

Direct hand-movement control in virtual space: a potential interface for virtual lab purposes

Ádám Skobrak

Budapesti Elektromos Művek Nyrt., Váci street 72-74, 1132 Budapest, Hungary, a.skobrak@gmail.com

Abstract

One of the most modern pedagogical trends is the project-based education, in which autonomous learning and student motivation play a key role. In this article the independent student activity and the effective communication between the student and the teacher who became a mentor are so important. In human-computer interaction, there is more and more opportunities beyond the use of traditional, customary tools. During the project task, using mobile motion detection, mobile robot speed control takes place in virtual space, which provides a good test environment for human control. The system tests show that 3D hand gesture-based control is initially uncertain compared to a computer mouse task, but can be used almost as effectively after a short practice. Based on the results obtained, it can be concluded that hand gesture sensing can serve as an appropriate interface for performing tasks in the future virtual lab, which offers new possibilities for distance education and virtual university also.

Keywords: project-based learning, mentor, human-machine interaction; Leap Motion; Robotino

Kézmozgás alapú irányítás a virtuális térben: a virtuális laboratórium egy lehetséges interfésze

Skobrák Ádám

Budapesti Elektromos Művek Nyrt., Váci street 72-74, 1132 Budapest, Hungary, a.skobrak@gmail.com

Absztrakt

A legkorszerűbb pedagógiai irányzatok közé tartozik a projekt alapú oktatás, amely során az autonóm tanulás és a tanulói motiváció kiemelt szerepet tölt be. A cikkben bemutatott feladat megvalósítása során kulcsfontosságúnak bizonyult az önálló tanulói tevékenység és a mentor szerepet betöltő pedagógus közötti hatékony kommunikáció. A projekt során kézmozgás érzékelés felhasználásával mobil robot sebességvezérlése valósul meg virtuális térben, mely jó tesztkörnyezetet ad az ember általi irányítás vizsgálatához. A rendszerrel végzett tesztek azt mutatják, hogy az irányítási feladat egérrel történő megvalósításához képest a 3D kézmozgás alapú irányítás kezdetben bizonytalan, azonban gyakorlással közel hasonlóan hatékonyan használható. A kapott eredmények alapján megállapítható, hogy a kézmozgás érzékelés megfelelő interfészként szolgálhat a jövő virtuális laboratóriumában végzett feladatok végrehajtásához, mely új lehetőségeket tartogat a távoktatás és a virtuális egyetem vonatkozásában is.

Kulcsszavak: projekt alapú oktatás, mentor, ember-számítógép interakció; Leap Motion; Robotino

1. Bevezető

Napjainkban a tanulói igényekben egyre inkább megjelenő, jól hasznosítható, gyakorlatiasabb és naprakész tudást kell biztosítani. Ráadásul az ilyen jellegű elvárások magukban hordozzák a tanár szerepének a megváltozását is, ahol az egyirányú, frontális ismeretközlés helyett nagyobb hangsúlyt kell fektetni a tanulási folyamat megszervezésére, támogatására és a

diákok aktív részvételére. Az ilyen jellegű igények kielégítése leginkább a projekt módszer alkalmazásával valósítható meg, ahol a tanulók nem szigorú keretek közé szorítva oldanak meg egy kisebb feladatot, hanem komplexebb összefüggéseket feltárva, akár csapatban együttműködve törekednek egy közös cél elérésre. A cikkben a jövő virtuális laboratóriumában végzett feladat végrehajtás során is alkalmazható rendszer kerül bemutatásra, amely a virtuális egyetem koncepció irányába is mutat. Immanuel, I. (2002), Eguchi, Y., Ando, Y., & Mizukawa, M. (2007), Zhang, Y., & Yang, B. (2011)

Az ember-számítógép kapcsolat beviteli eszközei folyamatosan fejlődnek, egyre több lehetőséget adva a számítógép, és azon keresztül egyéb más berendezés emberi irányítására vonatkozólag. A cikk első felében szekunder kutatás alapján az elméleti háttér áttekintésre kerül, a hagyományos egér alapú irányítás és a kézmozgás alapú vezérlés fontosabb részei tekintetében, úgy, mint a Leap Motion eszköz működése, hozzá hasonló eszközök bemutatása és összehasonlítása, valamint a kiválasztásuk indoklása. Kaur, H., & Rani, J. (2016), Sonkusare, J. S., Chopade, N. B., Sor, R., & Tade, S. L. (2015), Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., & Harvill, Y. (1986). Sziladi, G., et al. (2017) A cikk második felében a robot vezérlést végrehajtó tervezés, megvalósítás, tesztelés, valamint a tesztelés során felderített hibák, illetve azok javítása kerül bemutatásra, illetve az esetvizsgálat is olvasható, az irányítás eredményeinek összehasonlítása és kiértékelése tekintetében.

2. Hipotézisek, kutatási célok, motiváció

A kutatás során különböző tanulmányok Katona, J., et al. (2016), Katona, J., et al. (2019), Hartanto, R., Susanto, A., & Santosa, P. I., Wachs, J. P., Stern, H., & Edan, Y. (2005) és adatok kerültek felhasználásra, annak érdekében, hogy a megfogalmazott hipotézisek vizsgálhatók legyenek, amelyek a következők:

- Akkor hatékonyabb a kézmozgással vezérelt robot mozgatása, ha egy megközelítő eredményt ér el a hagyományos egér alapú irányítással.
- Fiatalkor könnyebben tudják elsajátítani a kézmozgással vezérelt robotot, mint az idősebbek.
- Összehasonlítás során többen maradnának a hagyományos egér alapú irányítás mellett.
- A jobbik kéz irányításától nem fog sokban eltérni a rosszabbik kéz.

A cikk során kézmozgás segítségével vezérelt mobil robot mozgatása valósul meg virtuális térben, mely tesztkörnyezetben vizsgálatok kerülnek elvégzésre a kézmozgás alapú irányítás végrehajtásának elemzése céljából, valamint ez összehasonlításra kerül a hagyományos egér

alapú irányítással. Az irányítással összefüggésben kapott eredmények kiértékelése alapján feltehetően új tudományos eredmények fogalmazhatók meg a kézmozgás hagyományostól eltérő irányítási funkciók megvalósítása terén, a kutatás célja ezen különbségek feltárása, vizsgálata.

Az ember-számítógép kapcsolat a hagyományostól eltérő megvalósításai számos új lehetőséget rejtenek egyes ember által végzett, elsajátított folyamatok elemzésében. A kéz a legáltalánosabban alkalmazott emberi beavatkozó szerv, ennek mozgásával számos irányítási vagy jelzési funkció megvalósítható. Az alkatrészek precíz összeszerelésében, berendezések irányításában, süket-némák beszédében, vagy épp a rendőr forgalomirányítás jelzéseiben is fontos szerepet játszanak, így ezek végrehajtásának elemzése új lehetőségeket, kutatási potenciált jelent, a téma érdekes kutatási területet jelent.

3. Felhasznált eszközök és technológiák

A cél eléréshez szükség volt egy Leap Motion controller-re és egy optikai egérre, melyek segítségével az egérmozgás illetve a kézmozgás megfigyelhetők, ezek segítségével a kitűzött feladat megvalósítható. A Leap Motion-t a kézmozgás alapú vezérlés módszerének kipróbálása miatt került beszerzésre és egyben azzal a szándékkal is, hogy az eszköz alkalmassá váljon az implementált alkalmazás, illetve, egyéb feladatok megvalósítására is. Ehhez természetesen hozzá kellett férni az eszköz által szolgáltatott információkhoz, vagyis meg kellett oldani az eszközzel történő kommunikációt. A Truts GXT 25 Gaming Mouse-t került kiválasztásra az egérmozgás alapú vezérléshez.

3.1. Leap Motion

Az eszköz optikai alapon működik, 850 nm infravörös tartományban. Ez kívül esik a látható fény spektrumán. Az eszköz kb. 135 fokos tartományba lát. A felismerés az eszköz felett kb. 25 mm-től 600 mm-ig működik. A Leap Motion controller akár 120 képkocka/másodperces képkockával képes nyomon követni a kezeket. Körülbelül 8 köbméter 3D-s interaktív teret biztosít.



1. ábra: Leap Motion robbantott ábrája

3.2. *Trust GXT 25 Gaming Mouse*

Az egér alapú irányításhoz, szükséges volt egy vezetékes optikai egér, amivel a kitűzött feladat megvalósítható. A pontosság tekintetében az egér választásánál a legfontosabb elem a szenzor volt, mivel a megvilágítási technológiája befolyásolja a mozgáskövetést. A Trust GXT 25 Gaming egér a cél megvalósításához elegendőnek bizonyult.



2. ábra: Trust GXT 25 Gaming egér

3.3. *Vezérlőprogramot futtató hardver*

A Leap Motion kontroller adatait feldolgozó és a virtuális környezetet, valamint robotot megjelenítő Windows PC-n futó program egy Acer E1-570G típusú laptopon hajtódik végre, melynek teljesítménye a Core i3 processzornak és a dedikált videokártyának köszönhetően teljes mértékben megfelelt.

3.4. Vezérelt virtuális mobil robot

A Festo Robotino® egy FESTO gyártmányú mobil robot, amely az oktatásban, különböző képzésekben és a Festo által támogatott kutatásokban játszik nagy szerepet. A Robotino® mozgatásának alapját az úgynevezett omnidrive teszi lehetővé, mely a robot három különálló 120° szögben elhelyezett omni kerekét, az omnidrive egység által szabályozott DC motor segítségével hajtja. Ennek előnye, hogy a robot, tetszőleges kezdőpontból tetszőleges pályán képes mozogni, akár egyhelyben forogni is. Az omnidrive-nak három bemenete van: V_x [mm/s] (float) sebesség, V_y [mm/s] (float) sebesség és a szögelfordulási sebesség – omega [deg/s] (float) és három kimenete, mely az egyes motorok sebességeit határozza meg. A motorok sebességét szabályozó blokk PID szabályozás segítségével szabályozza az egyes motorok gyorsulását, fékezését.



3. ábra: A Robotino® mobil robot

3.5. Robotino® View

A cikkben a robot alkalmazása mellett szól az is, hogy a Festo olyan programozási környezetet biztosít a robothoz, amely felhasználóbarát GUI-val rendelkezik. A Robotino® View egy interaktív „natív” grafikai programozásra szolgáló környezet. A program ingyenesen letölthető a Festo Didactic oldaláról. A funkcióblokkos fejlesztői környezetben vezérlési-, képfeldolgozó-, logikai-, vektoralgebrai stb. objektumok állnak rendelkezésre és ezek segítségével összetett vezérlés valósítható meg. A Robotino® View programban egyaránt rendelkezésre állnak képfeldolgozást segítő blokkok, oszcilloszkóp az érzékelt jelek időbeli változásának megfigyelésére és az irányításához szükséges objektumok is. A

fejlesztőkörnyezet lehetőséget biztosít külső programból történő adatkapcsolat létrehozására TCP/IP és UDP kapcsolaton keresztül.

3.6. *Robotino® Sim*

A Robotino® Sim egy robot szimulációs szoftver, amely virtuális 3D-s környezetben valósághűen modellezi le a robot mozgását, mely a fejlesztés során nagyon hasznos. A Robotino® szimulációs modell tartalmazza mindazt, amit egy valós robot is. Ahhoz, hogy megfelelően tudjuk használni a szoftvert szükségünk, van Windows 2000, XP, Vista vagy Windows 7 operációs rendszerre, illetve legalább egy 128 MB RAM-mal rendelkező OpenGL támogatásos videokártyára. Ennek a követelménynek az említett Acer laptop megfelel.

4. Adatfeldolgozó és vezérlő egység megvalósítása

A feldolgozott adatok segítségével valamilyen vezérlési funkció megvalósítására többféle beágyazott rendszer megoldás alkalmazható. A definiált feladatok megoldásához C# programozási nyelv került alkalmazásra.

4.1. *Célkitűzés és tervezés*

Az adatfeldolgozó és vezérlő egységgel szemben az alábbi követelmények kerültek megfogalmazásra:

- fogadni tudja a Leap Motion kontroller által USB-n alapuló kommunikáción küldött adatokat;
- a kiválasztott hardverek vezérléséhez szükséges adatkapcsolatot meg tudja valósítani.

4.2. *Adatfeldolgozó és virtuális mobilrobot vezérlő program*

A projekt létrehozása során szükséges volt pár fontosabb dolog beállítására. Először is referenciaként hozzá kellett adni a LeapCSharp.NET4.5.dll fájlt. Ebből a .dll állományból különböző verziók is léteznek, de ki kell választani a projektben használt .NET verziónak megfelelőt. Miután ez megvan meg kell határozni, hogy x86-os vagy x64-es architektúrájú platformra fejlesztünk. Végezetül egy „Post Build” eseményt kell létrehoznunk, ahol a natív könyvtárak átmásolódnak a projekt könyvtárában.

A forráskód négy osztályból épül fel: Az egyik a MainUI nevet kapta, amely lényegében az adatmegjelenítést egészét takarja. A második és a harmadik osztály kinézetben elég hasonló,

viszont az egyik az egér alapú mozgás irányítást valósítja meg ez a MouseContrUI nevet kapta. A másik a kézmozgás alapú irányítást valósítja meg ez pedig MotionContrUI nevet kapta. A negyedik osztály RobotinoUDPS nevet kapta. Feladata, hogy a Robotino® View programnak a megfelelő kommunikációs protokollnak megfelelően adatokat küldjön.

4.2.1. Az adatmegjelenítést megvalósító függvények

A MainUI osztály feladata a felhasználó által kiválasztott események lekezelése. Ezek az események fogják majd az adatfeldolgozó és megjelenítő függvényeket meghívni. A MotionContrUI és a MouseContrUI osztály feladata, hogy a megjelenített irányító felületen továbbítsa a virtuális robot számára szükséges adatokat.

4.2.2. Az adatfeldolgozást megvalósító metódusok

Az adatok feldolgozásáról mindkét osztályban külön metódusok gondoskodnak. Az alábbi programrészletekben bemutatásra kerül, hogy az adott irányításnál miként történik a beolvasás és a továbbítás a virtuális robot felé.

```
private void Mouse_Panel_MouseMove(object sender,
MouseEventArgs e)
{
    int cursorX = (e.X - 200) * -1;
    int cursorY = (e.Y - 200) * -1;
    udpconn.DataToRV((byte)cursorY, (byte)cursorX);
    int x = (cursorX - 100) * -1;
    int y = (cursorY - 100);
    Text_Box_X.Text = Convert.ToString("X: "+x);
    Text_Box_Y.Text = Convert.ToString("Y: "+y);
}
```

A MouseEventArgs egy olyan osztály, ami rendelkezik 6 tulajdonsággal (property) ezek a következők:

- Button;
- Clicks;
- Delta;
- Location;
- X;
- Y;

Ezekből, ami fontos volt, azaz utolsó kettő. Az X és Y egy szám értéket ad, amit az egéresemény során generál. A programrészleten is jól látható, hogy egy panelon lévő egér mozgás kerül megfigyelésre. Ebből az következik, hogy a panel jobb felső sarkánál van a 0;0 koordináta, ami a kutatás céljából egyáltalán nem volt megfelelő, ezért koordináta eltolást kellett beépíteni. Mivel a panel egy 200x200-as méret azért, elég lett volna csak 100-al eltolni a koordinátát, de az sem volt megfelelő, mert a Robotino View számára byte-ba kerül átküldésre, tehát az értékeket 0-255-ig értelmezi. A -1-el való szorzás azért kellett, mert a kivonás miatt negatív értékeket kapunk, amit át kellett számítani pozitívba. Miután a számítás sikerült a mért koordinátákat elküldi a Robotino View felé UDP kapcsolaton keresztül. A felhasználónak könnyen értelmezhető adatok kerültek megjelenítésre, ezért a kurzorból érkező koordináták úgy kerültek kiszámolásra, hogy ha az egér a koordináta rendszer közepén van, akkor legyen 0:0 az X és Y érték. Az X értéket vissza kellett szorozni mínusz 1-el, hogy az irányok megfelelőek legyenek a robot mozgáshoz képest.

```
public void HandTrack(object sender,
FrameEventArgs eventArgs)
{
    Frame frame = eventArgs.frame;
    if (frame.Hands.Count > 0)
    {
        Hand hand = frame.Hands[0];
        Vector position = hand.PalmPosition;
        float x = (hand.PalmPosition.x);
        float z = (hand.PalmPosition.z);

        int tx = Convert.ToInt32(x) -100;
        int tz = Convert.ToInt32(z)*-1 +100;

        Text_Box_X.Text = Convert.ToString("X: " + tx);
        Text_Box_Y.Text = Convert.ToString("Y: " + tz);

        radioButton1.Location = new Point((int)x, (int)z);

        udpconn.DataToRV((byte)tz, (byte)tx);
    }
}
```

A FrameEventArgs ami rendelkezik egy tulajdonsággal (property) ez pedig:

- frame

A frame osztályból a legfontosabb funkció a kéz azonosítását valósítja meg. A programrészletben megvizsgálásra kerül, hogy az eszköz egyáltalán lát-e érzékelhető kezét, mert ha nem akkor ne számoljon feleslegesen. Viszont, ha lát kezét akár többet is akkor is csak az elsőt fogja követni. A Vector-struktúra egy háromkomponensű matematikai vektort vagy pontot jelent, például egy irányt vagy pozíciót a háromdimenziós térben. Ebből az következik, hogy a robot mozgásához ebből a 3 vektorból kettőt fel kellett használni. A kiszámolt számok átadásra kerülnek egy gombnak, mivel itt a kurzoros megoldás nem volt megoldható, hogy szépen látható legyen a panelon a mozgás. Miután a számítás sikerült a mért koordinátákat elküldi a Robotino View felé UDP kapcsolaton keresztül. A felhasználó számára értelmezhető adatok kerültek megjelenítésre.

4.2.3. Virtuális robot mozgásvezérlésének megvalósítása

A Robotino® virtuális mobil robot mozgásvezérlését UDP kapcsolaton keresztül valósítottam meg, melyet a Robotino® View fejlesztőkörnyezet támogat. A robot mozgásvezérlését az X, Z koordinátából származó értékek felhasználásával végeztem. Az értékeket másodpercenként UDP kapcsolaton keresztül kerültek átküldésre a Robotino® View UDP szerverre, számára a localhost 127.0.0.1-es IP címre a 9180-as portra. A Robotino® View-ban az UDP szerver létrehozása és futtatása könnyen megvalósítható a fejlesztőkörnyezetben elérhető Data Exchange UDP funkcióblokk által. A Robotino® View-ban megvalósított egyszerű program a robot mozgási sebességének számítását funkcióblokkokkal valósítja meg. A program végtelen ciklusban fut addig, amíg a Stop gombbal a futást le nem állítódik. A Robotino View a Robotino® vezérlését WiFi kapcsolaton keresztül TCP/IP protokollal valósítja meg a robot IP címének megadásával. A Robotino mozgását a sajátgépen futó Robotino® Sim szimulációs környezetében is meg tudjuk figyelni, ami a 127.0.0.1 címen és a 8080-as UDP porton keresztül volt elérhető. A Robotino® View Robotino SIM-hez történő kapcsolódása után a robot virtuális térben történő mozgása követhető.

A DataToRV függvény a Robotino® View-val történő adatkapcsolat létrehozása. A függvény első része létrehozza az UDP kapcsolatot, majd összeállítja a Robotino® View kommunikációs protokolljának megfelelően az adatsomagot, amit egy checksum ellenőrzést követően elküld a Robotino® View UDP szerver számára.

```
public void DataToRV(byte a, byte b)
{
```

```
byte checksum;
byte [] data = new byte[36];
byte s0 = 0;

try
{
    UdpClient udpClient = new UdpClient(9181);

    udpClient.Connect("127.0.0.1", 9180);

    data[0] = 0;
    data[1] = 36;
    data[2] = 0;
    data[3] = 0;

    data[4] = a;
    data[5] = 0;
    data[6] = 0;
    data[7] = 0;

    data[8] = b;
    data[9] = 0;
    data[10] = 0;
    data[11] = 0;

    for (int i = 0; i < 36; ++i)
        s0 += data[i];
    checksum = (byte)(0xFF - s0);
    data[3] = checksum;
    udpClient.Send(data, 36);
    udpClient.Close();
}
catch (Exception)
{
    MessageBox.Show("Hiba a robot mozgatása során", "Figyelmeztetés!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}
```

A koordináták továbbítására az UDP protokoll került kiválasztásra, mivel azt is támogatja a Robotino View. A programkód elején a szükséges változók létrehozásra kerültek, majd a kapcsolat kiépítése következett a program és a Robotino View között, ha ez nem sikerült, akkor a hibát lekezelve hibaüzenet jelent meg. Viszont, ha sikerült akkor az adatok elküldésre kerültek. Az adatszerkezet előre specifikálva van így azt értelmezni és használni kellett. A protokoll a következőket követeli meg:

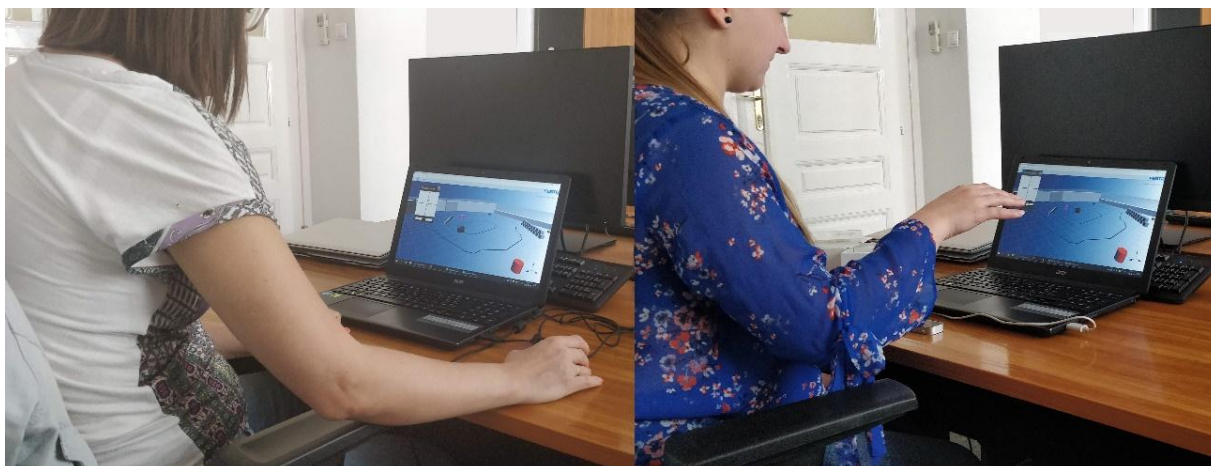
1. táblázat: Protokoll adatstruktúra

Byte	Funkció
0	Üzenetazonosító
1-2	Teljes N üzenet bájttjainak száma
3	Ellenőrző összeg
N-1	Üzenet utolsó bájttja

Mivel két adatot továbbítunk így 4-es és 8-as byte-ot fogjuk használni. Ha az üzenet rövidebb, mint 100 bájtt, akkor az s0 összeg az egész csomagból kerül kiszámításra egyetlen bájtt. Ha az üzenet 100 vagy több bájttot tartalmaz, az s0 az üzenetből kerül kiszámításra az első és az utolsó 50 bájtt. Mindkét esetben az ellenőrző összeg bájttját 0-val kell inicializálni.

5. Esetvizsgálatok, eredmények, felhasználói tapasztalatok

Az előző fejezetben ismertetett a Leap Motion eszközzel megvalósított adatfeldolgozó és vezérlő egység segítségével tesztalanyokon több esetvizsgálat került elvégzésre a Leap Motion robot vezérlés mind az egér alapú vezérlésével kapcsolatban. A tesztek során mind férfi mind női tesztalanyok közreműködtek. Mivel egy ember általi vezérlésű eszközökről van szó, ezért a használhatóságot jellemző értékelést csakis maguk a tesztalanyok adhatják, tehát az ő közreműködésükkel lehet az eszközök működését jellemző fontosabb paramétereket esetvizsgálatokkal meghatározni. Az első vizsgálandó szempont a kialakított rendszer használhatósága volt, ahol a tesztelőnek a véleménye került kiértékelésre egy 1-5-ig terjedő skálán. A másik szempont a virtuális mobil robot mozgásának pontossága, tehát a virtuális tesztpályán hányszor tért le a kijelölt útról vagy hányszor kezdett el forogni a kijelölt útvonalon. Ebben az esetben a sebesség, idő stb. csak másodlagos paraméterek.



4. ábra: Egér, illetve kézmozgás vezérlés tesztelése

A virtuális mobil robot mozgatásához, a virtuális környezet generál egy elég egyszerűnek mondható tesztpályát, ahol több pálya is található. Az első pálya egy ovális pálya, ahol körbe kell menni, tehát a kezdeti és a befejezési pont ugyan ott található. Nagyon figyelmesnek kell lennie, hogy tudja honnan indult el, mivel ugyan oda kell, visszaérkezzen, mert az is hibapontnak számít. A második pályarésznél egy kicsit összetettebb pályán kell körbe mennie. A kezdeti és a végpont ebben az esetben is ugyan ott található. Első körben mindkét pályarészt egyszer csinálják végig, amiből a kezdeti értékelést el tudjuk végezni.

A használhatósággal kapcsolatban alkotott kezdeti vélemények és tesztek alapján kapott eredmények nagyon megoszlottak, mivel a rendszer első alkalommal történő használata nem hozta számomra a kívánt eredményt. Az eszközök használata során egyes tesztalanyokat megzavarta az egér érzékenysége is, mivel nem ilyenekhez vannak szokva. Esetenként az is zavaró tényezőnek számított, hogy a robot szembe jövetelekor megfordul az irányítás, mintha egy kocsival tolatnánk. Tehát ha balra forgatja a kezét, akkor jobbra megy és ugyan ez fordítva. Érdekes volt, hogy jobban sikerültek az első körös kézmozgás vezérlés, mint az egerrel történő vezérlés.

Ezután az eszköz használatával kapcsolatosan összegyűjtött tapasztalataimat is átadva számukra, tanácsokkal ellátva őket arról, hogyan próbálják a kezüket mozgatni. A tesztelők az elején nem tudták, hogy a tesztpálya nézetét testre lehet szabni, ezzel megkönnyítve a tesztelésüket. Az eszköz használatával eltöltött idővel fokozatosan javult az eszköz használhatóságáról alkotott véleményük és a kívánt mozgási feladat elvégzésnek sikeressége is. A tesztalanyok által a használhatóságról adott értékelés, valamint a kísérletek alapján végzett tesztek eredményeit a 2. és a 3. táblázat foglalja össze.

2. táblázat: Virtuális robot vezérlése egér mozgatóssal

	Tesztalanyok	Kezdeti értékelés		Betanulás utáni értékelés			
		Használhatóság	Hibapontok / pálya		Használhatóság	Hibapontok / pálya	
			1.	2.		1.	2.
Fiatalok	1.	2	5	10	4	3	5
	2.	3	2	4	4	0	0
Idősek	1.	4	8	12	5	2	4
	2.	3	10	15	5	3	5
	Átlag	3	6,25	10,25	4,5	2	3,5

3. táblázat: Virtuális robot vezérlése kéz mozgatóssal

	Tesztalanyok	Kezdeti értékelés		Betanulás utáni értékelés			
		Használhatóság	Hibapontok / pálya		Használhatóság	Hibapontok / pálya	
			1.	2.		1.	2.
Fiatalok	1.	2	1	3	4	0	0
	2.	3	2	1	4	0	0
Idősek	1.	4	3	8	5	0	2
	2.	3	4	7	5	2	3
	Átlag	3	2,5	4,75	4,5	0,5	1,25

A kapott eredmények alapján is látható, hogy a tesztalanyok az eszközök használatának első lépései során nehezen boldogultak az egér megfelelő használatával. Robotmozgás esetén a tesztalanyoknak nehézséget okozott a robot megállítása, vagyis az irányító felület középpontját nehezen találták el. Ennek következtében előfordult, hogy a robot elindult a virtuális tesztpályán vagy esetleg egyhelyben forgott. Az kézmozgás vezérlés során jól kimutatható, hogy már a használatának első lépései során a felhasználó könnyen elsajátította a megfelelő használatát. Ez annak tudható be, hogy manapság az emberek többsége sűrűn használja a telefonját vagy esetleg touchpad-et.

Az eszközök több mint fél óra használata után már sokat javult a használatának eredményessége, melyet a táblázatban látható eredmények is mutatnak. A tesztalanyok által az eszköz használhatóságáról adott vélemények több mint 1,5 osztályzattal lettek jobbák és elérték a jó értéket.

A tesztalanyokkal a kísérletek elvégzése után elbeszélgettem, hisz e technológia alkalmazásának tapasztalatiról és továbbfejlesztési irányaihoz kapcsolódó információkról. Maguk a tesztalanyok tapasztalatai sok ismerettel tudnak szolgálni. Az elbeszélgetések során mindannyian kiemelték, hogy a kézzel sokkal hamarabb ráéreztek az irányításra. Könnyebben tudtak finomabb mozgást kivitelezni, mint az egérrel. Egyes tesztalanyok kiemelték, hogy az elején nehézkesen értelmezték a koordináta-rendszer és a robot mozgását. Nem tudták úgy

összehangolni a mozgatót és a vizuális látványt, de ez használat során nem volt tapasztalható. Az eszközök használatának eredményessége nagyban javult, egyre több sikerélménybe volt részük ezáltal egyre szívesebben használták az eszközt, és egyre jobb véleménnyel voltak az eszköz használhatóságát illetően is.

A tesztelések zárásaként, megkértem a tesztalanyokat, hogy csak a kézmozgás vezérléssel, próbáljuk ki, hogy mi van akkor, ha a rosszabbik kezükkel csinálják. A 4. táblázaton is jól látható a tesztelés eredménye.

4. táblázat: Virtuális robot vezérlése kézmozgatással, rosszabbik kézzel

	Tesztalanyok	Hibapontok / pálya	
		1.	2.
Fiatalok	1.	5	10
	2.	2	4
Idősek	1.	4	6
	2.	2	2
	Átlag	3,25	5,5

Ez a teszt szimplán csak érdekességként gondoltam, mivel egyáltalán nem volt kikötés, hogy melyik kezüket használhatják, de mindenki a jobbik kezét választotta. Az eredményen is jól látszik, hogy azért egy elég hosszú tesztelés után miután már mindenki ráérezett a robot mozgására utána a rosszabbik kezükkel még se tudtak olyan jó eredményeket elérni.

6. Hipotézisek értékelése

A készített esetvizsgálat és a felhasználói tapasztalatok alapján a következőket lehet levonni.

Az 1. számú hipotézis: Akkor hatékonyabb a kézmozgással vezérelt robot mozgása, ha egy megközelítő eredményt ér el a hagyományos egér alapú irányítással. Ez a hipotézis beigazolódott, mivel kiderült, hogy sokkal jobban vezérlik a kezükkel az emberek, mint egy egérrel. Sőt nem csak megközelítő eredmények születtek, hanem még túl is szárnyalta az elvárásaimat ezzel kapcsolatban.

A második hipotézis, miszerint a fiatalok könnyebben tudják elsajátítani a kézmozgással vezérelt robotot, mint az idősebbek ez csak részben igazolódott be, mert az idősebbek is ugyan olyan könnyen vették a kézmozgással történő akadályokat, mint a fiatalok.

A 3. számú hipotézisem nem igazolódott be, amely a következő volt: Összehasonlítás során többen maradnának a hagyományos egér alap irányítás mellett. A felhasználóktól azt a választ

kaptam, hogy szívesen használná munkája során az eszközt egér mozgatásra is. Elég megterhelő tud lenni az ember számára a kézkitartás.

A negyedik hipotézis, amelyben a következőt állítom: A jobbik kéz irányításától nem fog sokban eltérni a rosszabbik kéz irányítása ez nem igazolódott be, mivel ahhoz képest, hogy előtte mennyit teszteltek többet hibáztak a rosszabbik kezükkel, mint a jobbikkal. Kimutatható az eltérés nagy mértékben a kettő kéz között.

7. Összefoglalás

A legkorszerűbb pedagógiai irányzatok közé tartozó projekt alapú oktatással lehetőségem nyílt az autonóm tanulásra és a mentor tanárommal való hatékony együttműködésre és a cikkben bemutatott rendszer megvalósítása során ezek kulcsfontosságúnak bizonyultak. A projekt során kézmozgás érzékelés felhasználásával mobil robot sebességvezérlése valósul meg virtuális térben, mely jó tesztkörnyezetet ad az ember általi irányítás vizsgálatához. Az esetvizsgálati eredményeket tartalmazó fejezetben leírtaknak megfelelően látható, hogy az eszköz megfelelő betanulás nélkül még nem használható kielégítően, de a használatának elsajátítása után már jól használható alkalmazások születtek, mellyel a tesztalanyok szívesen kísérleteztek. Az eredmények alapján elmondható, hogy a fejlesztett rendszer – a felhasználói tapasztalatok, tesztek által bizonyítottan, akár a jövő virtuális laboratóriumában végzett feladat végrehajtás során is alkalmazható, amely a virtuális egyetem koncepció irányába is mutathat.

Irodalomjegyzék

Kaur, H., & Rani, J. (2016). A review: Study of various techniques of Hand gesture recognition. In *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)* (pp. 1-5). IEEE.

Sonkusare, J. S., Chopade, N. B., Sor, R., & Tade, S. L. (2015). A review on hand gesture recognition system. In *2015 International Conference on Computing Communication Control and Automation* (pp. 790-794). IEEE.

Immanuel, I. (2002). Collaborative Internet project management in education. In *32nd Annual Frontiers in Education* (Vol. 1, pp. T1E-T1E). IEEE.

Eguchi, Y., Ando, Y., & Mizukawa, M. (2007). Development of management system in mechatronics education applying project management method. In *SICE Annual Conference 2007* (pp. 552-557). IEEE.

Katona, J., et al. (2016). Speed control of Festo Robotino mobile robot using NeuroSky MindWave EEG headset based brain-computer interface. In *2016 7th IEEE international conference on cognitive infocommunications (CogInfoCom)* (pp. 000251-000256). IEEE.

Katona, J., et al. (2019). Electroencephalogram-based brain-computer interface for internet of robotic things. In *Cognitive Infocommunications, Theory and Applications* (pp. 253-275). Springer, Cham.

Hartanto, R., Susanto, A., & Santosa, P. I. (2014). Real time hand gesture movements tracking and recognizing system. In *2014 Electrical Power, Electronics, Communications, Control and Informatics Seminar (EECCIS)* (pp. 137-141). IEEE.

Sziladi, G., et al. (2017). The analysis of hand gesture based cursor position control during solve an IT related task. In *2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)* (pp. 000413-000418). IEEE.

Wachs, J. P., Stern, H., & Edan, Y. (2005). Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 35(6), 932-944.

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S., & Harvill, Y. (1986). A hand gesture interface device. *ACM SIGCHI Bulletin*, 18(4), 189-192.

Zhang, Y., & Yang, B. (2011). A mode of ability-targeted computer curriculum reform based on projects in engineering education. In *2011 6th International Conference on Computer Science & Education (ICCSE)* (pp. 154-157). IEEE.