UNIVERSITY OF GENOA

PHD PROGRAM IN MECHANICS, MEASUREMENT AND ROBOTICS

Motion Generation and Planning System for a Virtual Reality Motion Simulator: Development, Integration, and Analysis

by

Mohamed Sadiq IKBAL

Thesis submitted for the degree of *Doctor of Philosophy* (33° cycle)

February 2021

Prof. Ing. Matteo ZOPPI	Supervisor
Prof. Ing. Luca A. TAGLIAFICO	Head of the PhD program
Johan CASTEGREN, *CEO, Singular Perception s.r.l*	Industrial advisor

Thesis Reviewer:

Prof. Damien CHABLAT, *CNRS-LS2N*	External examiner
Prof. Perla MAIOLINO, *Oxford Robotics Institute*	External examiner

Department of Mechanical, Energy, Management and Transport Engineering

Thesis reviewers

Prof. Damien CHABLAT, Research Director, CNRS and Head of ReV (Robotique Et Vivant) team, LS2N.

He received the Ph.D. degree in Génie Mécanique from the École Centrale de Nantes and University of Nantes, Nantes, France in 1998. For one year, he was with the Centre of Intelligent Machines, McGill University, Montreal, QC, Canada. He joined the Centre National de la Recherche Scientifique (CNRS), Paris, France, in 1999 and became the research director CNRS at Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), Nantes, France, in 2011. Since 2017, he is the head of the ReV (Robotique Et Vivant) team at Laboratoire des Sciences Numériques de Nantes (LS2N), Centrale Nantes. His research interests include robotics, design of parallel manipulators, and virtual reality in design products.
Email: Damien.Chablat@ls2n.fr
Tel: +33 (0)2 40 37 69 88,
Location: Centrale Nantes (CN), Petit Port, 1, rue de la Noë, BP 92101, 44321 NANTES Cedex 3, Batiment S, étage 4, bureau 405

Prof. Perla MAIOLINO, Associate Professor of Engineering Science, Tutorial Fellow at Brasenose College, Oxford Robotics Institute.

She received the M.S. degree in robotics and automation and the Ph.D. degree in robotics from the University of Genoa, Genoa, Italy, in 2006 and 2010, respectively. She worked as a PostDoc at the University of Cambridge where she investigated aspects related to soft sensing and soft robotics. She moved to Oxford in September 2018 as Associate Professor at the Engineering Science Department and a member of Oxford Robotics Institute. Her current research interests include the design of new technological solutions for tactile sensors to be integrated on soft robots.
Email: perla.maiolino@eng.ox.ac.uk
Tel: +44 01865 283301,
Location: Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ

Acknowledgements

I would like to express my gratitude to Prof. Matteo Zoppi, my supervisor. His continuous guidance and constant inspiration throughout the period of Ph.D. were priceless. His outstanding problem-solving and project management skills will reflect throughout my career.

I am grateful to Johan Castegren of Singular Perception for his support to work in a unique field of research. His creative inputs were immensely important.

My sincere thanks to Prof. Francesco Crenna, Prof. Marco Testa, and Prof. Fabrizio Bracco for their valuable time and expertise during the discussion on various topics.

My heartfelt gratitude to the team of researchers at the PMAR laboratory of University of Genoa, Vishal Ramadoss, Keerthi Sagar, and Lugo Calles Jesús Hiram for their friendship, technical assistance, knowledge transfer, and huge moral support.

Last but not least, I would like to thank my family, my partner, and my friends back in India who have helped and cared for me throughout my career to become the first ever graduate in my family.

Special thanks to the University of Genoa for their support during the Covid-19 pandemic, especially while lock-downs.

Abstract

In the past five years, the advent of virtual reality devices has significantly influenced research in the field of immersion in a virtual world. In addition to the visual input, the motion cues play a vital role in the sense of presence and the factor of engagement in a virtual environment. This thesis aims to develop a motion generation and planning system for the SP7 motion simulator. SP7 is a parallel robotic manipulator in a 6RSS-R configuration. The motion generation system must be able to produce accurate motion data that matches the visual and audio signals. In this research, two different system workflows have been developed, the first for creating custom visual, audio, and motion cues, while the second for extracting the required motion data from an existing game or simulation. Motion data from the motion generation system are not bounded, while motion simulator movements are limited. The motion planning system commonly known as the motion cueing algorithm is used to create an effective illusion within the limited capabilities of the motion platform. Appropriate and effective motion cues could be achieved by a proper understanding of the perception of human motion, in particular the functioning of the vestibular system. A classical motion cueing has been developed using the model of the semi-circular canal and otoliths. A procedural implementation of the motion cueing algorithm has been described in this thesis. We have integrated all components together to make this robotic mechanism into a VR motion simulator. In general, the performance of the motion simulator is measured by the quality of the motion perceived on the platform by the user. As a result, a novel methodology for the systematic subjective evaluation of the SP7 with a pool of juries was developed to check the quality of motion perception. Based on the results of the evaluation, key issues related to the current configuration of the SP7 have been identified. Minor issues were rectified on the flow, so they were not extensively reported in this thesis. Two major issues have been addressed extensively, namely the parameter tuning of the motion cueing algorithm and the motion compensation of the visual signal in virtual reality devices. The first issue was resolved by developing a tuning strategy with an abstraction layer concept derived from the outcome of the novel technique for the objective assessment of the motion cueing algorithm. The origin of the second problem was found to be a calibration problem of the Vive lighthouse tracking system. So, a thorough experimental study was performed to obtain the optimal calibrated environment. This was achieved by benchmarking the dynamic position tracking performance of the Vive lighthouse tracking system using an industrial serial robot as a ground truth system. With the resolution of the identified issues, a general-purpose virtual reality motion simulator has been developed that is capable of creating custom visual, audio, and motion cues and of executing motion planning for a robotic manipulator with a human motion perception constraint.

Table of contents

List of figures

List of tables

List of abbreviations

VR	Virtual Reality.
DOF	Degree of Freedom.
SP7	Our custom built parallel robotic manipulator with 6RSS-R (6 revolute-spherical-spherical-Revolute) configuration and 6 + 1 redundant DOF.
IMU	Inertial Measurement Unit sensor for recording linear and angular acceleration data.
MCA	Motion Cueing Algorithm.
ECEF	Earth-Centered, Earth-Fixed.
UE	Unreal Engine
UDP	User Datagram Protocol
SDK	Software Development Kit
RT	Real Time
PID	Proportional–Integral–Derivative controller
SSQ	Simulation Sickness Questionnaire
MSQ	Motion Sickness Questionnaire
GUI	Graphical User Interface
VLTS	HTC Vive’s Lighthouse Tracking System.
BS	Base Station of HTC Vive.
HMD	Head Mount Display of HTC Vive.

T	Tracker device of HTC Vive.
C	Controller device of HTC Vive.
MOCAP	MOtion CAPture system.
ASTM	ASTM International.
d_{BS}	Distance between base stations.
e_{AD}	Average distance error.
σ	Standard deviation of error.
RMSE	Root Mean Square Error.
C16	Comau NS16 1.65 foundry hand industrial serial robot.
RMSD	Root Mean Square Deviation.
E(95)	Expected error at 95 percentile.
E(97)	Expected error at 97 percentile.
E(99.7)	Expected error at 99.7 percentile.
AET	Average Error Test.
MPET	Maximum Permissible Error Test.
$\bar{d}$	Average measurement deviation from mean.
d_{max}, d_{min}	Maximum and minimum measurement deviation from mean.
$\bar{e}$	Average measurement error.
e_{max}, e_{min}	Maximum and minimum measurement error.
δ_{avg}	Expected average error/deviation from mean.
δ_{max}	Expected maximum error/deviation from mean.
H_o	Null hypothesis of the hypothesis tests.
H_a	Alternative hypothesis of the hypothesis tests.

N Sample size of measured data.

s^2 Variance of measured error.

${}_{Left}H_{Right}$ Homogenous transformation matrix of right object with respect to the left in VLTS' coordinate frame.

${}_{Left}\hat{H}_{Right}$ Homogenous transformation matrix of right object with respect to the left in C16' coordinate frame.

P Position/linear degree of freedom.

O Orientation/angular degree of freedom.

CDF Cummulative Distribution Function.

%Loss Percentage loss of tracking.

Chapter 1

Introduction

The demand for total immersion in Virtual Reality (VR) is more than ever in recent times due to its rise in the number of VR products as well as the VR supported contents [5]. Numerous products have emerged to increase immersion while being in a VR environment focusing on various aspects of immersion like tactile feedback [6], infinite omnidirectional walking [7], and more. Motion simulators have also joined this line of products to achieve complete immersion in the virtual world like yawvr [8], paleblue platforms [9], KUKA amusement rides [10], and many more products of various price ranges.

This thesis is focused on the development of the motion generation and planning system for a virtual reality-based motion simulator namely “SP7”. SP7 is a hybrid parallel robotic manipulator with an RSS configuration. SP7 is a greenfield approach to build a VR motion simulator by PMAR Robotics laboratory in collaboration with a virtual reality gaming company Singular Perception s.r.l. The developed motion planning system will be a subsystem for the VR motion simulator product. The role of a motion simulator is to provide appropriate motion cues to the users such that they perceive their movements as motion in the virtual world without producing any ill effects like motion sickness, nausea, dizziness, etc. This is achieved by feeding the right input to the human sensory system to trick the user’s spatial awareness to be in the virtual world rather than the real world. In this thesis, extensive research has been conducted on various topics concerning the motion perception of the user in the system. The motion generation and planning system have been developed such that the immersion factor is increased with the motion cues and not degraded.

In this chapter, we introduce the subsystems needed for a VR motion simulator, the challenges of motion planning algorithms due to the human motion perception factor, the state-of-the-art motion planning algorithms, the contributions and the structure of this thesis.

1.1 Sub-systems of a VR motion simulator

The components required for the development of a VR motion simulator can be broadly grouped into three main sub-systems as shown in Fig. 1.1.

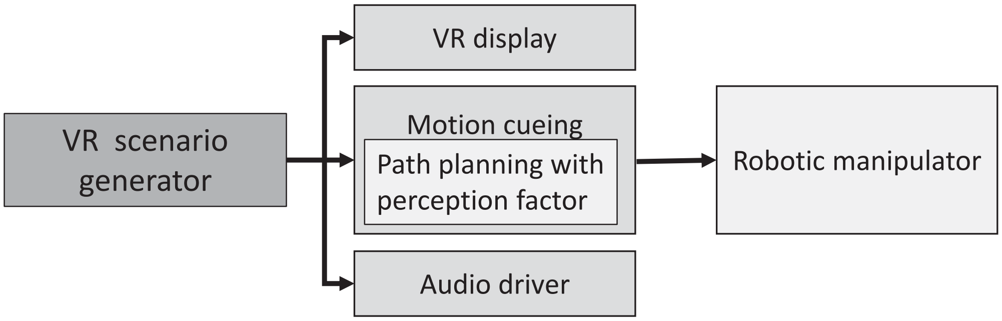


Figure 1.1 Abstract components for VR motion simulator development

1.1.1 Virtual reality scenario generator

This sub-system is responsible for creating the virtual environment to be experienced or interacted with by the user. It includes the generation of the visual data and its corresponding audio along with raw motion data. Game engines like Unreal Engine [11] and Unity[12] could be used for this purpose. The workflow for building a virtual environment is shown in Fig. 1.2 and is as follows: After the scenario planning phase, an animator uses software applications like Blender[13] to create the visual environment. Physics engines within the applications are used to generate the motion from the user frame. Then, the model of the virtual environment is imported into the Unreal engine/Unity where the interactions are programmed from the virtual reality visor (head mount display) point of view. Audio recordings and motion data are synchronized with the visual data here. Based on the requirement, both active and passive scenario could be generated in the same manner. Active/Online scenario is when the user in the virtual environment can interact within the virtual world, for instance: Flight, car simulators whereas in the passive/offline case, the user is a mere spectator with highly limited interaction with the virtual world like a VR roller coaster ride. For the passive case, one can even record the 360-degree video using cameras like [14] of a particular experience along with motion data using IMU to recreate a particular live experience. In most of our experiments, instead of Blender, we used CAD software namely PTC Creo [15] to generate the model and motion data from the simulation. Then, the Unreal engine was used to integrate the entire virtual world.

Upon the generation of the VR scenario, the final application must output the visual data to the head mount display of the VR system, the audio data to the audio driver of the system in which the application is running and, the motion data to the system running the motion cueing component. In this work, the VR system used was the HTC Vive Pro system, motion data was transmitted to the motion cueing component through the UDP network port.

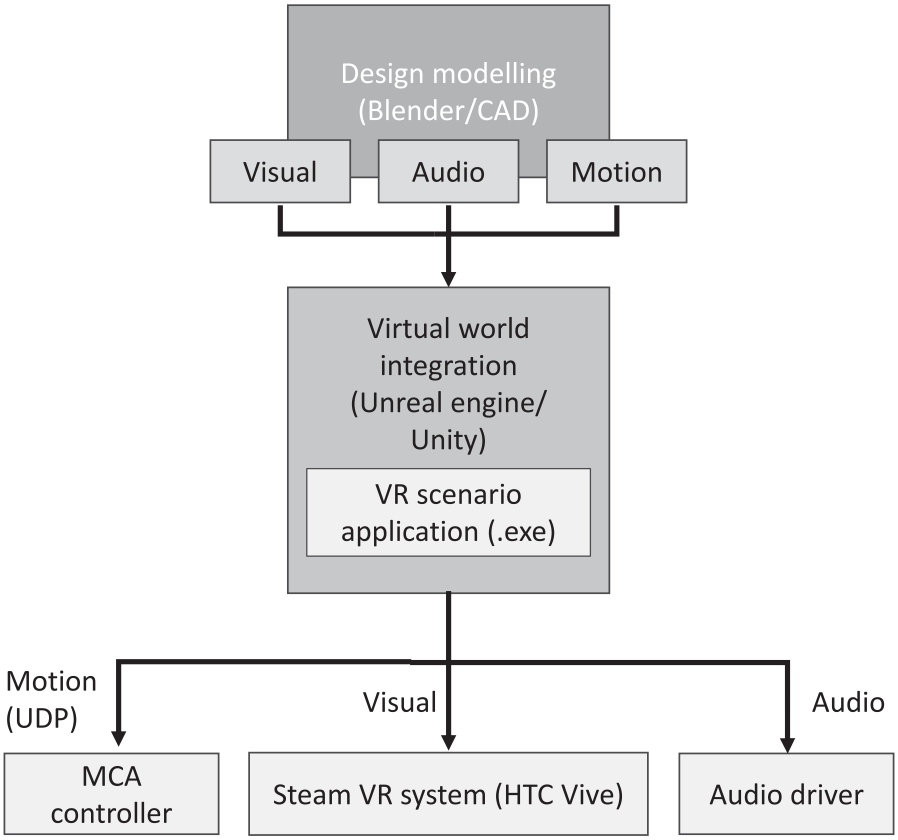

Design modelling (Blender/CAD)
Visual
Audio
Motion
Virtual world integration (Unreal engine/ Unity)
VR scenario application (.exe)
Motion (UDP)
Visual
Audio
MCA controller
Steam VR system (HTC Vive)
Audio driver

Figure 1.2 The workflow involved in the generation of a VR scenario for a motion simulator

1.1.2 Robotic manipulator

The sole function of a robotic manipulator in the context of the motion simulator is to move the user according to the output of a motion cueing algorithm. In our case, we have developed a parallel robot SP7 shown in Fig.1.3. SP7 is a 6RSS-R Stewart platform with an additional motor mounted below the platform (to yaw indefinitely). Currently, it operates in the following prescribed cuboidal workspace: $-0.05m \leq x, y \leq 0.05m$, $0.396m \leq z \leq 0.451m$,

$-5^\circ \leq \theta_1, \theta_2 \leq 5^\circ$, and $-180^\circ \leq \theta_3 \leq 180^\circ$. The platform was custom made at our laboratory.

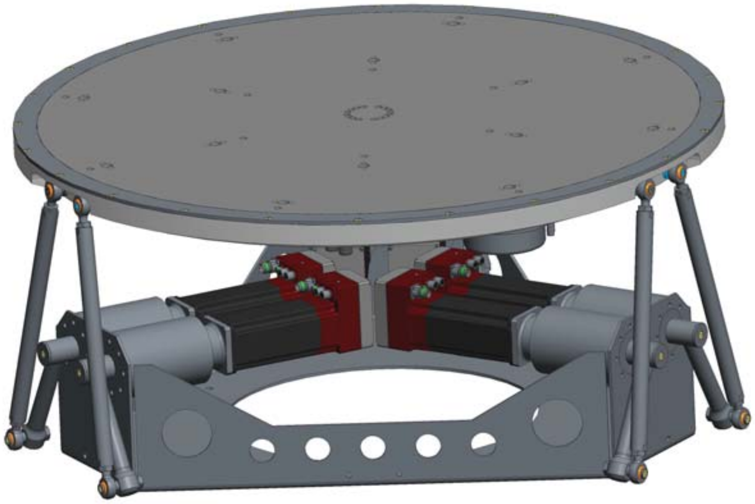


Figure 1.3 SP7 parallel robotic manipulator - CAD model

1.1.3 Motion planning

The process of mapping the actual motion data transmitted from the VR scenario application to the limited capabilities of the robotic manipulator while preserving the sense of realism is referred to as motion cueing. This is done through the motion cueing algorithms (MCA). The constraint of preserving the sense of realism makes the motion planning unique to the motion simulators. This requires knowledge of human motion perception in particular the vestibular system systems. Producing the appropriate motion cues is the vital part of a motion simulator which makes the difference between shaking the user on the platform and simulating the experience. The article [16] explains the significance of motion simulation in training or amusement experience. Motion cueing algorithms have addressed this constraint by basing the algorithm on the model of the vestibular system. The following two sub-sections 1.2 and 1.3 describes the challenges concerning human motion perception and briefly introduces the state of the art of MCA.

1.2 Motion planning and human motion perception

In addition to the usual challenges in motion planning a task for a robotic manipulation like workspace navigation, here the motion planning has a unique constraint “human motion perception” i.e., the user must be immersed in the virtual world. Before discussing the challenges in detail, first, we define the term human motion perception in the context of motion simulators. The term human motion perception is highly abstract and could refer to multiple topics as well as domain thus we define it in the context of a motion simulator as follows: ‘The perception of a motion by the user in the simulated world while the user is on the motion platform wearing a virtual reality head mount display with visual and auditory cues’.

To create a holistic experience in the virtual world, simulators must trick the human psyche by a combination of visual, aural, and motion cues. The immersion effect of vision and audio can be tricked quickly which happens most of the time when a user is watching a movie, listening to songs, etc. Whereas motion cues have powerful and irresistible effects as explained in [17]. For instance, you cannot trick a person into moving forward while being moved back even if the visual and the audio cues are highly immersive, the effect of convincing a false motion will not last as the human brain correct the false motion cues rapidly.

Extensive literature on the organs responsible for motion perception was conducted in [18] covering various aspects such as anatomy, physiology, psychophysiology, and neuropsychological. Here, we highlight a few necessary aspects pertaining to the challenges. There is also a common misconception between sensation and perception. The sensation is the initial process of detecting and encoding environmental energy whereas perception is the outcome of the organization and integration of sensations in the awareness of objects and environmental events. Multiple approaches have been proposed in [19], [20], and [21] to study sensation and perception such as structuralism, Gestalt psychology, constructive approach, direct perception, computational approach, and neurophysiological. We adopted the latest and eminent neurophysiological approach which invokes a form of reductionism: the idea that one can understand broad, apparently complex forms of behavior by studying their underlying biological processes. This approach is reasonable compared to others as our understanding of the sensory system draws extensively from neurophysiological mechanisms.

A study [22] in 1973, investigated the sensing mechanisms from a control systems point of view. The sensing mechanisms consider were the semicircular canals, the otoliths, head motion muscle spindle sensing, and body seat pressure sensing. They identified the possibility of providing false motion cues in the semicircular canals and the otoliths, mainly due to the adaptation phenomenon. A false cue is triggered by an adaptation effect several seconds after the stimulus has been removed. Another obvious yet significant finding is that the other sensory systems in the study do not measure the stimuli on the absolute scale. For instance, the pressure sensing mechanism can perceive the increase or decrease of pressure compared to the initial stimuli. It cannot say that the perceived pressure is of certain pascal units. The same applies to the body extremities, it can sense the direction but on a relative scale. In addition to the vestibular system, this small interval of error in perception can be utilized for effective illusions. Owing to the significance of the role of the vestibular system in motion simulation, the anatomy and the physiology of them was presented in [18] using the literature from [23]. Briefly, the vestibular system consists of three semicircular ducts (horizontal, anterior, and posterior) which sense the rotations in three perpendicular planes especially the angular velocity and two otolith organs (saccule and utricle) senses the translations in three planes, particularly the linear acceleration. The vestibular system employs receptors that are sensitive to the forces of gravity and acceleration acting on the head. The sensory receptors are based on minute displacements of the hair cells. The motion in the organ moves the hair cells which in turn generates neural signals. The generated independent, asymmetric, bidirectional perceived angular velocity and linear acceleration signals are summed synergistically in an AB push-pull manner [24]. The functions of the receptors and the hair cells are explained in [25].

The mathematical modeling of the vestibular system has been attempted by many but the most common and widely used model was derived by [1]. For the semicircular canal, the mechanical functioning of each canal was considered to be analogous to a torsional pendulum. In that note, the transfer function representing the input/output relation relating the cupula deflection to the head angular acceleration is given in Equation 1.1 and 1.2, and the whole model including the canal dynamics, linear adaptation dynamics, a central processing delay, and a threshold nonlinearity is shown in Fig. 1.4.

$$\frac{d_c(s)}{\ddot{\beta}(s)} = \frac{k}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.1}$$

where,

- $d_c(s)$ is the deflection of the cupula
- $\ddot{\beta}(s)$ is the head angular acceleration
- k is the stiffness of the cupula

It was found that the roots of the denominator are real and widely separated as they are overdamped and may be written as shown in Equation 1.2

$$\frac{d_c(s)}{\ddot{\beta}(s)} = \frac{k}{(s+a)(s+b)} \tag{1.2}$$

where,

- 0.04 < a < 0.2 radians/second
- $a_{nominal}$ = 0.0625 radians/second, and
- 4 < b < 300 radians/second
- $b_{nominal}$ = 10 radians/second

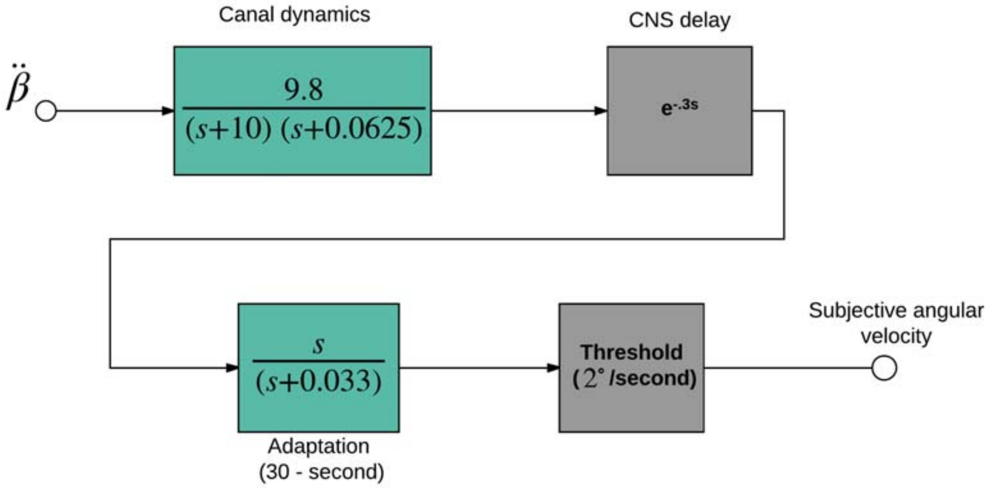


Figure 1.4 The model of the semicircular canal of the vestibular system presented in [1]

For the otoliths, the mechanical functioning was considered to be analogous to an overdamped spring-mass-dashpot linear accelerometer. To that effect, the transfer function relating the otolith displacement with respect to the macula to the linear acceleration of the head is shown in Equation 1.3 and 1.4, and the model including the otolith dynamics, a threshold nonlinearity, and a neural lead processing function is depicted in Fig. 1.5.

$$\frac{d_o(s)}{F_l(s)} = \frac{k}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.3}$$

where,

- $d_o(s)$ is the otolith displacement
- $F_l(s)$ is the linear head acceleration
- k is the stiffness of the hair attached to the maccula

It was found that the roots of the denominator are found to be overdamped and may be written as shown in Equation 1.4.

$$\frac{d_o(s)}{F_l(s)} = \frac{k}{(s+a)(s+b)} \tag{1.4}$$

where,

- $a_{nominal}$ = 0.19 radians/second, and
- $b_{nominal}$ = 1.5 radians/second

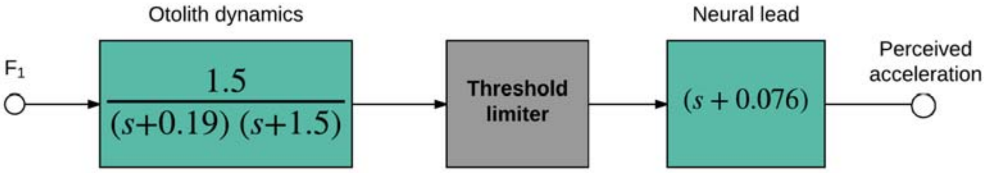


Figure 1.5 The model of the otolith organ of the vestibular system presented in [1]

Based on the observations made on the vestibular system pertaining to the human motion perception, motion planning for a motion simulator is done using the control strategy known as the motion cueing algorithm. The next sub-section 1.3 will cover the literature survey on the MCA.

1.3 Current motion planning systems for a motion simulator

As briefly introduced in the sub-section 1.1, the reproduction of realistic accelerations within a reduced workspace is done using the motion cueing algorithm which converts the stimulated physical state into actual motion for the robotic manipulator. Four major motion planning approaches have been developed for the MCA namely classical washout, adaptive washout, optimal control, and model predictive control [26]. In all these approaches, linear accelerations/specific force and angular velocities/body rate were taken as input and the pose of the robotic platform in cartesian space is returned as output at each time frame. The conversion must have enough resemblance with the real motion such that the user in the robotic platform perceives the motion to be realistic and it must be within the workspace of the robotic platform. The classical MCA was the entry point in motion planning for a motion simulator and was also implemented in our robotic platform, so we elaborate on it and briefly explain the other logic approaches.

1.3.1 Classical washout MCA

This approach was first published in [2], extensively studied in [27] and [3]. The rationale is based on the use of high-pass filters to eliminate the low-frequency motion content and to enable the replication of higher-frequency onset-type motion while ensuring that the commanded platform motion does not exceed the workspace of the platform, provided, of course, that the filter parameters are properly tuned. The schematic of the classical MCA is shown in Fig. 1.6.

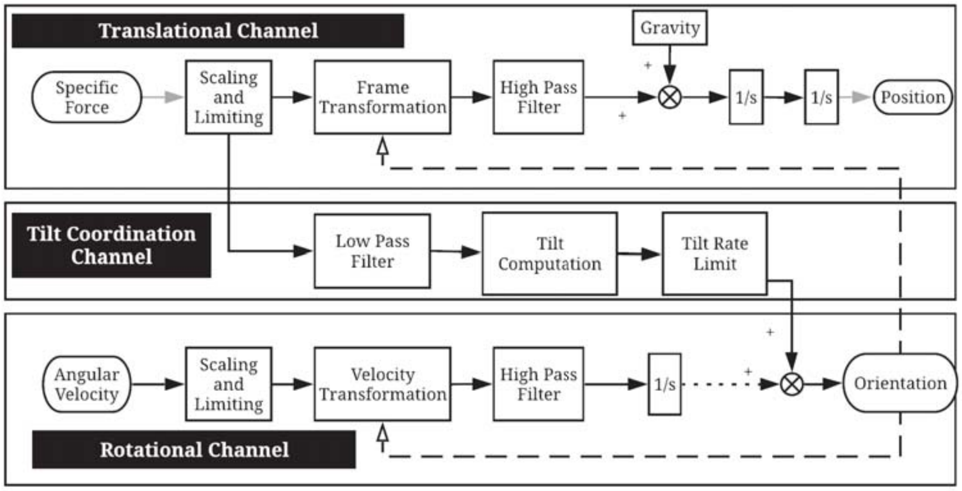


Figure 1.6 The schematic of the classical MCA. Our adaption based on [2]

The classical MCA is open-loop control. There are six inputs, three linear acceleration and three angular velocity from three perpendicular axis frame of reference. Each of the input linear accelerations and angular velocities is scaled to ensure that the workspace constraint of the robotic platform. There are two scaling techniques proposed in [3] namely linear scaling following the model of Equation 1.5 and nonlinear scaling following Equation 1.6. Fig. 1.7 shows a parallel comparison of the two scaling techniques.

$$a_x^{vr} = \begin{cases} S_x a_x^{ivr} & \text{if} \quad lim_{a_x}^{min} \leq a_x^{ivr} \leq lim_{a_x}^{max} \\ S_x lim_{a_x}^{max} & \text{if} \quad a_x^{ivr} \geq lim_{a_x}^{max} \\ S_x lim_{a_x}^{min} & \text{if} \quad a_x^{ivr} \leq lim_{a_x}^{min} \end{cases} \tag{1.5}$$

$$a_x^{vr} = \begin{cases} S_x a_x^{ivr} & \text{if} \quad lim_{a_x}^{min} \leq a_x^{ivr} \leq lim_{a_x}^{max} \\ S_x a_x^{ivr} - 0.7 S_x (a_x^{ivr} - lim_{a_x}^{max}) & \text{if} \quad a_x^{ivr} \geq lim_{a_x}^{max} \\ S_x a_x^{ivr} - 0.7 S_x (a_x^{ivr} - lim_{a_x}^{min}) & \text{if} \quad a_x^{ivr} \leq lim_{a_x}^{min} \end{cases} \tag{1.6}$$

where,

- a_x^{vr} is the limited x component of the acceleration in virtual reality frame
- S_x is the slope between $lim_{a_x}^{min}$ and $lim_{a_x}^{max}$
- a_x^{ivr} is the actual input x component of the acceleration in virtual reality frame
- $lim_{a_x}^{max}$ is the maximum limit of the x component of the acceleration of the platform
- $lim_{a_x}^{min}$ is the minimum limit of the x component of the acceleration of the platform

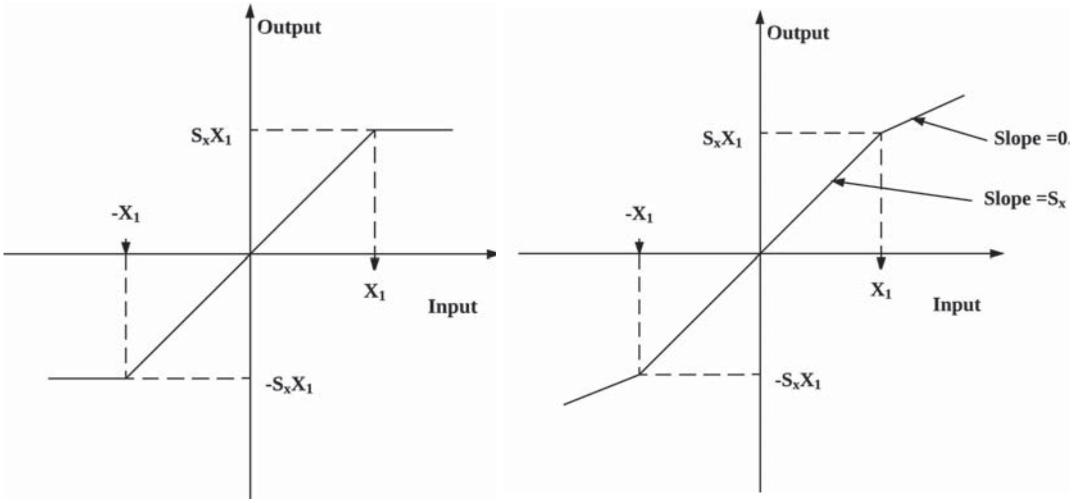


Figure 1.7 Input scaling strategies. Left: linear scaling, right: Nonlinear scaling. [3]

The scaled input is then processed through their corresponding channels: the translational DOF channel, rotational DOF channel, and tilt coordination channel.

Translational DOF channel

Here, the scaled linear acceleration is converted to specific force by removing its gravity component. Then, the coordinate transformation is done to convert the frame of reference in virtual reality to the base of the robotic platform, mainly the convention of the frame is aligned. The transformed specific force is fed through a high-pass third order filter whose transfer function $H_f(s)(s$ is the Laplace variable) is shown in Equation 1.7. The filtered specific force is integrated twice to obtain the position. This is done for all the three linear acceleration input.

$$H_f(s) = \frac{s^3}{(s+a_h)(s+b_h)(s+c_h)} \tag{1.7}$$

Rotational DOF channel

The scaled angular velocities are converted to Euler angle rates and then filtered using a second order high pass filter of the form in Equation 1.8 with transfer function $H_\omega(s)$. The filtered signals are integrated once to obtain the orientation command for the platform.

$$H_\omega(s) = \frac{s^2}{(s+a_r)(s+b_r)} \tag{1.8}$$

Tilt coordination channel

The purpose of this channel has been clearly explained in [17]. The body tilt can be used for the simulation of the sustained linear acceleration i.e., long duration, low-frequency linear motions. Thus, the scaled specific force is filtered using the low pass filter of the form shown in Equation 1.9.

$$L_f(s) = \frac{a_L b_L c_L}{(s+a_L)(s+b_L)(s+c_L)} \tag{1.9}$$

The filtered specific force are then transformed from the frame of the simulation body (ex. vehicle, flight, etc) to the ECEF frame using $\mathbf{R}$ in Equation. 1.10. The SP7 DOF can be represented in terms of its position in space and orientation defined by Euler angles (following the convention: x, y, z (roll-pitch-yaw)). Considering an ECEF reference frame for the SP7 and $\theta_1, \theta_2, \theta_3$ being the three Euler angles in the aforementioned order (roll-pitch-yaw), the rotation matrix is given by:

$$\mathbf{R} = \mathbf{R}_x(\theta_1) \cdot \mathbf{R}_y(\theta_2) \cdot \mathbf{R}_z(\theta_3),$$

where $\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$ are the matrices defining rotation along x, y, z axis by $\theta_1, \theta_2, \theta_3$ radians respectively. Therefore,

$$\mathbf{R} = \begin{bmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3 \end{bmatrix} \text{ with} \tag{1.10}$$

$$\mathbf{m}_1 = \begin{bmatrix} \cos\theta_2 \cos\theta_3 \\ \cos\theta_1 \sin\theta_3 + \cos\theta_3 \sin\theta_1 \sin\theta_2 \\ \sin\theta_1 \sin\theta_3 - \cos\theta_1 \cos\theta_3 \sin\theta_2 \end{bmatrix},$$

$$\mathbf{m}_2 = \begin{bmatrix} -\cos\theta_2 \sin\theta_3 \\ \cos\theta_1 \cos\theta_3 - \sin\theta_1 \sin\theta_2 \sin\theta_3 \\ \cos\theta_3 \sin\theta_1 + \cos\theta_1 \sin\theta_2 \sin\theta_3 \end{bmatrix},$$

$$\mathbf{m}_3 = \begin{bmatrix} \sin\theta_2 \\ -\cos\theta_2 \sin\theta_1 \\ \cos\theta_1 \cos\theta_2 \end{bmatrix}$$

Then the transformed angular velocities along x, y, z ($\omega = [\omega_x, \omega_y, \omega_z]^T$) to $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3]^T$ is given by the following relation:

$$\dot{\theta} = \mathbf{T} \cdot \omega \tag{1.11}$$

$$\mathbf{T} = \frac{1}{\cos\theta_2} \begin{bmatrix} 1 & \sin\theta_1 \sin\theta_2 & -\cos\theta_1 \sin\theta_2 \\ 0 & \cos\theta_1 \cos\theta_2 & \sin\theta_1 \cos\theta_2 \\ 0 & -\sin\theta_1 & \cos\theta_1 \end{bmatrix}$$

Finally, the tilt angle's relationship between low frequency acceleration, $\mathbf{f}_L = [f_{L_x}\ f_{L_y}\ f_{L_z}]^T$, and Euler angles is shown in Equation 1.12 and 1.13

$$\begin{bmatrix} f_{L_x} \\ f_{L_y} \\ f_{L_z} \end{bmatrix} = -g \cdot \begin{bmatrix} \sin\theta_2 \\ -\cos\theta_2 \sin\theta_1 \\ \cos\theta_1 \cos\theta_2 \end{bmatrix} \tag{1.12}$$

When these angles are small then,

$$\begin{aligned} \theta_2 &= -f_{L_x}/g \\ \theta_1 &= f_{L_y}/g \end{aligned} \tag{1.13}$$

This computation results in the tangent of the angle. These tilt angles are further restricted to guarantee that tilt occurs below the perception threshold of $3^\circ/s$ and $0.3^\circ/s^2$ for the roll motion and then added to the rotational channel's output. The usage of the perception threshold is debated towards an agreement in [28] and against in [29]. In our implementation, the perception threshold was used because it was found during our experiments that too much tilt disrupts the immersion.

1.3.2 Adaptive MCA

Two significant adaptive strategies were identified in [30] and [31]. There are a few more works of literature on this approach with no actual implementation on a motion simulator thereby no way of knowing the effectiveness of their method from a human motion perspective, so they were not considered. The difference of this technique is to adapt the severity of the washout filters according to the current state of the motion simulator thereby allowing maximum usage of the workspace. The schematic is similar to the classical MCA but the loop is closed here. A new parameter known as the adaptive gain is introduced in this method to tune the intensity of the acceleration and the angular velocity with the steepest descent method. The functionalities are summarized in [18] with implementable equations. The difference between [30] and [31] is that the latter has more parameters to tune as well as more terms in the cost function to increase the effect of the adaptive washout filter.

1.3.3 Optimal MCA

This approach was originally presented in [32] to serve as an improvement to the adaptive algorithm. This approach treats the motion cueing as a tracking problem. To be exact, the accelerations perceived in the simulator should track the accelerations that would be perceived in the real vehicle as closely as possible. The perception error between the real and the MCA output is minimized with the constraint of the capabilities of the robotic platform. Linear quadratic regulator tracking formulation is used to produce the washout filters. The implementation is also described in [27]. It is reported in [33] that this approach is not a common approach due to the requirement of a good model for perception and the subjective nature of tuning weights of the cost function.

1.3.4 Model predictive MCA

The actual model of the virtual world is used in the loop here. Introduced by [34] for a driving simulator, this approach is designed to be specific for a particular simulation like flight simulation, car simulation, etc. The model of the object to be simulated is inculcated in the algorithm to produce effective motion cues. The predictive strategy runs in parallel to the filtering components. An activation law is introduced in this approach, this law allows the motion cues output if the platform motion matches the vehicle motion otherwise will make the platform return to its home position. This approach does not require tuning, but it is a computationally expensive process in real-time, thus in [34], a method is employed where the reachable state set is precalculated such that the problem becomes a single-step optimization.

1.4 Contributions

This thesis aims to make a parallel robotic manipulator a full-fledged virtual reality motion simulator. This objective has been achieved by completing the following:

- Design, development, integration, and testing of the motion generation and planning system software architecture for converting the robotic manipulator to the VR motion simulator.
- A methodology for the systematic subjective evaluation of SP7 with a customized questionnaire focusing on human motion perception.
- A benchmark on the HTC Vive Lighthouse tracking system's dynamic pose tracking performance used to overcome motion compensation problems in the SP7.
- A novel procedure and software toolkit for tuning the motion cueing algorithm parameters using the principle of the abstraction layer that emerges from the failed exploration of a novel methodology for the objective assessment of motion perception in SP7.

1.5 Thesis structure

The thesis is split into six chapters. After the introduction in this chapter, the thesis organization is as follows:

Chapter 2 describes the software architecture developed for all three components of the virtual reality motion simulator SP7. This chapter presents two different workflows to generate the virtual reality scenario and a complete implementation procedure for the motion planning system and reports the trajectory tracking performance of the SP7 along with the constraints that were introduced to the robotic manipulator to enhance the motion perception.

Chapter 3 focuses on the evaluation of the SP7, in particular on the quality of the motion perceived by the user on the SP7. The procedure for converting SP7 into a flight simulator using X-Plane is elaborated along with a workaround workflow to conduct the subjective evaluation. The subjective evaluation of motion perception using the SP7 flight simulator is presented with a detailed study of the literature, the design of the experiment, the development of the questionnaire, and the results. Based on this outcome, several issues were highlighted in the current configuration of SP7.

Chapter 4 concentrates on solving the motion compensation problem using the Vive lighthouse tracking system. A comprehensive literature review has been presented here on the VLTS's motion tracking performance evaluations. This was followed by a description of the experimental setup that was developed using an industrial serial robot as a ground truth system. Statistically, the design and outcomes of static precision and accuracy have been documented. A benchmark for dynamic pose tracking has been developed in this chapter. The human wrist trajectory was utilized to validate the results. A procedure to calibrate the VLTS using our new method of error estimation is presented here in detail.

Chapter 5 addresses the issue of the parameter tuning of the motion cue algorithm. A literature study on the current approaches to tuning the MCA as well as the objective evaluation of the MCA is presented. A novel technique for objective evaluation of the MCA is explored here. After reviewing the results of our novel approaches, the procedure for tuning the parameters of MCA using a custom-developed abstraction layer concept has been reported.

Chapter 6 concludes this thesis by summarizing the contributions and points to a possible future direction of research in the MCA.

Appendix A: Blueprint flow of the custom cue generation system described in chapter 2.1.1.

Appendix B: A sample of questionnaire filled out by a jury for subjective evaluation of motion perception described in chapter 3.

Appendix C: Publication work during the period of this thesis

Appendix D: A list of videos produced for this thesis.

Chapter 2

Software architecture of SP7

As explained in Chapter 1, there are three main building blocks in the virtual reality motion simulator and the same classification can be extended to the software architecture, such as:

1. Virtual reality scenario generation

2. Motion cueing algorithm

3. Robotic platform manipulation

In this chapter, the software architecture of these modules are discussed in detail, concentrating on design and rationale, while briefly explaining the implementation

2.1 Virtual reality scenario generation

This segment of the system is responsible for the development of the virtual world in which the user is immersed, including visual, audio, and raw motion cues. For this reason, two approaches have been developed, firstly the custom cue generation method in which a personalized virtual world is created from scratch, while the second approach extracts the motion data from a specific game.

2.1.1 Custom cue generation

The primary aim of this approach is to build a framework from which customizable visual, audio, and motion cues could be produced as needed. The need for such a device stems from the fact that the SP7 is a custom-made platform with a custom motion cueing algorithm that needs thorough testing to verify each aspect of its functionality. It helps us to personalize all three cue signals sent to the SP7.

An outline of the workflow of this system can be found in Fig. 2.1. The first step is the modeling of objects that will be present in the virtual world. This is done by modeling software programs such as blender, PTC Creo, etc. The main components to be modeled in this process are those with which the user interacts, for example in the case of a driving simulator: vehicle, lane, tunnel, etc. In our system, we used PTC Creo to construct the model and the simulation motion data. The motion data is exported as '.csv' files with the timestamp, linear accelerations, and angular velocities. Then, The CAD model is imported into the Unreal Engine using the package "Datasmith" [35]. The motion data is imported into the table datatype of the UE. The virtual environment like mountains, a desert can be customized in the UE itself. Then, the visual, audio, and motion data timestamps are synchronized. The visual and audio data are connected to the SteamVR [36] plugin in the UE which will send the data to the VR device connected to the system. The motion data is sent through a UDP socket created in the UE application to the controller of the MCA.

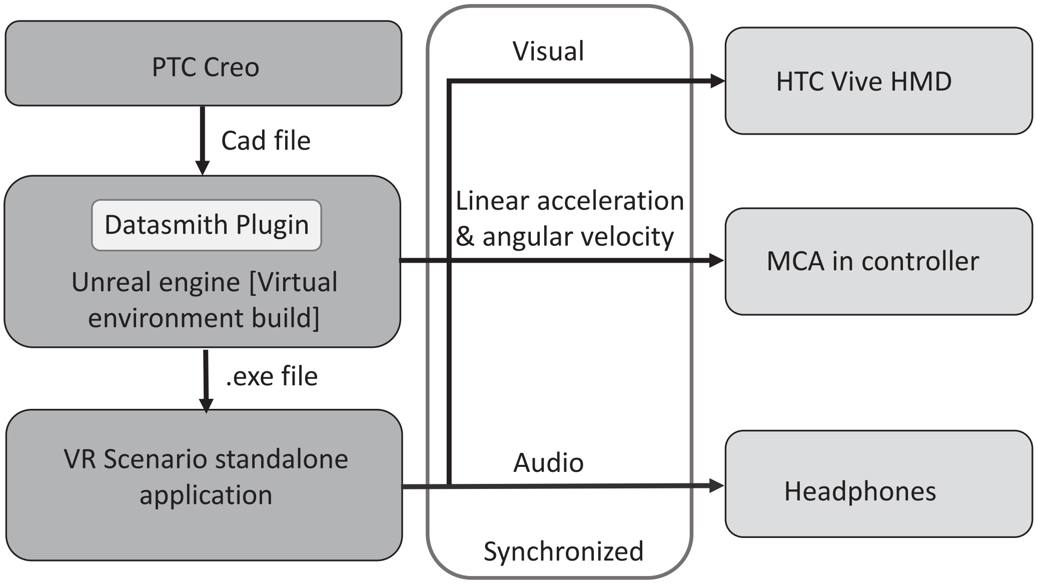

PTC Creo
Cad file
Datasmith Plugin
Unreal engine [Virtual environment build]
.exe file
VR Scenario standalone application
Visual
Linear acceleration & angular velocity
Audio
Synchronized
HTC Vive HMD
MCA in controller
Headphones

Figure 2.1 Workflow for custom cue generation framework.

The following is an illustration of the framework stated above. The purpose of the application to be built is to test the motion perception quality of the motion cueing algorithm. Nearly 80% of the immersion is achieved with visual cues (as mentioned in [37], [38], etc) so, to enable to user to focus on the motion, a tunnel with no distractive elements is created.

In the modeling phase, the model of the tunnel is designed in PTC Creo as shown in Fig. 2.2 and the CAD simulation data is exported ot a '.csv' file, a snippet of the file is shown in Fig. 2.3. In the integration phase, the CAD model is imported into the UE using the Datasmith plugin, Fig. 2.4 shows a preview of the imported tunnel. The motion data file is imported as an animation key frames inside the UE application. Fig. 2.5 shows the snippet of the imported simulation data as key frames. Then, the virtual world is built around the model, in our case, there should be no such model as it would contradict our application intent. Fig. 2.6 shows the snippet of the view of the tunnel in the virtual world. Once, the static components are integrated, the motion of the VR camera perspective is linked to the imported motion data key frames, so that the camera moves as per the simulation data imported. The VR camera corresponds to the vision of presented in the HMD display. Then, the VR camera motion data is sent to the MCA controller through a UDP port. Finally, to regulate the starting, resetting, pausing and more, key bindings are applied in the blueprint of the application. For example,"S" to start the motion, "R" to reset, "P" to stop, and so on. The blueprint of the flow of the integrated application is shown in the Appendix A. Finally, after testing the flow with the UE, a standalone application is created using the UE exporter. This standalone application will send visual and audio data to steam VR devices connected to the system as well as the motion data to the controller of the MCA.

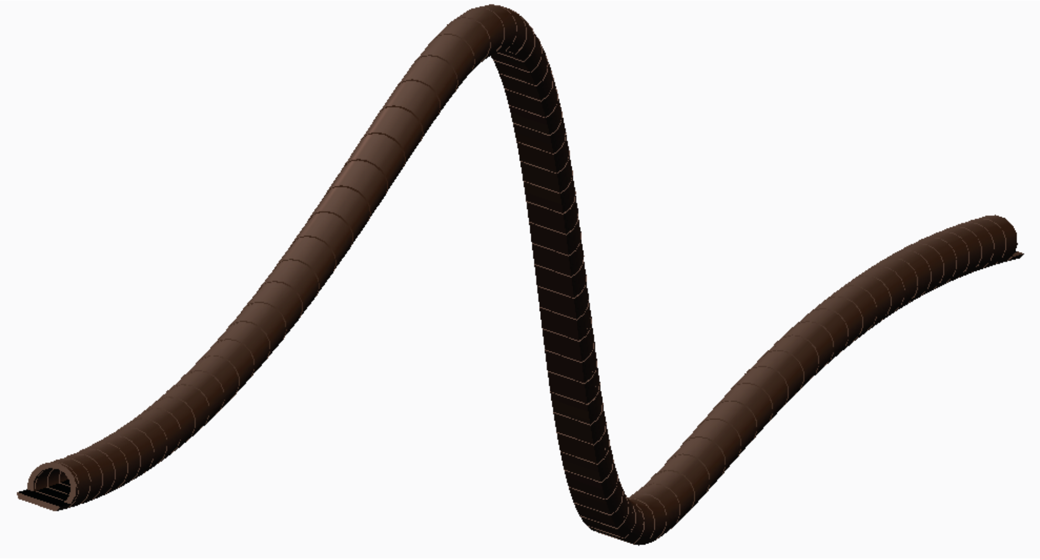
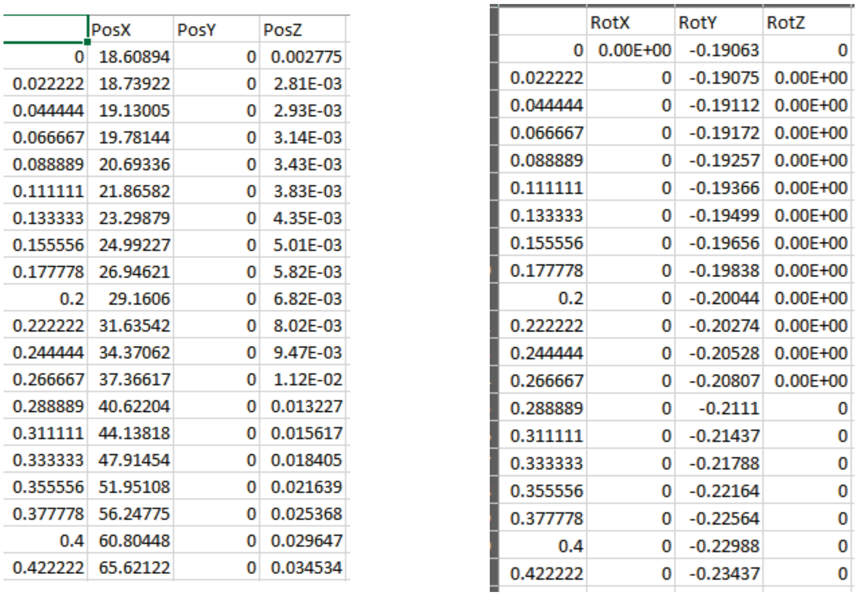


Figure 2.2 Perspective view of the CAD model created in PTC Creo Parametric 2.0. Pitch scenario model: a tunnel with equally spaced slots for lighting purpose



	PosX	PosY	PosZ
0	18.60894	0	0.002775
0.022222	18.73922	0	2.81E-03
0.044444	19.13005	0	2.93E-03
0.066667	19.78144	0	3.14E-03
0.088889	20.69336	0	3.43E-03
0.111111	21.86582	0	3.83E-03
0.133333	23.29879	0	4.35E-03
0.155556	24.99227	0	5.01E-03
0.177778	26.94621	0	5.82E-03
0.2	29.1606	0	6.82E-03
0.222222	31.63542	0	8.02E-03
0.244444	34.37062	0	9.47E-03
0.266667	37.36617	0	1.12E-02
0.288889	40.62204	0	0.013227
0.311111	44.13818	0	0.015617
0.333333	47.91454	0	0.018405
0.355556	51.95108	0	0.021639
0.377778	56.24775	0	0.025368
0.4	60.80448	0	0.029647
0.422222	65.62122	0	0.034534

	RotX	RotY	RotZ
0	0.00E+00	-0.19063	0
0.022222	0	-0.19075	0.00E+00
0.044444	0	-0.19112	0.00E+00
0.066667	0	-0.19172	0.00E+00
0.088889	0	-0.19257	0.00E+00
0.111111	0	-0.19366	0.00E+00
0.133333	0	-0.19499	0.00E+00
0.155556	0	-0.19656	0.00E+00
0.177778	0	-0.19838	0.00E+00
0.2	0	-0.20044	0.00E+00
0.222222	0	-0.20274	0.00E+00
0.244444	0	-0.20528	0.00E+00
0.266667	0	-0.20807	0.00E+00
0.288889	0	-0.2111	0
0.311111	0	-0.21437	0
0.333333	0	-0.21788	0
0.355556	0	-0.22164	0
0.377778	0	-0.22564	0
0.4	0	-0.22988	0
0.422222	0	-0.23437	0

Figure 2.3 Snippet of the .csv data file exported from the PTC Creo simulation.

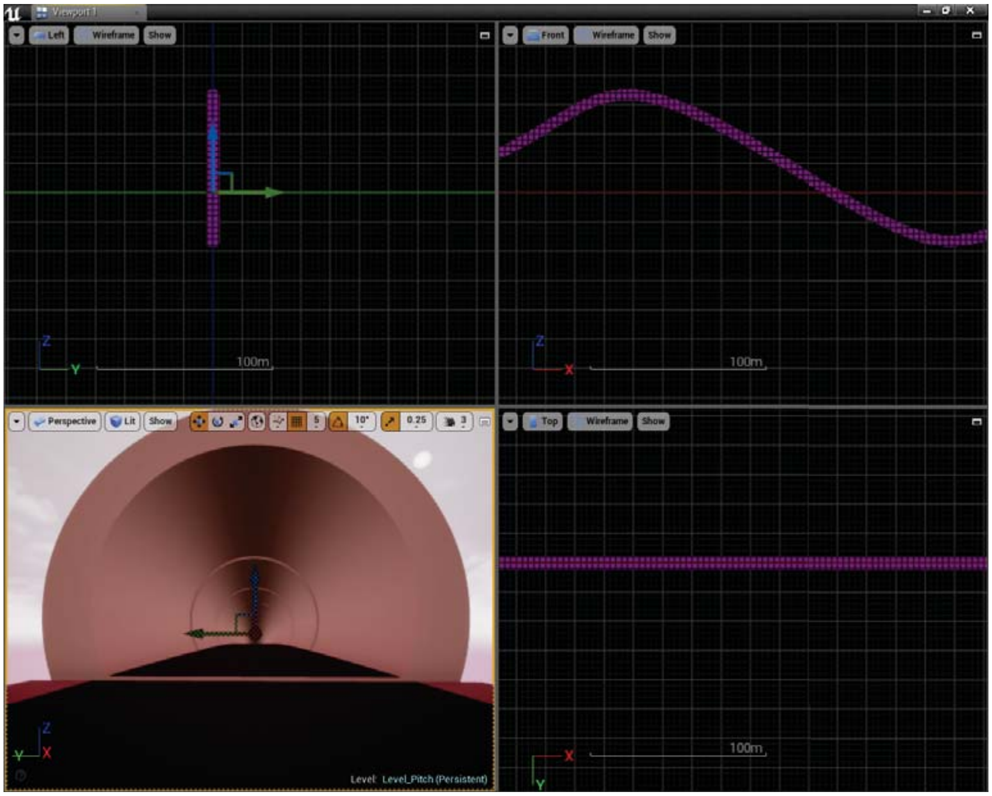


Figure 2.4 A preview of the tunnel CAD model imported in UE using the Datasmith plugin.

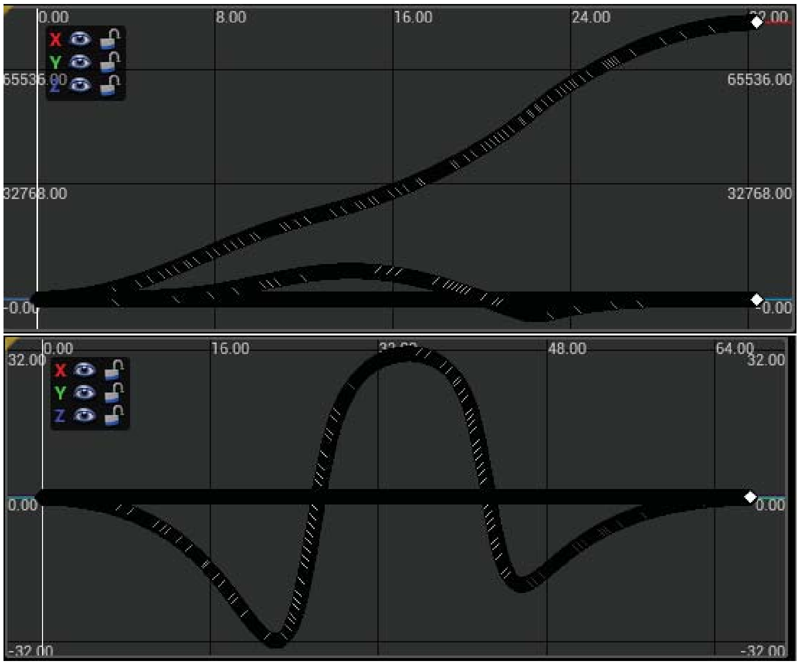


Figure 2.5 A snippet of the motion data file imported into Unreal engine as animation key frames. Top: position, Bottom: orientation

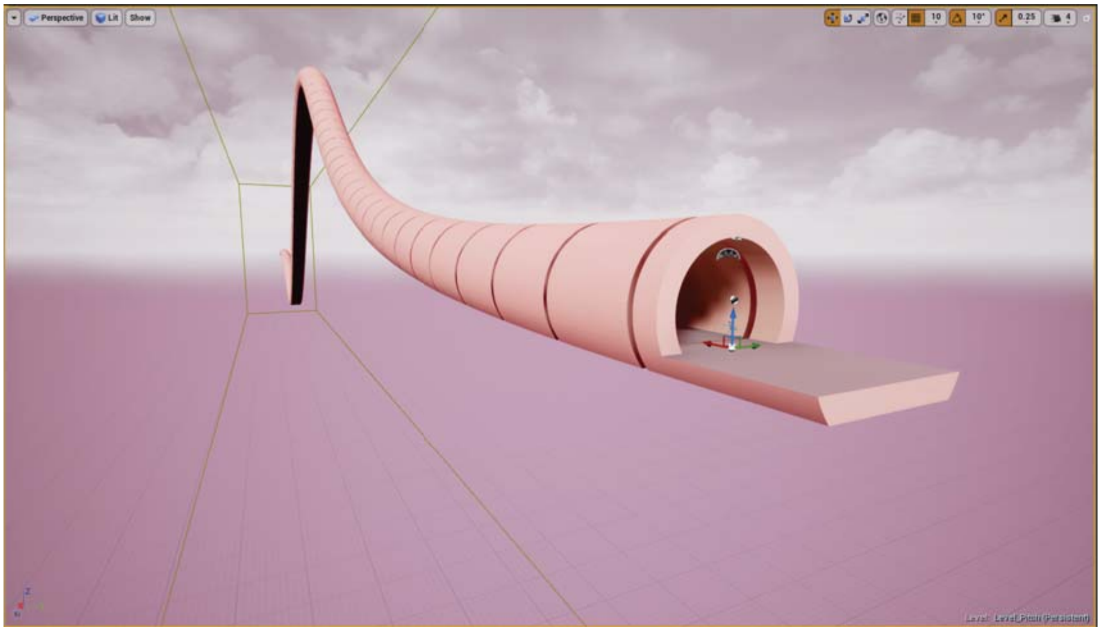


Figure 2.6 A preview of the tunnel in the virtual world.

Instead of using PTC Creo for modeling, Blender software may do the same process. An example of such an implementation is seen in Fig. 2.7. Here, the concept was to simulate a ship sailing experience in an ancient ruin. Apart from the model building, the rest of the process is the same as the one mentioned above. Here, instead of simulating the motion, the experience is made active where the user controls the motion of the ship. This feature has been implemented in the UE game engine.

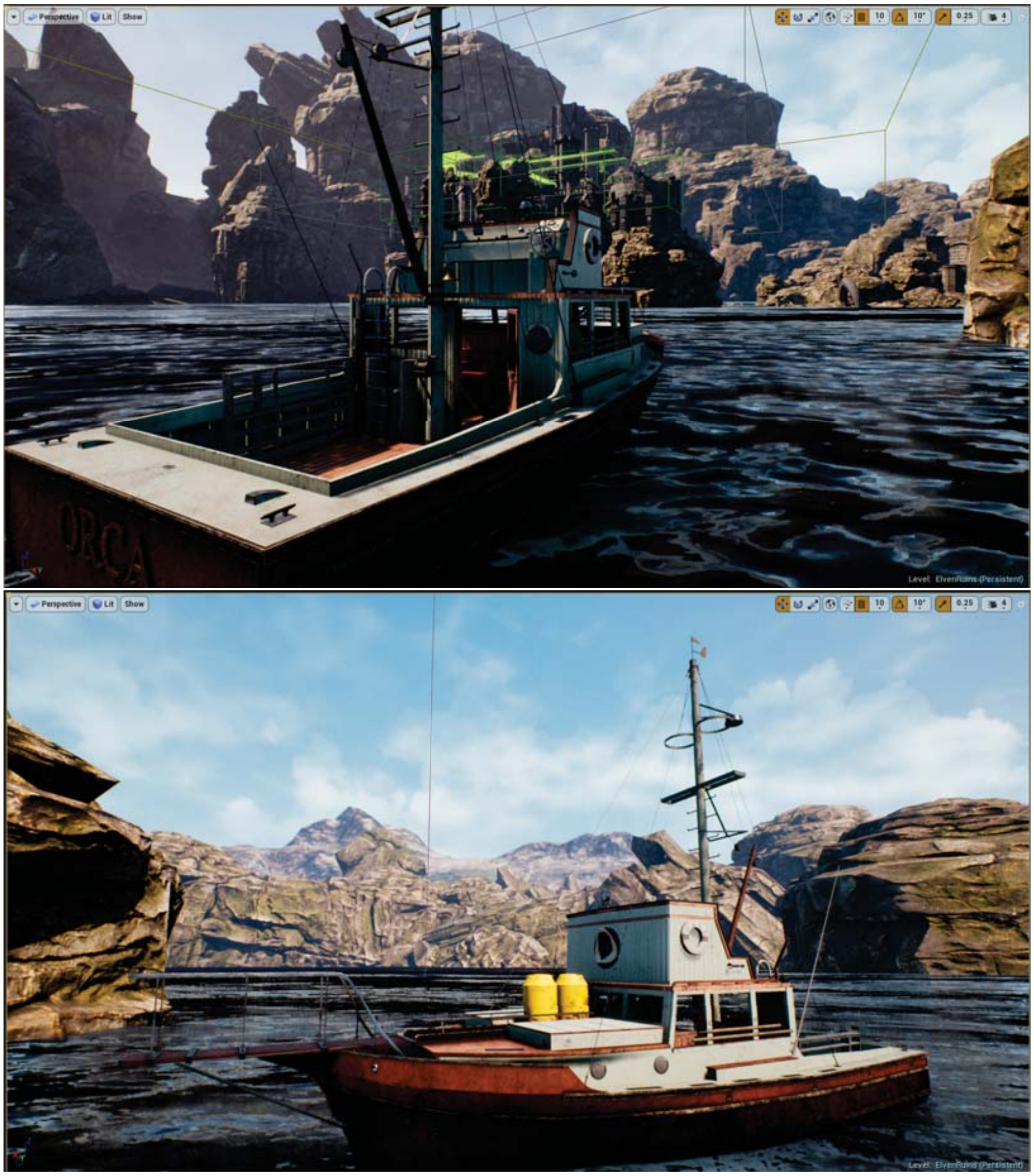

Perspective
Lit
Show
10
10°
0.25
4
ORCA
Level: ElvenRuins (Persistent)
Perspective
Lit
Show
10
10°
0.25
4
Level: ElvenRuins (Persistent)

Figure 2.7 A preview of the ship voyage simulation experience created with Blender and UE.

2.1.2 Extraction of motion cues from an existing game engine

This framework has been designed to allow the SP7 to be compatible with commercially available VR games. VR games come with visual and auditory cues for player immersion. Motion data from the game's physics engine must be extracted and sent to the MCA controller in the background. In our case, the flight simulator "X-Plane" was used during the initial experiments to allow flight simulation with SP7. In the X-Plane SDK [39], a plugin with a sample code is available in [40], this can be customized to extract metadata of the linear acceleration and angular velocity data at a required frame rate and send it to the MCA controller via a UDP port. As this system is merely an integration of the existing code in [40] with a UDP port, this part is not focused on this thesis.

2.2 Motion cueing algorithm

The classic motion cueing algorithm was the first algorithm to be introduced in the SP7. As explained in chapter 1.3.1, there is a great deal of literature describing the definition of the MCA and specifying the filter equations, but there is no current literature on the implementation methodology of the same. The main focus of the thesis is also on the architecture, development, integration, and testing of MCA. Therefore, for the above purposes, here I will elaborate on the algorithmic implementation of the MCA that has been developed during this research work. High-level principles for scaling, restricting, filtering, and tilt coordination are presented in Chapter 1.3.1. Now, this section will concentrate on discrete-time implementation and further explain the flow of algorithms in detail.

2.2.1 Discrete time implementation

The general architecture of the classical MCA is shown in Fig. 1.6. It has four sub parts namely scaling and limitation, translational DOF channel, rotational DOF channel, and tilt-coordination. For scaling and limitation, and tilt coordination, for the discrete time implementation we simply use the equations 1.5, 1.6, and 1.13 correspondingly at each frame. The translation DOF channel, rotational DOF channel and the first part of tilt coordination uses filters as shown in equation 1.7, 1.8 and 1.9 respectively. These filters are represented in continuous time Laplace domain, the discrete time implementation will require the use of bilinear transformation as shown in Equation 2.1.

$$2F_s\frac{z-1}{z+1} \longleftarrow s, \tag{2.1}$$

here z denotes the Z transform operator and F_s being the sampling frequency. As this is not a straight-forward conversion, the derivation is as follows:

For the translational DOF channel, the discrete time transfer function $H_f(z)$ of third order high pass filter described in Equation. 1.7 is given by:

$$H_f(z) = \frac{p_{4_h} \cdot (1 - 3z^{-2} + 3z^{-2} - z^{-3})}{1 + p_{1_h} z^{-1} + p_{2_h} z^{-2} + p_{3_h} z^{-3}} \text{ with} \tag{2.2}$$

$$p_{1_h} = \frac{B_h}{A_h},\ \ p_{2_h} = \frac{C_h}{A_h},\ \ p_{3_h} = \frac{D_h}{A_h},\ \ p_{4_h} = \frac{1}{A_h},$$

$$\begin{aligned} A_h &= \frac{\alpha_h}{2F_s} + \frac{\beta_h}{4F_s^2} + \frac{\gamma_h}{8F_s^3} + 1, \\ B_h &= -\frac{\alpha_h}{2F_s} + \frac{\beta_h}{4F_s^2} + 3\frac{\gamma_h}{8F_s^3} - 3, \\ C_h &= -\frac{\alpha_h}{2F_s} - \frac{\beta_h}{4F_s^2} + 3\frac{\gamma_h}{8F_s^3} + 3, \\ D_h &= \frac{\alpha_h}{2F_s} - \frac{\beta_h}{4F_s^2} + \frac{\gamma_h}{8F_s^3} - 1, \end{aligned}$$

$$\alpha_h = a_h + b_h + c_h,\ \beta_h = a_h b_h + b_h c_h + c_h a_h,\ \gamma_h = a_h b_h c_h.$$

Therefore, discrete time equation of the filter with input u and output y is given by:

$$y(k) = -\sum_{m=1}^{3} p_{m_h} y(k-m) + p_{4_h} \sum_{n=0}^{3} d_n \cdot u(k-n), \tag{2.3}$$

with $\mathbf{d} = \begin{bmatrix} 1 & -3 & 3 & -1 \end{bmatrix}^T$;$d_n$ represents n^{th} element of $\mathbf{d}$.

For the rotational DOF channel, the discrete time transfer function $H_\omega(z)$ of second order high pass filter described in Equation. 1.8 is given by:

$$H_\omega(z) = \frac{p_{3_r} \cdot (1 - 2z^{-1} + z^{-2})}{1 + p_{1_r} z^{-1} + p_{2_r} z^{-2}} \text{ with} \tag{2.4}$$

$$p_{1_r} = \frac{B_r}{A_r},\ \ p_{2_r} = \frac{C_r}{A_r},\ \ p_{3_r} = \frac{1}{A_r},$$

$$A_r = 1 + \frac{\alpha_r}{2F_s} + \frac{\beta_r}{4F_s^2},\ \ B_r = -2 + 2\frac{\beta_r}{4F_s^2},\ \ C_r = 1 - \frac{\alpha_r}{2F_s} + \frac{\beta_r}{4F_s^2},$$

$$\alpha_r = a_r + b_r \ \text{ and } \ \beta_r = a_r b_r$$

Therefore, discrete time equation of the filter with input u and output y is given by:

$$y(k) = -\sum_{m=1}^{2} p_{m_r} y(k-m) + p_{3_r} \sum_{n=0}^{2} c_n \cdot u(k-n), \tag{2.5}$$

with $\mathbf{c} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T$; c_n represents n^{th} element of $\mathbf{c}$.

For the tilt coordination channel, the discrete time transfer function $L_f(z)$ of third order low pass filter described in Equation. 1.9 is given by:

$$L_f(z) = \frac{p_{4_L} \cdot (1 + 3z^{-1} + 3z^{-2} + z^{-3})}{1 + p_{1_L} z^{-1} + p_{2_L} z^{-2} + p_{3_L} z^{-3}} \text{ with} \tag{2.6}$$

$$p_{1_L} = \frac{B_L}{A_L},\ p_{2_L} = \frac{C_L}{A_L},\ p_{3_L} = \frac{D_L}{A_L},\ p_{4_L} = \frac{\frac{\gamma_L}{8F_s^3}}{A_L},$$

$$\begin{aligned} A_L &= \frac{\alpha_L}{2F_s} + \frac{\beta_L}{4F_s^2} + \frac{\gamma_L}{8F_s^3} + 1 \\ B_L &= -\frac{\alpha_L}{2F_s} + \frac{\beta_L}{4F_s^2} + 3\frac{\gamma_L}{8F_s^3} - 3 \\ C_L &= -\frac{\alpha_L}{2F_s} - \frac{\beta_L}{4F_s^2} + 3\frac{\gamma_L}{8F_s^3} + 3 \\ D_L &= \frac{\alpha_L}{2F_s} - \frac{\beta_L}{4F_s^2} + \frac{\gamma_L}{8F_s^3} - 1, \end{aligned}$$

$$k_{1_L} = \frac{\alpha_L}{2F_s},\ k_{2_L} = \frac{\beta_L}{4F_s^2},\ k_{3_L} = \frac{\gamma_L}{8F_s^3},$$

$$\alpha_L = a_L + b_L + c_L,\ \beta_L = a_L b_L + b_L c_L + c_L a_L,\ \gamma_L = a_L b_L c_L.$$

Therefore, discrete time equation of the filter with input u and output y is given by:

$$y(k) = -\sum_{m=1}^{3} p_{m_L} y(k-m) + p_{4_L} \sum_{n=0}^{3} e_n \cdot u(k-n), \tag{2.7}$$

with $\mathbf{e} = \begin{bmatrix} 1 & 3 & 3 & 1 \end{bmatrix}^T$; e_n represents n^{th} element of $\mathbf{e}$.

The transfer function of an integrator is given by:

$$H_I(s) = \frac{1}{s} \tag{2.8}$$

By making use of the transformation in Equation. 2.1, following discrete time model can be obtained:

$$H_I(z) = \frac{1+z^{-1}}{2F_s(1-z^{-1})}$$

If u and y are input and output of the discrete time integrator at time instant k, then following can be written

$$y(k) = y(k-1) + \frac{1}{2F_S}\left[u(k) + u(k-1)\right] \tag{2.9}$$

2.2.2 Implementation procedure of the MCA

In this section, we make use of the mathematics presented in section 2.2.1 to formulate a procedure to implement MCA in RT. The input to the system is $\mathbb{I} = \{{}^b\mathbf{a}_b, {}^b\,\omega_b\}$ with output $\mathbb{O} = \{\mathbf{x}_p, {}^w\dot{\mathbf{x}}_p, \theta, \dot{\theta}\}$, where ${}^b\mathbf{a}_b, {}^b\,\omega_b \in \mathbb{R}^3$ denote linear acceleration and angular velocity of the moving body (b) expressed in the frame of body itself, whereas ${}^w\dot{\mathbf{x}}_p$ denotes the reference linear velocity of the platform, p, (of motion simulator) expressed in the world frame (fixed); $\dot{\theta}$ denotes gimbal angular velocity. Note that we consider an architecture where the input is linear acceleration of the actual motion and hence the addition of gravity can be ignored in fig. 1.6. The following denote user defined parameters: $\mathbb{P} = \{\mathbb{K}, \mathbb{S}, \mathbb{F}\}$, where $\mathbb{K}, \mathbb{S}, \mathbb{F}$ denote scaling, limiting/saturation and filtering parameters defined below.

$$\mathbb{K} = \{k_x, k_y, k_z, k_{\omega_x}, k_{\omega_y}, k_{\omega_z}\} \tag{2.10}$$

$$\mathbb{S} = \{\mathbf{a}^i_{\max}, \omega^i_{\max}, \mathbf{a}^i_{\min}, \omega^i_{\min}, \mathbf{x}^o_{p_{\max}}, \mathbf{x}^o_{p_{\min}}, {\theta_{\max}}^o, {\theta_{\min}}^o\} \tag{2.11}$$

$$\mathbb{F} = \{\mathbf{a}_r, \mathbf{b}_r, \mathbf{a}_h, \mathbf{b}_h, \mathbf{c}_h, \mathbf{a}_L, \mathbf{b}_L, \mathbf{c}_L\} \tag{2.12}$$

All the elements of $\mathbb{K} \in \mathbb{R}^+$ scale down the magnitude of input motion signal. The parameters k_x, k_y, k_z scale down the input linear acceleration along x, y, z respectively whereas $k_{\omega_x}, k_{\omega_y}, k_{\omega_z}$ scale down angular velocity along x, y, z. The elements of $\mathbb{S} \in \mathbb{R}^3$ denote the maximum and minimum limit of input motion signals (denoted by i) and the limits of workspace of the manipulator (denoted by o). The parameters $\mathbf{a}^i_{\max}, \omega^i_{\max}$ denote maximum input linear acceleration and angular velocity respectively; $\mathbf{a}^i_{\min}, \omega,^i_{\min}$ denote minimum input linear acceleration and angular velocity respectively; $\mathbf{x}^o_{p_{\max}}, \mathbf{x}^o_{p_{\min}}$ denote position limits of workspace and ${\theta_{\max}}^o, {\theta_{\min}}^o$ denote orientation limits of workspace. The elements of $\mathbb{F} \in \mathbb{R}^3$ denote CT cut-off frequencies associated with the high pass filter and low pass filter along translational, rotational and tilt-coordination channels. $\mathbf{a}_r = [a_{r_x} a_{r_y} a_{r_z}]^T, \mathbf{b}_r = [b_{r_x} b_{r_y} b_{r_z}]^T$ denote the high pass filter parameters along x, y, z for rotational channel, see Equation. 1.8.

$\mathbf{a}_h = [a_{h_x} a_{h_y} a_{h_z}]^T, \mathbf{b}_h = [b_{h_x} b_{h_y} b_{h_z}]^T, \mathbf{c}_h = [c_{h_x} c_{h_y} c_{h_z}]^T$ denote high pass filter parameters along x, y, z for translational channel, see Equation. 1.7. $\mathbf{a}_L = [a_{L_x} a_{L_y} a_{L_z}]^T, \mathbf{b}_L = [b_{L_x} b_{L_y} b_{L_z}]^T, \mathbf{c}_L = [c_{L_x} c_{L_y} c_{L_z}]^T$ denote low pass filter parameters for tilt-coordination channel along x, y, z, see Equation. 1.9.

After declaring all the parameters $\mathbb{P}$, MCA can be implemented in RT using the procedure described below:

Offline computation of discrete time filter parameters

The transformation of continuous time filter parameters to discrete time can be carried out before simulating a scenario. In order to do so, the methodology described in section 2.2.1 is used. Specifically, each discrete time filter is characterized by the following parameters.

$$H_{\omega_x} : [p1_{rx} p2_{rx} p3_{rx}], H_{\omega_y} : [p1_{ry} p2_{ry} p3_{ry}], H_{\omega_z} : [p1_{rz} p2_{rz} p3_{rz}]$$

$$H_{f_x} : [p1_{h_x}\ p2_{h_x}\ p3_{h_x}\ p4_{h_x}],\ L_{f_x} : [p1_{L_x}\ p2_{L_x}\ p3_{L_x}\ p4_{L_x}]$$

$$H_{f_y} : [p1_{h_y}\ p2_{h_y}\ p3_{h_y}\ p4_{h_y}],\ L_{f_y} : [p1_{L_y}\ p2_{L_y}\ p3_{L_y}\ p4_{L_y}]$$

$$H_{f_z} : [p1_{h_z}\ p2_{h_y}\ p3_{h_z}\ p4_{h_z}],\ L_{f_z} : [p1_{L_z}\ p2_{L_z}\ p3_{L_z}\ p4_{L_z}]$$

Note that $H_{\omega_x}, H_{\omega_y}, H_{\omega_z}$ denote high pass filter along x, y, z respectively for rotational channel (see Equation. 2.4); $H_{f_x}, H_{f_y}, H_{f_z}$ denote high pass filter along x, y, z respectively for translational channel (see Equation. 2.2); $L_{f_x}, L_{f_y}, L_{f_z}$ denote low pass filter along x, y, z respectively for tilt-coordination channel (see Equation. 2.6)

Declaration of limitation/saturation functions

In this section, we describe the saturation functions used by us to limit:

1. the acquired input signal from the game
2. to prevent the manipulator from going out of it's workspace

We make use of hard saturation to limit the input data; concretely speaking for an input u at time instant k, the output y is given by:

$$y(k) = u(k) \quad \text{if } u_{min} \leq u(k) \leq u_{max} \tag{2.13}$$

$$y(k) = u_{max} \qquad \text{if } u(k) > u_{max} \tag{2.14}$$

$$y(k) = u_{min} \qquad \text{if } u(k) < u_{min}, \tag{2.15}$$

where $u_{max}, u_{min} \in \mathbb{R}$ denote the maximum and minimum allowed input signal.

We make use of sigmoid function to limit the maximum and minimum velocity of the manipulator based on it's present position and orientation; concretely speaking, prospective linear and angular velocity (${}^w\dot{x}_p, \dot{\theta}$) of the simulator at a given position and orientation (x_p, θ) is saturated in between the following limits:

$$ {}^w\dot{x}_{p_{max}} = \frac{V}{1+\exp(\alpha x_p)} \tag{2.16} $$

$$ {}^w\dot{x}_{p_{min}} = \frac{V}{1+\exp(\alpha x_p)} - V, \tag{2.17} $$

where $V, \alpha \in R^+$ are saturation parameters and ${}^w\dot{x}_{p_{max}}, {}^w\dot{x}_{p_{min}}$ are the maximum/minimum permissible speed of the manipulator respectively. The velocities are saturated in this band, which is decided by the position and orientation of the manipulator. Saturation of velocity (based on position of the platform) is guided by the following equation:

$$ {}^w\dot{x}_{s_p} = {}^w\dot{x}_p \quad if \;\; {}^w\dot{x}_{p_{min}} \leq {}^w\dot{x}_p \leq {}^w\dot{x}_{p_{max}} \tag{2.18} $$

$$ {}^w\dot{x}_{s_p} = {}^w\dot{x}_{p_{max}} \qquad if \;\; {}^w\dot{x}_p > {}^w\dot{x}_{p_{max}} \tag{2.19} $$

$$ {}^w\dot{x}_{s_p} = {}^w\dot{x}_{p_{min}} \qquad if \;\; {}^w\dot{x}_p < {}^w\dot{x}_{p_{min}}, \tag{2.20} $$

where ${}^w\dot{x}_{s_p}$ is the saturated velocity. Note that in case of sigmoid saturation, both ${}^w\dot{x}_{p_{max}}, {}^w\dot{x}_{p_{min}}$ are functions of position, whereas when hard saturation is employed:

$$ {}^w\dot{x}_{s_p} = {}^w\dot{x}_p \quad if \;\; x_{p_{min}} \leq x_p \leq x_{p_{max}} \tag{2.21} $$

$$ {}^w\dot{x}_{s_p} = 0 \qquad \text{otherwise}, \tag{2.22} $$

with $x_{p_{max}}, x_{p_{min}}$ being the limits of workspace. Note that any reasonable function can be used to prevent the simulator from exiting it's workspace.

Real time computations

For the discrete time variable $k = 1, 2, 3, \cdots$, computations must be carried out on-line as per the pseudo code presented below.

```
Begin
    InitializeThePositionAndOrientationOfPlatform
        $\theta_1^i = 0;$        $x_i = a;$
        $\theta_2^i = 0;$        $y_i = b;$
        $\theta_3^i = 0$;    $z_i = c$;
    **DataAcquisition**
    $a_x(k) = acquire\_acceleration\_data(x)$
    $a_y(k) = acquire\_acceleration\_data(y)$
    $a_z(k) = acquire\_acceleration\_data(z)$
    $w_x(k) = acquire\_angular\_velocity\_data(x)$
    $w_y(k) = acquire\_angular\_velocity\_data(y)$
    $w_z(k) = acquire\_angular\_velocity\_data(z)$
    **Scaling**
        $a_{sx} = scale(a_x(k), k_x)$
        $a_{sy} = scale(a_y(k), k_y)$
        $a_{sz} = scale(a_z(k), k_z)$
        $w_{sx} = scale(w_x(k), k_{\omega_x})$
        $w_{sy} = scale(w_y(k), k_{\omega_y})$
        $w_{sz} = scale(w_z(k), k_{\omega_z})$
    **LowPassFiltering**
    $f_{Lx} = LPF(a_{sx}(k))$
    $f_{Ly} = LPF(a_{sy}(k))$
    **TiltCoordination**
        $[\theta_{1_{tilt}}, \theta_{2_{tilt}}, \theta_{3_{tilt}}](k) = find\_tilt(f_{Lx}, f_{Ly})$
    **RotationalChannel**
    **if** k==1 **then**
        $\mathbf{T}(k) = compute\_T(\theta_1^i, \theta_2^i, \theta_3^i)$
    **else**
        $\mathbf{T}(k) = compute\_T(\theta_1(k-1), \theta_2(k-1), \theta_3(k-1))$
    **end if**
    $[\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3](k) = \mathbf{T}(k) \cdot [w_{sx}, w_{sy}, w_{sz}]$
    $\dot{\theta}_{1_h}(k) = HPF\_R(\dot{\theta}_1(k))$
    $\dot{\theta}_{2_h}(k) = HPF\_R(\dot{\theta}_2(k))$
    $\dot{\theta}_{3_h}(k) = HPF\_R(\dot{\theta}_3(k))$
    **SigmoidSaturate**
        **if** k==1 **then**
            $[\dot{\theta}_{1_h}, \dot{\theta}_{2_h}, \dot{\theta}_{3_h}](k) = sigmoid(\dot{\theta}_{1_h}(k), \dot{\theta}_{2_h}(k), \dot{\theta}_{3_h}(k), \theta_1^i, \theta_2^i, \theta_3^i))$
        **else**
```

```
            $[\dot{\theta}_{1_h}, \dot{\theta}_{2_h}, \dot{\theta}_{3_h}](k) = sigmoid(\dot{\theta}_{1_h}(k), \dot{\theta}_{2_h}(k), \dot{\theta}_{3_h}(k), \theta_1(k-1), \theta_2(k-1), \theta_3(k-1)))$
        **end if**
        $[\theta_{1_{rot}}, \theta_{2_{rot}}, \theta_{3_{rot}}](k) = Integrator(\dot{\theta}_{1_h}(k), \dot{\theta}_{2_h}(k), \dot{\theta}_{3_h}(k))$
    **HardSaturationOfOrientationAndAngularVelocity**
    **TranslationChannel**
        **if** k==1 **then**
            $\mathbf{R}(k) = compute\_R(\theta_1^i, \theta_2^i, \theta_3^i)$
        **else**
            $\mathbf{R}(k) = compute\_R(\theta_1(k-1), \theta_2(k-1), \theta_3(k-1))$
        **end if**
        $[acc_x, acc_y, acc_z](k) = \mathbf{R}(k) \cdot [a_{sx}(k), a_{sy}(k), a_{sz}(k)]$
        $a_{h_x}(k) = HPF\_T(acc_x(k))$
        $a_{h_y}(k) = HPF\_T(acc_y(k))$
        $a_{h_z}(k) = HPF\_T(acc_z(k))$
        $[v_x, v_y, v_z](k) = Integrator(a_{h_x}(k), a_{h_y}(k), a_{h_z}(k))$
    **SigmoidSaturateTwo**
    **if** k==1 **then**
        $[v_x, v_y, v_z](k) = sigmoid(v_x(k), v_y(k), v_z(k), x_i, y_i, z_i)$
    **else**
        $[v_x, v_y, v_z](k) = sigmoid(v_x(k), v_y(k), v_z(k), x(k-1), y(k-1), z(k-1))$
    **end if**
    $[x, y, z](k) = Integrator(v_x(k), v_y(k), v_z(k))$
    **HardSaturationOfPositionAndVelocity**
    **OrientationOfThePlatform**
    $\theta_1(k) = \theta_{1_{rot}}(k) + \theta_{1_{tilt}}(k)$
    $\theta_2(k) = \theta_{2_{rot}}(k) + \theta_{2_{tilt}}(k)$
    $\theta_3(k) = \theta_{3_{rot}}(k) + \theta_{3_{tilt}}(k)$
    $send\_reference(x, y, z, v_x, v_y, v_z, \theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3)$
**End**
```

Explanation of routines

In this section, we relate the subroutines used in the pseudo code above, to the mathematics presented in section 2.2.1. *acquire_acceleration_data* and *acquire_angular_velocity_data* signify the process of reading data from a gaming engine at a specified frequency. *scale* is used to scale down the magnitude of incoming motion signal by a constant factor with K being the reciprocal of scaling factor. LPF is used to implement the recursive equation described in Equation. 2.7. $find\_tilt$ carries out the operations described in Equation. 1.13 with $\theta_{3_{tilt}} = 0$. $compute\_T$ computes the matrix described in Equation. 1.11. HPF_R implements the recursive equation described in Equation. 2.5. *sigmoid* routine is a sigmoid saturation described in section 2.2.2. The routine $Integrator$ is described by recursive equation, see Equation. 2.9. $Hard\_saturation$ is described in section 2.2.2. $compute\_R$ computes the matrix described in Equation. 1.10. Finally, $send\_reference$ sends the reference signal to the manipulator.

2.3 Robotic platform manipulation

SP7 is a parallel robotic manipulator with 7 DOF. The design of the mechanism is similar to the 6RSS-R parallel manipulator commonly known as the “Stewart platform” [41] with an additional yaw axis actuator making it a redundant axis. Fig. 2.8 shows the SP7 platform attached with flight simulator equipment. The mechanical design, kinematics, control strategies, workspace analysis, actuator selection process, etc. have been developed at the PMAR laboratory team. There are several works of literature on the various development aspects of this type of manipulator. For instance, [42] provides an implementation procedure for the inverse kinematics of the manipulator, [43] describes the actuator control and parameter tuning strategies, etc. Thus, the specifics of the SP7 platform are not added in this research. One simple yet essential constraint made on the SP7 is the identification of the prescribed workspace boundaries.

2.3.1 Prescribed workspace

For the user to have a smooth motion perception experience, the workspace of the platform has to be continuous at each configuration. The prescribed workspace has been identified inside the workspace envelope of the manipulator in order to have a continuous workspace The prescribed workspace is the set of poses that the robot end-effector can reach in such a way that any pose can be traversed without passing through singularities [44]. Even though this severely restricts the boundary of the platform, it is an essential step for a motion simulator platform. Fig. 2.9 shows the workspace envelope of the SP7 generated in the Maple application [45]. Table 2.1 lists the boundary limits of SP7 at each axis with the whole workspace and the reduced workspace capability.

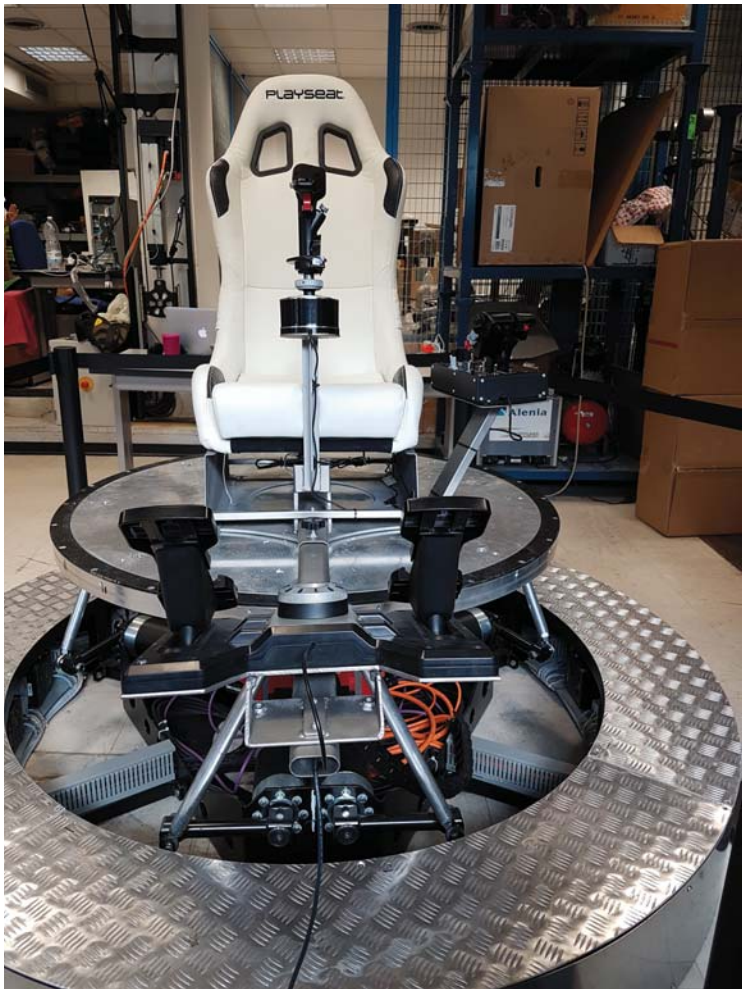
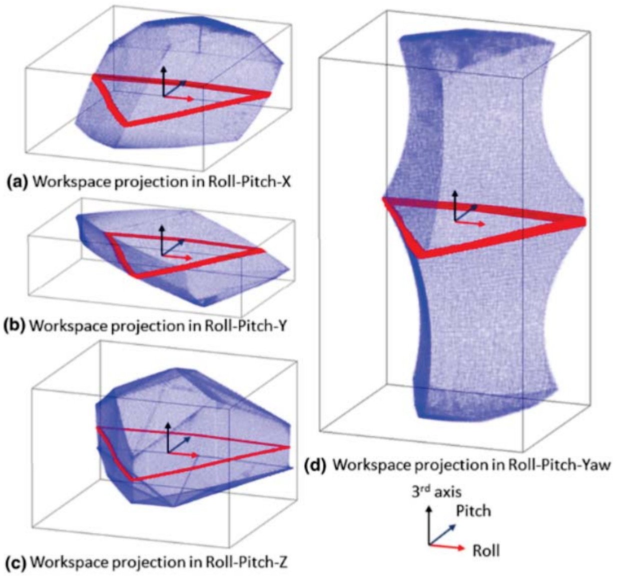


Figure 2.8 SP7 motion simulator with flight simulator gears.

Table 2.1 SP7 boundary limits: entire envelope and prescribed workspace limit.

DOF	**Entire workspace**	**Bounded workspace**
Roll angle	±15°	±5°
Pitch angle	+20/ − 13°	±7°
Yaw angle	±180°	±180°
Surge translation	+200/ − 120 mm	±50 mm
Sway translation	±120 mm	±55 mm
Heave translation	+150/ + 500 mm	+396 mm/ + 451 mm



(a) Workspace projection in Roll-Pitch-X

(b) Workspace projection in Roll-Pitch-Y

(c) Workspace projection in Roll-Pitch-Z

(d) Workspace projection in Roll-Pitch-Yaw

Figure 2.9 Workspace envelope projections of SP7 generated in Maple.

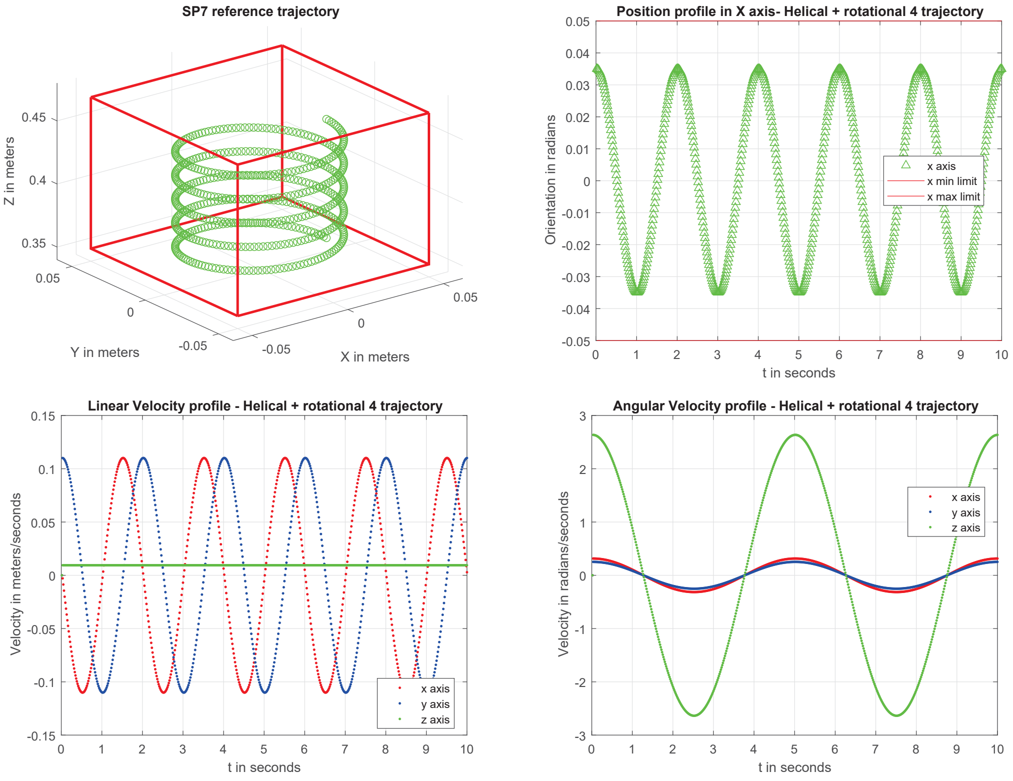


Figure 2.10 Some reference trajectories used to validate SP7 tracking performance.

2.3.2 Validation of trajectories tracking of the robotic manipulator

A simple one shot test was conducted using synthetically generated trajectories to verify the tracking performance of the SP7 manipulator. The platform's commanded trajectory pose and actual trajectory pose (recorded using the HTC Vive trackers) were compared and it was found that the average precision was less than 10 mm and, in most high-speed trajectories, the average accuracy was less than 30 mm. This high performance is consistent with the fact that SP7 is a parallel manipulator. Fig. 2.10 displays some of the reference trajectories used in this validation process.

2.4 Modular integration of all three sub-systems

The computer running the VR application created from the first subsystem sends visual and audio signals to the HTC Vive device connected to the same computer. The same application simultaneously sends motion data namely linear acceleration and angular velocity to the MCA controller in the second sub-system. MCA controller is a microcontroller with Linux operating system and contains the MCA algorithm is implemented in the C++ language. The motion signal, namely pose and velocity is transmitted from the MCA controller to the SP7 controller (third sub-system). The controller of the SP7 receives the Cartesian pose and the velocity of the end-effector, which is then processed into the joint angles with the inverse kinematics implemented in C++. To control the actuators, the joint angles are then processed via the PID controller and PWM. Via the Ethernet and UDP socket connections, all the communications between the three networks are linked to an Ethernet switch that manages data transmission efficiently. Fig. 2.11 shows the MCA controller along with the Ethernet switch. Fig. 2.12 displays the disassembled SP7 platform with the actuators and controllers in view. Fig. 2.13 shows the snippet of the application developed to control the SP7 platform with MCA controller as well as independently through multiple options like joystick, path file, and predefined motions.

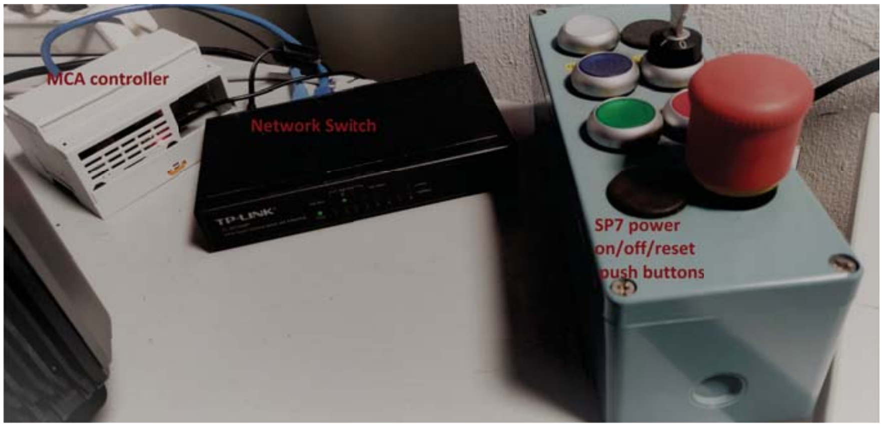


Figure 2.11 MCA controller connected to the Ethernet switch.

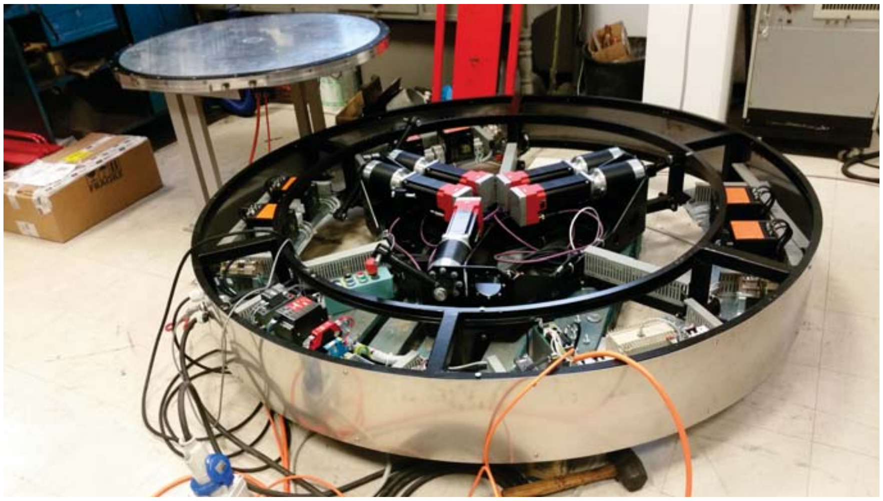
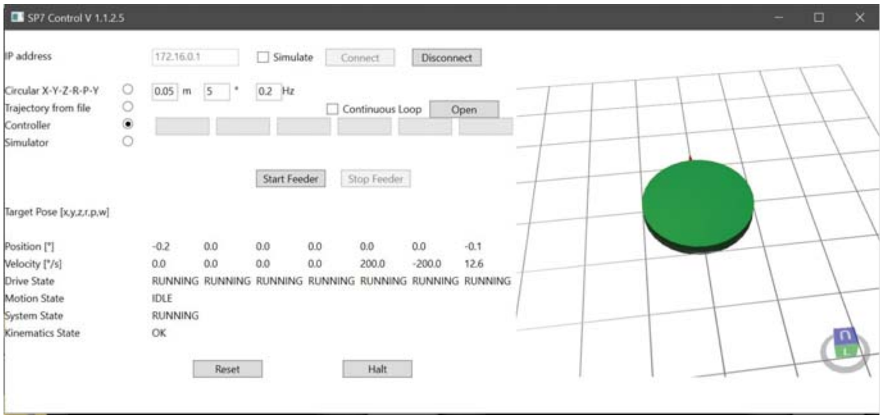


Figure 2.12 Disassembled SP7 platform with the actuators and controllers in view.



IP address 172.16.0.1 □ Simulate

○ Circular X-Y-Z-R-P-Y 0.05 m 5 ° 0.2 Hz
○ Trajectory from file □ Continuous Loop
● Controller
○ Simulator

Target Pose [x,y,z,r,p,w]

Position [°]	-0.2	0.0	0.0	0.0	0.0	0.0	-0.1
Velocity [°/s]	0.0	0.0	0.0	0.0	200.0	-200.0	12.6
Drive State	RUNNING	RUNNING	RUNNING	RUNNING	RUNNING	RUNNING	RUNNING
Motion State	IDLE						
System State	RUNNING						
Kinematics State	OK						

Figure 2.13 A screenshot of the SP7 controller application.

Chapter 3

Experimental evaluation of the quality of motion perception in SP7

This chapter will focus on the experimental evaluation of the integrated system "SP7" presented in the chapter 2. SP7 is a 7 DOF manipulator, so most VR games and experiences could be integrated with it. For the purpose of testing and to facilitate comparison with the literature, it was agreed by all stakeholders that SP7 will be first incorporated as a VR flight simulator. The technique used to achieve this integration is described in section 2.1.2. The chapter is structured as follows: First, the overview of the SP7 flight simulator implementation and testing. Followed by the assessment of motion perception both subjective and objective methods, and finally, the problems identified with the current setup are mentioned.

3.1 SP7 as flight simulator using X-plane simulator

Fig. 3.1 shows the setup of SP7 converted as a full-fledged flight simulator with gears such as seat (PLAYSEAT evolution white), joystick (HOTAS TM Warthog throttle), throttle (VPC T-50 stick), and rudder pedal (MFG CROSSWIND V2). Metadata extraction plugin was used to extract the motion data and send it to the MCA controller, which in turn sends the Cartesian pose and velocity of the end-effector to the SP7 controller.

After assembling all the components and checking the safety features, it was time to test the motion perception quality of a user on the SP7 flight simulator. The literature on the testing of a motion simulator for the quality of the motion perceived by the subject on the platform are presented in [46], [47], and more. In all the literature, the quality of the motion simulator is measured in terms of the intensity of motion sickness the subject experiences. Even though this form of evaluation does not explicitly consider a particular module of the

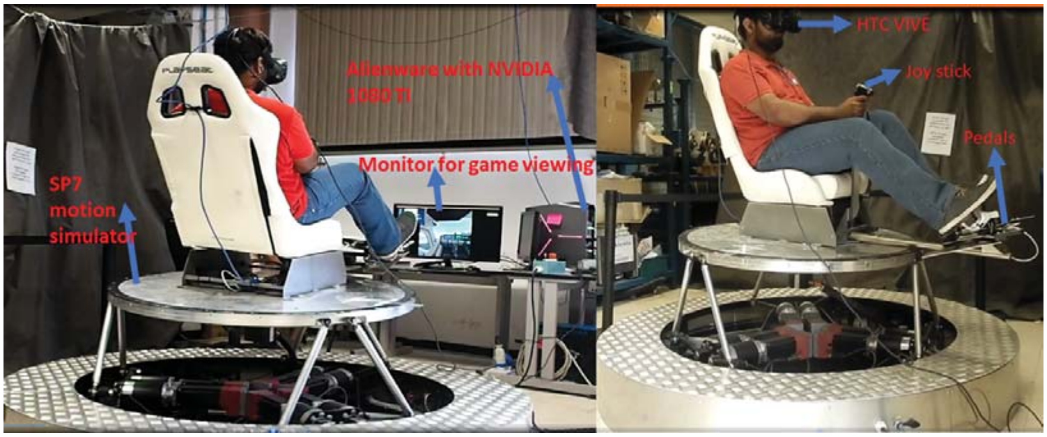


Figure 3.1 SP7 flight simulator set up with the user on the platform.

motion simulator, it provides a fair assessment of the motion simulator as a whole. But as the motion cueing algorithm is responsible for the motion generation, this can be considered as the evaluation of the MCA assuming the raw motion from the game/physics engine is error-free. As the end user's quality of experience relies on this sickness quotient, we adopt this type of approach for the subjective evaluation.

To conduct a subjective evaluation, the subjects must experience the same virtual simulation with the same visual, motion, and audio cues so that the questionnaire filled by them can be fairly compared and assessed. As of now, the X-Plane sim integration is for an active experience i.e., the user can fly the aircraft of his choice by himself. It is not feasible and convenient from the experimentation point of view to ask the user to fly through a trajectory as it will engage the user from a gaming perspective and might lead the subject not to experience the motion provided. Thus, a workaround in the current architecture was developed to use a recorded session of flight in X-plane. The workaround architecture is shown in Fig. 3.2 and it is quite straight-forward, an expert pilot was invited to record few scenarios of flight without the motion simulator as per the design of experiments. The recorded session's motion data is extracted and saved as well as the replay of the visuals. In the SP7 control API, a module is added to send the recorded motion data to the platform when the replay is triggered. This triggering and synchronization are achieved by a key binding in the joystick, when the button is pressed the replay starts playing in the HMD and the MCA controller starts sending motion data to the SP7. During this experience, the test subject will be passive and experience the entire session instead of engaging in piloting the aircraft.

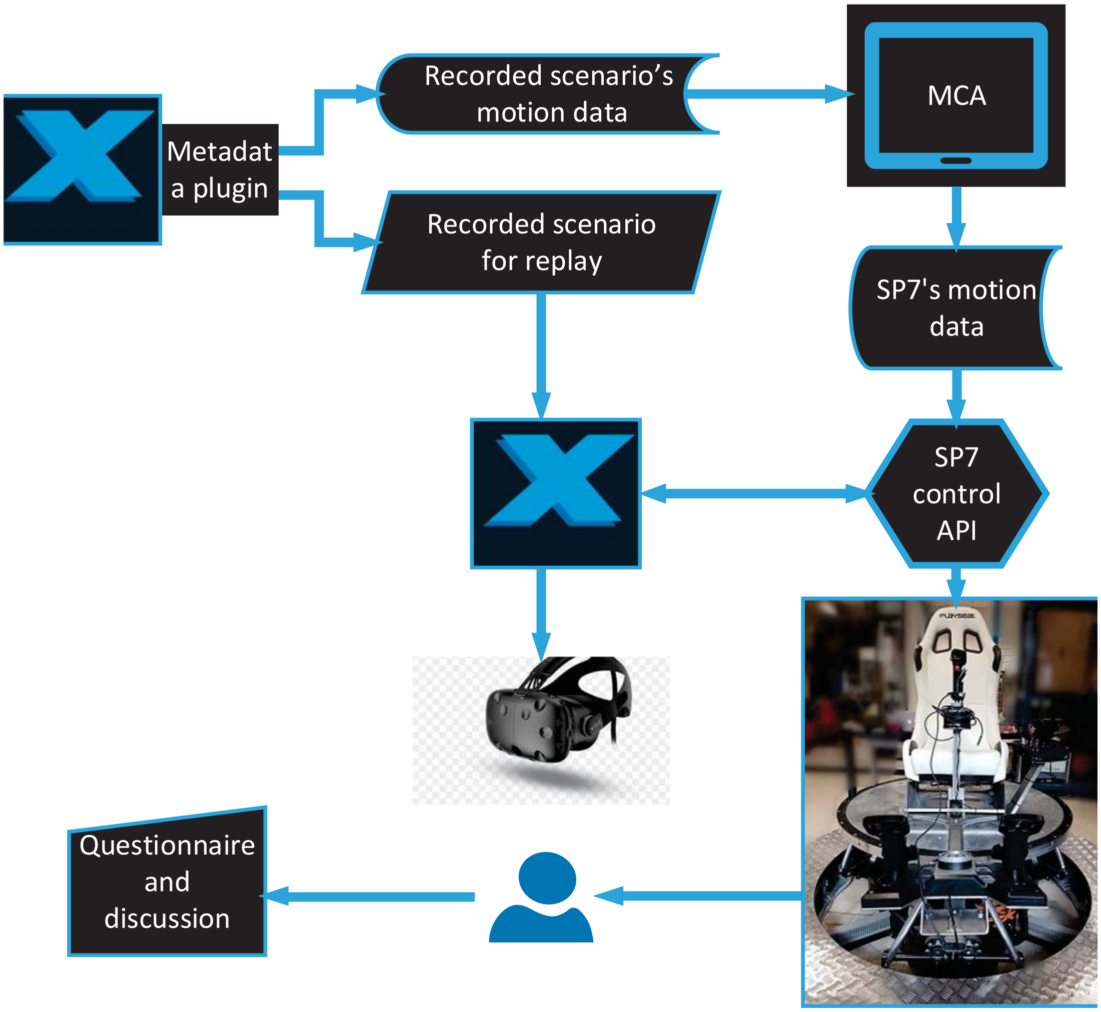


Figure 3.2 An overview of the workaround architecture of SP7 flight simulator setup to facilitate experimental evaluation with a recorded session.

The next section will present the literature on the subjective evaluation of motion perception in a flight simulator, expand on the protocol and questionnaire created for the SP7 assessment, and finally summarize the findings of the subjective assessment.

3.2 Subjective assessment of motion perception in SP7

The concept of human motion perception with regards to the motion simulator has been introduced in detail in the chapter 1.2. As mentioned earlier, there is numerous literature on the subjective evaluation of motion sickness like [46], [47], [48], [49], [50], [51], etc. but the current widely used literature in this approach is [46], the questionnaire presented in this study allows to evaluate the simulator sickness, this is commonly referred to as the simulation sickness questionnaire (SSQ). SSQ is an improvement of MSQ presented in [47] which is the standard questionnaire developed to evaluate motion sickness in aviation training. SSQ was formulated based on the weightage to the hierarchical factors of 28 symptoms of sickness like general discomfort, fatigue, headache, eyestrain, dizziness, vertigo, stomach awareness, etc. Multiple simulator studies like [52], [53], etc. have used SSQ in their evaluation of the motion cueing algorithm for their motion simulator. SSQ provides a one index evaluation of the sickness, a method has been proposed in [54] to decode and analyze the data of SSQ in an elaborate manner covering the entire cybersickness factors.

Apart from just the quality of the motion perceived, there are few more factors to be considered like the sense of presence in VR environments [55], duration of immersion with comfort [56], and other human factors mentioned [57]. These factors were not prioritized in the literature as most of them does not use VR device as the visual input. In our case, due to the level of immersion encountered by the subject, these considerations also play a major role. Thus, considering all these factors an experimental procedure with a modified SSQ questionnaire was developed in collaboration with Prof. Fabrizio Bracco of the Department of Education Science, University of Genoa.

3.2.1 Procedure for subjective evaluation of SP7

First, the subject is requested to fill out the pre-test questionnaire. The subject is asked to sit on the motion simulator platform seat and wear the HMD. The preference order of stimuli shall be selected by the subject at random. Then a short training session with visual to HMD and motion to the platform is given to let the subject get accustomed to the experience. Based on the order chosen by the subject, four sets of scenarios are given one after the other. One set consists of two cases, the visual stimuli remain the same for both cases, while the motion stimuli are provided only in the latter case. At the completion of each set, the participant will be asked to fill out a questionnaire to assess the experience. After 4 sets, the user is asked to fill another questionnaire for evaluating simulator sickness. The overall experimentation duration will be between 20 to 30 minutes depending on how fast the subject is responsive to the questionnaire. The procedure is organized as follows:

1. Explanation of the experiment
2. Agreement signing
3. Pre-test questionnaire filling
4. Choice (order of session) picking
5. Training session
6. If more training session requested, repeat 5 or proceed forward (Maximum 3 training sessions)
7. Session 1
    a. No motion
    b. With motion
    c. Mid test questionnaire
    d. Pause if needed (maximum 3 minutes)
8. Session 2
    a. No motion
    b. With motion
    c. Mid test questionnaire
    d. Pause if needed (maximum 3 minutes)
9. Session 3
    a. No motion
    b. With motion
    c. Mid test questionnaire
    d. Pause if needed (maximum 3 minutes)

10. Session 4 (Roll and Yaw combined)

    a. No motion

    b. With motion

    c. Mid test questionnaire

    d. Pause if needed (maximum 3 minutes)

11. Post-test questionnaire filling (Simulation sickness questionnaire + comments)

3.2.2 Questionnaire for subjective evaluation of SP7

As explained in the previous section, there are three questionnaires namely pre-test, mid-test, and post-test. The questionnaires are as follows:

Pre-test questionnaire

1. Have you had any previous virtual reality experiences?

   Yes

   No

2. Earlier experience with motion simulators?

   Yes

   No

3. Do you have any medical conditions?

   Yes

   No

   If yes, kindly specify below:

4. When have you eaten or had a drink last time? Kindly specify the approximate time.

5. How do you feel physically?
   On the scale of 0 to 100 (0 ⟹ worst condition ; 100 ⟹ Excellent condition)

6. How do you feel mentally?
   On the scale of 0 to 100 (0 ⟹ timid ; 100 ⟹ enthusiastic)

Mid-test questionnaire

1. How well you can distinguish case 1 (no motion) and case 2 (with motion)?
   On the scale of 0 to 100:
   0 ⟹ cannot feel any difference;
   100 ⟹ clearly distinct experiences

2. In case 1, how realistic was the experience?
   On the scale of 0 to 100:
   0 ⟹ completely fake;
   100 ⟹ completely real

3. In case 2, how realistic was the experience?
   On the scale of 0 to 100:
   0 ⟹ completely fake;
   100 ⟹ completely real

4. In case 1, how well you were engaged during the simulation?
   On the scale of 0 to 100:
   0 ⟹ So boring;
   100 ⟹ fully immersed

5. In case 2, how well you were engaged during the simulation?
   On the scale of 0 to 100:
   0 ⟹ So boring;
   100 ⟹ fully immersed

6. In case 1, how was the motion fidelity with respect to the visual display?
   On the scale of 0 to 100:
   0 ⟹ motion and visuals did not match at all;
   100 ⟹ motion was completely coherent with the visuals

7. In case 2, how was the motion fidelity with respect to the visual display?
   On the scale of 0 to 100:
   0 ⟹ motion and visuals did not match at all;
   100 ⟹ motion was completely coherent with the visuals

8. Rate the overall experience?
   On the scale of 0 to 100:
   0 ⟹ very poor;
   100 ⟹ Excellent
   Case 1:
   Case 2:

Post-test questionnaire (SSQ)

1. Did you feel nausea?
   Yes/No

2. Did you feel dizzy?
   Yes/No

3. Did you feel eye strain?
   Yes/No

4. Did you have any other eye trouble?
   Yes/No

   Did you have headache?
   Yes/No

5. Did you feel mental pressure?
   Yes/No

6. Were you tired?
   Yes/No

7. Did you feel anxiety (uneasiness)?
   Yes/No

8. Did you fear?
   Yes/No
   If yes, kindly specify a little on which point of simulation you felt fear.

9. Were you bored?
   Yes/No
   If yes, kindly specify a little on which point of simulation you felt boring.

If you have any comments about your overall experience, kindly mention below

3.2.3 Jury selection

A total of 18 subjects participated in the experiments lasting over 540 minutes. The jury was chosen on the basis of three major criteria, namely:

1. *Age distribution:* out of 18, 8 were in the range of 20-30 years, 7 in the range of 30-50 years, and 3 were in the range of more than 50 years.

2. *Experience in piloting aircraft at any capacity:* 1 licensed hobby pilot and 2 experienced flight simulator gamer.

3. *Previous exposure to simulator experience:* 5 jury participants have experience in a simulator of some kind, such as an arena or museum. 4 jury members had some prior experience in VR.

3.2.4 Results of subjective evaluation of SP7

A sample scanned copy of the filled questionnaire of one of the jury could be found in Appendix B. The scores of each jury were tallied to a total of 100 with a simple weighted average of each question. The weightage was allocated as per the SSQ model in [46]. The higher the score, the more comfortable, realistic, and immersive the simulator experience for a jury. Out of 18 juries, 7 scored 80+ (including 5 juries with prior simulator experience), 3 scored between 70 and 80 (the pilot is among them), 5 scored between 60 and 70, 1 experienced flight simulator player scored 57, and 2 scored less than 50. (1 gamer is in this category). The overall 61% (70+ score) of the jury was satisfied with the quality of the SP7 flight simulator experience. This kind of evaluation provides a single score on the performance, so it is quite vague on the cause of the low or high score. Based on literature and the discussions with the juries, the lower scores could be explained by the age factor as all three of the juries over 50 were not satisfied, the vestibular system functioning deteriorates as we age. But the 2 professional gamers in the 30-50 age group also gave scores less than 70, this could be explained by the phenomenon mentioned in [58] that the human brain may “quarantine” visual stimuli that were hard to experience in the real world (low-level realism), thus leading to less motion sickness. On the other hand, the high score given by the juries with not much flight simulation experience could be explained by the same phenomena. Therefore, in this subjective evaluation, we can only infer that the SP7 flight simulator had a 61% satisfactory user experience.

3.3 Identification of issues with the existing configuration

To analyze the cause of the unsatisfactory experience, the SP7 was further tested rigorously, focusing on each module of the system. There was almost no literature on the systematic testing of the whole system, thus each module of SP7 was tested for its functionality, like unit testing but focusing on the motion quality. After almost two months of intensive testing with feedback from friendly subjects, the following issues were identified

1. The first and obvious error found was in the motor control. The 7th axis (redundant axis) motor responsible for the yaw motion had an overshooting issue. The target pose was not reached precisely, the overshoot was in 3 degrees to 5 degrees range based on the initial acceleration of the motion. The fine-tuning of the PID controller did not resolve the issue, thus it was sent to the manufacturer who later found the issue to be in the shielding and returned the fixed motor.

2. During high frequency and high-intensity motions, the MCA washout filter's transient state ends abruptly, and the next input signal begins, leading the platform to halt suddenly with a heavy jerk and pulls the platform to return to the center for the next motion input. This is commonly referred to as the "turn around bump". The following two solutions were explored.

   a. The first solution implemented was the brute force solution in which a sigmoidal saturation function (ref. Equation. 2.13) was introduced at the output of the MCA controller i.e., the command pose and velocity. This method will ensure that the high-velocity data are limited while the low velocity performs as expected.

   b. The second solution is to add the sliding mode controller [59] to the output of the MCA controller. When the pose reaches any boundary of the SP7, the sliding mode controller is activated. Instead of abruptly stopping the motion and returning to the center, this sliding mode control will slide along the boundary of the platform and return to the center of the next motion signal. Through a series of experiments, the sliding action was found to be a distracting motion by few, and while few preferred the sliding motion to no motion at all.

3. Virtual reality games like X-Plane are designed for the user to be on a fixed base rather than the moving base, thus the motion of the simulator platform is not taken into account when the motion data is exported, i.e., the frame in which the motion data is sent, is the virtual world frame. MCA converted the virtual world frame to the platform base frame. This conversion is only applied to the motion cue and not to the visual cue, this causes a mismatch in visual and motion information. This mismatch of motion results in an increase of motion sickness in the user and destroys the immersion factor. This motion must be reflected in the visual data also and it is done by the motion

compensation algorithm. One such source code algorithm is available in [60], a fast implementation and testing of this algorithm has resolved the problem in less intensive motion, but during high-frequency motion, there was a shift in the user's view of the HMD, as if the user is suddenly out of the aircraft. When this happens, there was a complete loss of immersion. The solution to this issue of motion compensation using the Vive lighthouse tracking system is presented in chapter 4.

4. While conducting demonstration and discussion sessions with stakeholders, the parameter of the motion cueing was modified for each person as per their requirements. For instance, some like the higher intensity of initial motion in flight take-off scenarios while others preferred smoother intensities. Thus, with tedious empirical tuning, it was possible to achieve the best experience for a particular person and a particular VR scenario. This proved that the major cause of the problem with immersion could be solved with proper tuning of the MCA. But the issue is that there are 80 parameters to tune in the MCA which includes 63 filter parameters with no direct physical meaning. Tuning for one person for one scenario would take a minimum of 2 to 3 sessions of 30 minutes duration. SP7 being built for a multi-purpose motion simulator could not afford this problem. A strategy to address this issue has been discussed in chapter 5.

To achieve a full-fledged virtual reality motion simulator, the above-stated problems must be overcome. As a result, the remainder of this dissertation will concentrate on resolving the above-mentioned problems.

Chapter 4

Solving the motion compensation issue with the lighthouse tracking system

All the VR devices are designed such that the frame of reference for the motion is the user's head wearing the HMD. When the user moves his head, the frame of reference is moved with respect to the head. For a VR motion simulator, these devices do not take into account the contribution of motion provided by the motion platform. This leads to a huge disruption of motion perception leading to motion sickness and reduced time in the simulator. Motion compensation shall be incorporated in order to compensate for the motion of the robotic platform. The ideal method is to use the direct kinematics at each time frame to extract the current pose of the platform and then compensate it for the user HMD. However, our SP7 platform is a parallel robot manipulator for which the use of direct kinematics is not a viable option in real-time due to the computational complexity of providing a solution every 8 ms and the intricacy of navigating 40 solutions at each instance. The next most adapted algorithm to resolve the motion compensation is available in [60], which basically extracts data from the tracking system and then transforms the pose with respect to the HMD. When the tracking device is mounted on the motion platform at a known base offset, the motion of the platform is compensated for.

Our initial empirical test of this motion compensation showed that during some motion, the perceived motion was compensated and ignored during some motion. This resulted in the source of the problem being investigated, particularly the tracking system and the algorithm for motion compensation. The latter was tested by playing with the type and parameters of the filters used in the algorithm, while at the same time ensuring the slow motion of the device so that the pose tracking would never be lost. These empirical tests proved that there were no significant differences with the changes in the motion compensation algorithm.

The tracking system must be assessed before a deep dive into the development of a new model for motion compensation. Therefore, the tracking system used for SP7, the HTC Vive lighthouse tracking system was again empirically tested with few configurations and it was visually observed through a custom-made unreal engine application that the tracking device drifts away significantly from its original pose at multiple instances. It was also observed by empirically changing a few configurations that the drift is not present at some configuration and also at some speed of the platform. This observation led to the in-depth evaluation of the lighthouse tracking system of HTC Vive with the aim of identifying the ideal configuration and the dynamic limit to which the tracking of the device does not deteriorate. Obtaining a solution to the problem of motion compensation by VR monitoring devices will be beneficial for SP7 since it is a VR motion simulator, and no extra costs would be applied to the final product to rectify this problem.

An extensive literature review on the evaluation of the pose tracking performance of the HTC Vive was conducted and tabulated in Table 4.1. Developers [61] do not have thorough performance metrics for VLTS, as it is a VR utility system. Different studies studied its efficiency either broadly or for a particular application since gaining attention as an affordable motion tracking alternative. Table 4.1 summarizes studies which analyze the spatial tracking efficiency of VLTS. In [62], the idea of using the tracker as a ground truth tracking system was investigated and an algorithm was provided using only light data to boost dynamic efficiency by losing accuracy in static situations. Precision assessment of the controller against Vicon MOCAP for clinical testing was done by [63]. [64] measured the static output of HMD with grid lines drawn on the floor with three separate tracking volumes for scientific research. [65] has done another static evaluation analogous to [64] for more detailed ground facts. The precision analysis was performed by [66] and [67] for rehabilitation and medical tracking purposes. [68] assessed the use of the controller for the automated testing of an industrial robot.

There are few Vive system assessment references for parameters other than spatial tracking. Time performance assessment (latency and jitter) for neuroscience and biofeedback studies under extreme time restrictions ($< 10ms$) was conducted in [69], which indicates that events shorter than 22 ms cannot be registered. The performance bound evaluation of the ultra-wide lighthouse positioning method for micro unmanned aerial vehicles was carried out in [70], demonstrating that the former is a viable alternative. In general, the potential to expand the tracking space of the lighthouse tracking device was explored in [71]. The consistency of the perception of the Vive and Oculus Rift systems has been compared in [72], concluding that the Vive is slightly stronger. Numerous experiments utilizing Vive as one of the set-up devices are not included in this study as they fell beyond the scope of this research.

Table 4.1 Brief overview of the latest research on Vive evaluation studies.

Ref.	Device	Space calibration	Precision evaluation	Static analysis	Dynamic analysis	Ground truth system	Intended application
[62]	T	d_{BS}: 1) 93 mm, 2) 6271 mm	1) σ = 5 mm, 2) σ = 28 mm	$\sigma_{lin} < 0.5$ mm, $\sigma_{ang} < 0.006^\circ$	Velocity range = 10 - 60 mm/s	Astrobee robot	Robotics
[63]	C	1) Room scale, 2)Standing	–	e_{AD}: 1) 0.74±0.42 mm, 0.46±0.46°; 2) 0.63±0.27 mm, 0.66±0.40°	–	UR5 robot with Vicon MOCAP	Clinical research
[64]	HMD	d_{BS}: 1) 7.45 m, 2) 5 m, 3) 5.66 m	–	RMSE: 1) $<$ 0.08 mm, $< 0.0111^\circ$, 2) $<$ 0.064 mm, $< 0.0113^\circ$, 3) $<$ 0.066 mm, $< 0.0053^\circ$	–	Grid lines on floor with String and Chalk (positioning error, σ = 17 mm)	VR for scientific research
[65]	T, C, HMD	d_{BS} : 1) 7.6 m, 2) 6.3 m	–	T: 1) $\sigma <$ 2.81 mm 2) $\sigma <$ 2.85 mm C Left: RMSE 1) 14.42 mm, 2) 5.15 mm; C Right: RMSE 1) 9.15 mm, 2) 8.81 mm, HMD: RMSE = 1) 11.48 mm 2) 7.93 mm	–	Phase Space MOCAP	Similar to [64] with accurate ground truth

Table 4.1 (*Continued.*) Brief overview of the latest research on Vive evaluation studies.

[66]	T	Tracking volume: 2.5 × 2.5 m	–	RMSE: 6.8±3.2 mm, 1.64±0.18°	–	Vicon MOCAP with 1) SCORBOT ER VII robot, 2) Human & VR games	Lumbar Postural change (Vive vs Vicon)
[67]	T	d_{BS} = 5.40 m	$\sigma <$ 0.722 mm & $<$ 0.278°	RMSE $<$ 1.050 mm & $<$ 1.124°	–	UR5 robot with Optitrack MOCAP	Tracking medical device
[68]	C	90° between base stations	$\sigma <$ 0.231 mm	$\sigma <$ 0.3 mm, one active BS: $\sigma <$ 2.1mm	–	ABB-IRB 140 robot	Automated testing of industrial robot

To understand the VLTS performance we need to know its performance under dynamic motion states. This will help identify the limits to which the VLTS could be used in tracking. To this extent, we address the three potential shortcomings of the literature in this chapter.

First, the performance assessment results vary on the basis of the set-up. For eg, [62] reports millimeter precision, while [67] and [68] report sub-millimeter precision. Similar uncertainty may also be found in static analysis, with all publications showing sub-millimeter accuracy, except for [65] and [66]. The findings of [65] are higher since the layout of the tracking area is greater than the recommended 5 m, as the author was evaluating for extreme cases. The low static performance of [66] may be due to infrared interference from their Vicon MOCAP ground truth device, as we observed the same problem of inaccuracy during our initial test of Vive with Optitrack MOCAP as the ground truth framework. This variance in documentation is attributed to a lack of uniform protocols for measuring output across applications. Among other problems, the VLTS being a hybrid device of numerous sensor data fused to render VR experience seamless with human motion perception does not have a standard evaluation technique. In this study, we are seeking to standardize the assessment of VLTS' motion tracking by considering the ASTM International Standard for bench-marking optical tracking systems with markers. The nominal comparison for precision, static and dynamic analysis was selected as [73], [74] and [75] respectively. These standards have been chosen because the key percentage of VLTS data is from lighthouse monitoring with photodiode sensors as explained in [76].

Second, as can be seen from Table 4.1, most of the literature conducts precision assessment and static analysis for their various intended applications, but there is a lack of performance assessment under dynamic motion conditions. While [62] compared their algorithm with Vive's off-the-shelf algorithm at speeds ranging from 10 to 60 mm/s, it does not provide VLTS dynamic pose tracking performance. Since most applications involve spatial tracking with different velocity intensities under a wide range of motions, it is important to have clearer and more detailed VLTS output statistics under different dynamic conditions. In this study, we also conduct an in-depth dynamic pose tracking study that includes the structured evaluation according to the [75] protocols, determination of the upper bound velocity of reliable tracking, examining the efficiency of pose tracking and the conditions of reliability within the upper bound found, and finally validating the results obtained with human wrist motion trajectory

Third,the suggested approaches and guidelines to test the performance of HTC Vive spatial tracking for a custom application by using the findings of this study are presented.

This chapter is further structured as follows: a brief overview of the experimental setup and equipment used, a detailed description of the experimental design along with the results, substantiation of the results obtained, procedure for optimal configuration of VLTS and testing of SP7 results.

4.1 Experimental setup

4.1.1 System architecture

The architecture of the experimental setup is elucidated in Fig. 4.1. The four primary blocks are 1. Ground truth motion generation, 2. Hardware 3. Integration Setup and 4. Post Processing. The role of the first block is to generate trajectories for the COMAU NS 16 robot and check whether the trajectories are within prescribed workspace. The path file is generated and sent to COMAU using Python PDL Parser. With the use of Universal Robot Description Format (URDF) file, the motion generation was simulated in MATLAB for COMAU NS16 robot. The hardware block basically represent the Vive tracking system, the devices (as shown in Fig. 4.4) and the robot itself. The data from the encoder is read and fed

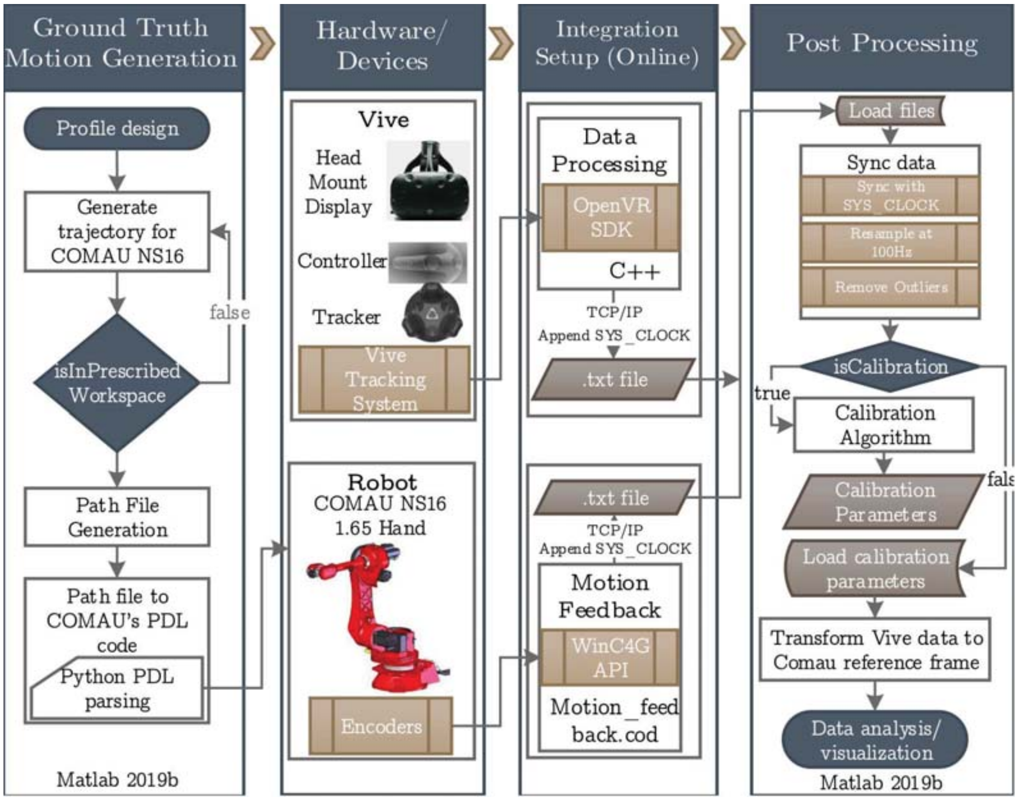


Figure 4.1 System architecture of the experimental setup. The source of the modules are made available in the GitHub repository [4]. The repository includes the C++ source code of the data collection API, the Python path to PDL parser script, and the MATLAB scripts developed for post-processing, transformation, generation of the performance metrics, and also for the tested registration algorithm.

to the motion feedback sub-block. The system integration for Vive tracking system is done using OpenVR SDK where the processing is performed using C++ programming language. The data processing and motion feedback sub-blocks are synchronized, re-sampled at 100 Hz and the outliers are removed in the post-processing section. The calibration algorithm is utilized and the corresponding parameters are obtained which are subsequently used to transform Vive data to COMAU reference frame. MATLAB 2019b is used for data analysis and visualization.

4.1.2 Vive lighthouse tracking system

Working of Vive's lighthouse tracking system is explained extensively by its developer [76]. Briefly, it is an optical navigation approach in which two base stations serve as a beacon with each one of them emitting a synchronization flash (blinking lights emitting diodes) and two infrared sweeping planes (laser beams) with two rotors spinning with constant velocity and orthogonally aligned. The objects to be tracked are equipped with photo-diodes and other sensors such as inertial measurement units (IMU) and gyroscope to measure linear and rotational accelerations respectively. The photo-diodes detect both the signals from the base stations (infrared sweep and LED flash) to triangulate the angle between the lighthouse's normal vector and the photo-diode with the time difference. Vive's off-the-shelf algorithm fuses all these sensor data to estimate the tracked pose relative to its world frame.

The physical space of the tracking system is configured as shown in the Fig. 4.2. The vendor recommended play area of 3500 × 3500 mm^2 and a maximum distance between the two bases stations (d_{BS}) of 5000 mm. Considering the above two criteria, each base station is kept 2500 mm away from the origin of the C16 at a height of 2500 mm. The same configuration of work volume is considered for all the evaluations.

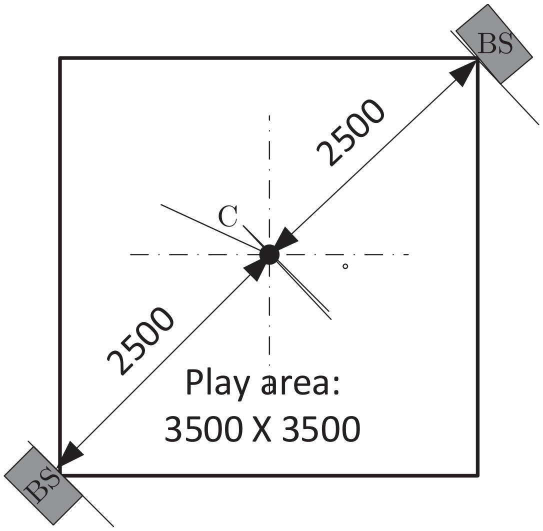


Figure 4.2 Work volume configuration sketch, top view. C: C16's Origin Point

Being an optics based system, tracking performance is highly dependent on various factors such as sunlight, presence of reflective surfaces, other IR rays interference and visibility of photo-diodes to the base stations. For instance; when used simultaneously with Optitrack Mocap system, Vive tracking became completely unreliable as the data kept shifting for a static pose. The following considerations were made while conducting the evaluation:

- The tracking area was covered with non reflective black sheets. Any reflective objects and windows were also covered. No other infrared emitting devices were placed in the tracking area.
- Base stations were synchronized using sync cable with channel "b" and "A" with room scale calibration
- Orientation of the robot's end-effector was configured such that the photo-diode of the tracking devices are visible to the base stations
- Receiver dongle of the tracker was set 45 cm far from the computer.

4.1.3 Ground truth system/ reference system

For the evaluation of the motion capture capabilities of VLTS, the best choice would be one of the gold standard system in that domain. In our initial tests with one such device, 'Optitrack MOCAP', the VLTS struggled to perform even in the best case static scenario. As contrasted to the millimeter scale errors in the standard tests, the errors with 'Optitrack MOCAP' were in the range of centimeters, and the jitter was high as well. This bad performance was found to be caused by the interference in infra-red bandwidth between MOCAP and VLTS. There have been some workaround solutions, such as capturing data without synchronization at various periods, utilizing relative steps, however these are not appropriate for dynamic performance evaluation.

We need a device that can be reliable and precise at high velocity and provide data with a latency of $<$ 20 ms [69] to evaluate the spatial tracking efficiency of VLTS, especially in dynamic cases. All these conditions are fulfilled by an industrial serial robot and, with its broad range of motion, it also facilitates a careful VLTS analysis. As the Ground Truth Unit, C16 was used (shown in Fig. 4.3). It has 0.05 mm repeatability and six DOF mobility as mentioned in [77]. The latency was estimated at an average of 8 ms and a maximum of 12 ms between the motion feedback frames of C16, which is well within the specifications.

Customized 3D printed mounts were developed for each of the devices to mount the VLTS devices to the C16 end-effector flange (ISO9409-1-A63), as shown in Fig. 4.4. The mounts for the dynamic performance assessment are shown in Fig. 4.5. As recommended by the ASTM [75] standard, all of the same tracking devices/object must be placed on either side of the metrology bar. During analysis, Metrology bar length serves as a reference quantity. The carbon fiber material for the metrology rod was recommended in the standard's

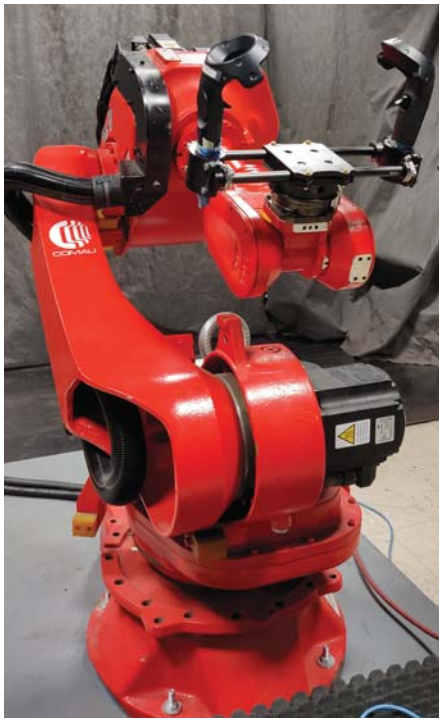


Figure 4.3 Ground truth system: Comau NS16 1.65 foundry hand robot with the mount for controller dynamic analysis attached to the end-effector

specifications such that the deflection will be less than or equal to 0.01 mm. The tool mount and the attachments are 3D printed.

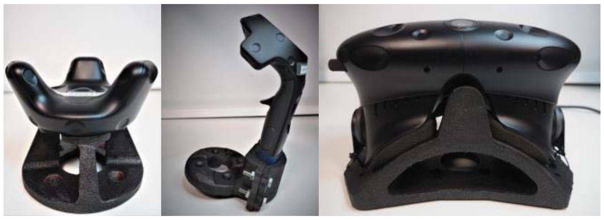


Figure 4.4 Vive tracking devices with their 3D printed mounts for flange ISO9409-1-A63. From left: tracker, Controller, and HMD.

For ease of trajectory planning, we are adopting the concept used to test the industrial robot known as the prescribed workspace. This workspace of the robot is a manifold inside the complete workspace envelope such that the trajectories are continuous in all DOFs. A detailed overview of this workspace is as follows.

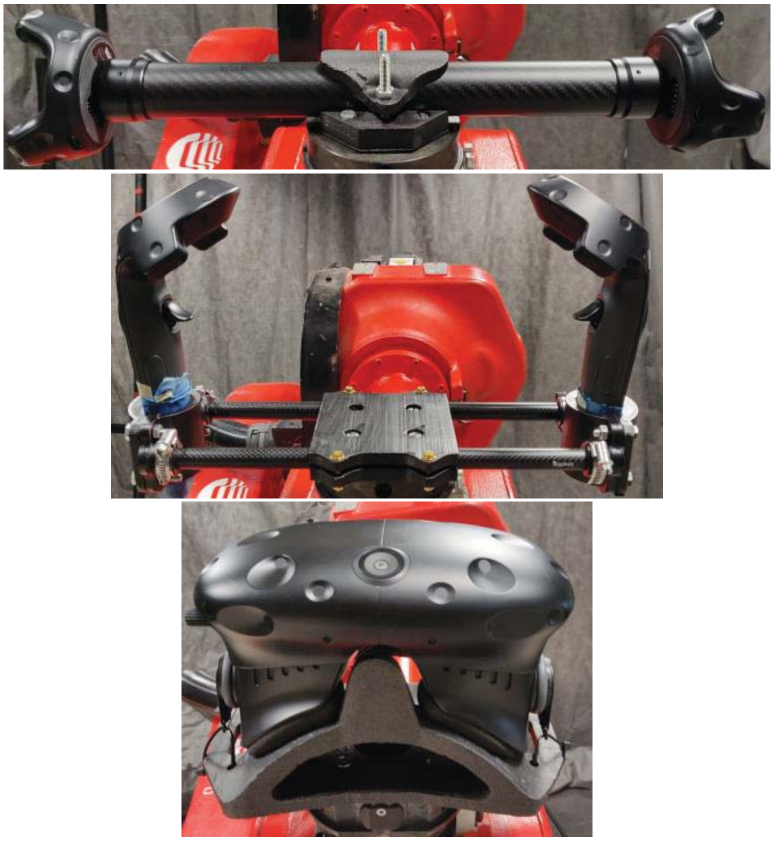


Figure 4.5 Metrology bar, with devices on both ends and its C16 tool mount. Top: A 350 mm long carbon fiber tube, with VLTS's tracker on both ends. Middle: An assembly of two carbon fiber rods and rack, with VLTS's controller on both ends. Bottom: HMD mounted on the end-effector of C16.

Prescribed workspace

The joint space of the 6-DOF industrial COMAU NS-16 foundry hand manipulator is denoted by $\mathbf{q} = [q_1, \ldots, q_6]$ and $\mathbf{X}$ represents the task space, such that $\mathbf{X} = f(\mathbf{q})$. The set of all reachable configurations constrained by joint limits is defined by the joint domain $\mathbf{Q}$, which is mathematically expressed as $\mathbf{Q} = \{\mathbf{q} | q_{imin} \leq q_i \leq q_{imax}, \forall i = 1, \ldots, 6\}$.

The image of joint domain $\mathbf{Q}$ by forward kinematic mapping using homogeneous transformation matrices defines the workspace W of the robot, $W = f(\mathbf{Q})$. Hence, the workspace W is the set of poses reachable by the robot end-effector. The workspace boundaries are defined by the joint limits and singularities. The analysis of the COMAU NS-16 hand's workspace is crucial for the motion planning and design of tracking experiments, which is calculated using the specifications given in manufacturer's datasheet [77].

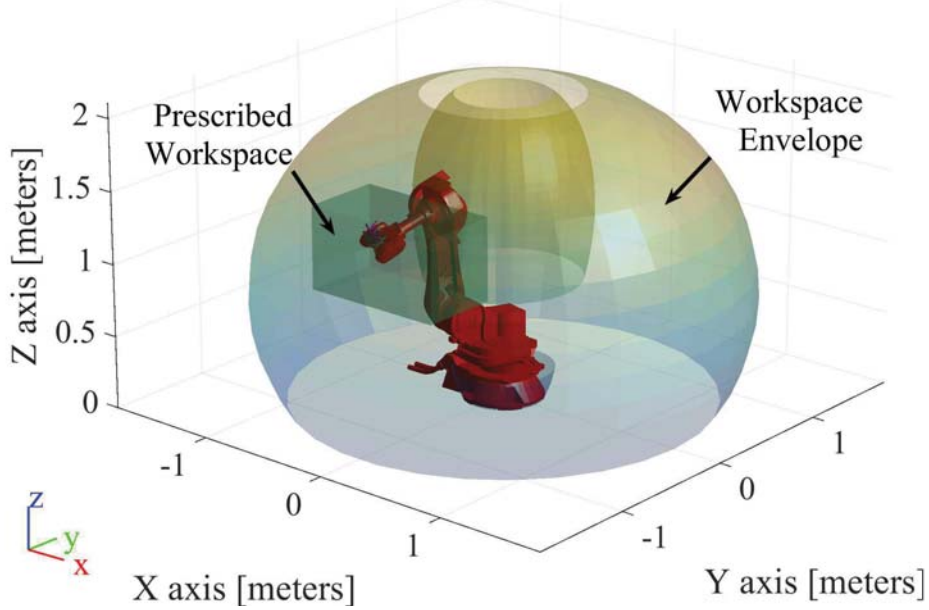


Figure 4.6 Prescribed workspace: A regular geometric form that contains the range of motion of robot's end-effector presented inside the COMAU NS16 hand workspace envelope

According to the ISO9283 standard [78], the test environment for manipulating industrial robots are defined within a specific region of the working space called ISO cube. The ISO cube is located inside the workspace envelope and has the maximum volume. For the proposed experimental scenario, we utilize the notion established by the standard and define the prescribed workspace [79]. The prescribed workspace contains the desired range of motion of the end-effector and has standard geometric form of a cuboid with dimensions 1 m x 0.55 m x 0.6 m. The prescribed workspace and the workspace envelope of the 6-DOF industrial robot is shown in Fig. 4.6.

4.1.4 Calibration procedure between VLTS and C16

VLTS and C16 have their own absolute reference frame. In order to compare motion data between two systems, they must be in the same frame of reference. The first and the most common method for calibrating two independent systems is to have an intermediate frame from one system's frame at a known precisely measured point with respect to the other systems origin frame. With the known offset, at each data collection the intermediate frame is used to transform from one system to another. This is not suitable for performance evaluation as the system error will be reflected in each data frame. The next method, widely used in the field of computer vision since 1986, is the least square fitting technique (calibration/registration process). This has been adapted for evaluating system performance (Vicon MOCAP assessment [80]). This method requires the collection of matching data points from both the system and estimating the parameters of the rigid body transformation matrix. This topic is itself vast with solutions explored from various aspects like closed form vs iterative solution, and etc. A detailed survey and categorization of the existing methods are presented in [81] and [82]. For our experimental setup, we assessed the effectiveness of two closed-form solution algorithms [83] and [84]. The mathematical notation to describe the relationship between different frames and components of a robotic system is as follows. The linear regression method will be introduced to find the transformation matrix between two sets of 3-D points. The Cartesian coordinates of an arbitrary point U with respect to the frame {M}, described by the origin O and the axes is denoted by (x_M,y_M,z_M). The homogeneous coordinates of U with respect to frame {M} are defined by $\begin{bmatrix} x_M & y_M & z_M & 1 \end{bmatrix}^\mathsf{T}$. The homogeneous transformation matrix represents the translation and/or rotation of a frame {N} into frame {M} and is given by:

$$^{N}T_M = \begin{bmatrix} ^{N}s_M & ^{N}n_M & ^{N}a_M & ^{N}P_M \end{bmatrix} = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

where $^{N}s_M$, $^{N}n_M$ and $^{N}a_M$ consists of the components of the unit vectors along the x_M, y_M and z_M axes respectively expressed in frame {N}. Given two sets of 3-D points from different coordinate systems: {M} and {N}, the following equation holds:

$$\begin{bmatrix} x_N \\ y_N \\ z_N \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_M \\ y_M \\ z_M \\ 1 \end{bmatrix} \tag{4.2}$$

The homogeneous transformation matrix is expressed as a set of linear equations. Expanding Eqn. 4.2 results in

$$\begin{aligned} x_N &= s_x x_M + n_x y_M + a_x z_M + p_x \\ y_N &= s_y x_M + n_y y_M + a_y z_M + p_y \\ z_N &= s_z x_M + n_z y_M + a_z z_M + p_z \end{aligned} \tag{4.3}$$

The objective of linear regression is to find the best transformation matrix in order to minimize

$$L^2 = \sum_{i=1}^{n} [{}^N U_i - (\mathbf{R}^M U_i + \mathbf{P})]^2 \tag{4.4}$$

where L represents the residual, the matrix **R** is the rotation and the column matrix **P** denotes the translation.

$$\begin{aligned} L^2_{x_N} &= \sum_{i=1}^{n} [x_{Ni} - (s_x x_{Mi} + n_x y_{Mi} + a_x z_{Mi} + p_x)]^2 \\ L^2_{y_N} &= \sum_{i=1}^{n} [y_{Ni} - (s_y x_{Mi} + n_y y_{Mi} + a_y z_{Mi} + p_y)]^2 \\ L^2_{z_N} &= \sum_{i=1}^{n} [z_{Ni} - (s_z x_{Mi} + n_z y_{Mi} + a_z z_{Mi} + p_z)]^2 \end{aligned} \tag{4.5}$$

The minimum value of the residuals are obtained by taking the derivative of Eqn.4.5.

$$\frac{\partial L^2_{x_N}}{\partial s_x} = 0 \frac{\partial L^2_{x_N}}{\partial n_x} = 0 \frac{\partial L^2_{x_N}}{\partial a_x} = 0 \frac{\partial L^2_{x_N}}{\partial p_x} = 0 \tag{4.6}$$

From the above equation, one can obtain the values for the first row of the transformation matrix which is as follows:

$$\begin{bmatrix} s_x \\ n_x \\ a_x \\ p_x \end{bmatrix} = \begin{bmatrix} \mathbf{H} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} x_{Ni} x_{Mi} \\ \sum_{i=1}^{n} x_{Ni} y_{Mi} \\ \sum_{i=1}^{n} x_{Ni} z_{Mi} \\ \sum_{i=1}^{n} x_{Ni} \end{bmatrix} \tag{4.7}$$

where **H** is given as, where all sums are from $i = 1$ to n,

$$\begin{bmatrix} \mathbf{H} \end{bmatrix} = \begin{bmatrix} \sum x^2_{Mi} & \sum x_{Mi} y_{Mi} & \sum x_{Mi} z_{Mi} & \sum x_{Mi} \\ \sum x_{Mi} y_{Mi} & \sum y^2_{Mi} & \sum y_{Mi} z_{Mi} & \sum y_{Mi} \\ \sum x_{Mi} z_{Mi} & \sum y_{Mi} z_{Mi} & \sum z^2_{Mi} & \sum z_{Mi} \\ \sum x_{Mi} & \sum y_{Mi} & \sum z_{Mi} & n \end{bmatrix} \tag{4.8}$$

The above procedure is repeated for second and third rows of the matrix defined in Eqn.4.2 thereby resulting in all components of the desired transformation matrix ${}^N T_M$. Once the transformation matrix is calculated, the Euler angles are obtained using ZYZ rotation sequence.

The specifications of the data for the approach being discussed are straightforward. The only prerequisite is the use of data pairs with point to point correspondence in two reference frames. In other words, these data sets must be paired in terms of pose in order to calculate the transformation matrix.

The angular distance between rotations is computed by utilizing the Rodrigues formula. The rotation matrix $\mathbf{R}$ is defined as:

$$\mathbf{R} = \cos\theta\mathbf{I} + \sin\theta[\mathbf{u}]_\times + (1-\cos\theta)\mathbf{u}\mathbf{u}^\mathsf{T} \tag{4.9}$$

where $\mathbf{u}$ is the rotation axis for a given rotation of unit magnitude, θ is the rotation angle and $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$. The angle of the difference rotation can be computed by calculating the trace of $\mathbf{R}$. In our case, the calibration was done by traversing the prescribed workspace (explained in 4.1.3) at very low speed of 1 millimeter per second for 13 hours. After post processing the data, a total of 1,364,011 number of samples were collected with point correspondence. Despite providing good transformation results with RMSE of 4.1 mm and 4.5 mm, we encountered a significant problem. Being a non-rigid body transformation, the outliers were also transformed i.e., the transformation includes even the deformed dataset. Also, for sub-millimeter evaluation, a RMSE of 4 mm is not acceptable. Thus, this method is also not suitable for our evaluation. The above-mentioned methods are used to obtain the absolute pose error in which there is a significant dependency between the two systems.

In our experimental system, as the time data are synchronized, we could evaluate the relative pose error in their own respective frame without transforming the frame from C16 to VLTS. This method fits our criteria perfectly because the error is measured without influencing either of the systems thereby removing any inter-system dependency. Let's say we have a dataset of N sample size from the C16 and VLTS. Each pose (2:N) is transformed with respect to the first pose in its own frame and then the pose error is calculated using equation (4.10).

$$\begin{aligned} 0 \le e_{RelAngle,k} &= \cos^{-1}(\frac{trace(R_k)-1}{2}) < \pi \\ e_{RelTran,k} &= \sqrt{(\hat{x_k}-\hat{x_1})^2) + (\hat{y_k}-\hat{y_1}) + (\hat{z_k}-\hat{z_1})^2)} \\ &\quad -\sqrt{(x_k-x_1)^2) + (y_k-y_1) + (z_k-z_1)^2)} \end{aligned} \tag{4.10}$$

where $e_{RelAngle,k}$ is the relative angle error, $e_{RelTran,k}$ is the relative translation error, $R_k = {}_1R_k\hat{R_k}^T$ represents the rotation difference expressed in [3×3] rotation matrix. The characters with $\wedge$ accent mark denotes the data measured in the coordinate frame of the VLTS, and the characters with no accent mark are data measured in the C16' coordinate frame.

4.2 Static pose tracking performance evaluation

The purpose of this experiment was also to obtain a reference measure of the quality of our setup compared to the literature [66, 68, 62] in addition to evaluating the system's performance under static conditions. Static pose tracking was evaluated for both precision and accuracy. Despite comprehensive literature on precision and accuracy analysis as illustrated in Table 4.1, they were undertaken for two key reasons: validating our experimental setup against the literature; and to address the variation in the literature's reports by introducing a standardized methodology and evaluation parameters.

4.2.1 Precision evaluation

Design of experiment

This test examines the precision of VLTS in static scenarios. The repeatability of the C16 (0.05 mm) makes it easy to revisit the command points precisely, which is suitable for precision evaluation. The procedure is as follows: the tracking device has been placed at 10 random points (as seen in Fig. 4.7) and the VLTS and C16 data has been recorded. This was repeated ten times in a random visit order. The randomness of the visits is to have "independent test results" [73]. It was an arbitrary choice of 10 points and 10 visits. The pose of each point is assessed 10 times, leading to a total of 100 assessments that will provide reliable data for one tracking device. For all three units, the same process was followed.

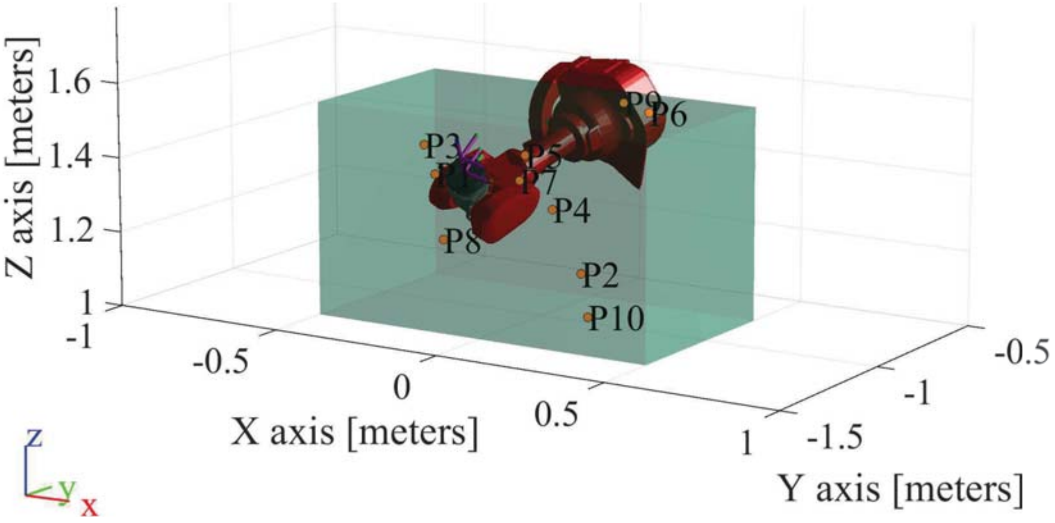


Figure 4.7 10 randomly selected points (P1 to P10) within the prescribed workspace. For each iteration, the order of visit to P1 to P10 was randomly selected.

In general, the measure of precision is evaluated in terms of imprecision and calculated as standard deviations (RMSD) [73]. Furthermore, *dmax*, E(50), E(95), and E(99.7) were determined to report the quality of the performance of pose tracking. *dmax* shows the highest error obtained, regardless of the number of occurrences, which includes the outliers. An overview of the confidence level of pose tracking is given by the percentile values. For eg, if the 99.7 percentile value is closer to the mean value, we can infer that the pose tracking is reliable with 99.7 % confidence.

In addition to the aforementioned criterion, two statistical hypothesis tests, AET and MPET were performed. The first test was done to verify whether VLTS, as the literature claims (Table. 4.1), has an average precision of less than 1 mm and 0.5°. The latter seeks to infer that the maximum deviation is less than 10 mm and 1 deg from the mean. The null and alternate hypotheses are set as seen in the table (4.2). The same hypothesis was tested for both location and orientation on all the three VLTS devices.

Table 4.2 Statistical tests for the precision evaluation of VLTS.

Test	Null Hypothesis	Alternative Hypothesis
AET	$H_0 : \bar{d} \leq \delta_{avg}$	$H_a : \bar{d} > \delta_{avg}$
MPET	$H_0 : d_{max} \leq \delta_{max}$	$H_a : d_{max} > \delta_{max}$

For the average error test, the sample size is $N > 30$ and the population standard deviation is unknown, so the Z-test is conducted to confirm the hypothesis [85]. In statistical sense, the precision is less than δavg if the AET accepts the null hypothesis with a p-value of > 0.95, i.e. if the equation (4.11) is not fulfilled.

$$\frac{\bar{d} - \delta_{avg}}{\sqrt{s^2/N}} > Z_\alpha \tag{4.11}$$

where, Z_α = 1.6449 (Z score at 0.95).

For MPET, the null hypothesis i.e., the maximum deviation is less than 1 cm and 1° if the equation (4.12) is false.

$$\frac{\delta_{max} - d_{max}}{d_{max} - d_{min}} < \frac{\alpha}{1 - \alpha} \tag{4.12}$$

where, $\alpha = 0.05$.

Results

According to the methods explained above, the precision measurement was evaluated and tabulated in the Table 4.3 along with the other parameters for accessing the quality of tracking. It can be found that, with RMSD less than 0.55 mm and 0.12°, all three devices showed high precision.

Table 4.3 Precision evaluation results: Statistics on deviations from mean.

Device	DOF	RMSD	E(50)	E(95)	E(99.7)	$\bar{d}$	s^2	d_{max}
T	P [mm]	0.5352	0.2675	1.0512	2.5625	0.3763	0.1448	3.4410
N: 61200	O [deg]	0.0600	0.0556	0.0915	0.1374	0.0550	0.0006	0.1592
C	P [mm]	0.5023	0.2454	1.0285	2.0286	0.3544	0.1267	2.4804
N: 61140	O [deg]	0.1155	0.1123	0.1585	0.1822	0.1116	0.0009	0.1993
HMD	P [mm]	0.1579	0.7401	0.3300	0.5243	0.1179	0.0110	0.873
N: 61140	O [deg]	0.0809	0.0803	0.1161	0.1607	0.0749	0.0009	0.2041

Fig. 4.8 shows that the average retro-projection error of each static point measured is less than 0.005 meters. Obtained results were consistent with the values reported in the literature with accuracy of 0.005864 meters, standard deviation of 0.000139 meters, root mean square error (RMSE) of 0.003105 meters.

From the CDF shown in Fig. 4.9, it is evident that HMD has an excellent precision with the peak value of 0.74 mm. Tracker and controller tracked 94.155% and 94.578% of data within 1 mm. For the angular DOF shown in Fig. 4.10, tracker oriented 97.259% of the sample within 0.1° precision. HMD and Controller tracked 78.471% and 0.333% within 0.1°. The worst deviation of 2.5625 mm at 99.7 percentile further confirms the excellent results obtained with VLTS on the overall quality of the static spatial tracking precision.

Using the value of $\bar{d}$ and s^2 in the Table. 4.3, the hypothesis tests were conducted and the results are tabulated in the Table. 4.4. All the H_0 was accepted. Thus, we can statistically infer the following for of all the three VLTS devices:

- The average deviation from mean is in sub-millimeter ($<$ 1 mm) and less than 0.5°.
- The maximum deviation from mean is less than 10 mm and 1°.

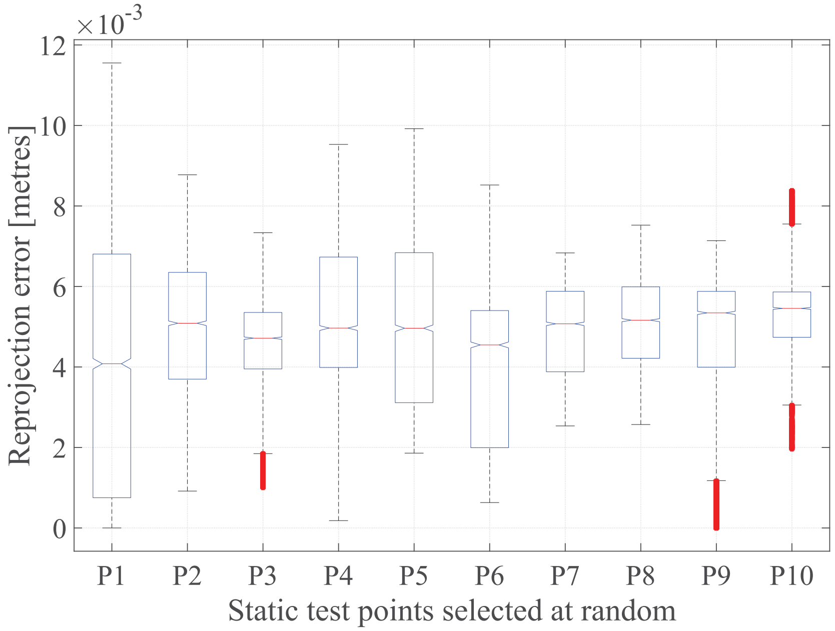


Figure 4.8 Retro-projection error for each measured static point

Table 4.4 Precision evaluation: Hypothesis test results.

Device	DOF	Z value	H_0	ME value	H_0
T	P	-405.4498	Accepted	1.9061	Accepted
	O	-4606.1015	Accepted	5.2800	Accepted
C	P	-448.4230	Accepted	3.0316	Accepted
	O	-3225.4311	Accepted	4.0188	Accepted
HMD	P	-2077.8760	Accepted	12.5117	Accepted
	O	-3446.2159	Accepted	3.8986	Accepted

4.2.2 Accuracy evaluation

Design of experiment

Accuracy analysis is an assessment of the accuracy of the system when the object to be tracked and the measuring system is both static in comparison to each other. This study is one of the most popular literature assessments available. Despite the variety of application of the VLTS, each analysis did not adopt a standard collection of metrics. There are also variations in the results, some claiming precision in millimeters and some report accuracy

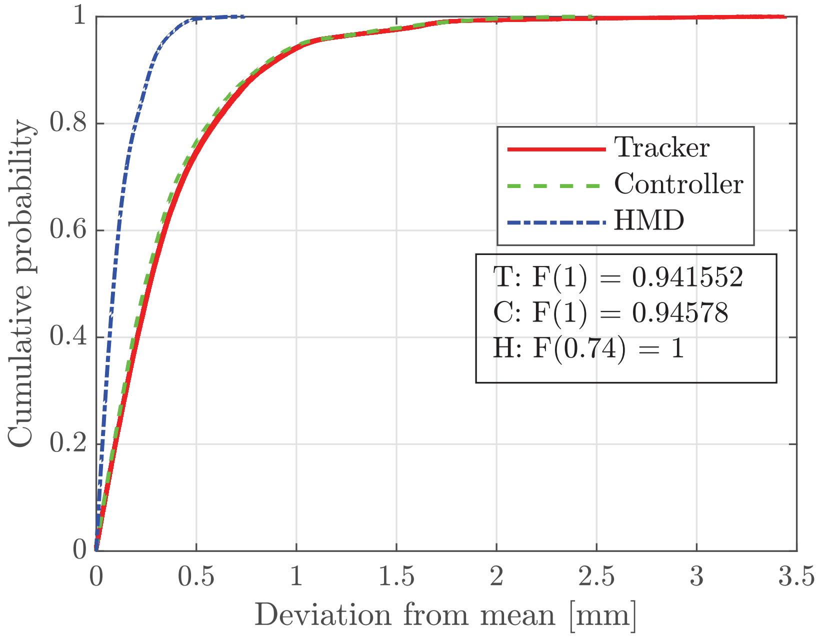


Figure 4.9 Precision evaluation: CDF of positional deviation from mean.

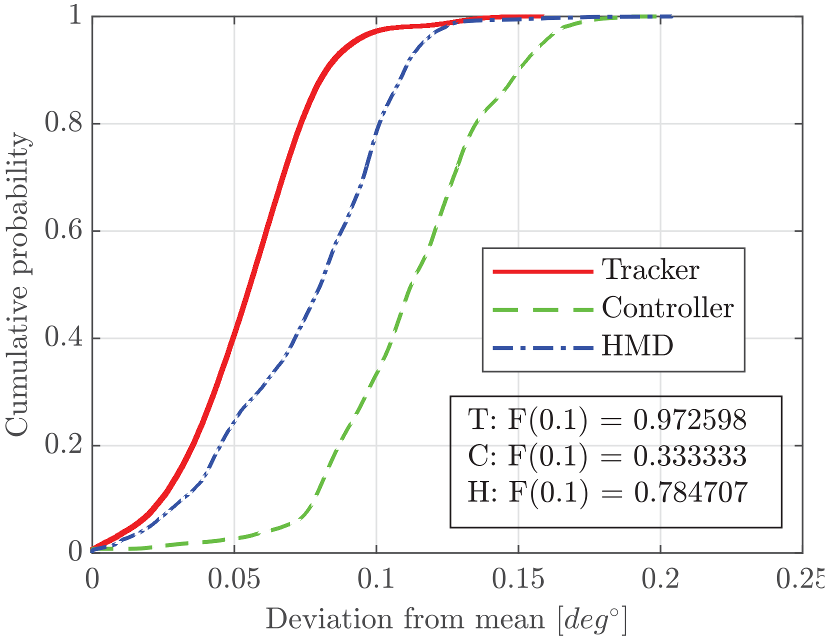


Figure 4.10 Precision evaluation: CDF of orientation deviation from mean.

in sub-millimeters. This is primarily induced by the chosen ground truth scheme, testing methodology, and assessment criteria. We intend to overcome the uncertainty by adapting a consistent approach and performance measurement metrics from the ASTM standard for optical motion tracking devices namely [74]. measurement of the selected ground truth system must be less than the test system (VLTS) measurement uncertainity, as is the case of our setup with C16. The technique suggested by [74] is a reasonably simple experimental procedure. The object/device to be tracked is placed at 'N' random locations and the pose is measured simultaneously by the reference system (C16) and the system under test (VLTS). To ensure that the average error follows a normal distribution according to the central limit theorem, the number of samples (N) should be greater than 32 (N$\geq$ 32). A sample size of 100 was arbitrarily selected to provide reliable results (N=100).

Using the technique described in the above section and the equation (4.10), the error at pose (*ek*) is calculated when data collection is concluded. The hypothesis tests were conducted with the same null and alternative hypothesis as tabulated in Table 4.4 but with δ_{avg} as 5 mm and 0.5°, and δ_{max} as 10 mm and 1°. The equations (4.11) and (4.12) were used to validate the null hypothesis by replacing $\bar{d}$ with $\bar{e}$, d_{max} and d_{min} with e_{max} and e_{min}.

Results

Table 4.5 includes the outcomes of accuracy analysis using the standard procedure described above. Instead of the sub-millimeter range of other literature, the RMSE was found to be in millimeter range. From the CDF of the position and orientation error shown in Fig. 4.11 and Fig. 4.12, it can be seen that the HMD tracked with 96.44% and 98.99% of the sample within 3 mm and 0.5°, controller's 85.61% and 95.48% of data has error below 3 mm and 0.5° whereas for the same error value, tracker has 67.04% and 51.56%.

The outcomes of the hypothesis tests are tabulated in Table. 4.6 with their corresponding test scores (Z value: Result of the AET in equation (4.11), ME value: Result of the MPET in equation (4.12)). All the null hypothesis was satisfied except for the liner DOF of the tracker device, the maximum permissible error's hypothesis was rejected. Upon further analysis, the maximum permissible positional error of the tracker was identified to be less than 11 mm.

Thus in a statistical sense, we can infer the following on the static performance (accuracy) of the three VLTS devices:

1. The average pose error is less than 3 mm and 0.5°

2. The maximum pose error is less than 11 mm and 1°

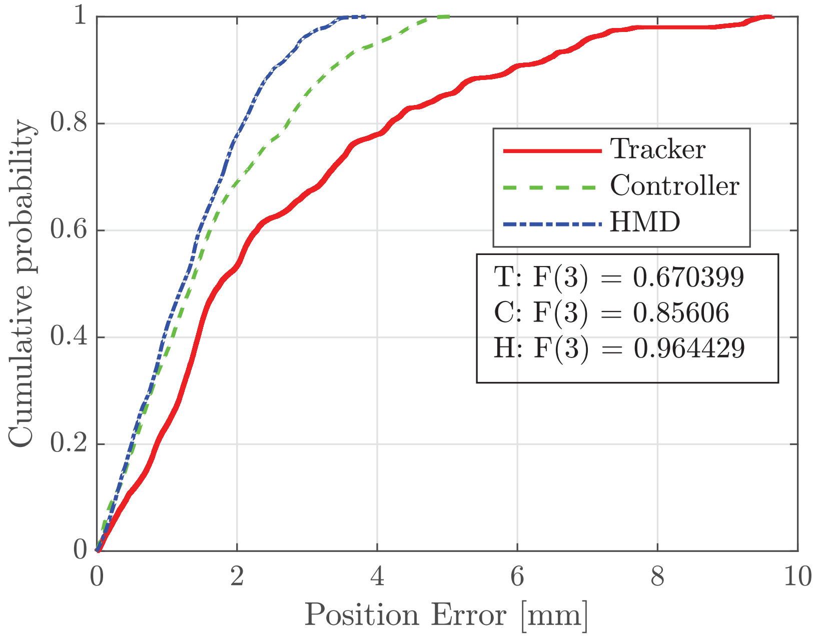


Figure 4.11 Accuracy analysis: CDF of position error.

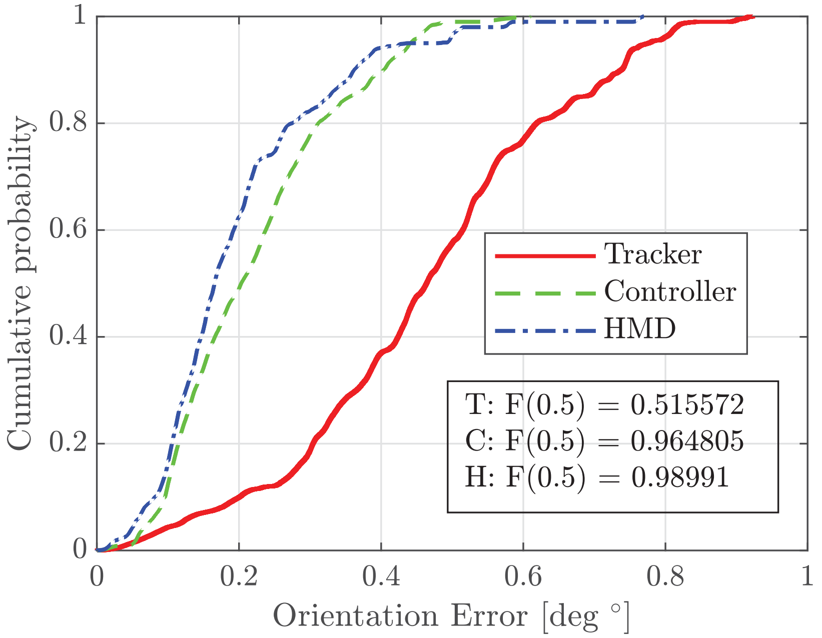


Figure 4.12 Accuracy analysis: CDF of orientation error.

Table 4.5 Accuracy analysis results: Statistics on relative pose error.

Device	DOF	RMSE	E(50)	E(95)	E(99.7)	$\bar{e}$	s^2 [mm^2, deg^2]	e_{max}
T	P [mm]	3.3091	1.7581	6.9137	9.4207	2.5499	4.4482	9.6425
N: 61185	O [deg]	0.5000	0.4638	0.7762	0.9131	0.4622	0.0364	0.9233
C	P [mm]	1.9826	1.3529	3.9879	4.7795	1.5963	1.3827	5.0362
N: 61243	O [deg]	0.2494	0.2031	0.4409	0.5835	0.2210	0.0133	0.6101
HMD	P [mm]	1.5688	1.2313	2.8782	3.4794	1.3209	0.7165	3.8349
N: 61140	O [deg]	0.2345	0.1657	0.4796	0.7643	0.1984	0.0156	0.7711

Table 4.6 Accuracy analysis: Hypothesis test results.

Device	DOF	Z value	H_0	ME value	H_0
T	P	-40.8321	Accepted	0.0371	*Rejected*
	O	-37.9068	Accepted	0.0831	Accepted
C	P	-295.4226	Accepted	0.9856	Accepted
	O	-597.7264	Accepted	0.6390	Accepted
HMD	P	-490.6218	Accepted	1.6076	Accepted
	O	-596.5022	Accepted	0.2968	Accepted

4.3 Planar pose tracking performance evaluation

Design of experiment

Before indulging spatial dynamic analysis, the planar pose tracking performance was explored so that it would yield some interesting observations such as how much error can happen in-phase and out-phase. The analysis is designed to have a bidirectional circular motion on three planar motions orthogonal to each other at three different peak velocities of robot. The circular motion was designed synthetically; one circular motion traverses from start position to end position and then back to the start. This bidirectional circular motion is repeated for 10 times for all the three chosen velocities. Planes chosen were XY, YZ, and XZ. Velocities chosen were 0.1 m/s, 0.75 m/s and 1.5 m/s.

Results

Fig. 4.13 investigates the distance error for each peak velocity. As expected, at the speed less than 0.1 m/s the tracker performs reasonably well with a peak error of 1.5 cm. At higher speed than 0.6 cm/s the values becomes irrelevant as the distance errors are in meters.

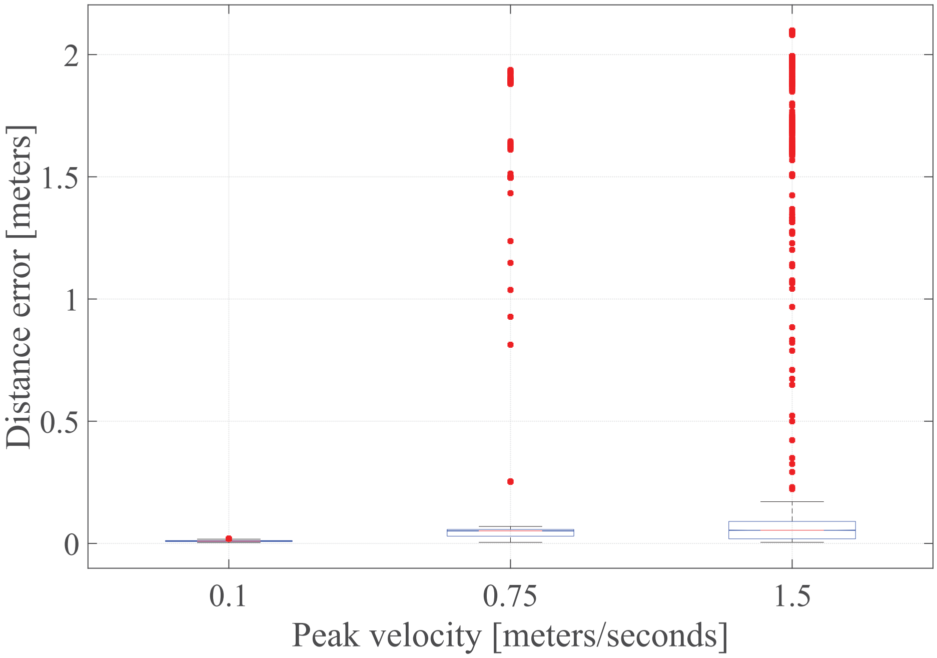


Figure 4.13 Retro-projection error for each measured static point

Fig. 4.14 shows that the tracking is better in XZ planes even at peak velocity of 1.5 m/s. While other planes suffer 4.14. This is because during its entire path, it's photo-diode were visible to the base station B1. This is consistent with the static analysis that Y axis had least error. The results of the experiments are tabulated in table 4.7.

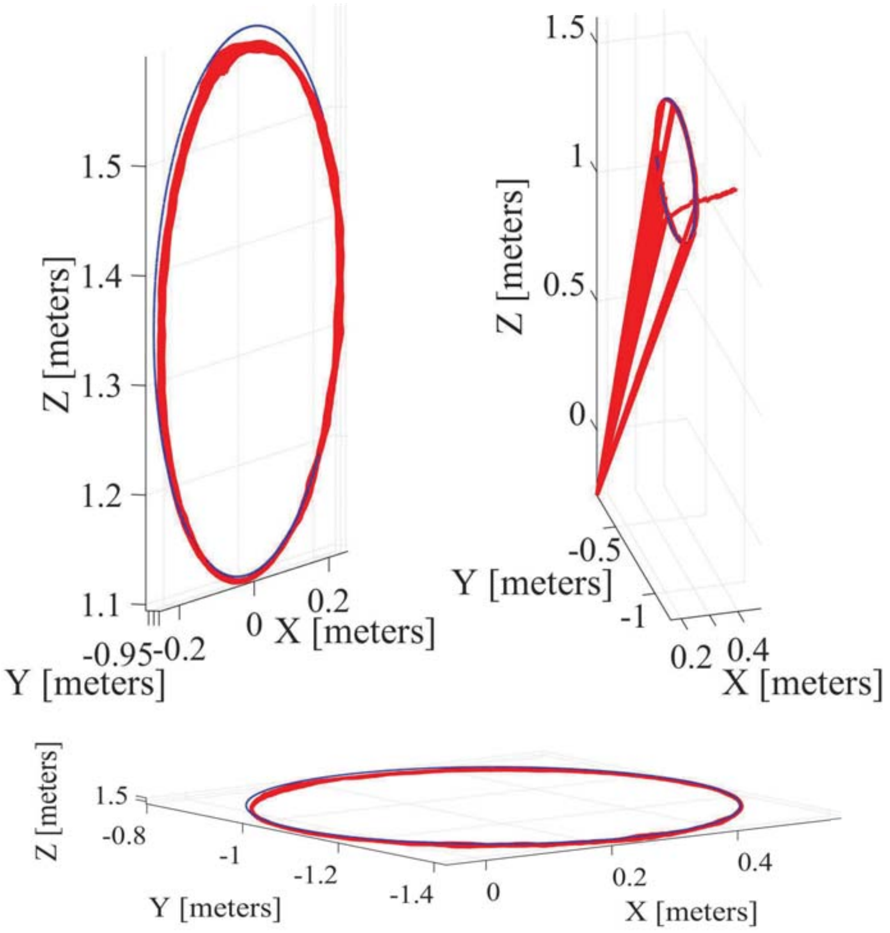


Figure 4.14 Three planar circular motion. Blue line indicates the robot trajectory and red line indicates Vive's trajectory. Top left: XZ planar motion at peak velocity of 1.5 m/s. Top right: YZ planar motion at peak velocity of 0.1 m/s. Bottom: XY planar motion at peak velocity of 0.75 m/s

4.4 Dynamic pose tracking performance evaluation

The objective of the analysis is the evaluation of VLTS's pose tracking capability while the system is in motion. Dynamic analysis is not prevalent in the literature, as described in the Table 4.1. We present the process for evaluating output against varying velocity, setting the upper limit, and checking the results with complicated and randomized trajectories in addition to the standard procedure of [75]. In order to prevent p-hacking ([86]), the whole experiment was done with one goal: to study the performance of the pose tracking against velocity variations.

Table 4.7 Error Analysis: Planar. Label is annotated as follows: first two indices represents the plane of the robot (X, Y, Z); third index represents the peak velocity (1: 0.1 m/s, 2: 0.75 m/s, 3: 1.5 m/s). MAE- Mean absolute error, σ_{dist} - Standard deviation distance error, RMSE-Root mean square error, TL- Time lag, MAVE- Maximum absolute velocity error

Label	MAE	σ_{dist}	RMSE	TL	MAVE
XY1	0.0050	0.0026	0.0099	0.2259	1.0566
XY2	0.0201	0.0182	0.0482	0.0989	0.2890
XY3	0.0311	0.1290	0.1461	0.1928	104.1149
XZ1	0.0064	0.0677	0.0689	0.0338	92.7520
XZ2	0.0183	0.0195	0.0464	0.0240	0.7294
XZ3	0.0221	0.0343	0.0612	0.0260	1.1318
YZ1	0.0064	0.0809	0.0820	0.1195	97.9673
YZ2	0.0236	0.1301	0.1404	0.0634	94.6722
YZ3	0.0491	0.3077	0.3262	0.0716	99.1022

4.4.1 Standard procedure evaluation

Design of experiment

The norm outlines the testing procedure and a collection of statistical performance metrics for an optical tracking device performance to be evaluated. One of the suggested test volumes of 3000 × 2000 × 2000 mm^3 was configured within our work volume of 3500 × 3500 × 2200 mm^3. The tracking device to be evaluated is connected to the ends of the metrology bar as is seen in Fig.4.5. The procedure requires moving the position of the centroid of the metrology bar at a relatively constant walking speed of 1500 mm/s (1200 ± 700 mm/s) along the work volume in two patterns(X, Y) as shown in Fig. 4.15 at a height of 1000 mm.

Via continuous smooth motion, the centroid of the metrology bar is moved around the test volume in two patterns (X and then Y) with three distinct orientations. The orientations are designed in such a way that the line going through the metrology bar and the devices aligned parallel to the three orthogonal planes. Thus, a total of 6 trajectories are tested for a single evaluation at the ordered average velocity.

Since the relative pose between the left and right tracking devices is unchanged at all times, the pose measurement error is determined by comparing ${}_{Left}\hat{H}_{Right}$ of VLTS and ${}_{Left}H_{Right}$ of C16. Timetamp is used to synchronize the data. Position error (e_p) and orientation error

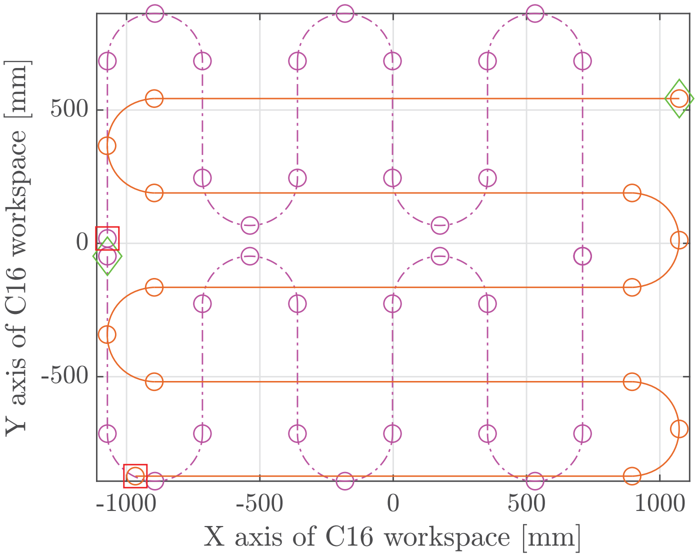


Figure 4.15 Pattern trajectories for dynamic analysis adapted to the C16's workspace. X pattern (horizontal lines-red) and Y pattern (vertical lines-cyan). Each pattern is parallel, straight line segments, back-and-forth along their corresponding axis with the paths separated by at most the length of the metrology bar. At maximum, one-half of the metrology bar length shall be the difference between the boundary lines and the test volume limits.

(e_o) at time instance t are determined as seen in 4.13.

$$\begin{aligned} e_p(t) &= ||\hat{T}(t)|| - ||T|| \\ e_o(t) &= \hat{\theta}(t) - 0 = \hat{\theta} \end{aligned} \tag{4.13}$$

where $\hat{T}(t)$ and $\hat{\theta}(t)$ are the position vector and the angle of rotation of ${}_{Left}\hat{H}_{Right}$, T is the position vector of ${}_{Left}H_{Right}$, $||\quad||$ denotes 2-norm of the vector. $\hat{\theta}(t)$ is calculated using the equation 4.14.

$$\hat{\theta}(t) = 2 * asin(\sqrt{\hat{q}_x^2(t) + \hat{q}_y^2(t) + \hat{q}_z^2(t)}) \tag{4.14}$$

where, $(\hat{q}_w(t), \hat{q}_x(t), \hat{q}_y(t), \hat{q}_z(t))^T$ is the unit quaternion representation of the rotation matrix $\hat{R}(t)$.

The error statistics computed are the RMSE, e_{max}, and the percentile error (E(99.7), E(95) and E(50)). Similar to previous section 4.2.1 and 4.2.2, the AET and MPET hypothesis tests were performed. Here, instead of checking the error in the millimeter or centimeter range, we intend to inspect the device in a broader perspective by examining whether the devices are capable of localizing themselves, i.e. being able to position themselves within the dimensions of the device. As such, we recommend that the average error limit should not be too far away from the sides of the device; rather, it should stay within the dimensions of the device. The half of the maximum of three dimensions (length, width and height) was chosen to be the expected average error limit δ_{avg} for each device as tabulated in Table. 4.8. The maximum error limit was selected to be 1.5 × δ_{avg} so that some extreme outliers are not penalized heavily. For angular DOF, an arbitrary value of 2° was chosen for δ_{avg}.

Table 4.8 δ_{avg} opted for dynamic analysis. $\delta_{max} = 1.5 \times \delta_{avg}$

Parameter	DOF	Tracker	Controller	HMD
δ_{avg}	P [mm]	49.8250	58.5	97.5
	O [deg]	2	2	2

This approach of fixing the δ_{avg} and δ_{max} was opted to have a general purpose results. The two hypothesis test performed will not provide direct information on the loss of tracking. Therefore, we propose another parameter “Percentage loss of tracking” to determine the efficiency of tracking in dynamic cases. It was calculated as the ratio between the number of data above the δ_{max} defined for MPET and the total number of sample data collected. This parameter along with the two hypothesis tests will provide sufficient information to infer statistics on the device’s performance.

Results

The findings of the standard procedural evaluation of the dynamic analysis of the VLTS are shown in Table 4.9. The average velocity of the trajectory during the test was 902.467 mm/s and a maximum of 1501.1 mm/s, which is limited to the recommendations of the standard. This velocity is approximately 15 times higher than the literature (60 mm/s). The tracking performance was highly poor which is clear from the CDF of the error (using equation (4.13) until δ_{max}) shown in Fig. 4.16 and 4.17. It was observed that only 18.7007 %, 44.3802 %, and 17.5275 % of data is less than 10 mm error and 61.5296%, 81.2617%, and 91.4467% of data is less than 2° of orientation error for tracker, controller, and HMD respectively.

The outcomes of the test hypothesis are tabulated in the table. 4.10 indicates that, except for HMD positional results, all other null assumptions have been refused.

The % loss of tracking computed, further confirms that all devices, with the exception of the HMD position tracking, are poorly spatially tracked.Since there was no substantial insight into the dynamic performance of VLTS by the standard procedure, we continue evaluating devices using the same trajectory but with a lower average velocity ($<$ 902.467 mm/s).

Table 4.9 ASTM, dynamic evaluation results: Statistics on pose error between the two tracking devices.

Device	Velocity [mm/s]	DOF	RMSE	E(50)	E(95)	E(99.7)	$\bar{e}$	s^2	e_{max}	%Loss
T	V_{avg} = 885.78	P [mm]	562.7750	33.3580	1164.0	3885.248	201.7611	276053.548	5125.9	33.7829
N: 6080	V_{max} = 1500.9	O [deg]	6.6700	1.5220	18.6890	27.5830	3.7284	30.5938	32.7830	
C	V_{avg} = 886.11	P [mm]	474.6540	11.8900	262.4120	3959.2	107.4543	213786.37	4792.4	11.3926
N: 61185	V_{max} = 1500.9	O [deg]	2.9890	0.6560	6.3880	17.9550	1.4778	6.7493	21.9330	
HMD	V_{avg} = 902.467	P [mm]	48.0560	32.8240	96.7290	117.9570	38.5553	823.0101	120.6060	0
N: 61185	V_{max} = 1501.1	O [deg]	1.0520	1.0300	1.6420	2.1480	0.9388	0.2263	3.7620	

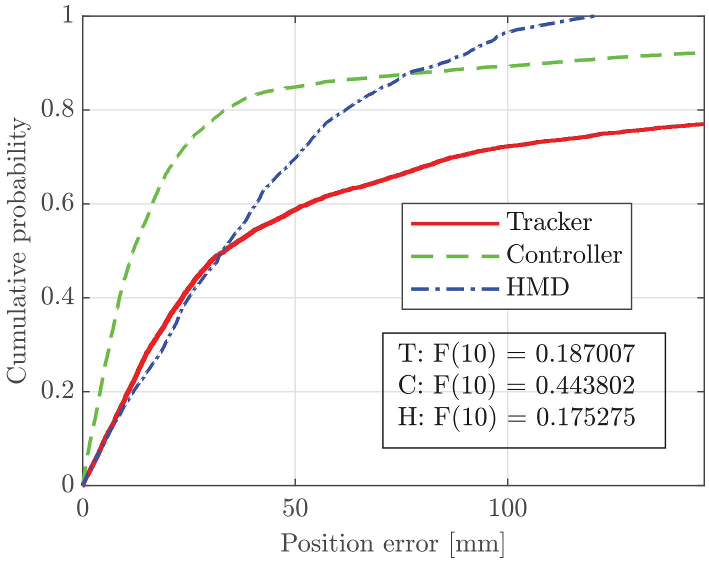


Figure 4.16 Dynamic standard evaluation: CDF of position error (up to 146.5 mm).

Table 4.10 ASTM, dynamic evaluation: Hypothesis test results.

Device	DOF	Z value	H_0	ME value	H_0
T	P	22.5484	*Rejected*	-0.9854	*Rejected*
	O	24.3661	*Rejected*	-0.9085	*Rejected*
C	P	8.1194	*Rejected*	-0.9817	*Rejected*
	O	-15.4156	Accepted	-0.8632	*Rejected*
HMD	P	-177.6190	Accepted	0.2126	Accepted
	O	-164.2445	Accepted	-0.2026	*Rejected*

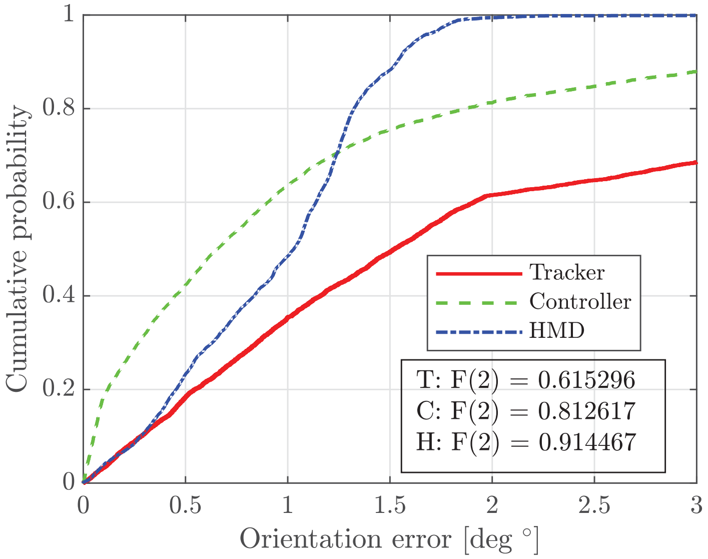


Figure 4.17 Dynamic standard evaluation: CDF of orientation error (up to 3°).

4.4.2 Determination of the velocity upper bound

Design of experiment

The aim of this test is to determine the upper bound velocity at which the monitoring is reliable and to analyze the output of VLTS devices below the upper bound limit. Here, the conditions for reliable tracking are laid down as follows:

- the acceptance of null hypothesis of the AET using equation (4.11)
- the percentage loss of tracking must be less than 0.5% (99.5% must be accepted)

Both of these conditions are considered as the necessary condition because less error and high loss of data is not favorable and the vice-versa is also not good.

The method is as follows: the same set of six ASTM test trajectories as seen in Fig. 4.15 will be used. Here, the average speed is decreased in steps and the results are checked for reliability at each stage until the upper bound of reliable tracking is established. Subsequently, a similar process is performed under this bound to determine whether the criteria of reliability are fulfilled. CDF, AET, MPET and % of tracking errors were used to infer statistics on the dynamic performance of the VLTS system. The same process has been replicated with all three devices.

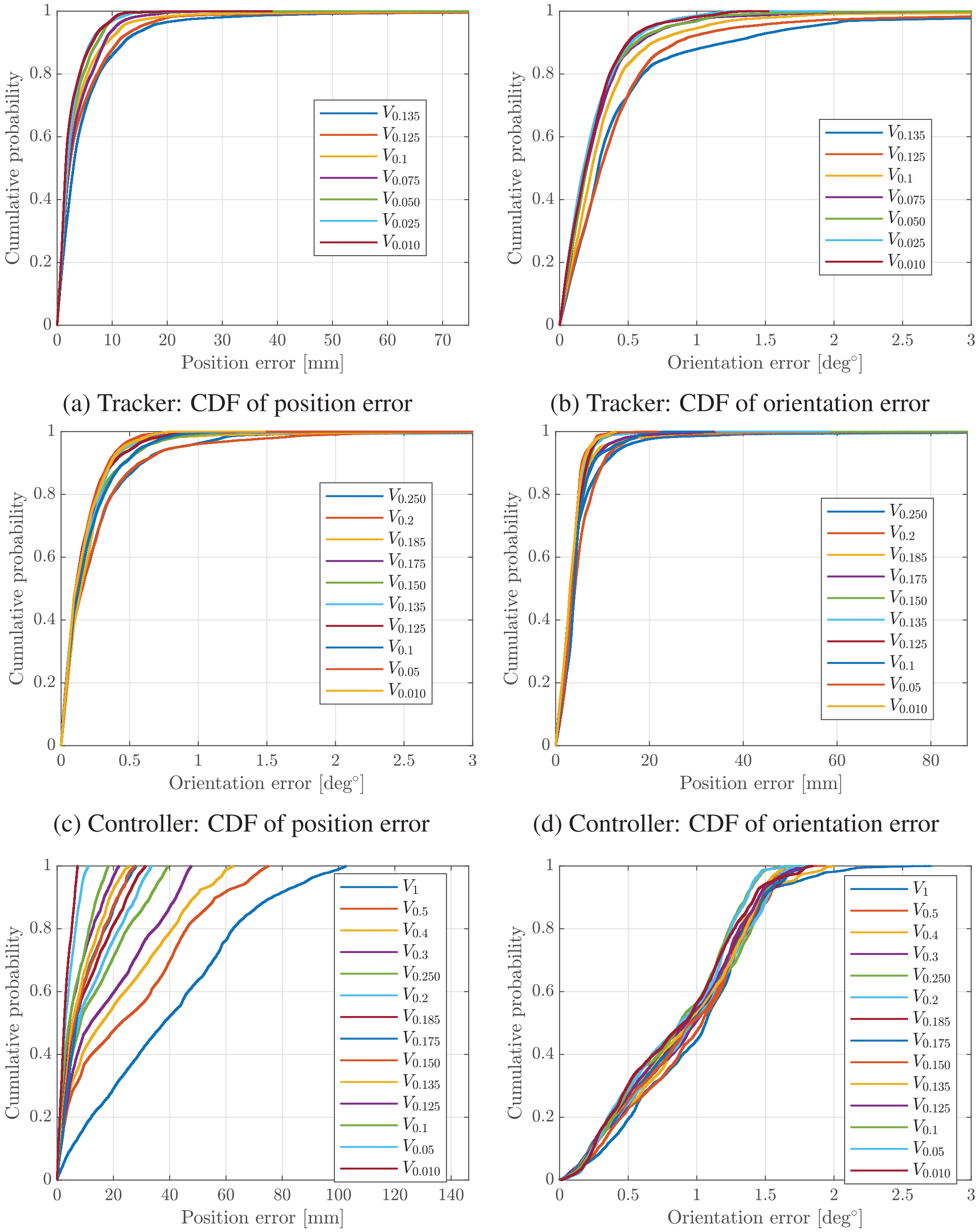
Results

Table 4.11, 4.12 and 4.13 summaries the performance analysis of the VLTS devices (Tracker, Controller and HMD correspondingly) under trajectories with varying velocities. The conditions of reliability (4.4.2) was checked at each velocity reduction step. Both the reliability conditions were found to be satisfied at 134.6 mm/s, 240.61 mm/s, and 853.46 mm/s for Tracker, Controller, and HMD respectively. HMD's superior results is due to many factors but the key factor is the number of photo-diodes in all the directions is high compared to the other two devices. CDF of the position and orientation error for the velocities lower than the previously mentioned reliable limit is shown from Fig. 4.18a to 4.18f. The Table 4.11, 4.12 and 4.13 displays others parameters such as percentile values, RMSE, $\bar{e}$, s^2, e_{max} for both position and orientation.

HMD's superior results are due to many factors, but the significant factor is that the number of photo-diodes is higher than the other two devices.

From the CDFs of position and orientation error shown in Fig. 4.18 for the trajectories less than the determined upper bound, we can observe that the curve signifies a good performance for all the devices i.e, within the bound the devices are able to localize themselves without losing more than 99.5% of data. Therefore from this analysis, we can infer the following on the upper bound of VLTS:

- The reliable upper bound velocity of the VLTS devices are 134.6 mm/s (Tracker), 240.61 mm/s (Controller), and 853.46 mm/s (HMD) with
  - the average error is less than the δ_{avg} (Refer Table. 4.8) and
  - the maximum error is less than the δ_{max} (= $1.5 \times \delta_{avg}$)



(a) Tracker: CDF of position error

(b) Tracker: CDF of orientation error

(c) Controller: CDF of position error

(d) Controller: CDF of orientation error

(e) HMD: CDF of position error

(f) HMD: CDF of orientation error

Figure 4.18 Velocity bound evaluation: CDF of position and orientation error up to δ_{max} for tests with velocities less than the upper bound

Table 4.11 Velocity bound evaluation results, Tracker: Statistics on pose error between the two tracking devices. P: Position, O: Orientation, N: Number of samples, $\bar{e}$: Average measurement error; s^2: variance of the measured error [mm^2, deg^2]; RMSE: root mean square error, Max.: maximum error, E(X) : estimated 'X' percentile deviation from the mean

Velocity [mm/s]	N	DOF	RMSE	e_{max}	E(50)	E(95)	E(99.7)	$\bar{e}$	s^2	H_0	% loss
$V_{avg} = 844.72$	6053	P [mm]	363.1460	3792.64	36.8620	487.5710	3268.269	124.7633	116328.1	*Rejected*	33.7519
$V_{max} = 1000.5$		O [deg]	7.4080	35.3360	1.6210	18.3830	33.7170	4.1969	37.2708	*Rejected*	
$V_{avg} = 484.01$	12826	P [mm]	343.1530	4815.7	12.8390	306.5680	3633.8	81.6145	111101.9	*Rejected*	15.6401
$V_{max} = 500.76$		O [deg]	4.8590	40.0370	0.9480	12.9760	23.3790	2.4616	17.5550	*Rejected*	
$V_{avg} = 247.59$	27082	P [mm]	298.7720	4899.3	5.8480	103.7330	3526.2	44.8902	87252.6	Accepted	7.4551
$V_{max} = 250.05$		O [deg]	3.001	38.6600	0.4850	5.4820	19.8450	1.2617	7.4129	Accepted	
$V_{avg} = 198.67$	33751	P [mm]	133.2400	2382.1	4.8110	43.6040	1512.7	20.8	17321.0	Accepted	2.2547
$V_{max} = 200.1$		O [deg]	2.6480	32.9840	0.3490	2.6060	27.4300	0.8909	6.2182	Accepted	
$V_{avg} = 183.94$	36430	P [mm]	75.377	2248.1	4.305	40.4790	402.639	14.3781	5475.1	Accepted	2.3936
$V_{max} = 185.13$		O [deg]	4.0520	37.2820	0.3660	6.3950	27.515	1.3769	14.5273	Accepted	
$V_{avg} = 174.10$	38649	P [mm]	31.9130	431.181	3.8560	31.9030	270.2170	10.776	902.3714	Accepted	2.3105
$V_{max} = 175.13$		O [deg]	1.7520	24.1030	0.356	2.3950	15.2680	0.7485	2.5103	Accepted	
$V_{avg} = 149.42$	45571	P [mm]	24.9910	911.09	3.059	26.7650	133.419	7.7316	564.787	Accepted	1.0116
$V_{max} = 150.08$		O [deg]	1.0470	11.446	0.3120	2.011	6.7270	0.5854	0.7527	Accepted	
$V_{avg} = 134.6$	50722	P [mm]	16.6200	945.0550	2.9390	16.7590	105.1940	5.8020	242.5590	Accepted	0.4921
$V_{max} = 135.05$		O [deg]	1.435	18.173	0.2810	1.7910	12.454	0.5838	1.7176	Accepted	
$V_{avg} = 124.65$	54706	P [mm]	13.8740	496.368	2.225	14.954	104.22	4.9429	167.7522	Accepted	0.4424
$V_{max} = 125.04$		O [deg]	1.5290	28.463	0.302	1.321	11.666	0.5243	1.7659	Accepted	
$V_{avg} = 99.84$	68817	P [mm]	8.8470	252.564	2.051	11.749	57.694	3.9	63.0563	Accepted	0.1410
$V_{max} = 100.07$		O [deg]	0.659	17.31	0.235	1.041	4.330	0.3495	0.3122	Accepted	
$V_{avg} = 74.94$	92038	P [mm]	5.2970	98.947	2.104	10.5810	22.451	3.5604	15.3867	Accepted	0.0109
$V_{max} = 75.05$		O [deg]	0.423	10.7250	0.197	0.794	2.293	0.2779	0.1017	Accepted	
$V_{avg} = 49.98$	139926	P [mm]	4.4770	197.92	1.915	8.839	18.897	2.955	11.3111	Accepted	0.0079
$V_{max} = 50.05$		O [deg]	0.3930	8.4180	0.1930	0.777	1.915	0.2653	0.0837	Accepted	
$V_{avg} = 24.99$	281257	P [mm]	3.796	20.513	1.818	8.323	14.685	2.7066	7.081	Accepted	0.0000
$V_{max} = 25.06$		O [deg]	0.324	1.9430	0.175	0.686	1.199	0.2369	0.0485	Accepted	
$V_{avg} = 9.99$	704694	P [mm]	3.721	39.132	1.527	8.336	15.438	2.5553	7.3185	Accepted	0.0000
$V_{max} = 10.05$		O [deg]	0.328	1.529	0.188	0.653	1.252	0.2413	0.0492	Accepted	

Table 4.12 Velocity bound evaluation results, Controller: Statistics on pose error between the two tracking devices. P: Position, O: Orientation, N: Number of samples, $\bar{e}$: Average measurement error; s^2: variance of the measured error; RMSE: root mean square error, Max.: maximum error, E(X) : estimated 'X' percentile deviation from the mean

Velocity [mm/s]	N	DOF	RMSE	e_{max}	E(50)	E(95)	E(99.7)	$\bar{e}$	s^2	% loss	H_0
$V_{avg} = 843.20$	6441	P [mm]	439.1060	4264.01	12.7170	962.2650	2756.7	136.7504	174140.5	*Rejected*	16.0534
$V_{max} = 1000.5$		O [deg]	4.8320	30.6040	1.0330	10.2410	29.6960	2.3345	17.9006	*Rejected*	
$V_{avg} = 484.45$	13029	P [mm]	249.9880	5375.2	5.4090	37.7980	2779.7	29.4976	61628.86	Accepted	2.1414
$V_{max} = 500.75$		O [deg]	1.0270	7.6570	0.1850	1.8140	6.8960	0.4616	0.8424	Accepted	
$V_{avg} = 389.17$	17235	P [mm]	101.3760	1825.6	4.8870	28.4040	1186.9	16.9618	9990.0	Accepted	1.4331
$V_{max} = 400.46$		O [deg]	0.7010	10.2600	0.2180	1.3490	4.2380	0.4041	0.3276	Accepted	
$V_{avg} = 295.02$	22746	P [mm]	46.1490	1547.3	4.9230	52.6810	366.7620	13.9838	1934.3	Accepted	3.4204
$V_{max} = 300.11$		O [deg]	0.8980	18.9910	0.2320	1.3670	4.2910	0.4070	0.6409	Accepted	
$V_{avg} = 247.61$	27296	P [mm]	27.3640	958.4530	3.2320	14.2800	110.0060	6.0842	711.8	Accepted	0.4103
$V_{max} = 250.05$		O [deg]	1.2860	26.7520	0.1520	0.8470	15.7730	0.3175	1.5533	Accepted	
$V_{avg} = 198.67$	33857	P [mm]	10.2180	287.3180	4.1360	12.8330	60.8330	5.5801	73.2805	Accepted	0.2068
$V_{max} = 200.10$		O [deg]	0.500	12.9080	0.1550	0.8450	2.8750	0.2677	0.1779	Accepted	
$V_{avg} = 183.68$	37258	P [mm]	8.2430	246.5230	3.4270	9.5630	39.2110	4.2109	50.2173	Accepted	0.1476
$V_{max} = 185.13$		O [deg]	0.3490	7.1770	0.1430	0.4810	2.4860	0.1987	0.0826	Accepted	
$V_{avg} = 174.11$	38985	P [mm]	7.4530	193.0960	3.4540	10.4670	321.0320	4.3891	36.2829	Accepted	0.1052
$V_{max} = 175.13$		O [deg]	0.2500	5.6870	0.1130	0.4630	1.4900	0.1668	0.0348	Accepted	
$V_{avg} = 149.29$	45844	P [mm]	5.3220	156.6310	3.5110	8.1610	16.7300	3.8603	13.4194	Accepted	0.0349
$V_{max} = 150.08$		O [deg]	0.2940	5.6380	0.1250	0.6100	1.3910	0.1941	0.0488	Accepted	
$V_{avg} = 134.49$	51269	P [mm]	4.6100	58.3530	3.4160	8.0790	19.9560	3.7402	7.2628	Accepted	0.0000
$V_{max} = 135.05$		O [deg]	0.2510	3.7700	0.1100	0.4710	1.2810	0.1703	0.0339	Accepted	
$V_{avg} = 124.65$	51466	P [mm]	4.5010	33.9030	3.4980	7.8780	13.4430	3.7365	6.2982	Accepted	0.0000
$V_{max} = 125.04$		O [deg]	0.2410	7.4890	0.1050	0.5310	1.0330	0.1632	0.0312	Accepted	
$V_{avg} = 99.78$	69266	P [mm]	5.7420	33.3940	3.9850	12.0900	20.2540	4.5338	12.4136	Accepted	0.0000
$V_{max} = 100.07$		O [deg]	0.2720	1.4240	0.1280	0.5910	1.0600	0.1910	0.0373	Accepted	
$V_{avg} = 49.97$	140318	P [mm]	4.0820	21.9090	3.4330	7.2730	11.9960	3.5062	4.3658	Accepted	0.0000
$V_{max} = 50.05$		O [deg]	0.2180	3.5390	0.1100	0.4320	0.7540	0.1560	0.0231	Accepted	
$V_{avg} = 9.99$	704464	P [mm]	4.0060	12.8560	3.0400	7.2370	12.3260	3.3431	4.8697	Accepted	0.0000
$V_{max} = 10.05$		O [deg]	0.2170	1.4910	0.1050	0.4520	0.7460	0.1581	0.0220	Accepted	

Table 4.13 Velocity bound evaluation results, HMD: Statistics on pose error. P: Position, O: Orientation, N: Number of samples, $\bar{e}$: Average measurement error; s^2: variance of the measured error; RMSE: root mean square error, Max.: maximum error, E(X) : estimated 'X' percentile deviation from the mean

Velocity [mm/s]	N	DOF	RMSE	e_{max}	E(50)	E(95)	E(99.7)	$\bar{e}$	s^2	% loss	H_0
V_{avg} = 853.46	7888	P [mm]	47.6370	102.6720	37.8510	88.3020	102.0160	39.5695	703.6343	Accepted	0.0000
V_{max} = 1000.5		O [deg]	1.0850	2.7120	1.0640	1.7290	2.3650	0.9781	0.2215	Accepted	
V_{avg} = 484.55	13688	P [mm]	33.9810	75.1200	22.6520	67.7220	74.4880	26.0019	478.6289	Accepted	0.0000
V_{max} = 500.75		O [deg]	1.0510	1.9390	1.0370	1.5700	1.7590	0.9477	0.2064	Accepted	
V_{avg} = 391.01	17545	P [mm]	28.1050	62.7540	15.8370	56.2230	62.1240	21.0501	346.8232	Accepted	0.0000
V_{max} = 400.47		O [deg]	1.0330	1.9990	0.9830	1.6630	1.9650	0.9195	0.2223	Accepted	
V_{avg} = 295.80	26865	P [mm]	23.3250	47.7350	11.9690	44.7850	47.5080	17.5017	237.7740	Accepted	0.0000
V_{max} = 300.11		O [deg]	1.0330	1.8490	0.9880	1.5910	1.8030	0.9207	0.2190	Accepted	
V_{avg} = 247.46	27665	P [mm]	17.6780	39.4800	8.2700	36.0390	39.2580	12.9919	143.7414	Accepted	0.0000
V_{max} = 250.04		O [deg]	1.0250	1.8300	0.9590	1.5470	1.7540	0.9079	0.2258	Accepted	
V_{avg} = 198.67	34865	P [mm]	15.1350	33.6590	7.5740	30.2380	33.3580	11.2635	102.2142	Accepted	0.0000
V_{max} = 200.10		O [deg]	1.0290	1.7390	0.9580	1.5740	1.7070	0.9157	0.2196	Accepted	
V_{avg} = 183.96	38166	P [mm]	13.4940	31.4470	6.3680	27.6240	31.1830	9.9602	82.8907	Accepted	0.0000
V_{max} = 185.13		O [deg]	1.0080	1.7500	0.9250	1.6200	1.7400	0.8933	0.2183	Accepted	
V_{avg} = 174.10	39540	P [mm]	12.0390	28.1090	6.7770	24.9000	27.9000	9.3159	58.1501	Accepted	0.0000
V_{max} = 175.13		O [deg]	1.0170	1.8360	0.9410	1.6160	1.7970	0.9013	0.2213	Accepted	
V_{avg} = 149.42	46063	P [mm]	11.7830	27.7610	6.0340	24.3840	27.2290	8.7538	62.2023	Accepted	0.0000
V_{max} = 150.08		O [deg]	1.0150	1.6490	0.9550	1.5350	1.6400	0.9062	0.2088	Accepted	
V_{avg} = 134.51	51273	P [mm]	10.4590	25.6890	5.3240	21.5690	25.2930	7.8093	48.3974	Accepted	0.0000
V_{max} = 135.05		O [deg]	0.9980	1.6490	0.9250	1.5230	1.6370	0.8863	0.2104	Accepted	
V_{avg} = 124.65	55259	P [mm]	9.0610	22.0280	4.3860	19.2820	21.7880	6.6694	37.6254	Accepted	0.0000
V_{max} = 125.04		O [deg]	0.9880	1.8010	0.9240	1.5560	1.7830	0.8752	0.2109	Accepted	
V_{avg} = 99.84	70447	P [mm]	7.9800	18.2900	4.2990	16.1770	18.1290	6.0265	27.3642	Accepted	0.0000
V_{max} = 100.07		O [deg]	0.9320	1.8000	0.8770	1.4250	1.6420	0.8331	0.1749	Accepted	
V_{avg} = 49.97	143515	P [mm]	4.6940	11.1090	2.9370	9.0610	10.9780	3.6951	8.3816	Accepted	0.0000
V_{max} = 50.05		O [deg]	0.9360	1.8080	0.8820	1.4140	1.6220	0.8343	0.1805	Accepted	
V_{avg} = 9.99	719274	P [mm]	3.5980	7.2880	2.4930	6.7290	7.2590	2.9391	4.3039	Accepted	0.0000
V_{max} = 10.05		O [deg]	0.9640	1.8450	0.9070	1.5440	1.8090	0.8504	0.2052	Accepted	

Fig. 4.19 portrays the overall percentage of loss for a range of velocities computed for all the trajectories (i.e., each error and its corresponding velocity was extracted from all the trajectory data. It can be clearly noticed that the above 300 mm/s the Tracker has a significant amount of error data ($> 10\%$) and the same can be observed for controller at velocity above 600 mm/s.

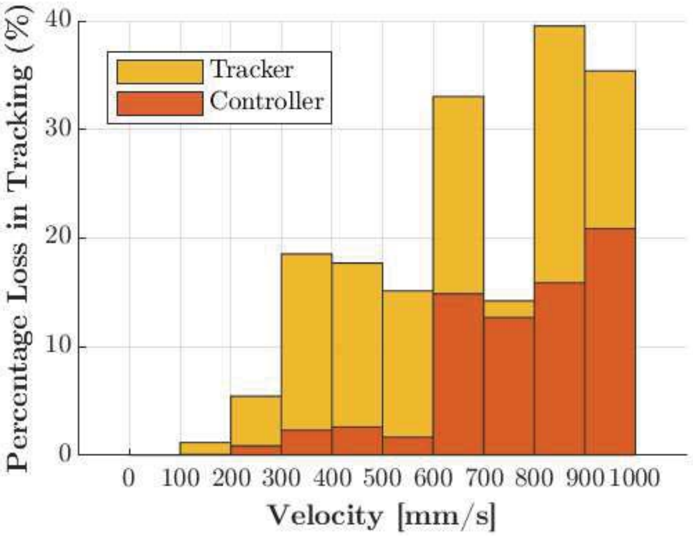


Figure 4.19 Overall percentage loss in tracking for tracker and controller (14 sets * 6 subsets, a total of 84 trajectories for tracker and controller) and percentage of loss was calculated over a range of 0 to 1000 mm/s in 10 steps).

4.5 Substantiation of the outcomes using human wrist trajectory tracking

The main objective of this experiment is to assure the findings of the studies performed in the section 4.4.2. The best approach will be to test the target application and, in essence, to check the usability of VLTS for that particular application. As the objective here is more general , the trajectory of human wrist motion was selected because of the likelihood of obtaining uninfluenced trajectories, not in the advantage of the performance of the device.

Design of experiment

Using the Optitrack MOCAP configured with 8 cameras, the ground truth trajectory profiles of generic human wrist motions were recorded. A wrist band was used to position the MOCAP reflective markers at the wrist joint centres, as can be seen in Fig. 4.20 fig:ZoppiHand (left). The subject was recommended to shift the joint extremes as much as possible to provide a more randomized and diverse trajectory, and no other details was presented. Eight trajectories were recorded with 3 spatial (subject instructed to move arbitrarily at various speeds) and 5 planar motions (subject instructed to position the palm when executing motions on a flat table and on a vertical board). The profile of the trajectory reported was then modeled with an average R^2 value of 0.64 using a smoothing spline curve fitting. The profile was converted to match inside the robot's prescribed workspace, as shown in Fig. 4.20 (right) (1 m x 0.55 m x 0.6 m cuboid, [78]. The workspace where the robot can travel linearly between any pose without reaching singularities or requiring joint space to change the manifold).

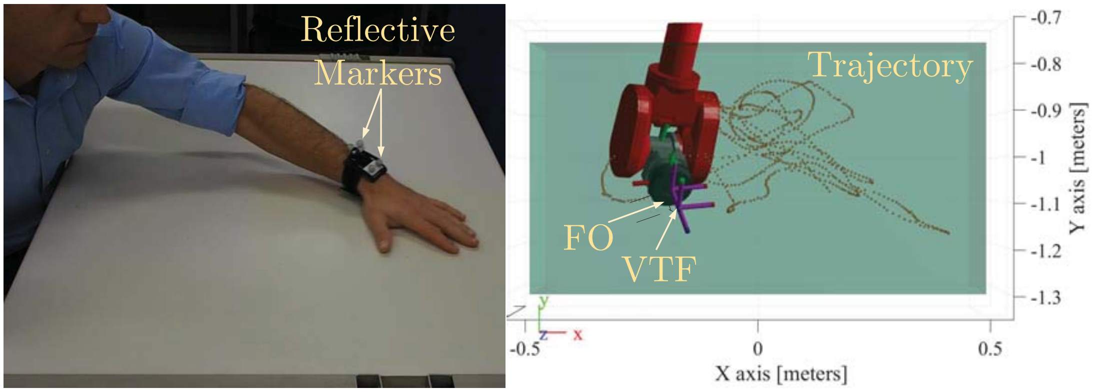


Figure 4.20 Left: Wrist centre of the upper arm is tracked by attaching a wrist band equipped with reflective markers. Right: the hand tracking trajectory fed to the C16, within the prescribed workspace. (FO is the offset distance of 45 millimetres where the mounting part is attached and VTF is the Vive tracker tool frame which is at distance of 19.6 millimetres from the top of the mount

The trajectory generation methodology for C16 is the difference in terms of configuration between this and the originally unsuccessful experiment with Optitrack MOCAP. The reference trajectories were manually produced according to specifications for all studies, but here the subject's wrist motion was recorded using Optitrack MOCAP. The primary aim of the Optitrack MOCAP is to collect from the subject the reference trajectories that are later fitted and provided as input to C16. C16 was then used to substantiate the dynamic efficiency of the VLTS. The same mount shown in Fig. 4.5 was used for tracker and controller. Due to practical difficulties of mounting two HMD, the single HMD mount shown in Fig. 4.4 (far right) was utilized. The centroid of the mounts were commanded to follow the prepared hand trajectory profiles. The error was calculated using the equation (4.13) for tracker and controller and equation (4.10) for HMD. Each trajectory's tracking data was examined for reliability and the corresponding velocity is verified against the upper bound reported by previous section 4.4.2.

Results

The results of tracking the fitted human wrist trajectories are displayed in Table. 4.14, 4.15, and 4.16 for Tracker, Controller, and HMD correspondingly. The non-reliable data is verified against the upper bound reported in section 4.4.2. For the HMD, as the upper bound was high, all the trajectories tracked were found reliable. For the Tracker, P4 trajectory was unreliable and the respectively average velocity was 166.67 mm/s ($>$ 134.6 mm/s). For the Controller, P3 was unreliable and the average velocity was 256.89 mm/s ($>$ 247.61 mm/s). Thus, all the trajectories resulted in unreliable tracking data was found to have average velocities higher than the upper bounds which in turn validates our obtained results on the dynamic performance bounds.

Table 4.14 Wrist trajectory evaluation results, Tracker: Statistics on pose error between the two tracking devices. P: Position, O: Orientation, N: Number of samples, $\bar{e}$: Average measurement error; s^2: variance of the measured error; RMSE: root mean square error, Max.: maximum error, E(X) : estimated 'X' percentile deviation from the mean

Traj./ N	Velocity [mm/s]	DOF	RMSE	e_{max}	E(50)	E(95)	E(99.7)	$\bar{e}$	$AE : H_0$	$MPE : H_0$	% loss
S1	$V_{avg} = 28.16$	P [mm]	3.7110	24.7980	2.5200	7.0830	17.1220	2.8081	Accepted	Accepted	0.0000
N: 35178	$V_{max} = 35.64$	O [deg]	0.2530	0.7810	0.2050	0.4250	0.7670	0.2179	Accepted	Accepted	
S2	$V_{avg} = 48.92$	P [mm]	3.3600	25.1800	1.9240	5.9430	17.6860	2.4470	Accepted	Accepted	0.0000
N: 63875	$V_{max} = 103.30$	O [deg]	0.1990	0.8750	0.1480	0.3600	0.7660	0.1619	Accepted	Accepted	
S3	$V_{avg} = 45.58$	P [mm]	3.0010	23.9530	1.6310	4.3860	18.9690	2.0075	Accepted	Accepted	0.0000
N: 55652	$V_{max} = 65.37$	O [deg]	0.2250	0.8120	0.1720	0.4180	0.6640	0.1880	Accepted	Accepted	
P1	$V_{avg} = 18.21$	P [mm]	4.8560	23.1790	2.5040	10.8960	20.5500	3.2866	Accepted	Accepted	0.0000
N: 23197	$V_{max} = 37.55$	O [deg]	0.3590	0.7620	0.3430	0.5260	0.6570	0.3308	Accepted	Accepted	
P2	$V_{avg} = 51.20$	P [mm]	3.1850	24.0930	1.4490	6.9270	17.9490	2.0550	Accepted	Accepted	0.0000
N: 70671	$V_{max} = 100.13$	O [deg]	0.2480	0.9410	0.1950	0.4220	0.8740	0.2105	Accepted	Accepted	
P3	$V_{avg} = 166.67$	P [mm]	37.31	220.26	13.6650	80.6990	209.4960	23.1034	Accepted	*Rejected*	6.4561
N: 11493	$V_{max} = 177.82$	O [deg]	0.9230	8.0430	0.6130	1.7830	4.4720	0.7150	Accepted	*Rejected*	
P4	$V_{avg} = 37.38$	P [mm]	6.0720	40.7720	3.4810	11.7670	28.2770	4.6310	Accepted	Accepted	0.0000
N: 47072	$V_{max} = 74.32$	O [deg]	0.4600	1.3340	0.1860	1.0140	1.2590	0.3214	Accepted	Accepted	
P5	$V_{avg} = 34.60$	P [mm]	3.3400	27.8030	1.1880	6.1030	20.5790	1.9056	Accepted	Accepted	0.0000
N: 153963	$V_{max} = 35.57$	O [deg]	0.2340	0.9600	0.1950	0.4000	0.7430	0.2041	Accepted	Accepted	

Table 4.15 Wrist trajectory evaluation results, Controller: Statistics on pose error between the two tracking devices. P: Position, O: Orientation, N: Number of samples, $\bar{e}$: Average measurement error; s^2: variance of the measured error; RMSE: root mean square error, Max.: maximum error, E(X) : estimated 'X' percentile deviation from the mean

Traj./ N	Velocity [mm/s]	DOF	RMSE	e_{max}	E(50)	E(95)	E(99.7)	$\bar{e}$	$AE : H_0$	$MPE : H_0$	% loss
S1	V_{avg} = 141.14	P [mm]	6.6130	24.2870	2.5970	20.8640	23.4870	4.1901	Accepted	Accepted	0.0000
N: 6441	V_{max} = 177.96	O [deg]	0.2080	1.1990	0.0470	0.4710	1.1700	0.1016	Accepted	Accepted	
S2	V_{avg} = 244.82	P [mm]	11.6700	89.4040	1.9400	22.9620	74.0030	5.2973	Accepted	*Rejected*	0.0079
N: 12643	V_{max} = 510.93	O [deg]	0.9040	5.9710	0.3110	1.4790	4.9540	0.5348	Accepted	*Rejected*	
S3	V_{avg} = 229.15	P [mm]	1.5620	4.4510	1.0800	3.0050	4.0030	1.2520	Accepted	Accepted	0.0000
N: 11015	V_{max} = 326.51	O [deg]	0.2090	0.5270	0.1680	0.3700	0.4840	0.1808	Accepted	Accepted	
P1	V_{avg} = 90.95	P [mm]	2.0910	3.6320	2.1080	2.8170	3.2500	2.0023	Accepted	Accepted	0.0000
N: 4561	V_{max} = 182.12	O [deg]	0.0720	0.3230	0.0470	0.1390	0.2920	0.0543	Accepted	Accepted	
P2	V_{avg} = 256.89	P [mm]	37.45	1014.2	4.3700	40.8050	283.9240	14.3968	Accepted	*Rejected*	3.3506
N: 14022	V_{max} = 500.61	O [deg]	1.0050	10.9920	0.2400	1.7550	6.5910	0.5308	Accepted	*Rejected*	
P3	V_{avg} = 139.93	P [mm]	1.3290	6.8520	0.9680	2.5090	4.0580	1.0826	Accepted	Accepted	0.0000
N: 14296	V_{max} = 177.83	O [deg]	0.0850	0.3660	0.0570	0.1780	0.2880	0.0662	Accepted	Accepted	
P4	V_{avg} = 187.59	P [mm]	1.0040	2.5660	0.8410	1.8030	2.2580	0.8541	Accepted	Accepted	0.0000
N: 9256	V_{max} = 364.21	O [deg]	0.2740	0.5410	0.2540	0.3960	0.5120	0.2632	Accepted	Accepted	

Table 4.16 Wrist Trajectory evaluation results, HMD: Statistics on pose error. P: Position, O: Orientation, N: Number of samples, $\bar{e}$: Average measurement error; s^2: variance of the measured error; RMSE: root mean square error, Max.: maximum error, E(X) : estimated 'X' percentile deviation from the mean

Traj./ N	Velocity [mm/s]	DOF	RMSE	e_{max}	E(50)	E(95)	E(99.7)	$\bar{e}$	$AE : H_0$	$MPE : H_0$	% loss
S1	V_{avg} = 140.99	P [mm]	7.5920	17.9790	4.9410	14.9110	17.1250	6.0573	Accepted	Accepted	0.0000
N: 6837	V_{max} = 177.91	O [deg]	0.3030	0.5800	0.2750	0.4870	0.5710	0.2721	Accepted	Accepted	
S2	V_{avg} = 244.92	P [mm]	18.2600	42.3410	14.0660	32.9210	41.3540	15.4773	Accepted	Accepted	0.0000
N: 12557	V_{max} = 508.68	O [deg]	0.2150	0.4690	0.1960	0.3290	0.4110	0.2016	Accepted	Accepted	
S3	V_{avg} = 230.09	P [mm]	12.9580	32.3630	8.6250	25.0180	31.1890	10.4078	Accepted	Accepted	0.0000
N: 10850	V_{max} = 326.55	O [deg]	0.1920	0.5220	0.1650	0.3100	0.4450	0.1760	Accepted	Accepted	
P1	V_{avg} = 90.63	P [mm]	6.1490	14.2970	4.4030	11.4700	13.9510	5.0988	Accepted	Accepted	0.0000
N: 4622	V_{max} = 182.39	O [deg]	0.1510	0.2350	0.1480	0.2070	0.2280	0.1468	Accepted	Accepted	
P2	V_{avg} = 256.55	P [mm]	15.5820	36.7460	11.0070	30.4170	34.9950	12.7158	Accepted	Accepted	0.0000
N: 14030	V_{max} = 500.61	O [deg]	0.1900	0.5090	0.1600	0.3140	0.4710	0.1690	Accepted	Accepted	
P3	V_{avg} = 139.84	P [mm]	9.4680	19.1780	7.1760	16.8790	18.6300	7.9549	Accepted	Accepted	0.0000
N: 14428	V_{max} = 177.88	O [deg]	0.2140	0.4590	0.1940	0.3240	0.4104	0.2014	Accepted	Accepted	
P4	V_{avg} = 187.88	P [mm]	9.9050	24.5660	7.8870	17.3580	23.4020	8.3184	Accepted	Accepted	0.0000
N: 9411	V_{max} = 363.62	O [deg]	0.1220	0.3350	0.1070	0.2000	0.2930	0.1113	Accepted	Accepted	
P5	V_{avg} = 172.89	P [mm]	7.8260	17.0940	6.5230	13.4010	15.7050	6.6888	Accepted	Accepted	0.0000
N: 30815	V_{max} = 177.66	O [deg]	0.1900	0.4050	0.1810	0.2760	0.3420	0.1801	Accepted	Accepted	

4.6 Procedure for calibration of the HTC Vive lighthouse tracking system using error estimation method

4.6.1 VLTS configuration recommendations

We recommend the following to be taken into account while configuration. These recommendations are made from observations and rectifying the failures while conducting this extensive study.

- Ensure that the sweep of the base station reaches the entire volume of work by specifying the workspace to track into a cube (play area * height). The best way to get this area completely enclosed is to have the base stations at the end of the play area diagonal. The most common error is to place the base stations at the midpoints of the cube's edges, this configuration would significantly decrease the tracking efficiency at the play area edges.

- Ensure the tracking devices are consistently visible to at least one of the base stations. It is better to make the devices visible to only one base station than visibility limited for both.

- Assign a unique role to the trackers and controller. This will ensure that the data received is not corrupted. For instance, if the role of "left-hand" is assigned to a tracker and a controller then the data from both these devices may be interchanged leading to swapping in visual and pose data of "left-hand" interpreted. This depends on the software application written to receive the data but we still suggest that the roles should be kept different from one another.

- The order from which each device is connected defines the SteamVR identification number. So if a series of experiments are carried out, the best way to maintain the Id's order is to connect the same way it was first paired. This is highly recommended for using VLTS as a Mocap solution.

4.6.2 Procedure to perform static performance evaluation for a custom application

Here, we explain how to use our results to evaluate VLTS for a custom application for an expected average and maximum error. This is an adaption from the hypothesis test proposed in [74]. To perform static analysis,

1. Select N $>$ 32 random pose within your workspace

2. Move the setup (robot or motion generator or just manually) with the device attached to these poses,
3. At each pose record the VLTS and the corresponding setup data (system to be tracked) (Note that if the data cannot be extracted from the system to be tracked, any other reference system can be used),
4. Calculate the error using the equation (4.10),
5. The above steps can be skipped if the error data for more than 32 pose exists,
6. Find $\bar{e}$ and e_{max}
7. Fix δ_{avg} and δ_{max} as per the average and maximum error the system is required to perform, (Note that if the interest is only on one of the values, the other value can be taken from the Table. 4.5)
8. Extract the s^2 for the specific device from the Table. 4.5,
9. Substitute the above values in the equation (4.11) and (4.12) to determine whether the null hypothesis of average error and maximum permissible error test is accepted or rejected,
10. If both the hypothesis is accepted, then we can concur that VLTS is suitable for the custom application and also the base stations are setup properly.
11. If any one of the hypothesis is rejected, then reconfigure the setup as per the recommendations mentioned later and repeat all the steps from the beginning until the hypothesis are accepted.

If the static analysis fails even after having the iterating with various configurations, then we can infer that VLTS is not the recommended system for the custom application to have an δ_{avg} and e_{max}. The next option is to increase the value of δ_{avg} and e_{max}, if the application can permit the increase in error and repeat the procedure.

4.6.3 Procedure to perform dynamic performance evaluation for a custom application

Here, we describe the procedure to conduct dynamic performance analysis for a custom application for an expected average error and percentage loss of data (Refer section 4.4.2). To perform dynamic analysis,

1. Select a trajectory based on the custom application with more than 32 samples (N>32 within your workspace) and an average velocity,
2. Perform the trajectory or motion with the tracking device attached,
3. At each pose record the VLTS and the corresponding setup data (system to be tracked),
4. Calculate the error using the equation (4.13) and (4.14).
5. Again, the above steps can be skipped if the error data exists for more than 32 samples,
6. Find $\bar{e}$,
7. Fix δ_{avg} and δ_{max} as per the average and maximum error the system is required to perform,
8. Calculate the percentage of loss in tracking (4.4.1)
9. Extract the s^2 for the specific device from the Table. 4.11 or 4.12 or 4.13.
10. Substitute the above values in the equation (4.11),
11. Check whether the reliability condition is accepted or rejected (Refer section 4.4.2),
12. If the reliability conditions are accepted, then we can concur that VLTS is suitable for the custom application with error δ_{avg} and δ_{max} ,
13. If the reliability condition fails, check whether the average velocity of the trajectory is within the upper bound specified in section 4.4.2. If the velocity is above the upper bound, we suggest to reduce the average velocity and repeat the procedure.

If the dynamic analysis fails repeatedly after reducing the velocity, then we can infer that the VLTS is not suitable for the custom application to have δ_{avg} and δ_{max}. The next option is to increase the value of δ_{avg} and e_{max}, if the application can allow increase in error and repeat the procedure.

4.6.4 Validating the error estimation method of VLTS for motion compensation in SP7

Through the sequence of experiments, a calibration technique and the limit of dynamic motion of VLTS were established. A straightforward experiment to assess the motion compensation of the SP7 platform was developed with this established knowledge. The motion cues were designed to be lower than the limit identified and the VLTS environment was calibrated in compliance to the procedure in the preceding section 4.6. The experiment was performed with 5 separate test subjects with one specific objective to determine the discrepancy with and without the motion compensation algorithm. The study participants were able to validate our predicted outcomes. Also, during the experiments, no major drifts in tracking were observed. Thus, by using our series of standardized statistical error estimation techniques, we were able to resolve the problem of SP7 motion compensation.

Chapter 5

Strategy for tuning the parameters of the motion cueing algorithm

The intricacy of tuning the parameters of the MCA and the significance of the effect of the proper tuning on the motion simulation experience was evident from our extensive testing period on SP7. Literature studies such as [87], [88], and[89] further substantiate the significance of the proper tuning approach to obtain a holistic experience of motion simulation. In order to have the right tuning strategy, one needs to have proper feedback from the user on the platform. The problem of relying on subjective feedback to adjust the parameters of the MCA is threefold.

1. The result applies only to that specific subject and to a specific scenario. The result cannot be transferred to another subject or scenario.

2. The number of parameters to be tuned for the classical MCA is 80. The number of experiments to be tested to tune a set of 9 parameters for one filter would be approximately 4 to 5. This adds an additional order of complexity to the tuning.

3. Apart from the complication of tuning all parameters, during this process, the subject concerned will experience simulation sickness which either affects the quality of the tuning or resets the whole tuning for all parameters.

The approach used to resolve the dependence of the parameter on the result of the subjective evaluation is to carry out an objective assessment of the same and to remove the human factor in the tuning process. After numerous attempts, such as [90],[91] have been made to produce an objective score over the subjective results, a completely objective approach has been proposed in [92] to test the perception of motion in a flight simulator and tune the MCA parameters based on that evaluation. Study [93] presents the implementation of the latter methodology for vehicle driving simulation. This approach is carried out by using data from the actual flight and the actual car as a reference motion and by reducing the error through optimization technique. Although this method is valid, it cannot be used in the case of SP7 since we do not have access to actual flight data which will increase the cost of tuning for a particular flight model. It is not practically possible to obtain the actual data for each scenario to be created with SP7 in the future. As a result, an attempt to conduct an objective evaluation using the HTC Vive lighthouse system trackers was made and presented in the following section. The trackers of the HTC Vive lighthouse system was made and presented in the following section.

5.1 Objective assessment of motion perception in SP7

In collaboration with Prof. Marco Testa (Department of Neuroscience, Rehabilitation, Ophthalmology, Genetics, and Maternal and Child Sciences (DINOGMI), University of Genoa), we explore a novel methodology to quantitatively measure the motion perception of a motion simulator. As mentioned in section 3.2, the subjective questionnaire-based measures are highly subjective and don't take into account all those complex interplays of multisensorial inputs that lead to balance and movement control [94]. For instance, maintaining equilibrium during a simple static standing task requires our body to resist the effect of gravity and a various range of perturbations like external forces, slippery or tilted surfaces, and moving visual scenes that can potentially happen simultaneously [95]. Balance control can be modeled as a closed-loop feedback system that registers body oscillations generate a proportional corrective motor action modifying the time course of sway and successively registering it again through the integration of multiple sensory stimuli. In particular visual, vestibular, and somatosensory systems are the three major sensory components of postural control [96]. To quantify each component's contribution to balance, a possible experimental protocol is to perturb the whole system using specific stimuli for one particular sensor modality and successively record the correlated responses with objective measures like angular or linear displacements, velocities, or accelerations of body segments, center-of-mass angular or linear motion, the center of pressure, joint torques and muscle activation through electromyogram [97]. If the information across all stimuli is appropriately coherent with each other, the internal estimation error of body orientation is reduced and consequently, this reduced variability is translated into reduced sway values [95]. Relying on these considerations, we speculate that monitoring the relative body segments' motions inside a virtual simulator can be an objective alternative to commonly used questionnaires and visual evaluation of the simulated experience.

Hence, we explore a novel methodology to quantitatively measure the accuracy of a motion simulator in giving realistic visual and motion stimuli, recording data from subjects' head and torso using Vive trackers and a head-mounted display. We have used two Vive trackers and the HMD to record motion data of the simulator, jury's sternum, and jury's head. A correlated parameter could be formulated to measure the effectiveness of motion perception using the collected data. The Vive lighthouse tracking system was chosen as the motion tracking system because SP7 already uses the Vive head mounted display for visual input and adding other tracking systems would not be economical.

5.1.1 Design of experiments

The initial discussion with the team led to the following design hypothesis and constraints for a prompt set of data to form the evaluation technique.

1. The visual cues should have minimum distractions
2. The motion cues must have almost null bumps and zero breaks in the presence
3. No audio input to avoid extra cues
4. Three sets scenarios with motion along three axes of the human body (sagittal plane - roll, horizontal plane - pitch, frontal plane - yaw)

With these constraints above, a scenario of a tunnel riding experience was designed using the custom cue generation workflow as described in section 2.1.1. The illustration mentioned in the section 2.1.1 was the tunnel scenario.

During the experimental session, both objective and subjective data will be recorded from the subject. The subjective response shall be recorded to validate the objective reasoning. Objective responses are recorded using Vive trackers, two trackers are used for recording the motion of the simulator platform and the subject's Torso motion. Data from the HMD is also recorded for tracking head motion. Vive trackers are setup through Unreal Engine along with the custom cue generation workflow. UE's “motion component" is used to assess and extract data from the trackers. Data of the trackers are also processed in UE by creating C++ plugins to handle frame transformations. At the end of each experimental session, in addition to the questionnaire response; the following tracker data will be computed and recorded

1. Motion of torso of subject with respect to SP7 platform (end effector position of the parallel manipulator)
2. Motion of head of subject with respect to SP7 platform
3. Motion of SP7 platform with respect to the base for reference motion.

Frame transformation: Two softwares (PTC Creo Parametric 2.0 and Unreal Engine 4.21.1) and two hardware (HTC Vive Tracker and SP7 motion simulator) were used for developing the experiments. It is of utmost importance to ensure proper reference frame transformations to properly integrate them.

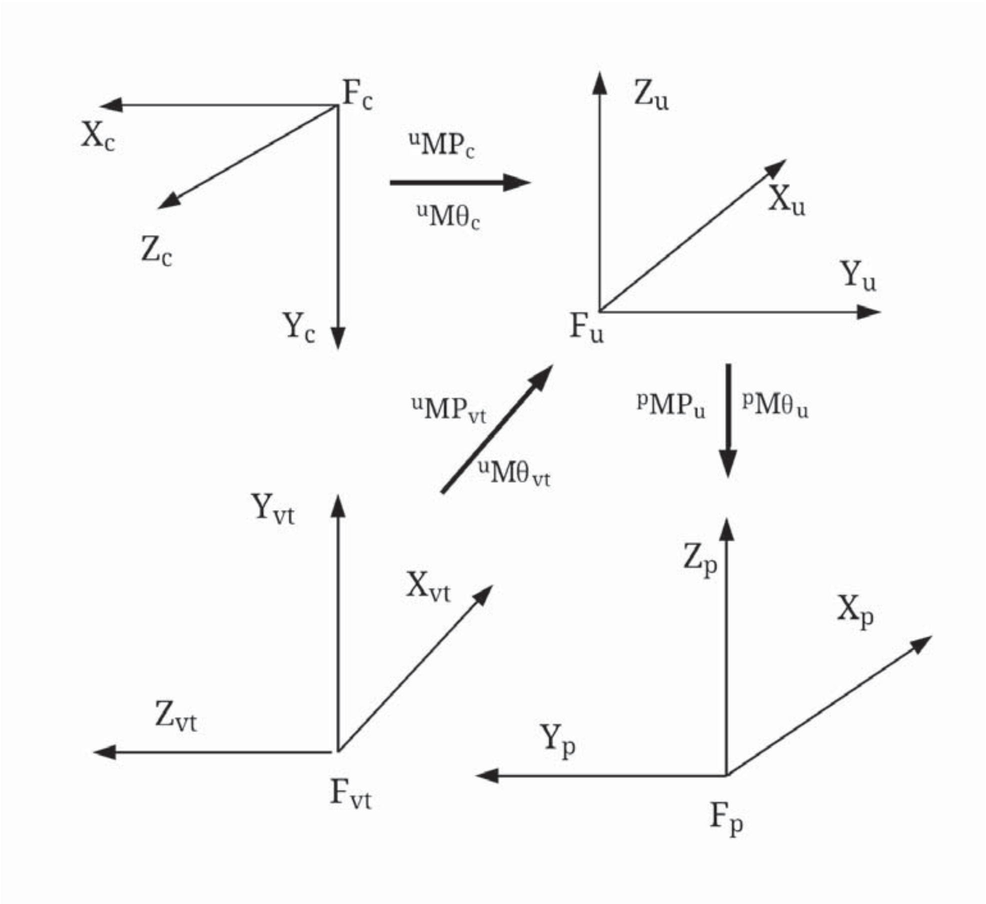


Figure 5.1 Representation of frames. Top left: PTC Creo Parametric reference frame F_c (Positive X_c is to the left, positive Y_c is facing down and positive Z_c is backwards) ; top right: Unreal Engine reference frame F_u; bottom left: Vive trackers reference frame F_{vt}; bottom right reference frame F_p

Frames F_c, F_{vt} and F_p follow right hand rule with configurations shown in Fig. 5.1 with positive rotation along anti-clockwise direction. Whereas F_u follow left hand rule with unique rotation directions (Rotation about X and y axis are in anti-clockwise direction and Rotation about z axis is in clockwise direction). Due to the uniqueness of the F_u, first right hand frame must be converted to left hand or vice versa based on the need, followed by necessary rotations to align the axes and finally negate necessary rotations. The Euler singularities were handled using [98].

For the ease of implementation in C++, the output of transformations explained above was converted to simple mapping matrices. Mapping the position and orientation of CAD model and data from Creo analysis in F_c to Unreal engine reference frame F_u is shown in (5.1) and (5.2) correspondingly,

$$^{u}P = {}^{u}MP_c *^{c} P \tag{5.1}$$

$$^{u}\theta = {}^{u}M\theta_c *^{c} \theta \tag{5.2}$$

where,

$$^{u}P = \begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix} \qquad {}^{c}P = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

$$^{u}\theta = \begin{bmatrix} \theta x_u \\ \theta y_u \\ \theta z_u \end{bmatrix} \qquad {}^{c}\theta = \begin{bmatrix} \theta x_c \\ \theta y_c \\ \theta z_c \end{bmatrix}$$

$$^{u}MP_c = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} {}^{u}M\theta_c = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

Similarly, mapping the position and orientation of Unreal Engine data in F_u to the SP7 motion simulator F_p is shown in (5.3) and (5.4) correspondingly,

$$^{p}P = {}^{p}MP_u *^{u} P \tag{5.3}$$

$$^{p}\theta = {}^{p}M\theta_u *^{u} \theta \tag{5.4}$$

where,

$$^{p}P = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} \qquad {}^{u}P = \begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix}$$

$$^{p}\theta = \begin{bmatrix} \theta x_p \\ \theta y_p \\ \theta z_p \end{bmatrix} \qquad {}^{u}\theta = \begin{bmatrix} \theta x_u \\ \theta y_u \\ \theta z_u \end{bmatrix}$$

$$^{p}MP_u = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^{p}M\theta_u = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Transformation from F_{vt} to F_u is done by Unreal Engine's motion controller component. But the motion data must be recorded in the frame of platform F_p as the juries will be seated on the SP7 motion simulator, so this transformation between F_u and F_p is done by mapping matrices shown in (5.3) and (5.4).

In order to validate our objective responses by comparing with the subjective evaluation, same procedure described in section 3.2.1 was used here with the same questionnaire in section 3.2.2. Fig. 5.2 shows subject prepared and seated on the SP7 simulator.

A post-processing script based on python and dash has been developed to process data in real-time and stream it to the localhost in a web browser. This was done in order to enable the real-time monitoring of the results, so that the visual observation could be correlated with the data processed from the tracker. Fig. 5.3 displays a sample streaming data app snippet running on Google Chrome.

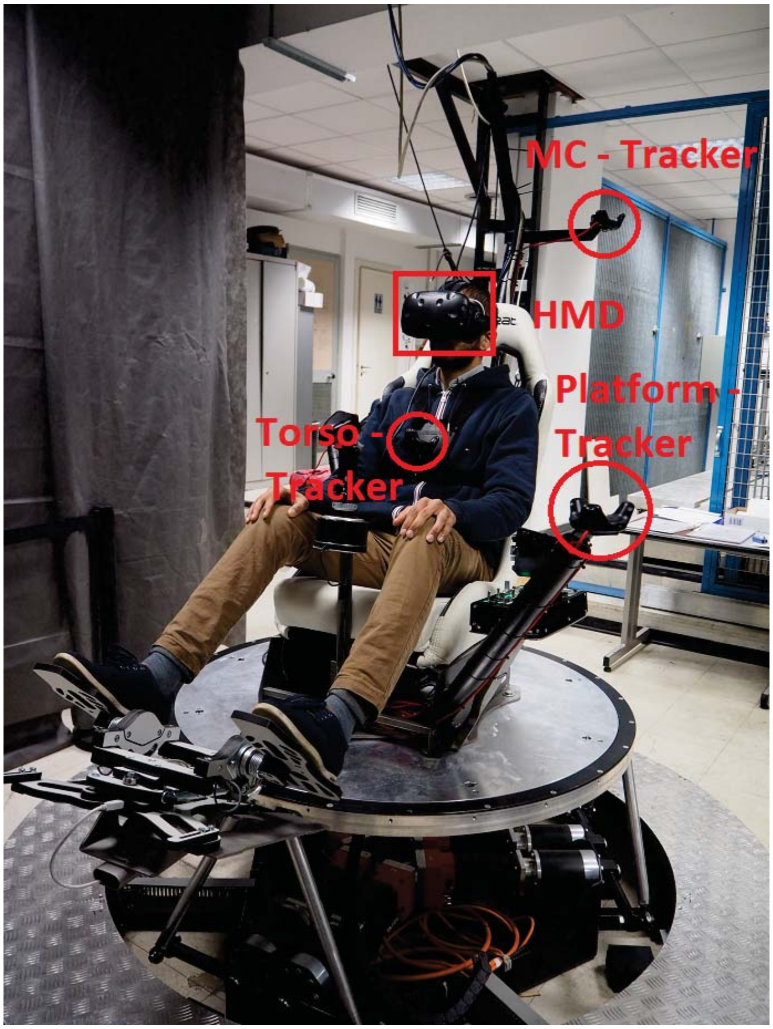
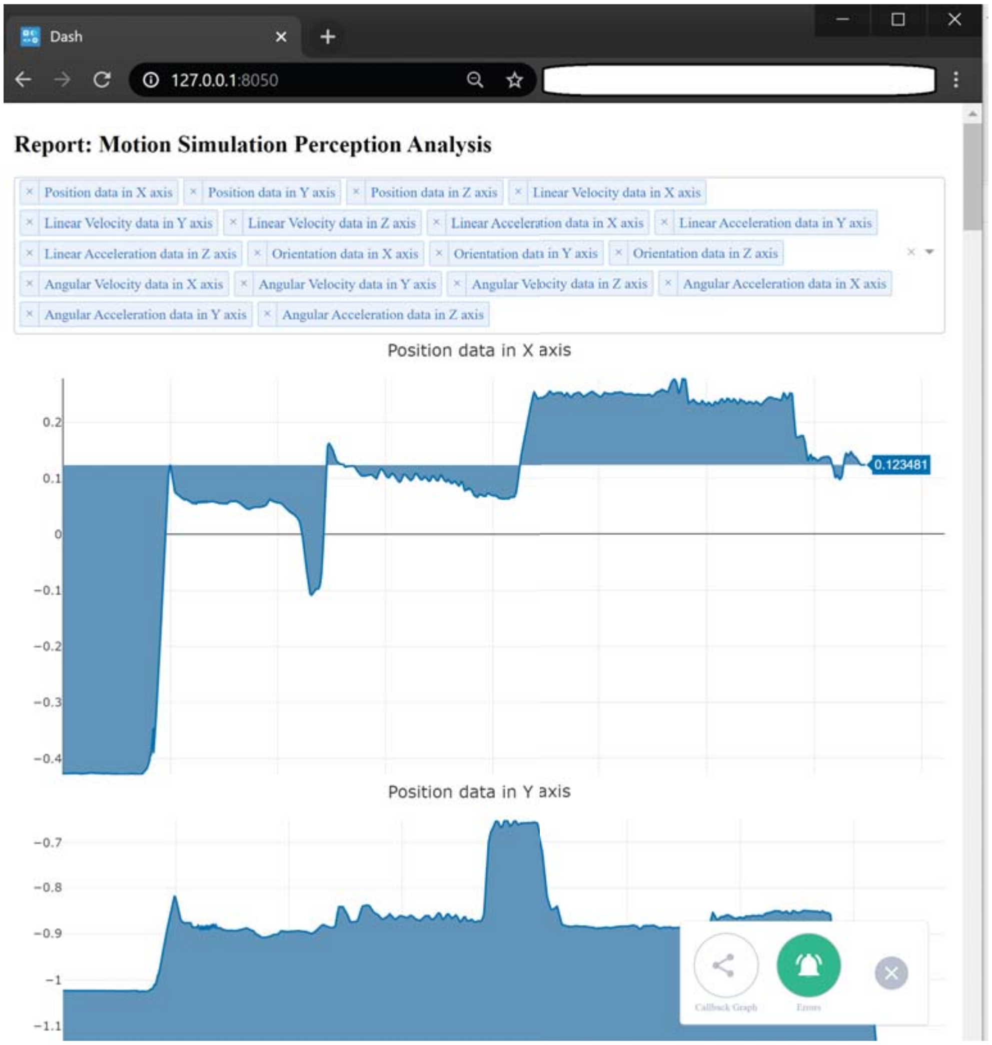


Figure 5.2 SP7 simulator - subject prepared and seated. Three Vive trackers are used; TR_P: platform motion tracking, TR_M: simulator motion compensation, TR_T: subject's Torso motion tracking (Camera strap used)).

5.1.2 Result analysis

Before evaluating with a large jury pool, three jury members were invited to participate in order to have a proof of the concept of the proposed method. Out of three, one jury have participated in the previous subjective evaluation of SP7 and the other two had no prior experience with a motion simulator. The experiments were conducted in the same way as the subjective evaluation, with the addition of the data recorded by the tracker. Fig. 5.4 shows a picture taken during an experiment session with a tunnel scenario. The recorded tracker data of the subject head with respect to the torso was then post-processed to filter instances



Report: Motion Simulation Perception Analysis

× Position data in X axis × Position data in Y axis × Position data in Z axis × Linear Velocity data in X axis × Linear Velocity data in Y axis × Linear Velocity data in Z axis × Linear Acceleration data in X axis × Linear Acceleration data in Y axis × Linear Acceleration data in Z axis × Orientation data in X axis × Orientation data in Y axis × Orientation data in Z axis × Angular Velocity data in X axis × Angular Velocity data in Y axis × Angular Velocity data in Z axis × Angular Acceleration data in X axis × Angular Acceleration data in Y axis × Angular Acceleration data in Z axis

Figure 5.3 Real-time post-processing of tracker data using Python-Dash

where there was a significant spike in motion. In Fig. 5.5, a sample dataset is shown. It was difficult to correlate the spikes with the subjective evaluation questionnaire. The vibration of the platform, along with the random movements of the user, further impairs the chance of obtaining a correlation parameter. After an intensive analysis of the data, we abandoned our objective evaluation process.

Although the objective of the study was not achieved, a significant observation was made while conducting the experiment and discussing specific scenario cases with the jury. While conducting a long session of experiments to tune the parameters of the MCA, the feedback given by the subject and the tuning parameters change does not necessarily have a positive effect on the perception of motion. For example, if the jury’s feedback is "the motion at the

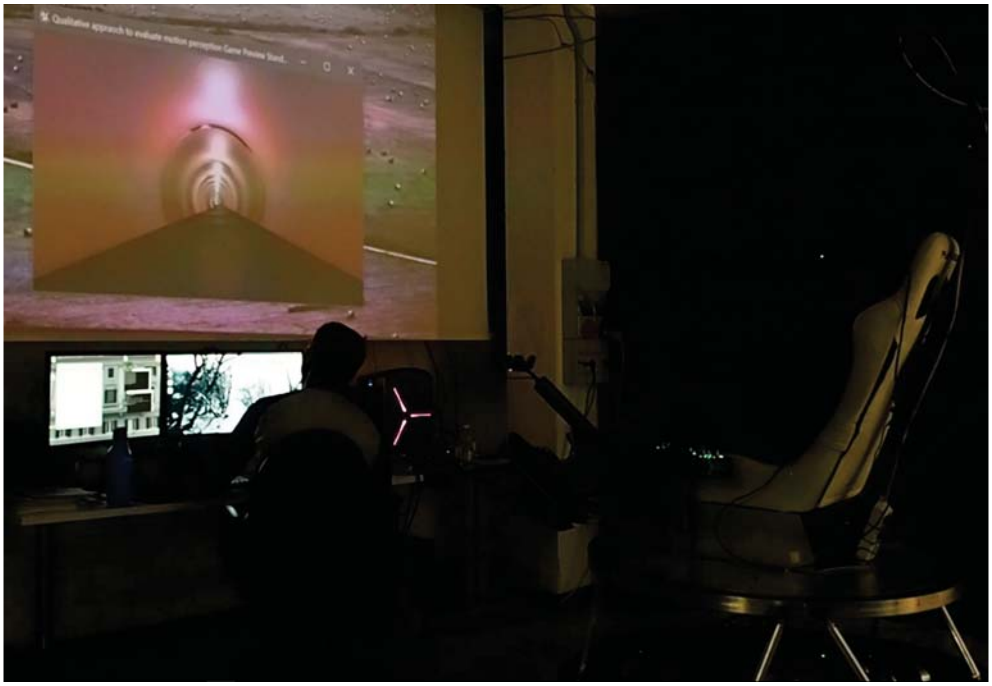

Qualitative approach to evaluate motion perception Game Preview Stand...

Figure 5.4 A picture taken during an experimentation session with tunnel scenario. The picture is dark because for VLTS to work better, the environment must be dark

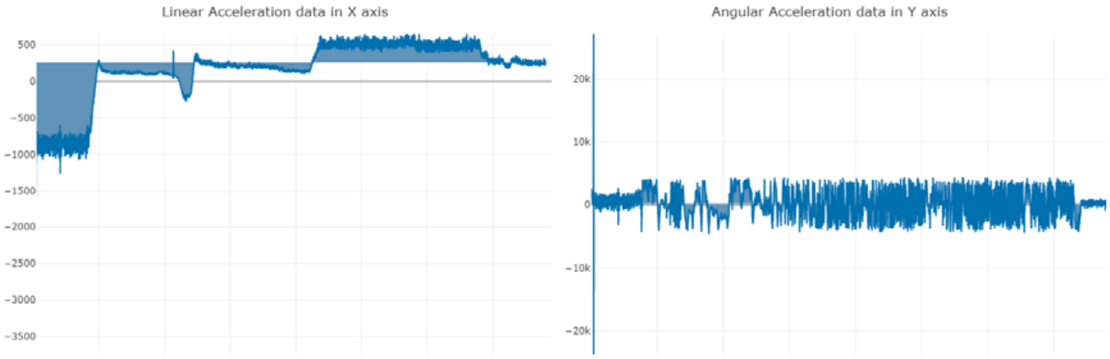


Figure 5.5 Tracker data plots for one of the tunnel scenario

turn wasn't properly felt," what's the right parameter to modify in such a way that the jury feels the right move? This is due to the lack of a direct correlation between the feedback given to the subject and the modification of the parameters. Therefore, to bridge this gap, an abstraction layer is required. The following section describes the attempt to develop such an abstraction layer.

5.2 An abstraction layer correlating the subjective feedback of the subject and the parameters of MCA

5.2.1 Experiment design

With the experience of conducting more than 60 sessions of empirical tuning and subjective evaluations of the motion perception in SP7 with multiple juries, few assumptions were taken to develop an abstraction layer. The assumptions are as follows:

- Scaling parameters can be associated with the intensity of motion.
- Saturation parameters can be associated with the comfort level during the experience.
- Filter parameters can be associated with the truthfulness of the perceived motion.

The three parameters mentioned above can be expressed in a cube (see Fig. 5.6) where the corners show the eight test conditions in a three-factor configuration of the experiment with two levels. In a full factorial design a total of 2×3 = 8 test conditions are involved which yield eight distinct results. Optimal tuning parameters could possibly be achieved by evaluating the eight test conditions.

On the basis of these assumptions, a graphical user interface has been developed to allow MCA parameters to be tuned on the fly. Fig. 5.7 shows a snippet of the interface. Basically, this interface modifies the parameters in the “config.yaml" file. The change of value in this file is reflected in the next frame input arriving from the VR scenario generator. Being a .yaml file, there is no need for compilation thus allowing on-the-fly tuning.

Thus, the new procedure followed for the tuning of MCA parameters are as follows:

1. In the scenario, select eight key distinctive motion segments. Each segment should refer to different motion in terms of the degree of freedom, initial acceleration, and duration.
2. After each segment, the subject is questioned on the intensity, comfort, and truthfulness of the motion perception on a scale of $0 \implies 100$.
3. Based on the response, the magnitude of the parameter is modified corresponding to the axis of motion.
4. Repeat steps 2 and 3 for each segment for a maximum of 3 times or before if the user is satisfied. The constraint of 3 times is because the subject will get accustomed to the scenario leading to no significant improvement in the tuning process.

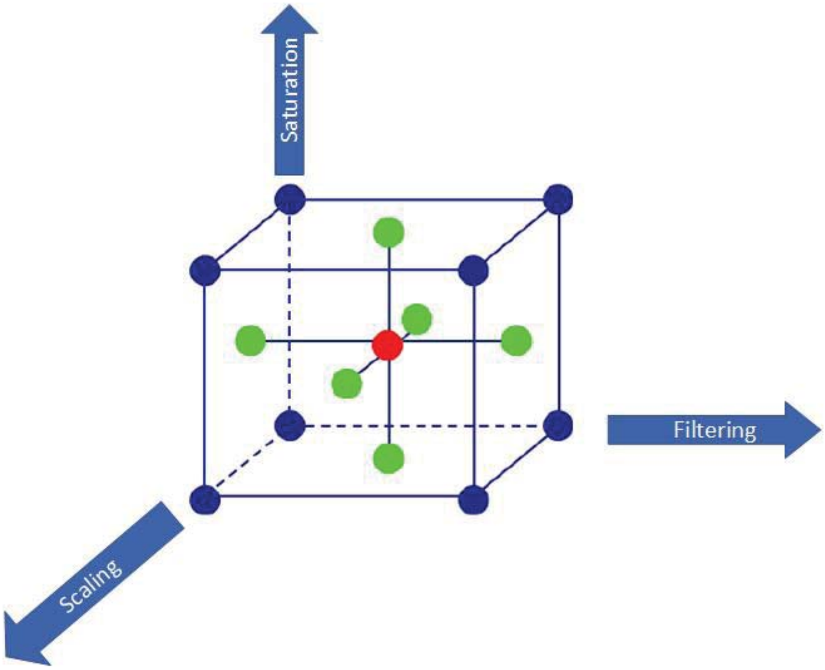
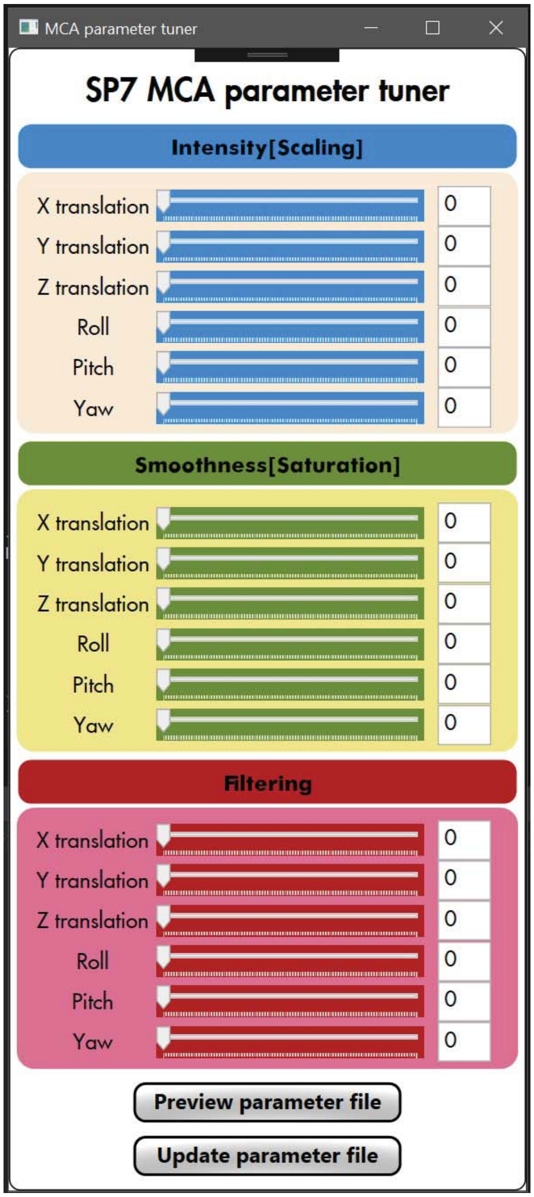


Figure 5.6 The three parameters are represented in the three-factor cube design of the experiment approach.

5. Upon completion of Step 4, pause for a minimum of 10 minutes to disengage the subject from the virtual world and relive any effects of motion sickness.
6. Full run of the scenario without a pause.
7. Filling the subjective assessment questionnaire described in section 3.2.2.

5.2.2 Result analysis

First, a small pool jury of three members was invited to test these assumptions. The average total score was 83% compared to 61% for the initial subjective assessment in chapter 2. The same experiment was repeated with 10 other members of the jury and the average total score for the subjective evaluation was 76%. This approach has yielded better results, but this abstraction has not yet resolved the dependence of each user on each scenario. The positive highlights of this approach are the reduction of almost one-fifth of the time taken to tune the 80 MCA parameters empirically. Also, the effect of motion sickness suffered by the subject while tuning is reduced significantly.



SP7 MCA parameter tuner

Intensity[Scaling]

X translation	0
Y translation	0
Z translation	0
Roll	0
Pitch	0
Yaw	0

Smoothness[Saturation]

X translation	0
Y translation	0
Z translation	0
Roll	0
Pitch	0
Yaw	0

Filtering

X translation	0
Y translation	0
Z translation	0
Roll	0
Pitch	0
Yaw	0

Figure 5.7 A snippet of the abstraction layer GUI for on-the-fly tuning.

Chapter 6

Conclusions

6.1 Summary

In this thesis, a motion generation system for building a virtual reality scenario application with custom visual, audio, and motion cues was developed using software such as PTC Creo and Unreal Engine. The workflow involved in the latter system was also illustrated with a sample scenario. In order to interface the SP7 motion simulator with a third-party virtual reality game, a custom script for extracting metadata from the game engine was introduced as part of the motion generation system. The motion output from the motion generation system was processed by a motion cueing algorithm to create an effective illusion while limiting the motion within the robotic manipulator workspace. Upon integration of the SP7 with the motion planning and motion generation system, a systematic subjective evaluation of the quality of motion perception was carried out with a jury pool of thirty members. The result was a mixed response with an overall average satisfaction score of 61%. During the evaluations, four key issues involved in the current setup were identified. The first issue was a technical issue with the hardware, an overshoot of the motor axis fixed by the manufacturer. The second was the turn-around bumps that were fixed by introducing a sigmoidal function to the output of the motion planning system. The remaining two issues were complicated and explored in detail. The third issue was the motion compensation for the movement of the platform in the virtual reality display. The default solution available in the literature was implemented, but there was still a significant amount of discrepancy in compensating for the motion. The source was found to be poor calibration of the Vive lighthouse tracking system after intensive testing. This calibration issue was resolved by evaluating the tracking system for its pose tracking capabilities with an industrial serial robot. In doing so, a benchmark for dynamic pose tracking performance evaluation has been developed. This benchmark allowed us to have an optimal calibrated environment for the Vive lighthouse tracking system using our error estimation technique. The calibrated environment was then tested and confirmed for resolution of the motion compensation problem. The final issue was the tuning of the parameters of the motion cueing algorithm. For each scenario, a total of 80 parameters must be adjusted for one user. As a result, objective feedback from the experiment was explored and no conclusive results could be derived from it. However, the observations made during these experiments led to the development of a strategy for tuning with the abstraction layer concept. The tuning process with this concept reduced the parameters from 80 to 9 and also reduced the tuning time by one fifth. The satisfaction score from the assessment of this approach was 76%. This approach reduces the complexity of tuning the motion cueing algorithm but does not address user dependence for each scenario. This could be a future line of research.

6.2 Future research

The tuning of the parameters for the motion cueing algorithm is a key issue in the current setup of the SP7 as well as in the general case of the motion simulator. Few studies have solved this problem by using the objective motion cueing algorithm tuning [92], but this requires the reference motion data from the actual vehicle or aircraft. This is not feasible for the practicality of obtaining the actual data from each object to be simulated. This approach also limits the possibility of what can be simulated. Our abstraction approach reduces the complexity, but still, the motion cueing parameters must be tuned for each scenario, which will be a major problem for the end-user due to the time taken and the motion sickness during the tuning process. Thus, on the basis of the abstraction concept, the idea to be explored is the possibility of creating a new motion cueing algorithm with meaningful parameters to tune. This could be done by building a physical model for each filter of an existing motion cueing algorithm. This concept could not be used for training purposes such as pilot training, but it has the potential to successfully replace the motion cueing algorithm used in the entertainment motion simulator.

References

[1] L. Young and C. Oman, "Model for vestibular adaptation to horizontal rotation," *Aerospace medicine*, vol. 40, no. 10, pp. 1076–1080, 1969.

[2] S. F. Schmidt and B. Conrad, *Motion drive signals for piloted flight simulators*. National Aeronautics and Space Administration, 1970, vol. 1601.

[3] R. J. Telban, W. Wu, F. M. Cardullo, and J. A. Houck, "Motion cueing algorithm development: Initial investigation and redesign of the algorithms," 2000.

[4] MSIhub, "Vlts evaluation system github repository," 2020. [Online]. Available: https://github.com/MSIhub/Vive_SpatialTrackingStudy.git

[5] GVR-1-68038-831-2, "Virtual reality market size: Industry report, 2020-2027." [Online]. Available: https://www.grandviewresearch.com/industry-analysis/virtual-reality-vr-market#:~:text=The%20global%20virtual%20reality%20market,visual%20experience%20to%20the%20users.

[6] T. Suit, "Haptic feedback vr suit for motion capture and vr training." [Online]. Available: https://teslasuit.io/the-suit/

[7] Infinadeck, "True omnidirectional treadmill." [Online]. Available: https://infinadeck.com/

[8] "Yaw virtual reality motion simulator." [Online]. Available: https://www.yawvr.com/

[9] "Paleblue: 6dof-2000-35as3." [Online]. Available: https://pale.blue/technology/motion-platform-simulators/motion-platforms/6dof-2000-35as3/

[10] "Automated amusement rides with kuka robots." [Online]. Available: https://www.kuka.com/en-de/industries/other-industries/entertainment-industry/amusement-rides

[11] "Unreal engine: The most powerful real-time 3d creation platform." [Online]. Available: https://docs.unrealengine.com/en-US/index.html

[12] "Unity real-time development platform." [Online]. Available: https://unity.com/

[13] "Blender: The free and open source 3d creation suite." [Online]. Available: https://www.blender.org/support/tutorials/

[14] "Insta360 one x2: A clever and endlessly creative 360 camera with easy ai editing." [Online]. Available: https://www.insta360.com/product/insta360-onex2

[15] "Creo cad software: Enable the latest in design." [Online]. Available: https://www.ptc.com/en/products/creo

[16] W. Wu, F. Cardullo, W. Wu, and F. Cardullo, "Is there an optimum motion cueing algorithm?" in *Modeling and Simulation Technologies Conference*, 1997, p. 3506.

[17] E. Groen and W. Bles, "How to use body tilt for the simulation of linear self motion," *Journal of Vestibular Research*, vol. 14, no. 5, pp. 375–385, 2004.

[18] M. S. Ikbal, "Generation of motion of a motion simulator for virtual reality experiences from real (measured) linear and rotational accelerations." Master's thesis, Ecole Centrale de Nantes and University of Genoa, Genoa, Italy, 2017.

[19] E. B. Goldstein and J. Brockmole, *Sensation and perception*. Cengage Learning, 2016.

[20] S. Millar *et al.*, *Space and sense*. Psychology Press, 2008.

[21] R. Blake and M. Shiffrar, "Perception of human motion," *Annual review of psychology*, vol. 58, 2007.

[22] D. R. Gum, "Modeling of the human force and motion sensing mechanisms," Ph.D. dissertation, Ohio State University, 1972.

[23] E. Braunwald, A. S. Fauci, D. L. Kasper, S. L. Hauser, D. L. Longo, and J. L. Jameson, *Harrison's principles of internal medicine*. McGraw Hill, 2001.

[24] R. C. Fitzpatrick and B. L. Day, "Probing the human vestibular system with galvanic stimulation," *Journal of applied physiology*, vol. 96, no. 6, pp. 2301–2316, 2004.

[25] "Vesitbular system." [Online]. Available: http://www.humanneurophysiology.com/vestibularsystem.htm

[26] S. Casas, R. Olanda, and N. Dey, "Motion cueing algorithms: a review: algorithms, evaluation and tuning," *International Journal of Virtual and Augmented Reality (IJVAR)*, vol. 1, no. 1, pp. 90–106, 2017.

[27] L. Reid and M. Nahon, "Flight simulation motion-base drive algorithms: part 1. developing and testing equations," *UTIAS Report, No. 296*, 1985.

[28] K. Stahl, G. Abdulsamad, K.-D. Leimbach, and Y. A. Vershinin, "State of the art and simulation of motion cueing algorithms for a six degree of freedom driving simulator," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 537–541.

[29] D. R. Berger, J. Schulte-Pelkum, and H. H. Bülthoff, "Simulating believable forward accelerations on a stewart motion platform," *ACM Transactions on Applied Perception (TAP)*, vol. 7, no. 1, pp. 1–27, 2010.

[30] R. V. Parrish, J. E. Dieudonne, R. L. Bowles, and D. J. Martin Jr, "Coordinated adaptive washout for motion simulators," *Journal of aircraft*, vol. 12, no. 1, pp. 44–50, 1975.

[31] M. Nahon, L. Reid, and J. Kirdeikis, “Adaptive simulator motion software with supervisory control,” *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 2, pp. 376–383, 1992.

[32] R. Sivan, J. Ish-Shalom, and J.-K. Huang, “An optimal control approach to the design of moving flight simulators,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 12, no. 6, pp. 818–827, 1982.

[33] A. H. J. Jamson, *Motion cueing in driving simulators for research applications*. University of Leeds, 2010.

[34] M. Dagdelen, G. Reymond, A. Kemeny, M. Bordier, and N. Maïzi, “Model-based predictive motion cueing strategy for vehicle driving simulators,” *Control Engineering Practice*, vol. 17, no. 9, pp. 995–1003, 2009.

[35] “Unreal datasmith: Gets your design data into unreal quickly and easily.” [Online]. Available: https://docs.unrealengine.com/en-US/WorkingWithContent/Importing/Datasmith/SoftwareInteropGuides/CAD/index.html#:~:text=CAD%20Workflow,-Datasmith%20uses%20a&text=Save%20your%20CAD%20scene%20to,Content%20into%20Unreal%20Engine%204%20.

[36] “Steamvr is the ultimate tool for experiencing vr content on the hardware of your choice.” [Online]. Available: https://store.steampowered.com/app/250820/SteamVR/

[37] S. Soto-Faraco, A. Kingstone, and C. Spence, “Multisensory contributions to the perception of motion,” *Neuropsychologia*, vol. 41, no. 13, pp. 1847–1862, 2003.

[38] R. M. Baños, C. Botella, M. Alcañiz, V. Liaño, B. Guerrero, and B. Rey, “Immersion and emotion: their impact on the sense of presence,” *Cyberpsychology & behavior*, vol. 7, no. 6, pp. 734–741, 2004.

[39] “Main page of x-plane sdk.” [Online]. Available: http://www.xsquawkbox.net/xpsdk/mediawiki/Main_Page

[40] “X-plane plug-in sample code.” [Online]. Available: https://docs.unrealengine.com/en-US/index.html

[41] D. Stewart, “A platform with six degrees of freedom,” *Proceedings of the institution of mechanical engineers*, vol. 180, no. 1, pp. 371–386, 1965.

[42] “The mathematics of the stewart platform.” [Online]. Available: https://web.archive.org/web/20130506134518/http://www.wokinghamu3a.org.uk/Maths%20of%20the%20Stewart%20Platform%20v5.pdf

[43] K. Liu, M. Fitzgerald, D. Dawson, and F. L. Lewis, “Modeling and control of a stewart platform manipulator,” in *Proceedings of the Symposium of Control of Systems with Inexact Dynamic Models*, 1991, pp. 83–89.

[44] Q. Jiang and C. M. Gosselin, “The maximal singularity-free workspace of the gough–stewart platform for a given orientation,” *Journal of Mechanical Design*, vol. 130, no. 11, 2008.

[45] "Maplesoft: Mathematics-based software and services for education, engineering, and research." [Online]. Available: https://www.maplesoft.com/

[46] H. K. Kim, J. Park, Y. Choi, and M. Choe, "Virtual reality sickness questionnaire (vrsq): Motion sickness measurement index in a virtual reality environment," *Applied Ergonomics*, vol. 69, pp. 66 – 73, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S000368701730282X

[47] R. S. Kennedy and A. Graybiel, *The Dial test: A standardized procedure for the experimental production of canal sickness symptomatology in a rotating environment*. US Naval School of Aviation Medicine, US Naval Aviation Medical Center, 1965.

[48] S. Alexander, M. Cotzin, C. Hill Jr, E. Ricciuti, and G. Wendt, "Wesleyan university studies of motion sickness: Vi. prediction of sickness on a vertical accelerator by means of a motion sickness history questionnaire," *The Journal of Psychology*, vol. 20, no. 1, pp. 25–30, 1945.

[49] R. S. Kennedy, "Motion sickness questionnaire and field independence scores as predictors of success in naval aviation training." *Aviation, space, and environmental medicine*, 1975.

[50] R. S. Kennedy, J. E. Fowlkes, K. S. Berbaum, and M. G. Lilienthal, "Use of a motion sickness history questionnaire for prediction of simulator sickness." *Aviation, Space, and Environmental Medicine*, vol. 63, no. 7, pp. 588–593, 1992.

[51] P. J. Gianaros, E. R. Muth, J. T. Mordkoff, M. E. Levine, and R. M. Stern, "A questionnaire for the assessment of the multiple dimensions of motion sickness," *Aviation, space, and environmental medicine*, vol. 72, no. 2, p. 115, 2001.

[52] M. Mollenhauer, R. Romano, and B. Brumm, "The evaluation of a motion base driving simulator in a cave at tacom," in *Transformational Science And Technology For The Current And Future Force: (With CD-ROM)*. World Scientific, 2006, pp. 323–330.

[53] B. Aykent, F. Merienne, C. Guillet, D. Paillot, and A. Kemeny, "Motion sickness evaluation and comparison for a static driving simulator and a dynamic driving simulator," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 228, no. 7, pp. 818–829, 2014.

[54] S. Bruck and P. A. Watters, "The factor structure of cybersickness," *Displays*, vol. 32, no. 4, pp. 153–158, 2011.

[55] F. Bracco, C. Chiorri, D. Glowinski, B. Hosseini Nia, and G. Vercelli, "Investigating visual discomfort with 3d displays: the stereoscopic discomfort scale," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 307–312.

[56] A. Murata, "Effects of duration of immersion in a virtual reality environment on postural stability," *International Journal of Human-Computer Interaction*, vol. 17, no. 4, pp. 463–477, 2004.

[57] E. Chang, H. T. Kim, and B. Yoo, "Virtual reality sickness: a review of causes and measurements," *International Journal of Human–Computer Interaction*, vol. 36, no. 17, pp. 1658–1682, 2020.

[58] J. F. Golding, K. Doolan, A. Acharya, M. Tribak, and M. A. Gresty, “Cognitive cues and visually induced motion sickness,” *Aviation, space, and environmental medicine*, vol. 83, no. 5, pp. 477–482, 2012.

[59] V. I. Utkin, “Sliding mode and its applications to variable structure systems,” *IEEE Transactions on Automatic Control*, vol. 22, no. 2, p. 2l2e222, 1978.

[60] “An openvr driver that allows to enable motion compensation.” [Online]. Available: https://ovrmc.dschadu.de/en/Download

[61] HTC, “Vive tracker developer guidelines,” Jun 2020. [Online]. Available: https://developer.vive.com/resources/knowledgebase/vive-tracker-developer-guidelines/

[62] M. Borges, A. Symington, B. Coltin, T. Smith, and R. Ventura, “Htc vive: analysis and accuracy improvement,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2610–2615.

[63] T. A. Jost, G. Drewelow, S. Koziol, and J. Rylander, “A quantitative method for evaluation of 6 degree of freedom virtual reality systems,” *Journal of biomechanics*, vol. 97, p. 109379, 2019.

[64] D. C. Niehorster, L. Li, and M. Lappe, “The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research,” *i-Perception*, vol. 8, no. 3, p. 2041669517708205, 2017.

[65] E. Luckett, “A quantitative evaluation of the htc vive for virtual reality research,” Ph.D. dissertation, The University of Mississippi, 2018.

[66] S. M. van der Veen, M. Bordeleau, P. E. Pidcoe, C. R. France, and J. S. Thomas, “Agreement analysis between vive and vicon systems to monitor lumbar postural changes,” *Sensors*, vol. 19, no. 17, p. 3632, 2019.

[67] T. Ameler, K. Blohme, L. Brandt, R. Brüngel, A. Hensel, L. Huber, F. Kuper, J. Swoboda, M. Warnecke, M. Warzecha, D. Heß, J. Frömke, A. Schmitz-Stolbrink, and C. M. Friedrich, “A comparative evaluation of steamvr tracking and the optitrack system for medical device tracking,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019, pp. 1465–1470.

[68] K. Sletten, “Automated testing of industrial robots using htc vive for motion tracking,” Master’s thesis, University of Stavanger, Norway, 2017.

[69] M. Le Chénéchal and J. Chatel-Goldman, “HTC Vive Pro time performance benchmark for scientific research,” in *ICAT-EGVE 2018*, Limassol, Cyprus, Nov. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01934741

[70] M. Greiff, A. Robertsson, and K. Berntorp, “Performance bounds in positioning with the vive lighthouse system,” in *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.

[71] Y. Yang, D. Weng, D. Li, and H. Xun, “An improved method of pose estimation for lighthouse base station extension,” *Sensors*, vol. 17, no. 10, p. 2411, 2017.

[72] M. Suznjevic, M. Mandurov, and M. Matijasevic, "Performance and qoe assessment of htc vive and oculus rift for pick-and-place tasks in vr," in *2017 Ninth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2017, pp. 1–3.

[73] A. International, "E177-19 standard practice for use of the terms precision and bias in astm test methods," West Conshohocken, PA, 2019, doi:. [Online]. Available: https://doi.org/10.1520/E0177-19

[74] ASTM-E2919-14, "Standard test method for evaluating the performance of systems that measure static, six degrees of freedom (6dof), pose," ASTM International, West Conshohocken, PA, 2014. [Online]. Available: http://www.astm.org/cgi-bin/resolver.cgi?E2919

[75] ASTM-E3064-16, "Standard test method for evaluating the performance of optical tracking systems that measure six degrees of freedom (6dof) pose," ASTM International, West Conshohocken, PA, 2016. [Online]. Available: http://www.astm.org/cgi-bin/resolver.cgi?E3064

[76] A. Yates, "Alan yates on the impossible task of making valve's vr work," 2016.

[77] Comau, "Ns-16-1.65: characteristics and technical specifics," 2015. [Online]. Available: https://www.comau.com/IT/le-nostre-competenze/robotics/robot-team/ns-16-165

[78] I. O. for Standardization, "Iso9283," 2015. [Online]. Available: https://www.iso.org/standard/22244.html

[79] M. Laribi, L. Romdhane, and S. Zeghloul, "Analysis and dimensional synthesis of the delta robot for a prescribed workspace," *Mechanism and machine theory*, vol. 42, no. 7, pp. 859–870, 2007.

[80] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A study of vicon system positioning performance," *Sensors*, vol. 17, no. 7, p. 1591, 2017.

[81] B. Sabata and J. Aggarwal, "Estimation of motion from a pair of range images: A review," *CVGIP: Image Understanding*, vol. 54, no. 3, pp. 309–324, 1991.

[82] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-d rigid body transformations: a comparison of four major algorithms," *Machine vision and applications*, vol. 9, no. 5-6, pp. 272–290, 1997.

[83] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Computer Architecture Letters*, vol. 13, no. 04, pp. 376–380, 1991.

[84] J. Cashbaugh and C. Kitts, "Automatic calculation of a transformation matrix between two frames," *IEEE Access*, vol. 6, pp. 9614–9622, 2018.

[85] W. M. Mendenhall and T. L. Sincich, *Statistics for Engineering and the Sciences*. CRC Press, 2016.

[86] M. L. Head, L. Holman, R. Lanfear, A. T. Kahn, and M. D. Jennions, "The extent and consequences of p-hacking in science," *PLoS Biol*, vol. 13, no. 3, p. e1002106, 2015.

[87] S. Casas, C. Portalés, P. Morillo, and M. Fernández, "To move or not to move? analyzing motion cueing in vehicle simulators by means of massive simulations," *Virtual Reality*, vol. 24, no. 1, pp. 93–108, 2020.

[88] P. R. Grant and L. D. Reid, "Motion washout filter tuning: Rules and requirements," *Journal of aircraft*, vol. 34, no. 2, pp. 145–151, 1997.

[89] S. Casas, I. Coma, J. V. Riera, and M. Fernández, "Motion-cuing algorithms: Characterization of users' perception," *Human factors*, vol. 57, no. 1, pp. 144–162, 2015.

[90] S. Casas, I. Coma, C. Portalés, and M. Fernández, "Towards a simulation-based tuning of motion cueing algorithms," *Simulation Modelling Practice and Theory*, vol. 67, pp. 137–154, 2016.

[91] M. Grottoli, D. Cleij, P. Pretto, Y. Lemmens, R. Happee, and H. H. Bülthoff, "Objective evaluation of prediction strategies for optimization-based motion cueing," *Simulation*, vol. 95, no. 8, pp. 707–724, 2019.

[92] R. Hosman and S. Advani, "Design and evaluation of the objective motion cueing test and criterion," *The Aeronautical Journal*, vol. 120, no. 1227, p. 873, 2016.

[93] M. Bruenger-Koch, "Motion parameter tuning and evaluation for the dlr automotive simulator," *2005 DSC NA CD-ROM Proceedings*, 2005.

[94] J. H. Pasma, D. Engelhart, A. C. Schouten, H. van der Kooij, A. B. Maier, and C. G. Meskers, "Impaired standing balance: the clinical need for closing the loop," *Neuroscience*, vol. 267, pp. 157–165, 2014.

[95] R. J. Peterka, "Sensory integration for human balance control," *Handbook of clinical neurology*, vol. 159, pp. 27–42, 2018.

[96] M. G. Gaerlan, "The role of visual, vestibular, and somatosensory systems in postural balance, 2010," *UNLV Theses/Dissertations/Professional Papers/Capstones*.

[97] I. D. Loram, C. N. Maganaris, and M. Lakie, "Active, non-spring-like muscle movements in human postural sway: how might paradoxical changes in muscle length be produced?" *The Journal of physiology*, vol. 564, no. 1, pp. 281–293, 2005.

[98] G. G. Slabaugh, "Computing euler angles from a rotation matrix," *Retrieved on August*, vol. 6, no. 2000, pp. 39–63, 1999.

[99] R. J. Krauzlis and S. A. Adler, "Effects of directional expectations on motion perception and pursuit eye movements," *Visual Neuroscience*, vol. 18, no. 3, p. 365, 2001.

[100] M. Chalk, A. R. Seitz, and P. Seriès, "Rapidly learned stimulus expectations alter perception of motion," *Journal of vision*, vol. 10, no. 8, pp. 2–2, 2010.

Appendix A

Blueprint flow of the custom cue generation system

The use of the UE to integrate custom visual, audio, and motion cues for the experience of virtual reality is described in section 2.1.1. The blueprint for the integration of one of the scenarios is shown in the Fig. A.1, A.2 and A.3. The shown blueprint is responsible for sending motion data to the SP7 controller and updating the pose of the visual object from the CAD simulation data. Few details of the flowchart are explained below.

In Fig A.1, the top right section, you can find the key to the letter "S" at the beginning of the scenario. At play start, there is a module to get the level name, if the level name has "NoMotion" then the motion data is not sent to the MCA controller of the SP7. For all other level names, the player pawn (UE object with VR camera component) is cast to the tunnel mesh and the motion data is sent to the MCA controller at each tick. The motion of the object is accomplished by changing the pose of the pawn at each tick interval. Fig A.2 is the continuation of Fig A.1, where the frame transformation from Creo to UE is executed at each tick update and then the visual mesh is transformed as can be seen in Fig A.3. These transformations of the framework have been described in section 5.1.1. Finally, the motion of the player pawn is updated by setting the calculated motion as the current position and orientation.

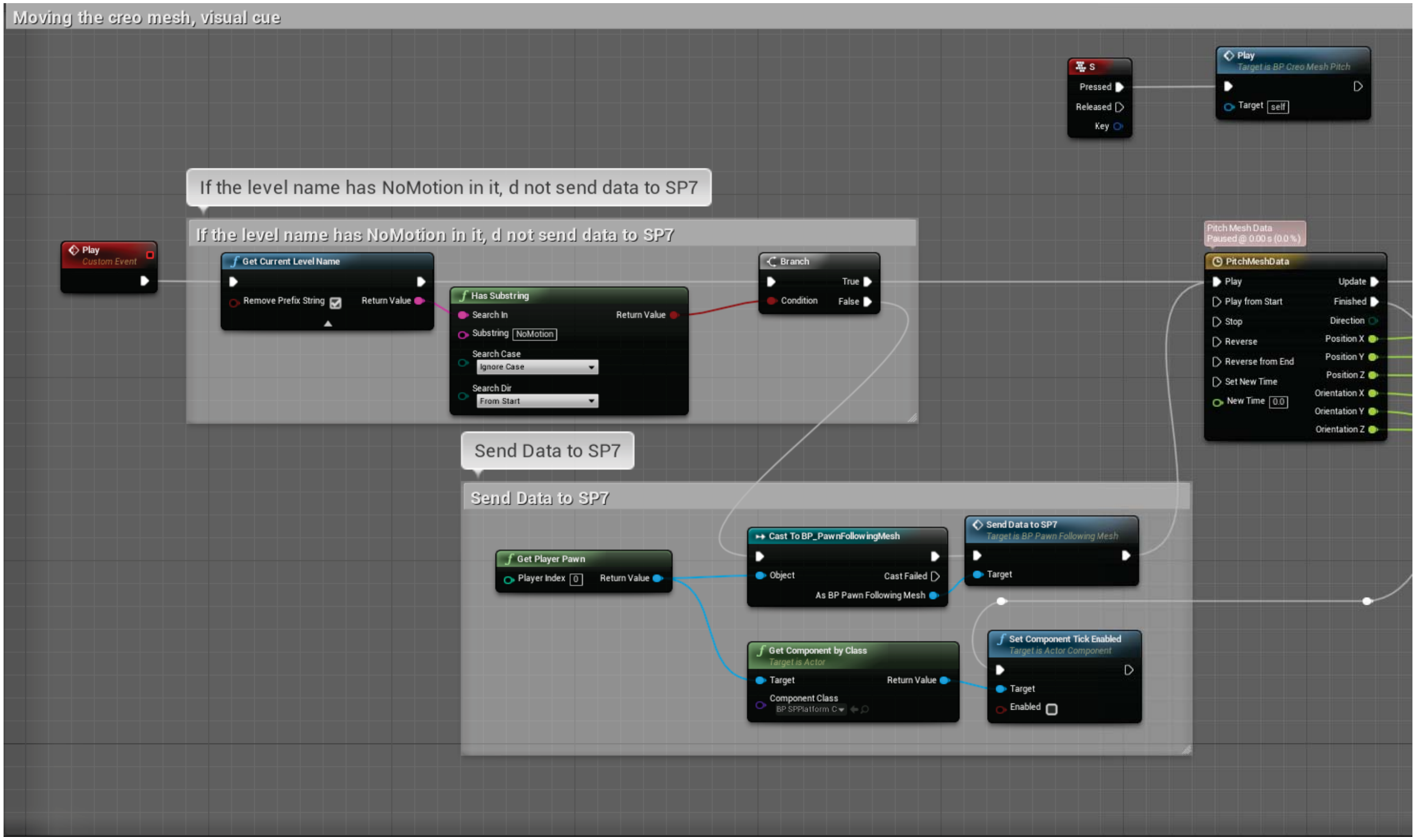


Figure A.1 A snippet of the blueprint flow of the tunnel pitch scenario application in UE (1/3).

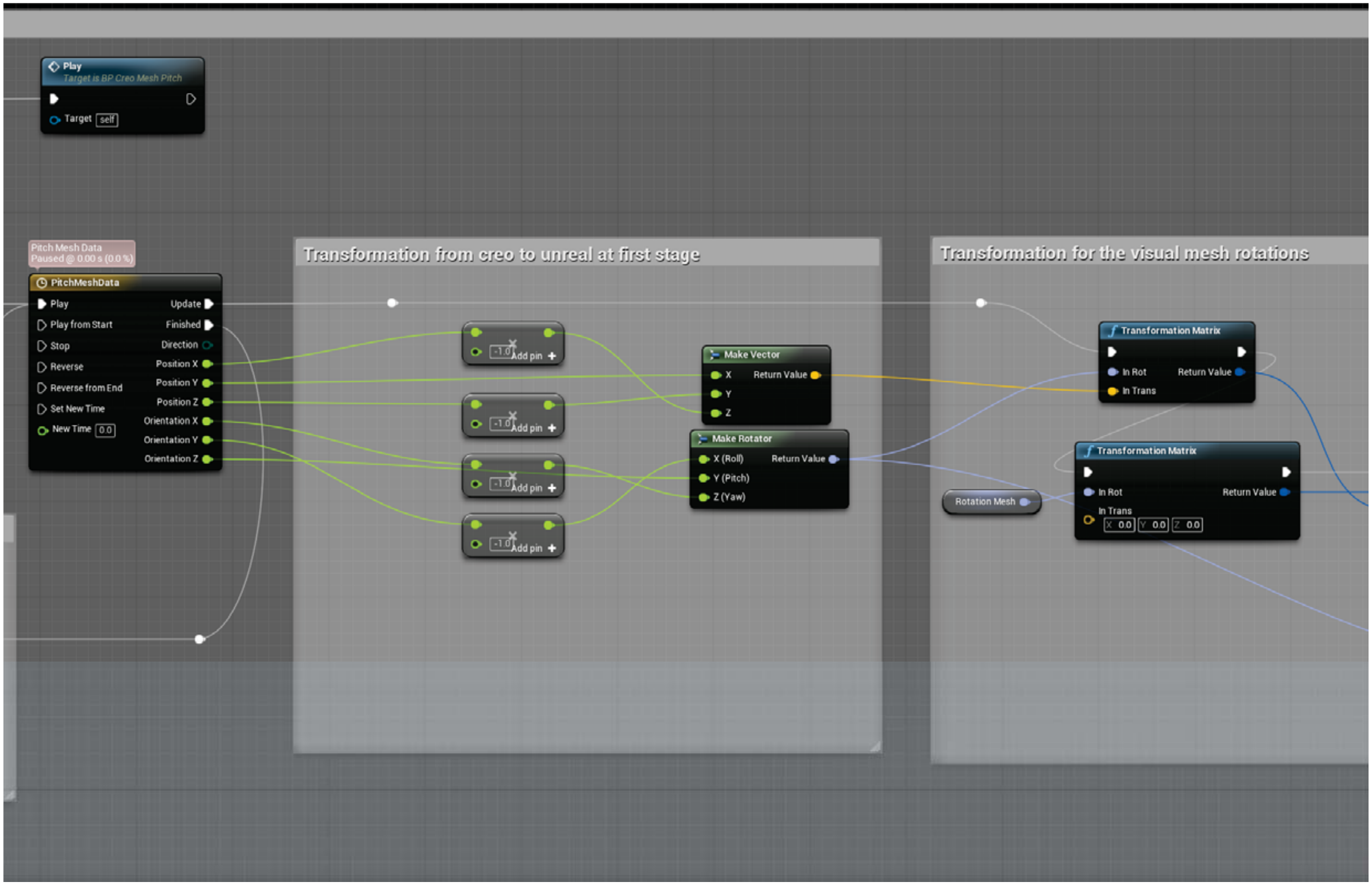


Figure A.2 A snippet of the blueprint flow of the tunnel pitch scenario application in UE (2/3).

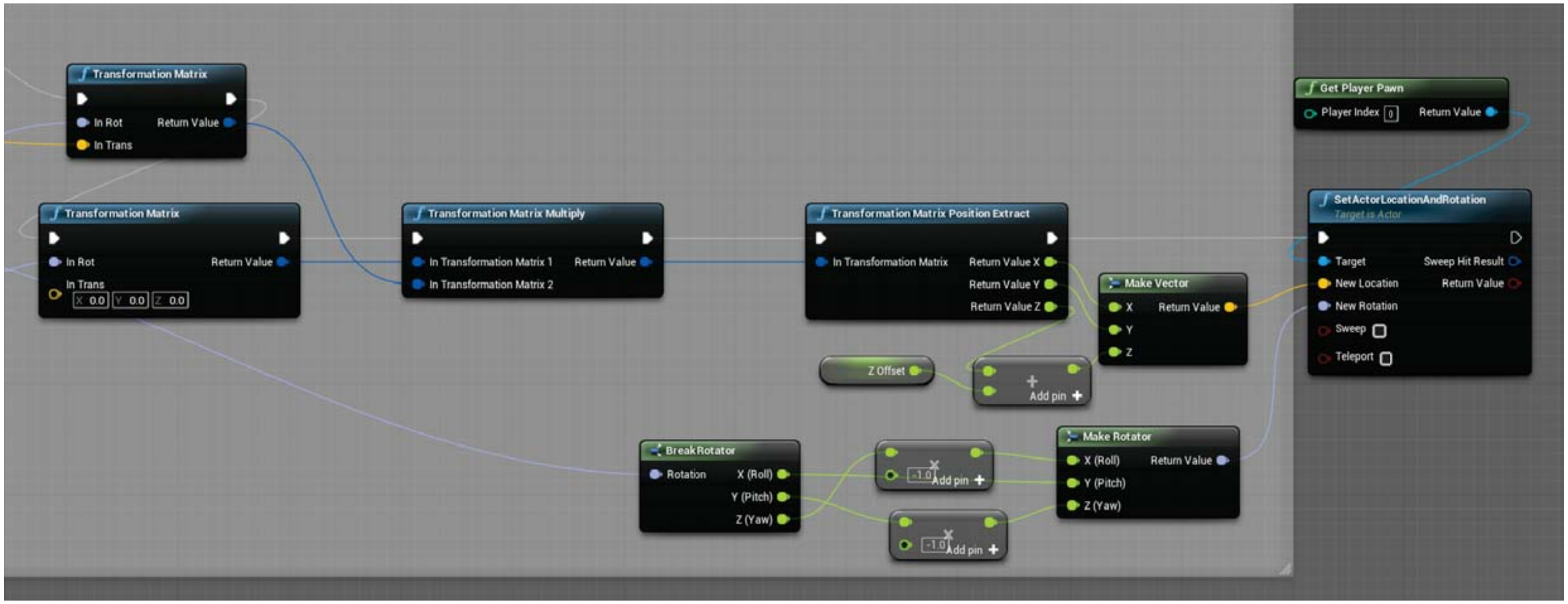


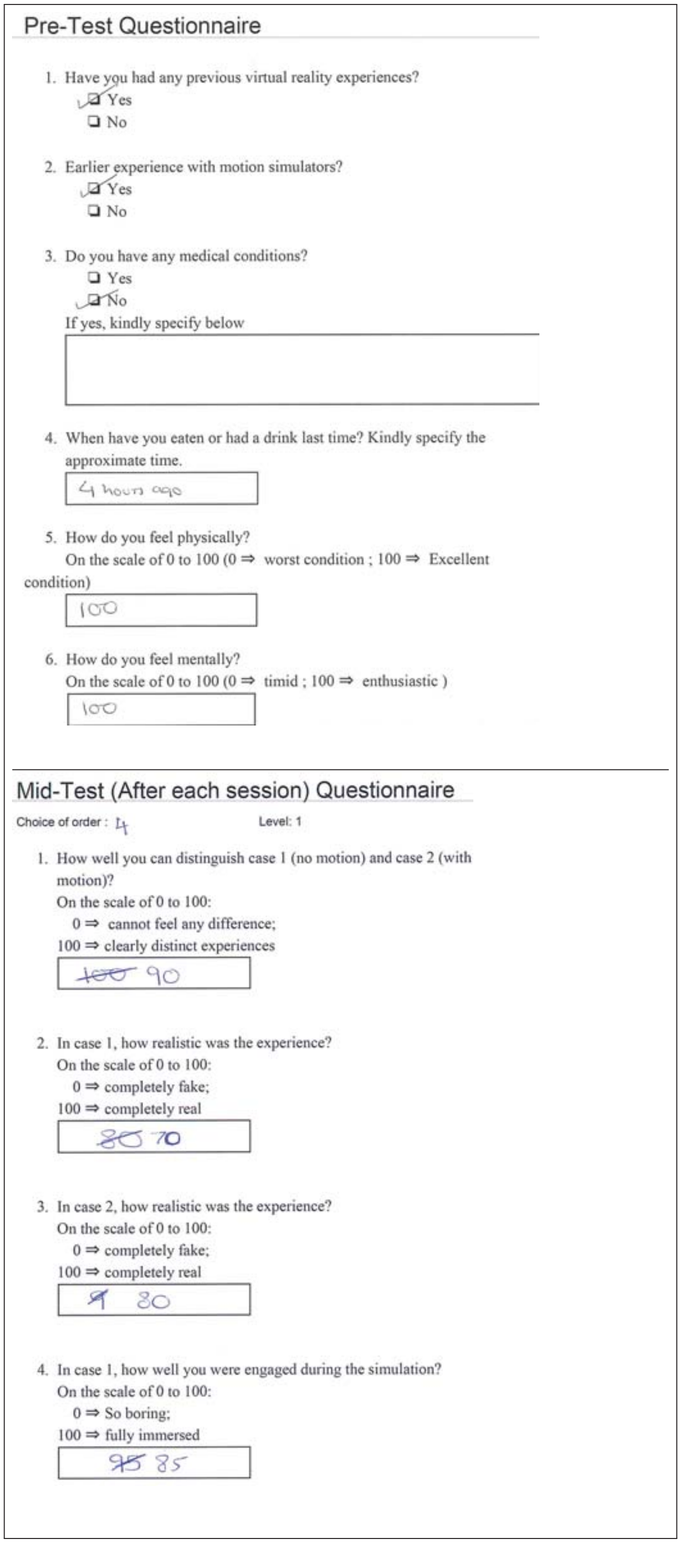
Figure A.3 A snippet of the blueprint flow of the tunnel pitch scenario application in UE (3/3).

Appendix B

A sample of questionnaire filled out by a jury for subjective evaluation of motion perception

As described in the section 3.2.4, a sample of the completed questionnaire is presented here. This is the evaluation of 1 out of 18 juries participated in the session. The run through of this experimental session was recorded and made available for reference at https://youtu.be/CmOhWla3blo. The procedure explained in 3.2.1 is followed systematically. Details on the jury whose response to the questionnaire is annexed to this appendix:

- Jury's age and gender: 27, Male
- Date and time of the experiment session: 06/02/2019, 17:45
- Previous experience: This jury has previous experience on SP7 motion simulation during the empirical tuning of MCA parameters.
- The average satisfaction score from this questionnaire = 65.49%



Pre-Test Questionnaire

1. Have you had any previous virtual reality experiences?
   ☑ Yes
   □ No

2. Earlier experience with motion simulators?
   ☑ Yes
   □ No

3. Do you have any medical conditions?
   □ Yes
   ☑ No
   If yes, kindly specify below

4. When have you eaten or had a drink last time? Kindly specify the approximate time.
   4 hours ago

5. How do you feel physically?
   On the scale of 0 to 100 (0 ⇒ worst condition ; 100 ⇒ Excellent condition)
   100

6. How do you feel mentally?
   On the scale of 0 to 100 (0 ⇒ timid ; 100 ⇒ enthusiastic)
   100

Mid-Test (After each session) Questionnaire

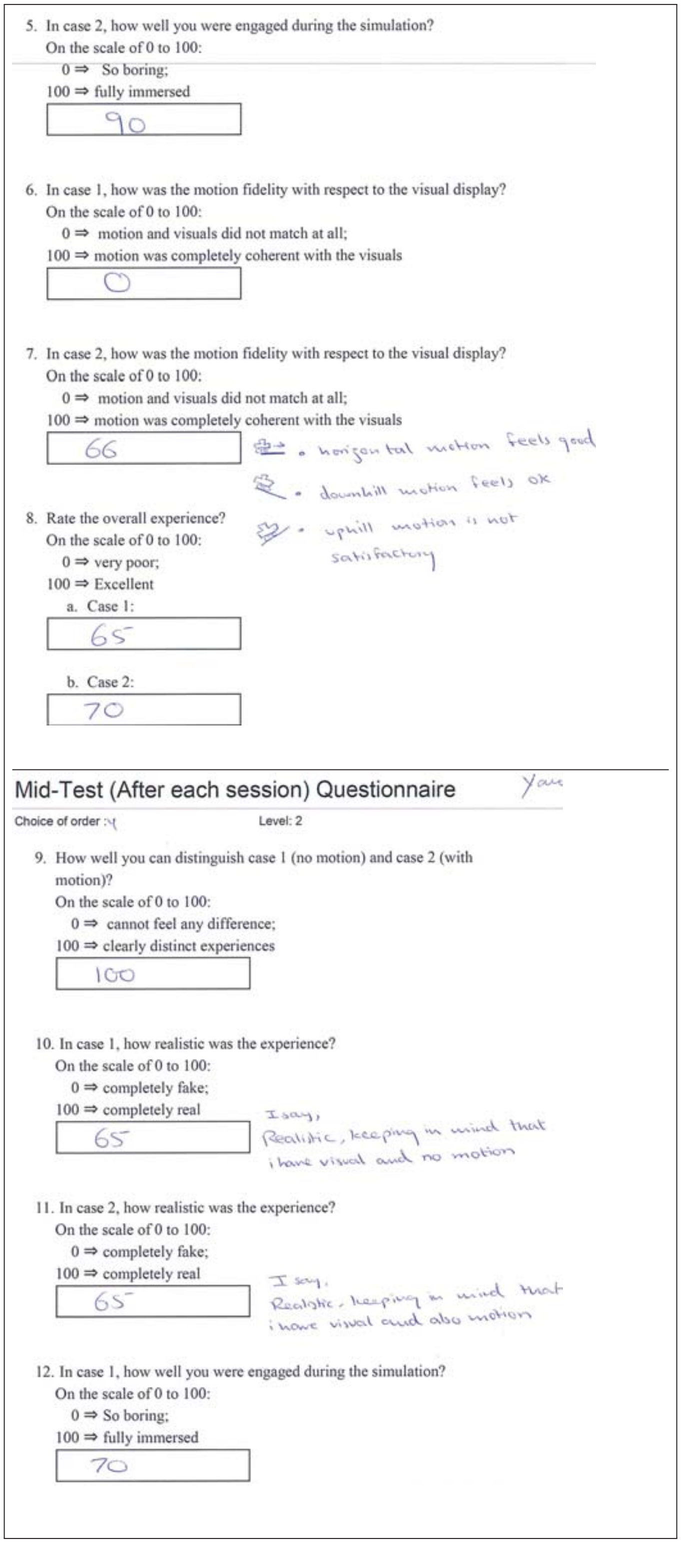
Choice of order : 4 Level: 1

1. How well you can distinguish case 1 (no motion) and case 2 (with motion)?
   On the scale of 0 to 100:
   0 ⇒ cannot feel any difference;
   100 ⇒ clearly distinct experiences
   ~~100~~ 90

2. In case 1, how realistic was the experience?
   On the scale of 0 to 100:
   0 ⇒ completely fake;
   100 ⇒ completely real
   ~~80~~ 70

3. In case 2, how realistic was the experience?
   On the scale of 0 to 100:
   0 ⇒ completely fake;
   100 ⇒ completely real
   ~~9~~ 80

4. In case 1, how well you were engaged during the simulation?
   On the scale of 0 to 100:
   0 ⇒ So boring;
   100 ⇒ fully immersed
   ~~95~~ 85



5. In case 2, how well you were engaged during the simulation?
   On the scale of 0 to 100:
   0 ⇒ So boring;
   100 ⇒ fully immersed
   90

6. In case 1, how was the motion fidelity with respect to the visual display?
   On the scale of 0 to 100:
   0 ⇒ motion and visuals did not match at all;
   100 ⇒ motion was completely coherent with the visuals
   0

7. In case 2, how was the motion fidelity with respect to the visual display?
   On the scale of 0 to 100:
   0 ⇒ motion and visuals did not match at all;
   100 ⇒ motion was completely coherent with the visuals
   66

   - horizontal motion feels good
   - downhill motion feels ok
   - uphill motion is not satisfactory

8. Rate the overall experience?
   On the scale of 0 to 100:
   0 ⇒ very poor;
   100 ⇒ Excellent
   a. Case 1:
   65
   b. Case 2:
   70

Mid-Test (After each session) Questionnaire

Yau

Choice of order : 4 Level: 2

9. How well you can distinguish case 1 (no motion) and case 2 (with motion)?
   On the scale of 0 to 100:
   0 ⇒ cannot feel any difference;
   100 ⇒ clearly distinct experiences
   100

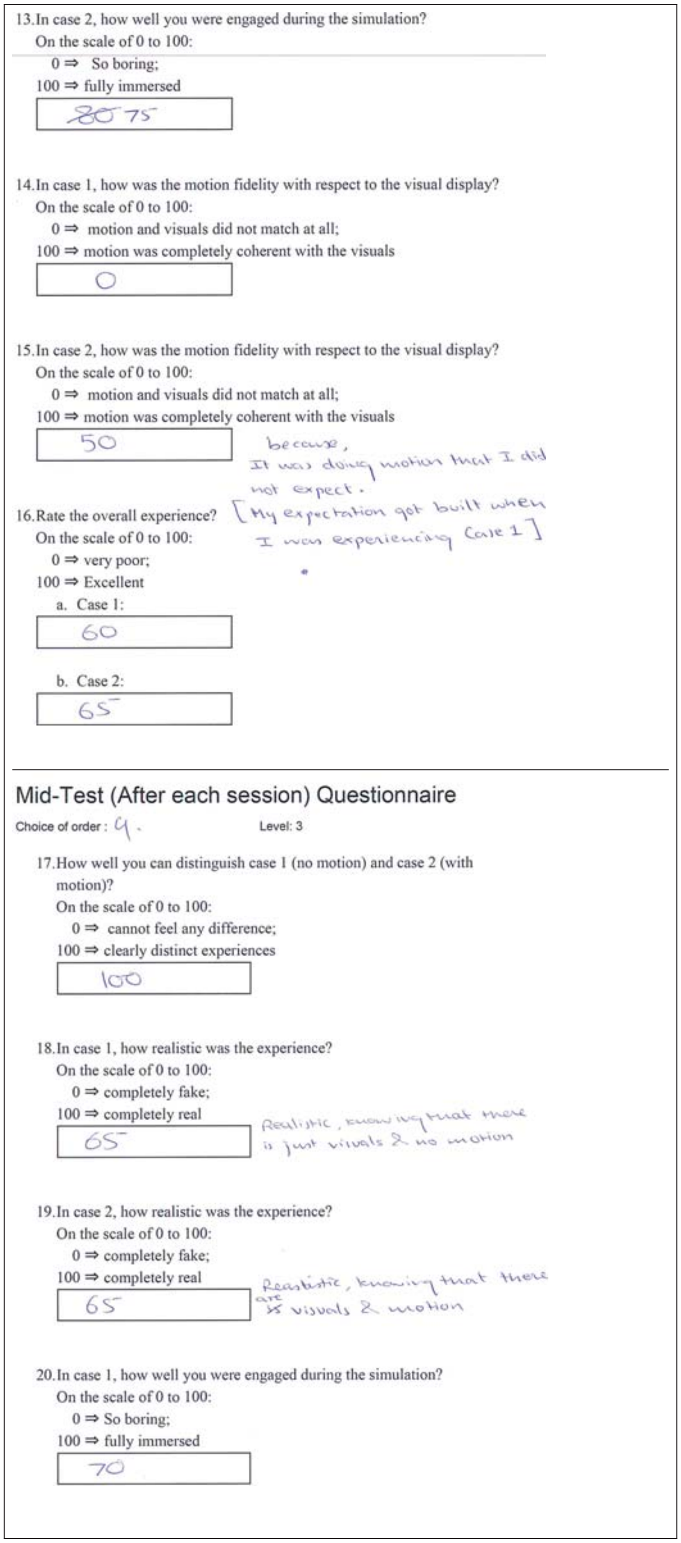
10. In case 1, how realistic was the experience?
    On the scale of 0 to 100:
    0 ⇒ completely fake;
    100 ⇒ completely real
    65

    I say,
    Realistic, keeping in mind that
    i have visual and no motion

11. In case 2, how realistic was the experience?
    On the scale of 0 to 100:
    0 ⇒ completely fake;
    100 ⇒ completely real
    65

    I say.
    Realistic, keeping in mind that
    i have visual and also motion

12. In case 1, how well you were engaged during the simulation?
    On the scale of 0 to 100:
    0 ⇒ So boring;
    100 ⇒ fully immersed
    70



13.In case 2, how well you were engaged during the simulation?
On the scale of 0 to 100:
0 ⇒ So boring;
100 ⇒ fully immersed

~~80~~ 75

14.In case 1, how was the motion fidelity with respect to the visual display?
On the scale of 0 to 100:
0 ⇒ motion and visuals did not match at all;
100 ⇒ motion was completely coherent with the visuals

0

15.In case 2, how was the motion fidelity with respect to the visual display?
On the scale of 0 to 100:
0 ⇒ motion and visuals did not match at all;
100 ⇒ motion was completely coherent with the visuals

50

because,
It was doing motion that I did
not expect.
[My expectation got built when
I was experiencing Case 1]

16.Rate the overall experience?
On the scale of 0 to 100:
0 ⇒ very poor;
100 ⇒ Excellent

a. Case 1:

60

b. Case 2:

65

Mid-Test (After each session) Questionnaire

Choice of order : 4. Level: 3

17.How well you can distinguish case 1 (no motion) and case 2 (with motion)?
On the scale of 0 to 100:
0 ⇒ cannot feel any difference;
100 ⇒ clearly distinct experiences

100

18.In case 1, how realistic was the experience?
On the scale of 0 to 100:
0 ⇒ completely fake;
100 ⇒ completely real

65

Realistic, knowing that there
is just visuals & no motion

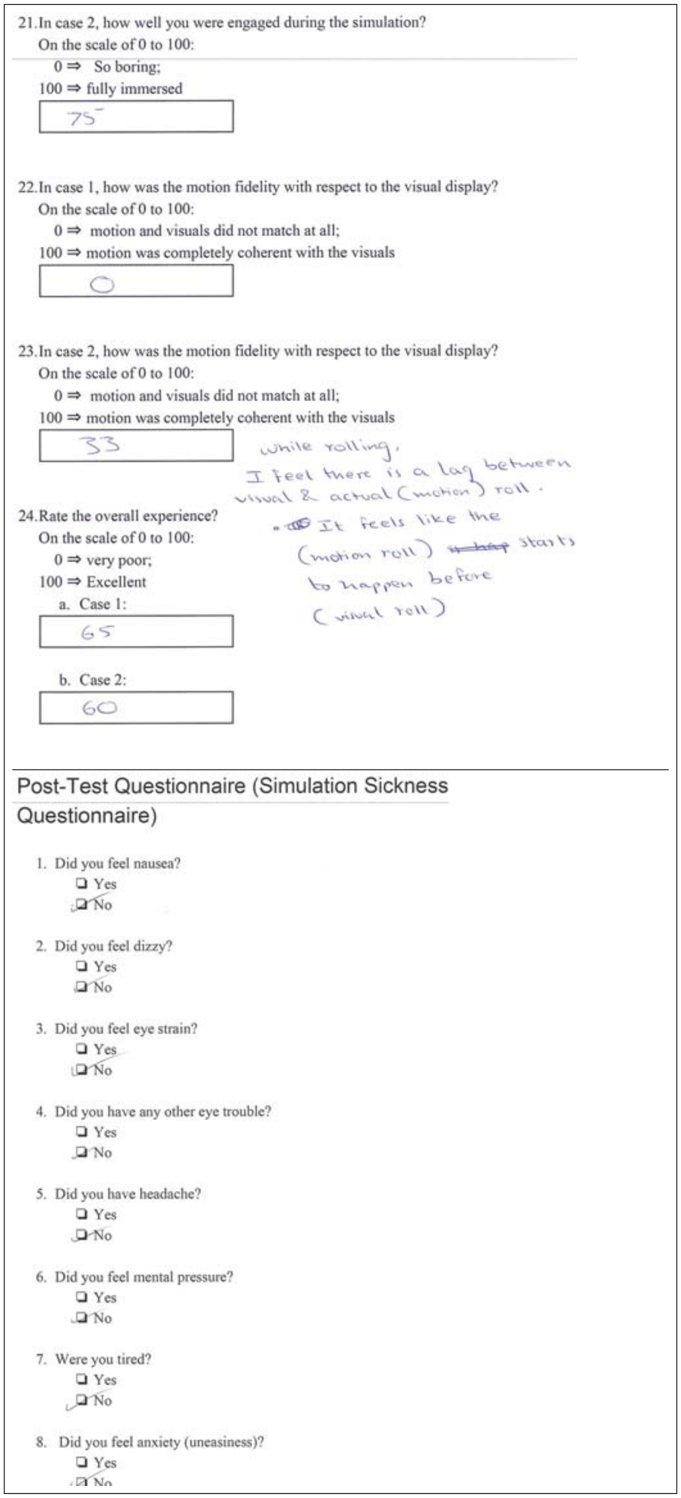
19.In case 2, how realistic was the experience?
On the scale of 0 to 100:
0 ⇒ completely fake;
100 ⇒ completely real

65

Reastistic, knowing that there
~~is~~ are visuals & motion

20.In case 1, how well you were engaged during the simulation?
On the scale of 0 to 100:
0 ⇒ So boring;
100 ⇒ fully immersed

70



21. In case 2, how well you were engaged during the simulation?
On the scale of 0 to 100:
0 ⇒ So boring;
100 ⇒ fully immersed

75

22. In case 1, how was the motion fidelity with respect to the visual display?
On the scale of 0 to 100:
0 ⇒ motion and visuals did not match at all;
100 ⇒ motion was completely coherent with the visuals

0

23. In case 2, how was the motion fidelity with respect to the visual display?
On the scale of 0 to 100:
0 ⇒ motion and visuals did not match at all;
100 ⇒ motion was completely coherent with the visuals

33

while rolling,
I feel there is a lag between
visual & actual (motion) roll.
- It feels like the
(motion roll) ~~it lag~~ starts
to happen before
(visual roll)

24. Rate the overall experience?
On the scale of 0 to 100:
0 ⇒ very poor;
100 ⇒ Excellent

a. Case 1:

65

b. Case 2:

60

Post-Test Questionnaire (Simulation Sickness Questionnaire)

1. Did you feel nausea?
☐ Yes
☑ No

2. Did you feel dizzy?
☐ Yes
☑ No

3. Did you feel eye strain?
☐ Yes
☑ No

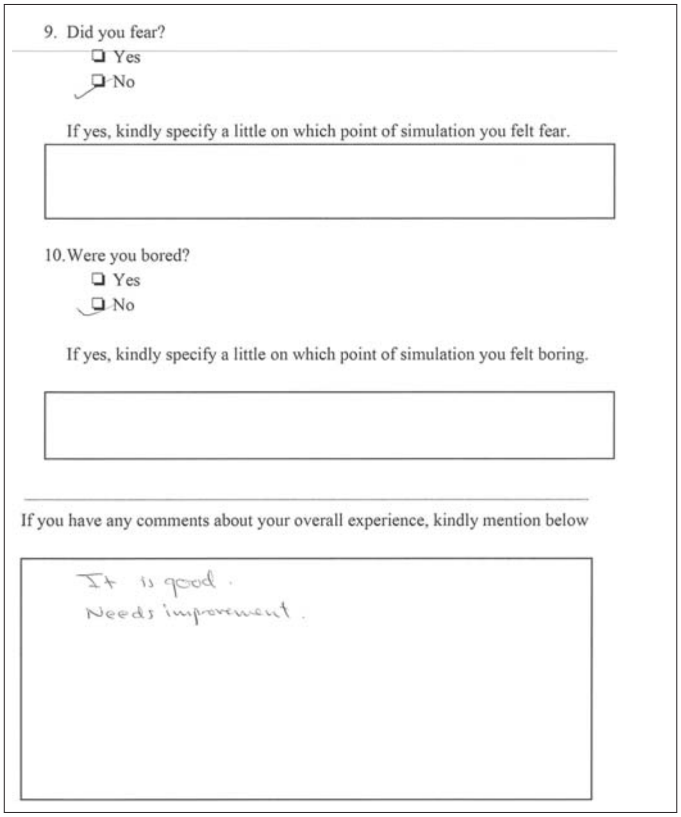
4. Did you have any other eye trouble?
☐ Yes
☑ No

5. Did you have headache?
☐ Yes
☑ No

6. Did you feel mental pressure?
☐ Yes
☑ No

7. Were you tired?
☐ Yes
☑ No

8. Did you feel anxiety (uneasiness)?
☐ Yes
☑ No



9. Did you fear?

   □ Yes

   ☑ No

   If yes, kindly specify a little on which point of simulation you felt fear.

10. Were you bored?

   □ Yes

   ☑ No

   If yes, kindly specify a little on which point of simulation you felt boring.

If you have any comments about your overall experience, kindly mention below

It is good.
Needs improvement.

Appendix C

Publication work during the period of this thesis

Journal publications

During this thesis work, three journal publications were made. The journal in section C.1 was on the bench-marking of the VLTS for its dynamic pose tracking evaluation. The calibration procedure was also included in this article. The journal is a by-product of the work on the solution of motion compensation issue explained in Chapter 4. Article in section C.2 was an attempt to solve the turn-around bump problem explained in section 3.3, comprehensive testing of the sliding mode solution received a mixed response and thus the brute force method was chosen in the current configuration. The research published in the article mentioned in section C.3 was performed to take advantage of the motion data available beforehand for passive scenarios. This was based on the principle that providing anticipatory cue would improve the efficiency of motion perception as the human body prepares for revealed cues, as described in [99] and [100]. For example, in a car driving scenario, if the user can see a massive slope down, the user's body can plan in such a way that even a small false motion will be well received. This idea was exploited using the acausal technique and a new MCA algorithm was developed using the classical MCA architecture. Extensive testing of the subjective analysis has been conducted for more than three months to demonstrate the efficacy of the algorithm. The detailed description of the algorithm is not stated in this thesis as it does not complement the flow of the main narrative and the contribution to the final product of SP7.

C.1 Dynamic Pose Tracking Performance Evaluation of HTC Vive Virtual Reality System

- **Authors:** Mohamed Sadiq Ikbal, Vishal Ramadoss, Matteo Zoppi
- **Published in:** IEEE Access, vol. 9, pp. 3798-3815, 2021, doi: 10.1109/ACCESS.2020.3047698.
- **URL:** https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9309218&isnumber=9312710
- **Abstract:** Virtual reality tracking devices are rapidly becoming the go-to system for cost-effective motion tracking solutions across different communities such as robotics, biomechanics, sports, rehabilitation, motion simulators, etc. This article focuses on the spatial tracking performance of HTC Vive's lighthouse tracking system (VLTS) devices (tracker, controller, and head mount display). A comprehensive literature survey on the performance analysis of VLTS on the various aspects is presented along with its shortcomings in terms of spatial tracking evaluation. The two key limitations have been identified: in static cases, there is a lack of standard procedures and criteria, and in dynamic cases, the entire study of spatial tracking. We address the first by assessing VLTS using the optical tracking system standard specified by ASTM International, and the latter by revising the standards to determine the upper-velocity limit for reliable tracking. The findings are substantiated with the trajectories of human wrist motion. Each evaluation's results are systematically analyzed with statistical hypothesis tests and criteria fulfillment. Comau NS16, an industrial serial robot, was used as the ground truth motion generator due to its repeatability and 6 degrees of workspace freedom. One of the major reasons for not having more generalized spatial tracking studies is that the tracking performance heavily depends on the configurations of the setup, work volume, environment, etc. Thus, the guidelines for configuring VLTS and the approach adapted from ASTM standards for evaluating VLTS for custom applications using our reported findings for both static and dynamic cases are included in the appendix.

C.2 A Sliding Mode-Based Approach to Motion Cueing for Virtual Reality Gaming using Motion Simulators

- **Authors:** Aman Sharma, Mohamed Sadiq Ikbal, Duc Trinh Cuong, Matteo Zoppi

- **Published in:** Springer: Virtual Reality (2020).
- **URL:** https://doi.org/10.1007/s10055-020-00439-5
- **Abstract:** Motion simulators have been of significant importance for the aviation sector in training pilots. However, the present boom in the utilization of robotics for virtual reality (VR) gaming has given rise to a new application of motion simulators. Motion cueing algorithms (MCA) play a key role in mapping the motions from a gaming scenario to the workspace of a simulator. This workspace is small (as compared to the gaming world), and on reaching the boundary, it becomes necessary to saturate the motion. Each degree of freedom, in the Cartesian space, is saturated between two fixed extremities. This hampers the perception of motion of a user enjoying the scenario. In order to address this practical problem, we make an attempt to enlarge the workspace and develop a mathematical methodology to prevent the simulator from exiting a non-cuboidal workspace. To do so, we propose sliding mode-based cueing algorithm (SMCA), which makes the simulator to slide in close proximity across the boundary of workspace. We make use of discrete-time models to present this methodology in order to ensure straightforward implementation by researchers in the future. Veracity of SMCA is testified by means of experimentation on SP7 motion simulator. The experimental results give evidence of a 57% increase in the considered sub-workspace, thereby reducing the relative necessity to saturate the motions as compared to classical MCA. This leads to a better experience of a user enjoying the VR scenario. On the other hand, the following drawbacks are reported: (1) necessity to analytically model the workspace boundary and ensuring that it is smooth with nonzero gradient, (2) SMCA parameter selection is more cumbersome than classical MCA, thereby making its utility restricted to recorded scenarios.

C.3 Acausal Approach to Motion Cueing

- **Authors:** Aman Sharma, Mohamed Sadiq Ikbal, Matteo Zoppi
- **Published in:** IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1013-1020, April 2019, doi: 10.1109/LRA.2019.2893709.
- **URL:** https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8616847&isnumber=8581687

- **Abstract:** Motion simulators have been used extensively by both industry and academia to train pilots, conduct psychological experiments on drivers, understand the perception of motion by humans, and cater to the burgeoning gaming industry among others. Working of a motion simulator can be summarized in following three steps: first, acquisition of motion signals; second, motion cueing: signal processing to generate motion references, and third, control: tracking the desired references. A motion cueing algorithm (MCA) acts as a bridge between the actual motions and the ones recreated by the simulator. Mathematically, MCA is constituted of the following four operations: scaling, saturation, filtering, and tiltcoordination (involves filtering too). The existing MCAs make use of causal filters to process the signals, thereby precluding the possibility of utilizing future motion signals to emulate pre-recorded scenarios. We present a new approach to generate motion cues by explicitly making use of future motion signals and causal linear filters. It is due to the usage of future motion signals (not the filter), we call the presented methodology as acausal cueing algorithm (ACA). Unlike most of the existing works on motion cueing, we choose to present the developed methodology using discrete-time models to facilitate its quick implementation by industry and other researchers in the future. The veracity of the presented methodology is examined by actuating a motion simulator (seven degrees of freedom parallel manipulator) based on the references generated by the ACA in response to test trajectories. The conducted experiments assert better performance of the ACA (over MCA) in the beginning, which eventually degrades in the last few seconds due to unavailability of future motion signals.

Parallel works: Conference articles

The conference papers referred to in section C.4, C.5, and C.6 were published with the other members of the PMAR robotics laboratory team while collaborating in the research projects.

C.4 Design of Serial Link Structure-Parallel Wire System for Virtual Reality Rehabilitation and Assessment

- **Authors:** Vishal Ramadoss, Mohamed Sadiq Ikbal, Dimiter Zlatanov, Matteo Zoppi

- **Published in:** Zeghloul S., Laribi M., Sandoval Arevalo J. (eds) Advances in Service and Industrial Robotics. RAAD 2020. Mechanisms and Machine Science, vol 84. Springer, Cham.

- **URL:** https://doi.org/10.1007/978-3-030-48989-2_45
- **Abstract:** Wearable robotics is a rapidly evolving field with assistive exoskeletons being developed for medical and industrial purposes. These exoskeletons can be used for gait rehabilitation of patients, arm rehabilitation and provide body support during training. These exoskeletal systems are anthropomorphically structured mechanisms where the rigid links form a serial kinematic chain and the cables are attached in a parallel configuration. Cable-driven system eliminates the need of rigid links and mechanical joints, making the system lightweight. Due to the serial unilateral constraints, it has to be ensured that all cables remain in tension at any point of time for system functionality. This results in redundancy in the actuation system. The aim of this paper is to provide the design and analysis of these multilink cable-driven robots (MCDRs). From the information of workspace, we can check whether the cable driven exoskeleton is able to perform the desired behavior (gait motion) with positive cable tensions and without violating defined constraints.

Publications in review process

C.5 HEDRA: A Bio-Inspired Modular Tensegrity Soft Robot With Polyhedral Parallel Modules

- **Authors:** Vishal Ramadoss, Keethi Sagar, Mohamed Sadiq Ikbal, Jesus Hiram Lugo Calles, Matteo Zoppi
- **Submitted to:** IEEE International Conference on Robotics and Automation (ICRA), 2021. Received November 2, 2020.
- **Abstract:** There is a surge of research interest in the field of tensegrity robotics. Robots developed under this paradigm provide many advantages and have distinguishing features in terms of structural compliance, dexterity, safety, and weight reduction. This paper proposes a new robotic mechanism based on tensegrity ('tension-integrity') robots and reconfigurable modular robots. The specific actuation schemes for this tensegrity robot with multiple degrees of freedom are presented. This article describes an easy-to-assemble 350 mm tensegrity based robot prototype by stacking a series of rigid struts linked with tensegrity joints that have no direct rigid contact with each other. The functionality of the proposed robot is validated by the experimental results by

integrating the polyhedral parallel structure as its skeleton and series of tensegrity joints. The proposed manipulator is capable of reaching bending angles up to 76 degrees. An adaptive cable driven underactuated robotic gripper is designed and attached to the tensegrity manipulator for grasping objects in different shapes, weights, and sizes.

C.6 Modeling and Stiffness Evaluation of Tendon-Driven Robot for Collaborative Human-Robot Interaction

- **Authors:** Vishal Ramadoss, Keerthi Sagar, Mohamed Sadiq Ikbal, Dimiter Zlatanov, Matteo Zoppi

- **Submitted to:** The proceedings of the 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR) (IEEE/ISR 2021), March 4-6, 2021, Nagoya, Japan

- **Abstract:** This paper presents a clear and detailed mathematical analysis of the stiffness evaluation of tendon-driven serial kinematic chains which allow single-point and multi-point routing with bundling per rigid link. The kinematic analysis of tendon driven serial chain with cable routing between links, re-routing within links and bundling is presented. Targeting a musculoskeletal two-link system with different tendon arrangements as a case study, the stiffness of each system is evaluated using a stiffness ellipse. The stiffness performance is analyzed to provide a holistic overview of passive stiffness and force production capabilities that have a pivotal role in the rehabilitation paradigm and human robot-interaction.

Appendix D

A list of videos produced for this thesis

- Dynamic pose tracking performance of HTC Vive lighthouse tracking system using an industrial serial robot
https://youtu.be/_PXVfkDHIGw

- SP7 flight simulator - Pilot Free Flight - MCA
https://youtu.be/HsL4DpKzB2M

- SP7 flight simulator - Subjective evaluation of MCA - Experiment run through
https://youtu.be/CmOhWla3blo

- SP7 Motion Simulator video presentation ACA and MCA
https://youtu.be/BHR2o6rgFPg

- DT implementation of motion cueing algorithm for real time applications
https://youtu.be/xWKiMC3CiSQ

- Experiment with SP7 flight simulator- Pilot 1 - MCA - ICRA Submission (1/4)
https://youtu.be/dg0VR5NCee8

- Experiment with SP7 flight simulator- Pilot 1 - ACA - ICRA Submission (2/4)
https://youtu.be/PwQWUqqrfTY

- Experiment with SP7 flight simulator- Pilot 2 - MCA - ICRA Submission (3/4)
https://youtu.be/GvnKVY8Jgkk

- Experiment with SP7 flight simulator- Pilot 2 - ACA - ICRA Submission (4/4)
https://youtu.be/yMPb1553K5M