Primjena virtualne stvarnosti u korištenju softvera pokretima ruku

Žabić, Filip

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:235:469233

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Filip Žabić

Zagreb, 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor: Student:

Prof. dr.sc. Zoran Kunica, dipl.ing. Filip Žabić

Zagreb, 2021.

ZADATAK

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: **602 - 04 / 21 - 6 / 1**	
Ur.broj: **15 - 1703 - 21 -**	

ZAVRŠNI ZADATAK

Student: **Filip Žabić** Mat. br.: 0035202083

Naslov rada na hrvatskom jeziku: **Primjena virtualne stvarnosti u korištenju softvera pokretima ruku**

Naslov rada na engleskom jeziku: **Application of virtual reality in use of software by hand movements**

Opis zadatka:

Napredak tehnologije omogućuje sve dublje razmatranje osnova ljudskog postojanja, što se odnosi i na motoričke radnje, neizostavne u radnim procesima, pa tako i u onim gdje se koriste računala.

U radu je potrebno:
1. objasniti značaj snimanja i analize pokreta za oblikovanje radnih procesa i opisati trenutačni stupanj razvijenosti tehnologije, posebno sa stanovišta pokreta koji se pojavljuju pri korištenju softvera
2. opisati opremu koja će se koristiti za snimanje pokreta te radni proces korištenja odabranog softvera
3. razviti aplikaciju virtualne stvarnosti za izvedbu pokreta rukama pri korištenju softvera te provesti eksperimente.

Zadatak zadan:
30. studenoga 2020.

Datum predaje rada:
1. rok: 18 veljače 2021.
2. rok (izvanredni): 5. srpnja 2021.
3. rok: 23. rujna 2021.

Predviđeni datumi obrane:
1. rok: 22.2. – 26.2.2021.
2. rok (izvanredni): 9.7.2021.
3. rok: 27.9. – 1.10.2021.

Zadatak zadao:
Prof. dr.sc. Zoran Kunica

Predsjednik Povjerenstva:
Prof. dr. sc. Branko Bauer

IZJAVA

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem mentoru prof. dr.sc. Zoranu Kunici na podršci, savjetima i opremi ustupljenoj za izradu ovog rada.

Također zahvaljujem mojoj obitelji, prijateljima i curi, koji su mi bili podrška prilikom studiranja i izradi ovog rada.

U Zagrebu, 2. veljače 2021.

Filip Žabić

SAŽETAK

Ljudsko djelovanje oduvijek teži optimiranju ljudskog tjelesnog rada bilo koje vrste radi postizanja što većeg profita i produktivnosti te humanizacije rada. Svaki takav rad sastoji se od niza pokreta te je zato snimanje i analiza pokreta ključno u oblikovanju i normiranju industrijskih procesa. Pomoću novih tehnologija, poput virtualne stvarnosti, snimanje i analiza pokreta znatno je lakše. Kontroler Leap Motion je uređaj kojim se pokreti ruku mogu snimiti te prikazati u virtualnoj stvarnosti. U sklopu rada prikazane su karakteristike, princip rada i mogućnosti uređaja te su izvedeni pokusi kojima je istražena upotrebljivost uređaja u raznim situacijama poput: korištenja CAD inženjerskog softvera za 3D modeliranje, kontrole kursora operativnog sustava, snimanja pokreta ruku koristeći deformabilno tijelo te općeg snimanja pokreta ruku. Sveukupno, rezultati navedenih pokusa pokazali su da uređaj ima velik potencijal u industrijskoj proizvodnji, ali i ne samo njoj, za snimanje i analizu općih putanja ruku u svrhu poboljšanja nekog procesa.

Ključne riječi: virtualna stvarnost, Leap Motion, šaka, ruka, pokret, praćenje, Python

SUMMARY

Human activity has always strived to optimise human physical labour of any kind in order to maximise profits and productivity and humanisation of work. Each such work consists of series of movements, so capturing and analysis of movements is crucial in design and standardisation of industrial processes. By using new technologies, like virtual reality, motion capturing and analysis is much easier. The Leap Motion controller is a device that can capture hand movements and display them in virtual reality. This work presents the characteristics, principles of operation and capabilities of the device as well as performed experiments to investigate the usability of the device in various situations such as: during use of CAD engineering software for 3D modelling, cursor movement control of the operating system, capturing hand movements having a deformable body and general hand movement capturing. Overall, the results of these experiments have shown that the device has a lot of potential in manufacturing and elsewhere, where it can be used for capturing and analysis of general hand movements in order to improve a certain process.

Key words: virtual reality, Leap Motion, hand, arm, movement, tracking, Python

S A D R Ž A J

ZADATAK I
IZJAVA II
SAŽETAK III
SUMMARY IV
POPIS OZNAKA I MJERNIH JEDINICA FIZIKALNIH VELIČINA VII
POPIS SLIKA VIII
POPIS TABLICA X

1. UVOD 1

2. ZNAČAJ SNIMANJA I ANALIZE POKRETA 3

3. TRENUTAČNI STUPANJ RAZVIJENOSTI TEHNOLOGIJE I METODE PRAĆENJA POKRETA 6

3.1. MEHANIČKO PRAĆENJE 8

3.2. OPTIČKO PRAĆENJE 9

3.3. ELEKTROMAGNETSKO (MAGNETSKO) PRAĆENJE 9

3.4. OSTALE VRSTE TEHNOLOGIJA PRAĆENJA POKRETA 10

4. KONTROLER *LEAP MOTION* 11

4.1. PRINCIP RADA UREĐAJA [31] 14

4.1.1. Hardver 14

4.1.2. Softver 15

4.2. LATENCIJA [32 i 33] 16

4.2.1. Hardverska razina 16

4.2.2. Sustav prikaza računala 17

4.2.3. Analiza čimbenika i optimiranje latencije 18

4.3. PROGRAMSKA PODRŠKA [34 i 35] 21

4.3.1. Koordinatni sustav 22

4.3.2. Podaci o praćenju kretanja 23

4.3.2.1. Šaka ... 24
4.3.2.2. Ruka ... 25
4.3.2.3. Prsti ... 26

5. POKUSI ... 28
5.1. KONTROLA KURSORA OPERATIVNOG SUSTAVA ... 29
5.2. INTERAKCIJA S CAD SOFTVEROM ... 32
5.3. RAD S DEFORMABILNIM TIJELOM ... 35
5.4. PRAĆENJE I ANALIZA OPĆIH PUTANJA RUKU ... 39
5.5. ZAKLJUČAK POKUSA ... 41
6. ZAKLJUČAK ... 42
7. LITERATURA ... 44
PRILOZI ... 47

POPIS OZNAKA I MJERNIH JEDINICA FIZIKALNIH VELIČINA

Oznaka	Mjerna jedinica	Opis
α	° ili rad	kut
a	m/s^2	ubrzanje
F	N	sila
l	mm	duljina
t	s	vrijeme
v	m/s	brzina

API	engl. *Application Programming Interface* – sučelje za programiranje
engl.	engleski
IoT	engl. *Internet of Things* – Internet stvari
IR	engl. *Infrared* – infracrveno
LED	engl. *Light Emitting Diode* – dioda koja emitira svjetlo
OpenCV	engl. *Open computer vision* – otvoreni računalni vid
SDK	engl. *Software Development Kit* – komplet za razvoj softvera
TCP	engl. *Transmission Control Protocol* – protokol upravljanja prijenosom
USB	engl. *Universal Serial Bus* – univerzalna serijska sabirnica
VR	engl. *Virtual Reality* – virtualna stvarnost

POPIS SLIKA

Slika 1. Postotak svjetske populacije zaposlen u poljoprivredi u razdoblju 1991.–2020. [1] ... 1

Slika 2. Frederick Winslow Taylor (1856.–1915.) [10] ... 3

Slika 3. Rast industrije VR u razdoblju 2017.–2021. [12] ... 4

Slika 4. Upotreba virtualne stvarnosti u obrazovanju [18] ... 5

Slika 5. Snimanje ljudskog hoda u virtualnoj stvarnosti [22] ... 7

Slika 6. Praćenje pokreta glave u virtualnoj stvarnosti [24] ... 8

Slika 7. Kontroler Leap Motion [26] ... 11

Slika 8. Kontroler Leap Motion u uporabi [28] ... 12

Slika 9. Postepeni razvoj Leap Motion kontrolera [28] ... 12

Slika 10. Gabaritne dimenzije kontrolera Leap Motion [30] ... 14

Slika 11. Oblik prostora za interakciju ... 14

Slika 12. Prikaz slike kamera ... 15

Slika 13. Različiti načini rada i njihov utjecaj na brzinu ... 17

Slika 14. Latencija u najgorem slučaju ... 19

Slika 15. Latencija uz Vsync isključen ... 19

Slika 16. Latencija uz USB 3.0 ... 20

Slika 17. Latencija u najboljem mogućem slučaju ... 20

Slika 18. Koordinatni sustav kontrolera Leap Motion ... 22

Slika 19. Objekti dostupni kroz API ... 23

Slika 20. Model šake uz prikaz vektora normale na dlan i orijentacije ... 24

Slika 21. 3D model šake u odnosu na pravu ... 25

Slika 22. Model prstiju i vektor smjera za svaki prst ... 26

Slika 23. Model kostiju Leap Motion softvera ... 27

Slika 24. Prikaz ljudske ruke u odnosu na različite nagibe šake (*pitch, roll, yaw*) [37] 28

Slika 25. Prikaz kontrole kursora pomoću dlana..... 30

Slika 26. Prikaz klika kursora pomoću geste 31

Slika 27. Grafovi nagiba *pitch* i *roll*..... 32

Slika 28. 3D model kvadra koji je korišten u pokusu 33

Slika 29. Prikaz kontrole modela pomoću obje ruke istovremeno..... 34

Slika 30. Početna pozicija mjerenja: tijelo u nedeformiranom stanju..... 36

Slika 31. Krajnja pozicija mjerenja: spužva u deformiranom stanju 37

Slika 32. Pozicija vrha kažiprsta u vremenu po svim osima kontrolera Leap Motion..... 38

Slika 33. Pozicija dlana ruke u vremenu po svim osima kontrolera Leap Motion 39

Slika 34. Brzina dlana ruke u vremenu po svim osima kontrolera Leap Motion..... 40

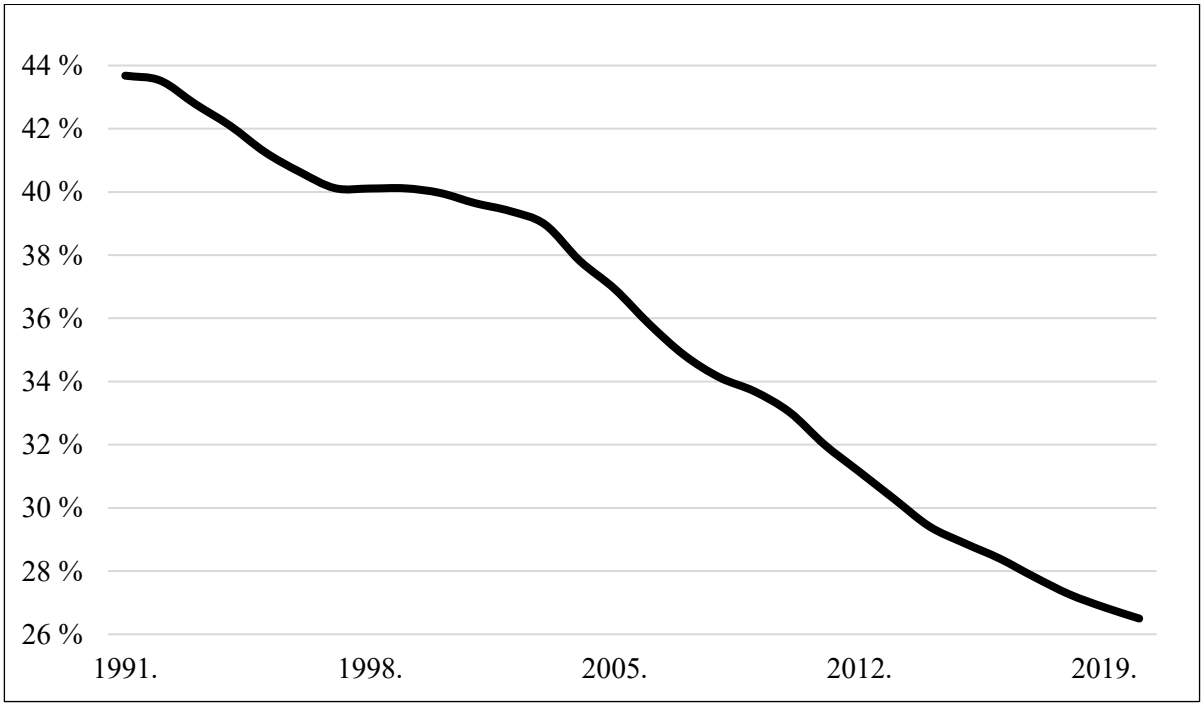
POPIS TABLICA

Tablica 1. Osnovne tehničke specifikacije uređaja [29] 13

Tablica 2. Mjerene fizikalne veličine Leap Motion Controllera 22

1. UVOD

Nekoć davno, svaki je čovjek morao biti lovac ili sakupljač da bi preživio. No, ljudi su svojom pameću i dosjetljivošću stvorili razne alate radi olakšanja toga posla. U poljoprivredi naprimjer, od jednostavnih alata poput štapova i plugova, do složenih strojeva poput automatiziranih traktora, napredovalo se do moderne poljoprivrede u kojoj sve manji postotak svjetske populacije proizvodi hranu, a svejedno je ima u izobilju. Slika 1. jasno prikazuje taj trend.



Slika 1. Postotak svjetske populacije zaposlen u poljoprivredi u razdoblju 1991.–2020. [1]

No, nije riječ samo o poljoprivredi, nego i o svim drugim područjima ljudskog djelovanja, pri čemu je stalna težnja da se umanji fizički rad bilo koje vrste.

Prije industrijskih revolucija, proizvodi su se izrađivali ručno i većinom pojedinačno. Obrtnik bi uobičajeno sâm stvarao svaki proizvod, a ako bi taj imao više dijelova, sve njih, da bi ih zatim sastavljao u konačan proizvod. [2]

Prva industrijska revolucija obilježila je prijelaz s ručnih metoda proizvodnje na strojeve koji koriste paru i vodu. Ovaj prijelaz trajao je duže vrijeme, pa se razdobljem na koje se to odnosi u Europi i Sjedinjenim Državama smatra ono između 1760. i 1840. godine. Tekstilna industrija bila je prva koja je usvojila promjene, a zatim su one uslijedile i u proizvodnji metala, poljoprivredi i rudarstvu.

Druga industrijska revolucija, poznata i kao Tehnološka revolucija, razdoblje je između 1871. i 1914. godine koje je proizašlo iz instalacija opsežnih elektroenergetskih, željezničkih i telegrafskih mreža, što je omogućilo i/ili ubrzalo prijenos električne energije, ljudi i ideja, kao i svake druge robe. Elektrifikacija je u tvornicama omogućila i razvoj moderne proizvodne linije.

Treća industrijska revolucija, poznata i kao Digitalna revolucija, dogodila se krajem 20. stoljeća. Velikom upotrebom računalnih i komunikacijskih tehnologija u proizvodnom procesu strojevi su počeli u potpunosti biti zamjena za ljude. [3]

Četvrta industrijska revolucija, poznata i kao industrija 4.0, prvi je puta spomenuta 2011. godine u tehnološkoj strategiji njemačke vlade koja promiče digitalizaciju proizvodnje. [4] Iste godine, na sajmu u Hannoveru, stručnjaci su ustanovili da je stigla nova industrijska revolucija s inovacijama koje su moderne tehnologije uvele u proizvodnju. Strojevi ovu automatizaciju duguju novim računalnim i komunikacijskim tehnologijama. [5] Neke od vodećih tehnologija koje to omogućuju su virtualna stvarnost, 3D ispis, *Internet of Things* (*IoT*) i tehnologije *blockchain*. [6]

Svaka od navedenih etapa razvoja imala je za glavnu svrhu postizanje što većeg profita i produktivnosti, a na temelju što učinkovitijeg iskorištenja rada. Budući da tjelesni rad podrazumijeva izvođenje niza pokreta, snimanje i analiza pokreta od suštinske je važnosti za oblikovanje i normiranje industrijskih procesa. [7] Stoga se upravo u ovome radu nastoji istražiti potencijal iskoristivosti virtualne stvarnosti (VR) za snimanje i analizu pokreta.

2. ZNAČAJ SNIMANJA I ANALIZE POKRETA

Kretanje je definirano kao promjena položaja tijela u odnosu na neki referentni sustav u vremenu. Opisuje se matematički pomoću pomaka, udaljenosti, brzine, ubrzanja i vremena. [8]

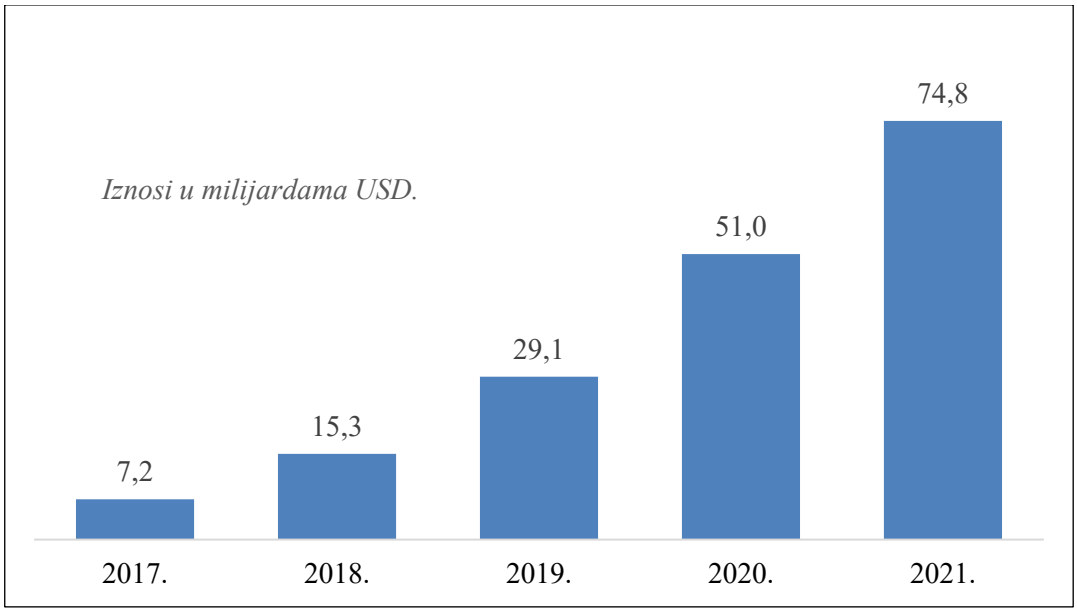
Iz navedenih veličina te mase tijela mogu se dobiti korisne informacije kao što su sila, rad, snaga i druge. Zbog toga je snimanje i analiza pokreta vrlo značajno u inženjerstvu, a razvoj tehnologija koje to omogućuju započeo je još tokom 19. stoljeća te se od tada kontinuirano nastavlja.

Prva industrijska revolucija dovela je do velikih promjena u kontekstu učinkovitosti industrijskog rada. Masovna proizvodnja učinila je bitnim i samu tehnologiju izrade pa su je zato tvrtke nastojale učiniti boljom kako bi maksimizirale profit. Menadžeri za upravljanje krajem 19. i početkom 20. stoljeća bili su jedni od prvih koji su primijenili znanstvene metode kako bi poboljšali učinkovitost proizvodnje. [9] Frederick Winslow Taylor jedan je od prvih zagovaratelja takvog znanstvenog pristupa.

Slika 2. Frederick Winslow Taylor (1856.–1915.) [10]

Taylor je sustavno razdvajao funkcije industrijskih procesa na postupke koji se mogu matematički analizirati kako bi se izvele promjene za učinkovitiju proizvodnju. Tijekom vremena i u različitim granama industrije, njegove su se metode koristile za primjenu na radnim mjestima, radnicima i onome što proizvode. Takav pristup naziva se još i tejlorizam (engl. *Taylorism*). Jedan od najpoznatijih primjera tejlorizma jest proizvodna linija.

Današnja tehnologija čini sve još jednostavnijim uz brzo i jeftino stvaranje, prikupljanje i obradu velikih količina podataka te djelovanje na osnovi njih. Stoga ne čudi da se u moderno doba tejlorizam proširio daleko izvan samih tvornica. Nove tehnologije također su omogućile mnogo složenije upravljanje učinkom zaposlenika nego za vrijeme Taylorova života. Odnedavno i male i velike tvrtke koriste digitalne alate. [11] Tako je i virtualna stvarnost sve prisutnija u snimanju i analizi pokreta. Slika 3. prikazuje tržište virtualne stvarnosti u periodu između 2017. i 2021. godine iz čega se vidi jasan trend rasta.



Slika 3. Rast industrije VR u razdoblju 2017.–2021. [12]

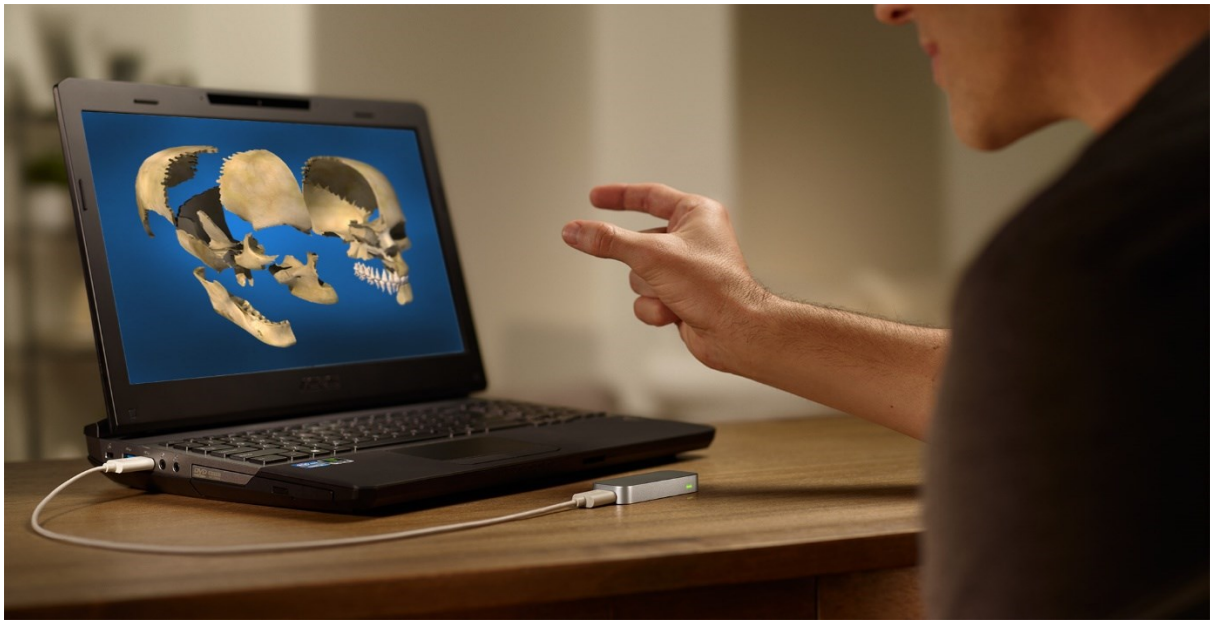
Sve više tvrtki prilagođava svoja radna mjesta za VR. To je postalo neophodno u današnjem svijetu kada mnogi radnici mogu raditi od kuće, posebice kod složenih projekata gdje rad na daljinu ne bi bio moguć samo uz pomoć klasičnih alata za video sastanke. Takav pristup omogućava nova interaktivna iskustva poput lakšeg donošenja tehničkih odluka, promoviranja boljeg timskog rada [13] te uvida u virtualne prototipove prije dostupnosti bilo kakvih fizičkih prototipova. [14]

Sveučilišta su najranije usvojila virtualnu stvarnost i još uvijek predvode u istraživanju i razvoju VR tehnologije. VR se također pokazala vrlo korisnim alatom u obrazovanju budućih

inženjera, i za nastavnike i studente. Ranije visoki troškovi u obrazovanju, zbog VR-a postaju mnogo niži. Najvažniji čimbenik je sposobnost upravljanja trodimenzionalnim modelima koji izgledaju i ponašaju se kao da su stvarni. Takav dodatan alat za obrazovanje omogućuje znatno lakše učenje u kontekstu shvaćanja složenih tema i njihovih primjena. [15]

VR može simulirati i stvarne radne prostore u svrhu zaštite na radu te za obuku osoblja u kojem ljudi mogu razvijati svoje vještine bez stvarnih posljedica neuspjeha. [16]

U današnje doba, VR tehnologija je mala, moćna i dovoljno jeftina da je mogu posjedovati milijuni ljudi [17].



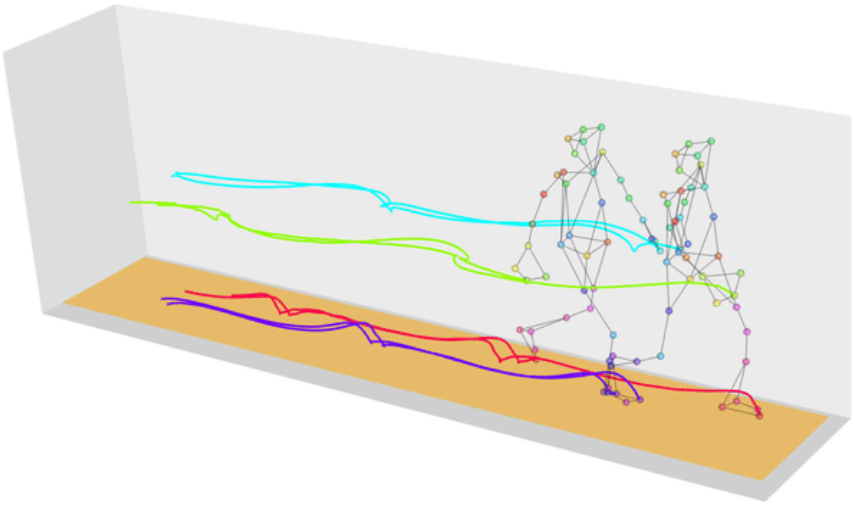
Slika 4. Upotreba virtualne stvarnosti u obrazovanju [18]

3. TRENUTAČNI STUPANJ RAZVIJENOSTI TEHNOLOGIJE I METODE PRAĆENJA POKRETA

U suvremeno doba proces snimanja i analize pokreta zahvaljujući modernoj tehnologiji lakši je nego ikada prije. Tradicionalan način primjenom tzv. *sustava unaprijed određenih vremena* (kao što je to naprimjer MTM-2) [19], još je uvijek prisutan, s time da se integrira s inženjerskim CAD/CAE/CAM softverima (Catia, Siemens NX) odnosno s novopridošlom opremom. [20]

Suvremeno praćenje i snimanje pokreta započelo je kao alat za fotogrametrijsku analizu u biomehaničkim istraživanjima između 1970. i 1980. godine, a kako je tehnologija sazrijevala proširila se i na obrazovanje, obuku, sport, animacije za televiziju, kino, video igre i ostalo.

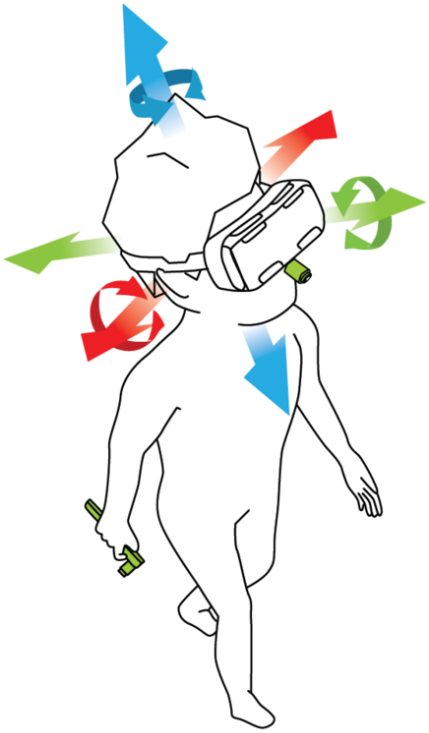
Od 20. stoljeća koriste se markeri koji se nose u blizini svakog zgloba kako bi se efektivno pratilo kretanje po položajima ili kutevima između markera. Akustičnim, inercijskim, LED, magnetskim ili reflektirajućim markerima i njihovim kombinacijama prate se pokreti optimalno najmanje dva puta brže od brzine željenog kretanja. Od početka 21. stoljeća zbog sve naprednije tehnologije razvijaju se nove metode. Većina modernih sustava može razlučiti između siluete korisnika i pozadine. Pomoću matematičkog modela iz siluete se mogu izračunati svi kutevi i položaji između zglobova. Dostupni su i hibridni sustavi koji mogu raditi oboje (markeri i silueta) uz manje markera nego prije. [21]



Slika 5. Snimanje ljudskog hoda u virtualnoj stvarnosti [22]

U kontekstu virtualne stvarnosti, snimanje pokreta uključuje precizno praćenje položaja VR naočala, kontrolera te drugih predmeta ili dijelova tijela unutar nekog fizičkog prostora. Budući da je svrha virtualne stvarnosti oponašanje prave stvarnosti, najvažnije je da praćenje bude točno i precizno kako ne bi razbilo iluziju trodimenzionalnog prostora.

Da bi se to postiglo, razvijeno je nekoliko metoda praćenja položaja i orijentacije VR naočala i svih povezanih objekata ili uređaja. Gotovo sve metode koriste senzore koji kontinuirano bilježe signale s odašiljača na ili u blizini praćenih predmeta (objekata), a zatim te podatke šalju računalu kako bi se spremila i obradila njihova približna fizička lokacija. Uglavnom se ta fizička mjesta identificiraju i definiraju pomoću jednog ili više koordinatnih sustava: kartezijev, polarni ili cilindrični sustav. Mnoga sučelja također su dizajnirana za praćenje i kontrolu nečijeg kretanja unutar virtualnog 3D prostora te interakciju s tim prostorom. Takva sučelja moraju usko surađivati sa sustavima za praćenje kako bi pružila ugodno korisničko iskustvo. [23]



Slika 6. Praćenje pokreta glave u virtualnoj stvarnosti [24]

Postoji više načina praćenja i snimanja pokreta, od kojih će neki od njih biti prikazani u nastavku [25].

3.1. MEHANIČKO PRAĆENJE

Korisnik nosi egzoskelet koji je zakačen za leđa korisnika. Dok se korisnik kreće, egzoskelet se također kreće, a senzori u svakom zglobu bilježe rotacije. Druge vrste praćenja pokreta uključuju rukavice, mehaničke ruke ili zglobne modele.

Prednosti: nema smetnji svjetlosti ili magnetskih polja.

Mane: tehnologija nema „svijest“ o tlu, tako da se ne može snimati skok korisnika, podaci sa stopala neprecizni su, oprema se mora često kalibrirati, nema podataka o smjeru u kojem pokazuje tijelo korisnika, apsolutni položaji nisu poznati.

3.2. OPTIČKO PRAĆENJE

Korisnik na tijelu nosi reflektirajuće oznake (markere) koje prati sustav kamera u svrhu određivanja pozicija navedenih markera u prostoru. Markeri su reflektirajući ili IR emitirajući, od kojih su mnogi razvijeni za glazbene primjene (poput dirigiranja). Koristi se za biomedicinsku primjenu (sportske ozljede, analiza sportskih performansi i slično).

Prednosti: korisnik se osjeća slobodno jer nema kabela koji povezuju tijelo s opremom, mogući veći predmeti rada, moguće je više korisnika, vrlo čisti (bez šuma) i detaljni podaci.

Mane: sklonost svjetlosnoj interferenciji, korisnici ili drugi objekti mogu blokirati reflektirajuće točke što uzrokuje gubitak podataka iako je moguća nadoknada softverom koji procjenjuje položaj nedostajućih točaka, rotacije dijelova tijela nisu apsolutne, korisnik mora nositi odijelo s točkicama i kuglicama što može biti neugodno, informacije moraju biti naknadno obrađene prije gledanja tako da izvođač ne može vidjeti svoju sliku pa ne može biti toliko kreativan ili prepoznati potencijalne probleme kod snimanja pokreta, veći trošak od (elektro)magnetskog.

3.3. ELEKTROMAGNETSKO (MAGNETSKO) PRAĆENJE

Korisnik nosi niz magnetskih prijamnika koji prate položaj u odnosu na statički magnetski odašiljač. Jedna od prvih primjena ove metode bila je u vojnoj industriji za praćenje kretanja glave pilota.

Prednosti: pozicije i rotacije su apsolutne, moguće određivanje orijentacije u prostoru, snimanje podataka moguće u stvarnom vremenu što omogućuje trenutno emitiranje i priliku da korisnici primaju trenutne povratne informacije (više spontanosti u izvedbi), jeftiniji od optičkog.

Mane: magnetska distorzija javlja se s povećanjem udaljenosti, podaci mogu sadržavati šum, sklonost smetnjama magnetskih polja, korisnik mora nositi kabele koji ga povezuju s opremom što ograničava slobodu kretanja, brzina uzorkovanja preniska je za mnoge sportske upotrebe.

3.4. OSTALE VRSTE TEHNOLOGIJA PRAĆENJA POKRETA

Ostale vrste tehnologija praćenja pokreta uključuju:

- Zvučnu, koja koristi ultrazvuk i podložna je nekoliko vrsta smetnji.
- *Biofeedback,* kojim se mjere tjelesni pokreti iz srca, mozga, mrežnice, očiju, kože i mišića, a koristi se u biomehaničke i sportske svrhe ali se može koristiti i za glazbene izvedbe.
- Snimanje električnog polja, u kojem tijelo djeluje kao odašiljač ili izvor smetnji koje se mjeri.
- Inercijske sustave, koji mjere ubrzanje, orijentaciju, kut nagiba i druge karakteristike.
- Video, koji koristi optičke tehnologije koje mogu otkriti promjene u luminiscenciji i boji.

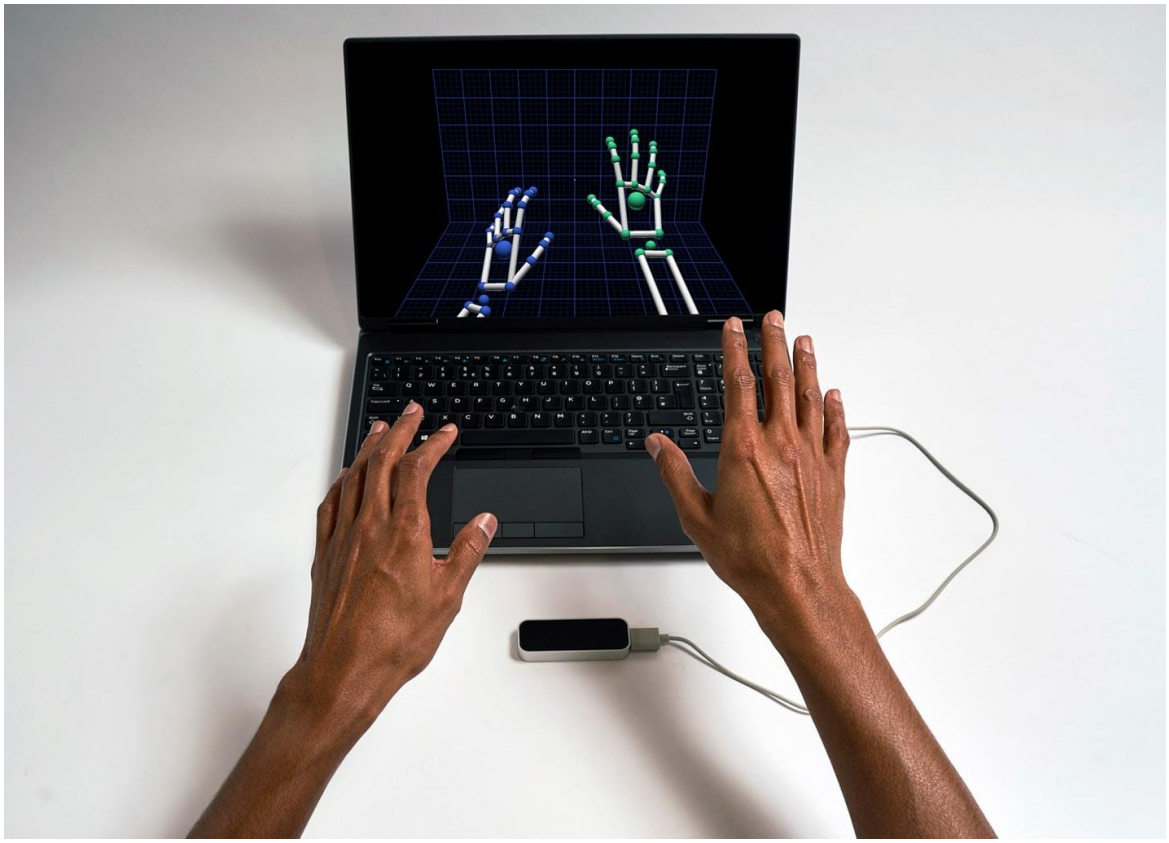
4. KONTROLER *LEAP MOTION*

Kontroler Leap Motion kompanije Ultraleap optički je uređaj koji bilježi kretanje ruku i prstiju korisnika za interakciju s računalom tj. digitalnim sadržajem na njemu (Slika 7.). Malen, brz i precizan, Leap Motion kontroler može se koristiti s računalima, biti integriran u hardverska rješenja ili zaslone, pričvršćen na VR naočale za izradu prototipa, istraživanje i razvoj i slično.

Slika 7. Kontroler Leap Motion [26]

Primjeri uporabe jesu [27] (Slika 8.):

- javna sučelja bez dodira (interaktivni kiosci, dizala)
- zabava (VR/AR, zabavni parkovi)
- zdravstvo (fizička rehabilitacija nakon moždanog udara, medicinsko snimanje)
- terapija i obrazovanje (anatomske vizualizacije, praktično učenje)
- obuka osoblja (simulatori leta, složeni kontrolni sustavi)
- industrijski dizajn i inženjerstvo (automobilska industrija, proizvodne linije)
- robotika
- suradnja na daljinu.



Slika 8. Kontroler Leap Motion u uporabi [28]

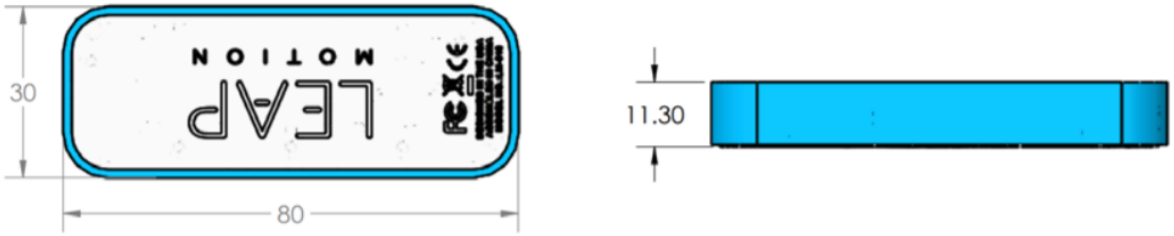
Od ranih hardverskih prototipova do najnovijeg softvera za praćenje, kontroler Leap Motion znatno je napredovao. Slika 9. prikazuje postepeni razvoj uređaja između 2011. i 2012. godine.

Slika 9. Postepeni razvoj kontrolera Leap Motion [28]

Osnovne tehničke specifikacije uređaja dane su tablicom 1. a njegove osnovne dimenzije slikom 10.

Tablica 1. Osnovne tehničke specifikacije uređaja [29]

Napajanje	5V DC pomoću USB kabla (minimalno 0,5 A)
Podatkovna veza	USB 2.0/3.0
Zaštita	Otporno na prskanje
Načini montaže	Radna površina, VR naočale ili ugradnja u postojeću hardversku instalaciju.
Interaktivna zona	60 do 80 cm. Vidno polje 140x120°.
Kamere	Dvije IR kamere, 640x240 piksela razmaka 40 mm, 850 nm (+/– 25 nm), 120 Hz, brzina okidača 1/2000 s.
Sučelje kamera	Podrška za programske jezike C, Python i Matlab te programski paket OpenCV.
LED oprema	Tri LED svjetla
Konstrukcija	Aluminij i staklo otporni na grebanje.
Optimalna temperatura rada	0 do 45ºC
Temperatura skladištenja	–10 do 50ºC
Relativna dopuštena vlažnost	5 do 85 % (bez kondenzacije)
Dopuštena nadmorska visina	0 do 3048 m
Usklađenost sa standardima	CE, FCC, CAN ICES-3, REACH, RoHS
Minimalni sistemski zahtjevi	Windows 7+ ili MacOS 10.7 (MacOS nije više formalno podržan), AMD Phenom II ili Intel Core i3/i5/i7 procesor, 2 GB RAM, USB 2.0 ulaz.

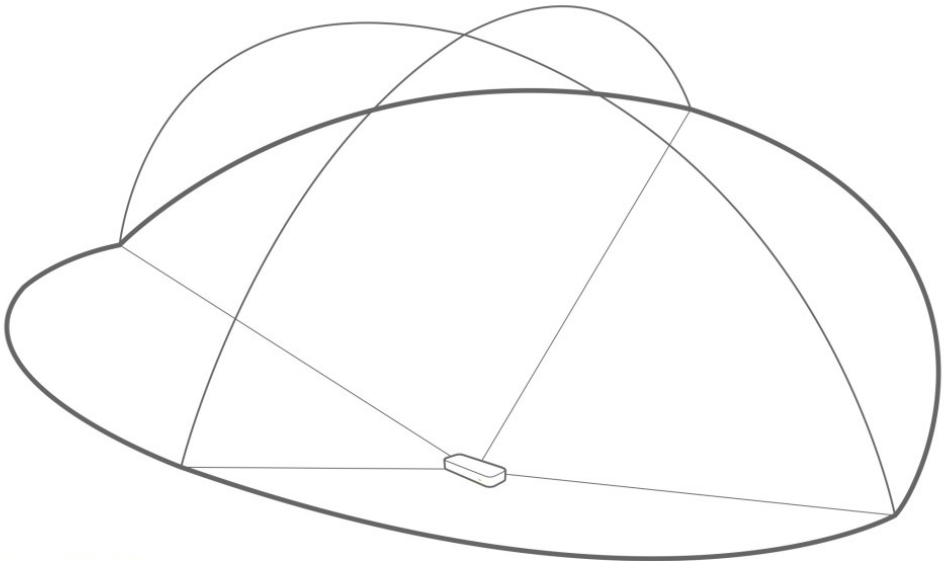


Slika 10. Gabaritne dimenzije kontrolera Leap Motion [30]

4.1. PRINCIP RADA UREĐAJA [31]

4.1.1. Hardver

U hardverskom smislu, kontroler Leap Motion zapravo je vrlo jednostavan uređaj. Srž kontrolera čine dvije kamere i tri infracrvene LED lampice koje stvaraju infracrvenu svjetlost valne duljine 850 nm koja je izvan spektra vidljive svjetlosti.

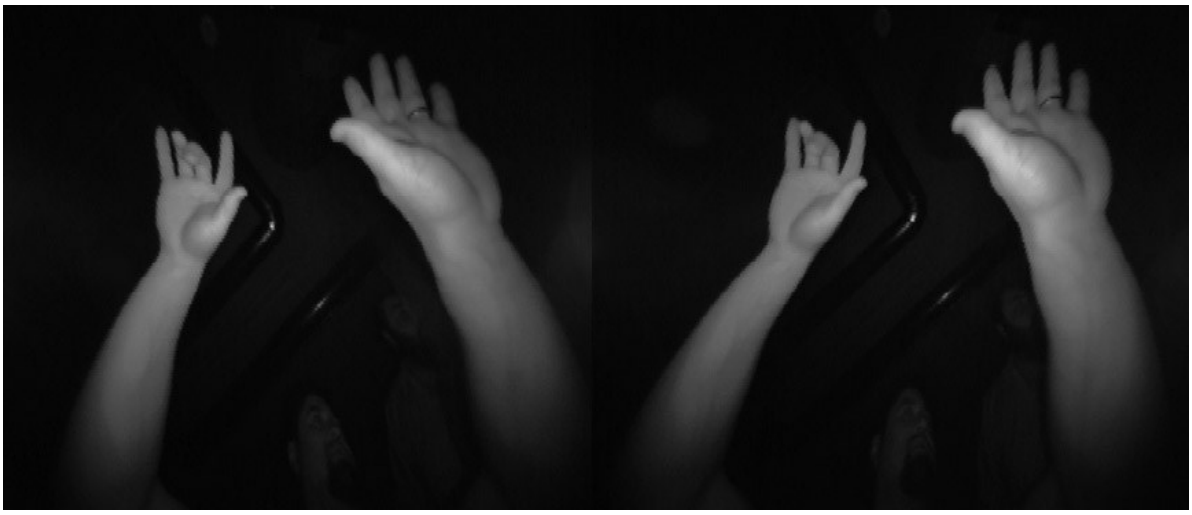


Slika 11. Oblik prostora za interakciju

Zbog svojih širokokutnih leća, kontroler Leap Motion ima veliki prostor interakcije od otprilike 225 dm^3 oblika obrnute piramide što je zapravo presjek vidnih polja dviju kamera. Maksimalni domet Leap Motion kontrolera je na otprilike 80 cm udaljenosti od uređaja. Ovaj raspon je ograničen širenjem infracrvenog LED svjetla kroz prostor jer postaje mnogo teže snimiti položaj ruku u prostoru izvan određene udaljenosti. Intenzivnost LED svjetla u konačnici je ograničena maksimalnom strujom koju USB veza podržava.

USB kontroler čita podatke senzora u vlastitu lokalnu memoriju i izvodi sve potrebne prilagodbe razlučivosti. Ti se podaci zatim putem sučelja USB prenose u istoimeni softver za praćenje Leap Motion.

Podaci imaju oblik sive stereo slike infracrvenog svjetlosnog spektra, odvojene u lijevu i desnu kameru. Tipično su jedini predmeti koji se mogu vidjeti oni izravno osvijetljeni IR LED svjetlom kontrolera Leap Motion. Međutim, žarulje sa žarnom niti, halogene žarulje i dnevna svjetlost također mogu osvjetljavati scenu unutar spektra infracrvenog zračenja. Tako se neke stvari, poput pamučnih košulja, mogu činiti bijele iako su zapravo tamne u vidljivom spektru.



Slika 12. Prikaz slike kamera

4.1.2. Softver

Kontroler Leap Motion primjenjuje napredne algoritme nad neobrađenim podacima senzora. Kada se podaci prenesu u računalo softver Leap Motion ih obrađuje. Nakon filtriranja pozadinskih objekata i osvjetljenja okoliša, podaci se analiziraju kako bi se dobio 3D prikaz onoga što uređaj vidi.

Nakon analize, dobivaju se detaljne informacije o pokretima poput broja prstiju i alata koji se koriste u datom trenutku. Napredni algoritmi također zaključuju i o položaju objekata koji se ne mogu vidjeti. Primjenjuju se i neke tehnike filtriranja kako bi se osigurala nesmetana vremenska usklađenost podataka. Leap Motion softver zatim te rezultate izražene kao niz snimaka (*Frame*) koji sadrže sve podatke praćenja uvodi u tzv. transportni protokol.

Kroz ovaj protokol softver komunicira s programom Leap Motion Control Panel, kao i s web klijentima putem standardnih web protokola (TCP lokalno, WebSocket za web). Podaci se također organiziraju u objektno orijentiranu API strukturu koja sadrži povijest svih snimaka (*Frame*) te pruža pomoćne funkcije i klase. Odatle se neka aplikacija može povezati na kontroler Leap Motion.

4.2. LATENCIJA [32 i 33]

Latencija je važan čimbenik u postizanju ispravnog doživljaja bilo kojeg interaktivnog sučelja. Kontroler Leap Motion ima manju latenciju od ostalih sličnih proizvoda na tržištu.

Postoji određena granična latencija koju podnosi ljudski vidni i živčani sustav kod kojeg je kašnjenje još uvijek neprimjetno. Ta granica varira od osobe do osobe, međutim otkriveno je da je zadovoljavajući prag u prosjeku otprilike 30 ms.

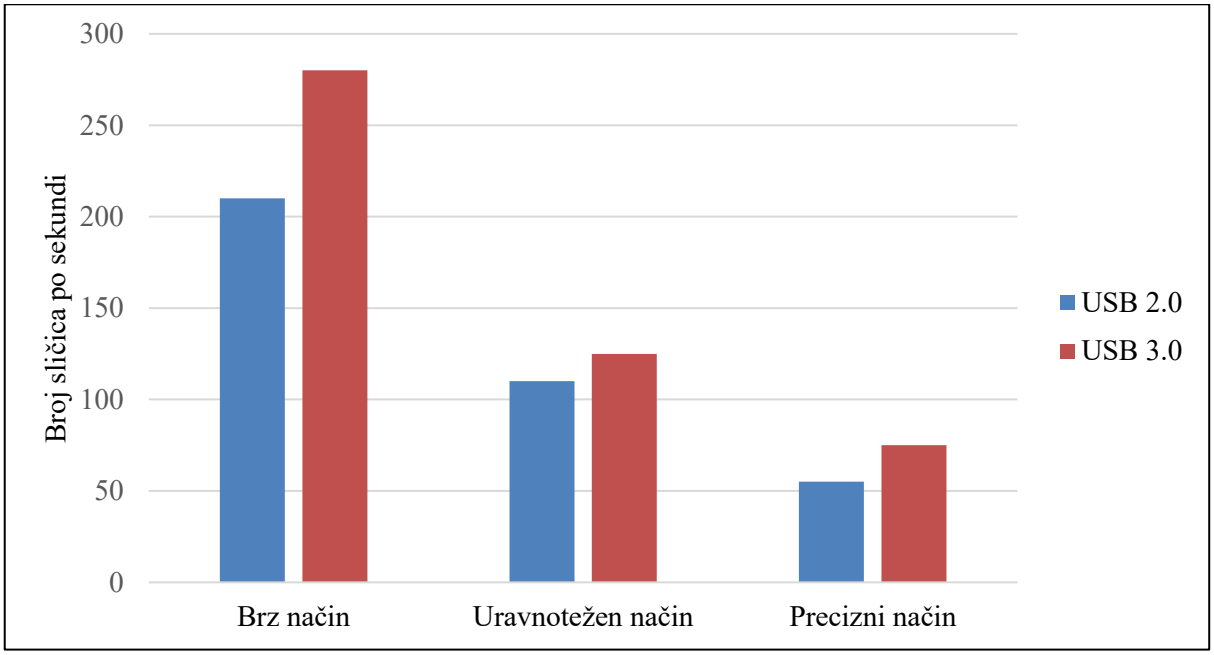
Postoje razni razlozi kašnjenja između vremena kada se pokret dogodi i dok se pokret u potpunosti ne obradi te njegov rezultat bude vidljiv.

4.2.1. Hardverska razina

Latencija započinje na hardverskoj razini. Kontroler Leap Motion tako sadrži tri postavke kvalitete načina rada:

- Brz način rada: smanjuje razlučivost infracrvenih senzora na četvrtinu njihove pune razlučivosti te na taj način skoro pa učetverostručuje brzinu praćenja.
- Precizan način rada: omogućuje smanjenje brzine praćenja na približno 40 % svoje nominalne vrijednosti bez utjecaja na razlučivost, što smanjuje iskorištenost procesora i kontrolera Leap Motion.
- Uravnotežen način rada: kombinacija gornjih načina.

Usporedba ova tri načina rada dana je na slici 13.



Slika 13. Različiti načini rada i njihov utjecaj na brzinu

Postavka načina rada uređaja ima velik utjecaj na rad hardvera. Osim promjene izlazne razlučivosti IR senzora, postavka utječe i na LED modulaciju te na USB kontroler unutar uređaja.

Postupak prijenosa podataka preko USB sučelja s uređaja na računalo najveći je izvor latencije na hardverskoj razini. Točno vrijeme izvođenja ovog prijenosa ovisi o postavci načina rada, koristi li se USB 2.0 ili USB 3.0, pa čak i specifikacijama USB kontrolera u glavnom računalu. Jeftinija računala imaju USB kontrolere koji jedva rade i na polovini brzine kvalitetnijih kontrolera.

4.2.2. Sustav prikaza računala

Zadatak sustava prikaza na računalu je primiti opis neke scene te ju prikazati na fizičkom ekranu. Na rad ovog sustava utječe više čimbenika:

- pozadinska aplikacija
- operativni sustav
- *driver* grafičke kartice
- grafička kartica
- elektronski sklop u samom ekranu.

Da bi ovaj kompliciran sustav ispravno radio, postoji mnogo privremenih međuspremnika koji djeluju kao posrednici između jedne faze obrade i sljedeće. Vrijeme potrebno za kopiranje, obradu i čitanje tih međuspremnika tim je veće što je veći broj međuspremnika.

Primarni faktori zbog kojih sustav kasni su vertikalna sinkronizacija (*VSync*) i elektronski sklop u samom ekranu. *VSync* je metoda za sprečavanje "kidanja slike“ na ekranu koje je uzrokovano istovremenim prikazivanjem više slika stvorenih u različito vrijeme. *VSync* osigurava da se na ekranu prikazuje slika iz samo jednog trenutka u vremenu. Da bi to bilo osigurano, moraju se odgoditi raniji koraci u procesu sve dok grafički upravljački programi nisu spremni za slanje sljedeće slike na zaslon. Takvo namjerno odgađanje puno doprinosi kašnjenju i može se samo donekle ublažiti.

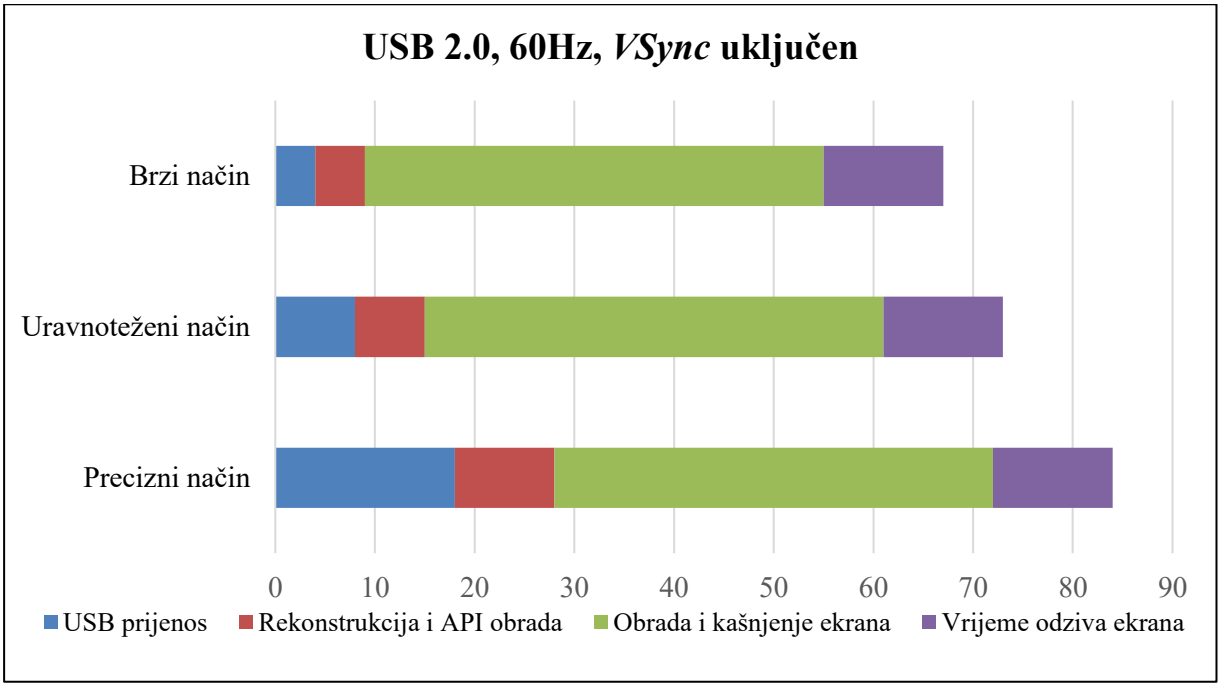
Kašnjenje prikaza na samom ekranu drugi je glavni faktor koji doprinosi kašnjenju u sustavu. Vrlo rijetko može se naći ekran koji istovremeno ima nisko kašnjenje, velike kutove gledanja, dobru točnost boja, visoku razlučivost i male dimenzije. Općenito, konzumeristički ekrani žrtvovat će točnost boja u korist brzine, dok će profesionalni ekrani učiniti suprotno. Ekran mora pretvoriti neobrađeni ulazni video signal s grafičke kartice računala u format koji hardver ekrana može razumjeti. Neobrađeni slikovni podaci zatim se obrađuju kako bi se prikaz prilagodio odgovarajućoj razlučivosti, izvršila podešavanja svjetline, kontrasta, boje te po želji izvršili neki dodatni zadaci poput izoštravanja.

Također, od utjecaja je i vrijeme odziva piksela unutar ekrana. Na primjer, u ekranu s tekućim kristalima (LCD) mali setovi kristala za svaku boju (crvena, zelena i plava) prilagođavaju svoju polarizaciju kako bi omogućili prolazak različitih količina bijelog pozadinskog osvjetljenja. Ti se kristali ne mogu odmah polarizirati, pa je potrebno određeno vrijeme odziva.

4.2.3. Analiza čimbenika i optimiranje latencije

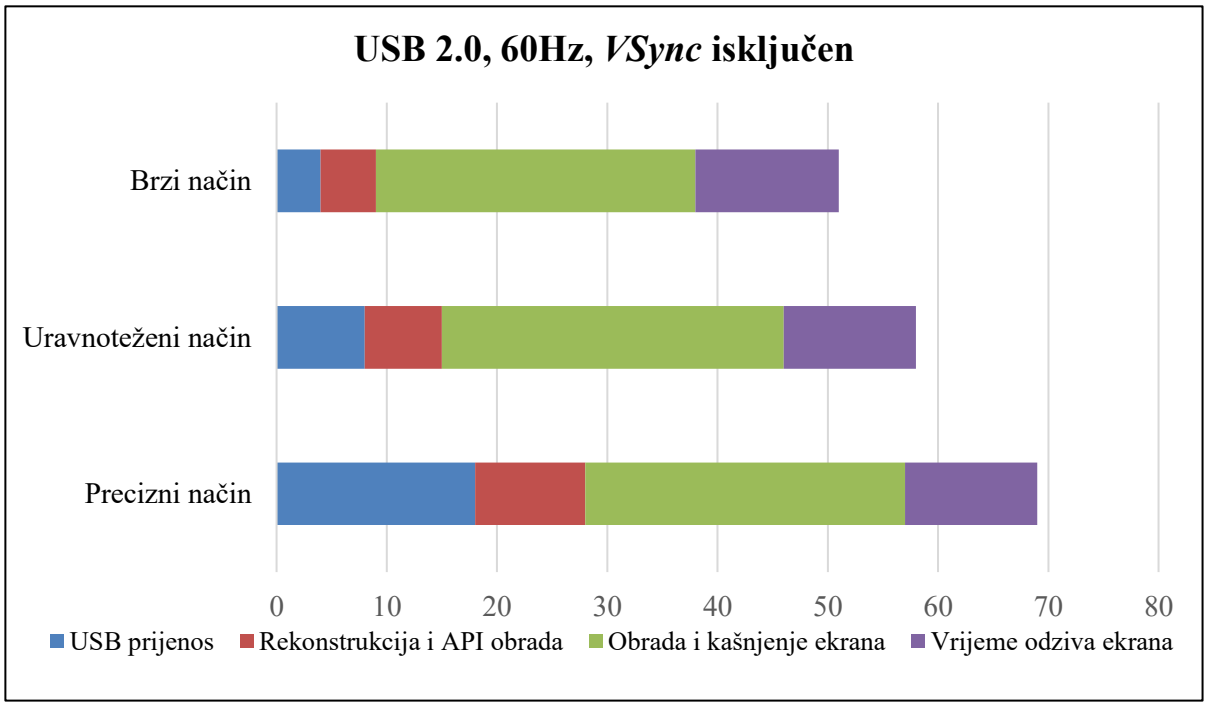
Najbolji način da se vide učinci različitih, gore navedenih čimbenika jest pogledati njihov doprinos u različitim scenarijima. Navedena mjerenja predstavljaju prosjek mjerenja s nekoliko različitih konfiguracija računala.

Na slici 14. prikazana je najgora situacija u kojoj se koristi sporije USB 2.0 sučelje i monitor od 60Hz uz *Vsync* uključen



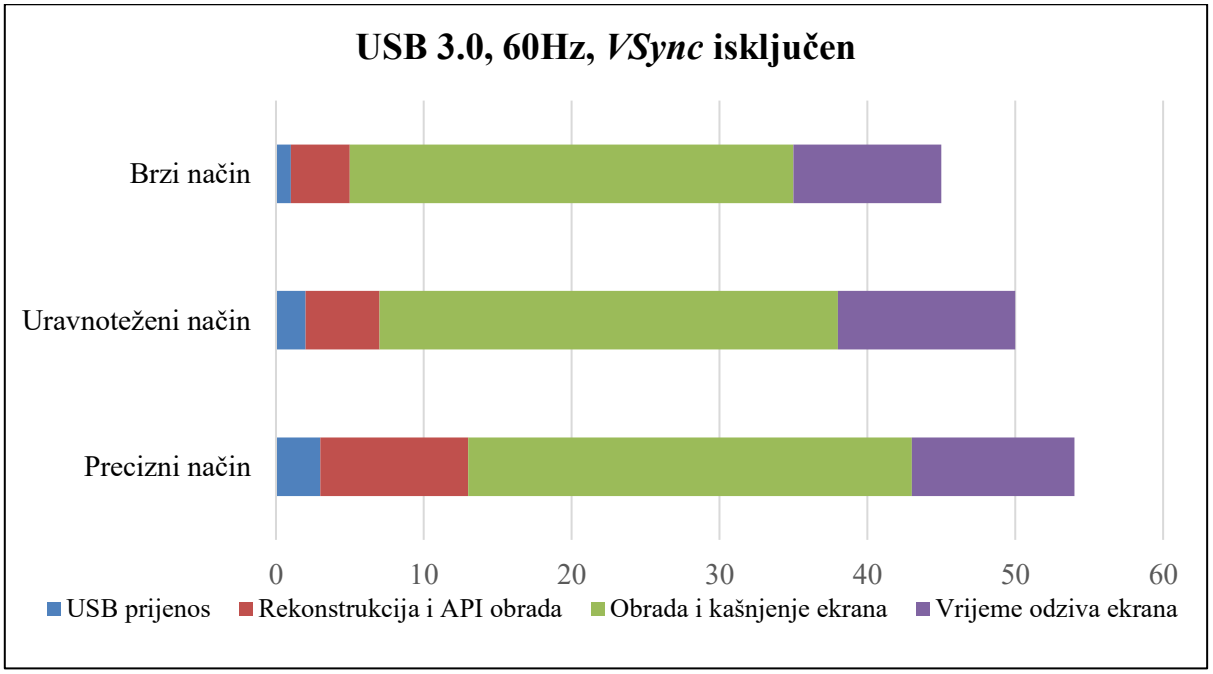
Slika 14. Latencija u najgorem slučaju

Isključivanjem Vsync opcije štedi se od 16 do preko 30 ms (Slika 15.). Međutim, nedostatak toga je prisutnost „kidanja slike“ (kao što je već rečeno).



Slika 15. Latencija uz Vsync isključen

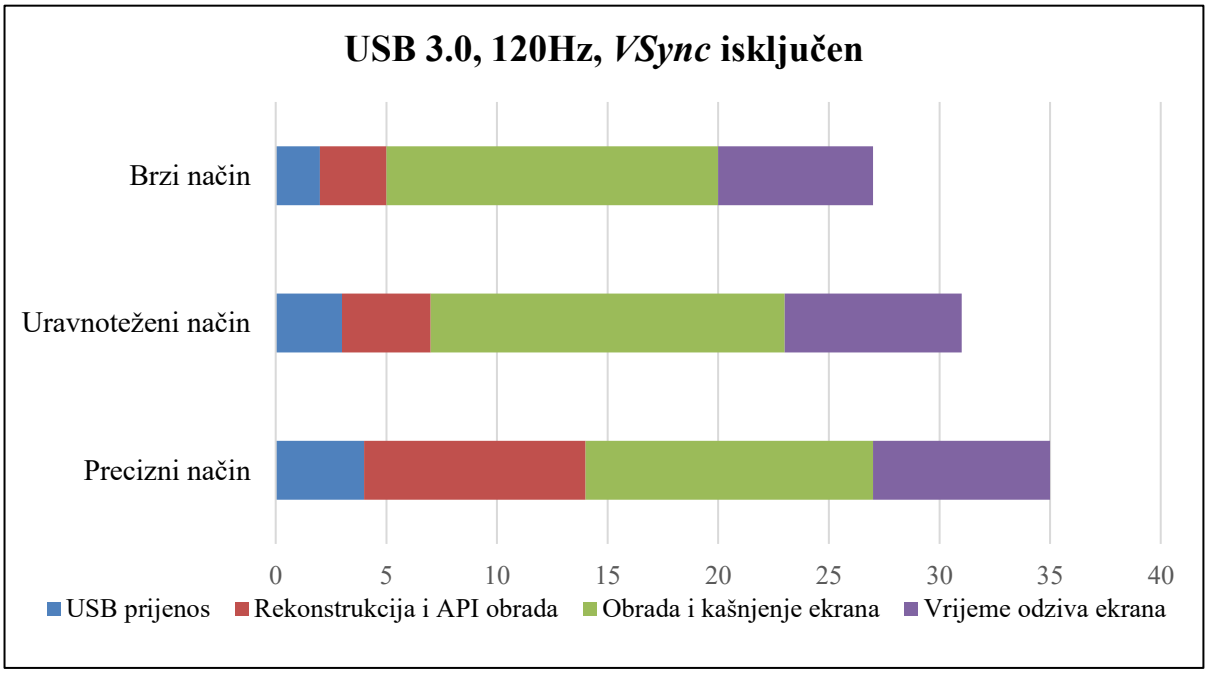
Uz USB 3.0 kabel i priključak može se drastično smanjiti vrijeme prijenosa podataka između uređaja i računala. Slika 16. prikazuje usporedbu s ostalim izvorima latencije.



Slika 16. Latencija uz USB 3.0

Sa slike 16. vidljivo je da u brzom načinu rada hardver i softver zajedno doprinose tek nešto više od 10 % ukupne izmjerene latencije.

Ekrani visokih performansi danas su prilično česti te pružaju smanjeno ulazno kašnjenje i vrijeme odziva ekrana. Slika 17. prikazuje najbolji mogući slučaj. U brzom načinu rada vidljivo je da je postignuta ranije spomenuta latencija manja od 30 ms.



Slika 17. Latencija u najboljem mogućem slučaju

Većina izmjerene latencije je izvan kontrole uređaja. Softver se može nadograditi kako bi se smanjilo opterećenje procesora i povećao odziv. Međutim, iz podataka je jasno da ukupna opažena latencija ovisi o puno čimbenika koji su izvan kontrole uređaja i softvera.

Za optimizaciju latencije preporuča se sljedeće:

- Korištenje računala s USB 3.0 priključkom i kabelom za veće brzine prijenosa podataka iz kontrolera Leap Motion.
- Monitor s mogućnošću osvježavanja slike od 120 Hz i malim vremenom odziva (do 5 ms). Danas postoji mnogo ekrana koji će raditi dobro, ali kao rezultat toga žrtvovat će široke kutove gledanja i točnost boja.
- U postavkama upravljačkog programa grafičke kartice najbolje je isključiti vertikalnu sinkronizaciju (*VSync*). Ako *VSync* ostane uključen, preporučljivo je uključiti trostruko međuspremništvo (*Triple buffering*) kako bi se nadoknadila neka dodatna kašnjenja.
- Ukoliko je potrebna minimalna latencija postoji način rada velike brzine koji smanjuje kvalitetu uzorkovanja u korist brzine.
- U krajnjem slučaju može se koristiti i ekran CRT. Ovi glomazni analogni ekrani nemaju ulazno kašnjenje, imaju veće brzine osvježavanja i efektivno nemaju vremena odziva. Međutim, sve ih je teže pronaći na tržištu.

4.3. PROGRAMSKA PODRŠKA [34 i 35]

Leap Motion ima veliku zajednicu od 350 000 programera diljem svijeta te jasnu i lako dostupnu tehničku dokumentaciju. Također sadrži i SDK te API koji služe za pristup sirovim podacima uređaja u svakom trenutku kako bi se mogle razvijati interaktivne aplikacije. Softver za praćenje Leap Motion znatno se mijenjao tijekom godina uz postepene nadogradnje.

Softver V2 nije više službeno podržan jer su mnogi razvojni resursi povezani s ovim softverom zastarjeli. Preporučuje se upotreba novijih inačica.

Softver V3 (poznatiji i kao Orion) dodatno je optimiziran za VR. Također je sačuvao API za mnoge programske jezike (C++, C#, Objective-C, Java, Python i JavaScript).

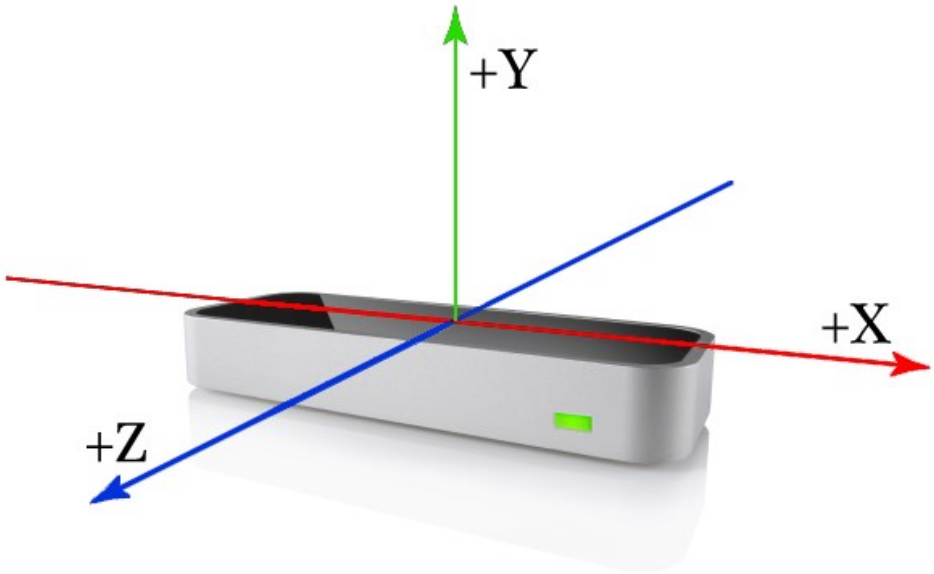
Softver V4 (druga generacija Oriona) odbacio je stariju API podršku za do tada podržane programske jezike u korist znatno brže LeapC API podrške. Ovo izdanje softvera

također sadrži i mogućnost vezivanja nepodržanih programskih jezika, što olakšava integraciju starijih aplikacija s LeapC API-jem.

Softver V5 (Gemini) trenutno je dostupan samo na Qualcomm Snapdragon XR2 5G platformi. Buduća izdanja proširit će softver i izvan XR2 čipa, a opće izdanje je u izradi.

4.3.1. Koordinatni sustav

Kontroler Leap Motion koristi desnokretni kartezijski koordinatni sustav s ishodištem na vrhu uređaja. Osi *X* i *Z* leže u vodoravnoj ravnini, a os *X* paralelna je s dužim rubom uređaja. Os *Y* je vertikalna, a pozitivne vrijednosti joj idu prema gore (za razliku od orijentacije prema dolje većine koordinatnih sustava u računalnoj grafici), os *Z* ima pozitivne vrijednosti koje se povećavaju prema korisniku.



Slika 18. Koordinatni sustav kontrolera Leap Motion

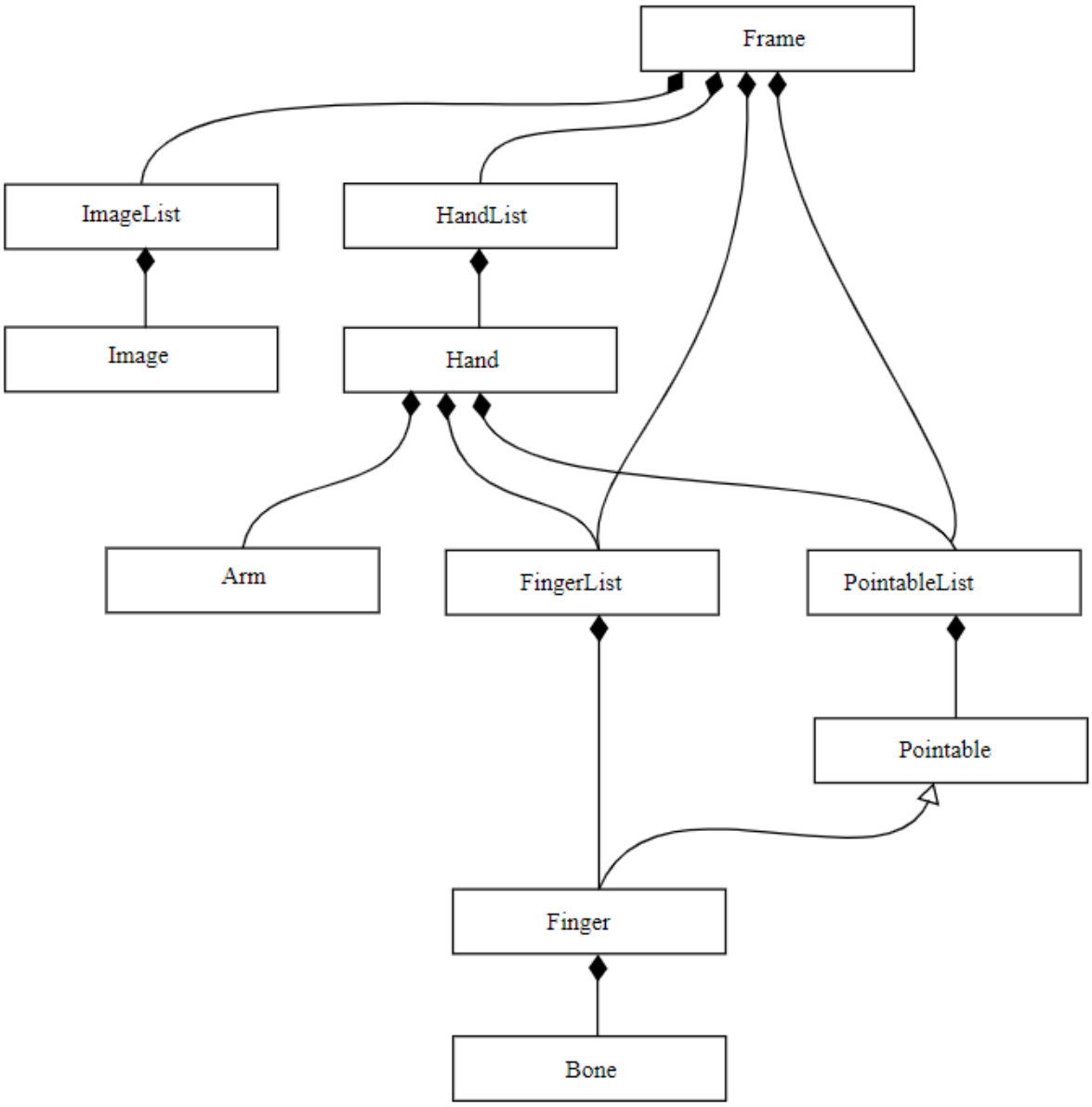
Kontroler Leap Motion mjeri fizikalne veličine u mjernim jedinicama danim u tablici 2.

Tablica 2. Mjerene fizikalne veličine kontrolera Leap Motion

Fizikalna veličina	Mjerna jedinica
Udaljenost	m
Vrijeme	s
Brzina	m/s
Kut	°

4.3.2. Podaci o praćenju kretanja

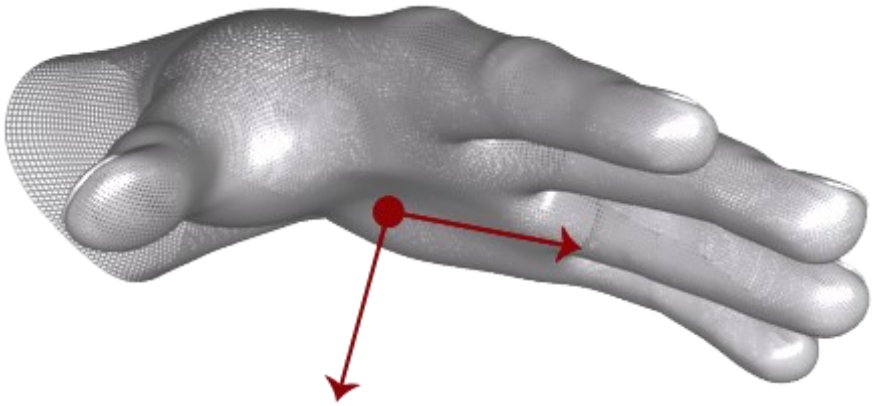
Dok kontroler Leap Motion prati šake, prste i ostale objekte u svom vidnom polju, pruža ažuriranja kao skup podataka. Preko klase *Frame* dobivaju se informacije o praćenim rukama s detaljima njihovih svojstava u datom trenutku. Klasa *Frame* u osnovi je korijen podatkovnog modela Leap Motion. Na slici 19. jasno se vide sve klase kojima neki programski jezik može pristupiti.



Slika 19. Objekti dostupni kroz API

4.3.2.1. Šaka

Model šake (Slika 20.) pruža informacije o identitetu (nazivu), položaju i drugim karakteristikama svake šake koja se nalazi u prostoru za interakciju, ruci na koju je šaka pričvršćena te popisima prstiju povezanih sa šakom. Šaku predstavlja klasa *Hand*.

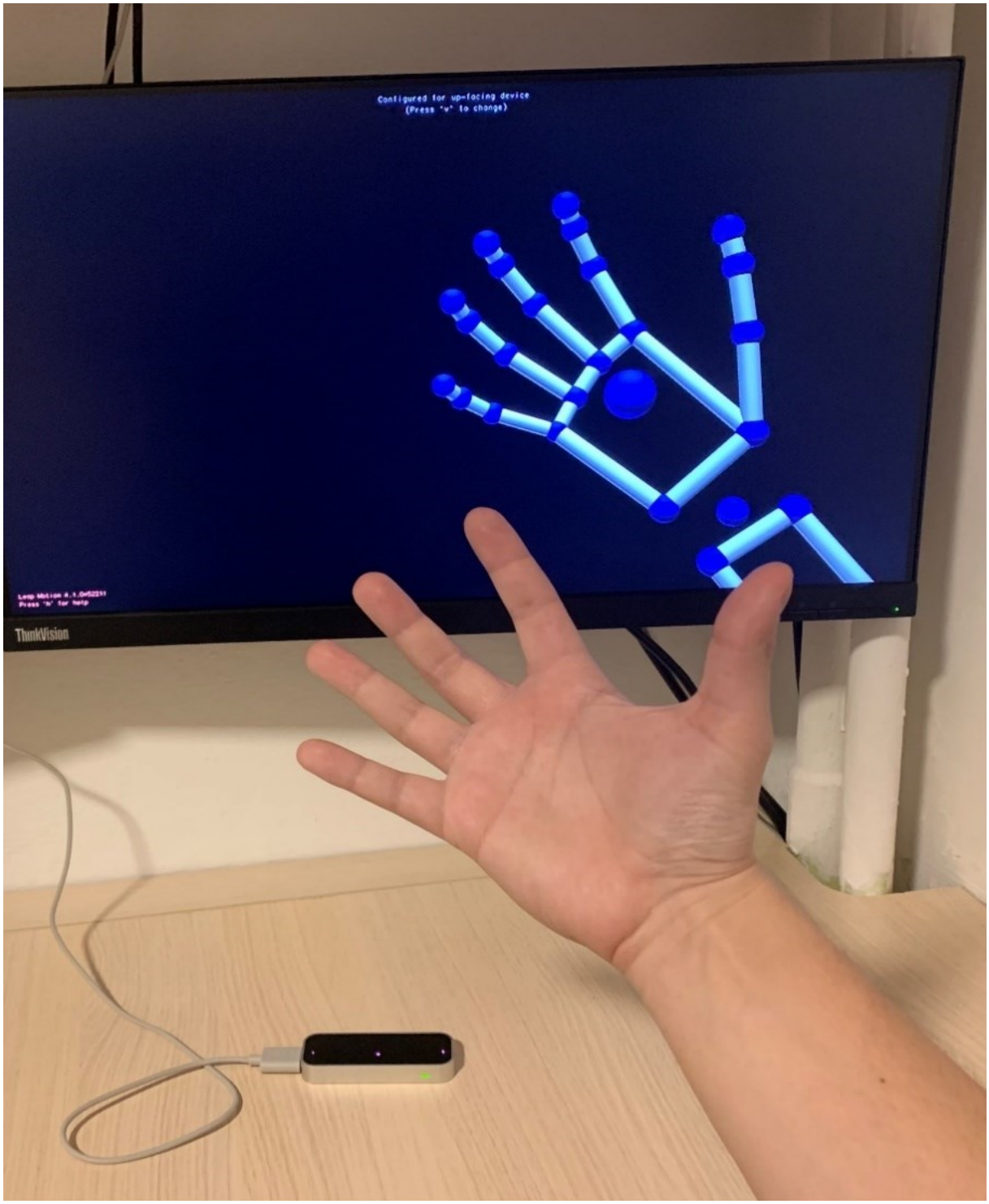


Slika 20. Model šake uz prikaz vektora normale na dlan i orijentacije

Slika 21. prikazuje usporedno zaslon s modelom šake Leap Motiona i stvarnu šaku.

Softver Leap Motion koristi unutarnji model ljudske šake kako bi pružio predviđanje praćenja čak i kada dijelovi šake nisu vidljivi. Model šake uvijek osigurava podatke o položajima svih pet prstiju, iako je praćenje optimalno kad se jasno vide silueta ruke i svi njezini prsti. Softver koristi vidljive dijelove šake, njegov unutarnji model i prošla promatranja za izračunavanje najvjerojatnijeg položaja dijelova koji trenutno nisu vidljivi. Suptilne pokrete prstiju prislonjenih uz ruku Leap Motion kontroler obično ne može otkriti. Ocjena pouzdanja *Hand.confidence* pokazuje koliko dobro promatrani podaci odgovaraju internom modelu.

Unutar *Frame* objekta mogu se pojaviti i više od dvije ruke. Međutim, preporuča se držanje najviše dvije ruke u vidnom polju kontrolera Leap Motion za optimalnu kvalitetu praćenja pokreta.



Slika 21. 3D model šake u odnosu na pravu

4.3.2.2. Ruka

Ruka je objekt koji pruža orijentaciju, duljinu, širinu i krajnje točke ruke. Kad lakat nije vidljiv, Leap Motion Controller procjenjuje njegov položaj na temelju prošlih opažanja i tipičnih ljudskih proporcija. Ruku predstavlja klasa *Hand.*

4.3.2.3. Prsti

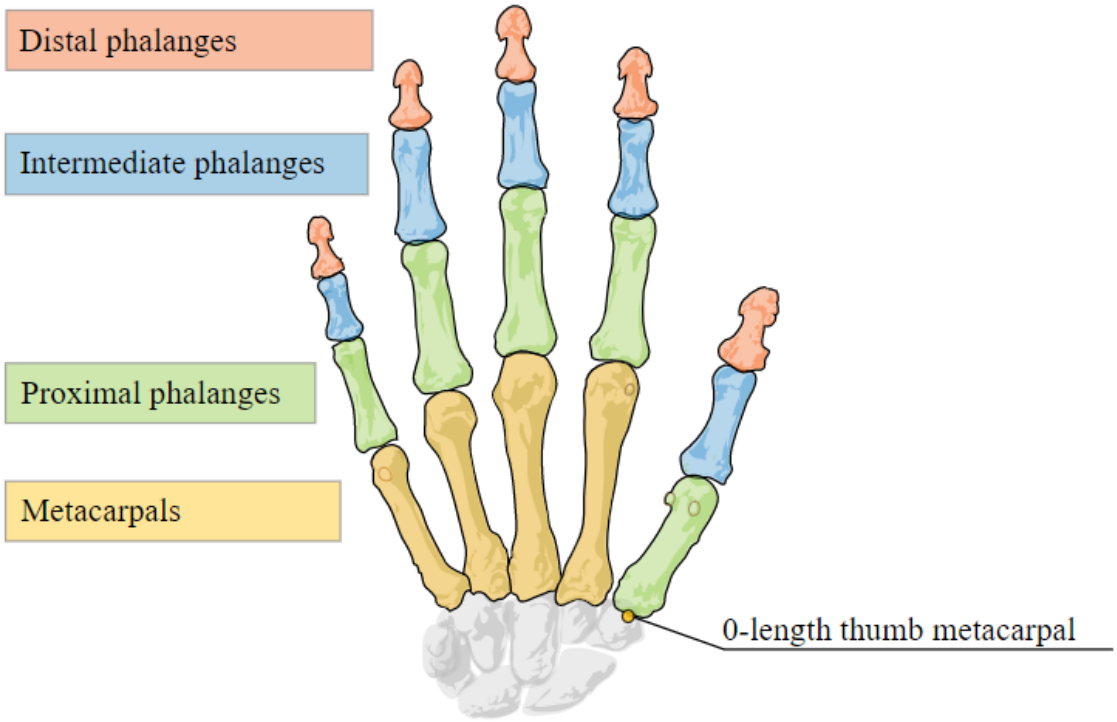
Leap Motion Controller pruža informacije i o svakom prstu na ruci. Ako se ne vidi cijeli prst ili njegov dio, karakteristike prsta procjenjuju se na temelju prošlih opažanja i anatomskog modela šake. Prsti se identificiraju prema nazivu. Prste predstavlja klasa *Finger*.

Slika 22. Model prstiju i vektor smjera za svaki prst

Finger klasa također pruža i *Bone* klasu koja opisuje položaj i orijentaciju svake kosti prsta. Svi prsti sadrže četiri kosti poredane od baze do vrha.

Kosti se identificiraju kao (Slika 23.):

- metakarpalna kost – kost unutar šake koja povezuje prst sa zapešćem (osim palca)
- proksimalna falanga – kost povezana s dlanom
- srednja falanga – srednja kost prsta, između vrha i baze
- distalna falanga – završna kost na kraju prsta.



Slika 23. Model kostiju Leap Motion softvera

Važno je napomenuti da model palca ne odgovara u potpunosti standardnoj anatomskoj nomenklaturi. Pravi palac ima jednu kost manje od ostalih prstiju. Međutim, radi jednostavnijeg programiranja, model palca Leap Motion Controllera uključuje metakarpalnu kost nulte duljine tako da palac ima isti broj kostiju na istim indeksima kao i ostali prsti. Kao rezultat toga, anatomska metakarpalna kost palca označena je kao proksimalna falanga, a anatomska proksimalna falanga označena je kao srednja falanga u modelu kosti prstiju softvera Leap Motion.

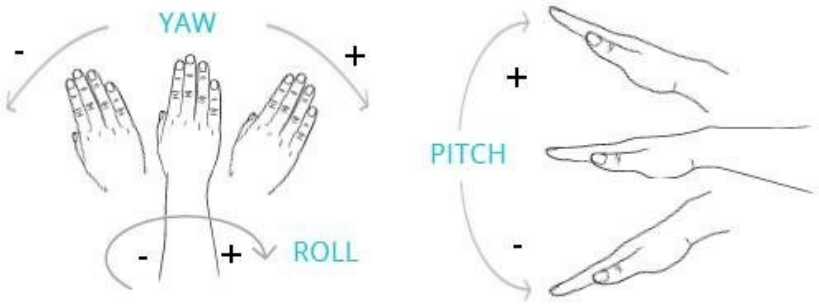
U studiji iz 2013. godine pokazalo se da je ukupna prosječna točnost uređaja 0,7 mm [36] što znači da bi uređaj trebao biti dovoljno dobar za korisnu uporabu.

5. POKUSI

U ovome poglavlju razvit će se programi virtualne stvarnosti za kontroler Leap Motion te eksperimentalno pokazati razne mogućnosti takve upotrebe.

Za sve pokuse koristit će se programski jezik Python uz programske pakete kao što su: matplotlib za crtanje grafova, numpy za lakšu obradu podataka te Leap Motion SDK. Kao operativni sustav koristit će se Microsoft Windows 10. Vrijeme izvođenja svakog pokusa odabrat će se proizvoljno i ne mora biti isto za sve pokuse jer ne utječe na rezultat.

Kako bi se što lakše upravljalo softverom, za većinu pokusa ruka će biti nepomična u prostoru dok će se kutevi nagiba šake oko odabranih osi koristiti kao ulazne naredbe za pojedini pokus. Slika 24. prikazuje takav način upravljanja.



Slika 24. Prikaz ljudske ruke u odnosu na različite nagibe šake (*pitch, roll, yaw*) [37]

Spomenuti nazivi za nagibe sa slike: engl. *pitch, roll, yaw* – posrtanje, valjanje, skretanje – standardni su i česti nazivi u zrakoplovstvu i robotici a za potrebe narednih ispitivanja matematički se definiraju kao:

$$pitch = \tan^{-1}\frac{Z}{Y} \tag{1}$$

$$roll = \tan^{-1}\frac{X}{Y} \tag{2}$$

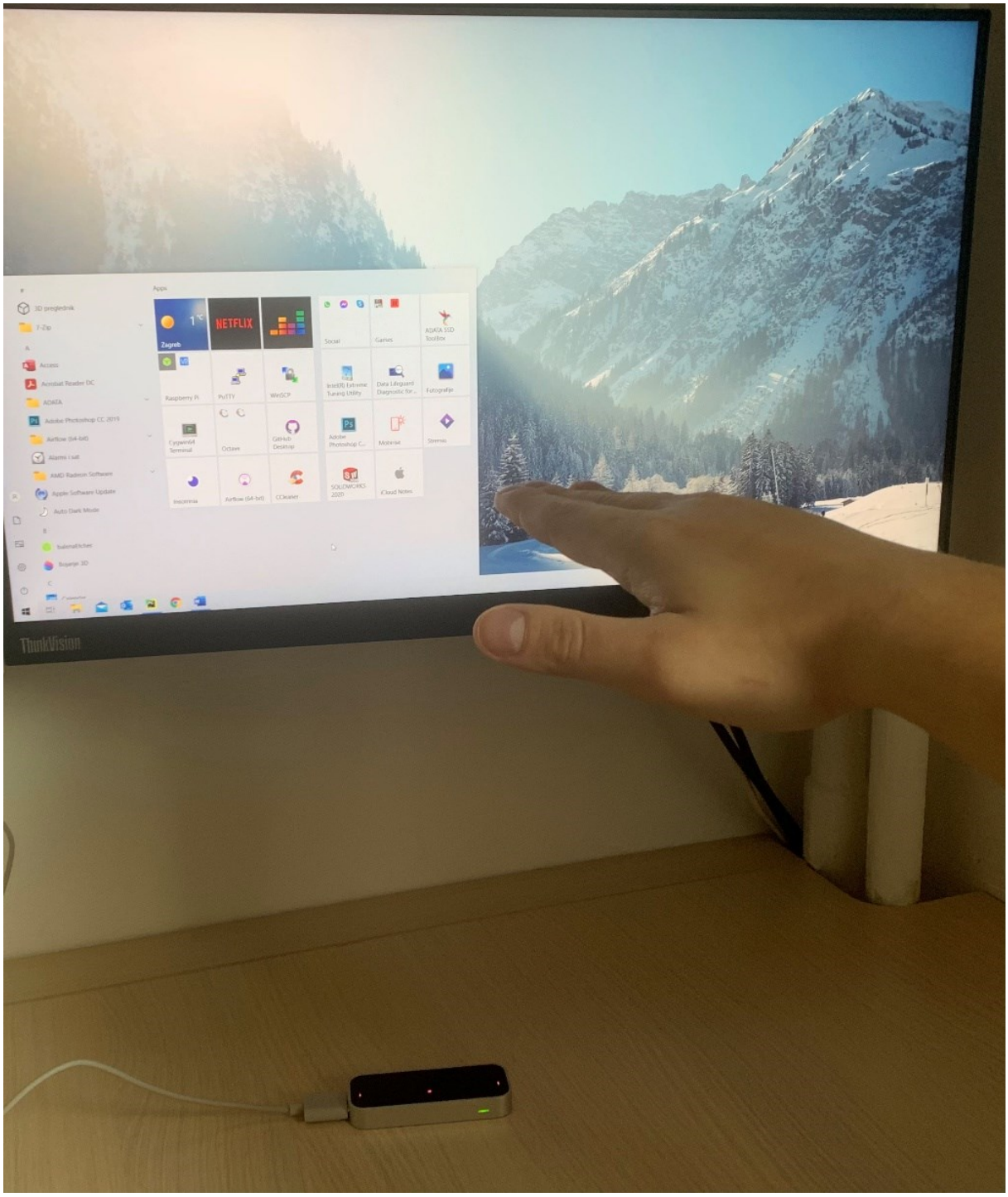
$$yaw = \tan^{-1}\frac{X}{Z} \quad (3)$$

gdje su *X*, *Y* i *Z* iznosi komponenata vektora normale na dlanu (Slika 24.).

5.1. KONTROLA KURSORA OPERATIVNOG SUSTAVA

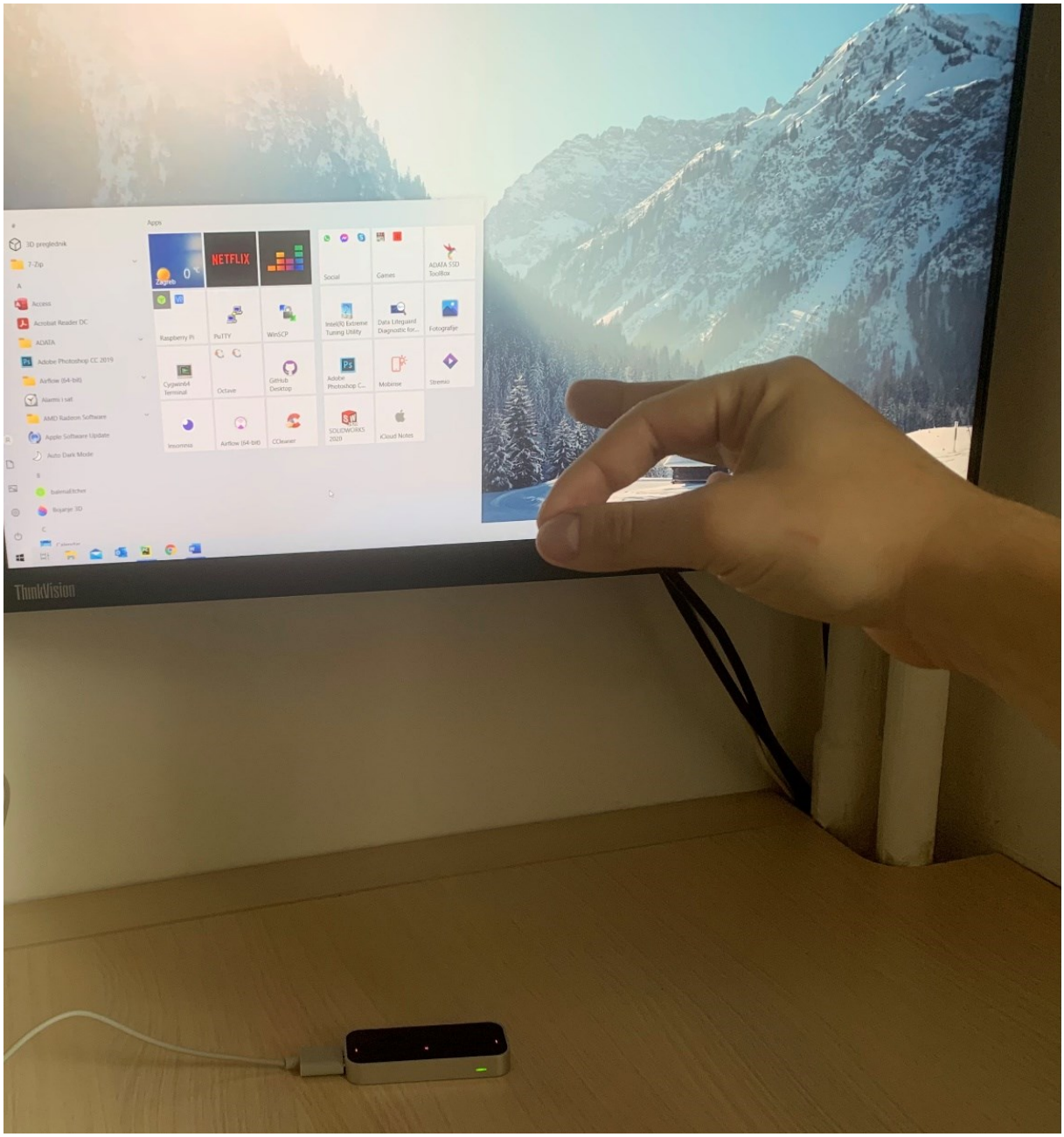
U ovome pokusu ispitala se preciznost i točnost korištenja kontrolera Leap Motion u svrhu upravljanja kursorom operativnog sustava Microsoft Windows 10. Također se ispitalo i korištenje gesti. Cilj pokusa je dobiti uvid u praktičnost takve uporabe. Naime, praktičnost se ogleda u mogućnosti lakšeg upravljanja softverom nego kod klasične metode uporabom tipkovnice i miša.

Program *kontrola_kursora.py* izračunava potrebne nagibe iz podataka Leap Motion kontrolera u svakom trenutku uzorkovanja. Ti nagibi se također bilježe u pozadini za crtanje grafova jednom kad korisnik prekine radnju. Naredbama kursora pristupa se preko naredbi Windows API-ja. Slika 25. prikazuje takvu uporabu.



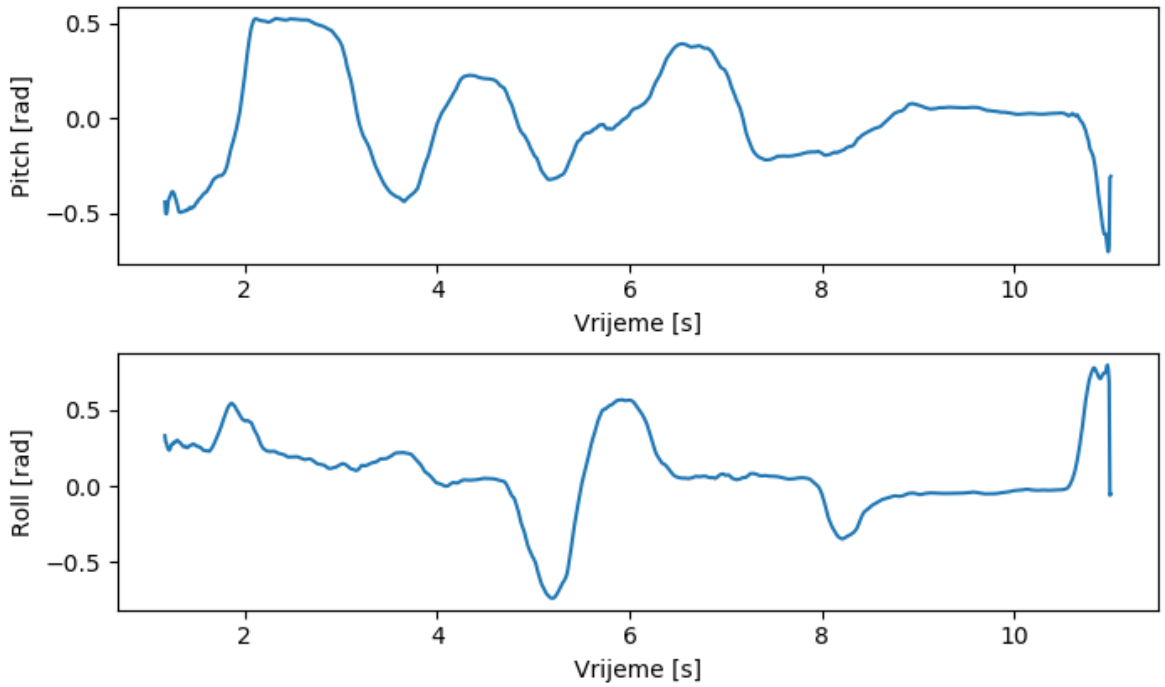
Slika 25. Prikaz kontrole kursora pomoću dlana

Pomoću Leap Motion API-ja moguće je također i prepoznavanje određenih gesti kojima se mogu pridružiti bilo koje naredbe. Slika 26. prikazuje gestu spajanja kažiprsta i palca kojoj je pridružena naredba klika.



Slika 26. Prikaz klika kursora pomoću geste

Slika 27. prikazuje grafove nagiba šake (slike 25. i 26.) u vremenu koje program generira nakon završetka korištenja. Iz grafova je vidljivo kako nagibi mogu poprimiti negativne ili pozitivne vrijednosti, ovisno o smjeru u kojem se želi ostvariti kretanje kursora. Također je na grafovima vidljivo da podaci ne sadrže previše šuma niti diskontinuitete, što znači da se s lakoćom mogu dalje analizirati kako bi se poboljšao neki proces.



Slika 27. Grafovi nagiba *pitch* i *roll*

Korištenje miša jednostavno je i intuitivno, no ipak nije toliko precizno kao u klasičnog računalnog miša. Problem je također i u ergonomiji jer brzo postane teško držati ruku podignutu dulje vremena. Ovakav način korištenja mogao bi biti pogodan kada se želi postići higijensko korištenje računala i/ili strojeva u medicinske svrhe.

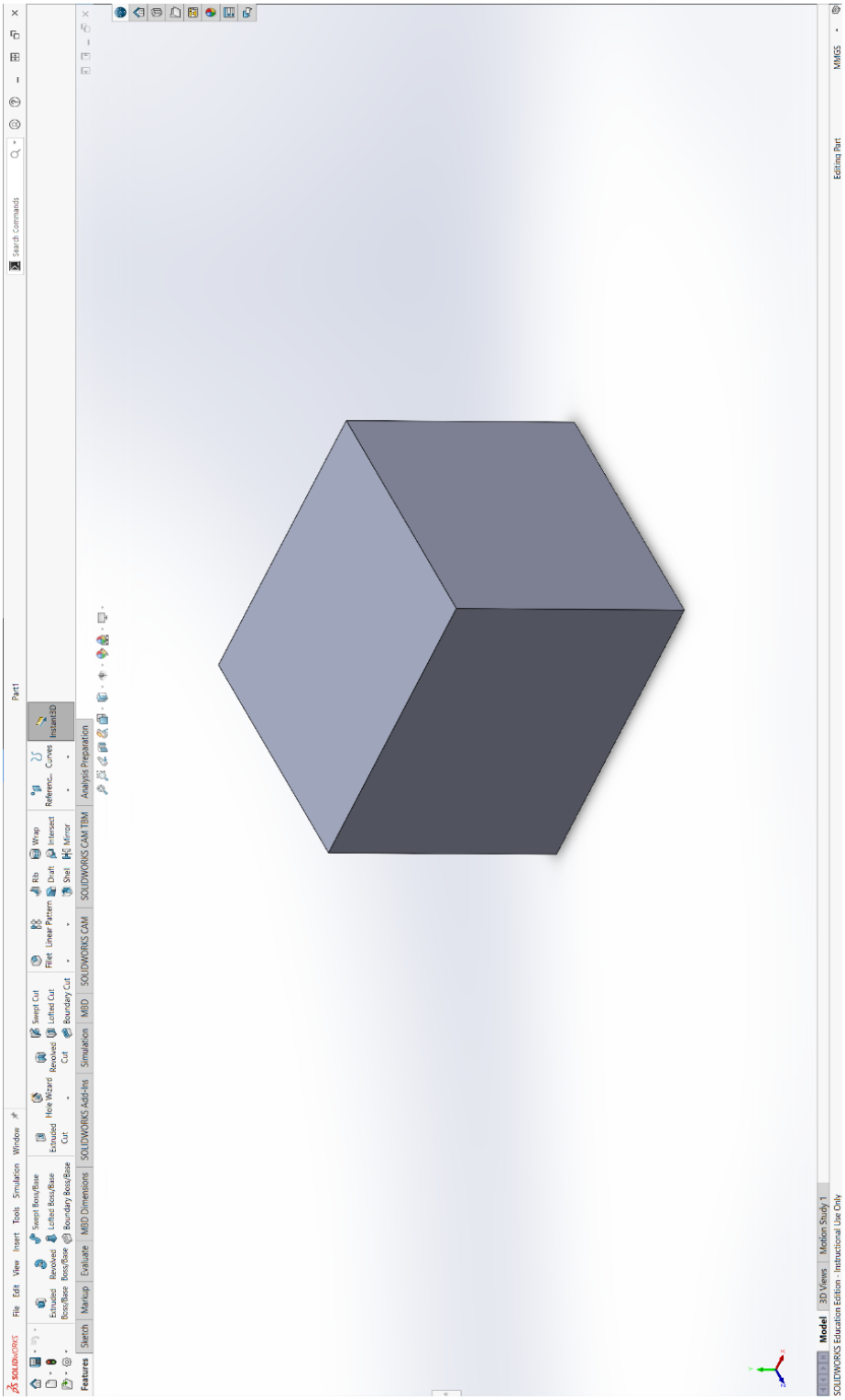
5.2. INTERAKCIJA S CAD SOFTVEROM

U ovome pokusu nastojala se istražiti mogućnost uporabe kontrolera Leap Motion za lakši pregled 3D modela unutar SolidWorks CAD softvera. Također se nastojala istražiti i mogućnost upotrebe više ruku odjednom.

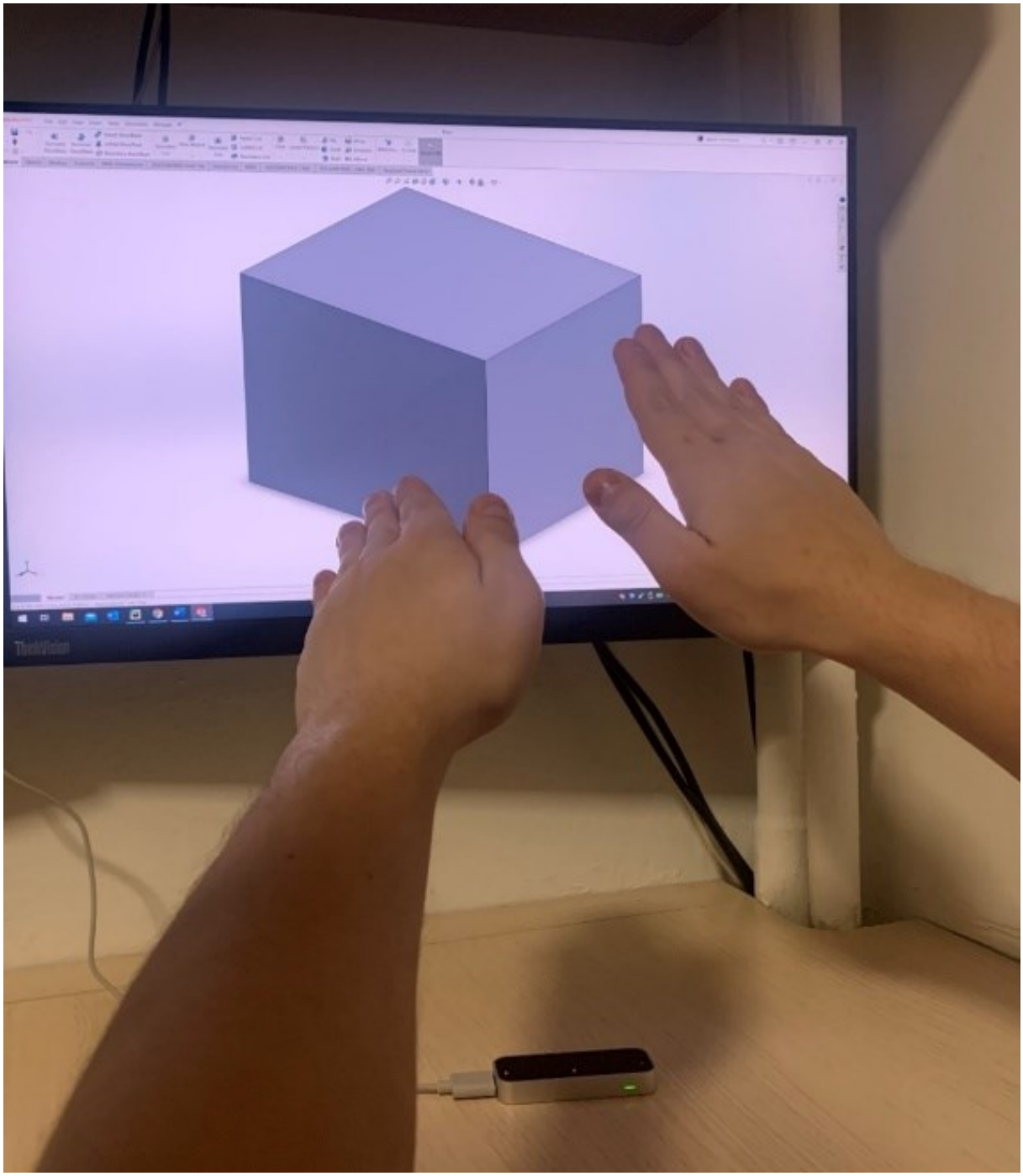
SolidWorks podržava programiranje pomoću programskog jezika VBA (*Visual Basic for Applications*) koji je dostupan unutar svakog Microsoft Windows operativnog sustava. Tako je pozivanjem VBA naredbi za SolidWorks iz Python programskog jezika moguće direktno upravljanje 3D modelima. Dodatnim povezivanjem s Leap Motion SDK paketom moguće je upravljanje 3D modelom koristeći pokrete ruku.

Za potrebe ovog eksperimenta napravljen je jednostavan 3D model kvadra proizvoljnih dimenzija (Slika 28.).

Program *solidworks_pomicanje_i_zumiranje.py* koristi prethodno spomenute nagibe (*pitch i roll*) svakog dlana zasebno kako bi se 3D model kvadra mogao istovremeno pomicati i zumirati. Također je moguće i korištenje samo jedne (desne) ruke za pomicanje 3D modela.



Slika 28. 3D model kvadra koji je korišten u pokusu



Slika 29. Prikaz kontrole modela pomoću obje ruke istovremeno

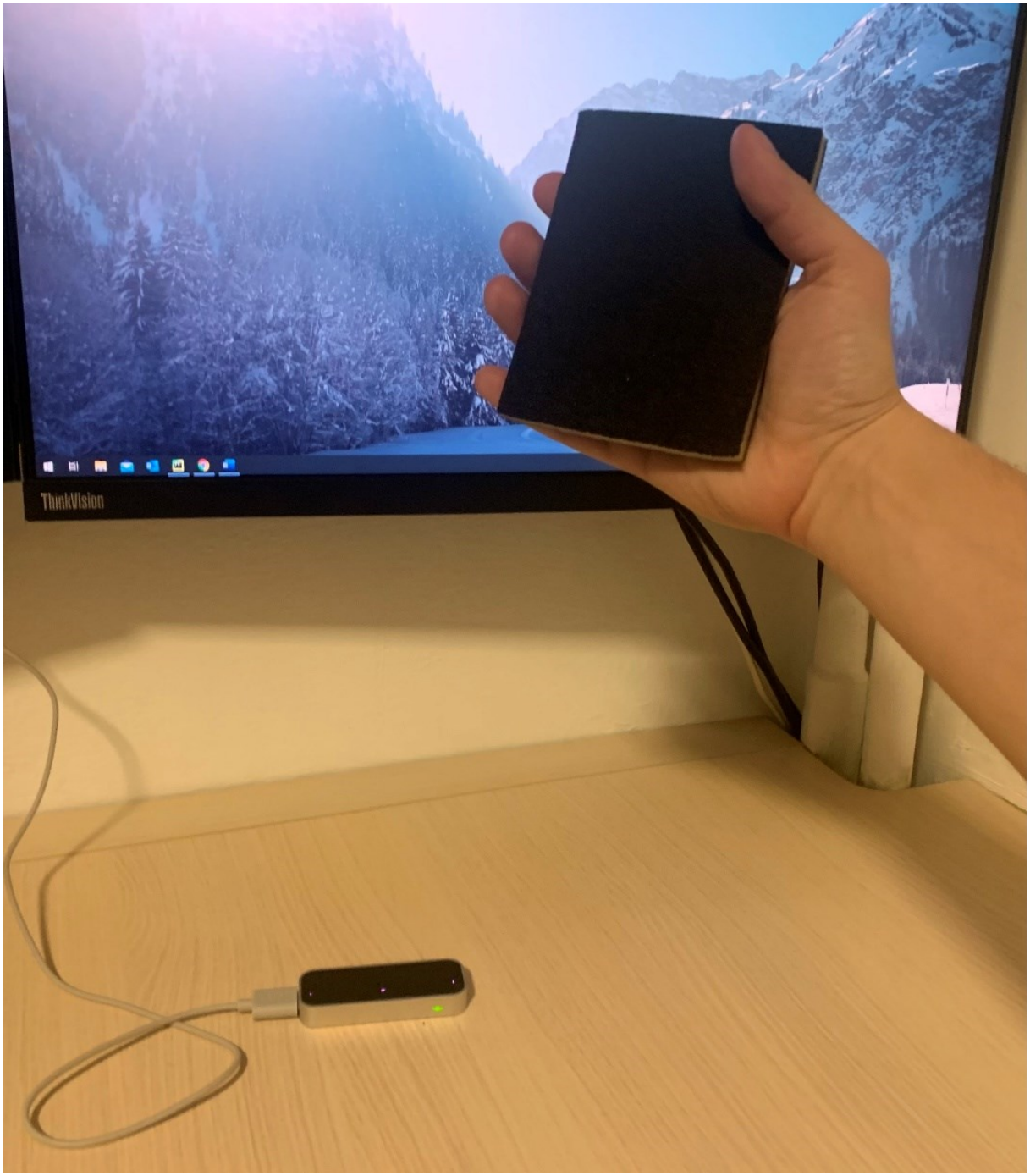
Ovakav način kontrole izrazito je neergonomičan jer obje ruke korisnika moraju biti podignute. Upravljanje je također vrlo nespretno i neprecizno zbog više istovremenih motoričkih radnji. Prisutan je i problem prilagodbe vrijednosti nagiba za pojedinu CAD aplikaciju. Naime, svaka CAD aplikacija ima svoje interne vrijednosti kojima raspolaže za zumiranje ili pomicanje 3D modela pa je tako potrebna prilagodba da te vrijednosti budu u

skladu s vrijednostima kontrolera Leap Motion. U ovom slučaju to je izvedeno jednostavnom matematičkom relacijom, no to ne mora nužno uvijek biti tako.

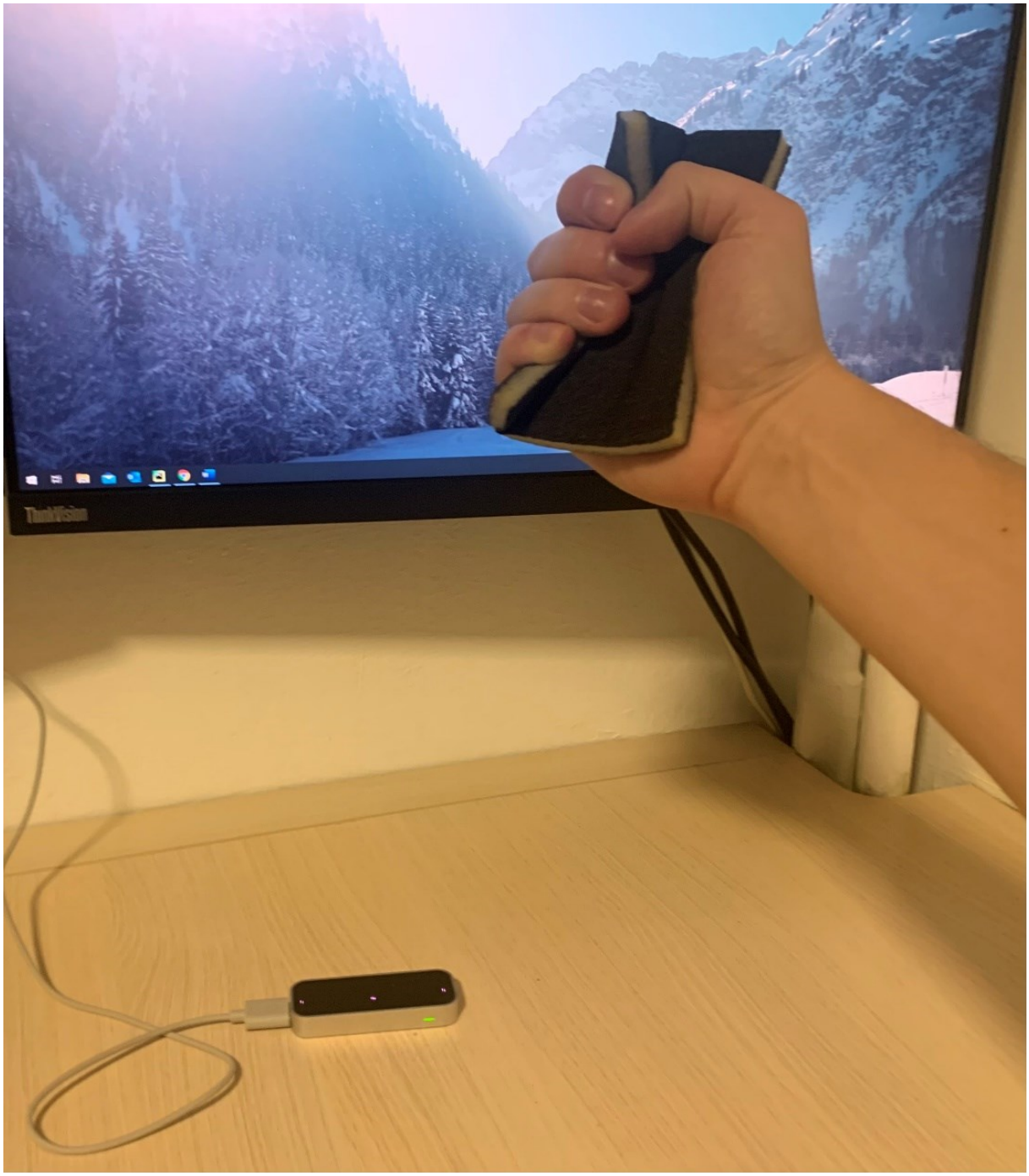
Ukoliko se koristi samo jedna (desna ruka), 3D model se može samo pomicati i upravljanje je u tom slučaju lakše. Međutim, kao i u prvom pokusu, i dalje nije ergonomično zbog brzog umaranja ruke. Zbog svega navedenog, ovaj način upravljanja nije preporučljiv.

5.3. RAD S DEFORMABILNIM TIJELOM

U ovome pokusu nastojala se istražiti mogućnost uporabe Leap Motion kontrolera za snimanje i analizu pokreta koji se javljaju u radu s deformabilnim tijelom. Deformabilno tijelo koje se koristi za ove potrebe je spužva. Slika 30. prikazuje stanje tijela prije deformiranja dok slika 31. prikazuje stanje tijela nakon deformiranja.

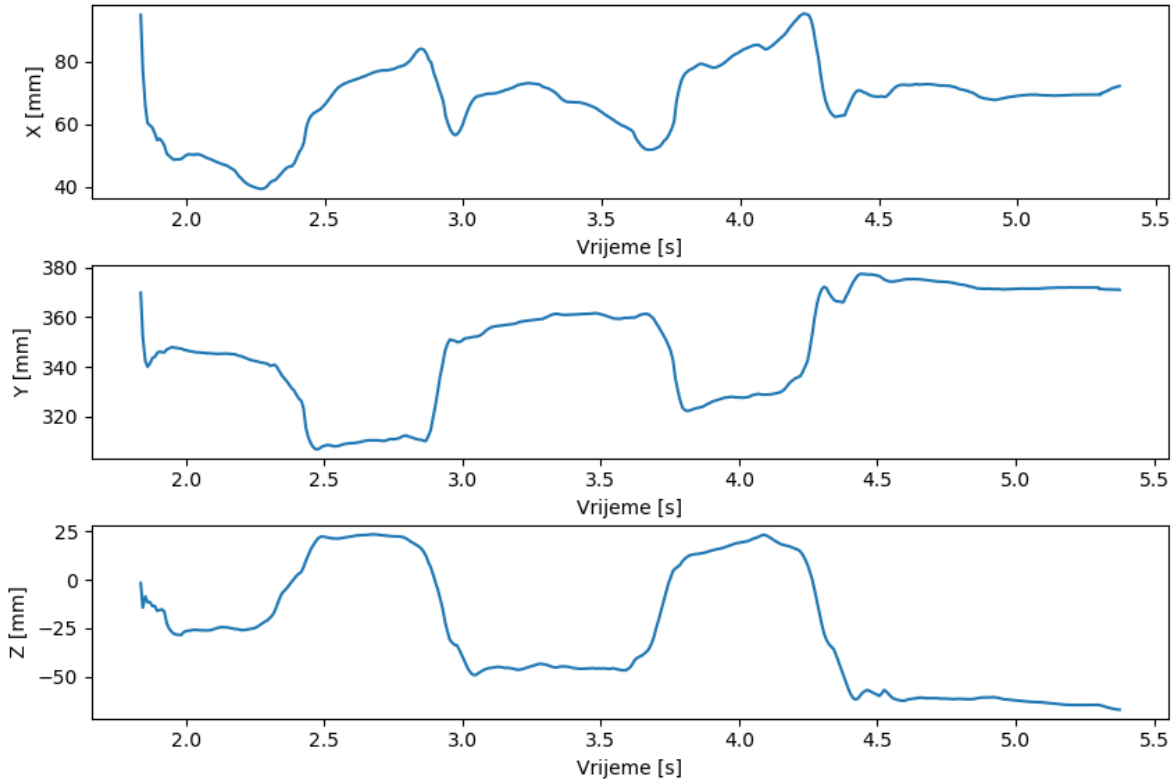


Slika 30. Početna pozicija mjerenja: tijelo u nedeformiranom stanju



Slika 31. Krajnja pozicija mjerenja: spužva u deformiranom stanju

Za vrijeme pokusa, program *snimanje_parametara_deformabilni_objekti.py* bilježi pozicije vrha kažiprsta po svim osima u pozadini. Kad korisnik prekine radnju, dobivaju se grafovi pozicija po svim osima u vremenu. Grafovi se mogu vidjeti na slici 32.



Slika 32. Pozicija vrha kažiprsta u vremenu po svim osima kontrolera Leap Motion

Slično kao i kod nagiba dlana, pozicije po svim osima mogu poprimiti pozitivne i negativne vrijednosti, ovisno o smjeru kretanja od ishodišta Leap Motion kontrolera.

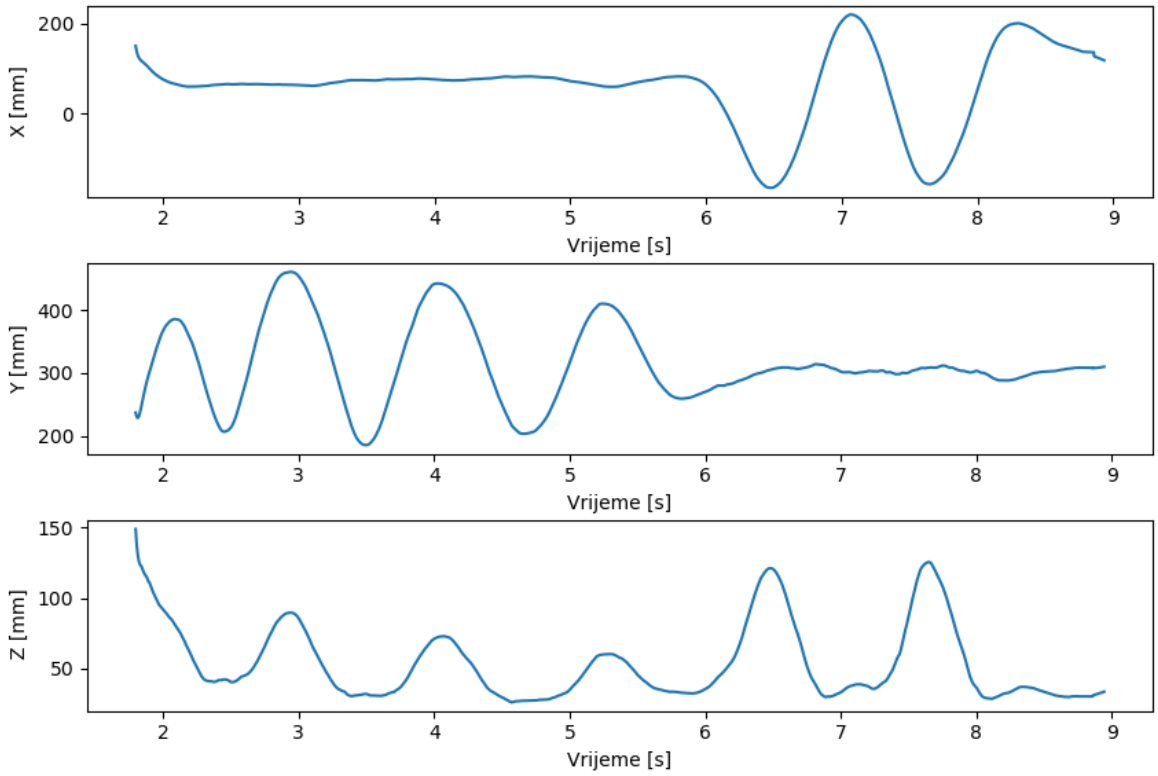
Po osima *Y* i *Z* jasno se vide periodi deformiranog stanja. Prvi period nalazi se između 2,2 i 3 s dok se drugi period nalazi između 3,5 i 4,5 s. Iz navedenog korisno je promatrati zbivanja po osi *Z* jer je spužva najviše deformirana u tom smjeru. U prvom periodu jasno se vidi da ta deformacija iznosi otprilike 50 mm dok u drugom periodu ona iznosi otprilike 75 mm. Iz navedemih podataka mogu se dobiti korisne informacije poput ukupnog izvršenog rada, separacije vremena trajanja deformacije od vremena mirovanja i slično.

Ovakva uporaba Leap Motion kontrolera ima mnogo potencijala, posebice u proizvodnoj industriji za lako i neinvazivno mjerenje i analizu učinkovitosti rada nad deformabilnim proizvodima. Ipak, ovakav način korištenja nije izrazito točan zbog osjetljivosti uređaja, stoga je preporučljiv samo za aproksimativne izračune.

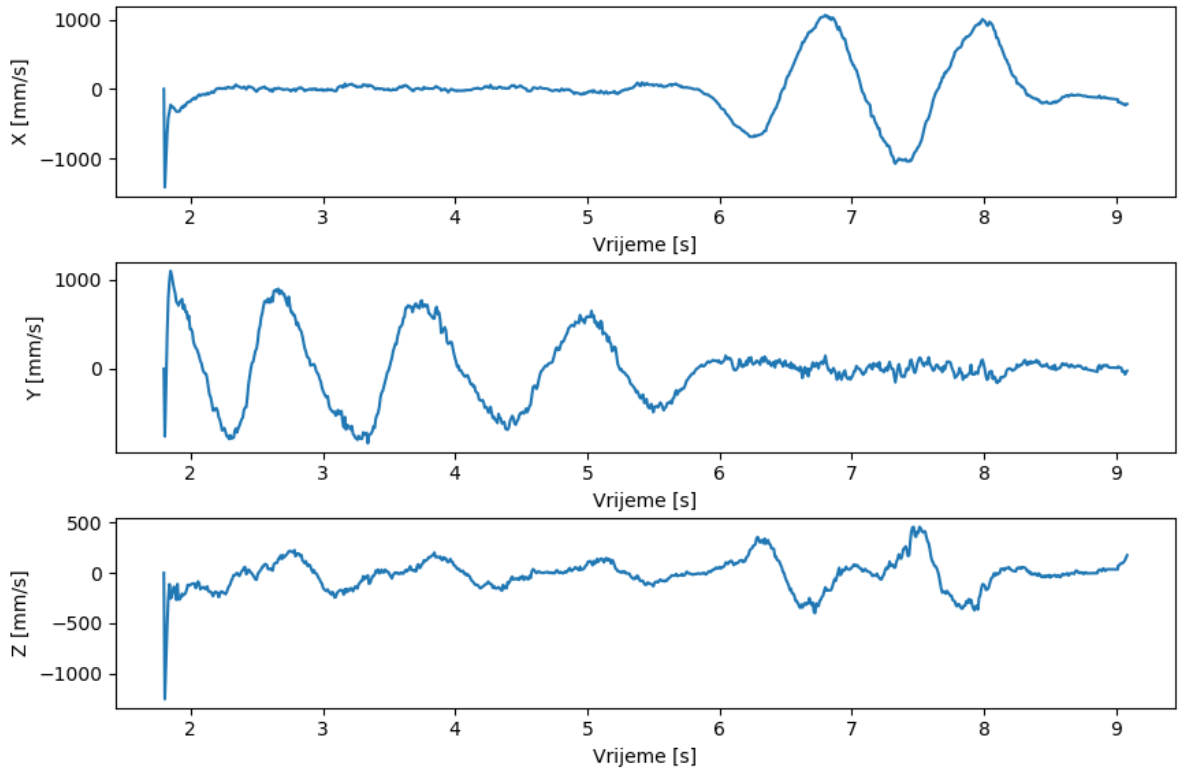
5.4. PRAĆENJE I ANALIZA OPĆIH PUTANJA RUKU

Kao što je do sada pokazano, pomoću Leap Motion kontrolera mogu se dobiti razni podaci o pokretima ruku. Osim specifičnih namjena tako je moguće i praćenje općih putanja ruku.

Program *snimanje_parametara_ruku.py* bilježi pozicije i brzine dlana jedne ruke po svim osima u pozadini. Kad korisnik prekine radnju, dobivaju se grafovi pozicija i brzina po svim osima u vremenu. Grafovi se mogu vidjeti na slikama 33. i 34. Putanja kretanja je izvedena proizvoljno.



Slika 33. Pozicija dlana ruke u vremenu po svim osima kontrolera Leap Motion



Slika 34. Brzina dlana ruke u vremenu po svim osima kontrolera Leap Motion

Osim pozicija, u ovome pokusu pokazano je da se mogu dobiti i brzine kretanja dlana ruke kao dodatna informacija za daljnju analizu.

Iz grafova se jasno mogu vidjeti periodi u kojima se ruka podiže, spušta ili miruje. Iako su grafovi brzina po svim osima šumovitiji nego grafovi pozicija, daljnjim filtriranjem brzina mogu se dobiti korisne nove informacije.

Daljnjom analizom brzina tako bi se mogle dobiti informacije poput prijeđenog puta, ubrzanja, količine gibanja (ukoliko je poznata i masa ruke) i slično. Navedeno također može biti korisno u proizvodnji kako bi se procesi lakše snimili, analizirali i poboljšali.

5.5. ZAKLJUČAK POKUSA

Izvedeni pokusi jasno pokazuju kako je moguće dobiti mnoštvo podataka o pokretima ruku pomoću kontrolera Leap Motion. Dobivene podatke vrlo je lako dalje upotrijebiti za razne svrhe poput kontrole kursora ili upravljanja CAD aplikacijom. Također je moguće lako spremanje tih podataka te njihova analiza i prikaz pomoću bilo kojeg softvera. Međutim, doživljaj upravljanja kursora ili CAD aplikacije može biti neergonomičan za korisnika te neprecizan u odnosu na klasične metode upravljanja. Ipak, takva upotreba je preporučljiva samo kada se želi izbjeći dodirivanje sučelja računala i/ili strojeva.

Prilikom snimanja pokreta pri radu s deformabilnim tijelima pokazano je da uređaj ima mnogo potencijala za upotrebu u industriji (ali i svugdje drugdje) gdje može biti koristan za **analizu rada nad deformabilnim proizvodima**. Pokusi su pokazali da se na taj način mogu dobiti korisne informacije poput izvršenog rada, separacije vremena trajanja deformacije od vremena mirovanja i slično. Pokazano je i da takva mjerenja sadrže značajna odstupanja zbog osjetljivosti uređaja, stoga je preporučljivo koristiti ih samo za aproksimativne izračune.

Nadalje, Leap Motion kontroler može također biti koristan i za snimanje i analizu općih putanja ruku. Iz snimljenih podataka moguće je izračunati **mnoštvo varijabli** koje bi bile korisne **za poboljšanje nekog procesa**, **bilo da je riječ o industrijskom ili neindustrijskom procesu**.

6. ZAKLJUČAK

Automatizacija fizičkog ljudskog rada oduvijek je bila ključna u postizanju što veće produktivnosti i profita. Esencijalnu ulogu u svakom fizičkom radu ima pokret, stoga je snimanje i analiza pokreta vrlo bitno u ostvarivanju i standardizaciji što efikasnijih industrijskih procesa. Iz svakog pokreta mogu se dobiti vrlo korisne informacije poput sile, rada, snage i slično. Te informacije mogu se analizirati te se temeljem njih mogu donijeti korisne promjene u nekom procesu.

Moderne tehnologije i digitalni alati omogućuju snimanje i analizu pokreta mnogo lakšim nego ikada prije. Virtualna stvarnost nova je tehnologija koja je sve prisutnija u snimanju i analizi pokreta. Pokazala se vrlo korisnom u raznim industrijama, uz jasan trend rasta zastupljenosti te veliki potencijal za poboljšanje mnogih procesa.

Kontroler Leap Motion je malen, brz i precizan uređaj koji služi za snimanje i analizu pokreta ruku i prstiju korisnika te njihovu upotrebu u virtualnoj stvarnosti. Zbog malih dimenzija i niske cijene može koristiti u robotici, industrijskom dizajnu, zdravstvu i slično.

Kontroler Leap Motion dolazi s naprednim istoimenim softverom koji može brzo obraditi velike količine podataka. Napredni algoritmi i tehnike filtriranja pružaju detaljne informacije o pokretima ruku i prstiju korisnika u svakom trenutku uz nesmetanu vremensku usklađenost.

Latencija, koja je bitna za dobro korisničko iskustvo interaktivnih sučelja virtualne stvarnosti, dovoljno je niska za praktičnu porabu. Iako mnogo faktora koji doprinose latenciji ne ovisi o samom uređaju, kontroler Leap Motion ima manju latenciju nego ostali slični proizvodi na tržištu. Poznato je da latencija, koja bi trebala biti manja od 30 ms kako ju ljudski vidni i živčani sustav ne bi mogao primijetiti, može biti postignuta uz određene postavke poput korištenja USB 3.0 sučelja.

Programska podrška kontrolera Leap Motion nudi sve alate i resurse potrebne za razvoj vlastitih aplikacija. Jasna tehnička dokumentacija i velika zajednica od preko 350 000 programera diljem svijeta čine razvoj VR aplikacija lakšim nego prije. Kontroler Leap Motion također sadrži SDK i API koji služe za pristup sirovim podacima uređaja u svakom trenutku uz podršku za tri najzastupljenija operativna sustava i mnoštvo programskih jezika.

U radu su korištenjem kontrolera Leap Motion provedeni pokusi za razvoj jednostavnijih aplikacija virtualne stvarnosti koje su imale za cilj istražiti praktičnost kontrole softvera pokretima ruku. Također je provedeno snimanje, prikaz i analiza različitih parametara navedenih pokreta, korištenje gesti, dvoruki rad, rad s deformabilnim tijelom te snimanje i analiza opće putanje pokreta ruke. Pokazano je da je upravljanje softverom preporučljivo samo u pojedinim industrijama, poput proizvodnje ili medicine gdje je beskontaktno upravljanje uređajima poželjno zbog zahtjeva sterilnosti i higijene. Za sve ostale potrebe klasične metode poput računalnog miša još uvijek imaju više smisla.

Snimanje i analiza pokreta općih putanja pokreta ruku ili rad s deformabilnim tijelima pokazali su se kao vrlo zanimljivo područje uporabe zbog mnogo podataka koji se mogu dobiti te analizirati kako bi se poboljšao neki proces. To je zanimljivo s inženjerskog stanovišta jer se tako mogu dobiti korisne informacije poput izvršenog rada, vremena trajanja deformacije, prijeđenog puta, ubrzanja i slično. Ipak, pokazano je i da takva mjerenja sadrže značajna odstupanja zbog osjetljivosti uređaja, stoga je preporučljivo koristiti ih samo za aproksimativne izračune.

Za daljnji rad od posebnog je interesa provedba složenijih pokusa i analiza prikupljenih podataka za neke specifične kontekste (primjerice ORL kirurški zahvati), čime bi se stekle osnove za normiranje, mehanizaciju i automatizaciju procesa te razvoj novih alata.

7. LITERATURA

[1] https://data.worldbank.org/indicator/SL.AGR.EMPL.ZS?end=2020&start=1991 Pristupljeno: 2020-12-01.

[2] https://en.wikipedia.org/wiki/Assembly_line Pristupljeno: 2020-12-01.

[3] https://en.wikipedia.org/wiki/Fourth_Industrial_Revolution Pristupljeno: 2020-12-02.

[4] https://www.bmbf.de/de/zukunftsprojekt-industrie-4-0-848.html Pristupljeno: 2021-01-23.

[5] https://www.sanayidegelecek.com/en/sanayi-4-0/tarihsel-gelisim Pristupljeno: 2020-12-05.

[6] https://www.salesforce.com/blog/what-is-the-fourth-industrial-revolution-4ir/ Pristupljeno: 2020-12-02.

[7] Šare N., Snimanje i analiza pokreta u virtualnoj stvarnosti [Diplomski rad]. Zagreb: Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje; 2020. Dostupno na: https://urn.nsk.hr/urn:nbn:hr:235:270575

[8] https://en.wikipedia.org/wiki/Motion Pristupljeno: 2020-12-05.

[9] https://www.youtube.com/watch?v=BJWPuiqznhI Pristupljeno: 2020-12-06.

[10] https://en.wikipedia.org/wiki/File:F._Taylor_1856-1915.jpg Pristupljeno: 2020-12-06.

[11] https://www.theguardian.com/commentisfree/2017/sep/25/robots-taken-over-work-jobs-economy Pristupljeno: 2020-12-07.

[12] https://www.roadtovr.com/global-virtual-reality-industry-reach-7-2-billion-revenues-2017/ Pristupljeno: 2020-12-07.

[13] https://blog.techviz.net/everything-to-know-about-vr-in-engineering- Pristupljeno: 2020-12-07.

[14] Seu; et al. (2018). "Use of gaming and affordable VR technology for the visualization of complex flow fields". *Proceedings of the 6th European Conference on Computational Mechanics (ECCM 6) & 7th European Conference on Computational Fluid Dynamics (ECFD 7), Glasgow, Scotland*.

[15] Abulrub, Abdul-Hadi G.; Attridge, Alex N.; Williams, Mark A. (April 2011). "Virtual reality in engineering education: The future of creative learning". *2011 IEEE Global Engineering Education Conference (EDUCON)*: 751–757. doi:10.1109/EDUCON.2011.5773223. ISBN 978-1-61284-642-2

[16] https://en.wikipedia.org/wiki/Virtual_reality Pristupljeno:: 2020-12-09.

[17] https://blog.leapmotion.com/vr-universities/ Pristupljeno: 2020-12-10.

[18] https://ww1.prweb.com/prfiles/2013/07/30/10961515/LeapMotion-LaptopFullsize.jpg Pristupljeno: 2020-12-10.

[19] Buzjak D., Prema imerzivnom projektiranju proizvodnih procesa korištenjem tehnika virtualne stvarnosti [Diplomski rad]. Zagreb: Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje; 2020. Dostupno na: https://urn.nsk.hr/urn:nbn:hr:235:990908

[20] Strsoglavec, Filip i Kunica, Zoran (2011) Unapređenje projektiranja i sklapanja plinske mjerno-redukcijske stanice. PLIN2011: 9. skup o prirodnom plinu, toplini i vodi; 2. međunarodni skup o prirodnom plinu, toplini i vodi, 28-30.09.2011., Osijek, Hrvatska.

[21] https://en.wikipedia.org/wiki/Motion_capture Pristupljeno: 2020-12-10.

[22]https://upload.wikimedia.org/wikipedia/commons/9/97/Two_repetitions_of_a_walking_sequence_of_an_individual_recorded_using_a_motion-capture_system.gif Pristupljeno: 2020-12-11.

[23] https://en.wikipedia.org/wiki/VR_positional_tracking Pristupljeno: 13.2020-12-

[24]https://en.wikipedia.org/wiki/VR_positional_tracking#/media/File:Positional_tracking_in_virtual_reality.png Pristupljeno: 2020-12-13.

[25] http://web.mit.edu/comm-forum/legacy/papers/furniss.html Pristupljeno: 2020-12-13.

[26] https://blog.leapmotion.com/wp-content/uploads/2013/04/device-hero.png Pristupljeno: 2020-12-13.

[27] https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf Pristupljeno: 2020-12-15.

[28] https://cms.ultraleap.com/app/uploads/2020/02/HERO-UltraLeap_Product05342_edit.jpg Pristupljeno: 2020-12-15.

[29]http://media.tumblr.com/54ddd23606f6d449f6660329787f48c8/tumblr_inline_mm8stt1uiO1qz4rgp.gif Pristupljeno: 2020-12-15.

[30] https://www.ultraleap.com/datasheets/Leap_Motion_Controller_Datasheet.pdf Pristupljeno: 2020-12-18.

[31] https://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/ Pristupljeno: 2020-12-18.

[32] https://blog.leapmotion.com/understanding-latency-part-1/ Pristupljeno: 2020-12-23.

[33] https://blog.leapmotion.com/understanding-latency-part-2/ Pristupljeno: 2020-12-23.

[34]https://developerarchive.leapmotion.com/documentation/python/devguide/Leap_Overview.html Pristupljeno: 2020-12-28.

[35] https://developer.leapmotion.com/documentation Pristupljeno: 2021-01-28.

[36] Weichert, Frank; Bachmann, Daniel; Rudak, Bartholomäus; Fisseler, Denis (2013-05-14). "Analysis of the Accuracy and Robustness of the Leap Motion Controller". *Sensors (Basel, Switzerland)*. **13** (5):6380–6393. doi:10.3390/s130506380. ISSN 1424-8220. PMC 3690061. PMID 23673678

[37]https://www.researchgate.net/profile/Martin_Molina4/publication/304816895/figure/fig11/AS:667910355812385@1536253646890/Hand-gesture-commands-used-for-drone-flight-In-this-FFsetup-positive-pitch-roll-and-yaw.jpg Pristupljeno: 2021-01-28.

PRILOZI

Prilog 1. Program kontrola_kursora.py 48

Prilog 2. Program solidworks_pomicanje_i_zumiranje.py 50

Prilog 3. Program snimanje_parametara_deformabilni_objekti.py 51

Prilog 4. Program snimanje_parametara_ruku.py 53

Prilog 1. Program kontrola_kursora.py

```
import sys
import math
import Leap
import ctypes
import win32api
import win32con
import time
import matplotlib.pyplot as plt
import numpy as np
win32 = ctypes.windll.user32


class SnimacDlana(Leap.Listener):
    def __init__(self):
        time.clock()
        Leap.Listener.__init__(self)

        self.vremena = np.array([])

        self.pitchevi = np.array([])
        self.rollovi = np.array([])

        self.x = 0
        self.y = 0

        self.x_max = win32.GetSystemMetrics(0) - 1
        self.y_max = win32.GetSystemMetrics(1) - 1

        self.granicna_jacina_stiska = 0.98
        self.granicni_nagib_za_scroll_gore = 0.175
        self.granicni_nagib_za_scroll_dolje = -0.2

    def on_connect(self, controller):
        print "Spojen"

    def on_frame(self, controller):

        frame = controller.frame()

        if not frame.hands.is_empty:

            roll =
math.atan(frame.hands[0].palm_normal.x/frame.hands[0].palm_normal.y)
            pitch =
math.atan(frame.hands[0].palm_normal.z/frame.hands[0].palm_normal.y) * -1

            self.vremena = np.append(self.vremena, time.clock())
            self.pitchevi = np.append(self.pitchevi, pitch)
            self.rollovi = np.append(self.rollovi, roll)

            x_pomak = 8 * roll
            y_pomak = 8 * pitch

            self.x = self.x + x_pomak
```

```
            self.y = self.y + y_pomak

            if self.x > self.x_max:
                self.x = self.x_max
            elif self.x < 0.0:
                self.x = 0.0

            if self.y > self.y_max:
                self.y = self.y_max
            elif self.y < 0.0:
                self.y = 0.0

            win32.SetCursorPos(int(self.x), int(self.y))

            if frame.hands[0].pinch_strength > self.granicna_jacina_stiska:

                win32api.mouse_event(win32con.MOUSEEVENTF_LEFTDOWN,
int(self.x), int(self.y), 0, 0)
                win32api.mouse_event(win32con.MOUSEEVENTF_LEFTUP,
int(self.x), int(self.y), 0, 0)

                time.sleep(0.2)


def main():

    listener = SnimacDlana()

    controller = Leap.Controller()
    controller.set_policy_flags(Leap.Controller.POLICY_BACKGROUND_FRAMES)
    controller.add_listener(listener)

    print "Pritisni Enter za izlaz..."
    sys.stdin.readline()

    fig, axs = plt.subplots(2)
    fig.set_size_inches(8, 5)

    axs[0].plot(listener.vremena, listener.pitchevi)
    axs[0].set_ylabel("Pitch [rad]")
    axs[0].set_xlabel("Vrijeme [s]")

    axs[1].plot(listener.vremena, listener.rollovi)
    axs[1].set_ylabel("Roll [rad]")
    axs[1].set_xlabel("Vrijeme [s]")

    plt.subplots_adjust(hspace=0.35)
    plt.show()

    controller.remove_listener(listener)


main()
```

Prilog 2. Program solidworks_pomicanje_i_zumiranje.py

```
import Leap
import math
from ctypes import windll
import win32com.client

win32 = windll.user32

solidworks = win32com.client.Dispatch("SldWorks.Application")
part = solidworks.ActiveDoc
model = part.ActiveView

controller = Leap.Controller()
controller.set_policy_flags(Leap.Controller.POLICY_BACKGROUND_FRAMES)

while True:
    frame = controller.frame()

    if not frame.hands.is_empty:

        pitch = -1.0 * math.atan(frame.hands.rightmost.palm_normal.z /
frame.hands.rightmost.palm_normal.y)

        if pitch < 0:

            pitch = abs(pitch) + 0.99 * (1 - abs(pitch))
            model.ZoomByFactor(pitch)

        elif pitch > 0:

            pitch = 1 + 0.1 * pitch
            model.ZoomByFactor(pitch)

        if frame.hands[1].palm_normal.x != 0:

            pitch2 = math.atan(frame.hands.leftmost.palm_normal.z /
frame.hands.leftmost.palm_normal.y) * -1
            roll = math.atan(frame.hands.leftmost.palm_normal.x /
frame.hands.leftmost.palm_normal.y)

            x_pomak = 0.008 * roll
            y_pomak = 0.008 * pitch2

            model.TranslateBy(x_pomak, y_pomak)
```

Prilog 3. Program snimanje_parametara_deformabilni_objekti.py

```
import sys
import Leap
import matplotlib.pyplot as plt
import time
import numpy as np


class SnimacDeformabilnihObjekata(Leap.Listener):
    def __init__(self):
        time.clock()
        Leap.Listener.__init__(self)

        self.vremena = np.array([])

        self.pozicije_x = np.array([])
        self.pozicije_y = np.array([])
        self.pozicije_z = np.array([])

    def on_connect(self, controller):
        print "Spojen"

    def on_frame(self, controller):

        frame = controller.frame()
        kaziprst = frame.fingers.finger_type(Leap.Finger.TYPE_INDEX)[0]
        pozicija_kaziprsta = kaziprst.stabilized_tip_position

        if not frame.hands.is_empty and len(frame.hands) < 2:

            self.vremena = np.append(self.vremena, time.clock())

            self.pozicije_x = np.append(self.pozicije_x,
pozicija_kaziprsta[0])
            self.pozicije_y = np.append(self.pozicije_y,
pozicija_kaziprsta[1])
            self.pozicije_z = np.append(self.pozicije_z,
pozicija_kaziprsta[2])


def main():
    listener = SnimacDeformabilnihObjekata()
    controller = Leap.Controller()
    controller.add_listener(listener)

    print "Pritisni Enter za izlaz..."
    sys.stdin.readline()

    fig, axs = plt.subplots(3)
    fig.set_size_inches(10, 7)

    axs[0].plot(listener.vremena, listener.pozicije_x)
    axs[0].set_ylabel("X [mm]")
    axs[0].set_xlabel("Vrijeme [s]")
```

```
    axs[1].plot(listener.vremena, listener.pozicije_y)
    axs[1].set_ylabel("Y [mm]")
    axs[1].set_xlabel("Vrijeme [s]")

    axs[2].plot(listener.vremena, listener.pozicije_z)
    axs[2].set_ylabel("Z [mm]")
    axs[2].set_xlabel("Vrijeme [s]")

    plt.subplots_adjust(hspace=0.35)
    plt.show()

    controller.remove_listener(listener)


main()
```

Prilog 4. Program snimanje_parametara_ruku.py

```
import sys
import Leap
import matplotlib.pyplot as plt
import time
import numpy as np


class SnimacRuku(Leap.Listener):
    def __init__(self):
        time.clock()
        Leap.Listener.__init__(self)

        self.vremena = np.array([])

        self.pozicije_x = np.array([])
        self.pozicije_y = np.array([])
        self.pozicije_z = np.array([])

        self.brzine_x = np.array([])
        self.brzine_y = np.array([])
        self.brzine_z = np.array([])

    def on_connect(self, controller):
        print "Spojen"

    def on_frame(self, controller):

        frame = controller.frame()

        if not frame.hands.is_empty and len(frame.hands) < 2:

            self.vremena = np.append(self.vremena, time.clock())

            self.pozicije_x = np.append(self.pozicije_x,
frame.hands[0].palm_position[0])
            self.pozicije_y = np.append(self.pozicije_y,
frame.hands[0].palm_position[1])
            self.pozicije_z = np.append(self.pozicije_z,
frame.hands[0].palm_position[2])

            self.brzine_x = np.append(self.brzine_x,
frame.hands[0].palm_velocity[0])
            self.brzine_y = np.append(self.brzine_y,
frame.hands[0].palm_velocity[1])
            self.brzine_z = np.append(self.brzine_z,
frame.hands[0].palm_velocity[2])


def main():
    listener = SnimacRuku()
    controller = Leap.Controller()
    controller.add_listener(listener)

    print "Pritisni Enter za izlaz..."
```

```
    sys.stdin.readline()

    fig, axs = plt.subplots(3)
    fig.set_size_inches(10, 7)

    axs[0].plot(listener.vremena, listener.pozicije_x)
    axs[0].set_ylabel("X [mm]")
    axs[0].set_xlabel("Vrijeme [s]")

    axs[1].plot(listener.vremena, listener.pozicije_y)
    axs[1].set_ylabel("Y [mm]")
    axs[1].set_xlabel("Vrijeme [s]")

    axs[2].plot(listener.vremena, listener.pozicije_z)
    axs[2].set_ylabel("Z [mm]")
    axs[2].set_xlabel("Vrijeme [s]")

    plt.subplots_adjust(hspace=0.35)

    fig2, axs2 = plt.subplots(3)
    fig2.set_size_inches(10, 7)

    axs2[0].plot(listener.vremena, listener.brzine_x)
    axs2[0].set_ylabel("X [mm/s]")
    axs2[0].set_xlabel("Vrijeme [s]")

    axs2[1].plot(listener.vremena, listener.brzine_y)
    axs2[1].set_ylabel("Y [mm/s]")
    axs2[1].set_xlabel("Vrijeme [s]")

    axs2[2].plot(listener.vremena, listener.brzine_z)
    axs2[2].set_ylabel("Z [mm/s]")
    axs2[2].set_xlabel("Vrijeme [s]")

    plt.subplots_adjust(hspace=0.35)
    plt.show()

    controller.remove_listener(listener)


main()
```