

POLITECNICO DI TORINO

Master's Degree in Ingegneria del Cinema e dei Mezzi
di Comunicazione



Master's Degree Thesis

Interfaccia di Locomozione a Mappatura di Pressione su Seduta per Realtà Virtuale

Supervisors

Prof. Andrea Giuseppe BOTTINO

Dott. Francesco STRADA

Candidate

Leonardo VEZZANI

Luglio 2022

Summary

In questa tesi proponiamo un nuovo approccio alla locomozione in realtà virtuale illustrando come è stata sviluppata un'interfaccia a mappatura di pressione da posizionare su una sedia, per ricavare la posizione del torso dell'utente nello spazio. Con i dati ricavati si procede a controllare un avatar virtuale delegando il controllo di direzione e velocità di movimento all'inclinazione del busto dell'utente.

In questo elaborato illustreremo le ragioni che ci hanno portato a questa scelta per poi passare ad analizzare quali sistemi di locomozione esistono attualmente nel panorama videoludico e sperimentale.

In seguito esporremo come è stata costruita l'interfaccia e l'hardware utilizzato per la realizzazione del dispositivo.

Successivamente analizzeremo la tipologia dei segnali generati e la loro elaborazione nell'applicativo.

Infine illustreremo l'ambiente di test e i risultati raccolti, confrontando l'interfaccia realizzata con altre tecnologie.

Concluderemo con un'analisi generale e delle riflessioni su possibili sviluppi e migliorie.

Acknowledgements

Il raggiungimento di questo traguardo non sarebbe stato possibile senza tutte le persone che hanno creduto in me e mi hanno accompagnato in questi anni. Voglio ringraziare i miei genitori, i quali mi hanno supportato e aiutato a completare questo percorso in un periodo difficile come quello degli ultimi anni: ringrazio mia madre, che mi ha aiutato a non perdere di vista le cose importanti; e ringrazio mio padre, che mi ha sempre spronato ad impegnarmi al massimo ogni giorno.

Ringrazio mia sorella, che mi ha aiutato a crescere e migliorare in una prospettiva ricca come quella accademica.

Ringrazio mio fratello, che mi ricorda l'importanza di nutrire i propri sogni e dedicarsi ad essi ogni giorno.

Un grosso ringraziamento va ai miei zii che sono stati un punto di riferimento in questi anni di crescita e cambiamento.

Ringrazio i miei cugini per l'affetto e il sostegno dimostratomi nonostante la distanza dalla nostra città natale.

Ringrazio anche i miei nonni che, sebbene non abbiano potuto vedere questo giorno, sono sicuro sarebbero stati orgogliosi di questo mio traguardo.

Ringrazio Angelica, che in tutti questi anni mi è stata vicino come nessun altro, nei momenti felici e in quelli più tesi ha sempre trovato il modo di sostenermi e spronarmi nonostante fossi solo un "*sognatore perennemente insoddisfatto*".

Ringrazio Francesco che, nonostante quel puzzone ci abbia traditi andandosene a Milano, è stato una presenza costante, tra avventure e disavventure, aiutandomi a nutrire e accrescere il mio amore per lo studio e l'ingegneria. Ringrazio Fabio, che con la sua gentilezza e il suo impegno è stato un compagno di studio come nessun altro, sempre pronto ad aiutarmi e consigliarmi nelle mie idee più ambiziose.

Voglio ringraziare anche i miei amici storici di Firenze, Viera, Äittam, Guiccia, Scalo, Mammonxzzz, Barto, Nicco e Tommy che in questi anni mi hanno tenuto compagnia tra una birra e una kill, da remoto e in presenza, regalandomi ore di spensieratezza.

Grazie anche a Silvia, Stefano, Cesare, Caffa, Simone, Amedeo, Mattia, Rosy e tutti gli UCCM, emigrati e non, con cui mi sono ritrovato a condividere questa avventura politecnica giorno per giorno.

Un ringraziamento va anche a tutti gli amici di Roma come Matteo, Alessandro, Silvia, Paola e molti altri, che mi accolgono a braccia aperte ad ogni occasione.

Voglio anche ringraziare tutte le persone che hanno intrapreso percorsi diversi e che ora son distanti, ma che hanno comunque caratterizzato questi anni di studio, tra lezioni e aperitivi, tra caffè e esami, rendendoli unici e indimenticabili.

Dulcis in fundo, ringrazio Yuki che negli ultimi mesi è diventata la parte più bella della mia giornata.

Table of Contents

List of Tables	VIII
List of Figures	IX
Acronyms	XI
1 Introduzione	1
2 Locomozione di Realtà Virtuale	5
3 La Seduta	10
4 Progettazione Board	14
5 Analisi numerica segnali	29
6 Calcolo del movimento	35
7 Test	43
8 Risultati	50
9 Conclusioni	53
Bibliography	56

List of Tables

3.1	Componenti seduta	12
4.1	ESP8266 scheda tecnica	15
4.2	MPU6050 scheda tecnica	16
4.3	HX711 scheda tecnica	17
7.1	Confronto dei principali studi relativi organizzati in classi.	45

List of Figures

3.1	Rappresentazione della pressione media su una seduta	10
3.2	Schema costruttivo seduta	13
4.1	ESP8266 prodotto da AzDelivery	15
4.2	Mpu6050 schema costruttivo	16
4.3	HX711 schema costruttivo	17
4.4	Schema costruttivo del dispositivo Walking Seat	18
5.1	Risposta lineare del sistema a spostamenti di peso lineari . .	30
5.2	Segnale x affetto da rumore	31
6.1	Illustrazione delle direzioni consentite con WSL	36
6.2	Confronto T_y curva camminata(sinistra) e T_y curva corsa(destra)	38
6.3	Curva T_x in valore assoluto	39
7.1	Scenari: S1 Movimento rettilineo, S2 Direction Controll, S3 Movimento disaccoppiato, S4 agilità, S5 Interazione con oggetti.	46

Acronyms

AI

artificial intelligence

HMD

HeadMounted Display

LET

Locomotion Evaluation Testbed

JST

Joystick technique

WST

Walking Stea Technique

LU

Locomotion Usuer

Introduzione

L'idea alla base di questa tesi è quella di sfruttare un comportamento naturale come il muoversi sulla sedia durante una sessione di gioco, osservabile nei giocatori meno esperti, e trasformarlo in una tecnica di locomozione per applicazioni di realtà virtuale. Si può osservare frequentemente durante una sessione di gioco di giochi come Mario Kart o di un qualunque altro gioco di gare automobilistiche, che i giocatori ruotino il controller o inclinino il busto durante una curva particolarmente impegnativa o in momenti tesi e avvincenti. Questo comportamento, che sembra essere più frequente nei giocatori meno esperti e nei bambini, è la chiave dell'intuizione alla base di questo progetto. Riuscire a sfruttare questo movimento naturale e spontaneo per controllare un avatar virtuale può apparire un mero esercizio tecnico in applicazioni desktop, ma potrebbe diventare un interessante approccio in applicativi VR data la natura immersiva del medium. La naturalezza con cui ogni giorno ci avvaliamo di sedie e poltrone e le innumerevoli azioni che compiamo da seduti ci forniscono un'ampia conoscenza e competenza nello stare seduti.

L'obiettivo di questa tesi è validare la possibilità di usare l'inclinazione del busto dell'utente per la navigazione di ambienti virtuali. Consentire una locomozione libera, semplice e con ridotti effetti collaterali è un punto fondamentale per la fruizione di applicazioni di realtà virtuale e uno dei principali problemi della realtà virtuale per come la conosciamo oggi.

La navigazione di un mondo digitale tramite un visore di realtà virtuale (Head Mounted Display) è oggi possibile grazie a HMD economici e tool di sviluppo accessibili. Creare un mondo virtuale per fruizione da HMD pone problematiche ben diverse da quelle affrontate negli ultimi decenni dagli sviluppatori di mondi virtuali fruiti da desktop. Lo sviluppo di ambienti per realtà virtuale (VR) richiede un design centrato sulle possibilità e limitazioni del binomio utente-HMD.

Le limitazioni fisiche, quali l'altezza dell'utente, la lunghezza delle braccia, la forma fisica e l'ambiente in cui l'utente si trova, e quelle percettive, come la diversa risposta propriocettiva, vestibolare e tattile, devono essere tenute in alta considerazione durante la progettazione dell'esperienza in VR. Spesso le soluzioni proposte per affrontare una di queste limitazioni comporta l'exasperazione di altre problematiche.

La locomozione in VR pone una serie di sfide facilmente risolvibili singolarmente ma raramente affrontate con una prospettiva globale. Nei sistemi di locomozione è necessario tenere a mente, in fase di sviluppo, le limitazioni dell'ambiente fisico in cui l'utente fruisce dell'esperienza: non solo pareti e mobili limitano il movimento dell'utente ma anche lo stesso HMD può imporre restrizioni legate al sistema di tracking e alla presenza di eventuali cavi.

Con queste limitazioni a mente, nei primi anni di sviluppo per il grande pubblico, sono state proposte soluzioni che hanno dato alla luce molteplici sistemi di locomozione più o meno diegetici. I più diffusi prevedono una qualche combinazione di teletrasporto, movimento col joystick e movimento diretto che rendono l'esperienza fruibile in modo soddisfacente, anche se non sempre esprimono al meglio il potenziale del sistema umano-HMD.

Altre soluzioni hanno fatto di necessità virtù creando applicativi che facessero leva sul sistema di locomozione come parte integrante e caratterizzante dell'esperienza: titoli come Gorilla Tag, Kayak VR: Mirage o Echo VR hanno

fatto del sistema di locomozione il loro punto di forza volontariamente rinunciando alla possibilità di creare un sistema di locomozione bipede realistico. Nel tentativo di trovare un sistema di locomozione universale e che assicurasse una maggiore immersività, si è ricercata una soluzione a questo problema progettando hardware apposito come, ad esempio, tapis roulant omnidirezionali che consentissero all'utente una maggiore libertà di movimento nel mondo virtuale pur restando in un limitato volume nel mondo fisico. Queste soluzioni, però, oltre a essere molto costose, richiedono la creazione di spazi dedicati alla realtà virtuale e una vera e propria "vestizione" per poterne fruire.

La soluzione proposta in questa tesi mira ad offrire all'utente un'esperienza facile da imparare, rapidamente accessibile ed economica che non richiede uno spazio dedicato. L'idea alla base di questa interfaccia prevede di sfruttare l'inclinazione del torso e della parte superiore del corpo durante le sessioni in VR stando comodamente seduti ed evitando così di aver bisogno di ampi volumi nel mondo fisico.

Creando un dispositivo capace di mappare la posizione del baricentro sulla seduta si può determinare l'inclinazione dello spazio del torso dell'utente e usarla come input per determinare direzione ed entità dello spostamento nel mondo virtuale.

La creazione di un cuscino o di un rivestimento da applicare ad una sedia o a una poltrona già presente nell'ambiente dell'utente medio, consentirebbe di avere un dispositivo sempre accessibile e dal costo ridotto con cui entrare e uscire da mondi virtuali con facilità.

Per validare questa ipotesi è necessario verificare la relazione tra l'utente e la percezione del sé sia nello spazio virtuale che in quello fisico al fine di capire quanto sia importante, in applicazioni di realtà virtuale, la percezione del corpo fisico e quanto la risposta propriocettiva possa aiutare ad avvicinare l'utente al proprio avatar virtuale.

Al fine di restituire un'interfaccia coerente con le capacità fisiche dell'utente

medio, è necessario analizzare come l'essere umano cammina, quale siano le sue abitudini e come la parte superiore ed inferiore del corpo siano in relazione tra loro.

Nei capitoli seguenti analizzeremo i principali punti inerenti la percezione del sé nello spazio per come la conosciamo oggi, procederemo verificando quali interfacce di locomozione esistono e quali sono quelle interfacce basate sullo stesso principio della seduta in esame. In seguito analizzeremo la progettazione e la creazione del dispositivo soffermandoci sulle componenti, il codice e i segnali generati dal device. Infine analizzeremo la Locomotion Evaluation Testbed soffermandoci sugli aspetti inerenti l'aggiornamento dalla versione Unity 2018 a 2020 e analizzeremo i dati raccolti in fase di test. A concludere alcune riflessioni su possibili miglioramenti e applicazioni future.

Locomozione di Realtà Virtuale

Percezione del movimento nell'essere umano

Per comprendere a pieno il problema della locomozione in ambiente virtuale è necessario analizzare la camminata umana nella sua complessità. Il sistema locomotorio umano è complesso e articolato e non si può analizzare la camminata limitandoci ad una determinata area del corpo: la camminata è un'azione che viene compiuta da tutto il corpo umano, dal sistema locomotorio così come dal sistema nervoso.

Il controllo dell'equilibrio esercitato dal sistema nervoso sul corpo durante la camminata consente, infatti, di eseguire quel movimento, più volte definito come "*caduta controllata*", a carico delle gambe, torso e braccia che ogni giorno esercitiamo.

Questo movimento ricco e articolato è frutto di un insieme di input e output che il nostro cervello processa per mantenere l'equilibrio e navigare lo spazio senza sovraccaricare i singoli canali percettivi. Per quanto vista e tatto siano fondamentali allo svolgimento della camminata in modo corretto, questi non sono gli unici stimoli che usiamo per camminare: la propriocezione, ossia la percezione e la consapevolezza della posizione del nostro corpo nello spazio, e le risposte vestibolari, ossia la percezione dell'equilibrio, ci forniscono informazioni ad ogni passo.

Uno dei motivi per cui determinati sistemi di locomozione generano forte motion sickness e nausea è proprio dovuto all'assenza di questi feedback

nel nostro corpo. Se gli stimoli cui siamo sottoposti non concorrono alla determinazione di un risultato univoco circa la nostra presenza dello spazio, il sistema nervoso risulta affaticato con conseguenze spiacevoli.

In diversi studi si riporta come il flusso ottico nelle esperienze virtuali non sia sufficiente a creare una illusione di movimento credibile: uno degli stimoli più utili a contribuire alla creazione di tale illusione è la risposta vestibolare. Un sistema di locomozione che sfrutta l'inclinazione del corpo potrebbe fornire una migliore illusione di movimento.

Inoltre, il movimento del corpo durante le attività ad intenso sforzo mentale è dimostrato aumentare l'engagement e il senso di presenza nel contesto di azione. Nelle applicazioni di realtà virtuale, tanto più l'utente si muove ed è fisicamente attivo, e tanto più questo movimento è riportato, descritto e concorde col movimento rappresentato dal flusso ottico, tanto più il senso di presenza sarà valido e opportunamente riportato.

Sistemi di Locomozione

Il problema della locomozione in ambienti virtuali è tanto attuale quanto complesso. La differenza tra i vincoli fisici e virtuali percepiti dell'utente può portare ad una rottura dell'immersività e della qualità della simulazione.

Le soluzioni ad oggi proposte in letteratura sono varie e forniscono soluzioni a diversi livelli di complessità e fedeltà. I sistemi di locomozione possono essere gestiti da diversi dispositivi di input quali:

- **Controller:** si usano i sensori e gli input nativamente montati sui controller dedicati
- **Indossabili:** si sfruttano sensori dedicati per tracciare la posizione nello spazio del corpo dell'utente
- **Dedicati:** sono dispositivi complessi sviluppati appositamente per il task in esame

Le implementazioni su ciascuna di queste tipologie di dispositivi possono essere plurime e integrare più tipologie di device. In questa tesi prenderemo in esame alcune delle più diffuse.

La prima di queste tecniche è denominata "*Arm Swinging*": a l'utente viene richiesto, stando fermo in posizione eretta, di dondolare le braccia lungo i fianchi simulando il movimento compiuto dalle braccia durante la camminata. La direzione di camminata può essere legata sia alla direzione di vista sia alla direzione di movimento delle braccia.

Un'altra tecnica di locomozione, analoga alla precedente, è la "*Walk in place*", dove l'utente indossa dei sensori alle caviglie capaci di fornire la posizione dei piedi nello spazio: camminando sul posto l'utente si muoverà in ambiente virtuale lungo la direzione indicata da un terzo sensore montato sulla schiena dell'utente.

Tra i dispositivi dedicati troviamo i tapis roulant omnidirezionali, grossi e ingombranti macchinari in cui l'utente viene assicurato a bracci meccanici che lo tengono in posizione, mentre il tapis roulant sotto ai suoi piedi gli permetterà di muoversi in piena libertà camminando. Questi sistemi consentono un'elevata fedeltà e risposta propriocettiva e vestibolare ma sono anche costosi e richiedono spazi dedicati.

Una delle soluzioni più comuni e diffuse prevede l'uso del joystick montato sui controller base forniti con gli HMD e ha due declinazioni principali: la prima prevede un movimento analogo a quello ampiamente diffuso nel mondo dei videogiochi, muovendo l'utente in modo continuo della direzione indicata dalla direzione del joystick; l'altro prevede il casting di un raggio nell'ambiente virtual e il "*teletrasporto*" dell'utente nella posizione indicata.

Sistemi di Locomozione Leaning

I sistemi basati su leaning per il movimento in realtà virtuale sono piuttosto rari. Secondo le nostre ricerche solo tre sistemi sono stati testati ad oggi per l'implmenetazione di movimento basato sull'inclinazione di parti o della

totalità del corpo dell'utente.

Il più semplice tra questi, e anche il più particolare è il sistema Joyman. Joyman prevede l'uso di tutto il corpo umano come un joystick. sfruttando una postazione basculante dotata di parapetto, l'utente deve inclinarsi nella direzione desiderata per muoversi in realtà virtuale. Questo sistema leaning ha dimostrato come questi sistemi possano fornire un soddisfacente livello di immersione e senso di presenza. Durante gli esperimenti riportati con Joyman viene descritto un comportamento ricorrente: gli utenti tendono ad esasperare l'inclinazione nella direzione indicata muovendo il torso oltre la posizione suggerita.

Questo comportamento non deve sorprendere: l'essere umano percepisce la propria posizione nello spazio riconoscendosi occupare un punto compreso tra la parte superiore del torso e il volto. Un sistema basato su inclinazione che considera tutto il corpo come equipollente a livello percettivo ignora la differente relazione tra una persona e la sua percezione del se e tra una persona e la percezione dei propri arti.

Un altro sistema di locomozione proposto in letteratura è denominato NaviChair.

Navichair prevede l'uso di una seduta basculante, capace di inclinarsi per indicare la direzione verso cui dirigersi. Anche questo sistema fallisce nel proporre una soluzione valida. Lo studio riporta una media di errore nella definizione della direzione di movimento elevata, probabilmente dovuta all'uso della seduta come sistema di puntamento non concorde all'orientazione del bacino nello spazio. Inoltre la difficoltà di usare la seduta con questa tecnologia rende navichair una implementazione discutibile per l'uso come dispositivo di input.

Uno studio successivo, condotto con l'uso di HMD illustra un miglioramento delle prestazioni e suggerisce un possibile miglioramento ulteriore avvalendosi di un sistema con seduta girevole.

Flexperch è un sistema di locomozione per il volo libero che consente

all'utente di usare il proprio peso e le proprie gambe come input. Sebbene lo studio in questione sia parziale e evidenzi alcuni errori di design nella seduta, è interessante notare come la posizione del corpo e delle gambe in realtà virtuale influenzi la percezione dell'azione compiuta.

In questo caso una seduta comune è stata preferita a Flexperch per comodità e senso di sicurezza e per la piacevolezza di esperienza durante il volo.

La Seduta

Distribuzione del peso

Al fine di progettare una seduta che ci consenta di avvalorare l'ipotesi illustrata è necessario conoscere come si distribuisca il peso dell'utente sulla seduta.

La distribuzione del peso di una persona su una qualunque seduta non è omogenea e dipende da molteplici fattori: il tipo di sedia, la forma fisica dell'utente e le sue condizioni di salute e le azioni compiute sulla stessa possono determinare diverse distribuzioni.

La distribuzione del peso di una persona su una seduta si manifesta come da

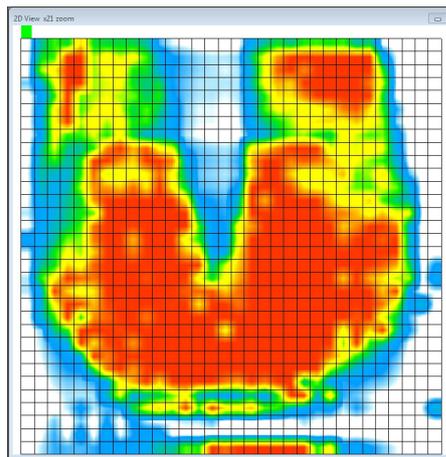


Figure 3.1: Rappresentazione della pressione media su una seduta

figura 3.1: si può vedere come la maggior parte della forza venga applicata in corrispondenza del bacino e come si possano distinguere nettamente la posizione delle gambe sulla superficie.

Inoltre è noto come la stessa sedia di cui ci si avvale modifichi la postura dell'utente e deve essere presa in considerazione come fattore determinante la distribuzione del peso corporeo.

Progettazione della Seduta

Esistono molte soluzioni industriali che consentono di ricavare queste informazioni: Tekscan si avvale del BPMS Body Pressure Measurement System per raccogliere ed illustrare la distribuzione del peso in diversi ambiti con elevata precisione. Utilizzare un dispositivo come il BPMS consentirebbe di ottenere molte informazioni circa non solo la posizione del baricentro dell'utente ma anche la posizione delle gambe, la variazione di peso esercitata da ciascuna gamba o ciascun gluteo e molto altro.

Gestire una mole di informazioni di questo tipo è sicuramente complesso ma avvalendosi di un campione di utenti cospicuo, e di un altrettanto cospicuo campione di sedie, si potrebbe mirare a creare un algoritmo che consenta non solo di riconoscere la posizione del baricentro dell'utente sulla seduta ma anche la posizione dell'utente stesso tramite l'uso di algoritmi di machine learning. Per quanto interessante questo approccio ricade al di fuori degli obiettivi di questa tesi.

Nel tentativo di ovviare alla complessità di una mappatura fine della seduta si è pensato di creare una seduta con diverse aree sensibili, capaci di leggere la distribuzione del peso per ciascuna gamba e gluteo: creare un cuscino con diversi sensori distribuiti sui lati, al centro e in altre posizioni strategiche avrebbe potuto dare risultati interessanti. Questa possibilità è stata scartata per l'elevata complessità: realizzare una seduta così articolata richiede una conoscenza tecnica non disponibile nel contesto di questa tesi.

Si è deciso dunque di ridurre il problema alla sua essenza al fine di verificare

l'ipotesi proposta riducendo la possibilità di problemi e imprevisti dovuti ad elementi eccessivamente complessi che non avrebbero contribuito ad avvalorare la tesi.

Il dispositivo realizzato per questa tesi presenta caratteristiche minime al fine di analizzare un caso dalla complessità minima. La seduta è composta da due superfici rigide distanziate da quattro celle di carico. Al centro della superficie della seduta è presente il giroscopio. Uno dei due lati della seduta è uniformemente imbottito per un maggiore comfort. L'utilizzo di una struttura rigida consente una ripartizione delle forze sui sensori più semplice grazie all'assenza di deformazioni che altererebbero la distribuzione del peso sui sensori. La seduta così realizzata è posizionabile su un qualsiasi tipo di sedia, sgabello o poltrona e consente la sperimentazione in contesti diversi. La seduta è stata quindi realizzata sotto forma di piattaforma composta da due stati di legno multistrato di che racchiudono quattro celle di carico posto agli angoli estremi dei pannelli, ciascuno affiancato da una molla distanziatrice capace di esprimere una forza di 2kg al fine di separare i pannelli in assenza di un utente. La circuiteria e il giroscopio sono stati montati al centro del pannello di legno inferiore.

1	pannello inferiore
2	distanziatori
3	dadi posizionamento
4	molle distanziatrici
5	celle di carico
6	board Arduino
7	alloggio cella di carico
8	pannello superiore imbottito

Table 3.1: Componenti seduta

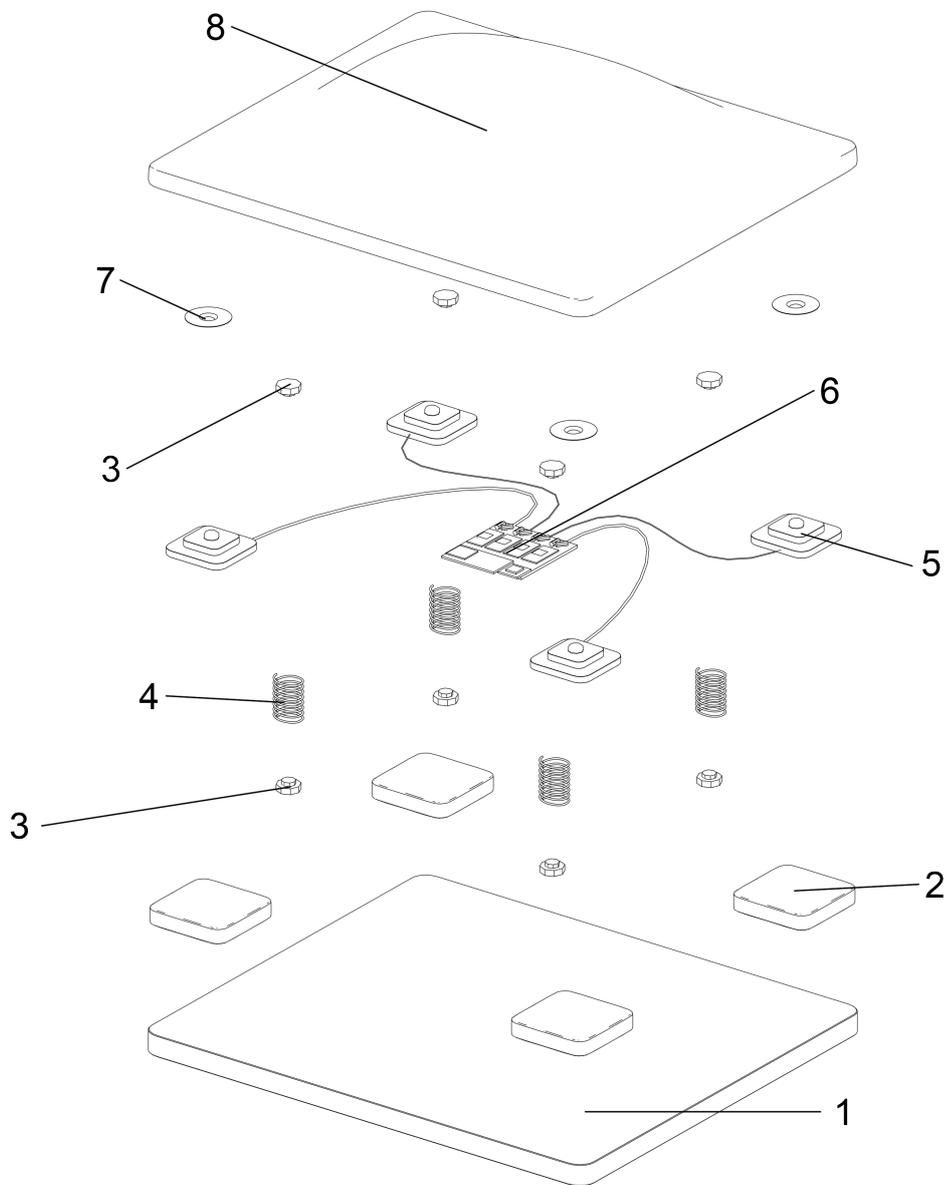


Figure 3.2: Schema costruttivo seduta

Progettazione Board

Hardware

La componente elettronica è stata realizzata utilizzando la piattaforma Arduino. Dopo diverse iterazioni e modifiche progettuali è stata trovata una soluzione ottimale nell'impiego delle seguenti schede:

1. Board ESP 8266 ESP-12F WiFi Lolin v3 AZDelivery
2. Giroscopio GY-521 MPU6050 ARCELI
3. Sensore pressione HX711 ARCELI
4. Cella di carico mezzo ponte HALJIA

L'uso di questi componenti è stato dettato dal ridotto costo e dalle funzioni fornite, che offrivano un'adeguata ricchezza di possibilità che ci hanno permesso di tentare anche approcci diversi da quelli illustrati nella soluzione raggiunta.

ESP8266 microcontrollore

ESP8266 è un microcontrollore con antenna WiFi (standard IEEE 802.11) integrata che supporta il protocollo TCP/IP.

Usare un microcontrollore con protocollo internet integrato ci consente di mantenere un maggior numero di GPIO disponibili per l'elevato numero di HX711 necessari. L'uso di protocollo TCP in questo sistema è sconsigliato rispetto a UDP, ma la vicinanza tra i dispositivi, il ridotto numero di errori e le criticità legate all'uso di UDP su Unity ci hanno portato a prediligere TCP.

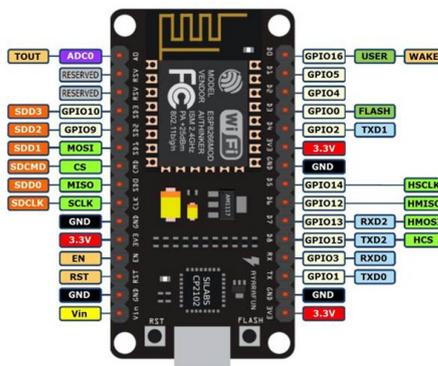


Figure 4.1: ESP8266 prodotto da AzDelivery

Microcontrollore	Tensilica 32-bit RISC CPU Xtensa LX106
Tensione funzionamento	3.3V
Tensione di input	7-12V
I/O Pin digitali (DIO)	16
Input Pin analogici (ADC)	1
UARTs	1
SPIs	1
I2Cs	1
Memoria Flash	4 MB
SRAM	64 KB
Clock	80 MHz

Table 4.1: ESP8266 scheda tecnica

MPU6050 Accelerometro e Giroscopio

Il giroscopio e accelerometro mpu6050 consente di ottenere informazioni circa l'orientazione e la velocità di rotazione nello spazio della componente. È inoltre dotato di un termometro che in questo applicativo non è stato usato.

Il giroscopio ci consente di sapere l'orientazione della seduta nello spazio, consentendo così di creare un sistema di locomozione che rispetti le differenze tra rotazione della testa e del bacino.

il giroscopio è stato montato al centro della seduta, equidistante dalle quattro celle di carico.

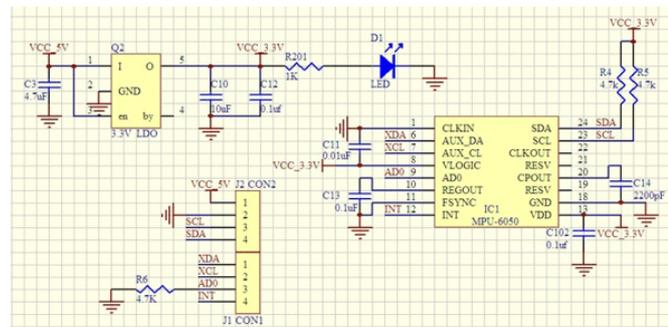


Figure 4.2: Mpu6050 schema costruttivo

Tensione funzionament	2.3–3.4 V
Consumi	3.9 mA max.
Intervalli di misura Accelerometro	± 2 g ± 4 g ± 8 g ± 16 g
Tolleranza di calibrazione Accelerometro	$\pm 3\%$
Intervallid i misura Giroscopio	$\pm 250/500/1000/2000$ °/sar
Tolleranza di calibrazione Giroscopio	$\pm 3\%$
Interfaccia	I2C
Sensore di temperatura	si
Temperature di funzionamento	-40 °C to +85 °C

Table 4.2: MPU6050 scheda tecnica

HX711 ADC sensore pressione

Per leggere correttamente il valore di una cella di carico si misura la variazione della resistenza costituita dalla cella stessa. Al fine di misurare correttamente una variazione piccola in modulo come quella della cella usata, si costruisce un ponte di wheatstone al fine stabilizzarne la lettura. Nel nostro caso si sono usate delle resistenza fisse da 1kohm.

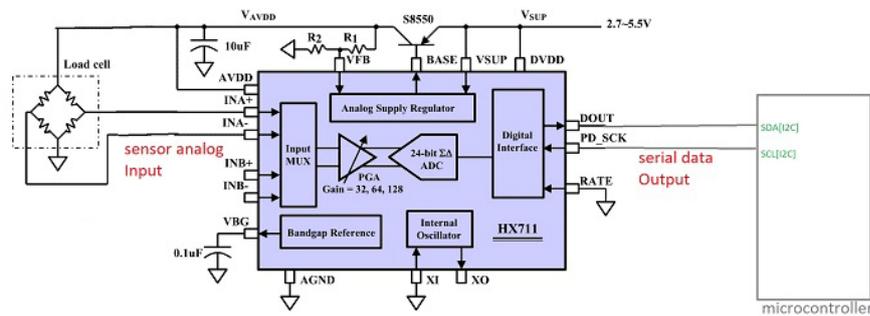


Figure 4.3: HX711 schema costruttivo

Tensione funzionamento	2.6V to 5.5 V
Consumi	< 1.5mA mA max.
PGA basso rumore con guadagno variabile	32,64,128bit
frequenza dati uscita	10-80SPS
Canalidi ingresso differenziabili	2
Temperature di funzionamento	-40 °C to +85 °C

Table 4.3: HX711 scheda tecnica

Schema Board

La board viene assemblata come in figura 4.4: i sensori condividono porta di alimentazione e ground. la conversione da porte analogiche a digitale consente la costruzione dello schema come illustrato.

Di fondamentale importanza la costruzione del ponte di wheatstone con resistenze costanti, al fine di avere un valore leggibile su un numero ridotto di celle di carico.

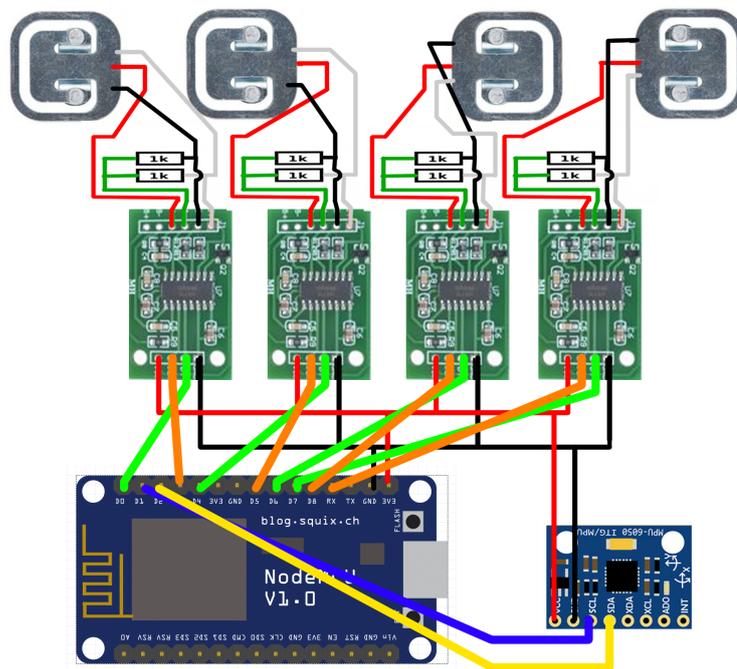


Figure 4.4: Schema costruttivo del dispositivo Walking Seat

Software

Il codice, riportato integralmente, si occupa di registrare i segnali ricevuti dai sensori e trasformarli in un segnale grezzo da trasmettere all'applicativo desktop tramite socket TCP. Il segnale in uscita è composto dai cinque valori x , y , dx , dy e vel a cui vengono affiancati, in fase di trasmissione, anche i valori misurati su ciascuna cella. Dx e dy sono le derivate calcolate sui valori discreti rispettivamente di x e y .

```
1 #include <MPU6050_tockn.h>
2 #include <HX711_ADC.h>
3 #include <ESP8266WiFi.h>
4 #include <Wire.h>
5 #include <WiFi.h>
```

Oltre ad includere le librerie ottimizzate per ciascuna componente indicata sulle relative guid tecniche è necessario includere libreria *Wire.h* e *Wifi.h*. *Wire.h* è necessaria per utilizzare la porta seriale RX come porta GPIO 3: dal momento che la board ha bisogno di un totale di 10 porte dati (4 coppie per le celle di carico e una coppia per il giroscopio) è stato necessario ricorrere alle porte seriali della board. *WiFi.h* è necessaria per accedere alle funzionalità TCP della board ESP8266 e avere il massimo controllo sul protocollo TCP.

```
1 /*
2   HX711 - SCALES
3  */
4 const int SA_sck = 14; //D5
5 const int SA_out = 2; //D4
6 const int SB_sck = 15; //D8
7 const int SB_out = 12; //D6
8 const int SC_sck = 3; //RX
9 const int SC_out = 13; //D7
```

```

10 const int SD_sck = 0; //D3
11 const int SD_out = 16; //D0
12
13 /*
14  CELLE DI CARICO
15 */
16 HX711_ADC LoadCellA(SA_out, SA_sck);
17 HX711_ADC LoadCellB(SB_out, SB_sck);
18 HX711_ADC LoadCellC(SC_out, SC_sck);
19 HX711_ADC LoadCellD(SD_out, SD_sck);
20
21 float a, b, c, d, velAng = 0;
22 float x, y, dx, dy, xOld, yOld = 0.0f;

```

Per ciascuna cella viene poi istanziato un elemento *HX711_ADC*, responsabile della lettura del controllo sull'omonimo sensore, fornendo le porte di serial clock e di data. successivamente allochiamo la memoria necessaria per manipolare i dati in ingresso (riga 27) e per calcolare i dati in uscita (riga 28).

```

1
2 /*
3  ESP8266
4 */
5 WiFiServer wifiServer(34197);
6
7 const char* ssid = "XRNetwork";
8 const char* password = "Evipcinema_xr22";
9
10 long tSent = 0;
11 int datagramPerSeconds = 39;
12 long datagramInterval = 1;

```

Successivamente istanziamo la rete WiFi a cui il dispositivo andrà ad accedere fornendo la porta, ssid e password della rete locale. L'indirizzo IP

viene assegnato staticamente dal router per ragioni di gestione della rete locale. Vengono inoltre allocati valore per il controllo del throughput sul socket. *DatagramPerSecond* è responsabile del numero di pacchetti trasmessi al secondo ed è calcolato come:

$$datagramPerSeconds = frameRate + DenoiserWindow - 1 \quad (4.1)$$

```
1 /*  
2  MPU6050 TOCKN  
3 */  
4 MPU6050 mpu6050(Wire);
```

Infine procediamo a creare l'istanza *MPU6050* che sarà responsabile in fase di setup di inizializzare le porte del sensore omonimo.

```
1 void setup() {  
2   pinMode(SC_sck, FUNCTION_3);  
3   Serial.begin(115200);
```

In fase di setup si procede col convertire il pin RX seriale in GPIO tramite l'apposita funzione e successivamente si apre la comunicazione seriale TX per esigenze di debug: la sovrascrittura della funzionalità della porta *SC_sck* impedirà la ricezione da parte della board di comandi runtime tramite comunicazione seriale, lasciando la trasmissione di linee di debug possibile.

```
1 /*  
2  CALIBRATION HX711  
3 */  
4 LoadCellA.begin();  
5 LoadCellB.begin();
```

```
6 LoadCellC.begin();
7 LoadCellD.begin();
8
9 float calibrationValue = 6000;
10 long stabilizingtime = 50000;
11 boolean _tare = true;
12
13 Serial.print("Calibrating...");
14
15 LoadCellA.start(stabilizingtime, _tare);
16 if (LoadCellA.getTareTimeoutFlag()) Serial.println("Timeout,
17     check A MCU>HX711 wiring and pin designations");
18 else LoadCellA.setCalFactor(calibrationValue);
19
20 LoadCellB.start(stabilizingtime, _tare);
21 if (LoadCellB.getTareTimeoutFlag()) Serial.println("Timeout,
22     check B MCU>HX711 wiring and pin designations");
23 else LoadCellB.setCalFactor(calibrationValue);
24
25 LoadCellC.start(stabilizingtime, _tare);
26 if (LoadCellC.getTareTimeoutFlag()) Serial.println("Timeout,
27     check C MCU>HX711 wiring and pin designations");
28 else LoadCellC.setCalFactor(calibrationValue);
29
30 LoadCellD.start(stabilizingtime, _tare);
31 if (LoadCellD.getTareTimeoutFlag()) Serial.println("Timeout,
32     check D MCU>HX711 wiring and pin designations");
33 else LoadCellD.setCalFactor(calibrationValue);
```

Successivamente si procede con l'inizializzazione delle celle di carico e dei relativi sensori: ogni sensore viene avviato, vengono definiti i parametri e i tempi di calibrazione per procedere alla tara di ciascuna cella.

Ciascuna cella viene avviata indicando i parametri di tara e effettuati controlli sulla risposta. In caso di una o più celle non correttamente avviate sarà necessario riavviare la board e, se il problema persiste, verificare le saldature

della stessa.

```
1  /*
2     CALIBRATION MPU6050
3  */
4  Wire.begin(4, 5);
5  mpu6050.begin();
6  mpu6050.calcGyroOffsets(true);
```

Si procede poi a avviare il giroscopio usando i metodi forniti dalle librerie MPU6050_tockn.h e Wire.h che ci consentono di inizializzare e tarare il giroscopio.

```
1  /*
2     SET FRAMERATE
3  */
4  datagramInterval = 1000 / datagramPerSeconds;
5  Serial.print("  datagram interval is :");
6  Serial.println(datagramInterval);
```

Tramite una semplice divisione si procede ad impostare l'intervallo di tempo tra un pacchetto e il successivo; intervallo durante il quale il dispositivo si occuperà di raccogliere dati.

```
1  /*
2     TCP
3  */
4  WiFi.begin(ssid, password);
5  while (WiFi.status() != WL_CONNECTED) {
6     Serial.println("Connecting..");
7     delay(1000);
8  }
```

```
9  
10 Serial.print("Connected to WiFi. IP:");  
11 Serial.println(WiFi.localIP());  
12 wifiServer.begin();  
13 }
```

Infine, si procede a aprire la connessione con la rete WiFi. In caso di insuccesso il dispositivo effettuerà tentativi a oltranza. In caso di riuscito handshake si procederà a fornire l'IP assegnato e ad aprire il server a cui l'applicativo dovrà connettersi.

```
1 void loop()  
2 {  
3   WiFiClient client = wifiServer.available();  
4   client.setNoDelay(true);
```

Nel loop, che viene eseguito ciclicamente, si procede a verificare la presenza di un client e a impostare la comunicazione con lo stesso come asincrona, al fine di inviare i pacchetti nell'istante in cui i calcoli vengono completati e con la frequenza desiderata.

```
1  
2 while (client) {  
3   int whileCounter = 0;  
4   float xSum = 0.0 f;  
5   float ySum = 0.0 f;  
6   float dxSum = 0.0 f;  
7   float dySum = 0.0 f;  
8   float velAngSum = 0.0 f;  
9  
10  do {  
11    /*  
12     SCALE  
13    */
```

```

14     if (LoadCellA.update()) a = (LoadCellA.getData());
15     if (LoadCellB.update()) b = (LoadCellB.getData());
16     if (LoadCellC.update()) c = (LoadCellC.getData());
17     if (LoadCellD.update()) d = (LoadCellD.getData());
18
19     x = (a + d) - (c + b);
20     y = (c + d) - (a + b);
21
22     xSum += x;
23     ySum += y;
24
25     if (x != xOld) dx = x - xOld;
26     if (y != yOld) dy = y - yOld;
27
28     dxSum += dx;
29     dySum += dy;
30
31     xOld = x;
32     yOld = y;
33
34     mpu6050.update();
35     velAngSum += mpu6050.getGyroZ();
36
37     whileCounter++;
38 } while ( millis() < tSent + datagramInterval-3); // 3
millis = tempo di elaborazione e invio

```

Verificata la presenza di un client e fintanto che il client resta connesso si procede a calcolare il segnale in uscita. Si azzerano i valori cumulativi, definiti dal suffisso *SUM*: In questi valori vengono sommate le singole misure effettuate nell'intervallo di tempo definito da *datagramPerSeconds* che verranno poi divise per il numero di cicli effettuati nel *while* applicando così un primo denoising a media.

In questo passaggio si calcolano anche le *dx* e *dy* dopo aver verificato l'aggiornamento corretto del valore delle celle. Lo stesso processo viene

applicato alla misura della velocità di rotazione del giroscopio.

Si contano infine il numero di cicli effettuati nell'intervallo di misura stabilito.

Si noti come vengono sottratti 3 millisecondi all'intervallo di misura: questi rappresentano il tempo necessario all'esecuzione dei comandi successivi e devono essere tenuti in considerazione per un corretto controllo del throughput sul socket.

```
1    x = xSum / whileCounter ;
2    y = ySum / whileCounter ;
3
4    dx = dxSum / whileCounter ;
5    dy = dySum / whileCounter ;
6
7    velAng = velAngSum/whileCounter ;
```

Una volta effettuate le misurazioni si procede a dividere i valori per il numero di cicli di misura, ottenendo così un segnale in uscita affettoda minor rumore.

```
1    byte DataBuffer [36] = {
2        ((uint8_t*)&x) [0] ,
3        ((uint8_t*)&x) [1] ,
4        ((uint8_t*)&x) [2] ,
5        ((uint8_t*)&x) [3] ,
6
7        ((uint8_t*)&y) [0] ,
8        ((uint8_t*)&y) [1] ,
9        ((uint8_t*)&y) [2] ,
10       ((uint8_t*)&y) [3] ,
11
12       ((uint8_t*)&dx) [0] ,
13       ((uint8_t*)&dx) [1] ,
14       ((uint8_t*)&dx) [2] ,
15       ((uint8_t*)&dx) [3] ,
16
```

```
17 ((uint8_t*)&dy)[0],
18 ((uint8_t*)&dy)[1],
19 ((uint8_t*)&dy)[2],
20 ((uint8_t*)&dy)[3],
21
22 ((uint8_t*)&a)[0],
23 ((uint8_t*)&a)[1],
24 ((uint8_t*)&a)[2],
25 ((uint8_t*)&a)[3],
26
27 ((uint8_t*)&b)[0],
28 ((uint8_t*)&b)[1],
29 ((uint8_t*)&b)[2],
30 ((uint8_t*)&b)[3],
31
32 ((uint8_t*)&c)[0],
33 ((uint8_t*)&c)[1],
34 ((uint8_t*)&c)[2],
35 ((uint8_t*)&c)[3],
36
37 ((uint8_t*)&d)[0],
38 ((uint8_t*)&d)[1],
39 ((uint8_t*)&d)[2],
40 ((uint8_t*)&d)[3],
41
42 ((uint8_t*)&velAng)[0],
43 ((uint8_t*)&velAng)[1],
44 ((uint8_t*)&velAng)[2],
45 ((uint8_t*)&velAng)[3]
46
47 };
```

Si procede creando il pacchetto da inviare allocando i singoli byte. questo approccio è reso necessario dall'uso del protocollo TCP senza sovrastrutture al fine di mantenere una misura di pacchetto costante ed evitare variazioni di lunghezza dovute all'alternanza di segni positivi e negativi presenti in altre

formattazioni.

```
1   client.write(DataBuffer, 36);  
2   tSent = millis();  
3   delay(5);  
4   }  
5 }
```

Infine, si invia il pacchetto così confezionato e si registra il tempo di invio per la misurazione dell'intervallo di misura successivo.

si introduce inoltre un ritardo di 5ms al fine di assicurare un aggiornamento dei valori dei sensori dall'ultima misura effettuata.

Analisi numerica segnali

Obiettivi

Al fine di controllare un avatar virtuale usando l'inclinazione del torso è necessario conoscere la direzione dell'inclinazione e l'entità dell'inclinazione. Altre informazioni utili sono la velocità di variazione di posizione e il peso dell'utente.

La posizione del baricentro sulla seduta viene registrata in relazione alla posizione delle celle. Quattro celle posizionate a disegnare un quadrato sono in grado di fornire la posizione del baricentro di un qualunque greve posizionato al loro interno tramite triangolazione.

In questo caso non è possibile triangolare un punto fuori dall'area descritta dalle celle a causa della rigidità del pannello superiore: l'applicazione di un greve fuori dalla sopracitata area porterà il pannello ad inclinarsi e a separare almeno due celle di carico dal corrispettivo punto di appoggio, portando il numero di punti di triangolamento sotto la quantità minima necessaria per una triangolazione di precisione.

Inoltre, nel caso in cui si desideri descrivere il punto come una coppia di coordinate (x,y) , nel caso in cui si abbiano quattro celle distribuite in un quadrangolo rettangolo, sarà sufficiente fare la differenza tra la somma di celle adiacenti per ricavare (x,y) ;

$$x = (a + b) - (c + d); y = (b + c) - (d + a); \quad (5.1)$$

Celle di carico

Uno dei primi elementi da valutare per interpretare correttamente i valori registrati è verificare che tipo di rapporto sussiste tra la posizione del baricentro e la posizione registrata. Per verificare la funzione di trasferimento abbiamo usato come elemento di test una grave di 7kg con superficie di appoggio di $9,42\text{cm}^2$ (un manubrio da palestra). Posizionando il grave in diverse posizioni misurando la distanza dal centro e registrando questi valori si sono raccolti i seguenti dati:

Come si può vedere, allo spostarsi del peso tra diverse posizioni il seg-

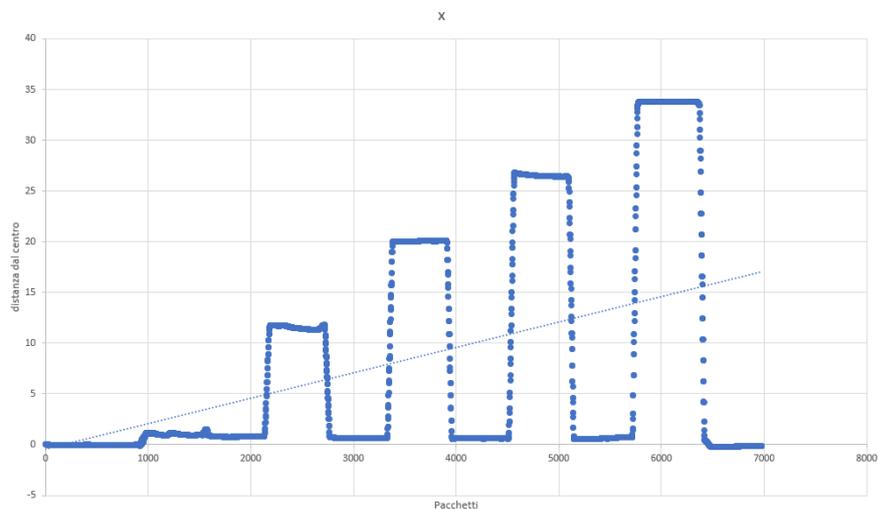


Figure 5.1: Risposta lineare del sistema a spostamenti di peso lineari

nale registrato cresce linearmente: in questo caso si provvedeva a spostare e rimuovere il corpo di test in posizioni distanti tra loro di 2 centimetri. la posizione registrata segue una legge lineare come il grave posizionato.

La scala non è proporzionale alle misure effettive della seduta; la coppia di valori (x, y) registrata, dunque, non indica la posizione in centimetri sulla seduta del baricentro. Si è preferito mantenere la scala di misura quanto più ampia possibile per mantenere una precisione elevata nei calcoli successivi.

Si è proceduto, dunque, a verificare il comportamento del segnale in quiete e sotto stress, al fine di identificare gli eventuali rumori e derive che affliggono il sistema al fine di correggerli.

Per verificare il comportamento delle celle si è proceduto a lasciare il sistema funzionante in quiete per un esteso periodo di tempo. La presenza delle molle distanziatrici assicura l'assenza di pressione sui sensori. Il sistema in quiete è affetto da una leggera deriva, non superiore a $0.0001\text{grammo}/\text{sec}$ che non rappresenta un problema per la lettura del dato, visto il modulo molto minore del valore registrato.

Per misurare correttamente il rumore dei segnali x e y si è proceduto ad analizzare il rumore delle singole celle e del segnale complessivo.

come si evince dal grafico, le singole celle sono affette da rumore bianco a

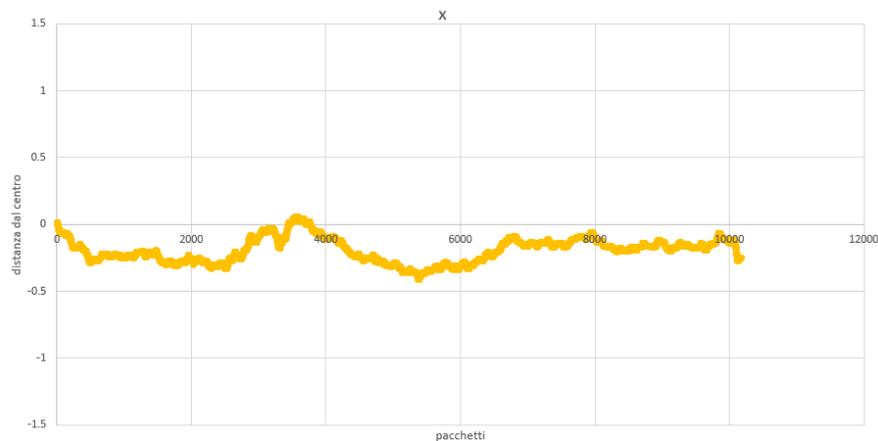


Figure 5.2: Segnale x affetto da rumore

media nulla, di modulo sufficientemente piccolo da essere un considerato un o-piccolo del segnale.

I segnali x e y , analogamente, presentano un rumore bianco a media nulla ma di modulo maggiore. Al fine di ridurre questo rumore che influenzerebbe la stabilità di alcuni calcoli successivi si è implementato un filtro a media applicato lato dispositivo prima di trasmettere la misura all'applicazione.

Come già illustrato nel capitolo precedente, il filtro consiste in una sommatoria del segnale nella finestra di tempo che intercorre tra una trasmissione e la successiva. Durante i 25ms che intercorrono tra due pacchetti TCP, il dispositivo legge il valore delle celle, calcola gli output x e y e li somma tra loro. Al termine della misurazione si divide il valore cumulato per il numero di iterazioni e si ottiene così la media del valore sull'intervallo desiderato. Grazie all'elevata frequenza di campionamento si riesce ad ottenere un segnale stabile e affetto da rumore minimo, su cui si può procedere a calcolare le derivate senza che queste siano affette da rumori eccessivi.

Altro elemento da tenere in considerazione è la velocità di applicazione delle forze. È importante sapere se la velocità di variazione del peso può influenzare la misura del peso stesso. Per verificare ciò si è proceduto in due direzioni parallele: in primis si è verificato se muovendo rapidamente un grave posato sulla seduta le variazioni fossero correttamente registrate, in secundis si è proceduto a verificare se il posizionamento di pesi più o meno velocemente potesse influenzare la lettura dei dati.

La variazione di posizione rapida di un peso all'interno dell'area di rilevamento viene correttamente registrata dai sensori e non sono state identificate velocità minime o massime oltre le quali il segnale risultasse incapace di leggere correttamente il valore. Per quanto riguarda, invece la velocità di applicazione delle forze si sono riscontrati dei limiti non eliminabili: dal momento che le celle di carico riescono a registrare al massimo 50kg di peso ciascuna, l'applicazione di forze maggiori genera una saturazione della cella sul fronte di salita che comprime l'intero sistema. Infatti, alla saturazione della cella sul fronte di salita corrisponde un mancato ritorno a 0 sul fronte di discesa, portando così il sistema a calcolare i segnali in uscita in modo errato. L'applicazione di forze impulsive è dunque da evitare durante l'uso di questa specifica implementazione della Walking seat. Conseguentemente si è proceduto ad eliminare dalle feature implementate la possibilità di saltare

sulla sedia per far saltare l'avatar virtuale.

Una volta verificata la pulizia e correttezza del segnale si è implementato un calcolo della derivata sui singoli campioni. La necessità di calcolare la derivata nasce dalla natura del segnale misurato, che impiega un tempo considerevole per passare da positivi a negativi. Questo transitorio, dovuto alla qualità delle celle di carico, è trattato grazie all'uso di un algoritmo PID, illustrato del capitolo 6.

Giroscopio

Il sensore MPU6050 offre la possibilità di leggere sia dati dal giroscopio che dall'accelerometro. In questa applicaione si è prediletto l'uso del giroscopio che resituisce all'applicaizone la velocità di rotazione intorno all'asse prescelto in ciascun istante.

Usare la velocità, espressa in *grad/sec*, consente una trasmissione rapida ai metodi di controllo della rotazione in Unity.

Il segnale registrato è affetto da rumore bianco a media nulla corretto dal filtraggio a media applicato durante la lettura tra un pacchetto e il successivo.

TCP

Per questo dispositivo si è prediletto l'uso di un socket web per la trasmissione dei dati. La creazione di un device bluetooth è stata esclusa per l'elevata complessità dei dispositivi coinvolti e per consentire una pipeline di comunicazione semplice ed efficace.

Usando una board ESP8266 con TCP/IP integrato si è deciso di sfruttare il protocollo sopracitato. Il protocollo TCP (Transmission Control Protocol) è un protocollo di rete che si occupa del controllo della trasmissione rendendo affidabile la comunicazione tra i dispositivi coinvolti. Questo protocollo offre un controllo sulla trasmissione e previene la perdita di pacchetti. Data la natura del dispositivo, l'uso del protocollo UDP sarebbe sembrata la scelta

più ovvia, ma l'implementazione non è stata possibile. Esistono librerie che consentono l'implementazione di UDP su ESP e alcune prove preliminari mostravano una buona qualità di trasmissione, dovuta anche alla prossimità tra i dispositivi. Nonostante il socket UDP venisse correttamente impostato e i pacchetti venissero correttamente intercettati e letti dal software di sniffing *wireshark*, il server UDP in unity non riusciva a ricevere correttamente i dati. La rimozione dei firewall che bloccano pacchetti UDP in ingresso e in uscita in unity non hanno alterato l'esito del tentativo.

Il socket TCP mantiene delle ottime performance: grazie alla prossimità tra device e computer, l'utilizzo di un protocollo TCP asincrono non bloccante le performance sono ottime: la board invia un pacchetto ogni $25ms$, in media i pacchetti vengono ricevuti ogni $25.59ms$ con una deviazione standard di $0.19ms$. Nelle analisi preliminari condotte *wireshark* ha rivelato una percentuale di errori di trasmissione inferiore allo 0.0001% in un intervallo di 33.34 min.

Per favorire delle performance migliori abbiamo anche provveduto a separare le frequenze della rete locale, in modo da dedicare le frequenze WiFi 2.4Ghz al dispositivo walking seat e le frequenze a 5Ghz alla trasmissione tra HMD e computer.

Calcolo del movimento

Al fine di trasdurre il segnale grezzo in input chiari per l'utente si è proceduto seguendo un processo di sviluppo agile con numerosi test durante tutto il processo di design.

Nelle prime fasi di sviluppo si è tentato di utilizzare il baricentro in modo perfettamente analogo al joystick di un controller, usando un vettore bidimensionale per controllare i movimenti dell'avatar. Questo primo approccio estremamente semplicistico forniva numerosi spunti di riflessione.

Sebbene il dispositivo apparisse intuitivo il suo uso si è presto rivelato fallace e controproducente: consentire all'utente di muoversi uniformemente in ogni direzione indipendentemente da entità dell'inclinazione e posizione fisica dell'utente portava ad una scarsa immersione e un difficile controllo del movimento.

Alla luce delle prime sperimentazioni si è valutata l'ipotesi di usare un sistema ibrido, che delegasse una porzione di controlli ai controller dell'HMD o altri input dal lato dell'utente. Al fine di mantenere l'esperienza quanto più immersiva e semplice possibile si è giunti al compromesso di delegare al controller destro una parte della modulazione della velocità.

È stato anche necessario programmare un tasto di reset, data la possibilità di disallineamenti tra la seduta virtuale e la seduta fisica, dovuti a movimenti involontari dell'utente sulla seduta e la possibilità non nulla, sebbene rara, di ritardi sul socket di trasmissione, che consentisse di riallineare la

seduta virtuale all'headset: in caso di necessità l'utente è dunque abilitato a posizionarsi composto sulla sedia e riallineare la seduta virtuale rapidamente. Si è proceduto quindi ad implementare delle aree di stasi, in modo tale da fornire all'utente dei margini di spostamento sulla seduta senza che il sistema rispondesse a delle variazioni di modulo minore alla soglia di sensibilità. La possibilità di muoversi in ogni direzione, ha costituito un problema a livello percettivo e propriocettivo: muoversi in determinate direzioni, in particolare movimenti laterali puri, costituisce un movimento molto raro per l'essere umano e consentire un movimento del genere si è dimostrato essere *oleoso* ed estremamente innaturale, trasformando il sistema di locomozione in un sistema più adatto alla guida in un gioco arcade.



Figure 6.1: Illustrazione delle direzioni consentite con WSL

Alla luce di queste considerazioni si è optato per consentire all'utente di muoversi principalmente lungo l'asse postero-frontale con un margine di correzione lungo l'asse destra-sinistra al fine di consentire una maggiore flessibilità in curva e nel controllo di movimenti morbidi e continui o, più in generale, su tutti i controlli che potrebbero risentire del movimento del torso

durante la rotazione intorno all'asse di rotazione della sedia.

Questa scelta è stata compiuta anche per aumentare la similitudine tra la risposta vestibolare durante la camminata fisica, legata all'inclinazione del busto in avanti o indietro, con l'inclinazione necessaria durante l'uso della seduta.

Uso del PID

Il PID o controllo Proporzionale-Integrale-derivativo è un algoritmo usato nei sistemi automatici a retroazione negativa. È un sistema che consente di regolare la reazione all'errore ponderando il contributo di ciascuna componente PID, rendendolo molto flessibile e modulare.

$$u(t) = K_P e(t) + K_I \int (e(\tau) d\tau) + K_D \frac{de(t)}{dt}$$

La componente proporzionale contribuisce in base a un fattore K_P contribuisce alla stabilizzazione del segnale pur mantenendo un errore non nullo; l'azione integrale, proporzionale di fattore k_I , consente di portare il processo al punto di riferimento richiesto; mentre l'azione derivativa, con fattore K_D consente di compensare rapidamente le variazioni dell'errore nel segnale. Data la natura dei segnali esposta nel capitolo 5, l'uso del PID è un passaggio fondamentale per stabilizzare gli input in ingresso all'applicazione. Con questa implementazione infatti si nota una sensibile riduzione del rumore e una migliorata reattività al variare della velocità.

Curve di trasferimento

Si è proceduto ad implementare delle curve di trasferimento che portassero il segnale in un intervallo compreso tra $x \in (-1,1)$ e $y \in (0,1)$.

Dette curve sono disegnate in modo da consentire un adeguato controllo sia

delle variazioni fini di velocità a basse velocità, sia di variazioni rapide ad elevare velocità, per consentire una buona precisione di movimento durante le manovre più delicate un passaggio da corsa a stasi sufficientemente rapido da simulare quanto avviene nella realtà fisica.

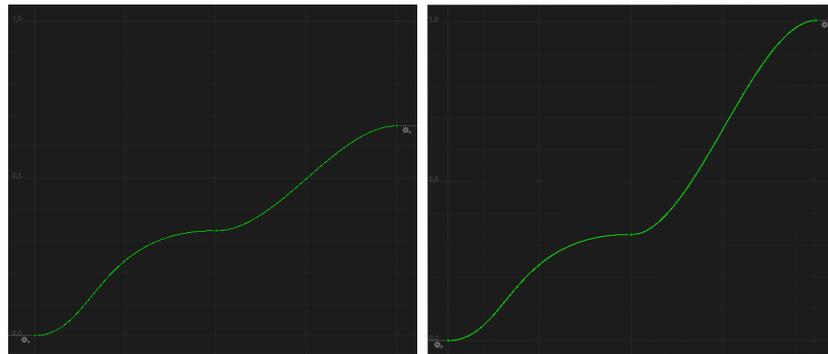


Figure 6.2: Confronto T_y curva camminata (sinistra) e T_y curva corsa (destra)

Come si può osservare la curva di trasferimento T_y è asimmetrica e presenta un plateau per $T_y = 1/3$.

Il valore dell'intervallo stazionario è dovuto all'implementazione del sistema di locomozione del sistema di valutazione, che vedremo nel prossimo paragrafo.

L'asimmetria nella curva T_y , invece, è dovuta al diverso uso e alla diversa percezione nella camminata frontale e posteriore dell'essere umano: mentre la camminata frontale, che compone la quasi totalità dei movimenti compiuti da un essere umano medio nell'arco della giornata, è ricca di dettagli e variazioni fini di cui l'essere umano è capace e che devono essere consentite anche in ambiente virtuale; la camminata posteriore, invece, è soggetta ad un controllo più grossolano, dovuto sia alla sua minor frequenza nelle nostre vite, sia alla struttura del nostro corpo, che non consente un controllo fine come quello del controllo frontale.

La curva T_x , invece, è una curva simmetrica che consente le stesse correzioni laterali su ogni tipo di curva, purché l'utente si mantenga inclinato nel cono

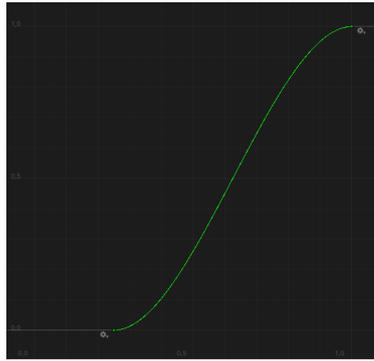


Figure 6.3: Curva T_x in valore assoluto

di 90° frontali o posteriori.

T_x potrebbe essere personalizzata per compensare o simulare eventuali asimmetrie individuali ma la dimostrazione di tale applicazione esula dai fini di questa tesi.

Implementazione in LET

Per usare il dispositivo nel sistema di valutazione di locomozione, di cui parleremo nel prossimo capitolo, si è proceduto ad adattare il sistema di locomozione joystick già presente.

L'interfaccia joystick prevede l'uso di un vettore monodimensionale per modulare la velocità del movimento e la rotazione del controller nello spazio per indicare la direzione: all'inclinazione della levetta del joystick l'utente si sarebbe mosso nella direzione indicata dal controller. Con questa tecnica si abilita l'utente a movimenti facilmente controllabili dalla combinazione del movimento di mano e polso ma si manca sensibilmente di realismo e immersività, oltre ad avere una sensibile propensione a indurre motion sickness.

Si è dunque intervenuti a livello di script, modificando il prefab e aggiungendo alcune componenti. Rispetto al joystick locomotion il nostro sistema di locomozione prevede l'assenza di un empty nella gerarchia che rappresenti l'orientazione della seduta nello spazio, orientazione che lo script di

locomotion legge e interpreta per consentire di curvare all'utente.

Per consentire all'utente di avere un controllo fine sulla velocità di movimento si rimappa il valore in uscita dalle funzioni T_x e T_y usando una funzione di interpolazione lineare (*Mathf.Lerp*) che rimappasse i valori da $(-1, 1) \rightarrow (-walkingSpeed, runningSpeed)$ in modo tale da offrire all'utente la possibilità di camminare e correre a proprio piacimento, usando l'inclinazione della seduta e il *grip* come illustrato in precedenza.

```
1
2 private Vector3 CalculateMotion()
3 {
4     Vector3 movement = Vector3.zero;
5
6     if (_TCPSocket.isMoving)
7     {
8         _lastDir = _walkingSeat.transform.rotation;
9         tp = _TCPSocket.movement;
10
11         t = (tp.y) ;;
12         tx = (tp.x + 1) / 2;
13
14         _lastSpeed = Mathf.Lerp(-_walkSpeed, RunSpeed, t);
15
16         _lastSpeedX = Mathf.Lerp(-_walkSpeed, _walkSpeed,
17 tx);
18
19         if (_lastSpeed > 0)
20         {
21             _TCPSocket.FlattenRearAsym(_slopeControl);
22             _TCPSocket.ModCurveFRAsym(_speedControl);
23         }
24         else if (_lastSpeed < 0)
```

```
25     {
26         _TCPSocket.FlattenFrontAsym(_slopeControl);
27     }
28     else
29     {
30         _TCPSocket.resetCurveFR();
31         _TCPSocket.ModCurveFRAsym(_speedControl);
32     }
33     }
34     movement = _lastDir.getForwardXZ(Time.deltaTime *
35     _factor * _lastSpeed) + _lastDir.getRightXZ(Time.deltaTime *
36     _factor * _lastSpeedX );
37     _factor += _smoothStep;
38     if (_factor > 1)
39         _factor = 1;
40     _bigArrow.gameObject.SetActive(true);
41 }
42 else if (_factor > 0)
43 {
44     movement = _lastDir.getForwardXZ(Time.deltaTime *
45     _factor * _lastSpeed);
46     _factor -= _smoothStep;
47     if (_factor < 0)
48         _factor = 0;
49     _bigArrow.gameObject.SetActive(false);
50 }
51 else if (_lastSpeed > 0)
52 {
53     _lastSpeed = 0;
54     _lastDir = Quaternion.identity;
55     _bigArrow.gameObject.SetActive(false);
56 }
57 return movement;
58 }
```

```
58     private void OnSlopeControl(InputAction.CallbackContext obj)
59     {
60         _speedControl = obj.action.ReadValue<float>();
61     }
```

Test

Data l'unicità di questo sistema di locomozione si è deciso di testare la bontà di questo dispositivo sottoponendolo a un test articolato e omnicomprensivo che riguardasse diversi casi applicativi frequenti in VR. Per testare la qualità del sistema di locomozione a mappatura di pressione, si è proceduto ad utilizzare il sistema di valutazione Locomotion Evaluation Testbed **??**. L'uso di questo test è stato suggerito dalla necessità di confrontare diverse tecniche di locomozione in modo strutturato.

Usando il sistema LET, infatti, si possono confrontare i punteggi e i risultati ottenuti da 4 diversi sistemi di movimento e confrontare in questo modo la qualità, complessiva e per specifici compiti, di ciascuna tecnica di locomozione.

Nonostante le altre tecniche con cui la seduta a mappatura di pressione è stata paragonata siano tutte tecniche che prevedono che l'utente stesse in piedi per usufruire della tecnica non si sono riscontrate significative differenze nella conduzione del test e nella difficoltà delle prove a cui gli utenti sono stati sottoposti.

I test sono stati condotti posizionando la seduta illustrata in precedenza su una sedia girevole, ad altezza variabile, dotata di freno al fine di limitare i movimenti accidentali della stessa durante il test.

Locomotion Evaluation Test

Lo studio nasce dall'esigenza di confrontare diverse tecnologie e diverse soluzioni proposte al problema della locomozione in VR.

Come abbiamo visto nel primo capitolo, le tecniche di locomozione sono molteplici e fanno affidamento a diverse metafore e interfacce per consentire all'utente di muoversi nello spazio. Alcune di esse possono essere performate usando i joystick o altre semplici soluzioni che non necessitano di hardware aggiuntivo oltre le componenti base di un set per realtà virtuale; altre invece possono diventare molto costose e ingombranti, come i treadmill omnidirezionali, dispositivi che consentono all'utente di correre e camminare vincolando l'utente in una posizione fissa nello spazio.

Il Locomotion Evaluation Testbed costruisce un sistema di testing a partire da diversi altri test condotti in precedenza, unendo e ampliando diversi sistemi di valutazione al fine di creare un test comune di riferimento per ogni sistema di locomozione.

In ?? viene evidenziato come il test sia stato disegnato basandosi su diversi altri studi precedenti e analizzando le più comuni azioni compiute in VR insieme alla locomozione.

Come si può vedere nella tabella 7.1, le funzioni e le azioni da valutare per in un test di locomozione sono molteplici e variegate ed è stato necessario trovare un sistema di test che consentisse una valutazione omnicomprensiva di questi compiti.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)
[8] Bowman 1997	✓	✓			✓	✓			✓					✓		
[7] Nilsson 2013	✓	✓	✓						✓		✓		✓			
[40] Wilson 2016		✓			✓		✓		✓							
[41] Calandra 2018		✓	✓	✓	✓	✓	✓	✓	✓	✓						
[42] Calandra 2019		✓	✓	✓	✓	✓	✓	✓	✓	✓						
[43] Nilsson 2013		✓	✓						✓		✓		✓			
[9] Whitton 2005	✓	✓	✓					✓						✓		
[44] Sarapuri 2018										✓	✓					
[45] Albert 2018	✓		✓						✓	✓	✓					
[38] Pai 2017		✓	✓	✓					✓	✓	✓	✓				
[47] Ferracani 2016	✓	✓	✓	✓	✓	✓	✓						✓			
[48] Wilson 2018	✓	✓	✓	✓	✓	✓				✓	✓	✓				
[49] Mayor 2019	✓	✓	✓			✓		✓	✓	✓						
[50] Lapointe 2009	✓	✓	✓		✓	✓					✓		✓			✓
[51] Loup 2018	✓	✓	✓		✓		✓	✓		✓		✓				
[10] Schuemie 2005	✓	✓	✓		✓		✓		✓	✓						
[52] Suma 2010	✓	✓	✓						✓	✓						
This work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7.1: Confronto dei principali studi relativi organizzati in classi.

La Tabella 7.1 sopra riportata illustra i principali studi relativi organizzati nelle seguenti classi: Lavoro presentante un insieme di scenari e compiti includendo metriche di valutazione (1), raccoglie obiettivi(2), o misure soggettive(3), considera altre interazioni oltre la locomozione(4), pone l'accento su aspetti inerenti la precisione, la sensibilità e responsività o il livello di controllo dell'interfaccia (5), la velocità di operazione (6), la facilità di compiere errori(7), l'usabilità (8), il senso di presenza e immersione(9), la motion sickness(10), lo sforzo fisico, comfortevolezza, senso di movimento, acclimatazione (11), onere mentale e carico cognitivo(12), facilità d'uso(13), appropriatezza e efficacia(14), apprendibilità (15), apprezzamento, soddisfazione/frustrazione(16). Come si può vedere dalla tabella diversi studi coprono diversi aspetti della locomozione in vr e solo il Locomotion Evaluation Testbed copre tutti gli aspetti evidenziati dagli altri studi, aggiungendo, in più, un'analisi sull'apprendibilità del sistema di locomozione, che nel caso specifico della seduta a mappatura di pressione è particolarmente importante. Il test è strutturato in diversi scenari, ciascuno dei quali raccoglie diversi task. Ogni scenario si occupa di valutare un determinato aspetto dell'uso

della tecnica di locomozione in esame.

hubOgni sessione di test è preceduta da una sessione di training, in cui l'utente può imparare a usare propriamente la tecnica di locomozione in esame, successivamente si sottopone ciascun utente agli scenari progettati per sottoporre ad un'adeguato stress le capacità della tecnica di locomozione.

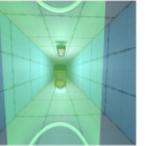
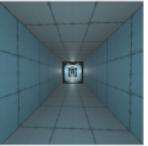
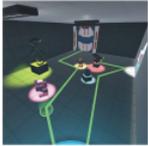
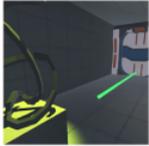
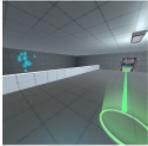
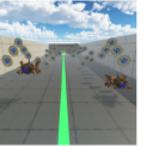
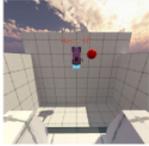
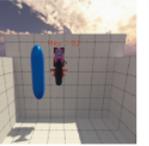
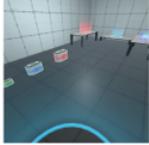
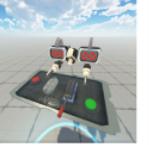
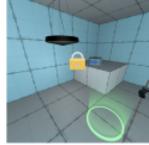
Scenario	Task				
<p>■ S1. Straight movements</p>	 <p>■ T1. Straight line walking (a)</p>	 <p>■ T2. Over/Under-shooting (b)</p>	 <p>■ T3. Chasing (c)</p>	 <p>■ T4. Sprinting (d)</p>	
<p>■ S2. Direction control</p>	 <p>■ T1. Multi-straight line walking (e)</p>	 <p>■ T2. Backward walking (f)</p>	 <p>■ T3. Curved walking (g)</p>	 <p>■ T4. Stairs & ramps (h)</p>	 <p>■ T5. Fear (i)</p>
<p>■ S3. Decoupled movements</p>	 <p>■ T1. Decoupled gaze (j)</p>	 <p>■ T2. Stretched-out hands (k)</p>	 <p>■ T3. Decoupled hands (l)</p>		
<p>■ S4. Agility</p>	 <p>■ T1. Dynamic agility (m)</p>	 <p>■ T2. Stationary agility (n)</p>	 <p>■ T3. Evasion (o)</p>		
<p>■ S5. Interaction with objects</p>	 <p>■ T1. Grabbing (p)</p>	 <p>■ T2. Manipulation (q)</p>	 <p>■ T3. Interaction in motion (r)</p>		 <p>Training (s)</p>

Figure 7.1: Scenari: S1 Movimento rettilineo, S2 Direction Control, S3 Movimento disaccoppiato, S4 agilità, S5 Interazione con oggetti.

Prima di iniziare la sperimentazione vengono richieste all'utente informazioni circa la sua esperienza con la VR e la tecnica di locomozione che andrà ad utilizzare, così come viene richiesto di compilare un questionario per la valutazione della motion sickness.

La sperimentazione inizia con uno scenario di training, in cui l'utente, guidato passo passo dal conduttore, impara ad usare la tecnica di locomozione, l'interfaccia e compie esperienze significative al fine di condurre al pieno delle proprie capacità, e di quelle del sistema di locomozione, i test successivi.

Lo scenario 1 (s1) è progettato per valutare la tecnica di locomozione nella più semplice condizione possibile. A l'utente viene chiesto di attraversare un corridoio compiendo quattro task: il primo task (t1) prevede di camminare seguendo una linea retta il più fedelmente possibile; il secondo task (t2) mira a verificare la precisione della tecnica nel fermarsi nel punto verificato, richiedendo all'utente di fermarsi al centro di destinazioni sempre più piccole; successivamente (t3) si richiede all'utente di seguire un oggetto in movimento, con velocità variabile, ad una distanza costante, infine (t4) si richiede all'utente di correre lungo un corridoio valutandone le performance. Questo scenario, concentrato sul movimento rettilineo, è un buon trait d'union tra l'ambiente di training e le parti più complesse del test.

Lo scenario 2 (s2) è ideato per porre l'utente in una situazione di locomozione più articolata e valutarne le capacità di controllo della direzione. Il test inizia in una galleria d'arte in cui, come primo task (t1), l'utente viene fatto muovere, seguendo una linea guida, attraverso diverse opere che, una volta raggiunte, deve osservare per poi procedere oltre; successivamente l'utente deve camminare all'indietro per varcare una porta aperta solo mentre non la si sta guardando (t2); completata questa fase nel primo ambiente si procede a far muovere l'utente in un tunnel curvilineo (t3) per verificarne le capacità di velocità e precisione di movimento; il task successivo (t4) prevede che l'utente salga diverse rampe di scale, lisce e con scalini, per misurare quale risultato più gradevole per la tecnica in esame; infine nell'ultimo task (t5) si

richiede all'utente di attraversare un ponte sospeso in condizioni di paura crescente.

Il terzo scenario (s3) si occupa di valutare le capacità di separare e distinguere le direzioni di camminata da quelle di vista e movimento. Nel primo task (t1) viene richiesto all'utente di camminare in direzione rettilinea, come in s1t1, ma guardando un punto in direzione quasi ortogonale a quella di vista; nella sezione successiva (t2) l'utente viene posto in controllo di due robot controllati dai controller, con cui deve collezionare degli oggetti camminando lungo una linea retta; infine (t3) viene richiesto all'utente di compiere lo stesso task appena compiuto ma usando robot più piccole e distribuendo gli oggetti da collezionare in posizioni più complesse.

Il penultimo scenario (s4) mette al centro dell'attenzione l'agilità offerta dal sistema di locomozione. Nel primo task (t1) l'utente deve attraversare un corridoio mentre dei muri mobili gli si lanciano contro per poi raggiungere una posizione (t2) in cui a l'utente viene richiesto di schivare proiettili lanciati verso la sua posizione senza usare la tecnica di locomozione, successivamente (t3) usando la tecnica di locomozione per schivare una seconda serie di proiettili.

L'ultimo scenario (s5) prevede un'ampia gamma di interazione con diversi oggetti. Nel primo task (t1) all'utente viene richiesto di collezionare oggetti per aprire delle porte, prima deve prendere un oggetto, poi due oggetti e infine spostare due oggetti senza farli collidere con le pareti; nel secondo task (t2) viene richiesto all'utente di manipolare degli oggetti per posizionarli in posizioni specifiche e in ordini precisi al fine di valutare la capacità del sistema di locomozione di consentire questo tipo di azioni; infine, l'ultimo task di tutto il test, (t5) prevede che l'utente inseguia un robot che cerca di scappare mentre si operano degli strumenti come bottoni e leve tenute dal robot.

Al termine di ciascuno scenario all'utente viene richiesto di compilare un questionario sullo scenario appena concluso.

Dopo l'ultimo questionario l'utente può compilare la seconda parte del questionario per la motion sickness e un ultimo questionario su tutta l'esperienza.

Modifiche

Utilizzando il locomotion evaluation testbed nelle nuove versioni di unity alcuni aggiornamenti e modifiche si sono rese necessarie al fine di aggiornare il sistema e utilizzare l'applicativo su un più ampio numero di dispositivi HMD.

Per passare dalla versione 2018 alla versione 2020 di unity si è dovuto procedere ad aggiornare il sistema di gestione degli input dall' sdk di SteamVR al toolkit OpenXR. Durante questo aggiornamento non sono state applicate modifiche al test a meno dell'aggiunta di un controllo nel test s1 t3: al termine del movimento del robot lungo il proprio percorso, poteva capitare che l'utente si trovasse in un posizione tale da impedire il corretto svolgimento del task successivo s1t4. Si è dunque proceduto ad implementare un controllo per assicurarsi che il task successivo non venisse influenzato dai risultati del task precedente.

Risultati

Somministrazione del test

Il test è stato condotto con il contributo di 12 tester, (9 maschi e 3 femmine), di età media di 25 anni \pm 1 anni/o, sani. Ciascun tester riportava un'esperienza diversa con la realtà virtuale e software di editing e modellazione 3d.

Il test prevede che ciascun utente si eserciti nell'uso della tecnica di locomozione per poi essere sottoposto a cinque prove e rispondere ad altrettanti questionari circa la qualità dell'interfaccia. I test sono preceduti e seguiti da un questionario circa le condizioni psicofisiche dell'utente al momento dell'esecuzione del test e di apprezzamento e valutazione generale della tecnica.

Ciascuna sessione di test inizia con la compilazione da parte del candidato della sezione del test inerente le proprie informazioni anagrafiche e le informazioni inerenti la frequenza d'uso di tecnologie 3d e VR.

Successivamente si richiede all'utente di esprimere il grado di disagio psicofisico provato all'inizio del test, al fine di confrontarlo con quello della fine del test.

Si procede dunque a informare l'utente della struttura del test e a spiegare l'interfaccia e i suoi comandi. Successivamente si inizia la pratica nello scenario di training: in questo scenario l'utente andrà a sperimentare e provare tutte le azioni da svolgere con la tecnica di locomozione che gli saranno

richieste durante il test, dal camminare linearmente all'afferrare oggetti.

Una volta che l'utente si sentirà a proprio agio e dimostrerà padroneggiare la tecnica di locomozione, si procede con i test veri e propri.

Si inizia con la sezione inerente il movimento rettilineo (S1) e si procede in ordine passando da controllo di direzione (S2), movimento disaccoppiato(S4), agilità (S4) e infine manipolazione oggetti (S5).

All'inizio di ogni scenario di test si misura il battito cardiaco dell'utente e prima di ogni task si forniscono spiegazioni circa i compiti da svolgere. Al termine di ogni sessione si provvede a misurare nuovamente il battito cardiaco e si fornisce all'utente il questionario da compilare circa la recente esperienza.

Al termine di tutti i test e dopo la compilazione del questionario conclusivo si prende nota anche di eventuali commenti, consigli o osservazioni.

Risultati

Dai test sono emerse alcune interessanti conclusioni. Il movimento tramite inclinazione del torso è molto rapido da imparare e molto intuitivo. L'uso di una seduta girevole aiuta nella scelta della direzione e rende l'interfaccia semplice da usare.

Molti problemi emergono, però, quando viene richiesto un controllo fine e rapido della direzione della camminata: il sistema non può performare bene in termini di tempo o errori come altri sistemi più facili da controllare, come il joystick. In particolare nelle prove di agilità la WS risulta poco soddisfacente, probabilmente dovuto alla latenza di input e la zona morta nell'input troppo ampia.

Nella manipolazione di oggetti la WS non compete con altre tecniche: qualunque oggetto con cui interagire troppo in basso per non essere raccolto restando in equilibrio sulla sedia porta ad un movimento involontario. In più molti utenti non sono riusciti a concludere l'ultimo scenario a causa di

motion sickness elevata.

In generale l'interfaccia piace e viene riconosciuta come valida e comoda, ma questa implementazione è ancora lontana dall'essere effettivamente utilizzabile da tutti.

Conclusioni

Il dispositivo realizzato per questo studio, nella sua semplicità, apre la strada a molte riflessioni e possibilità.

In un periodo storico in cui la realtà virtuale evolve, per passare dall'essere un prodotto di nicchia, figlio di esperimenti teorici e dalle scarse applicazioni pratiche e commerciali, a essere un prodotto dalle ricche potenzialità e dall'ampia accessibilità, è bene chiedersi quanto questa "*altra*" realtà in cui stiamo entrando possa e debba replicare la realtà fisica in cui già viviamo. La dimensione della simulazione può beneficiare estremamente da una tecnologia che replichi il più fedelmente possibile la realtà fisica in un ambiente protetto, ma è uno sforzo tecnico e tecnologico necessario per l'uso quotidiano del futuro utente medio?

In questa tesi, riflettendo su quanto la realtà virtuale debba imitare quella fisica, abbiamo proposto una soluzione che offrisse all'utente il giusto compromesso tra comodità e realismo, creando un'interfaccia d'interazione pensata tenendo a mente le esigenze dell'utente di VR.

Walking Seat

Questa interfaccia di locomozione nasce per sfruttare uno dei comportamenti più frequenti nei neo-videogiocatori e nei bambini: tanto più il gioco si fa concitato, tanto più questi utenti si muoveranno sulla sedia, tanto più

"piegheranno in curva" in una gara automobilistica e salteranno sul divano per scansare un ostacolo.

Per sfruttare questo comportamento abbiamo realizzato una seduta capace di mappare il baricentro dell'utente seduto su di essa per usare l'inclinazione del torso come input per controllare l'avatar virtuale.

Abbiamo realizzato un dispositivo usando la suite Arduino, montando quattro sensori di pressione e un giroscopio su una board con connessione WiFi. Abbiamo creato un socket wireless per trasmettere nel modo più semplice possibile i dati raccolti dalla seduta ad un'applicazione per la realtà virtuale. Con numerose iterazioni, test e sperimentazioni, abbiamo progettato un'interfaccia che consentisse di valutare la validità dell'ipotesi sopraccitata. Sfruttando e aggiornando un sistema di valutazione omnicomprendente, che valutasse il sistema di locomozione in diverse situazioni e applicazioni pratiche, confrontandolo al tempo stesso con altre tecniche.

Al termine di questa sperimentazione possiamo affermare che il sistema di locomozione si è dimostrato piacevole ed relativamente efficace oltre che molto rapido da apprendere. Questa prima implementazione, così semplice ed essenziale, non è stata in grado di offrire una soluzione valida per tutti gli scenari in cui è stata testata ma ha illustrato chiaramente le proprie criticità e potenzialità e posto delle solide basi per eventuali versioni successive.

Sviluppi futuri

Oltre ad un miglioramento della sensoristica di pressione con sensori a membrana, non solo per una migliore velocità e densità di lettura ma anche per un maggior livello di comfort, la seduta necessita di un miglioramento software che consenta di riconoscere l'utente, la sua posizione e potenzialmente le sue intenzioni, indipendentemente dalla sua posizione rispetto al centro della seduta; probabilmente il training di una rete neurale o una AI potrebbe portare enormi benefici ad un dispositivo di questo tipo.

Una seduta con una precisione come quella appena descritta avrebbe, inoltre, altre applicazioni oltre la realtà virtuale. In un mondo post Covid, con lo smart working pratica sempre più comune nel contesto lavorativo, un dispositivo capace di verificare la correttezza della posizione che si assume davanti al computer può diventare uno strumento fondamentale per la prevenzione e il trattamento di condizioni cliniche.

Un dispositivo trasparente come una seduta, di cui quotidianamente facciamo uso in diversi ambienti, offre la possibilità di passare rapidamente da realtà fisica ad altre realtà miste; senza bisogno di training specifico ne di attrezzatura dedicata ingombrante, questa tecnologia può aprire la strada ad un mondo in cui la realtà virtuale è alla portata di tutti.

Bibliography

- [1] John C. Tuthill and Eiman Azim. «Proprioception». In: *Current Biology* 28.5 (2018), R194–R203. ISSN: 0960-9822. DOI: <https://doi.org/10.1016/j.cub.2018.01.064>. URL: <https://www.sciencedirect.com/science/article/pii/S0960982218300976>.
- [2] Nadia Bianchi-Berthouze, Whan Woong Kim, and Darshak Patel. «Does Body Movement Engage You More in Digital Game Play? and Why?» In: *Affective Computing and Intelligent Interaction*. Ed. by Ana C. R. Paiva, Rui Prada, and Rosalind W. Picard. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 102–113.
- [3] Vaibhav V. Unhelkar, Claudia Pérez-D’Arpino, Leia Stirling, and Julie A. Shah. «Human-robot co-navigation using anticipatory indicators of human walking motion». In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 6183–6190. DOI: [10.1109/ICRA.2015.7140067](https://doi.org/10.1109/ICRA.2015.7140067).
- [4] Shohei Yamashita, Tomohiro Kurihara, Tetsushi Ikeda, Kazuhiko Shinozawa, and Satoshi Iwaki. «Evaluation of robots that signals a pedestrian using face orientation based on analysis of velocity vector fluctuation in moving trajectories». In: *Advanced Robotics* 34.20 (2020), pp. 1309–1323. DOI: [10.1080/01691864.2020.1811763](https://doi.org/10.1080/01691864.2020.1811763). eprint: <https://doi.org/10.1080/01691864.2020.1811763>.

- [//doi.org/10.1080/01691864.2020.1811763](https://doi.org/10.1080/01691864.2020.1811763). URL: <https://doi.org/10.1080/01691864.2020.1811763>.
- [5] Frank Steinicke, Yon Visell, Jennifer L. Campos, and Anatole Lcuyer. «Human Walking in Virtual Environments: Perception, Technology, and Applications». In: 2013.
- [6] Frank Steinicke, Yon Visell, Jennifer Campos, and Anatole Lécuyer. *Human Walking in Virtual Environments: Perception, Technology, Applications*. Mar. 2013. DOI: 10.1007/978-1-4419-8432-6.
- [7] Shawn M. O'Connor and Arthur D. Kuo. «Direction-Dependent Control of Balance During Walking and Standing». In: *Journal of Neurophysiology* 102.3 (2009). PMID: 19553493, pp. 1411–1419. DOI: 10.1152/jn.00131.2009. eprint: <https://doi.org/10.1152/jn.00131.2009>. URL: <https://doi.org/10.1152/jn.00131.2009>.
- [8] Adrian J.T. Alsmith and Matthew R. Longo. «Where exactly am I? Self-location judgements distribute between head and torso». In: *Consciousness and Cognition* 24 (2014), pp. 70–74. ISSN: 1053-8100. DOI: <https://doi.org/10.1016/j.concog.2013.12.005>. URL: <https://www.sciencedirect.com/science/article/pii/S1053810013001645>.
- [9] Mehrangiz Ashiri, Brian Lithgow, Behzad Mansouri, and Zahra Mousavi. «Comparison between Vestibular Responses to a Physical and Virtual Reality Rotating Chair». In: *Proceedings of the 11th Augmented Human International Conference*. AH '20. Winnipeg, Manitoba, Canada: Association for Computing Machinery, 2020. ISBN: 9781450377287. DOI: 10.1145/3396339.3396392. URL: <https://doi.org/10.1145/3396339.3396392>.
- [10] Costas Boletsis. «The New Era of Virtual Reality Locomotion: A Systematic Literature Review of Techniques and a Proposed Typology».

- In: *Multimodal Technologies and Interaction* 1.4 (2017). ISSN: 2414-4088. DOI: 10.3390/mti1040024. URL: <https://www.mdpi.com/2414-4088/1/4/24>.
- [11] Nadia Bianchi-Berthouze. «Understanding the Role of Body Movement in Player Engagement». In: *Human–Computer Interaction* 28.1 (2013), pp. 40–75. DOI: 10.1080/07370024.2012.688468. eprint: <https://www.tandfonline.com/doi/pdf/10.1080/07370024.2012.688468>. URL: <https://www.tandfonline.com/doi/abs/10.1080/07370024.2012.688468>.
- [12] Mark R. Mine, Frederick P. Brooks, and Carlo H. Sequin. «Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction». In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 19–26. ISBN: 0897918967. DOI: 10.1145/258734.258747. URL: <https://doi.org/10.1145/258734.258747>.
- [13] Alasdair G. Thin, Lisa Hansen, and Danny McEachen. «Flow Experience and Mood States While Playing Body Movement-Controlled Video Games». In: *Games and Culture* 6.5 (2011), pp. 414–428. DOI: 10.1177/1555412011402677. eprint: <https://doi.org/10.1177/1555412011402677>. URL: <https://doi.org/10.1177/1555412011402677>.
- [14] Ryan P. McMahan et al. «Evaluating natural interaction techniques in video games». In: *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. 2010, pp. 11–14. DOI: 10.1109/3DUI.2010.5444727.
- [15] P.K. Nag, S. Pal, S.M. Kotadiya, A. Nag, and K. Gosai. «Human–seat interface analysis of upper and lower body weight distribution». In: *International Journal of Industrial Ergonomics* 38.5 (2008). Seating Dynamics, pp. 539–545. ISSN: 0169-8141. DOI: <https://doi.org/10.1016/j.ergon.2008.05.011>.

- 1016/j.ergon.2007.10.024. URL: <https://www.sciencedirect.com/science/article/pii/S016981410700203X>.
- [16] Guillaume Loup and Emilie Loup-Escande. «Effects of Travel Modes on Performances and User Comfort: A Comparison between ArmSwinger and Teleporting». In: *International Journal of Human–Computer Interaction* 35.14 (2019), pp. 1270–1278. DOI: 10.1080/10447318.2018.1519164. eprint: <https://doi.org/10.1080/10447318.2018.1519164>. URL: <https://doi.org/10.1080/10447318.2018.1519164>.
- [17] Graham Wilson, Mark McGill, Matthew Jamieson, Julie R. Williamson, and Stephen A. Brewster. «Object Manipulation in Virtual Reality Under Increasing Levels of Translational Gain». In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–13. ISBN: 9781450356206. DOI: 10.1145/3173574.3173673. URL: <https://doi.org/10.1145/3173574.3173673>.
- [18] Andrea Ferracani, Daniele Pezzatini, Jacopo Bianchini, Gianmarco Biscini, and Alberto Del Bimbo. «Locomotion by Natural Gestures for Immersive Virtual Environments». In: *Proceedings of the 1st International Workshop on Multimedia Alternate Realities*. AltMM '16. Amsterdam, The Netherlands: Association for Computing Machinery, 2016, pp. 21–24. ISBN: 9781450345217. DOI: 10.1145/2983298.2983307. URL: <https://doi.org/10.1145/2983298.2983307>.
- [19] Volkan Sevinc and Mehmet Ilker Berkman. «Psychometric evaluation of Simulator Sickness Questionnaire and its variants as a measure of cybersickness in consumer virtual environments». In: *Applied Ergonomics* 82 (2020), p. 102958. ISSN: 0003-6870. DOI: <https://doi.org/10.1016/j.apergo.2019.102958>. URL: <https://www.sciencedirect.com/science/article/pii/S0003687019301759>.

- [20] Julia Diemer, Georg W. Alpers, Henrik M. Peperkorn, Youssef Shiban, and Andreas Mühlberger. «The impact of perception and presence on emotional reactions: a review of research in virtual reality». In: *Frontiers in Psychology* 6 (2015). ISSN: 1664-1078. DOI: 10.3389/fpsyg.2015.00026. URL: <https://www.frontiersin.org/articles/10.3389/fpsyg.2015.00026>.
- [21] Adam Jones, J. Edward Swan, Gurjot Singh, and Eric Kolstad. «The Effects of Virtual Reality, Augmented Reality, and Motion Parallax on Egocentric Depth Perception». In: *2008 IEEE Virtual Reality Conference*. 2008, pp. 267–268. DOI: 10.1109/VR.2008.4480794.
- [22] Craig D. Murray and Judith Sixsmith. «The Corporeal Body in Virtual Reality». In: *Ethos* 27.3 (1999), pp. 315–343. ISSN: 00912131, 15481352. URL: <http://www.jstor.org/stable/640592> (visited on 07/12/2022).
- [23] Giuseppe Riva. «From virtual to real body: Virtual reality as embodied technology». In: *Journal of Cyber Therapy and Rehabilitation* 1 (Jan. 2008), pp. 7–22.
- [24] Jennifer Wu. «Choosing My Avatar & the Psychology of Virtual Worlds: What Matters?» In: 2014.
- [25] Daniel Roth, Jean-Luc Lugrin, Dmitri Galakhov, Arvid Hofmann, Gary Bente, Marc Erich Latoschik, and Arnulph Fuhrmann. «Avatar realism and social interaction quality in virtual reality». In: *2016 IEEE Virtual Reality (VR)*. 2016, pp. 277–278. DOI: 10.1109/VR.2016.7504761.
- [26] Anthony Steed, Ye Pan, Fiona Zisch, and William Steptoe. «The impact of a self-avatar on cognitive load in immersive virtual reality». In: *2016 IEEE Virtual Reality (VR)*. 2016, pp. 67–76. DOI: 10.1109/VR.2016.7504689.

- [27] Arvid Kappas. «To Our Emotions, with Love: How Affective Should Affective Computing Be?» In: *Affective Computing and Intelligent Interaction*. Ed. by Sidney D’Mello, Arthur Graesser, Björn Schuller, and Jean-Claude Martin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–1. ISBN: 978-3-642-24600-5.
- [28] Yaying Zhang, Bernhard E. Riecke, Thecla Schiphorst, and Carman Neustaedter. «Perch to Fly: Embodied Virtual Reality Flying Locomotion with a Flexible Perching Stance». In: *Proceedings of the 2019 on Designing Interactive Systems Conference*. DIS ’19. San Diego, CA, USA: Association for Computing Machinery, 2019, pp. 253–264. ISBN: 9781450358507. DOI: 10.1145/3322276.3322357. URL: <https://doi.org/10.1145/3322276.3322357>.
- [29] Thinh Nguyen-Vo, Bernhard E. Riecke, Wolfgang Stuerzlinger, Duc-Minh Pham, and Ernst Kruijff. «NaviBoard and NaviChair: Limited Translation Combined with Full Rotation for Efficient Virtual Locomotion». In: *IEEE Transactions on Visualization and Computer Graphics* 27.1 (2021), pp. 165–177. DOI: 10.1109/TVCG.2019.2935730.
- [30] Maud Marchal, Julien Pettré, and Anatole Lécuyer. «Joyman: A human-scale joystick for navigating in virtual worlds». In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. 2011, pp. 19–26. DOI: 10.1109/3DUI.2011.5759212.
- [31] Alexandra Kitson, Abraham M. Hashemian, Ekaterina R. Stepanova, Ernst Kruijff, and Bernhard E. Riecke. «Comparing leaning-based motion cueing interfaces for virtual reality locomotion». In: *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. 2017, pp. 73–82. DOI: 10.1109/3DUI.2017.7893320.

- [32] Alberto Cannavo, Davide Calandra, F. Gabriele Prattico, Valentina Gatteschi, and Fabrizio Lamberti. «An Evaluation Testbed for Locomotion in Virtual Reality». In: *IEEE Transactions on Visualization and Computer Graphics* 27.3 (Mar. 2021), pp. 1871–1889. DOI: 10.1109/tvcg.2020.3032440. URL: <https://doi.org/10.1109/2Ftvcg.2020.3032440>.
- [33] Hiroyuki Kidokoro, Takayuki Kanda, Drazen Brscic, and Masahiro Shiomi. «Will I bother here? - A robot anticipating its influence on pedestrian walking comfort». In: Mar. 2013, pp. 259–266. ISBN: 978-1-4673-3099-2. DOI: 10.1109/HRI.2013.6483597.
- [34] Carlo Flemming, Benjamin Weyers, and Daniel Zielasko. «How to Take a Brake from Embodied Locomotion x2013; Seamless Status Control Methods for Seated Leaning Interfaces». In: *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2022, pp. 728–736. DOI: 10.1109/VR51125.2022.00094.