

UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET

Master rad

**Razvoj sistema za ispitivanje efekta haptičkog interfejsa na
učenje pokreta**

Mentor:

Prof dr. Dejan Popović

Student:

Petar Kovačević

Br. indeksa:2010/3061

Beograd, avgust 2011.

Development of a system for testing effects of haptic interface on motor learning

REZIME

Napravljen je sistem za testiranje efekata haptičkog interfejsa na učenje pokreta. On se sastoji iz računara, na koji je povezan haptički uređaj, i softvera napisanog korišćenjem H3DAPI platforme. U ovom sistemu korišćen je Phantom Omni haptički robot. Cilj softvera je da simulira crtanje unapred zadatih primitiva na ekranu pomoću haptičkog uređaja. Kako bi proverili ispravnost sistema, kao i efekte haptičkog interfejsa, izveden je eksperiment u kojem su učestvovali šest zdravih ispitanika. Oni su ne dominantnom rukom vežbali precrtavanje primitive, sa i bez pomoći haptike, koristeći haptičkog robota. Primitive su odabrane iz standardnog Bender-Gestalt testa. Rezultati eksperimenta su pokazali da su ispitanici koji su vežbali bez pomoći imali mnogo veći napredak, ali i da pomoć haptičkog interfejsa ne poboljšava postojeće već razvija nove veštine. Sistem se pokazao upotrebljivim, tako da bi mogao biti korišćen u budućim eksperimentima.

ABSTRACT

System has been created for testing haptic effects on motor learning. It consists of computer, on which is connected haptic device, and software written using H3DAPI platform. In this system it has been used Phantom Omni haptic robot. The purpose of the software is to simulate drawing of already given primitives on the screen using haptic device. In order to verify the correctness of the system, and the effects of haptics interface, an experiment has been done which involved six healthy subjects. They were practicing drawing primitives with their non-dominant hand, with and without help of haptics, using haptic robot. Primitives were selected from the standard Bender-Gestalt test. Experimental results showed that subjects who were practicing without the aid had a much better improvement, but also that help of haptic interface does not improve present but develops new skills. The system has proven to be useful, so it might be used in future experiments.

ZAHVALNICA

Ovaj rad je nastao kao rezultat istraživanja u okviru laboratorije za Biomedicinsku instrumentaciju i tehnologije, odseka za Signale i sisteme, Elektrotehničkog fakulteta u Beogradu.

Na pomoći prilikom izrade ovog rada posebno se zahvaljujem:

Prof. dr. Dejanu Popoviću

M.Sc. Milošu Kostiću

Sadržaj

1. Uvod.....	7
2. Haptički interfejs.....	9
2.1 HAPI.....	9
2.2 H3DAPI	9
3. Sistem	11
4. Eksperiment	19
5. Obrada rezultata	22
6. Rezultati eksperimenta.....	24
7. Diskusija.....	26
8. Literatura.....	28

1. Uvod

Pokreti rukom su veoma bitni u svakodnevnom životu. Koriste se pri pisanju, jelu, kujanju, oblačenju, itd. Kako bi pravilno izvodio ove pokrete čovek mora prvo da ih nauči. E. R. Guthrie je 1952. u knjizi "Psihologija: principi i značenja" definisao učenje pokreta kao relativno trajnu promenu, izazvanu vežbanjem i novim iskustvom, u mogućnostima reagovanja.

Kao posledica moždanog udara ili povreda mozga može se javiti hemiplegija. Hemiplegija je stanje potpunog gubitka neuromišićnih funkcija, ispoljenih na jednoj strani tela, izazvano oštećenjem suprotne strane mozga. Kako bi se vratile neuromišićne funkcije pacijentu je neophodna rehabilitacija. Proces učenja je ključan u vraćanju motornih funkcija paretičnih ekstremiteta kod hemiplegičnih pacijenata [1]. Ove vežbe često umeju da budu monotone, takodje terapeut uvek mora biti prisutan, kako bi korigovao pacijenta u slučaju da neku vežbu radi nepravilno. Proces učenja može biti uvećan pravilnom korišćenjem pomoći robota u okviru vežbi sa zadatim ciljem [2].

Haptika je dobila ime po grčkoj reči *haptesthai*, što znači dodirnuti, osetiti. To je tehnologija koja omogućava korisniku da manipulira virtualnim objektom, ali takodje i da dobija povratne informacije o njihovoj fizici i kretanju. Koristeći haptički uređaj (džojstik, robot, rukavica...) on može delovati silom na objekat i time promeniti njegovu strukturu ili delovati na njegovo kretanje. Uređaj takođe pruža informacije o fizičkoj strukturu objekata (masi, hrapavosti površine, otpornosti na ugibanje...) kao i silama koje postoje u sceni. Haptika je prvi put primenjena u avionima za upravljanje kontrolnim sistemom, a kasnije je našla primenu u mašinama za iskopavanje. Danas se koristi u različite svrhe: u medicini za simuliranje hirurških zahvata, u dizajnu za modelovanje 3D modela, kontroleri igračkih konzola (Wii, X Box, Play Station...) su haptički uređaji.

Upotreba haptičkog interfejsa između paretičnog ekstremiteta i robota predložena je kao prihvatljivo rešenje za vežbe sa zadatim ciljem [2-8]. Ona pruža mogućnost da kroz zabavu, igrajući igru na računaru, pacijent vežba zadate pokrete. On ima povratnu informaciju, preko haptičkog uređaja, da li dobro izvodi pokret, i u slučaju da ga ne radi pravilno, uređaj ga koriguje. Na ovaj način pacijent, bez pomoći terapeuta, može pravilno da vežba. Moguće je takođe snimati sve parametre (trajektorija, brzina, ubrzanje...) vežbanja pacijenta i na osnovu rezultata imati tačan uvid u napredak pacijenta, ali i informacije o kvalitetu same vežbe kao i celog sistema.

Cilj ovog rada je razvoj i testiranje sistema, za vežbanje pokreta rukom, koji bi koristio haptički interfejs. Sistem je morao biti projektovan tako da ne zavisi od haptičkog uređaja koji se koristi i da bude lak za korišćenje. Kako nisu bili poznati efekti haptičkog interfejsa na korisnika sistema, testiranje je vršeno u okviru eksperimenta u kojem su učestvovali zdravi ispitanici. U ovom radu je dat opis sistema kao i samog eksperimenta.

2. Haptički interfejs

Kako bi napravili sistem koji koristi mogućnosti koje pruža haptika bilo je neophodno iskoristi mogućnosti nekog od postojećih haptičkih interfejsa. Pošto haptika postaje sve popularnija oblast, broj haptičkih interfejsa raste velikom brzinom. Trenutno najzastupljeniji su: H3DAPI, OpenHaptics i CHAI3D. U ovom sistemu korišćen je H3DAPI zbog kvalitetne dokumentacije i mogućnosti jednostavnog korišćenja u okviru programskog jezika C++.

2.1 HAPI

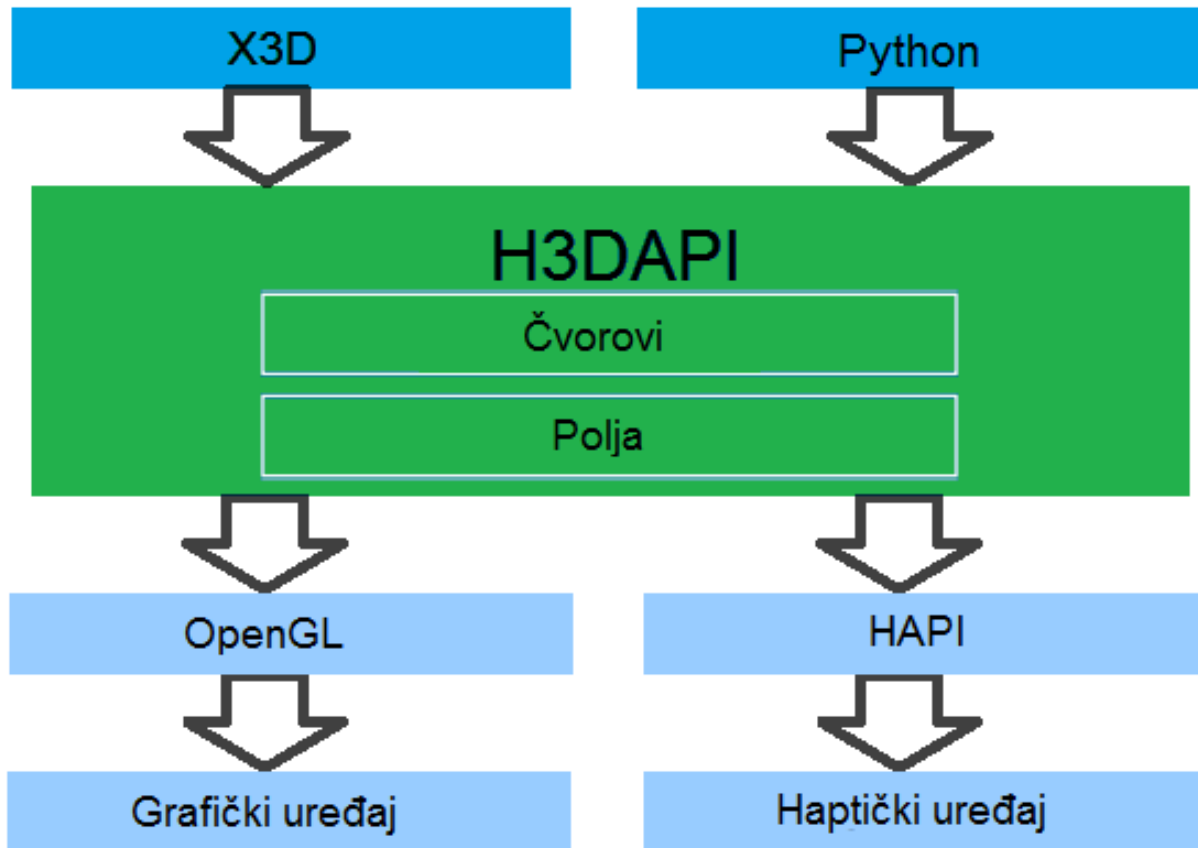
HAPI (Haptics Application Programming Interface) je otvoren više-platformski endžin za renderovanje haptike napisan u potpunosti u programskom jeziku C++. On je u potpunosti nezavisan od haptičkog uređaja koji se koristi, tako da kod napisan za jedan, potpuno isto radi i na svim ostalim uređajima. Korisniku se pruža mogućnost da bira između različitih algoritama za renderovanje, različitih efekata sila i nekoliko vrsta površina objekata. HAPI obuhvata četiri funkcionalnosti:

- Upravljanje uređajima koje omogućava HAPI-ju da radi sa različitim haptičkim uređajima.
- Haptika geometrije, sadrži tri celine: obradu sudara, haptičke algoritme za renderovanje i algoritme reakcije sa površinom.
- Haptiku slobodnog prostora, sa kolekcijom efekata sila.
- Upravljanje nitima.

2.2 H3DAPI

H3DAPI(Haptics 3D Application Programming Interface) je otvorena platforma za razvoj softvera razvijen, u programskom jeziku C++. Ona koristi OpenGL za grafičko i HAPI za haptičko renderovanje. Suštinski to je sloj koji obuhvata grafiku i haptiku u jednu platformu i implementira X3D i Python kao interfejs za razvoj aplikacija (Slika 1). Sastoji se iz čvorova i polja. Polja su osnovni gradivni blokovi u H3DAPI-ju. Ona mogu da sadrže podatke, da budu zavisni jedni od drugih i računaju svoju vrednost u odnosu na trenutne vrednosti polja od kojih zavisi. Čvorovi su strukture koje sadrže polja, koja

su grupisana tako da okružujući čvor predstavlja novi entitet (slično klasama i atributima). Tako na primer postoji čvor Sfera sa poljima: ime,poluprečnik,boja...Zavisnost između polja se definiše pomoću ruta. Ona koji učestvuju u jednoj ruti moraju biti istog tipa.



Slika 1: Način rada H3D API-ja.

3. Sistem

Sistem se sastoji iz PC računara koji poseduje IEEE-1394a FireWire port., Haptičkog robota Phantom Omni i softvera napisanog u jeziku C++ koristeći H3DAPAPI platformu.

Phantom Omni je haptički robot firme Sensable (Slika 2). Povezuje se sa računarom preko IEEE-1394a FireWire porta. Posедуje šest stepeni slobode kao i “force feedback” po sve tri ose. Korisnik kontroliše robota tako što pomera ručicu koja ima oblik olovke. Koordinate metalnog vrha ručice se transformišu u koordinate pokazivača u virtualnom svetu. Pošto robot ima male dimenzije samim tim je i sloboda pokreta ograničena tako da je ovaj robot korišćen samo za testiranje ispravnosti rada sistema. Sistem je razvijen tako da može da radi, bez izmena, sa bilo kojim drugim haptičkim uređajem.



Slika 2: Haptički robot Phantom Omni.

Softver je napisan u Microsoft Visual Studi-u 2005, u jeziku C++. Cilj mu je da simulira crtanje određenih primitiva pomoću haptičkog robota. On iscrtava primitive na ekranu, koje korisnik treba da precрта, u ravni paralelnoj sa ravni ekrana. Kako bi korisnik imao bolji osećaj prostora dodate su horizontalne i vertikalne paralelne bele linije, u istoj ravni sa primitivama. Koordinate primitiva se čitaju iz fajla i kao XML stringovi se prosleđuju H3D-API-ju (Listing 1). Pri iscrtavanju scene se zadaju i haptičke sile koje mogu da deluju na haptički uređaj. Ovde je upotrebljena sila koja drži uređaj u ravni u kojoj se nalaze primitive koje se precrtavaju i sila koja vuče pokazivač ka ivičnim linijama primitiva. Sila koja drži robota u ravni primitiva realizovana je u funkciji pozicije pokazivača. Ako su koordinate pokazivača x,y i z funkcija ove sile je 0,0 i -100z. Sila koja vuče robota ka ivičnim linijama primitiva je definisana rastojanjem. Tako da ako je pokazivač u okviru zadatog rastojanja od primitive neće biti nikakvog efekta, ali ako se odalji preko dozvoljenog rastojanja sila će početi da koriguje njegovu putanju tako što će ga vući ka primitivi sve dok se pokazivač ne nađe na prihvatljivoj udaljenosti od primitive.

```

ifstream myfile;
myfile.open("C:\\Users\\Pera\\Desktop\\Robot Crtanja\\krug.txt");
string pom;
getline(myfile,pom);
while(!myfile.eof())
{
    pom+=" 0";
    s+=pom;
    s+=" ";
    br++;
    getline(myfile,pom);
}
myfile.close();
}
stringstream ssl;
ssl << br;
string scena= "<Group>\n"
"<Viewpoint position=\"0 0 0.25\" />\n"
" <Group DEF=\"GROUP\" />\n"
    "<DeviceInfo>\n"
        "<PhantomDevice DEF=\"device\">\n"
            "<GodObjectRenderer/>\n"
            "<Shape containerField=\"stylus\">\n"
                "<Appearance>\n"
                    "<Material/>\n"
                "</Appearance>\n"
            "<Sphere radius=\"0.0025\"/>\n"
        "</Shape>\n"
    "</PhantomDevice>\n"
"</DeviceInfo>\n"
"<Transform translation=\"-0.066 0 0 \" scale=\" 0.066 0.066 1\">\n"

```

```

"<Shape>\n"
  "<Appearance>\n"
    "<Material emissiveColor=\"1 0 0\"/>\n"
  "</Appearance>\n"
  "<LineSet DEF=\"KRUG\" vertexCount=\""+ssl.str()+"\">\n"
    "<Coordinate point=\""+s+"\"/>\n"
  "</LineSet>\n"
"</Shape>\n"
"<MagneticGeometryEffect DEF=\"MGEKRUG\" enabled=\"false\" startDistance=\"0.1\" escapeDistance=\"0.15\">\n"
"<LineSet USE=\"KRUG\"/>\n"
"</MagneticGeometryEffect>\n"
"</Transform>"
"<Transform scale=\" 0.066 0.066 1\">\n"
  "<Shape>\n"
    "<Appearance>\n"
      "<Material emissiveColor=\"1 0 0\"/>\n"
    "</Appearance>\n"
    "<LineSet DEF=\"KVADRAT\" vertexCount=\"5\">\n"
      "<Coordinate DEF=\"coorline\" point=\"0 0 0 1 1 0 2 0 0 1 -1 0 0 0 0\" />\n"
    "</LineSet>\n "
  "</Shape>\n"
  "<MagneticGeometryEffect DEF=\"MGEKVADRAT\" enabled=\"false\" startDistance=\"0.1\" escapeDistance=\"0.15\">\n"
  "<LineSet USE=\"KVADRAT\"/>\n"
  "</MagneticGeometryEffect>\n"
"</Transform>"
"<Shape >\n"
  "<LineSet vertexCount=\"2\">\n"
  "<Coordinate DEF=\"coorline\" point=\"-1 0 0 1 0 0\" />\n"
  "</LineSet>\n "
  " <Appearance>\n"
    " <Material emissiveColor=\"1 1 1\" transparency=\"0.3\" shininess=\"0.4\" />\n"
  " </Appearance>\n"
" </Shape> \n"
"<Shape >\n"
  "<LineSet vertexCount=\"2\">\n"
  "<Coordinate DEF=\"coorline\" point=\"-1 0.033 0 1 0.033 0\" />\n"
  "</LineSet>\n "
  " <Appearance>\n"
    " <Material emissiveColor=\"1 1 1\" transparency=\"0.3\" shininess=\"0.4\" />\n"
  " </Appearance>\n"
" </Shape> \n"
"<Shape >\n"
  "<LineSet vertexCount=\"2\">\n"

```

```

        "<Coordinate DEF=\"coorline\" point=\"-1 0.066 0 1 0.066 0\"
        />\n"
        "</LineSet>\n "
        " <Appearance>\n"
            "          <Material          emissiveColor=\"1          1          1\"
            transparency=\"0.3\" shininess=\"0.4\" />\n"
        " </Appearance>\n"
    " </Shape> \n"
    "<Shape >\n"
        "<LineSet vertexCount=\"2\">\n"
        "<Coordinate DEF=\"coorline\" point=\"-1 -0.033 0 1 -0.033
        0\" />\n"
        "</LineSet>\n "
        " <Appearance>\n"
            "          <Material          emissiveColor=\"1          1          1\"
            transparency=\"0.3\" shininess=\"0.4\" />\n"
        " </Appearance>\n"
        " </Shape> \n"
    "<Shape >\n"
        "<LineSet vertexCount=\"2\">\n"
        "<Coordinate DEF=\"coorline\" point=\"-1 -0.066 0 1 -0.066
        0\" />\n"
        "</LineSet>\n "
        " <Appearance>\n"
            "          <Material          emissiveColor=\"1          1          1\"
            transparency=\"0.3\" shininess=\"0.4\" />\n"
        " </Appearance>\n"
        " </Shape> \n"
    "<Shape >\n"
        "<LineSet vertexCount=\"2\">\n"
            "<Coordinate DEF=\"coorline\" point=\"0 1 0 0 -1 0\"
            />\n"
        "</LineSet>\n "
        " <Appearance>\n"
            "          <Material          emissiveColor=\"1          1          1\"
            transparency=\"0.3\" shininess=\"0.4\" />\n"
        " </Appearance>\n"
    " </Shape> \n"
    "<PositionFunctionEffect>\n"
    "<GeneralFunction containerField=\"xFunction\" function=\"0\"
    params=\"x,y,z\"/>\n"
    "<GeneralFunction containerField=\"yFunction\" function=\"0\"
    params=\"x,y,z\"/>\n"
    "<GeneralFunction containerField=\"zFunction\" function=\"-
    100*z\" params=\"x,y,z\"/>\n"
    "</PositionFunctionEffect>\n"
    "</Group>";

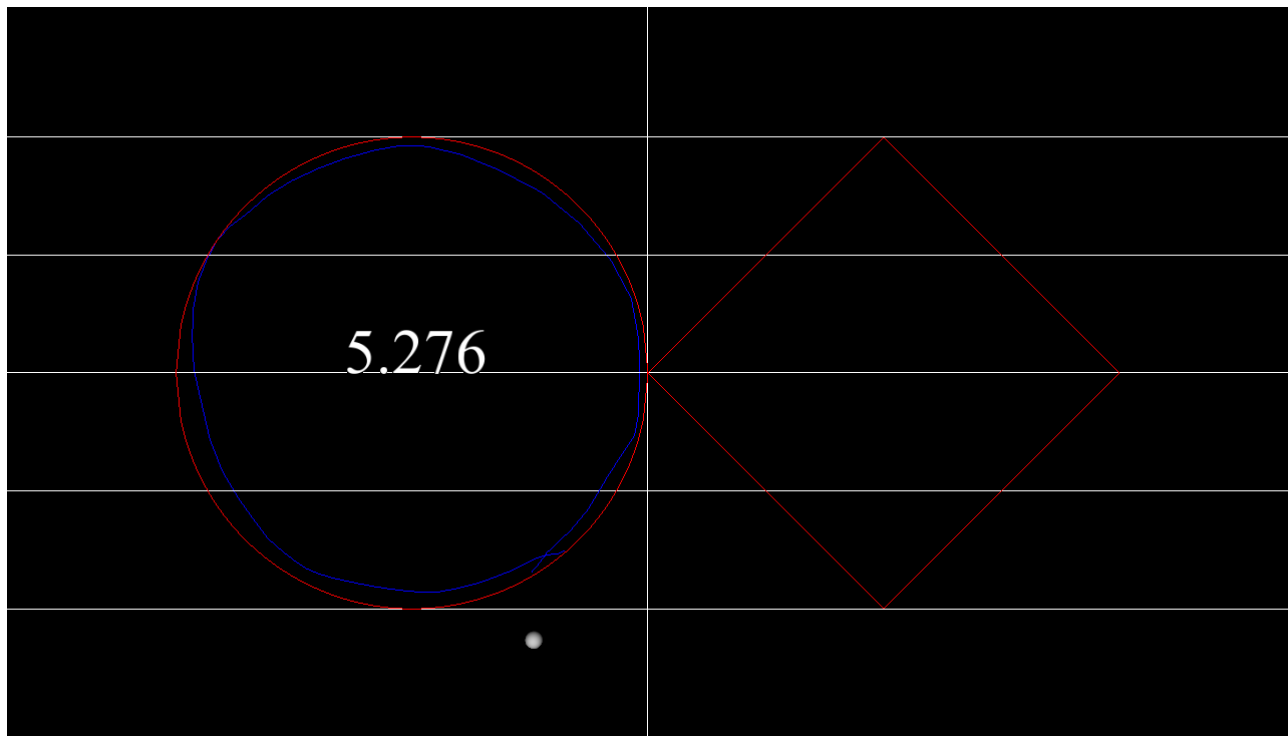
    AutoRef< Node > group(
        X3D::createX3DNodeFromString(scena,
        &myDefNodes ) )AutoRef< Scene > scene( new Scene );
    GLUTWindow *glwindow = new GLUTWindow;

```

```
scene->window->push_back( glwindow );
scene->sceneRoot->setValue( group.get() );
Scene::mainLoop();
```

Listing 1: Kreiranje scene.

Korisnik precrtava primitive tako što drži dugme na robotu i povlačeći ručicu robota, a pri tome pokušava da kursorom na ekranu prati linije primitiva koje se nalaze na ekranu. Kursor pri tome ostavlja plavi trag za sobom kako bi korisnik imao povratnu informaciju o uspešnosti precrtavanja. Kako bi mogli da se vrše eksperimenti sa i bez pomoći haptičkih sila u okviru iste aplikacije, ona je napravljena da radi u dva režima. Prvom, u kojem haptička sila pomažu samo držanjem robota u ravni primitiva i drugom u kojem pomažu i sile koje vuku pokazivač ka ivicama primitiva. Režim rada se određuje držanjem odgovarajućeg dugmeta. Po završetku precrtavanja na ekranu se ispisuje vreme koje mu je za to bilo potrebno (Slika 3). U fajl se upisuju koordinate vrha ručice robota za vreme precrtavanja sa frekvencijom od 30Hz (Listing 2).



Slika 3: Izgled aplikacije u toku vežbanja

```

class Button : public AutoUpdate< SFBool >
{
    virtual void update() {
        AutoUpdate<SFBool >::update();
        if (!(clockind)&&(value))
        {
            pocetak=clock();
            clockind=true;
            if (brisivreme)
            {
                Group *g1;
                myDefNodes.getNode( "GROUP", g1 );
                g1->children->pop_back();
                brisivreme=false;
            }
        }
        if ((clockind)&&! (value))
        {
            kraj=clock();
            double pomvreme=(double) (kraj-pocetak)/CLOCKS_PER_SEC;
            stringstream ss (stringstream::out);
            ss<<pomvreme;
            string vreme=ss.str();
            string transformacija;
            if (orijentacija<0)
            {
                transformacija="-0.085 0 0";
                tl+=pomvreme;
            }
            else
            {
                transformacija="0.045 0 0";
                tr+=pomvreme;
            }
            orijentacija=0;
            clockind=false;
            X3D::DEFNodes *tempDefNodes = new X3D::DEFNodes();
            AutoRef< Transform > tempTransform(
                static_cast< Transform
* >( X3D::createX3DNodeFromString(
            "<Transform      scale=\"      0.02      0.02      1\"
translation=\"\"+transformacija+\"\">\n"
            " <Shape DEF=\"SHAPE\" >\n"
            "<Text string='\""+vreme+\"' />\n"
            " <Appearance>\n"
            " <Material diffuseColor=\"1 1 1\"/>\n"
            " </Appearance>\n"
            " </Shape> \n"

```



```

        "</Transform>", tempDefNodes ).get() ) );

        Group *g;
myDefNodes.getNode( "GROUP", g );
g->children->push_back( tempTransform.get() );
brisivreme=true;

    }
ind=value;
}

};

class Crtanje : public AutoUpdate< SFVec3f> {
Vec3f pom;
ofstream myfile;
int br;
public:
Crtanje()
{
    pom[0]=0;
    pom[1]=0;
    pom[2]=0;
    myfile.open("C:\\Users\\Pera\\Desktop\\Robot
Crtanja\\Kole1.txt");
    br=0;
}
virtual void update() {
AutoUpdate<SFVec3f>::update();
if ((br>4)&& (ind))
{
    orijentacija+=value[0];
X3D::DEFNodes *tempDefNodes = new X3D::DEFNodes();
AutoRef< Transform > tempTransform(
static_cast< Transform * >( X3D::createX3DNodeFromString(
"<Transform>\n"
" <Shape DEF=\"SHAPE\" >\n"
"<LineSet vertexCount=\"2\">"
"    <Coordinate DEF=\"coorline\" point=\"0 0 0 0
0\" />"
"</LineSet>\n "
" <Appearance>\n"
" <Material emissiveColor=\"0 0 1\"/>\n"
" </Appearance>\n"
" </Shape> \n"
"</Transform>", tempDefNodes ).get() ) );
Coordinate *c;
tempDefNodes->getNode( "coorline", c );
stringstream ss (stringstream::in | stringstream::out);
ss << pom[0]<<" " <<pom[1]<<" " <<pom[2]<<" " <<value[0]<<"
"<<value[1]<<" " <<value[2];

```

```

    string s=ss.str();
    c->point->setValueFromString(s);
    Group *g;
    myDefNodes.getNode( "GROUP", g );
    g->children->push_back( tempTransform.get() );
    if(value[0]<0) xpoz=-1;
    else xpoz=1;
    myfile<<value[0]<<" "<<value[1]<<" "<<value[2]<<endl;
    }
    else br++;
    pom=value;
}

```

```

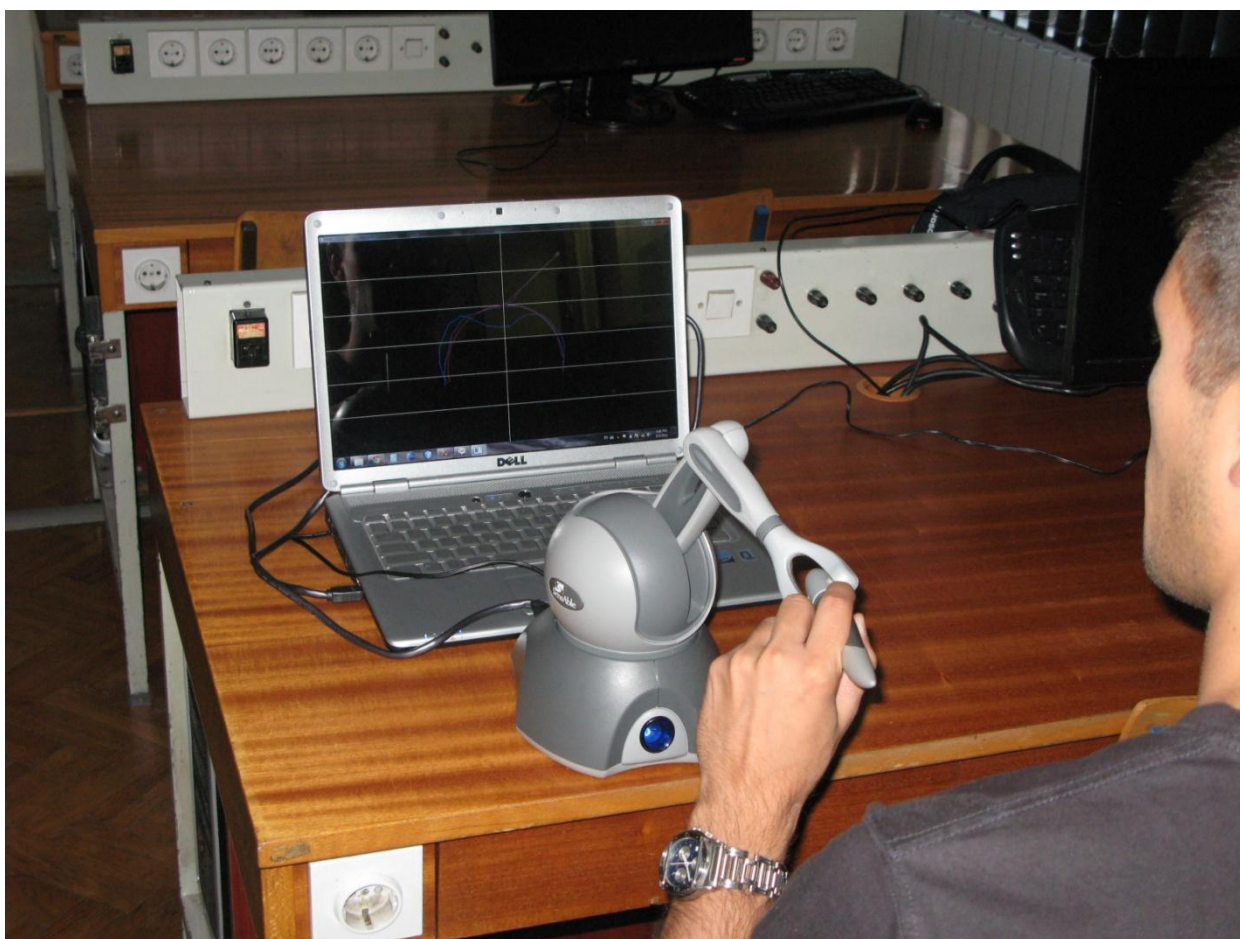
Crtanje *crtanje = new Crtanje();
Button *button=new Button();
PhantomDevice *dev;
MagneticGeometryEffect *mgeKRUG,*mgeKVADRAT;
myDefNodes.getNode( "device", dev);
myDefNodes.getNode( "MGEKVADRAT", mgeKVADRAT);
myDefNodes.getNode( "MGEKRUG", mgeKRUG);
dev->weightedProxyPosition->route( crtanje );
dev->mainButton->route( button);
button->route( mgeKVADRAT->enabled);
button->route( mgeKRUG->enabled);

```

Listing 2: Precrtavanje primitive.

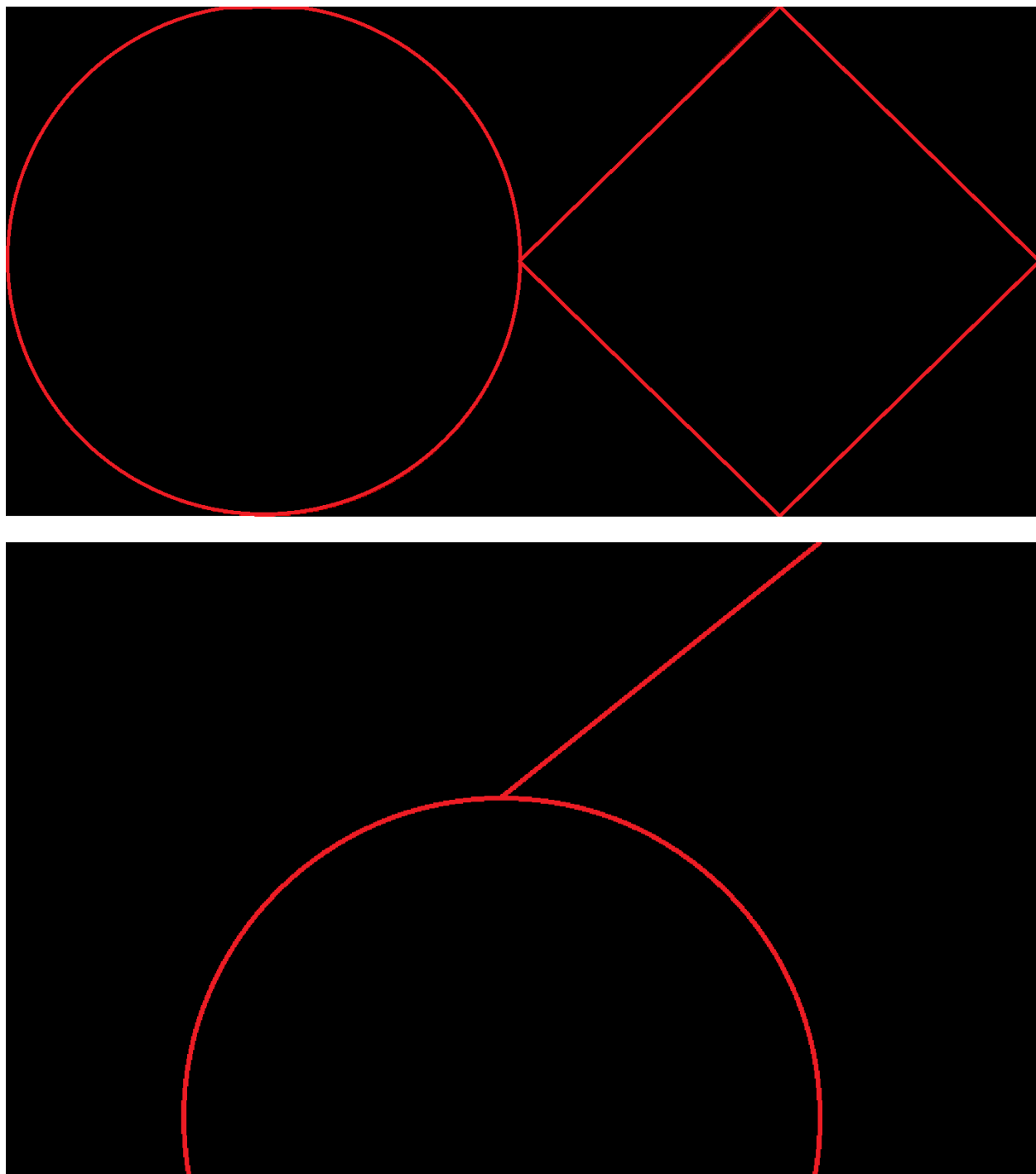
4. Eksperiment

Cilj eksperimenta je bio da se utvrdi ispravnost razvijenog sistema i da se vidi da li i koje efekte ima haptički interfejs na učenje pokreta. Zamišljen je da ispitanici sede za stolom na kojem se nalaze monitor, a tačno ispred njega haptički robot. Oni bi sedeli tako da mu se robot nalazi ispred ramena ne dominantne ruke. Na monitoru su bile prikazane primitive koje treba da precrta rukom koristeći haptičkog robota, u ravni paralelnoj sa monitorom (Slika 4).



Slika 4: Eksperimentalna postavka.

Ispitanici bi pri tome robota kontrolisali ne dominantom rukom. Primitive koje su crtali su odabrane iz standardnog Bender-Gestalt testa. To su krug, kvadrat i potkovica (Slika 5).



Slika 5: Izgled primitiva koje su ispitanici precrtavali tokom eksperimenta. Krug i kvadrat (gore), i potkovica(dole).

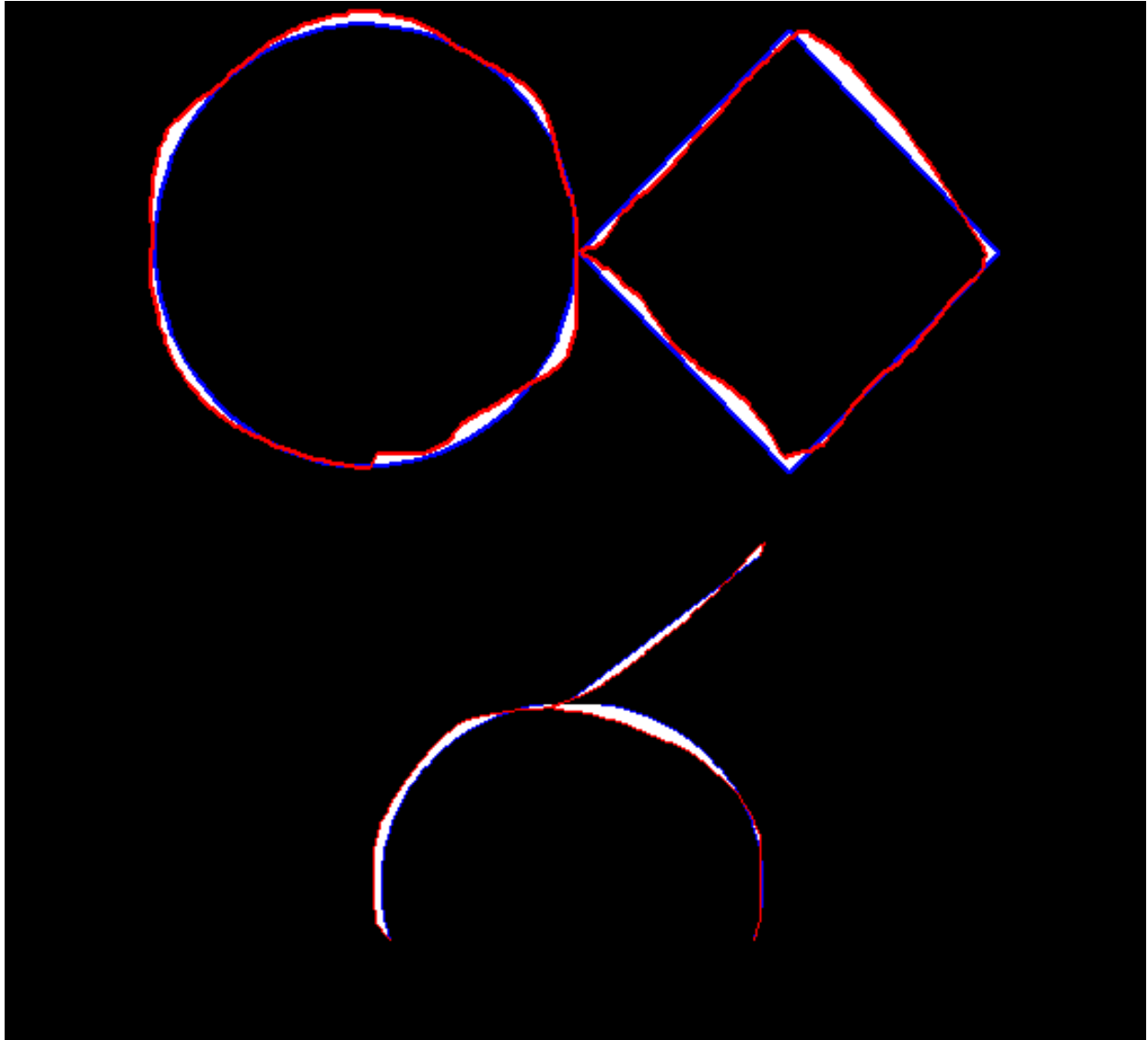
U eksperimentu je učestvovalo šest ispitanika prosečne starosti 26 godina. Podeljeni su slučajnim odabirom u dve grupe od po troje. Jedna grupa je vežbala bez pomoći haptičkog sila koje vuku kursor ka ivicama primitiva, dok je drugoj grupi pružana ova pomoć.

Eksperiment se sastojao iz 3 faze. Prve u kojoj se svaki ispitanik upoznao sa opremom, načinom izvođenja eksperimenta i tada je izvršen prvi test. Pokazano im je kako se upravljanja haptičkim robotom i primitive koje će crtati u okviru eksperimenta. Potom im je dato slobodno vreme da se naviknu na rukovanje robotom. Pri tome su koristili dominantnu ruku kako se ne bi javio efekat učenja. Na kraju je izvršen početni test. Ispitanici su ne dominantom rukom nacrtali tri puta svaku primitivu. Od njih je traženo da svaku primitivu pokušaju da nacrtaju za 6 ± 1 sekundi. Vremensko ograničenje je uvedeno kako bi otprilike istom brzinom nacrtali primitive u okviru početnog i krajnjeg testa. U drugoj fazi su vežbali pravilno da nacrtaju zadate primitive. Svako je imao 9 sesija u okviru kojih je po 5 puta nacrtao svaku primitivu. Poslednja treća faza se sastojala iz završnog testa u okviru kojeg su ponovo nacrtali tri puta svaku od primitiva ne dominantnom rukom. Ponovo im je traženo da primitive nacrtaju za 6 ± 1 sekundi.

5. Obrada rezultata

Kompletna obrada rezultata je urađena u programskom paketu MATLAB. Sistem kao izlaz, pri crtanju haptičkim robotom, daje koordinate tačaka koje korisnik nacrtao. Za svakog ispitanika su postojale koordinate početnog i krajnjeg testa za sve tri primitive. Kako su svi u svakom testu tri puta nacrtali primitivu u obradi je uzeta njihova srednja vrednost. Pošto je potrebno odrediti razliku između zadatih primitiva i primitiva koje je korisnik nacrtao, izlazne koordinate se transformišu u sliku. U slučajevima kada korisnik nije “zatvorio” krug ili kvadrat iskorišćena je prava linija za njihovo “zatvaranje”.

U sledećem koraku određeni su odnosi nacrtanih i zadatih slika kako bi predstavili brojnom vrednošću rezultat koji je ispitanik postigao u testu. Krug i kvadrat su poređeni na osnovu apsolutne razlike u površinama zadatih i nacrtanih primitiva. Njihov relativni pomeraj je uzet u obzir, i predstavljena je koeficijentom koji je proporcionalan pomeraju. U slučaju idealnog položaja ovaj faktor iznosi jedan. Za potkovicu razlika je računata između površina dobijenih povezivanjem krajnjih tačaka zadate i nacrtane primitive (Slika 6).



Slika 6: Prikaz razlike u površinama između nacrtanih i precrtanih primitiva.

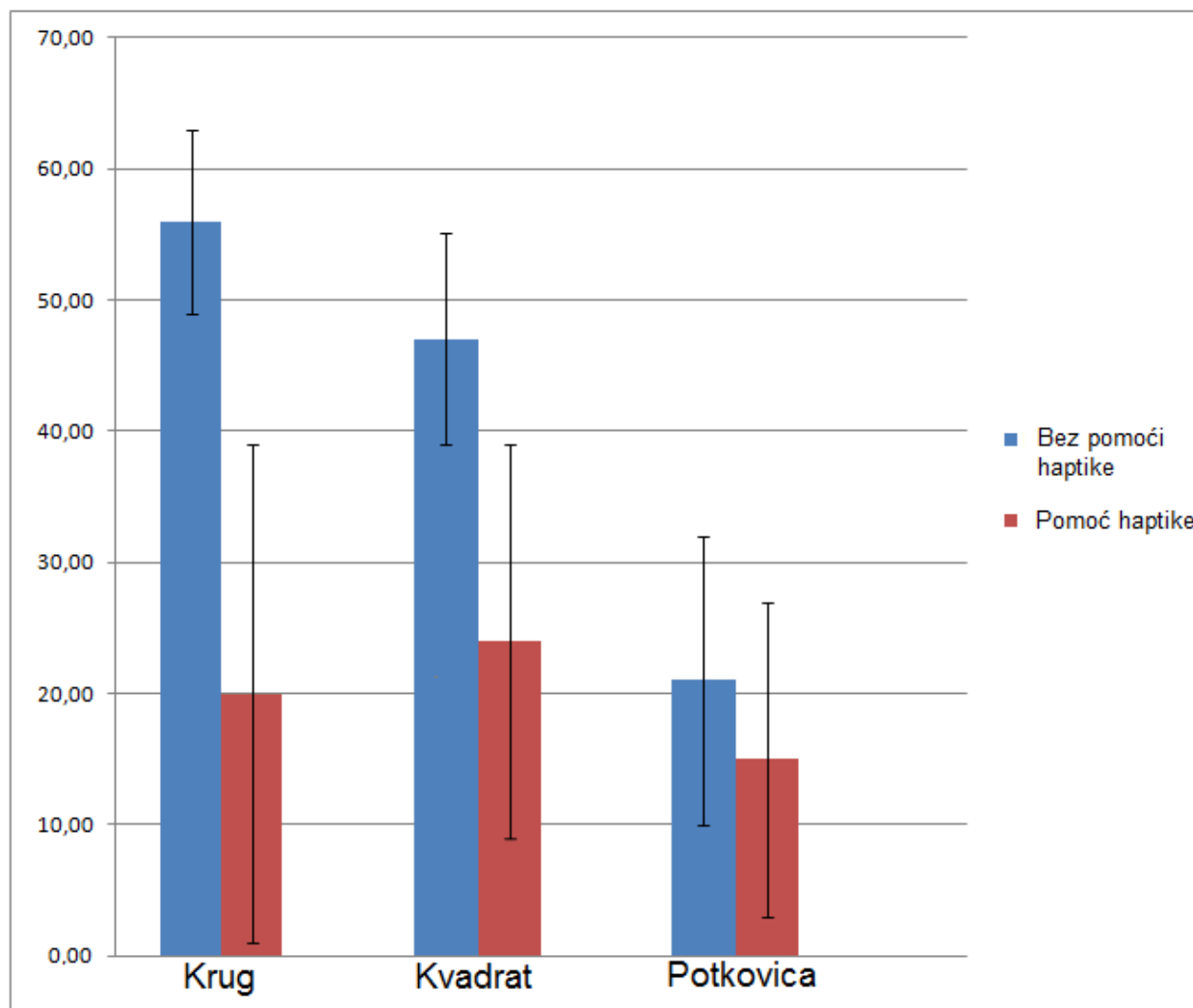
Indeks učenja se određuje na osnovu početnog i krajnjeg testiranja po formuli:

$$I = \left(1 - \frac{T_k}{T_p}\right) * 100$$

gde je T_p rezultat početnog, a T_k rezultat krajnjeg testiranja.

6. Rezultati eksperimenta

Rezultati eksperimenta su prikazani na slici 7.



Slika 7: Rezultati eksperimenta.

Svi ispitanici su pokazali napredak u krajnjem u odnosu na početni testu. Kod crtanja kruga, ne dominantnom rukom, veći indeks učenja su imali ispitanici koji su vežbali bez pomoći haptičkih sila i on iznosi 56, sa standardnom devijacijom koja iznosi 7. Indeks

učenja kod grupe koja je imala pomoć haptičkog uređaja iznosi 20, a standardna devijacija 19.

Indeks učenja je manji kod crtanja kvadrata u odnosu na krug kod ispitanika koji nisu imali pomoć haptike i on iznosi 47, sa standardnom devijacijom od 8. Sa druge strane indeks grupe koja je imala pomoć haptike, kod crtanja kvadrata je veći u odnosu na kruga i jednak je 24, sa standardnom devijacijom od 15.

Najmanji napredak, kod obe grupe, prikazan je pri crtanju potkovice. Kod ispitanika bez pomoći haptike indeks učenja iznosi 21, a standardna devijacija 11, dok je druga grupa imala indeks od 15 sa standardnom devijacijom od 12.

7. Diskusija

Rezultati eksperimenta pokazuju da je posle devet vežbanja grupa koja je vežbala bez pomoći robota više napredovala. Odavde se izvodi zaključak da učenje pokreta pomoću haptičkog robota nije efikasno kao samostalno učenje, ali kao što se vidi iz rezultata daje kakav takav napredak tj. nema negativne efekte.

Napredak kod testa se potkovicom je duplo manji kod ispitanika koji nisu imali pomoć haptike. Ovo je posledica činjenica da je crtanje ovog oblika novi zadatak za sve ispitanike, dok su se sa crtanjem kruga i kvadrata sigurno često sretali do sada. Odavde se može zaključiti da devet vežbanja nije dovoljno da se savlada ovaj pokret. Sa druge strane indeks učenja pri crtanju potkovice, kod ispitanika koji su omali pomoć haptike, je sličan indeksima učenja kod druga dva oblika, tj ispitanici su razvili novu veštinu za nepoznat zadatak. Odavde dolazimo do zaključka da vežbanje pomoću haptičkog robota ne poboljšava postojeću već razvija novu veštinu. Ovakva veština je slična veštini koja se dobija tradicionalnim učenjem, ali nije ista.

U daljem radu treba ispitati da li zajedničko učenje tradicionalnog načina i učenja pomoću haptičkog robota daje bolje rezultate nego pojedinačno. Ovakav način vežbanja se može primeniti kod pacijenata u rehabilitaciji koji ne mogu da izvedu zadati pokret. Oni bi vežbali pomoću haptičkih sila sve dok ne budu u stanju da pokret izvedu samostalno. Nakon toga bi im bile ugašena pomoć i oni bi nastavili sa vežbanjem na tradicionalni način. Takodje treba ispitati kakav efekat bi bio kada bi se ravan u kojoj se crta postavila normalno na ravan monitora. U ovom slučaju on bi crtao u horizontalnoj dok bi mu grafički prikaz bio u vertikalnoj ravni. Pošto ovo nije tradicionalni način crtanja, rezultati ovog eksperimenta ne mogu se predvideti. Zato postoji opcija da se i monitor stavi u horizontalnu ravan pa da ispitanik crta direktno po njemu. Ovo bi trebalo da simulira standardno crtanje, ali je pitanje koliki bi napredak(indeks učenja) bio u ovom slučaju.

Razvijeni sistem se pokazao kao dobro rešenje i može biti korišćen u daljem radu. Razvijen da bude nezavisan od haptičkog uređaja koji se koristi tako da se mogu koristiti složeniji roboti, koji omogućavaju mnogo veću slobodu pokreta.

Sistem je ispoljio i neke nedostatke. Sile koje drže haptički uređaj na ivicama primitiva nisu dovoljno dobro definisane. Dešava se da na prelazima sa jednog pravca na drugi dolazi do sabiranja sila ova dva pravca, što prouzrokuje nepravilan rad robota i grešaka pri crtanju. Ovo se može rešiti boljom definicijom sila, tako što bi se palile i gasile sile u

zavisnosti koji segment ispitanik trenutno crta. Intenzitet sila takodje treba definisati boljom funkcijom jer se dešavalo da sila previše snažno deluje na haptičkog robota, a samim tim i na pacijenta. Ovo je posebno problem ako robota koriste pacijenti u rehabilitaciji ili deca.

Tokom rada primećeni su određeni nedostaci H3DAPI platforme. Često se događao konflikt između H3DAPI-ja i drajvera grafičke karte. U ovom slučaju bilo je neophodno restartovati računar. Javio se i problem zahtevnosti grafičkih resursa računara pošto se u isto vreme renderuje grafika i haptika. Ovo je rešeno prelaskom na računar sa boljom grafičkom kartom, ali ni to nije dalo očekivane performanse. Aplikacija bi počela da koči već posle nekoliko nacrtanih primitiva. U daljem radu biće testirane ostale haptičke platforme sa ciljem pronalaska platforme sa najboljim odnosom mogućnosti i performansi.

8. Literatura

1. Popović DB, Sinkjær T, Popović MB (2009) Electrical stimulation as a means for achieving recovery of function in stroke patients. *J Neuro Rehabilitation*. 25:45-58.
2. Krebs HI, Hogan N, Aisen ML, Volpe BT (1999) Robot-Aided Neurorehabilitation. *IEEE Trans Rehabil Eng*. 6(1): 75–87.
3. Hogan N, Krebs HI, Charnnarong J (1992) MIT-MANUS: a workstation for manual therapy and training. In *Proc. IEEE International Workshop on Robot and Human Communication*, Tokyo, Japan, 2003, ISBN: 0-7803- 0753-4, 1992.
4. Hogan N, Krebs HI, Charnnarong J, et al. (1993) MITMANUS: a workstation for manual therapy and training. In *Proc. SPIE 1833*, 28, doi:10.1117/12.142124, 1993.
5. Casadio M, Sanguineti V, Morasso PG, Arrichiello V (2006) Braccio di Ferro: A new haptic workstation for neuromotor rehabilitation. *Technology and Health Care* 14:123–142.
6. Hesse S, Schmidt H, Werner C, Bardeleben A (2003) Upper and lower extremity robotic devices for rehabilitation and for studying motor control. *Current Opinion in Neurology* 16, 6: 705-710.
7. Casadio M, Morasso PG, Ide AN, et al. (2009) Measuring functional recovery of hemiparetic subjects during gentle robot therapy. *Measurement* 42(8): 1176-1187.
8. Zihel J, Podobnik J, Sikic M, Munih M (2009) Pick to place trajectories in human arm training environment. *J. of Technology and Health Care*, 17 (4): 323-335.