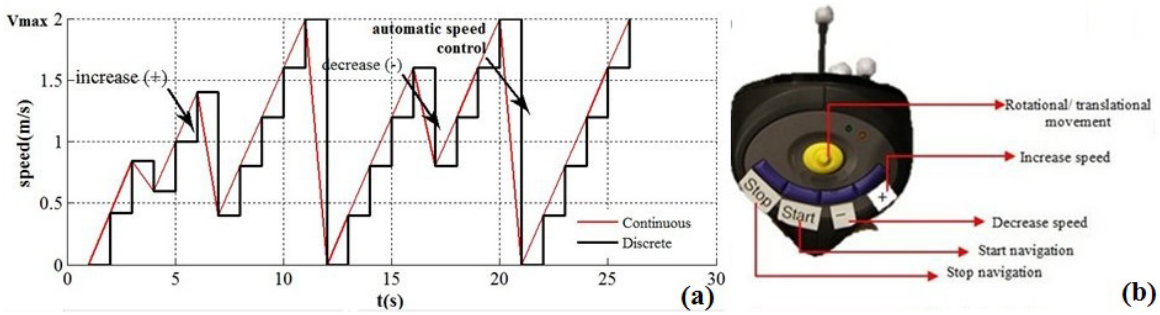


Figure 6.33: a) Mechanism of the speed controller and sequence of its operation (e.g., the speed controller automatically operates at $t = 12$ s and decreases the speed, because the user crossed the topper speed limit, while at $t = 7$ s the speed is decreased by the user manually), b) Fly-stick and navigation sub-task assigned to each button.

to the operation curve shown in Fig. 6.33.a. The "speed controller function" has two modes of operation; automatic and manual. The automatic mode becomes active when the user crosses the upper or lower speed limit. The manual mode functions by a user command and during the experiment keeps on-hold to receive the user command and consequently change the speed. For instance, the user decreases the speed intentionally at time $t = 7$ s and similarly at $t = 17$ s while the speed decreases automatically at $t = 12$ s because of the upper limit crossing in Fig. 6.33.a. The joy-stick handle (yellow handle at the top, Fig. 6.33.b) sends the forward/backward command to the "navigation function".

The rotation is commanded by turning the handle to the left and right (first option). However, the "navigation function" can perform the rotation task by reading the instantaneous orientation of the optic tracker attached to the Fly-stick and redirecting the user movement along the Fly-stick orientation (second option). The navigation task is initiated and terminated when the blue buttons 3, 4 (labeled by "start" and "stop" in Fig. 6.33.b) respectively are pushed by the participant. The navigation interface based on the Fly-stick represents a huge group of interfaces called device-based navigation interface (DBNI). Few examples of this type of interfaces were explained in chapter 3.

Natural Gesture-Based Navigation Interface (NGBNI)

The second group of interfaces based on natural body gesture such as walking, hand and head movements and so on will be presented here. This group is referred to as Natural Gesture-Based Navigation Interface (NGBNI). Here we will use a Microsoft Kinect Xbox 360 that captures the user motion and a FAAST VRPN server[198] that interprets the motion into gesture and streams the gesture on VRPN. As mentioned in chapter 3 (section 3.5), any travel in VEs either is rotation or translation or a combination of both. Translational movement (Fig. 6.34.a) is commanded and controlled by up/down movements of the feet during walking-in-place (WIP), and rotational movement (Fig. 6.34.b) is commanded when the left/right hand moves above the shoulder as explained in chapter 3 (see section 3.6.2) and shown in Fig. 6.34. The user nat-

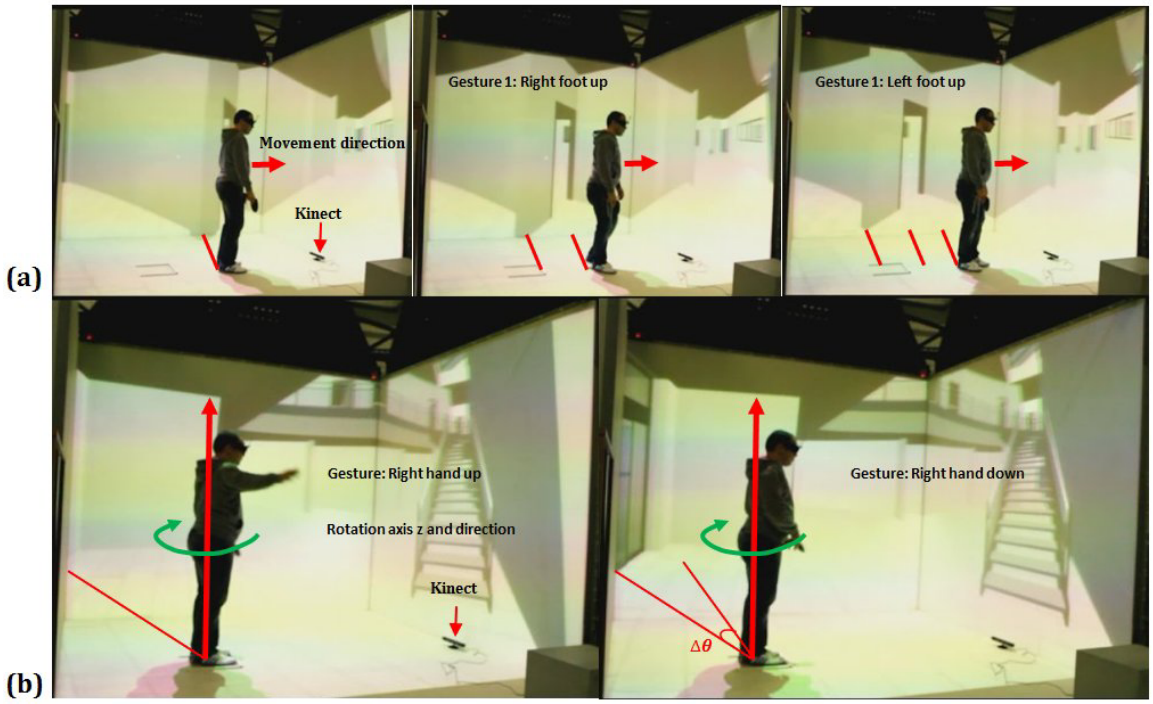


Figure 6.34: Navigation based on natural gesture: (a) walking forward, (b) rotation to the left/right.

ural walking controls the forward/backward movement speed which means, when the user walks faster the movement is faster and vice versa. The rotational speed is controlled by the height of the hand which means when the hands go above the shoulder the scene rotates faster and other way around.

VIMS can appear due to various situations[17] such as the texture quality, an unsuitable velocity and acceleration of the visual flow, self-motion and so on. The effect of the speed, the distance and the navigation interface on VIMS will be studied in three experiments while some of the previous features will be used for the evaluation, validation and verification.

6.4.2 Experiment I: effect of the speed variation on VIMS

This experiment aims at studying the effect of the velocity variation on the level of VIMS during navigation while keeping the scene texture and distance from the virtual scene constant for all the participants. This study will focus only on translational movement. Translational movement is defined as a straight movement from one point to another without doing any rotation along the path in between. A translational movement in VEs can be simply expressed by $V_L = f_L(v, d)$ where, independent variables v and d represent the velocity (m/s) and the distance (m) from a virtual barrier respectively (distance between the participants and VE objects or scene, see Fig. 6.35).

Participants and questionnaires

Seventeen subjects (13 males and 4 females: 31.58 ± 12.69 years, 74.65 ± 15.22 kg), participated in the experiment. There was a briefing to give enough information about the test procedure and possible risks before each experiment individually. All the subjects participated voluntarily in the experiments. A pre-exposure questionnaire (Q1) was obtained from each subject to know their backgrounds and to evaluate their health condition (see appendix A). The result of the questionnaire showed that there was no test subject whose participation in the experiment would be unsuitable due to health issues. Kennedy's SSQ questionnaire with the same scoring method as detailed in[98] (appendix B) applied to the experiments to calculate the SSQ score.

Experiment design

A row of balls were lined up in parallel with a virtual wall as indicated by "path 1" (Fig. 6.35.a). These balls were demonstrated in yellow and named "path indicator" in Fig. 6.35.a. The distance between path 1 and the virtual wall is set to 2 meters as shown and tagged by D_1 in Fig. 6.35.a (the real setup is shown in Fig. 6.35.b). The path indicators are set to 1.25 m above the ground to keep the user in a constant distance from the virtual barrier. A Fly-stick is used to navigate along the path using the DBNI navigation definition.

Test procedure

The participants navigated along path 1 during the experiment as follows:

1. The COG of each participant is recorded for t_1 s (pre-exposure measurement).
2. The participants navigate along path 1 (detailed in Fig. 6.35.b) for t_2 s with a speed of $v = 2$ m/s.
3. The participants increase the speed one step (by pressing button 1, Fig. 6.33.b) and navigate for t_2 s.
4. Step (3) is repeated 6 times for equal periods of t_2 s.
5. The post-exposure COG of the participants is recorded for t_1 s.
6. An SSQ (Q2) is filled out and the test is terminated.

The test procedure was planned, "scheduled plan", to be performed in 7 steps. However, the examiner could stop the experiment if the participants would not feel good (if they felt abnormal uneasiness in their stomach and head) or the sickness score changed abruptly similar to that explained above (section 6.3). We did not have any case with serious sickness during the experiment.

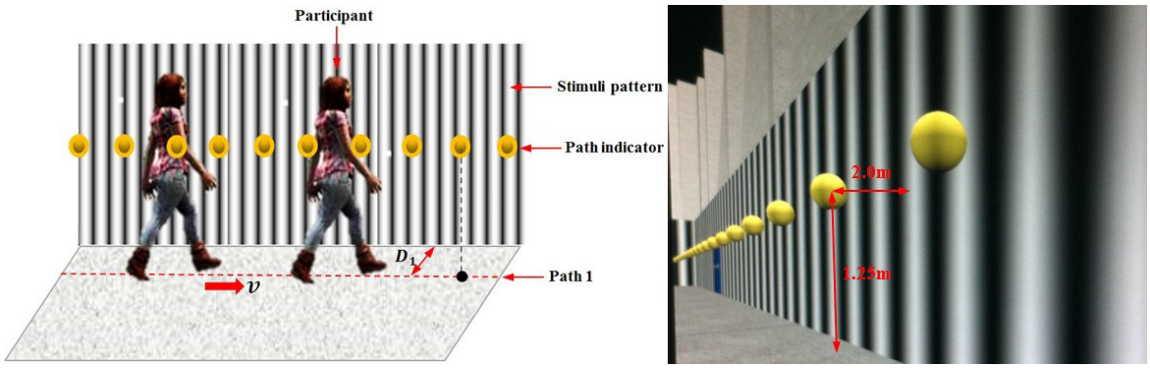


Figure 6.35: (a) Test parameters (velocity, distance= D_1 constant), path planning for navigation inside a 3D VE, (b) real path setup inside a scale-one 3D display.

Statistical data analysis

The level of post-exposure sickness for 17 participants is calculated using the SSQ report and Kennedy's correction factor[98] (the sickness level in pre-exposure is zero). Statistical analysis using an F-test shows that the incremental variation of the speed significantly increases ($F(1,16) = 121.86, p = 0.042$) the SSQ scores ($M = 325.56$, $SD = 47.15$). The speed variation leads to a noticeable post-exposure area dilation with respect to the pre-exposure area ($F(1,8) = 133.86, p = 0.0015$). The same effect has been observed on the COG shape ($F(1,8) = 112.21, p = 0.0054$). At the same time, the difference between the LF and HF components of COG sway significantly increases ($F(1,8) = 29.41, p < 0.0001$). The difference between the LF and HF components, the COG area dilation and a higher level of sickness appear only due to a velocity increase. Based on the pre-test questionnaire (Q1) most of the selected subjects (13 out of 17) were daily computer user and playing computer/video games at least once a week. This experience can affect the level of the SSQ score, however the fact that sickness increases due to a velocity increase is unavoidable.

The model proposed by Oman[144] (see chapter 2, section 2.7) helps to understand why the SSQ score increases by modifying the navigation parameters. When the participants are navigating inside a real-scale 3D model, due to the conflict between proprioceptive data (stored pattern, Purkinge cell effect) and measurement of the sensory organs, K (Kalman gain in the Bayesian model, red color in Fig. 6.36) and $\hat{\mathbf{S}}$ matrices are overestimated or underestimated (yellow color in Fig. 6.36).

This parameter, K, is calculated mathematically by applying $\frac{\partial P_j}{\partial K_j} = \frac{\partial E\{(x_j - \hat{x}_j)(x_j - \hat{x}_j)^T\}}{\partial K_j} = 0$, where x_j and $\hat{x}_j$ demonstrate the actual and estimated values of a target state of the body such as rotation, translation and so on (any parameter related to navigation). To the best of our knowledge, the biological and cognitive mechanisms which lay under this estimation are almost unknown. The calculation usually has a certain amount of error. This error leads to more divergence between actual and estimated sensory data,

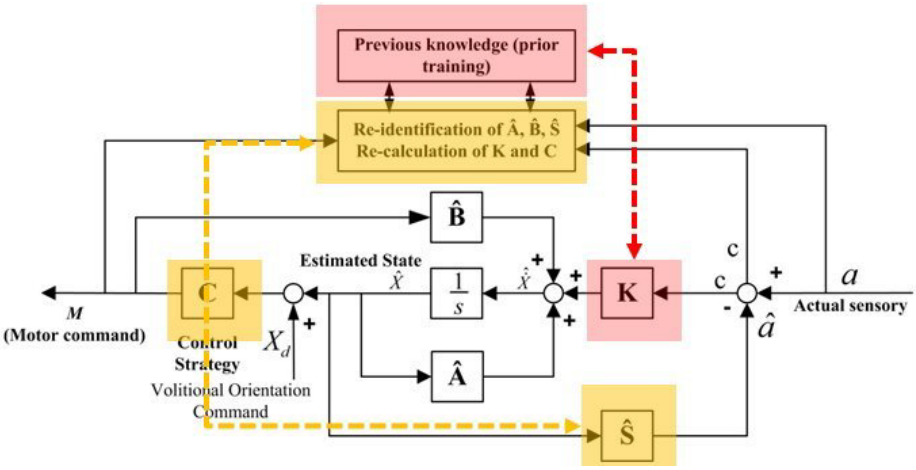


Figure 6.36: CNS model for explaining the reason of VIMS by the theory of sensory conflict [144].

however it is dissolved sometimes by adaptation or habituation in the real environment. But in a virtual environment, the conflict vector grows due to some missing non-vestibular and vestibular sensory which in turn gives rise to more sickness. When the sickness level crosses a certain level of perception tolerance, the symptoms of sickness will appear. Since the first effect of a re-calculation error appears in sensorymotor commands (control mechanism shown in Fig. 2.42 in chapter 2 and Fig. 6.36 in the current chapter), it can be easily sensed in the stance posture stability and the central gravity sway area.

That is the reason why the COG was selected as a psychophysiological measure for our study. A little difference between F/B and L/R sways makes the COG area elliptical. The body starts turning around its base in a circular shape with the presence of sickness and mal-operation of C function (control strategy as seen in Fig. 6.36). Based on subjective studies, the frequency of this marginally stable post-exposure sway appears in the bandwidth of $1.3 - 2.75$ Hz and can move to further frequencies up to 20 Hz. However, our observation shows that after 5 Hz the participants will be severely sick, this process is definitely time dependent (see Fig. 6.31). The longer the participants navigate inside a VE, the more difference appears between voluntary and involuntary components Δf (see Fig. 6.14).

6.4.3 Experiment 2: effect of the distance from the virtual barrier on VIMS

Experimental design

The experiments were designed in the same system and 3D model as section 6.4.2. As seen, the only difference between the current and the previous experiments is the number of paths which is 4 in this experiment. The participants navigate along paths 1 through 4 by a Fly-stick2 using a DBNI throughout the experiment. The navigation speed was kept constant and set to 2 m/s as it was the minimum speed

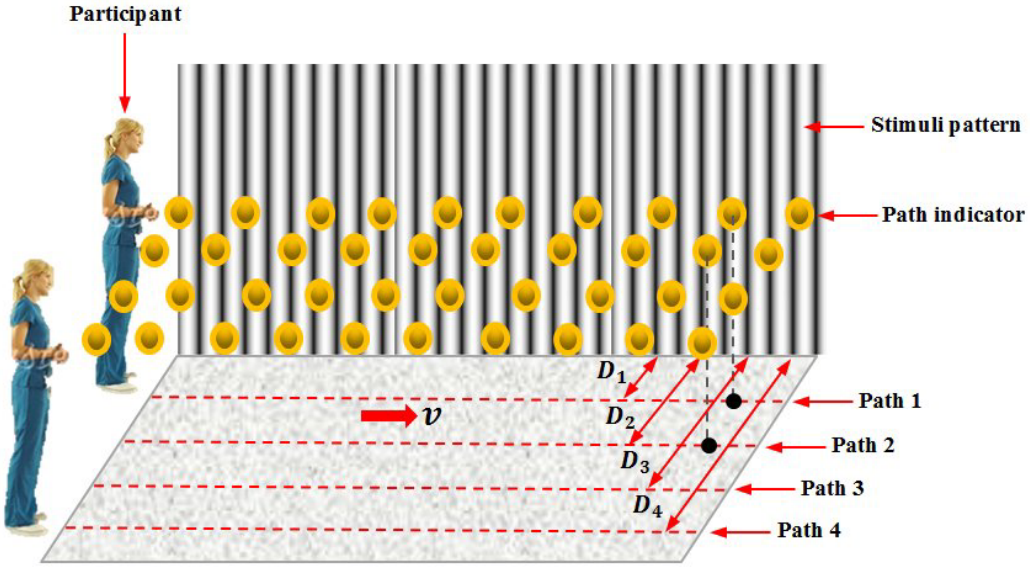


Figure 6.37: Experimental design for studying the effect of the distance on VIMS.

in the previous experiment (section 6.4.2) and induces the lowest level of sickness. This time the distance increases and the effect of this modification on VIMS will be studied. The distance between two consecutive lines is approximately 1 m. As shown in Fig. 6.37, path 1 through path 4 are adjusted 1 m ($D_1 = 1$ m), 2 m ($D_2 = 2$ m), 3 m ($D_3 = 3$ m) and 4 m ($D_4 = 4$ m) away from the virtual wall, respectively. The difference between frequency components (HF and LF) will be used as a feature for data analysis. Therefore, the pre and post-exposures COG are recorded in four points (at the end of each path) to prove the hypothesis.

Hypothesis

When the speed of the navigation is kept constant by increasing the distance between the virtual barriers and the participants, the level of sickness decreases.

To be sure about our findings and double check the hypothesis, we have set up a trial experiment with few subjects. As seen, the strategy and the test procedure is exactly the same except the number of paths in this experiment is greater than the previous one and the speed is constant. The distance between two adjacent lines as well as the first line and the wall were set to 1 m ($D_i = 1$ m). 10 males from the first experiment participated in this experiment. Kennedy's SSQ was collected from each participant as psychological measurements.

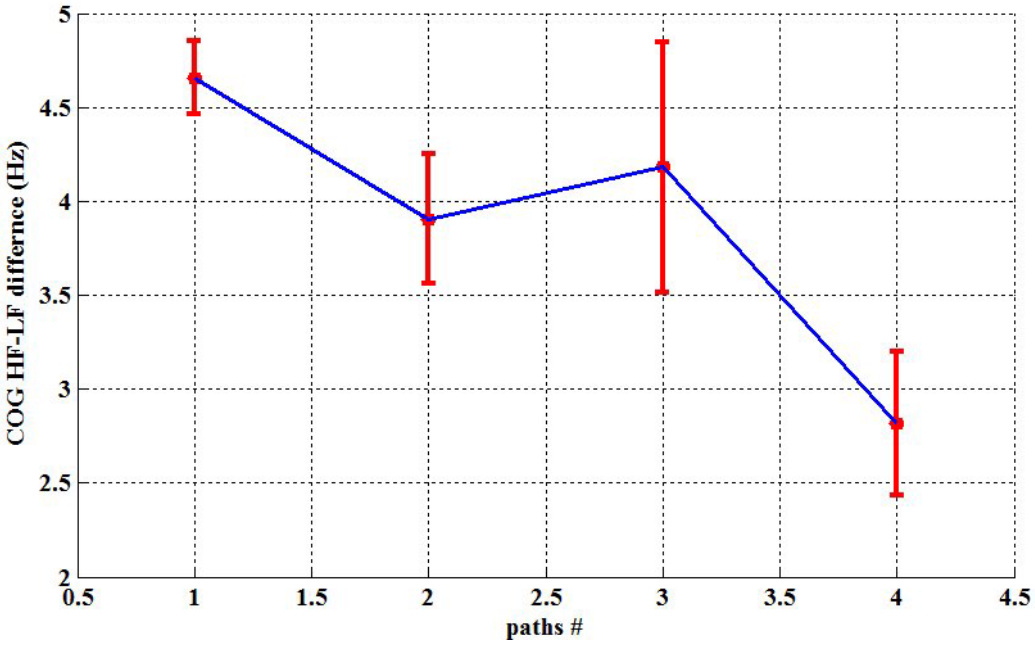


Figure 6.38: Frequency components analysis of the COG signal.

Statistical data analysis

The following steps were taken to calculate the statistical parameters in the frequency domain:

1. The FFT transform of the COG signal was calculated;
2. The mean and variance of the signal for each frequency was calculated;
3. A confidentiality interval (C.I.) was extracted via the following equation;

$$95\%\text{C.I.} = \frac{\sum_{i=1}^{n} \sqrt{\frac{(X-\mu)^2}{n-1}}}{\sqrt{n}} \times 1.96 \tag{6.13}$$

4. All the previous parameters were combined and plotted.

In general, the signal spectrum can expand from $-\infty$ Hz to $+\infty$ Hz. Practically, the frequency spectrum is limited to $0 - 20$ Hz[205] in this experiment. Since the body movement is not extremely fast due to its natural inertia, high frequency cannot extend to higher than a specific limit. If we consider very high frequency, then we might mix up noise and the signal. On the other hand, the movement can contain 0 Hz which means the body is in a standing posture without any voluntary and involuntary movements. We expect frequencies up to 20 Hz appear in the spectrum. By increasing the distance, the effect of the virtual stimuli on the participants decreases.

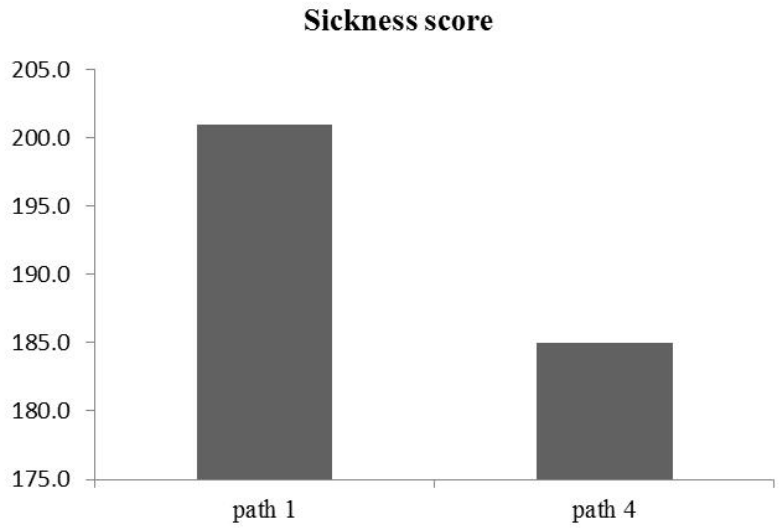


Figure 6.39: Post-exposure sickness score associated with paths 1 and 4.

The post exposure COG signal was recorded at the end of each line and the SSQ filled. The COG signals were analyzed in the frequency domain and the difference of LF and HF components were calculated. Fig. 6.38 shows the difference between each path with the next path. Statistical data analysis using a T-test shows VIMS in path 1 ($M = 4.65$, $SD = 0.61$) is significantly higher ($t(9) = 5.21$, $p < 0.05$, $p = 0.0124$) than in path 2 ($M = 3.9$, $SD = 1.1$). However, VIMS in path 3 ($M = 4.18$, $SD = 2.1$) does not significantly ($f(9) = 0.95$, $p = 0.705$) reduce with respect to path 2. On the contrary, VIMS in path 4 ($M = 2.8$, $SD = 1.2$) is significantly lower ($f(9) = 3.56$, $p < 0.01$, $p = 0.0023$) than in path 4 and path 3 ($f(9) = 3.03$, $p < 0.05$, $p = 0.0324$). SSQ data analysis shows significant ($f(9) = 3.15$, $p < 0.05$, $p = 0.042$) difference between the sickness score associated with path 1 and path 4. It means closer distance induces more vection and in turns more visually induced motion sickness, as shown in Fig. 6.39.

6.4.4 Experiment 3: comparison between DBNI and NGBNI

The effect of the navigation interface will be studied in the experiment. Two navigation interfaces, i.e., DBNI (fly-stick) and NGBNI (gesture detection with a Kinect, WIP) are selected for this step. As mentioned above, these two navigation interfaces represent two types of navigation interfaces, natural and artificial. The first type is navigation based on user gesture and the second type is navigation via devices such as a gamepad, a Fly-stick, a mouse and so on. Based on the theory that has already been discussed in chapter 2, when a gesture-based interface is employed for a navigation purpose, it seems that the motion

pattern perceived by sensory afferents is more in accordance with the proprioceptive pattern, which in turn creates less sensory conflict at the onset and the cessation of the sensory rearrangement. For that reason, it makes absolutely sense to hypothesis that navigation interfaces such as NGBNI creates less sickness comparing to DBNI.

Experiment design and participants

Fifteen subjects were selected to participate in the experiment. The experiment consists of moving along a straight path in a virtual environment. The balls were aligned in the same way as in the previous test-bench (see section 6.4.2) and the same model as in the previous experiments is used in this experiment. The experiment setup is exactly the same as that of section 6.4.2. This time the speed and the distance are kept unchanged and the navigation interface is changed. However, if the parameter such as the speed, the acceleration and the spatial virtual distance from the model are miss-adjusted, severe sickness will be induced and the experiment is then totally biased. Avoiding the intervention of other navigation parameters, the speed and the distance are set to 2.5 m/s and 4 m, respectively. Experiments 1 and 2 show that this speed and distance induce the lowest sickness. The speed is approximately close to the human natural walking speed.

Now, we will benefit from the spectrum analysis of postural sway to study the effect of different navigation interfaces on the user performance. We know that the difference between the LF and HF components of COG sway will grow up dramatically if the level of VIMS increases. Based on this finding, it is easier to hypothesis that the distance between HF and LF in NGBNI is smaller than in DBNI. Therefore, the pre and post-exposure COG signals are collected from the subjects in the same way as done in section 6.4.2.

Data analysis

First the pre and post-exposures F/B signals are transformed into the frequency space and the spectrum is calculated for each participant using the FFT. Then, at each frequency, the spectrum mean, variance and confidential interval (C.I.) are calculated by equation (6.13) as explained in section 6.4.3 across the subjects. The amount of the difference between HF and LF components is evaluated by a T-test (as seen in Figs. 6.40 and 6.41). When we change the navigation interface from DBNI to NGBNI, the HF components of the post-exposure sway (Fig. 6.40, signal in blue) get closer to 1 Hz as shown in Fig. 6.40 which means less sickness is induced by the navigation interface. Because we have better postural sway responses, it means better stability in turn. The first component seems to be more voluntary body movement due to presence and the second component refers to the degree of instability of the user posture due to presence and the interface. Interval $4.32 - 4.35$ Hz represents the HF components of the post-exposure sway for DBNI which is significantly far from the LF components ($t(14) = 4.19, p < 0.001$). Interval $1.26 - 1.35$

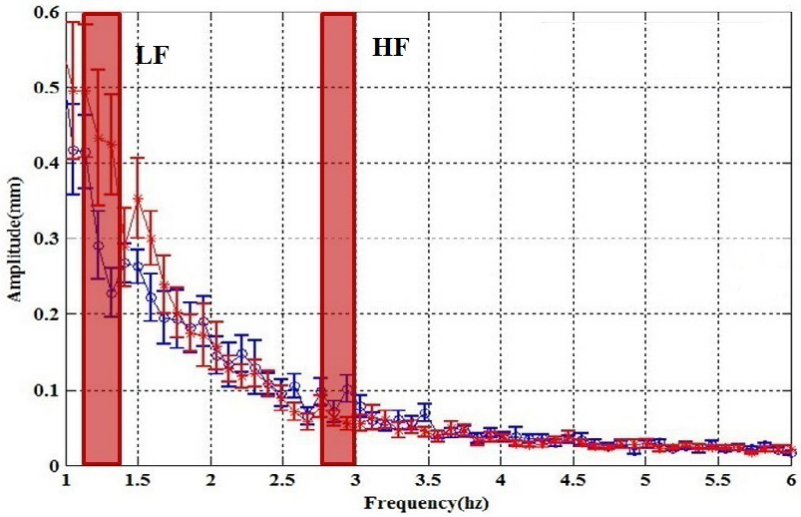


Figure 6.40: Spectrum analysis for NGBNI.

Hz contains the LF components of the post-exposure sway. At the same time, the HF components of post-exposure is quite far away from the LF components of pre-exposure too ($t(14) = 2.93, p < 0.05$) which creates a difference nearly 3.1 Hz, as shown in Fig. 6.41. On the other hand, an HF component of the post-exposure sway for NGBNI is located in interval $2.94 - 2.97$ Hz. This HF component is significantly further away from the LF components, $1.1 - 1.4$ Hz, ($t(14) = 2.33, p < 0.05$) which creates a difference of 1.9 Hz as shown in Fig. 6.40. Comparing the difference of the HF-LF components of NGBNI and DBNI shows a difference of 1.2 Hz which means the Fly-stick (DBNI) induced more sickness comparing to walking in place (NGBNI). As a result, we can conclude that NGBNI provides a better navigation interface for the users comparing to DBNI.

6.5 Conclusion

It has been shown that VIMS affects the postural body sway which in the frequency domain, means the body sway switches from a stable state, with only voluntary movements, to a marginally stable state, with voluntary and involuntary movements. Simulation results show that the second frequency component (HF) of the body sway moves to the far end of the spectrum. In fact, the difference between the low frequency (LF) and high frequency components grows. Therefore, we can say the difference between LF and HF components is an indicator of VIMS susceptibility. This is equivalent to the dilation of the COG area and changing from an elliptical shape to a circular shape in the time domain. Besides, binocular disparity can be used as a simple feature to calculate the level of VIMS and its onset.

VIMS increases linearly by translational velocity as shown in Fig. 6.42.b while the effect of the rotational

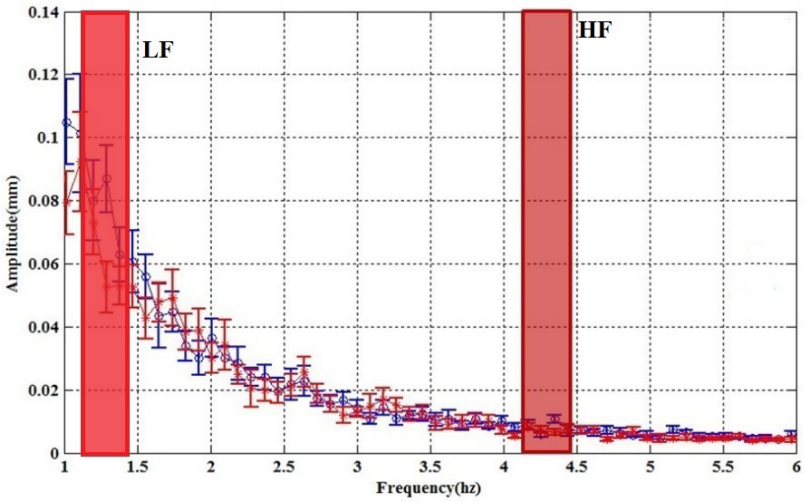


Figure 6.41: Spectrum analysis for DBNI.

Parameters	*Range*		
ω *(°/s)*	$15.0 - 55.0$	$15.0 - 55.0$	$15.0 - 55.0$
v *(m/s)*	$0.5 - 2.5$	$2.5 - 6.5$	> 6.5
d *(m)*	1.0	$1.0 - 4.0$	> 4.0

Table 6.6: Possible settings for navigation parameters.

velocity on VIMS has a Gaussian distribution, with a mean located in $\mu = 60°/s$ as shown in Fig. 6.42.a. Moreover, VIMS decreases linearly by increasing the distance from the virtual barrier as seen in Fig. 6.42.c. In addition, the variation of VIMS during rotation changes independent of the rotation radius, while VIMS during translation depends on the distance and the velocity. For instance, when the velocity is low (2.5 m/s) and the distance is far (4 m) the level of VIMS is low (150), while when the distance is close to the virtual barrier (1 m) and the velocity is low (2.5 m/s) VIMS is high (400). Thus, when the user is navigating inside a narrow corridor, he needs to decrease the speed and shift his trajectory to the middle of the corridor.

To make the long story short, the navigation parameters are suggested to be selected from the range specified in Table 6.6. Although setting the values proposed in table 6.6 as navigation parameters decreases the VIMS level, to minimize the VIMS level, an optimization problem needs to be solved in real-time to correct the traveling trajectory. The update rate of the target 3D display should be above 60 fps to be able to implement this speed control system.

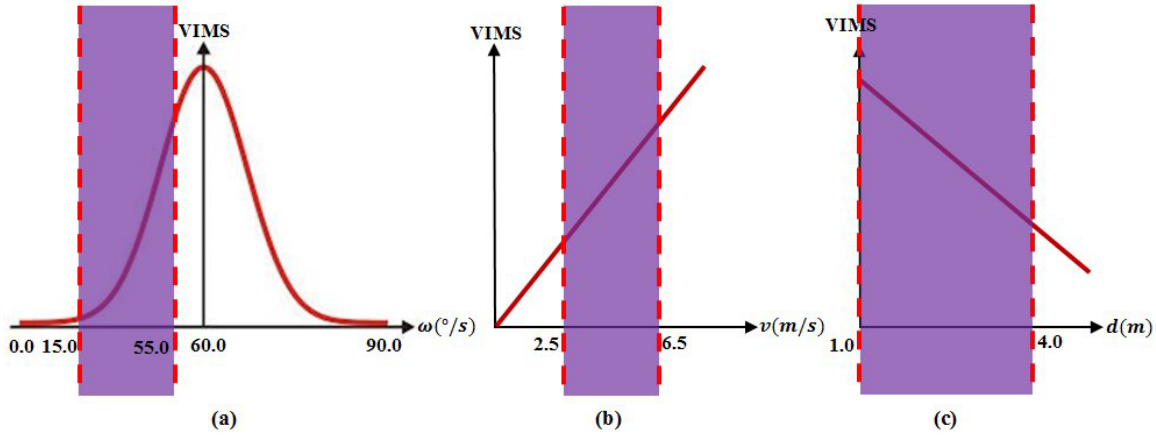


Figure 6.42: (a) VIMS-rotational velocity, (b) VIMS-translational velocity, (c) VIMS-distance from the virtual barrier.

7 Conclusion

7.1 Conclusion

The oculo-vestibular dynamics was modeled and discussed using Bayesian and Kalman theories, by incorporating the CNS model (artificial neural network). It has been shown that a sensory conflict (MS and VIMS) can appear at the onset or cessation of any sensory rearrangement in the vestibular nuclei (VN). The upright postural sway modeling revealed that an inappropriate visual flow can dilate the COG area, modify the COG shape and increase the difference between the frequency components of the post-exposure upright stance. More precisely, the participant's SSQ score increases by growing the difference between LF and HF components and dilating the COG area simultaneously. For that reason, the COG area/shape as well as the frequency components can be used as an alternative solution for VIMS prediction and estimation in a real-time process. Since VIMS is mostly provoked by the visual flow, the mathematical model of binocular vision was derived to study the effect of rotational and translational movements on VIMS susceptibility. It has been shown that vergence is deteriorated during rotational movement by increasing the rotational velocity of a 3D scene. As a result, the difference between the two eyes (residual signal) increases during the transition from one point to another. Such a divergence between the two eyes gives rise to a higher SSQ score. So, it can be effectively used as an indicator of a VIMS occurrence.

Another reason of the sensory conflict is the divergence between vestibular and non-vestibular measurements due to the lack of sensory information. It means the pattern perceived by the user via sensory inputs is not the same as the stored pattern (a previous knowledge is made by the latest translation in the real environment). Avoiding this conflict, a navigation/manipulation interface either needs to be designed in a very similar manner that we interact in the real environment (gesture-based interfaces), or maximum sensory inputs (texture, pressure, gravity, weight and so on) should be incorporated in the interface. Six interaction interfaces were developed to study the effect of the navigation and manipulation interface on VIMS. The navigation/manipulation interfaces are:

1. Artificial interfaces
    (a) Device-based interface
    (b) Navigation using iDevice such as an iPhone, an iPad
2. Natural interfaces
    (a) Walking in place based on gesture analysis and body motion classification
    (b) Walking in place by head movement analysis
    (c) Navigation with sound command (speech processing)
    (d) Bracelet and sensor fusion

This study, as explained on chapter 6, shows that artificial interfaces induce more sickness compared to natural interfaces. However, the display technology, the interaction (navigation/manipulation) interfaces and the parameters (velocity/acceleration), the traveling trajectory, the system parameters (frame rate, delay, 3D rendering, ...) highly influence the user performance and contribute largely in VIMS and other cyber effects.

Manipulation of an object in a virtual environment is very different than the one in real environment. This is because a lot of sensory inputs is missing in the virtual environment. As a result, the brain cannot generate exactly the same action as produced in the real environment with the full sensory inputs.

7.2 Future work

Due to the fact that the EEG sensor was not accessible in the current research, the VOG signal was recorded and analyzed to see what is going on when VIMS occurs, as shown in chapters 5 and 6. But a better solution for measuring VIMS is to provoke the ocular system while measuring an EEG signal in the V3a and V5 areas of the visual cortex as they are responsible for motion, as explained in chapter 2. Since these two areas are

responsible for motion estimation, therefore the characteristics of VIMS can be better understood in this way. The brain map can be used to study the effect of the visual flow on VIMS susceptibility.

This thesis studied binocular vision very deeply using the classic control theory and artificial intelligence. Both of these two technologies are widely used in robotics, machine vision and some other similar disciplines. We have not gone deep inside the simulation of binocular vision. However, the model can be separately studied and the result can be used in robot intelligence and man-robot interactions.

A

Q1: Examinee general information (pre-exposure)

A.1 English

1. First name:
2. Family name:
3. Email address: @gmail.com or
4. Age: 15-20 years
5. Weight: 0-50
6. Gender: man ☐ woman ☐
7. Profession: student ☐ engineer ☐ programmer ☐ office clerk ☐ other professions ☐ :
8. Experience with a computer:

(a) professional gamers ☐
Level: every day: ☐ at least once a week: ☐ at least once a month ☐ played before a lot but not now ☐ played but I don't know exactly how much ☐

(b) computer users ☐
Level: daily users (office clerk, ...) ☐ use the computer not so much ☐

9. From the gadgets/devices listed below which one you have used so far:
Kinect ☐ gamepad ☐ Fly-stick ☐

10. Have you played to a laser game so far: yes ☐ no ☐

11. Do you get sick when you travel: yes ☐ no ☐

12. If the answer to item 11 is yes which of the following travel vehicles makes you sicker:
car ☐ train ☐ airplane ☐ boat ☐ ship/ferry ☐

13. When the traveling vehicle is in motion do you study: yes ☐ no ☐

14. If the answer to item 13 is yes do you feel (during study): headache ☐ eyestrain ☐ fatigue ☐

15. If you are a woman have you been pregnant: yes ☐ no ☐

16. Do you have any auditorial disorder: yes ☐ no ☐

17. If the answer to item 16 is yes what is the level of disorder:
slight ☐ moderate ☐ headphone implanted ☐ severe ☐ near to deaf ☐

18. Do you have any visual disorder: slight ☐ moderate ☐ sever ☐ other ☐ :

19. Do you wear glasses: yes ☐ no ☐

A.2 French

1. Prénom :

2. Nom de famille :

3. Adresse Email : @gmail.com ou

4. Age : 25-30 ans

5. Poids : 60-70

6. Sexe : homme ☐ femme ☐

7. Profession :
étudiant ☐ ingénieur ☐ employé de bureau ☐ programmeur ☐ autres professions ☐ :

8. Expérience avec un ordinateur :

   (a) joueur professionnel ☐
   Niveau : tous les jours ☐ au moins une fois par semaine ☐ au moins une fois par mois ☐ joué avant beaucoup mais pas maintenant ☐ joué mais je ne sais pas exactement combien de temps ☐

   (b) utilisateur d'ordinateur ☐
   Niveau : utilisateur quotidien (employé de bureau, ...) ☐ ordinateur pas souvent utilisé ☐

9. Parmi les gadgets / appareils énumérés ci-dessous lequel avez-vous utilisé jusqu'à présent :
Kinect ☐ gamepad ☐ Fly-stick ☐

10. Avez-vous déjà joué au laser game : oui ☐ non ☐

11. Tombez-vous malade lorsque vous voyagez : oui ☐ non ☐

12. Si la réponse à la question 11 est oui, quel mode de transport vous rend davantage malade :
voiture ☐ train ☐ avion ☐ bateau ☐ navire / ferry ☐

13. Lorsque le véhicule est en mouvement avez-vous l'habitude d'étudier : oui ☐ non ☐

14. Si la réponse à la question 13 est oui, sentez-vous (pendant que vous étudiez) :
maux de tête ☐ fatigue oculaire ☐ fatigue ☐

15. Si vous êtes une femme avez-vous été enceinte : oui ☐ non ☐

16. Avez-vous des troubles d'audition : oui ☐ non ☐

17. Si la réponse à la question 16 est oui, quel est le niveau de trouble :
léger ☐ modéré ☐ appareil implanté ☐ sévère ☐ proche de la surdité ☐

18. Avez-vous des troubles visuels : léger ☐ modéré ☐ autre ☐:

19. Portez-vous des lunettes : oui ☐ non ☐

B

Kennedy's Simulator Sickness Questionnaire (SSQ)

B.1 Questionnaire and report example

The Simulator Sickness Questionnaire (SSQ) introduced by Kennedy et al.[98] was used as a measure in visually induced motion sickness experiments (see tables B.1 and B.2). The symptoms used and their weightings are given in table B.3 (adapted from[98]). The SSQ is based on three components: nausea, oculomotor problems, and disorientation. These can be combined to produce a total SSQ score, as described in table B.3.

B.2 Calculation instruction

The participants report the degree to which they experience each of the above symptoms as one of "None", "Slight", "Moderate" and "Severe". These are scored respectively as 0, 1, 2 and 3. To compute the scale scores for each column, the reported value for each symptom is multiplied by the weight in each column and then summed down the columns. The total SSQ score is obtained by adding the scale scores across

	Level			
Human factors	*Non*	*Slight*	*Moderate*	*Severe*
	0	$X0$	$X1$	$X2$
General discomfort	☐	☐	☐	☐
Fatigue	☐	☐	☐	☐
Headache	☐	☐	☐	☐
Eyestrain	☐	☐	☐	☐
Difficulty focusing	☐	☐	☐	☐
Salivation increasing	☐	☐	☐	☐
Sweating	☐	☐	☐	☐
Nausea	☐	☐	☐	☐
Difficulty concentrating	☐	☐	☐	☐
Fullness of the head	☐	☐	☐	☐
Blurred vision	☐	☐	☐	☐
Dizziness with eyes open	☐	☐	☐	☐
Dizziness with eyes closed	☐	☐	☐	☐
Vertigo	☐	☐	☐	☐
Stomach awareness	☐	☐	☐	☐
Burping	☐	☐	☐	☐

Table B.1: Questionnaire.

	Level				
Human factors	*Non*	*Slight*	*Moderate*	*Severe*	*Val.*
	0	$X0 = 1$	$X1 = 2$	$X2 = 3$	
General discomfort	☐	☒	☐	☐	1
Fatigue	☒	☐	☐	☐	0
Headache	☐	☒	☐	☐	1
Eyestrain	☐	☐	☒	☐	2
Difficulty focusing	☐	☒	☐	☐	1
Salivation increasing	☒	☐	☐	☐	0
Sweating	☒	☐	☐	☐	0
Nausea	☐	☒	☐	☐	1
Difficulty concentrating	☐	☐	☒	☐	2
Fullness of the head	☐	☒	☐	☐	1
Blurred vision	☐	☒	☐	☐	1
Dizziness with eyes open	☒	☐	☐	☐	0
Dizziness with eyes closed	☒	☐	☐	☐	0
Vertigo	☒	☐	☐	☐	0
Stomach awareness	☒	☐	☐	☐	0
Burping	☒	☐	☐	☐	0

Table B.2: Example of filled questionnaire.

	Weight		
SSQ Symptom	*Nausea*	*Oculomotor*	*Disorientation*
General discomfort	1	1	0
Fatigue	0	1	0
Headache	0	1	0
Eyestrain	0	1	0
Difficulty focusing	0	1	1
Salivation increasing	1	0	0
Sweating	1	0	0
Nausea	1	0	1
Difficulty concentrating	1	1	0
Fullness of the head	0	0	1
Blurred vision	0	1	1
Dizziness with eyes open	0	0	1
Dizziness with eyes closed	0	0	1
Vertigo	0	0	1
Stomach awareness	1	0	0
Burping	1	0	0

Table B.3: Computation of the SSQ Scores.

the three columns and multiplying by 3.74. The weighted scale scores for each column individually can be found by multiplying the "Nausea" scale score by 9.54, the "Oculomotor" subscale by 7.58 and the "Disorientation" subscale by 13.92[98].

C

List of publications

C.1 Journal papers

1. M. Pedrotti, M.-A. Mirzaei, A. Tedesco, J.-R. Chardonnet, F. Merienne, S. Benedetto, and T. Baccino, "Automatic Stress Classification With Pupil Diameter Analysis," *International Journal of Human-Computer Interaction*, vol. 30, no. 3, pp. 220–236, 2014.

2. M.-A. Mirzaei, F. Merienne, J. H. Oliver, and J.-R. Chardonnet, "New wireless connection between user and VE using speech processing," *Virtual Reality*, vol. 18, no. 4, pp. 235–243, 2014.

C.2 Conference papers

C.2.1 International peer-reviewed conferences

1. M. Pedrotti, T. Baccino, and M.-A. Mirzaei, "Driver stress-state classification using pupil diameter signal analysis," presented at the 28th International Congress of Applied Psychology, Paris, France, 2014, pp. 37–39.

2. M.-A. Mirzaei, J.-R. Chardonnet, F. Merienne, and A. Genty, "Navigation and interaction in a real-scale digital mock-up using natural language and user gesture," presented at the Laval Virtual VRIC'14, Laval, France, 2014, pp. 1–4.

3. M.-A. Mirzaei, S. Prianto, J.-R. Chardonnet, C. Père, and F. Merienne, "New motherwavelet for pattern detection in IR image," presented at the IEEE Video Communication and Image Processing, Kuching, Malaysia, 2013, pp. 1–6.

4. M.-A. Mirzaei, J.-R. Chardonnet, C. Père, and F. Merienne, "Improvement of the real-time gesture analysis by a new mother wavelet and the application for the navigation inside a scale-one 3D system," presented at the IEEE International Conference on Advanced Video and Signal-Based Surveillance, Krakow, Poland, 2013, pp. 270–275.

5. M.-A. Mirzaei, J.-R. Chardonnet, C. Père, and F. Merienne, "Sensor fusion for interactive real-scale modeling and simulation systems," presented at the 18th International Conference on Computer Games (CGAMES USA), Louisville, USA, 2013, pp. 149–153.

6. O. E. Ramos Ponce, M.-A. Mirzaei, and F. Merienne, "Tracking in Presence of Total Occlusion and Size Variation using Mean Shift and Kalman Filter," presented at the IEEE/SICE International Symposium on System Integration, Kyoto, Japan, 2011.

C.2.2 Domestic conferences

1. G. Hattar, M.-A. Mirzaei, S. Garbaya, F. Merienne, and J.-R. Chardonnet, "Joint Platform for the Real-Scale 3D Visualization of Conceptual Design Incorporating Semantic," presented at the AFRV, Laval, France, 2013, pp. 29–34.

2. M.-A. Mirzaei, J.-R. Chardonnet, C. Père, and F. Merienne, "Designing a 3D Navigation System Using Cognitive Factors," presented at the AFRV, Strasbourg, France, 2012.

3. M.-A. Mirzaei, J.-R. Chardonnet, C. Père, and F. Merienne, "Test-bench for evaluating navigation and interaction in large virtual databases," presented at the Confere, Venice, Italy, 2012.

References

[1] 3CPM (2008). 3cpm electrogastrogram (http://www.3cpmcompany.com/Products.htm).

[2] Andre, D., Pelletier, R., Farringdon, J., Safi er, S., Talbott, W., Stone, R., Vyas, N., Trimble, J., Wolf, D., Vishnubhatla, S., Boehmke, S., Stivoric, J., & Astro, T. (2006). *The Development of the SenseWear armband, A Revolutionary Energy Assessment Device to Assess Physical Activity and Lifestyle*. Technical report, BodyMedia, Inc.

[3] Antonini, M., Barlaud, M., Mathieu, P., & Daubechies, I. (1992). Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2), 205–220.

[4] Barnes, G. & Asselman, P. (1991). The mechanism of prediction in human smooth pursuit eye movements. *The Journal of Physiology*, 439(1), 439–461.

[5] Becker, W. (1989). The neurobiology of saccadic eye movements. metrics. *Reviews of oculomotor research*, 3, 13–67.

[6] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions PAMI, special issue Learning Deep Architectures*, 35(8), 1798–1828.

[7] Bennett, J., Maguire, A. M., Cideciyan, A. V., Schnell, M., Glover, E., Anand, V., Aleman, T. S., Chirmule, N., Gupta, A. R., Huang, Y., Gao, G.-P., Nyberg, W. C., Tazelaar, J., Hugues, J., Wilson, J. R., & G, J. S. (1999). Stable transgene expression in rod photoreceptors after recombinant adeno-associated virus-mediated gene transfer to monkey retina. *Proceedings of the National Academy of Sciences*, 96(17), 9920–9925.

[8] Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., & Cruz-Neira, C. (2001). Vr juggler: A virtual platform for virtual reality application development. In *IEEE Virtual Reality* (pp. 89–96).: IEEE.

[9] Biopack, I. Bionomadix wirless transmiter (http://www.biopac.com/BioNomadix-2Ch-Wireless-EMG).

[10] Biopack, I. Biopack electromyography systems (http://www.biopac.com/emg-electromyography).

[11] Bolt, R. (1980). "put-that-there": Voice and gesture at the graphics interface. In *ACM Siggraph Computer Graphics* (pp. 262–270).

[12] Borah, J., Young, L. R., & Curry, R. E. (1979). *Sensory Mechanism Modeling*. Technical report, DTIC Document.

[13] Borah, J., Young, L. R., & Curry, R. E. (1988). Optimal estimator model for human spatial orientationa. *Annals of the New York Academy of Sciences*, 545(1), 51–73.

[14] Boritz, J. & Booth, K. (1997). A study of interactive 3d point location in a computer simulated virtual environment. In *ACM symposium on Virtual reality software and technology* (pp. 181–187).

[15] Born, R. T. & Bradley, D. C. (2005). Structure and function of visual area mt. *Annu. Rev. Neurosci.*, 28, 157–189.

[16] Bos, J. E. (2003). Method for the prevention of motion sickness, and apparatus for detecting and signaling potentially sickening motions.

[17] Bos, J. E., Bles, W., & Groen, E. L. (2008). A theory on visually induced motion sickness. *Displays*, 29(2), 47–57.

[18] Bowman, D. & Hodges, L. (1997). An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Symposium on Interactive 3D graphics* (pp. 35–ff).

[19] Brandt, T., Dichgans, J., & Koenig, E. (1973). Differential effects of central versus peripheral vision on egocentric and exocentric motion perception. *Experimental Brain Research*, 16(5), 476–491.

[20] Cannon, S. & Robinson, D. (1987). Loss of the neural integrator of the oculomotor system from brain stem lesions in monkey. *Journal of Neurophysiology*, 57(5), 1383–1409.

[21] Carter, J. (2012). Improving the performance of gis/spatial analysts though novel applications of the emotiv epoc eeg headset. Master's thesis, Michigan Technological University.

[22] Chandrasekaran, M., Muralidhar, M., Krishna, C. M., & Dixit, U. (2010). Application of soft computing techniques in machining performance prediction and optimization: a literature review. *The International Journal of Advanced Manufacturing Technology*, 46(5-8), 445–464.

[23] Chauvin, Y. & Rumelhart, D. E. (1995). *Backpropagation: theory, architectures, and applications*. Psychology Press.

[24] Chen, J. & Bowman, D. A. (2006). Effectiveness of cloning techniques for architectural virtual environments. In *Virtual Reality Conference* (pp. 103–110).

[25] Chen, W., Chen, J., & So, R. (2011). Visually induced motion sickness: Effects of translational visual motion along different axes. In *Contemporary Ergonomics and Human Factors 2011: Proceedings of the international conference on Ergonomics & Human Factors 2011* (pp. 281–287).

[26] Chen, Y.-C., Duann, J.-R., Chuang, S.-W., Lin, C.-L., Ko, L.-W., Jung, T.-P., & Lin, C.-T. (2010). Spatial and temporal eeg dynamics of motion sickness. *NeuroImage*, 49(3), 2862–2870.

[27] Chen-Harris, H., Joiner, W. M., Ethier, V., Zee, D. S., & Shadmehr, R. (2008). Adaptive control of saccades via internal feedback. *The Journal of Neuroscience*, 28(11), 2804–2813.

[28] Cheung, B., Howard, I., & Money, K. (1991). Visually-induced sickness in normal and bilaterally labyrinthine-defective subjects. *Aviation, Space, and Environmental Medicine*, 1, 1–10.

[29] Cheung, B., Howard, I., Nedzelski, J., & Landolt, J. (1989). Circularvection about earth-horizontal axes in bilateral labyrinthine-defective subjects. *Acta oto-laryngologica*, 108(5-6), 336–344.

[30] Chittaro, L., Gatla, V. K., & Venkataraman, S. (2005). The interactive 3d breakaway map: A navigation and examination aid for multi-floor 3d worlds. In *International Conference on Cyberworlds* (pp. 59–66).

[31] Chow, E., Ji, J., So, R., & Cheung, R. (2007). The effect of visual stimulus oscillation frequency on postural disturbance in roll and in fore-and-aft direction. In *First International Symposium on Visually Induced Motion Sickness, Fatigue, and Photosensitive Epileptic Seizures (VIMS2007)* (pp. 10–12).

[32] Ciresan, D. C., Meier, U., Gambardella, L. M., & Schmidhuber, J. (2010). Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12), 3207–3220.

[33] Couclelis, H., Golledge, R. G., Gale, N., & Tobler, W. (1987). Exploring the anchor-point hypothesis of spatial cognition. *Journal of Environmental Psychology*, 7(2), 99–122.

[34] Cullen, K. E. (2004). Sensory signals during active versus passive movement. *Current opinion in neurobiology*, 14(6), 698–706.

[35] Dangelmaier, W., Fischer, M., Gausemeier, J., Grafe, M., Matysczok, C., & Mueck, B. (2005). Virtual and augmented reality support for discrete manufacturing system simulation. *Computers in Industry*, 56(4), 371–383.

[36] Darken, R. P., Cockayne, W. R., & Carmein, D. (1997). The omni-directional treadmill: A locomotion device for virtual worlds. In *10th annual ACM Symposium on User Interface Software and Technology* (pp. 213–221).

[37] Darken, R. P. & Sibert, J. L. (1993). A toolset for navigation in virtual environments. In *6th annual ACM symposium on User Interface Software and Technology* (pp. 157–165).

[38] DiZio, P. & Lackner, J. (1998). *Alleviation of motion sickness and postural instability during and after virtual environment exposure*. Technical report, Naval Air Warfare Center Training Systems Division.

[39] Ebenholtz, S. M., Cohen, M. M., & Linder, B. J. (1994). The possible role of nystagmus in motion sickness: a hypothesis. *Aviation, Space, and Environmental Medicine*, 65(11), 1032–1035.

[40] Edlow, J., Newman-Toker, D., & Savitz, S. (2008). Diagnosis and initial management of cerebellar infarction. *Lancet Neurol*, 7, 951–964.

[41] Enderle, J. D. & Wolfe, J. W. (1987). Time-optimal control of saccadic eye movements. *IEEE Transactions on Biomedical Engineering*, 1(1), 43–55.

[42] Erickson, T. (1990). Working with interface metaphors. *The art of human-computer interface design*, 1, 65–73.

[43] Everson, R. & Roberts, S. J. (2000). Independent components analysis. In *Artificial Neural Networks in Biomedicine* (pp. 153–168). Springer.

[44] Fernandez, C. & Goldberg, J. M. (1971). Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. ii. response to sinusoidal stimulation and dynamics of peripheral vestibular system. *J Neurophysiol*, 34(4), 661–675.

[45] Forsberg, A., Herndon, K., & Zeleznik, R. (1996). Aperture based selection for immersive virtual environments. In *9th annual ACM symposium on User Interface Software and Technology* (pp. 95–96).

[46] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4), 193–202.

[Garmin] Garmin. Foretrex 401 (https://buy.garmin.com/en-US/US/on-the-trail/wrist-worn/foretrex-401/prod30026.html).

[48] Gatev, P., Thomas, S., Kepple, T., & Hallett, M. (1999). Feedforward ankle strategy of balance during quiet stance in adults. *The Journal of Physiology*, 514(3), 915–928.

[49] Gillner, S. & Mallot, H. A. (1998). Navigation and acquisition of spatial knowledge in a virtual maze. *Journal of Cognitive Neuroscience*, 10(4), 445–463.

[50] Giordano, P. R., Souman, J., Mattone, R., De Luca, A., Ernst, M., & Bulthoff, H. (2008). The cyberwalk platform: Humna-machine interaction enabling unconstrained walking through vr. In *First Workshop for Young Researchers on Human-Friendly Robotics*.

[51] Goldberg, J. M. & Fernandez, C. (1971). Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey. i. resting discharge and response to constant angular accelerations. *J Neurophysiol*, 34(4), 635–660.

[52] Guo, C., Ji, J., & So, R. (2011). Could okan be an objective indicator of the susceptibility to visually induced motion sickness? In *IEEE Virtual Reality Conference (VR)* (pp. 87–90).

[53] Harris, C. M. (1998). On the optimal control of behaviour: a stochastic perspective. *Journal of neuroscience methods*, 83(1), 73–88.

[54] Harris, C. M. & Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature*, 394(6695), 780–784.

[55] Harris, C. M. & Wolpert, D. M. (2006). The main sequence of saccades optimizes speed-accuracy trade-off. *Biological cybernetics*, 95(1), 21–29.

[56] Hart, S. G. & Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52, 139–183.

[57] Harwood, M. R., Mezey, L. E., & Harris, C. M. (1999). The spectral main sequence of human saccades. *The Journal of Neuroscience*, 19(20), 9098–9106.

[58] Haslwanter, T., Jaeger, R., Mayr, S., & Fetter, M. (2000). Three-dimensional eye-movement responses to off-vertical axis rotations in humans. *Experimental Brain Research*, 134(1), 96–106.

[59] Hastad, J. (1994). On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3), 484–492.

[60] Hettinger, L. J. & Riccio, G. E. (1992). Visually induced motion sickness in virtual environments. *Presence: Teleoperators and Virtual Environments*, 1(3), 306–310.

[61] Hill, A. S. (2008). Ygdrasil ide (`http://www.alexshill.com/projects/ygdrasil.htm`).

[62] Himi, N., Koga, T., Nakamura, E., Kobashi, M., Yamane, M., & Tsujioka, K. (2004). Differences in autonomic responses between subjects with and without nausea while watching an irregularly oscillating video. *Autonomic Neuroscience*, 116(1), 46–53.

[63] Hinckley, K., Pausch, R., Goble, J., & Kassell, N. (1994). A survey of design issues in spatial input. In *7th annual ACM symposium on User Interface Software and Technology* (pp. 213–222).

[64] Hinckley, K., Tullio, J., Pausch, R., Proffitt, D., & Kassell, N. (1997). Usability analysis of 3d rotation techniques. In *10th annual ACM symposium on User Interface Software and Technology* (pp. 1–10).

[65] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6), 82–97.

[66] Hinton, G. E. (2007). Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10), 428–434.

[67] Hintzman, D. L., O'Dell, C. S., & Arndt, D. R. (1981). Orientation in cognitive maps. *Cognitive Psychology*, 13(2), 149–206.

[68] Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen netzen. Master's thesis, Institut für Informatik, Technische Universität, München.

[69] Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

[70] Holmes, S. R. & Griffin, M. J. (2001). Correlation between heart rate and the severity of motion sickness caused by optokinetic stimulation. *Journal of Psychophysiology*, 15(1), 35–42.

[71] Hosman, R. & Stassen, H. (1999). Pilot's perception in the control of aircraft motions. *Control engineering practice*, 7(11), 1421–1428.

[72] Howarth, P. & Finch, M. (1999). The nauseogenicity of two methods of navigating within a virtual environment. *Applied Ergonomics*, 30(1), 39–45.

[73] Hu, S., Stern, R. M., Vasey, M. W., & Koch, K. L. (1989). Motion sickness and gastric myoelectric activity as a function of speed of rotation of a circular vection drum. *Aviation, Space, and Environmental Medicine*, 1, 1–7.

[74] Huang, G.-B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, 2(2), 107–122.

[Igou] Igou, S. W. The anatomy of the brain (http://www.braininjury.com/anatomy.shtml).

[76] Irwin, J. (1881). The pathology of sea-sickness. *The Lancet*, 118(3039), 907–909.

[77] Ito, M. & Kano, M. (1982). Long-lasting depression of parallel fiber-purkinje cell transmission induced by conjunctive stimulation of parallel fibers and climbing fibers in the cerebellar cortex. *Neuroscience letters*, 33(3), 253–258.

[78] Iwata, H. (1999). Locomotion interface for virtual environments. In *9th International Symposium of Robotics Research (ISRR'99)* (pp. 220–226).

[79] Iwata, H., Yano, H., & Tomiyoshi, M. (2007). String walker. In *SIGGRAPH*.

[80] Jacoby, R., Ferneau, M., & Humphries, J. (1994). Gestural interaction in a virtual environment. In *Proceedings of SPIE*, volume 2177 (pp. 355).

[81] James, W. (1881). Sense of dizziness in deaf-mutes. *Mind*, 23, 412–413.

[82] Jansen-Osmann, P. (2002). Using desktop virtual environments to investigate the role of landmarks. *Computers in human behavior*, 18(4), 427–436.

[83] Jansen-Osmann, P. & Wiedenbauer, G. (2004). The representation of landmarks and routes in children and adults: A study in a virtual environment. *Journal of Environmental Psychology*, 24(3), 347–357.

[84] Ji, J., So, R., Lor, F., Cheung, T., Howrth, P., & Stanney, K. (2005). A search for possible neural pathways leading to visually induced motion sickness. *Vision*, 17(2), 131–134.

[85] Ji, J. T., So, R. H., & Cheung, R. T. (2009). Isolating the effects of vection and optokinetic nystagmus on optokinetic rotation-induced motion sickness. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 51(5), 739–751.

[86] Johnson, W. H., Sunahara, F. A., & Landolt, J. P. (1999). Importance of the vestibular system in visually induced nausea and self-vection. *Journal of Vestibular Research*, 9(2), 83–87.

[87] Juhola, M. (1988). Detection of nystagmus eye movements using a recursive digital filter. *IEEE Transactions on Biomedical Engineering*, 35(5), 389–395.

[88] Jul, S. & Furnas, G. W. (1997). Navigation in electronic worlds: a chi 97 workshop. *Sigchi Bulletin*, 29, 44–49.

[89] Kaga, K. (1992). Memaino kouzo: Structure of vertigo. *Kanehara, Tokyo*, 1, 23–26.

[90] Kakeya, H. (2007). Moevision: simple multiview display with clear floating image. In *Electronic Imaging 2007* (pp. 64900J–64900J).

[91] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.

[92] Kardamakis, A. A. & Moschovakis, A. K. (2009). Optimal control of gaze shifts. *The Journal of Neuroscience*, 29(24), 7723–7730.

[93] Kellogg, R. S., Kennedy, R. S., & Graybiel, A. (1964). *Motion sickness symptomatology of labyrinthine defective and normal subjects during zero gravity maneuvers*. Technical report, DTIC Document.

[94] Kennedy, R. & Stanney, K. (1988). Gaps in our knowledge about motion sickness and cybersickness. In *Motion Sickness, Simulator Sickness, Balance Disorders and Sopite Syndrome Conference* (pp. 9–11).

[95] Kennedy, R., Stanney, K., Drexler, J., Compton, D., & Jones, M. (1999). Computerized methods to evaluate virtual environment aftereffects. In *Driving simulation conference* (pp. 273–287).

[96] Kennedy, R. S., Fowlkes, J. E., & Lilienthal, M. G. (1993a). Postural and performance changes following exposures to flight simulators. *Aviation, Space, and Environmental Medicine*, vol.1, 912–920.

[97] Kennedy, R. S., Graybiel, A., McDonough, R. C., & Beckwith, D. (1968). Symptomatology under storm conditions in the north atlantic in control subjects and in persons with bilateral labyrinthine defects. *Acta oto-laryngologica*, 66(1-6), 533–540.

[98] Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993b). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international Journal of Aviation Psychology*, 3(3), 203–220.

[99] Khalil, H. K. (2002). *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River.

[100] Khater, T., Quinn, K., Pena, J., Baker, J., & Peterson, B. (1993). The latency of the cat vestibulo-ocular reflex before and after short-and long-term adaptation. *Experimental Brain Research*, 94(1), 16–32.

[101] Kim, D. & Ko, H.-S. (2007). Eulerian motion blur. In *NPH* (pp. 39–46).

[102] Kim, E. (2012). 'digital pill' with chip inside gets fda green light (http://money.cnn.com/2012/08/03/technology/startups/ingestible-sensor-proteus/).

[103] Koenig, E., Allum, J., & Dichgans, J. (1978). Visual-vestibular interaction upon nystagmus slow phase velocity in man. *Acta oto-laryngologica*, 85(1-6), 397–410.

[104] Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1), 59–69.

[105] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541–551.

[106] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

[107] Liang, J. & Green, M. (1994). A highly interactive 3d modeling system. *JDCAD*, 1, 262–270.

[LinkInstruments] LinkInstruments. Dual prob oscilloscope (http://www.linkinstruments.com/).

[109] Lisberger, S., Evinger, C., Johanson, G., & Fuchs, A. (1981). Relationship between eye acceleration and retinal image velocity during foveal smooth pursuit in man and monkey. *Journal of Neurophysiology*, 46(2), 229–249.

[110] Lo, W. & So, R. H. (2001). Cybersickness in the presence of scene rotational movements along different axes. *Applied ergonomics*, 32(1), 1–14.

[111] Loram, I. D. & Lakie, M. (2002). Direct measurement of human ankle stiffness during quiet standing: the intrinsic mechanical stiffness is insufficient for stability. *The Journal of Physiology*, 545(3), 1041–1053.

[112] Lou Maher, M., Liew, P.-S., Gu, N., & Ding, L. (2005). An agent approach to supporting collaborative design in 3d virtual worlds. *Automation in Construction*, 14(2), 189–195.

[113] Luenberger, D. G. (1964). Observing the state of a linear system. *IEEE Transactions on Military Electronics*, 8(2), 74–80.

[114] Lynch, K. (1972). *What Time is this Place?* MIT Press.

[115] MacKenzie, I. & Ware, C. (1993). Lag as a determinant of human performance in interactive systems. In *INTERACT'93 and CHI'93 conference on Human factors in computing systems* (pp. 488–493).

[116] MacNeilage, P. R., Ganesan, N., & Angelaki, D. E. (2008). Computational approaches to spatial orientation: from transfer functions to dynamic bayesian inference. *Journal of Neurophysiology*, 100(6), 2981–2996.

[117] Magoulas, G. D., Vrahatis, M. N., & Androulakis, G. S. (1997). Effective backpropagation training with variable stepsize. *Neural networks*, 10(1), 69–82.

[118] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), 674–693.

[119] Mandryk, R. L., Inkpen, K. M., & Calvert, T. W. (2006). Using psychophysiological techniques to measure user experience with entertainment technologies. *Behaviour & Information Technology*, 25(2), 141–158.

[120] Masani, K., Popovic, M. R., Nakazawa, K., Kouzaki, M., & Nozaki, D. (2003). Importance of body sway velocity information in controlling ankle extensor activities during quiet stance. *Journal of Neurophysiology*, 90(6), 3774–3782.

[121] Masson, G., Busettini, C., & Miles, F. (1997). Vergence eye movements in response to binocular disparity without depth perception. *Nature*, 389(6648), 283–286.

[122] Mayne, R. (1950). The dynamic characteristics of the semicircular canals. *Journal of Comparative and Physiological Psychology*, 43(4), 309–319.

[123] Mays, L. E., Porter, J. D., Gamlin, P., & Tello, C. A. (1986). Neural control of vergence eye movements: neurons encoding vergence velocity. *J Neurophysiol*, 56(4), 1007–1021.

[124] McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3), 419–457.

[125] McClure, J. A., Fregly, A. R., Molina, E., & Graybiel, A. (1971). *Response from arousal and thermal sweat areas during motion sickness*. Technical report, DTIC Document.

[126] McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133.

[127] Merfeld, D. & Zupan, L. H. (2002). Neural processing of gravitoinertial cues in humans. iii. modeling tilt and translation responses. *Journal of Neurophysiology*, 87(2), 819–833.

[128] Merfeld, D. M., Young, L. R., Oman, C. M., & Shelhamer, M. J. (1993). A multidimensional model of the effect of gravity on the spatial orientation of the monkey. *Journal of Vestibular Research: equilibrium & orientation*, 1, 1–10.

[129] Mergenthaler, K. & Engbert, R. (2007). Modeling the control of fixational eye movements with neurophysiological delays. *Physical Review Letters*, 98(13), 138104.

[130] Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH* (pp. 1045–1048).

[131] Mine, M. (1995). *Virtual environment interaction techniques*. Technical Report TR95-018, UNC Chapel Hill.

[132] Mine, M., Brooks Jr, F., & Sequin, C. (1997). Moving objects in space: exploiting proprioception in virtual-environment interaction. In *24th annual conference on Computer graphics and interactive techniques* (pp. 19–26).

[133] Minor, L. B. & Goldberg, J. M. (1991). Vestibular-nerve inputs to the vestibulo-ocular reflex: a functional-ablation study in the squirrel monkey. *The Journal of Neuroscience*, 11(6), 1636–1648.

[134] Money, K. (1972). Measurement of susceptibility to motion sickness. In *AGARD Conference Proceedings* (pp. B2–1–B2–4).

[135] Montello, D. R. (1998). A new framework for understanding the acquisition of spatial knowledge in large-scale environments. *Spatial and temporal reasoning in geographic information systems*, 1, 143–154.

[136] Morasso, P. G. & Schieppati, M. (1999). Can muscle stiffness alone stabilize upright standing? *Journal of Neurophysiology*, 82(3), 1622–1626.

[137] Mujber, T., Szecsi, T., & Hashmi, M. (2004). Virtual reality applications in manufacturing process simulation. *Journal of materials processing technology*, 155, 1834–1838.

[NCI] NCI. Nci dictionary of cancer terms: Hypothalamus (http://www.cancer.gov/dictionary?CdrID=46359).

[139] Newman, M. C., McCarthy, G. W., Glaser, S. T., Bonato, F., & Bubka, A. (2013). Motion sickness adaptation to coriolis-inducing head movements in a sustained g flight simulator. *Aviation, Space, and Environmental Medicine*, 84(2), 104–109.

[140] Ngoc, T. V. (2008). Medical applications of wireless networks (http://www.cse.wustl.edu/~jain/cse574-08/ftp/medical/).

[141] Nordahl, R., Serafin, S., Turchet, L., & Nilsson, N. C. (2011). A multimodal architecture for simulating natural interactive walking in virtual environments. *PsychNology Journal*, 9, 245–268.

[142] Noton, D. (1998). A bright future for light therapy (http://www.positivehealth.com/article/light-and-colour/a-bright-future-for-light-therapy).

[143] Okawa, T., Tokita, T., Shibata, Y., Ogawa, T., & Miyata, H. (1995). Stabilometry. significance of locus length per unit area (l/a) in patients with equilibrium disturbances. *Equilibrium Research*, 55(3), 283–293.

[144] Oman, C. M. (1990). Motion sickness: a synthesis and evaluation of the sensory conflict theory. *Canadian Journal of Physiology and Pharmacology*, 68(2), 294–303.

[145] Optican, L. M., Zee, D. S., & Chu, F. C. (1985). Adaptive response to ocular muscle weakness in human pursuit and saccadic eye movements. *J Neurophysiol*, 54(1), 110–122.

[146] O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation*, 8(5), 895–938.

[147] Pan, Z., Cheok, A. D., Yang, H., Zhu, J., & Shi, J. (2006). Virtual reality and mixed reality for virtual learning environments. *Computers & Graphics*, 30(1), 20–28.

[148] Pedrotti, M., Mirzaei, M. A., Tedesco, A., Chardonnet, J.-R., Mérienne, F., Benedetto, S., & Baccino, T. (2014). Automatic stress classification with pupil diameter analysis. *International Journal of Human-Computer Interaction*, 30(3), 220–236.

[149] Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., & Mine, M. (1997). Image plane interaction techniques in 3d immersive environments. In *1997 symposium on Interactive 3D graphics* (pp. 39–ff).

[150] Pierce, J. S., Stearns, B. C., & Pausch, R. (1999). Voodoo dolls: Seamless interaction at multiple scales in virtual environments. In *1999 Symposium on Interactive 3D Graphic* (pp. 141–145).

[151] Piltan, F., Haghighi, S. T., Sulaiman, N., Nazari, I., & Siamak, S. (2011). Artificial control of puma robot manipulator: A-review of fuzzy inference engine and application to classical controller. *International Journal of Robotics and Automation*, 2(5), 401–425.

[152] Pinho, M., Bowman, D., & Dal Sasso Freitas, C. (2008). Cooperative object manipulation in collaborative virtual environments. *Journal of the Brazilian Computer Society*, 14(2), 54–67.

[153] Poupyrev, I., Billinghurst, M., Weghorst, S., & Ichikawa, T. (1996). The go-go interaction technique: non-linear mapping for direct manipulation in vr. In *9th annual ACM symposium on User Interface Software and Technology* (pp. 79–80).

[154] Presson, C. C. & Montello, D. R. (1994). Updating after rotational and translational body movements: Coordinate structure of perspective space. *Perception*, 23(12), 1447–1455.

[155] Raina, R., Madhavan, A., & Ng, A. Y. (2009). Large-scale deep unsupervised learning using graphics processors. In *ICML*, volume 9 (pp. 873–880).

[156] Raphan, T., Matsuo, V., & Cohen, B. (1979). Velocity storage in the vestibulo-ocular reflex arc (vor). *Experimental Brain Research*, 35(2), 229–248.

[157] Reason, J. (1978). Motion sickness adaptation: a neural mismatch model. *Journal of the Royal Society of Medicine*, 71(11), 819–829.

[158] Reason, J. & Brand, J. (1975). *Motion sickness*. Academic Press, London.

[159] Riccio, G. E. & Stoffregen, T. A. (1991). An ecological theory of motion sickness and postural instability. *Ecological psychology*, 3(3), 195–240.

[160] Riva, G. (2004). *Cybertherapy: Internet and virtual reality as assessment and rehabilitation tools for clinical psychology and neuroscience*, volume 99. IOS Press.

[161] Robinson, D. (1977). Vestibular and optokinetic symbiosis: an example of explaining by modelling. *Control of gaze by brain stem neurons*, 1, 49–58.

[162] Robinson, D. A. (1973). Models of the saccadic eye movement control system. *Kybernetik*, 14(2), 71–83.

[163] Robinson, D. A., Gordon, J., & Gordon, S. (1986). A model of the smooth pursuit eye movement system. *Biological cybernetics*, 55(1), 43–57.

[164] Rolnick, A. & Lubow, R. (1991). Why is the driver rarely motion sick? the role of controllability in motion sickness. *Ergonomics*, 34(7), 867–879.

[165] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386–408.

[166] Ruddle, R., Payne, S., & Jones, D. (1997). Navigating buildings in desktop virtual environments: experimental investigations using extended navigational experience. *Journal of Experimental Psychology: Applied*, 3, 143–159.

[167] Ruddle, R. A., Payne, S. J., & Jones, D. M. (1999). The effects of maps on navigation and search strategies in very-large-scale virtual environments. *Journal of Experimental Psychology: Applied*, 5(1), 54–75.

[168] Ruddle, R. A. & Péruch, P. (2004). Effects of proprioceptive feedback and environmental characteristics on spatial learning in virtual environments. *International Journal of Human-Computer Studies*, 60(3), 299–326.

[169] Sadalla, E. K., Burroughs, W. J., & Staplin, L. J. (1980). Reference points in spatial cognition. *Journal of Experimental Psychology: Human Learning and Memory*, 6(5), 516–528.

[170] Saeb, S., Weber, C., & Triesch, J. (2009). A neural model for the adaptive control of saccadic eye movements. In *International Joint Conference on Neural Networks (IJCNN 2009)* (pp. 2740–2747).

[171] Sainath, T. N., Mohamed, A.-r., Kingsbury, B., & Ramabhadran, B. (2013). Deep convolutional neural networks for lvcsr. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)* (pp. 8614–8618).

[172] Sanctorum, J. (2014). Een brug voor 'tomorrowland' (http://visionairbelgie.wordpress.com/2014/02/22/hersendood/).

[173] Sanjeev Arulampalam, M., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2), 174–188.

[174] Sayers, H. (2004). Desktop virtual environments: a study of navigation and age. *Interacting with Computers*, 16(5), 939–956.

[175] Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2), 234–242.

[176] Schmidhuber, J. (2013). My first deep learning system of 1991+ deep learning timeline 1962-2013. *arXiv:1312.5548*, 1, 1–7.

[177] Scibora, L. M., Villard, S., Bardy, B., & Stoffregen, T. A. (2007). Wider stance reduces body sway and motion sickness. In *Proc. VIMS* (pp. 18–23).

[178] Setareh, M., Bowman, D. A., & Kalita, A. (2005). Development of a virtual reality structural analysis system. *Journal of architectural engineering*, 11(4), 156–164.

[179] Sharples, S., Cobb, S., Moody, A., & Wilson, J. R. (2008). Virtual reality induced symptoms and effects (vrise): Comparison of head mounted display (hmd), desktop and projection display systems. *Displays*, 29(2), 58–69.

[180] Sheridan, T. B. & Furness III, T. A. (1992). Various articles in 'spotlight' on simulator sickness. Presence.

[181] Sjölinder, M., Höök, K., Nilsson, L.-G., & Andersson, G. (2005). Age differences and the acquisition of spatial knowledge in a three-dimensional environment: evaluating the use of an overview map as a navigation aid. *International Journal of Human-Computer Studies*, 63(6), 537–564.

[182] Skalak, R. & Chien, S. (1987). *Handbook of bioengineering*. McGraw-Hill New York.

[183] Slater, M., Usoh, M., & Steed, A. (1995). Taking steps: The influence of a walking technique on presence in virtual reality. *ACM Transactions on Human Interaction*, 2, 201–219.

[SMI] SMI, V. Smi eye tracking (http://eyetracking-glasses.com/).

[185] Smith, J. (1957). The forces operating at the human ankle joint during standing. *Journal of Anatomy*, 91(Pt 4), 545–564.

[186] So, R. H., Finney, C. M., & Goonetilleke, R. S. (1999). Motion sickness susceptibility and occurrence in hong kong chinese. *Contemporary ergonomics*, 1, 88–92.

[187] So, R. H. & Griffin, M. J. (1991). Effects of time delays on head tracking performance and the benefits of lag compensation by image deflection. In *AIAA Flight Simulation Technologies Conference*, 124–130.

[188] So, R. H. & Lo, W. (1999). Cybersickness: An experimental study to isolate the effects of rotational scene oscillations. In *IEEE Virtual Reality* (pp. 237–241).

[189] Sonneborn, L. & Van Vleck, F. (1964). The bang-bang principle for linear control systems. *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, 2(2), 151–159.

[190] Sorenson, H. W. (1985). *Kalman filtering: theory and application*, volume 38. IEEE press New York.

[191] Souman, J. L., Giordano, P. R., Frissen, I., Luca, A. D., & Ernst, M. O. (2010). Making virtual walking real: Perceptual evaluation of a new treadmill control algorithm. *ACM Transactions on Applied Perception*, 7(2), 11:1–11:14.

[192] Spain, E. & Holzhausen, K. (1991). Stereoscopic versus orthogonal view displays for performance of a remote manipulation task. In *Proceedings of SPIE*, volume 1457 (pp. 103).

[193] Sphinx (2010). Multilayer perceptron (http://deeplearning.net/tutorial/mlp.html).

[194] Stanney, K. M. & Hash, P. (1998). Locus of user-initiated control in virtual environments: Influences on cybersickness. *Presence: Teleoperators and Virtual Environments*, 7(5), 447–459.

[195] Stern, R. M., Hu, S., Anderson, R. B., Leibowitz, H. W., & Koch, K. L. (1990). The effects of fixation and restricted visual field on vection-induced motion sickness. *Aviation, Space, and Environmental Medicine*, 1, 1–10.

[196] Stoakley, R., Conway, M., & Pausch, R. (1995). Virtual reality on a wim: interactive worlds in miniature. In *SIGCHI conference on Human factors in computing systems* (pp. 265–272).

[197] Stoffregen, T. A., Hettinger, L. J., Haas, M. W., Roe, M. M., & Smart, L. J. (2000). Postural instability and motion sickness in a fixed-base flight simulator. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 42(3), 458–469.

[198] Suma, E. A., Lange, B., Rizzo, A., Krum, D. M., & Bolas, M. (2011). Faast: The flexible action and articulated skeleton toolkit. In *IEEE Virtual Reality Conference (VR)* (pp. 247–248).

[199] Sylvestre, P. A., Choi, J. T., & Cullen, K. E. (2003). Discharge dynamics of oculomotor neural integrator neurons during conjugate and disjunctive saccades and fixation. *Journal of Neurophysiology*, 90(2), 739–754.

[200] Tahboub, K. A. (2009). Biologically-inspired humanoid postural control. *Journal of Physiology-Paris*, 103(3), 195–210.

[201] Tai, C.-T. (1986). Unified definition of divergence, curl, and gradient. *Applied Mathematics and Mechanics*, 7(1), 1–6.

[202] Takada, H., Fujikake, K., Miyao, M., & Matsuura, Y. (2007). Indices to detect visually induced motion sickness using stabilometry. In *First International Symposium on Visually Induced Motion Sickness, Fatigue, and Photosensitive Epileptic Seizures (VIMS2007)*, volume 1 (pp. 178–183).

[203] Takemura, A. (2000). *Change in neuronal firing patterns in the process of motor command generation for ocular following response*. PhD thesis, University of Tsukuba, Medical science.

[204] Taylor, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., & Helser, A. T. (2001). Vrpn: a device-independent, network-transparent vr peripheral system. In *ACM symposium on Virtual reality software and technology* (pp. 55–61).

[205] TechnoConcept (2007). *Techno-concept balance board manual, Notice Logiciel SabotSoft*.

[206] Tendick, F., Downes, M., Goktekin, T., Cavusoglu, M. C., Feygin, D., Wu, X., Eyal, R., Hegarty, M., & Way, L. W. (2000). A virtual environment testbed for training laparoscopic surgical skills. *Presence: Teleoperators and Virtual Environments*, 9(3), 236–255.

[207] Thabet, W., Shiratuddin, M., & Bowman, D. (2002). *Virtual reality in construction: a review*, chapter 2, (pp. 25–52). Civil-Comp press.

[208] Tonnis, M., Sandor, C., Lange, C., & Bubb, H. (2005). Experimental evaluation of an augmented reality visualization for directing a car driver's attention. In *4th IEEE/ACM International Symposium on Mixed and Augmented Reality* (pp. 56–59).

[UJI] UJI. Emg sensor (http://be.uji.es/equipamiento.htm).

[210] Ujike, H., Ukai, K., & Nihei, K. (2008). Survey on motion sickness-like symptoms provoked by viewing a video movie during junior high school class. *Displays*, 29(2), 81–89.

[211] Ukai, K. (2006). Human factors for stereoscopic images. In *IEEE International Conference on Multimedia and Expo* (pp. 1697–1700).

[Unige] Unige. Inauguration du brain behaviour laboratory (http://www.unige.ch/presse/static/BBL/).

[213] Vidal, M., Amorim, M.-A., & Berthoz, A. (2004). Navigating in a virtual three-dimensional maze: how do egocentric and allocentric reference frames interact? *Cognitive Brain Research*, 19(3), 244–258.

[214] Vingerhoets, R. A., Medendorp, W. P., & Van Gisbergen, J. A. (2006). Time course and magnitude of illusory translation perception during off-vertical axis rotation. *Journal of Neurophysiology*, 95(3), 1571–1587.

[215] Vinson, N. G. (1999). Design guidelines for landmarks to support navigation in virtual environments. In *SIGCHI conference on Human Factors in Computing Systems* (pp. 278–285).

[216] Waller, D. (2005). The walkabout: Using virtual environments to assess large-scale spatial abilities. *Computers in human behavior*, 21(2), 243–253.

[217] Waller, D., Beall, A. C., & Loomis, J. M. (2004). Using virtual environments to assess directional knowledge. *Journal of Environmental Psychology*, 24(1), 105–116.

[218] Waller, D., Hunt, E., & Knapp, D. (1998). The transfer of spatial knowledge in virtual environment training. *Presence: Teleoperators and Virtual Environments*, 7(2), 129–143.

[219] Ware, C. (1990). Using hand position for virtual object placement. *The Visual Computer*, 6(5), 245–253.

[220] Ware, C. & Jessome, D. (1988). Using the bat: A six-dimensional mouse for object placement. *IEEE Computer Graphics and Applications*, 8(6), 65–70.

[221] Watson, B., Spaulding, V., Walker, N., & Ribarsky, W. (1997). Evaluation of the effects of frame time variation on vr task performance. In *IEEE Virtual Reality Annual International Symposium* (pp. 38–44).

[222] Webb, N. A. & Griffin, M. J. (2002). Optokinetic stimuli: motion sickness, visual acuity, and eye movements. *Aviation, Space, and Environmental Medicine*, 73(4), 351–358.

[223] Wei, C.-S., Ko, L.-W., Chuang, S.-W., Jung, T.-P., & Lin, C.-T. (2011). Eeg-based evaluation system for motion sickness estimation. In *5th International IEEE/EMBS Conference on Neural Engineering (NER)* (pp. 100–103).

[224] Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harward University.

[225] Westerdahl, B., Suneson, K., Wernemyr, C., Roupé, M., Johansson, M., & Martin Allwood, C. (2006). Users' evaluation of a virtual reality architectural model compared with the experience of the completed building. *Automation in construction*, 15(2), 150–165.

[226] Wickens, C. & Baker, P. (1995). *Cognitive issues in virtual reality*, volume 1, chapter 13, (pp. 514–541). Oxford University Press.

[Wikipedia] Wikipedia. Visual cortex (http://en.wikipedia.org/wiki/Visual_cortex).

[228] Williams, B., Bailey, S., Narasimham, G., Li, M., & Bodenheimer, B. (2011). Evaluation of walking in place on a wii balance board to explore a virtual environment. *ACM Transactions on Applied Perception*, 8(3), 19:1–19:14.

[229] Witmer, B. G. & Kline, P. B. (1998). Judging perceived and traversed distance in virtual environments. *Presence: Teleoperators and Virtual Environments*, 7(2), 144–167.

[230] Witmer, B. G. & Singer, M. J. (1998). Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 7(3), 225–240.

[231] Yokota, Y., Aoki, M., Mizuta, K., Ito, Y., & Isu, N. (2005). Motion sickness susceptibility associated with visually induced postural instability and cardiac autonomic responses in healthy subjects. *Acta oto-laryngologica*, 125(3), 280–285.

[232] Young, L., Forster, J., & Van Houtte, N. (1968). A revised stochastic sampled data model for eye tracking movements. In *Fourth Annual NASA-University Conference on Manual Control*: University of Michigan.

[233] Youtube (2008). Ewc using eog (https://www.youtube.com/watch?v=q0PNihcQ1gw).

[234] Zee, D., Fitzgibbon, E., & Optican, L. (1992). Saccade-vergence interactions in humans. *Journal of Neurophysiology*, 68, 1624–1624.

[235] Zhai, S., Buxton, W., & Milgram, P. (1994). The □silk cursor□: investigating transparency for 3d target acquisition. In *SIGCHI conference on Human factors in computing systems: celebrating interdependence* (pp. 459–464).

[236] Zhai, S. & Milgram, P. (1993). Human performance evaluation of manipulation schemes in virtual environments. In *IEEE Virtual Reality Annual International Symposium* (pp. 155–161).

[237] Zhang, X. & Wakamatsu, H. (2001). A unified adaptive oculomotor control model. *International Journal of Adaptive Control and Signal Processing*, 15(7), 697–713.

[238] Zhang, X. & Wakamatsu, H. (2002). Mathematical model for binocular movements mechanism and construction of eye axes control system. *Journal of the Robotics Society of Japan*, 20(1), 89–97.

[239] Zheng, Y., McCaleb, M., Strachan, C., & Williams, B. (2012). Exploring a virtual environment by walking in place using the microsoft kinect. In *ACM Symposium on Applied Perception* (pp. 131–131).

[240] Zuber, B., Semmlow, J., & Stark, L. (1968). Frequency characteristics of the saccadic eye movement. *Biophysical journal*, 8(11), 1288–1298.

INFLUENCE DES TECHNIQUES D'INTERACTIONS SUR LE MAL DE SIMULATEUR DANS LES ENVIRONNEMENTS VIRTUELS : ESTIMATION ET PREDICTION

RESUME : La compréhension de la dynamique oculo-vestibulaire pendant un réarrangement sensoriel dans le système nerveux central joue un rôle extrêmement important pour mieux appréhender la perception humaine, et améliore la technologie dans de nombreux domaines de l'ingénierie. Par ailleurs, le conflit sensoriel qui apparaît entre les systèmes oculaire, vestibulaire et proprioceptif pendant un réarrangement sensoriel à certains instants peut affecter grandement la performance d'un utilisateur dans un grand nombre d'applications telles que la simulation de vol/conduite, les systèmes d'affichage 3D échelle 1, les affichages de grande taille, les serious games, etc. De fait, connaître la condition dans laquelle le conflit sensoriel apparaît est d'une grande importance. Cette étude a pour objectif la compréhension de la nature du conflit sensoriel par la modélisation et des études subjectives, ainsi que les conditions dans lesquelles ce conflit apparaît dans un environnement synthétique. Les résultats sont ensuite utilisés pour concevoir de meilleures interfaces de navigation et de manipulation dans des environnements virtuels immersifs et interactifs. Plusieurs éléments, tels que la surface/taille projetée du balancement postural d'un utilisateur (mesurée par le centre de gravité) après exposition au conflit, la différence entre les composantes hautes et basses fréquences du balancement, et le temps d'exposition, seront proposés comme indicateurs de ce conflit dans des processus temps réel. Enfin, la méthode proposée sera utilisée pour évaluer plusieurs interfaces de navigation. Celles-ci sont basées sur des périphériques, la marche sur place, la reconnaissance vocale, les supports nomades, ou une fusion multi-sensorielle. Nous montrerons que les interfaces naturelles engendrent moins de conflit que celles artificielles. De plus, les trajectoires de l'utilisateur et de mauvais réglages des paramètres de navigation peuvent amener à un conflit plus élevé.

Mots clés : conflit sensoriel, dynamique oculo-vestibulaire, interfaces d'interaction, éléments temps réel, environnement virtuel à l'échelle 1, réalité virtuelle.

INFLUENCE OF INTERACTION TECHNIQUES ON VIMS IN VIRTUAL ENVIRONMENTS : ESTIMATION AND PREDICTION

ABSTRACT: Understanding oculo-vestibular dynamics during sensory rearrangement in the central nervous system plays an extremely important role in better understanding human perception, and improves the technology in many engineering fields. Besides, the sensory conflict that occurs between ocular, vestibular and proprioception during sensory rearrangement at certain occasions might adversely affect the user performance in a wide variety of domains including flight/car simulators, scale-one 3D systems, large-scale displays, serious games, and so on. Therefore, knowing the condition in which the sensory conflict happens has a great deal of importance. This study aims at understanding the nature of sensory conflict by modeling and subjective studies, and the conditions in which it takes place in a synthetic environment. The results then will be used to design better navigation and manipulation interfaces in immersive and interactive Virtual Environments. A set of novel features including the area/shape of the user's COG post-exposure postural sway, the difference between LF and HF components of the sway in a frequency space, and the time of exposure will be proposed as indicators of this conflict in real-time processes. Finally, the proposed method will be used to evaluate a set of navigation interfaces. The interfaces include device-based, walking in place, speech processing, iDevice-based, and finally sensor fusion. It will be shown that naturally inspired interfaces create less conflict comparing to artificial ones. Moreover, user trajectories and inappropriate settings of navigation parameters can lead to higher conflict.

Keywords : sensory conflict, oculo-vestibular dynamics, interaction interfaces, real-time features, scale-one virtual environment, virtual reality.



ARTS ET MÉTIERS ParisTech

ParisTech INSTITUT DES SCIENCES ET TECHNOLOGIES PARIS INSTITUTE OF TECHNOLOGY