ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar

REHABILITÁCIÓS GÉP TERVEZÉSE ÉS MEGVALÓSÍTÁSA

OE-BGK
2022.

Hallgató neve: **Turner Máté**
Hallgató törzskönyvi száma: **T009500/FI12904/B**

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar
Anyag- és Gyártástudományi Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Turner Máté
Szakdolgozat száma: SZD22052513059164
Törzskönyvi száma: T009500/FI12904/B **Neptun kódja:** [redacted]

Szak: gépészmérnöki
Specializáció: gépészmérnöki - CAD-CAM-CNC

A dolgozat címe: Rehabilitációs gép tervezése és megvalósítása
A dolgozat címe angolul: Design of a rehabilitation machine
A feladat részletezése:

1. Végezzen kutatást a szakirodalomban a megoldandó feladattal kapcsolatban!

2. Tervezzen meg egy emberi kéz rehabilitására szolgáló gépet!

2. Építse meg a rehabilitációs gép csukló mozgatásáért felelős részét!

3. Tervezze meg a rendszer vezérlését!

4. Végezzen méréseket a csukló mozgástartományaiban és értékelje az eredményt!

5. Írjon programot a rehabilitációs mozgatásokat vezérlő gcode-ok előállítására!

6. Foglalja össze az eredményeket, fogalmazzon meg további fejlesztési javaslatokat!

Intézményi konzulens neve: Dr. Czifra György

Külső konzulens neve: Kovács Attila
Munkahelye: M-Technologies KFT

A kiadott téma elévülési határideje: 2024. 05. 15.
Beadási határidő: 2022. 12. 15.

A szakdolgozat: Nem titkos.

Kiadva: Budapest, 2022. 10. 21.

ÓBUDAI EGYETEM Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar Anyag- és Gyártástudományi Intézet P.H. 2.

.....
Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

Dr. Czifra György
.....
belső konzulens

.....
külső konzulens

Digitálisan aláírta: Dr. Czifra György
DN: cn=Dr. Czifra György, o=BGK AGI, ou=Óbudai Egyetem, BGK, email=czifra.gyorgy@bgk.uni-obuda.hu, c=HU
Dátum: 2022.12.06 20:00:44 +01'00'

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat/diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatban/diplomamunkában található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Budapest, 2022.12.15.

.....
hallgató aláírása

Tartalomjegyzék

1. Köszönetnyilvánítás 4
2. Bevezetés 5
   2.1 A feladat ismertetése 6
3. Szakirodalmi háttér 7
   3.1 Az akut agyi érkatasztrófa (stroke) 7
   3.2 A kéz alapvető anatómiája, mozgásának működése 10
   3.3 3D nyomtatás és 3D szkennelés az egészségügyben 13
   3.4 A léptetőmotorok és vezérlésük 17
   3.5 Programozási nyelvek 20
4. A csukló általi gépi főtengelyek meghatározása 21
5. A gép mozgatóelemeinek kiválasztása 23
6. A kéz 3D szkennelése 28
7. A kéz rögzítésének és a gép mozgatási pályáinak meghatározása 30
8. A gép tervezésének véglegesítése 34
9. A gép alkatrészeinek gyártása és összeszerelése 37
10. A vezérlés megépítése és üzembehelyezése 40
11. A printer.cfg konfigurációs fájl létrehozása 44
12. Tesztmozgások és léptetőmotor kalibrálások 45
13. Alap csuklómozgatási program megírása programmal 46
14. Összefoglalás 47
15. Summary 48
16. Irodalomjegyzék 49
17. Mellékletek listája 52

1. Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani feleségemnek Turner-Kiri Mónikának a szakadatlan támogatásáért, megértéséért, minden bátorító gondolatáért, szakdolgozatom lektorálásáért. Valamint, hogy kitartott mellettem az utóbbi évek alatt, amikor oly sok közös szabadidőt áldoztunk az egyetemi tanulmányaim előrehaladásába.

Köszönettel tartozom Kuznyecov Gábornak. Szakmai és emberi támogatása nélkül ez a szakdolgozat nem jöhetett volna létre.

Köszönöm Dr. Czifra Györgynek, hogy vállalta témám vezetését és egyengette utam a beszámolóm megírása során.

Köszönöm Dr. Bajzik Éva doktornőnek, hogy útbaigazított az anatómia világában és hogy bátorított, támogatott célom megvalósításában.

Külön köszönöm az egész családomnak, hogy mindig számíthattam rájuk és hogy lehetővé tették számomra, hogy ideáig eljussak.

Tanulmányaim során nagy hatással voltak rám barátaim szavai és tettei, melyek ösztönöztek az elmúlt évek alatt. Ezért köszönettel tartozom Juhász Ádámnak, Makay Zsoltnak, Timár Zoltánnak és Major Dávidnak.

Végül, de nem utolsó sorban azoknak a szaktársaknak szeretnék köszönetet mondani, akik személyében az egyetemi évek alatt barátokra leletem. Köszönöm az elmúlt évek közös munkáit és hogy annyi mindent tanulhattam tőletek: Darázsi János, Jelinkó Tibor, Kernbaum Márk.

„Nagy lépéseket általában akkor tehetünk, ha korábban szétválasztott tudományterületeket elegyítünk.”

(Katie Patrick)

2. Bevezetés

Napjainkban az orvostudományban alkalmazott gépekre fokozottabb szükség van. Egyre több betegséget, baleset általi sérülést gépekkel kezelnek, gyógyítanak. Vannak olyan megbetegedések, betegségből származó mellékhatások vagy balesetből eredő sérülések melyeket gépi beavatkozások nélkül nem is lehet kezelni. A sérülésekből, bénulásokból hosszú rehabilitációs folyamat vezet a felépüléshez, sokszor nem is lehetséges a 100%-os regenerálódás.

Az akut agyi érkatasztrófa (stroke) következményeként fellépő bénulással és részleges bénulással küzdő emberek segítése motivált a munkám során, hogy tervezzek egy kézrehabilitációs gépet. Fontos, hogy az ilyen betegségben szenvedők minél előbb újra vissza tudjanak illeszkedni a mindennapi életükbe, tudjanak dolgozni, de a legfontosabb, hogy a mindennapi élethez szüksége mozgásokat el tudják végezni és ezzel önállóak és önellátóak tudjanak lenni.

A számottevő stroke-os betegek mozgásrehabilitációja nagyon leterheli a gyógytornászok és rehabilitációs orvosok kapacitását. Ennek hatására a korszerű rehabilitációs gépek iránti igény rohamos tempóban növekszik, hiszen ezekkel gyorsítani és pontosítani lehet a terápiás eljárásokat. Kutatást végeztem a jelenleg alkalmazásban lévő kézrehabilitációs gépek terén, hogy milyen technológiával és elven működnek. Arra a következtetésre jutottam, hogy bonyolultságából és lehetséges befoglaló méretek miatt nincs olyan gép, amely foglalkozik a kéz csukló és ujjak rehabilitálásával egyaránt.

Ezért célkitűzésem egy olyan rehabilitációs gép prototípusának megtervezése, amely képes a beteg bénult kezét, csuklótól az ujjak végéig rehabilitációs mozgatásokkal kezelni, valamint a csukló mozgatásáért felelős részegységet megépíteni. A stroke-ot kapott emberek keze képes ideggörcs által fixálni izmokat egy pozícióba. Nem tudják mozgatni a kezüket, de hosszú, megtervezett, passzív rehabilitációs tornáztatást követően a beteg elkezdi aktívan mozgatni ujjait, csuklóját. A gép melyet létrehozok alkalmas lehet egy korai passzív tornáztatásra és a későbbiekben pedig az aktív rehabilitációban is alkalmazható lehet.

Mivel a projektet önfinanszírozás keretein belül hozom létre, ezért törekszem arra, hogy költséghatékony legyen. Ez azzal jár, hogy felhasználok, mindent magam körül, ami alkalmas a gép létrehozásához, de ezek nem az optimális megvalósítások.

2.1 A feladat ismertetése

Feladatom elején ismertetem az akut agyi érkatasztrófának (stroke) nevezett betegséget, az ezzel járó kézbetegségeket és mozgásszervi hiányosságokat.
Továbbá összegzem röviden a kéz alapvető anatómiai felépítését, működését, funkcióit, melyek elengedhetetlenek a rehabilitációs gép megtervezéséhez és megépítéséhez. Kitérek az agyi-ideg-izom-mozgás kapcsolatokra, kihangsúlyozva azokat a részeket, amelyeket a szakdolgozatomhoz célirányosan gyűjtöttem össze.
A következő feladatokat kell elvégeznem:

- Végezzen kutatást a szakirodalomban a megoldandó feladattal kapcsolatban!
- Tervezzen meg egy emberi kéz rehabilitálásra szolgáló gépet!
- Építse meg a rehabilitációs gép csuklómozgatásáért felelős részét!
- Tervezze meg a rendszer vezérlését!
- Végezzen méréseket a csukló mozgástartományaiban és értékelje az eredményt!
- Írjon programot a rehabilitációs mozgatásokat vezérlő gcode-ok előállítására!
- Foglalja össze az eredményeket, fogalmazzon meg további fejlesztési javaslatokat!

3. Szakirodalmi háttér

3.1 Az akut agyi érkatasztrófa (stroke)

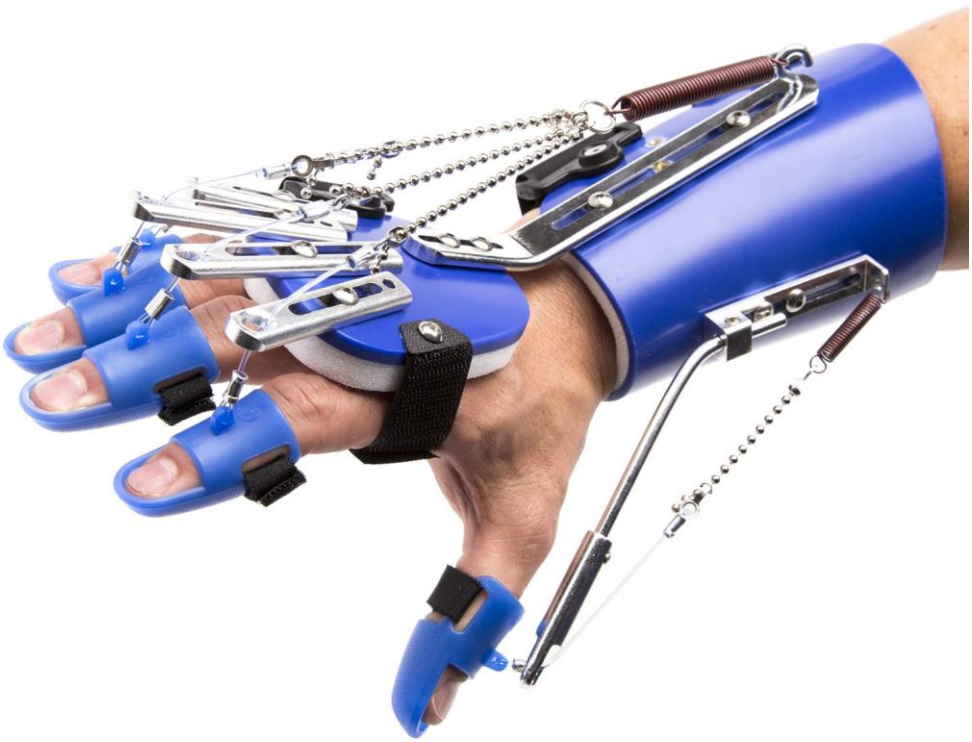
Az irodalomkutatás a szakdolgozattal szorosan összefügg, ezért ismertetnem kell a projektemhez szükséges részegységek alapvető ismereteit. A munkámat az ehhez szükséges információk felkutatásával kezdtem, melyet a következőkben ismertetek. A szakdolgozatban a forrásmegjelölés nélkül használt képek saját képek, minden más képnél a forrásokat megjelöltem.

A stroke (magyar köznyelvben gutaütés, szélütés) egy hirtelen fellépő agyi érelzáródási vagy érrepedési bevérzés miatt következik be. Ilyenkor az agy bizonyos része nem kap megfelelő vérellátást és ezzel az agy oxigénellátása megszakad. Ez bárkivel és bármikor előfordulhat, kortól és nemtől függetlenül. Magyarországon éves szinten megközelítőleg 40-50 ezer ember kap stroke-ot. [1]

„A stroke kialakulásában számos rizikófaktor játszik kiemelkedő szerepet: a magas vérnyomás, a szívbetegségek, a dohányzás, a kettes típusú cukorbetegség, a túlzott alkoholfogyasztás, a túlsúly és a mozgásszegény életmód mind elősegítik az agyi érkatasztrófa megjelenésének lehetőségét.” [2]

1. ábra. Agyi sérülés a testben, ellentétes oldalon vált ki hatást [3]

A stroke azonnali tünetei a száj félrehúzódása, zavaros beszéd, végtaggyengeségek. Tipikus esete a stroke-nak, hogy az agy egyik felében fellépő sérülés a test ellentétes oldalán okoz bénulásokat, végtagelégtelenségeket. (1. ábra.) Az agyi érkatasztrófa sérüléseit két nagy kategóriába soroljuk. Az egyik az elsődleges sérülések, amelyek közvetlenül az érelzáródás után következnek be. Ennek kezelése szinte lehetetlen. A másik a másodlagos, melyek hosszan elhúzódnak órákra, napokra, hetekre, hónapokra az eset bekövetkezése után. Itt fennáll a hosszú terápiás ablak lehetősége, hiszen ezeket az időszakosan maradandó sérüléseket már hosszabb távon lehet kezelni. A stroke után elengedhetetlenül fontos feltérképezni az agyszövetben keletkezett károkat, így pontos képet kapnak arról, hogy melyek azok a folyamatok, amelyeket meg kell állítani ahhoz, hogy az idegszövetekben ne keletkezzenek további sérülések és a megfelelő kezelést lehessen alkalmazni. [1]



2. ábra. Kézrehabilitációs eszköz [4]

A stroke utáni bénulással járó következményeket a végtagokon rehabilitációs orvosok végzik. Alapvetően kétfelé kell bontani az ilyen kezeléseket. Az első az aktív rehabilitáció, amikor a páciens a kezét valamilyen szinten tudja mozgatni, erőt tud vele kifejteni. Ebben az esetben olyan feladatok elé állítják a beteget, amikor neki kell egy adott tárgyat mozgatnia vagy arrébb tennie, például kis labdát, pénzérmét egy megadott helyre vagy pozícióba kell mozgatnia. A kezelések előrehaladásával egyre összetettebb mozgásokat igénylő feladatot kap, például egy vizes palackból pohárba kell töltenie vizet. A második kezelési forma a passzív rehabilitáció, amikor a páciens még nem tudja mozgatni a kezét és az ujjait. Ilyenkor a kezelőorvosok kézzel, speciális tornáztatásokat hajtanak végre, hogy az ernyedt vagy épp görcsben álló izmokat stimulálják.

Egyre több orvos és cég használ ezekre a folyamatokra robotrehabilitációs gépeket. Az ilyen rehabilitálások az agyban sérült vagy elpusztult idegsejteknek segítenek újra működésbe lépni. Ahol ezek az idegsejtek és kapcsolataik megszűntek, ott a végtagok mozgásához szükséges idegi kapcsolatokat létesítik újra. A robotok által végzett rehabilitáció abban könnyíti meg a rehabilitációs kezelőorvos segítségét, hogy egy-egy mozgáskombinációt pontosabban, gyorsabban, magas ismétlésszámmal tudnak végrehajtani, melyek együttese hozzájárul a motorikus fejlődéshez. Emellett pontos adatokat kapnak arról, hogy mik a páciens mozgásainak jelenlegi hiányosságai és melyek a már elért eredményei.

„Számos tanulmány alátámasztja, hogy a robottechnológia és a hagyományos gyógytorna egyesítése kedvező eredményeket mutat a rehabilitáció folyamán” [5]

3.2 A kéz alapvető anatómiája, mozgásának működése

A kéz rehabilitációjához szükséges gép megtervezéséhez és kivitelezéséhez elengedhetetlen az alapvető anatómia ismerete.

A kéz mozgása során a mozgásokba bekapcsolódik a csukló, az alkar, a könyök, a felkar és a váll is. A vállat a kulcscsont rögzíti a lapockához és a szegycsonthoz. A felkar egyetlen csontból áll mely közvetlenül kapcsolódik a vállízületbe. A felkar alsó részén kapcsolódnak az alkar csontjai az singcsont és az orsócsont és az alkar feszítő és hajlító izmai. A két alkari csont ízületesen kapcsolódik egymáshoz és a köztük lévő üres teret inas hártya tölti ki. Az alkar és a felkar csontjai közösen alkotják a könyökízületet. A könyökben a singcsont hengeres ízülettel, míg az orsócsont gömbízülettel kapcsolódik. Továbbá a két alkarcsont egymáshoz forgóízületben végződik. A könyök kétféle mozgást képes produkálni. Hajlítás-feszítést, vagy forgó mozgatást. Amikor a forgó mozgás létrejön a két alkari csont keresztben állnak ezzel forgatva a tenyeret lefelé néző irányba. A két csont visszaforgatásával a tenyér felfelé néz és az alkari csontok párhuzamosak lesznek egymással.

Tovább haladva az alkaron lefelé, megtalálható a csukló, mely a kezet az alkarhoz rögzíti. A csukló valójában az alkari csontok és a kéztőcsontok kapcsolódása egy ellipszoid alakú ízületben. A kéztőcsontok 8 kis csontból alkotják meg a kéz alsó részét. Ezek a csontok 2 sorban rendeződve domborulnak a kézfej felé, míg homorulatot képeznek a tenyér felé. Mozgást csak a hüvelykujj kézközépízületével tud végrehajtani, ezenkívül helyzetük és pozíciójuk egymáshoz viszonyítva fixált.

A kéztőcsontokhoz kapcsolódik 5 csöves kézközépcsont, melyek között izmok és ínak feszülnek. A kézközépcsontokhoz ízületesen az ujjak csöves csontjai, ujjpercei és körömpercei illeszkednek.

Az izomműködés az izom összehúzódó képességén alapul. Idegi inger alapján az izom képes összehúzódni, azaz rövidülni és elernyedni, tehát kinyúlni a másodperc tört része alatt. Izomösszehúzódás esetén elektromos jelenségek is megfigyelhetőek, izommunka során működési áram is vezethető le.

A kéz megjelenését tenyéroldalról az ujjakhoz és kézközépcsontokhoz tartozó izmok, míg a kézfejnél a kézközépcsontok formái szabják meg. A tenyér felől jól szembetűnik, hogy

a hüvelykujj jóval fejlettebb, hiszen a kézközépcsonthoz való kapcsolódása köré jóval több izomtömeg tornyosul. A kézközépcsontok között csontközi izmok feszülnek. Az ujjak tenyér felöli oldalán az úgynevezett hajlító, míg kézfej felöli részén a feszítő izmok találhatóak.

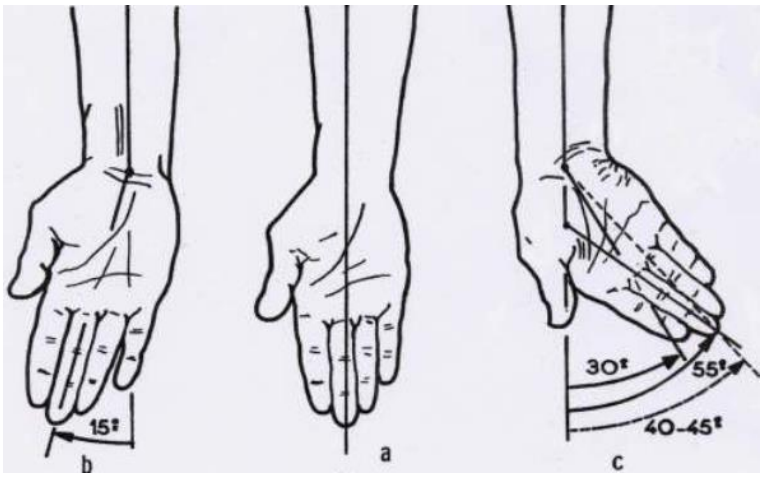
„Akár akaratlagos, akár automatikus a mozgás, a benne részt vevő izmok működésének összerendezettnek, koordináltnak kell lennie. Ezt a koordináló feladatot a kisagy (cerebellum) látja el” [6]

A kisagy jelzéseket fogad a test számos területéről, köztük az egyensúlyi rendszerből, a szem által érzékelt távolságokból is így folyamatosan ismerve a test éppen aktuális állapotát. Ezen információk alapján adja ki az izommozgáshoz szükséges jelek sorozatát az idegrendszeren keresztül. Egyik esete annak, amikor a kisagy nem funkcionál megfelelően, hogy a kiadott jelzései alapján a végtag nem képes a kívánt helyzetbe pozícionálni, csak ha a mozgást az illető a szemével végig tudja kísérni. Csak a sértetlen és egészséges idegrendszer képes a mozgás pontos irányítására és megállítására. A kisagy mozgásszervi kontrolljának működése egy egyszerű gyakorlattal ellenőrizhető. A mutatóujjat a testtől kinyújtott kézzel eltartva, majd csukott szemmel az orrot megérintve tökéletesen kell levezényelnie. A kisagy ezen képességének hiánya súlyos működészavarra utal. Továbbá a kisagy a felelős az egyensúly megtartásáért, kontrollálásáért, a már említett izomkoordinációkért és a beszédért.

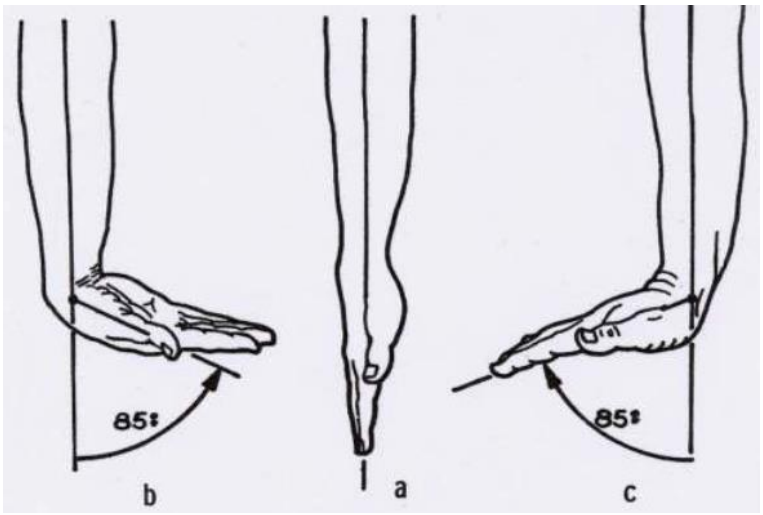
„A test különböző pontjait érő ingerek mindvégig egymástól élesen elkülönült neuronokon haladnak a thalamuson keresztül az érző agykéreg felé (gyrus postcentralis), ahol a testfelszín a maga teljességében pontról pontra térképszerűen képviselve van. Az érző kéreg feladata tehát az inger helyének lehető legpontosabb meghatározása, és ezen továbbmenően az inger elemzése.” [6]

Az ilyen ingerek felismerése egy korábbi emléképpel kerül összehasonlításra. Ez az oka annak, hogy egy idegsérült kéz rehabilitációja során korábbról ismerős mozgáskombinációkat próbálnak felidéztetni az aggyal azáltal, hogy speciális tornagyakorlatokat hajtanak végre az adott végtagon. [6]

A kéz mozgástartományait az izmok és csontok együttes korlátjai határolják be. Ezen mozgástartományokat a csukló két tengelye mentén szemlélteti a (3. ábra.) és (4. ábra.). A csukló a harmadik forgástengelye mentén való forgómozgásban már részt vesz a könyök és a váll is, így képes ez a tengely körül a 225°-os elfordulásra.



3. ábra. A csukló mozgástartományai I. [7]



4. ábra. A csukló mozgástartományai II. [7]

3.3 3D nyomtatás és 3D szkennelés az egészségügyben

A 3D nyomtatók alkalmazása napjainkban egyre elterjedtebbek ipari és háztartási környezetben egyaránt, valahol hobbi, valahol munkaeszközként. A három legelterjedtebb 3D nyomtatási eljárás az SLS, SLA és FDM technológián alapul. Az SLS (szelektív lézerszinterezés) technológia során vékony porrétegeket hordanak fel egy munkaasztalra és a tárgyat így rétegenként egy lézer segítségével olvasztással állítják elő. Az SLA (sztereó litográfia) nyomtatás esetén egy fényre keményedő műgyantával állítják elő a rétegeket. Ez a típus nagyon sima felületeket és nagyon pontos nyomtatást eredményez. Az FDM azaz műanyagolvasztási eljárás során műanyag szálanyagot húz be a nyomtatófej – úgynevezett filamentet –, melyet felmelegít, megolvaszt és ez után rétegenként építi fel belőle az adott alkatrészt. [8]

Sokan gondolják, hogy a már említett olcsóbb nyomtatási technológiákkal nem lehet az egészségügy területén használatos dolgokat készíteni, pedig a legtöbb 3D nyomtatott orvosi eszköz, alkatrész ilyen eljárásokkal készül.

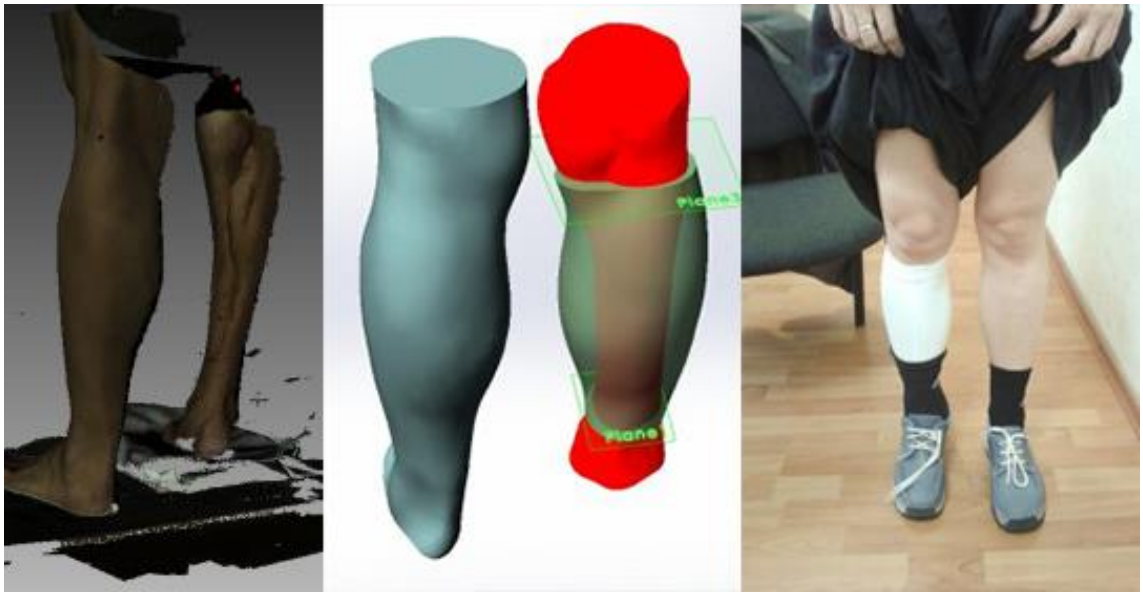
5. ábra. A kéz csontjai - 3D nyomtatással készített, vizualizációt segítő eszköz

Az első nagyobb kategória a vizualizációt segítő termékek, orvosoknak tartott oktatásokon, műtétek megtervezését segítő modellek, páciensekkel való egyeztetések során felmerülő vizualizációs segédeszközök. (5. ábra.)

A második kategóriába tartoznak az orvosi segédeszközök, azaz műtéti sablonok, egyedi eszközök és kellékek. Ezek azok a tárgyak, melyek segítik az orvos munkáját a beteg ellátásában, viszont fontos megjegyezni, hogy ezen eszközök nem válnak a páciens testének részévé. „… ezen a területen a mai magyarországi szabályozás szempontjából minősíteni kell a tárgyakat, ami csak speciális engedélyeztetési folyamat után lehetséges.”

A harmadik és egyben legnagyobb kategória az alkalmazott ergonómia területe. Ide tartoznak a 3D nyomtatással előállított implantátumok, protézisek, sérült ízületi és izomzati tehermentesítő ezközök (ortézisek) és az egészség megőrzéséért felelős eszközök. Ezen területeket bár egymástól külön kell kezelni, mégis szorosan összefonódnak. A 3D nyomtatás mellett a 3D szkennelés is jelentős szakterületté fejlődött. A két digitális eljárás kiegészítve egymást egyre korszerűbb, gyorsabb és pontosabb prototípusgyártást tesz lehetővé. Az implantátumokat a testen belülre helyezik, ezek előállítása a legzártabb rendszerű az összes orvosi segédeszköz közül. Az elemek létrehozásához az orvosoknak kell szkennelni az emberi test belső részeit. Erre rendszerint CT, MRI vagy CBCT eljárást alkalmaznak. Az így szkennelt és létrejött fájlokat 3D nyomtatásra alkalmas fájlokká kell konvertálni, azaz .stl kiterjesztésű fájlokká. Az .stl fájlokat már könnyen fel tudják dolgozni és ezek segítségével hatékonyan pontos protéziseket lehet tervezni.

A protézisek 3D nyomtatása nagyon precízen és pontosan működő nyomtatókat igényel. Olyan anyaghasználat az engedélyezett, amely sterilizálható vagy megfelel az orvosi alkalmazásokhoz szükséges követelményeknek. Az EOSP800 egy olyan nyomtató, ami képes biokompatibilis műanyagot nyomtatni SLS technológiával. Az így létrehozott eszközök alkalmasak a műtéti úton való használatra is. [9]



6. ábra. Szkennelt modell, 3D tervezés, 3D nyomtatott protézis [9]

Ahhoz, hogy ezek az anyagok az emberi szervezettel (szövetek, csontok, bőr, nyálkahártya stb.) érintkezhessenek, biológiailag összeférhetőnek és biofunkcionálisnak kell lenniük.

„Biológiai összeférhetőség, biokompatibilitás alatt az élő szervezet és a belé helyezett mesterséges rendszer vagy anyag zavartalan együttműködését, összeférhetőségét értjük. Az összeférhetőségnek ki kell terjednie a rendszer szerkezetére és felületére is.” [10]

„Biofunkcionalitás alatt azt értjük, hogy az élő szervezetben alkalmazott idegen anyagnak át kell vennie a biológiai rendszer legfontosabb funkcióit. Kiemelt pontok a terhelésátadás, ízületi funkciók.” [10]

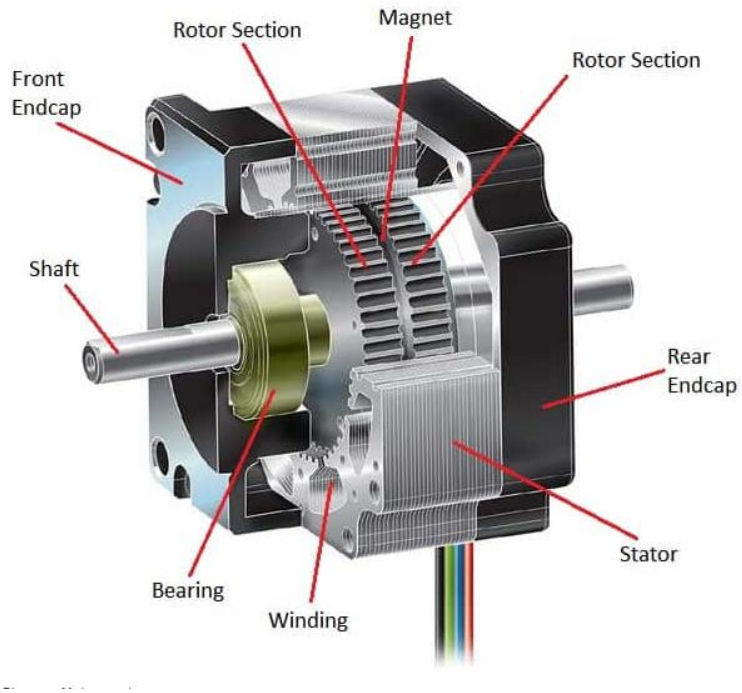
A 3D nyomtatott egészségügyi eszközöknél, mint minden ilyen tárgynál fontos a sterilizálás. Ez folyamat nem egyenlő a fertőtlenítéssel vagy szennyeződésmentesítéssel. A sterilizálási folyamat célja, hogy mentesítse az egészségügyi termékeket minden életképes mikroorganizmustól. „A sterilizálás, definíció szerint a legellenállóbb, a legtöbb környezeti körülmény túlélésére képes mikrobiális bakteriális spórák elpusztítására vagy kiküszöbölésére való képesség.” [10] A polimerek esetén nélkülözhetetlen a sterilizálási eljárások és megismételhetőségüknek pontos ismerete mivel ez akár roncsolhatja is az adott tárgy minőségét, anyagi épségét. A sterilizálást a legtöbbször gőzzel, besugárzással vagy száraz hővel végzik.

A gőzsterilizálás a legegyszerűbb módszer. 80-134 °C-on, minimum 15 percig hajtanak végre. Hátránya, hogy kevés ilyen polimer van, ami kibírja a magas hőmérsékletet, de a 3D nyomtatásban is alkalmazható ABS-t, PVC-t, PEEK-t (poliéter-éter-keton) képes sterilizálni. Nagy előnye, hogy olcsó, gyors, nincs egészség- és környezetkárosító hatása.
A besugárzással végzett sterilizálást elterjedtebb sugárfajtákkal, mint például a röntgennel, elektronsugárral, UV-sugárral végzik. Hátránya, hogy körülményes a folyamathoz szükséges feltételeket megteremteni, előnye, hogy rendkívül hatékony. Általában egy sugárzásra van szükség, de a besugárzási dózis több millió rad.
Magas hőmérsékleten 105-170 °C-on, és a hőmérséklet függvényében minimum 1-16 órán keresztül, szárazhővel sterilizálják azokat az anyagokat, amik nem bírják a párát, valamint az éles eszközöket, hogy csökkentsék a korróziójukat. [10]

3.4 A léptetőmotorok és vezérlésük

„A léptetőmotor egy szénkefe nélküli egyenáramú motor, amelyben minden egyes forgás (fordulat) a motor szerkezetétől függően bizonyos számú lépésre van felosztva. Jellemzően a teljes 360°-os tengelyfordulat 200 lépésre oszlik, ami azt jelenti, hogy 1,8°-os osztásonként egyetlen lépést hajt végre. Vannak olyan motorok is, amelyek egy lépést 2; 2,5; 5, 15 vagy 30°-os osztásonként hajtanak végre.” [11]

A léptetőmotorokat alkalmazzák 3D nyomtatókban, plotterekben, szkennerekben, lemezmeghajtókban, gravírozógépekben, robotokban és ipari gépekben. A léptetőmotor precíz és pontos működése a speciális felépítésének köszönhető. [11] (7. ábra.)



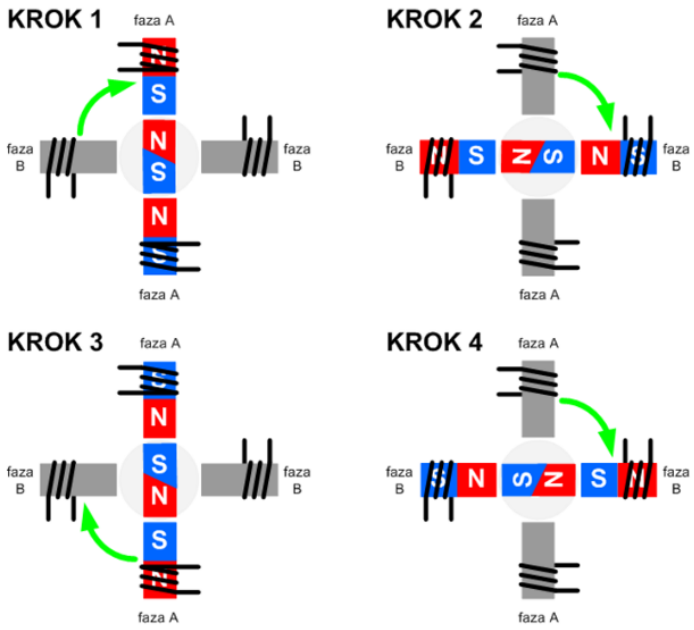
7. ábra. A léptetőmotor felépítése [12]

A léptetőmotor részei:

- magnet - mágnes;
- rear endcap - hátsó zárósapka;
- stator - stator, állórész;
- winding – elektromos tekercselés;
- bearing – csapágy;
- shaft – tengely;
- front endcap – zárófedél.

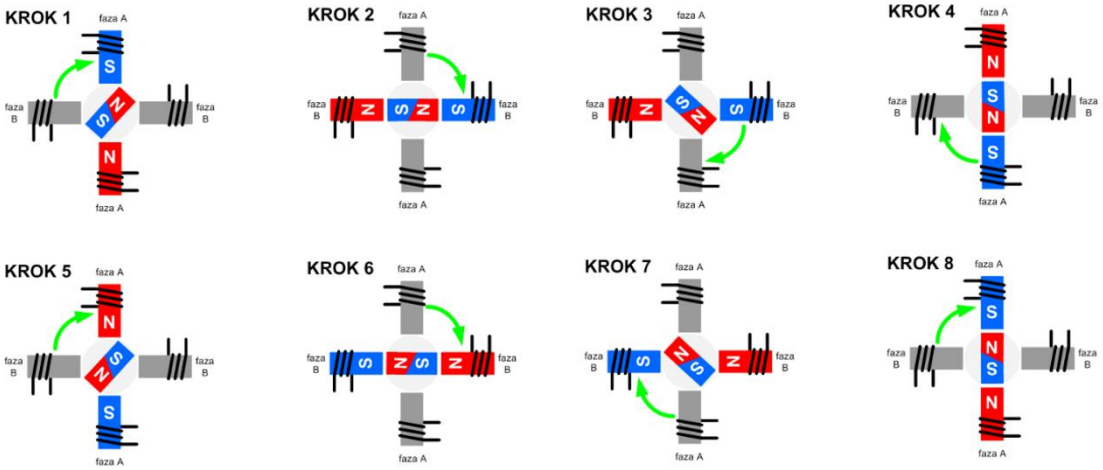
A léptetőmotorok valójában nem forognak, hanem szakaszos mozgást végeznek. A vezérlőimpulzusok következtében meghatározott szögelfordulást hajtanak végre. Az impulzusok sorrendje a motor forgásirányát az impulzusok frekvenciája a forgási sebességét deklarálja. [13]

Ezek alapján a léptetőmotort elektromágnesek sorozata mozgatja, amelyek így a rotor mágneseit pozícionálják. [11] (8. ábra.) (9. ábra.)



8. ábra. A teljes léptetés elvi ábrája kétfázisú tápegységgel [11]

A teljes léptetés működésének lényege, hogy itt a rotor és a stator kialakítása határozza meg, hogy hány lépés tesz ki egy teljes kört. (8. ábra.) Tehát ha 30°-os osztásonként van egy állás akkor a teljes kör – 360°- megtételéhez szükséges lépés 12 lesz. Ha aprólékosabb mozgásra van szükségünk akkor vannak olyan léptetőmotorok, amik 1,8°-os lépést is le tudnak produkálni. Egy ilyen motornál a teljes kör megtételéhez szükséges lépések száma 200. A tengely elfordulása az elektromágnesek aktiválása (egyszerre egy vagy kettő) után történik meg. Ha egy tekercset kapcsolnak, annak nincs szüksége nagy teljesítményre. Ha viszont egyszerre két átellenes tekercset aktiválnak, akkor növekedni fog a leadható nyomaték és a sebesség is, ehhez viszont kétszer akkora teljesítményre lesz szüksége a motornak.



9. ábra. A féllépéses üzemmód elvi ábrája kétfázisú tápegységgel [11]

A következő üzemmódja a léptetőmotoroknak a féllépéses mód. (9. ábra.) A képen könnyen felfedezhetőek az úgy nevezett félállások (a rotor raszter kettő, egyenlő részre osztásával és a raszterek által bezárt szög felével) a teljes léptetés üzemmódhoz képest. Ha az előző 1,8°-os raszterközöket vesszük alapul és ha ennek a felével számolunk - azaz 0,9°-kal –, akkor a teljes körbeforgáshoz szükséges egész lépések száma 400 lesz. A stator tekercseire váltakozó kapcsolót kell kötni, ezzel a kétszeresére növelhető a szögfelbontás, valamint nőni fog a nyomaték és akadálytalanabb, zökkenőmentesebb lesz a motormunka.

A harmadik változat a mikrolépéses mód. Ebben a módban még kisebb rasztertávolságokról beszélünk. Egy 1/32 mikrólépéses, 1,8°-os egész lépéses, kétfázisú szabványos léptetőmotor esetén a korábbi számítások mentén a teljes kör megtételéhez szükséges lépések száma 6400. [11]

A léptetőmotorok lépésszámát az impulzus frekvenciája, míg a forgás irányát a feszültséggel ellátott tekercsek sorrendje képes programozottan beállítani. A léptetőmotorokat mikrovezérlőkkel lehet irányítani. Számos egylapkás számítógép (például a Raspberry Pi, Arduino) vagy több léptetőmotor vezérléséhez gyártott alaplap képes fogadni a léptetőmotor meghajtókat. Ezek a meghajtók általában véve egyszerű eszközök. Feladatuk, hogy a gépvezérlőtől érkező jeleket átalakítsák a léptetőmotor meghajtásához szükséges mintává. A minta sorrendjétől függően feszültség alá helyezi a fázisokat és így lépésenként meghajtja meg a motort, ezzel forgatva az egyik vagy a másik irányba. [14]

3.5 Programozási nyelvek

A programozási nyelvekre azért van szükség, hogy a számítógépekkel lehetőség legyen a kommunikációra és ezáltal a vezérlésre. Ehhez a kommunikációhoz speciális nyelvekre van szükség.

Az alacsonyszintű programozási nyelveket közvetlenül a processzor futtatja, így ezek nagyon gyorsan tudnak futni. A programozási nyelvek két főbb csoportja a gépi kód - ami bináris vagy hexadecimális formában van - és az assembly nyelv, mely a parancsokat szimbolikusan kezeli, így az ember számára is értelmezhető.

A középszinten elhelyezkedő nyelvek kötik össze a magasszintű programozási nyelveket az alacsonyokkal. Előnye, hogy könnyen értelmezhető és felhasználóbarát nyelvek alkotják, amely szorosan kapcsolódik az emberi nyelvhez.

A magasszintű nyelvek felhasználóbarát programok fejlesztésére szolgál. Futtatásukhoz szükség van egy fordító környezetre mely alacsony szintű nyelvre fordítja a kódokat. Könnyen írható és olvasható, valamint karbantartása is egyszerű. Főbb programozási nyelvek a C#, C, C++, Java, JavaScript, Python, Pascal. [15]

```c
#include <stdio.h>

int main() {
  printf("Hello World\n");
  return 0;
}
```

10. ábra. Egyszerű program C nyelven [16]

Egy magasszintű programkód futtatásához a fejlesztői környezetbe implementált fordítószoftver leellenőrizi a kód szintaktikáját. A szintaktika az adott programozási nyelv úgynevezett nyelvtana. Az írt program lefutása előtt a fordítóprogram felkutatja a lehetséges szintaktikai hibákat és ha az helytelen, nem fordítja le alacsonyabb szintű nyelvre a kódot. Ha a szintaktika hibátlan akkor megtörténik a fordítás és a program lefut. A szemantika a használt programozási nyelv lehetséges építőkövei. Szemantikában vétett hibák a program lefordulása után, a program futása közben derülnek ki. [17]

4. A csukló általi gépi főtengelyek meghatározása

A kézrehabilitációs gép tervezése kezdetén, az első dolog amire megoldást kellett találjak, hogy egy olyan vázat és szerkezetet tervezzek, amellyel reprodukálni tudom a csukló és az ujjak mozgását. A gépet a saját kezemen fogom tesztelni és alkalmazni, ezért a továbbiakban ennek megfelelően fogom folytatni a tervezést. Alaposan tanulmányoztam a kéz mozgástartományait, mozgástengelyeinek irányait és tengelyek körül lehetséges elfordulási szögeit. A tervezési munkám kezdetén, megállapítottam a csukló és az ujjak főtengelyeit.

A csukló mozgása három tengely mentén történik. Ennek megfelelően három tengelyvonalat kellett meghatároznom a rehabilitációs gépben is. Ahhoz, hogy ezeket a mozgatásokat összehangoltan, egyszerre, minden tengely mentén és külön-külön is tudjam reprodukálni, a tengelyek között sorrendet állítottam fel. A csukló szempontjából belülről kifelé haladva raktam sorba a három forgástengelyt. A csukló egyhelyben való mozgatása során, viszonyítási pontnak a középső ujj végét választottam meg.

Belülre került a legkisebb mozgástartományban mozgó tengely. Ha az emberi test elé, egy asztal fölé, tenyérrel lefelé kinyújtott jobb kezet veszem referencia pozíciónak, akkor az asztal lapjára merőleges tengelyt – a csukló középpontján keresztül – választottam z tengelynek. A z tengely mentén a jobb kézhez tartozó csukló egy balra 15°-os, míg jobbra 45°-os szöghöz tartozó körívszakaszon tud elmozdulni. Tehát a középső ujj végének bejárható útja összesen egy 60°-os szöghöz tartozó ívpálya.

Az eggyel nagyobb mozgástartományba eső tengely az asztal lapjával és a két vállal is párhuzamos, a csuklón keresztül haladó x tengely. Ennek mozgástartománya a vízszintes állástól nézve lefelé és felfelé fordítva is 85°-85°. Ez azt jelenti, hogy az x tengely körüli forgás mozgástartománya, így 170°.

A csukló legnagyobb mozgástartományában mozgó tengely került legkívülre. Ez a tengely a könyökön, csuklón és a középső ujjon átmenő, vállakra merőleges tengely, azaz y tengely. Mozgástartománya a tenyeret óramutató járásával ellentétesen forgatva 45° és óramutató járásával megegyezően 180°. Így a teljes mozgáshoz tartozó szög 225°.

Bár a készülék megépítése és tervezése jelen állás szerint a csuklóra összpontosul, számolnom kellett a gép továbbfejlesztési lehetőségeivel is, ahol szükségszerű az ujjak mozgatásával is kalkulálnom. Az ujjak tengelyeinek meghatározása az ujj ízületein

keresztül történt. Az ujjak ízületeinek szabadságfoka 1, tehát egy forgástengelye körül forgatható hengerrel lehet helyettesíteni. Itt nem egy tengelyt deklaráltam minden ízületnél, hanem az összes ízület tengelyére merőleges hatásvonalat húztam, ezáltal csökkentve a gépbe beépítendő motorok számát. Ezek alapján ujjanként egy tengellyel számoltam, mert az a rehabilitáció szempontjából elég mozgás lehetőségeket tud lefedni az ujjak számára. Bár az ujjak tövénél az első ízületnél nem csak egy szabadságfokú ízület van, a rehabilitáció szempontjából ez a mozgás elhanyagolható.

A megállapításaim alapján jutottam arra a következtetésre, hogy a csukló mozgatásához három, míg az ujjak mozgatásához elegendő ujjanként egy, azaz összesen nyolc mozgatóegységgel számolnom.

5. A gép mozgatóelemeinek kiválasztása

A témában végzett kutatásaim során olyan gépekkel találkoztam, amik csak az ujjak rehabilitációs mozgatását végzik. Ennek az az oka, hogy egy gépbe beletervezni a teljes kéz rehabilitációs mozgatását nagyon nehéz. Rengeteg az összetett mozgás, amik akadályozni tudnák az ilyen gépek működését.

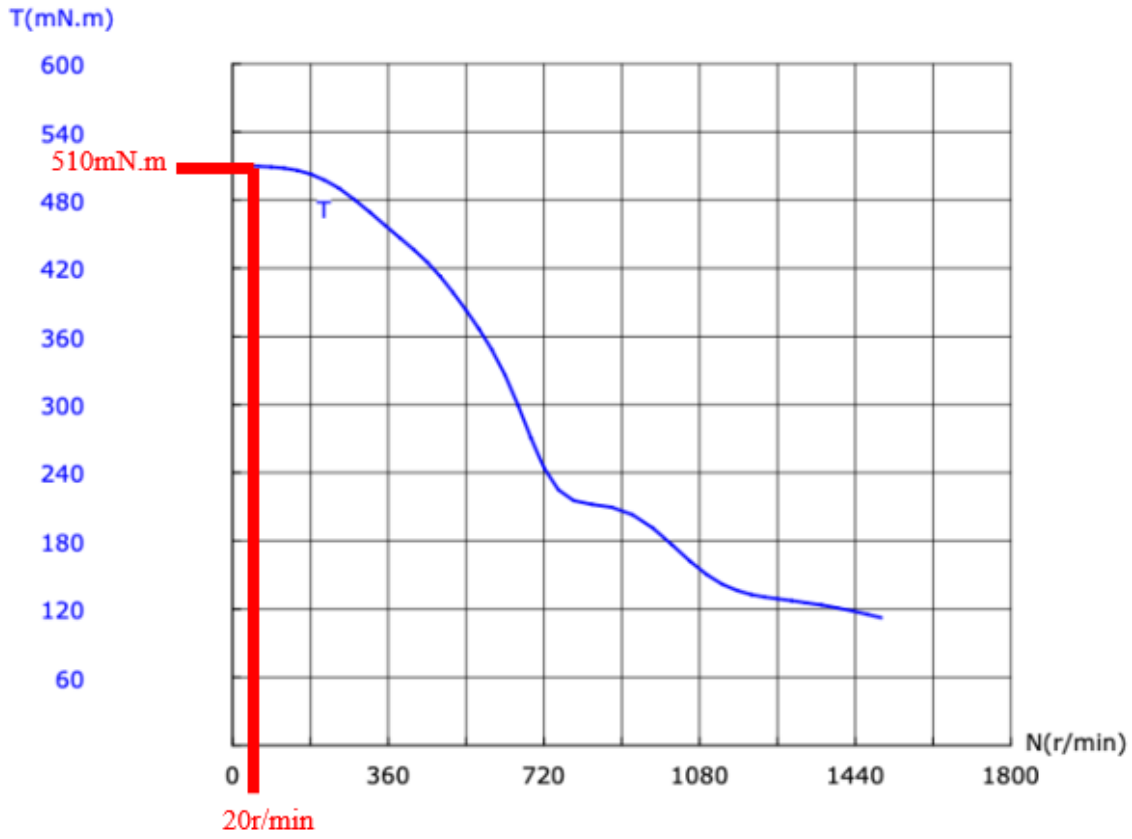
Vannak gépek melyek pneumatikus, hidraulikus vagy szervomotoros rendszereket alkalmaznak. Rehabilitációs orvosokkal folytatott beszélgetéseim során arra jutottam, hogy ezeket a gépeket nem lehet nagyon pontosan vezérelni, akár mm-ről mm-re. Ezért választottam a végtag mozgatásához a léptetőmotorokat. Régóta foglalkozom 3D nyomtatással és ezen motorok működését és a hozzájuk tartozó vezérlést is elég jól ismerem ahhoz, hogy tudjak velük dolgozni a projektem során és meg tudjam általuk valósítani a szükséges precíz mozgásokat.

A csukló mozgatásához három léptetőmotorra van szükségem, tengelyenként egy motorral számoltam. Annyit biztosan tudtam, hogy a gép mozgatásáért felelős részét tehermentesíteni fogom, hogy a kéz teljes súlyát ne a motoroknak kelljen viselnie, hanem közvetlenül egy támasztó állványnak a gép előtt. Ezzel számolva és a csukló mozgástartományainak figyelembevételével szükségem volt 2 nagyobb teljesítményű léptetőmotorra, valamint egy kisebbre.

A kisebbik léptetőmotornak súlyt alig kell elbírnia mert a csuklót vízszintes irányban mozgatja a z tengelye mentén. Erre a 2,5amperes, 2fázisú, NEMA17-es 48mm-es léptetőmotort választottam. (11. ábra.) Dimenziói 42mm*42mm*48mm-esek és mindössze 340g-os, így ez egy könnyebben beszerelhető kisebb motor valóban. Abban az esetben, ha a motort szükséges a gépen belül más mozgatott elemekhez rögzítenem akkor sem okoz számottevő problémát. 1,8°-os teljes lépésszögei elegendő precizitást biztosítanak a kéz z irányában történő aprólékos mozgatásához. A tartónyomatéka 0,59Nm, ami azt jelenti, hogy feszültség alatt a motor képes megtartani 0,061kg-ot egy, 1m-es karon lépésveszítés nélkül0. Mivel ennek a motornak a tengelyére egy maximum 30mm átmérőjű tárcsát szerelnék fel, így az a 0,59Nm-rel számolva már 4kg súly megtartásához elegendő. (5.1) (5.2)

11. ábra. NEMA17 léptetőmotor 48mm 2,5A [18]

A léptetőmotor tartónyomatéka nem egyenlő a forgómozgás közben kifejtett nyomatékkal. Ezen érték meghatározásához a léptetőmotor leírásában szereplő nyomatéki diagramot használtam, hogy le tudjam olvasni a fordulatszámhoz tartozó nyomatékot. (12. ábra.) A motornak nem szükséges, hogy gyorsan forogjon, rehabilitációs szempontból lassú mozgásokra van szükségem. Ezért 20 fordulat/perccel számoltam, hogy biztosan a maximális igénybevételt vegyem alapul, mivel minél magasabb a fordulatszám, a kifejtett nyomaték annál kevesebb. Így a leolvasott nyomaték 0,51Nm. Ezzel számolva a motor képes maximum 20 fordulat/perccel 3,46kg teher mozgatására. (5.3)



12. ábra. NEMA17 48mm 2,5A léptetőmotor nyomatékdiagramja [19]

A forgatónyomaték számításánál használt összefüggés [20]:

$$M = F * \frac{d}{2} (5.1)$$

rendezve az egyenletet az erő, kg meghatározásához:

$$F = \frac{M}{\frac{d}{2}} (5.2)$$

$$F = \frac{0{,}59Nm}{\frac{0{,}03m}{2}} = 39{,}33N = 4kg (5.2)$$

$$F = \frac{0{,}51Nm}{\frac{0{,}03m}{2}} = 34N = 3{,}46kg (5.3)$$

$$F = \frac{2Nm}{\frac{0{,}05m}{2}} = 80N = 8{,}1kg (5.4)$$

ahol: M: forgatónyomaték [Nm];

F: kerületi erő [N];

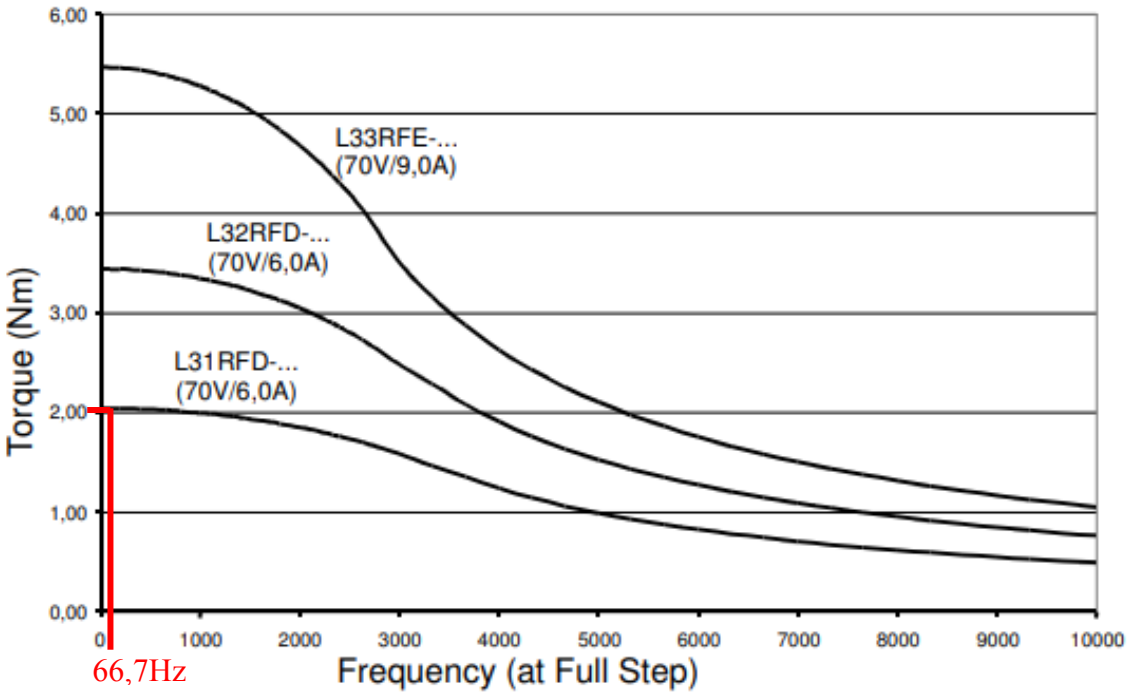
$\frac{d}{2}$: erőkar [m];

$1N = 0{,}10197kg$

Egy emberi kéz a csuklótól az ujjakig megközelítőleg 0,6-0,9kg. Így a számított értékek alapján ez a motor megfelelő lesz a kéz z tengely körüli forgatásához.

A másik két nagyobb motor adott volt a számomra, mert itthoni körülmények között már rendelkezésemre álltak, ezért ezeket terveztem felhasználni. Ezek a motorok már nagyobb terhet kell, hogy megmozdítsanak, így valóban szükségesek az erősebb léptetőmotorok. Mindkét motor egy L31RFD-00N-NN-00 típusú 6 amperes két fázisú léptetőmotor, amelyek darabonként 1,6kg-osak. Tartónyomatékuk 2,8Nm, ami a korábbi számítások alapján már bőven túlteljesít, de mivel ez nagyobb gépalkatrészeket fog mozgatni így szükségszerű, de nem optimális. A motorok fordulatszáma nem fogja meghaladni a 20 fordulat/perc értéket. Ezeknek a léptetőmotoroknak a nyomatékdiagramját a nyomaték mellett nem fordulat/percben adták meg, hanem frekvenciaértékben, ami teljes lépésre vonatkozik. Az x tengelyen van jelölve a teljes fordulatokat Hz-ben kifejezve, ezres léptékekben. A motor 200 teljes léptéket tesz meg egy fordulat alatt. A diagramon jelölt 1Hz 1 teljes lépték/másodpercnek felel meg. Ahogy a legkisebb motornak, ezeknek sem kell, hogy gyorsan forogjanak, ezért itt is egy maximum fordulatszámmal számoltam, 20 fordulat/perccel. A 20 fordulat az 20*200 teljes lépés a motornak. Ez azt jelenti, hogy az így kapott 4000lépték/perc értéket le kell osztanom 60 másodperccel, hogy megkapjam hány Hz-en üzemel a motor 20fordulat/perc fordulatszámon. Az így kapott 66,7Hz-et leolvastam a diagramról, mely szerint a motor nyomatéka forgás közben hozzávetőlegesen 2Nm 6 amperen. (13. ábra.)

A két nagyobb léptetőmotorra maximum 50mm-es tárcsával számolva, 8,1kg-ot tud folyamatos forgással mozgatni, 20fordulat/perc alatt. (5.4) A magasabb forgónyomatékra szükség lesz, mert ezek a motorok fogják mozgatni a csuklót x és y tengelyei körül, más alkatrészekkel együtt, amik így mind hozzáadódnak a nagyobb súlyhoz és indokolt lesz ezen motorok használata.



13. ábra. A L31RFD-00N-NN-00 motor nyomatékdiagramja a nyomaték és a Hz-teljes lépték függvényében [21]

A továbbiakban ezekkel a léptetőmotorokkal és forgatónyomatékaikat figyelembevéve méretezem a gép alkatrészeit és paramétereit, szem előtt tartva, hogy a szükséges alkatrészeket mozgatni tudjam. A forgatónyomatékokat a lehetséges maximum erőkarokra számoltam ki, így, ha ennél kisebb tárcsákat választok meg, a motorok által kifejthető nyomatékok tovább emelkednek.

6. A kéz 3D szkennelése

Mielőtt a tervezési fázisba belekezdtem volna, felkerestem egy olyan személyt, aki 3D szkenneléssel foglalkozik, hogy a jobb kezemről készítsen egy szkennelt fájlt. A kéz 3D szkennelése egy EinScan Pro 2X kézi szkennerrel készült. A leképezés nehézsége, hogy az ujjak közötti területeket nehéz letérképezni, valamint a kéz apró bemozdulásai tovább rontják a folyamat pontosságát. Az eredményes szkenneléshez az vezetett, hogy a poligonok db számát alacsonyabbra állítottuk és a körbefényképezés sebességét gyorsítottuk. A művelet során egy igen pontos 3D-s modellt kaptunk, ellenben nem éles textúrával. Jövőre nézve hasznos észrevételeim egyike, hogy egy ilyen típusú szkenneléshez elengedhetetlen egy, a kéz befogására alkalmas szerkezet, hogy a mozdulatlanságot könnyebben fenn tudja tartani a páciens. Ezáltal még több poligon leképezése lenne lehetséges, amivel még pontosabb eredményeket érhetnénk el.

14. ábra. A jobbkéz 3D szkennelése EinScan Pro 2X 3D szkennerrel

A modell létrehozására azért volt szükségem, mert így pontosabb méreteket, irányokat és síkokat tudtam meghatározni a kéz 3D-s modelljén. Mindemellett a gép tervezése során

is hasznos lehet, ha a 3D-s modellt be tudom illeszteni a készülékem terveibe, elősegítve ezzel a tervezési fázist és a rehabilitációs gépem szemléltetését egyaránt.

A gépet a jobbkezem mozgatásához gyártom le, hogy a későbbiekben más személy bevonása nélkül is tudjak a fejlesztéssel foglalkozni. A tervezést a népszerű CAD szoftverben, az Autodesk Inventor-ban folytattam. A programban dolgozva a kézmodellen segédsíkok definiálásával meghatároztam a csukló 3 forgástengelyének metszéspontját és ettől a ponttól kiindulva a középső ujj végéig lemértem a kéz hosszát, amely esetemben 185mm lett.

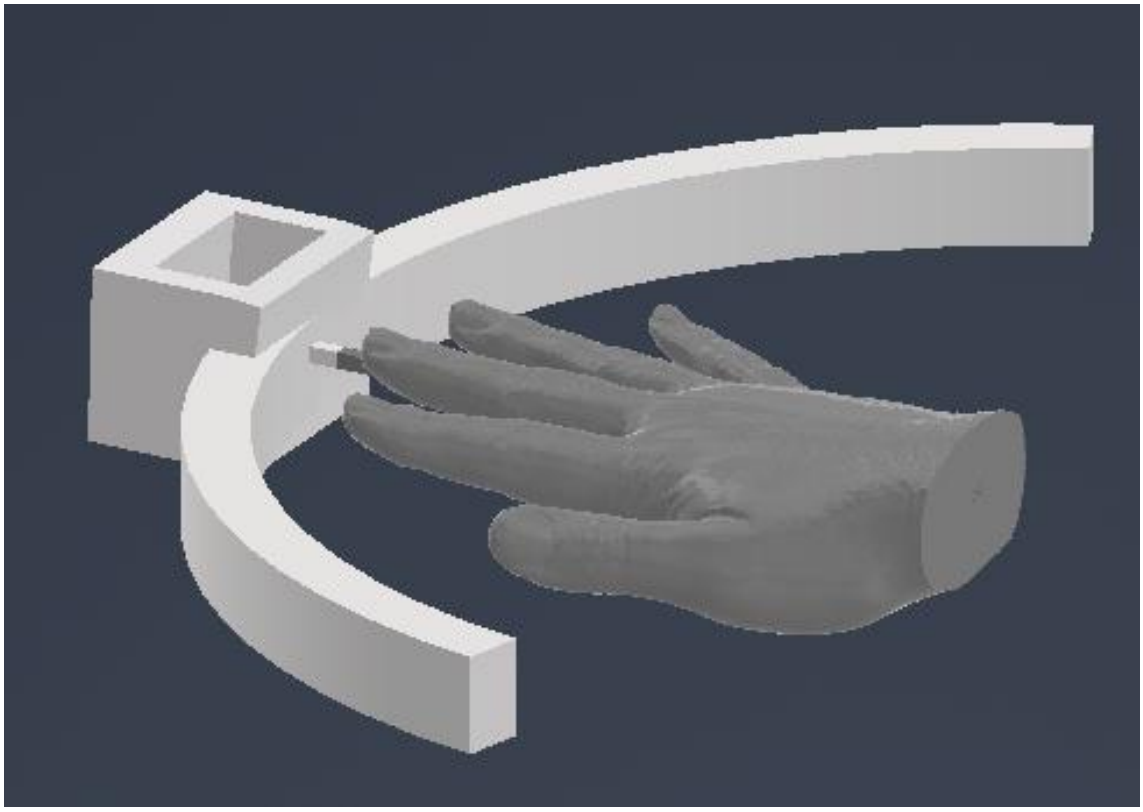
15. ábra. 3D szkennelt kéz modellje

7. A kéz rögzítésének és a gép mozgatási pályáinak meghatározása

A kéz alátámasztását egy párnázott állvány fogja biztosítani a gép előtt közvetlenül. Ehhez az alátámasztáshoz egy laza tépőzár pánttal fogom az alkart hozzárögzíteni a csukló és a könyök között, tehermentesítve ezzel a gépet, amennyire csak lehetséges.

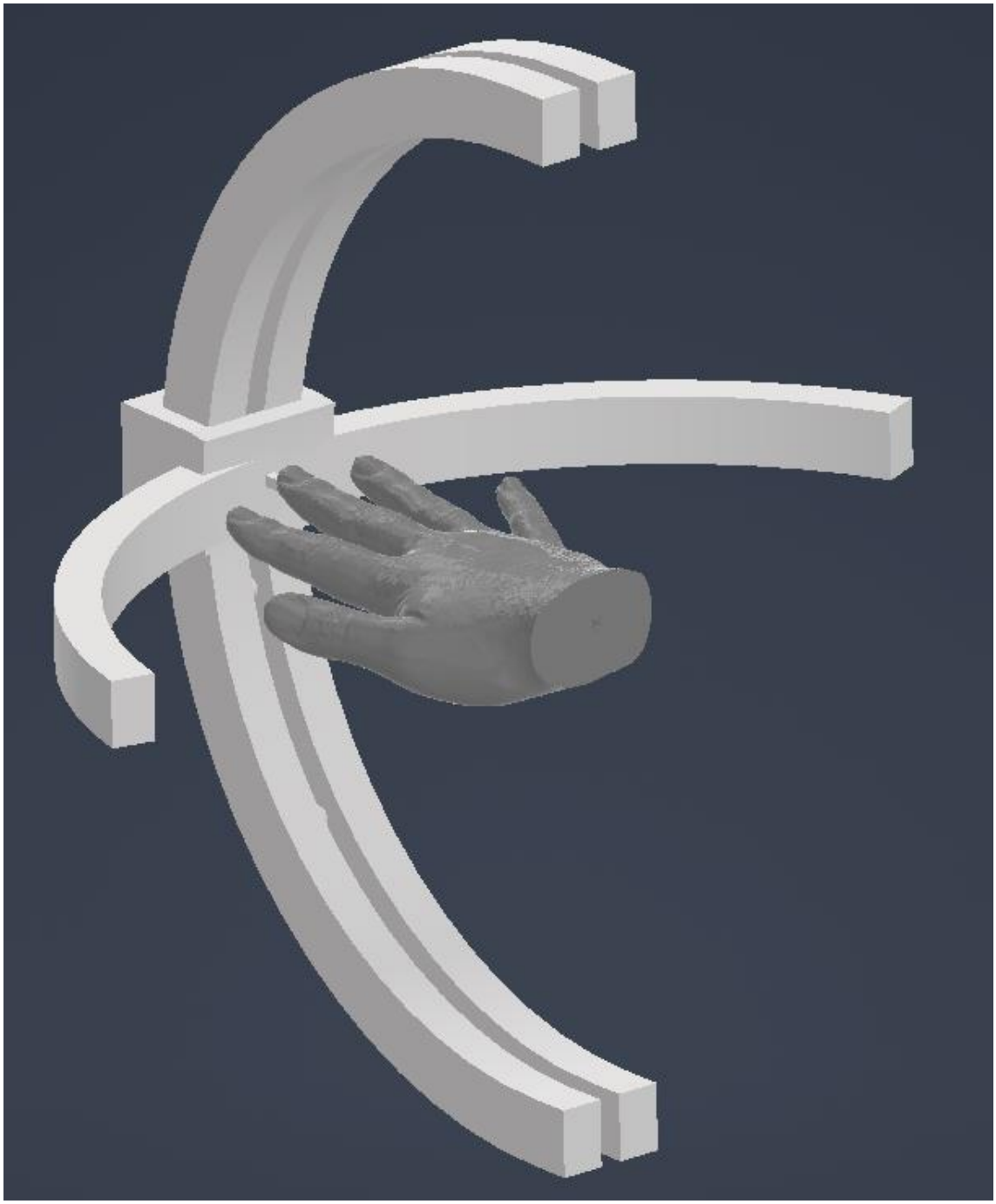
A kéz géphez való rögzítése a 3D szkennelt kéz segítségével valósítható meg. A 3D-s modell tenyér részéből egy olyan kimetszés készíthető, ami 3D nyomtatás után tökéletesen belesimul a páciens tenyerébe. Ehhez a tenyér darab alakzathoz egy sztenderd felső rész csavarozható, ami a kézfejen illeszkedik a kézhez. Ehhez az alkatrész pároshoz ugyancsak szíjjal rögzíthető a kéz. A kézfejnél lévő alkatrész tovább terjed a középsőujj végéig és itt történik a kéz gépbe való rögzítése. Ahhoz, hogy több kézzel is kompatibilis legyen ez az alkatrész és a gép is maga, a kapcsolódó konzolt egy reteszhorony szerű kivágással lehet rögzíteni a géphez. A csúszó alkatrészt úgy kell beállítani, hogy a kéznél meghatározott forgástengelyek metszéspontja a gép által deklarált forgástengelyek metszéspontjába kerüljön.

A csukló körívpályák mentén mozog. A gép tervezése során is ezt tartottam szem előtt, hogy hogyan tudnám ezt megvalósítani. Itt is, mint ahogyan a kéz tengelyeinek meghatározásánál a kisebb mozgástartományokkal kezdtem. A kéz z tengely körüli forgását terveztem meg elsőként. Meghatároztam egy olyan körívszakaszt, amelynek középpontja a csuklóforgástengelyeinek metszéspontjával esik egy pontba és sugara nagyobb, mint a kéz és a géphez való rögzítéséhez szükséges alkatrész hossza. Ezt a körívszakaszt egy sínben terveztem rögzíteni, amiben csúszó mozgást tud végezni, ha a NEMA17-es motor mozgatja. A hosszát a körívszakasznak úgy terveztem meg, hogy balra 35°-ot, jobbra pedig 45°-ot csúszva is a rögzített sínben maradjon az alkatrész. Ennek elvi ábrázolása látható az ábrán. (16. ábra.)



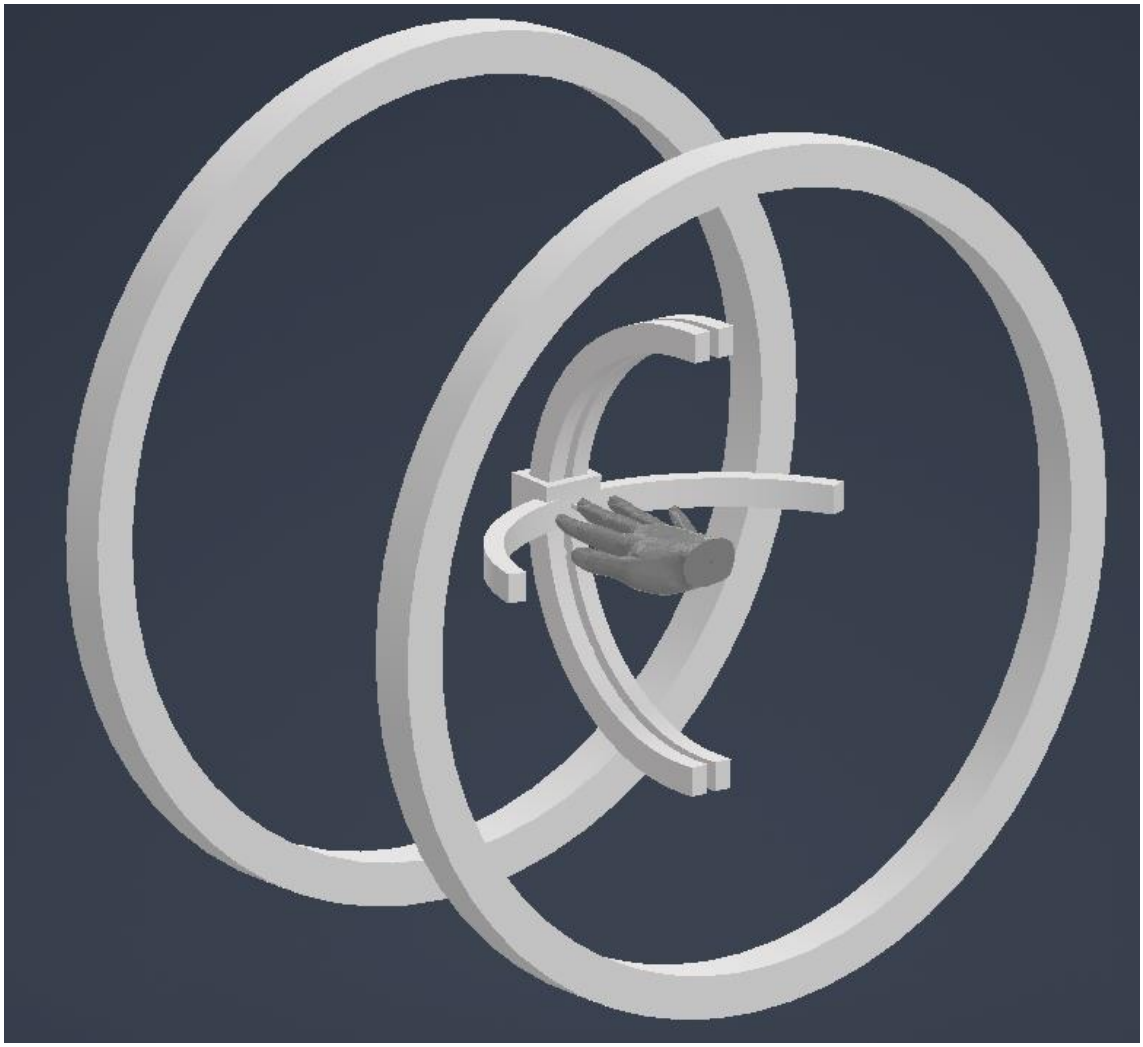
16. ábra. A kéz z tengely körüli forgatásának vázlatos megtervezése

A következő nagyobb körívalkatrész tengelye - ami mentén a kéz is forog - az x tengely. Ezt merőlegesen a z ívszakaszra vettem fel. Ez egy nagyobb, felfelé és lefelé is 85°-os mozgást igénylő körívszakasz, aminek sugarát úgy növeltem, hogy nagyobb legyen, mint a z tengelyhez tartozó körívszakaszé. Ennek érdekében a körívszakasz hosszát úgy választottam meg, hogy legyen elegendő hely a nagyobb gépegységhez való rögzítéshez és a 85°-85°-os mozgást meg tudjam rajta vezetni. Itt egy sínpárt fogok megvalósítani, hogy masszívabb legyen a szerkezet, mert ennek már el kell bírnia a belső z tengelyhez tartozó körívszakaszt és a rá rögzített kezet is. Ezt szemlélteti az ábra (17. ábra.).



17. ábra. A kéz x tengely körüli forgatásának vázlata

Az y forgástengelyhez nem egy körívszakaszt, hanem két teljes kört rendeltem hozzá. Erre azért volt szükség, mert ez lesz a legkülső két alkatrészem, ami mentén a kéz tud forogni és ez lesz a gépem külső masszív váza is. Ezeknek az alkatrészeknek el kell bírniuk a belső két tengelyt, valamint az azokat mozgató motorokat is. (18. ábra.)



18. ábra. A kéz y tengely körüli forgó mozgatásának vázlata

A tervezés kezdetén mellékes célként tűztem ki, hogy a gép alkalmas legyen a jobb és a bal kéz mozgatására egyaránt, ne legyen szükség másik gépre a kéz megcserélése esetén. Ezzel a gépet alkalmazó cégek jelentős anyagiakat spórolhatnak meg, hiszen elegendő egy gépet megvásárolniuk és annak további karbantartási költségeit fedezniük.

Az imént felvázolt konstrukcióval a kétkezes rehabilitáció lehetséges lenne, hiszen függőlegesen a gép szimmetrikus - a kéz irányából tekintve- így nincs jelentősége annak, hogy melyik kezet illesztik a készülékbe. Legfeljebb pár kisebb alkatrészt kell átszerelni vagy passzív, illetve aktív állapotba helyezni, hogy ez megvalósítható legyen.

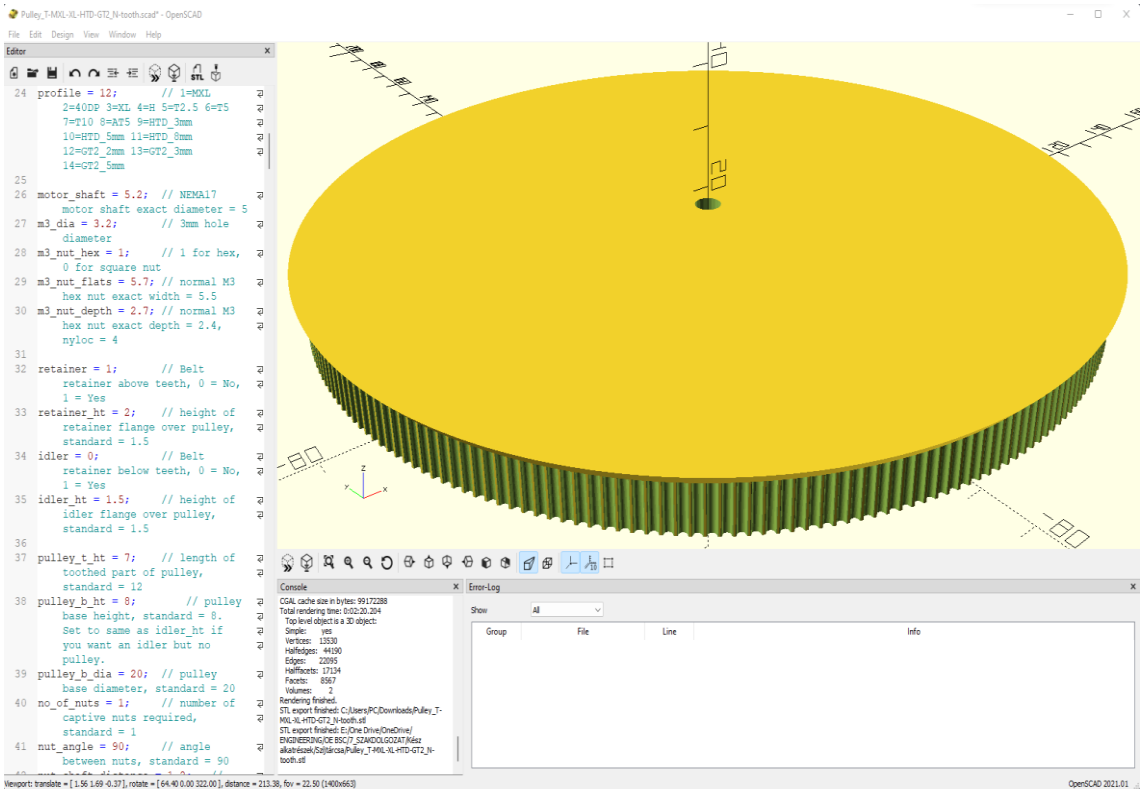
8. A gép tervezésének véglegesítése

Miután meghatároztam a gép fő mozgásainak alapelemeit a terveket elkezdtem kidolgozni, továbbra is az Autodesk Inventor szoftvert alkalmazva.

A tervezés ezen fázisát a motorok rögzítési pontjainak megkeresésével kezdtem és köréjük építettem fel a már meghatározott íves elemeket. A motort úgy helyeztem el, hogy a hozzátartozó elemet zsinegek felcsévélésével és leengedésével tudja mozgásba hozni. A legkisebb, azaz a NEMA17-es motort a z-x tengelyeket összekötő konzol hátoldalára helyeztem. A konzolba kötöttsínpályával rögzítettem a z tengelyhez tartozó elemet.

A konzol az x tengelyhez tartozó dupla körívpályán mozog, 6db görgő segítségével. A következő feladat az x tengely körül mozgató motor elhelyezése volt úgy, hogy ezt a konzolt tudja mozgatni zsinórok segítségével. Ezt az úgynevezett egy szinttel kintebb lévő mozgatott elemen helyeztem el, az y tengelyhez tartozó teljes körben. A motor a kör közepére helyeztem azért, hogy ha forgatom a teljes gépet akkor a motor ne jelentsen visszatartó erőt az 1,6kg-os súlyával, hanem legyen az y forgástengely vonalán. Ehhez a motorhoz így gond nélkül hozzá tudtam tervezni a zsinegeket, hiszen az x motor tengelye, az x tengelyhez tartozó íves elem és a z-x konzol is egy síkra estek.

Mint ahogyan az előző két esetben is, az y tengely mentén mozgató léptetőmotort is a tőle kintebb lévő elemen helyeztem el, az egész gépet tartó bakok hátsó darabján. Ennek a motornak a mozgatásához nem zsineget, hanem GT2-es, 6mm-es bordásékszíjat terveztem használni. A motor egyedi tengelykialakítása miatt szabványos ékszíjtárcsát nem tudtam alkalmazni, ezért 3D nyomtatással terveztem kivitelezni ezt. A másik, azaz hajtott tárcsát és a hajtó tárcsát is OpenSCAD tervezői programban hoztam létre. A program lényege, hogy programozottan, parametrikus tervezéssel lehet benne létrehozni 3D modelleket. Egy előre generált programot kerestem internet segítségével és ezen program attribútumain változtatva lépésről lépésre programozottan generáltam le szükséges alkatrészeket.

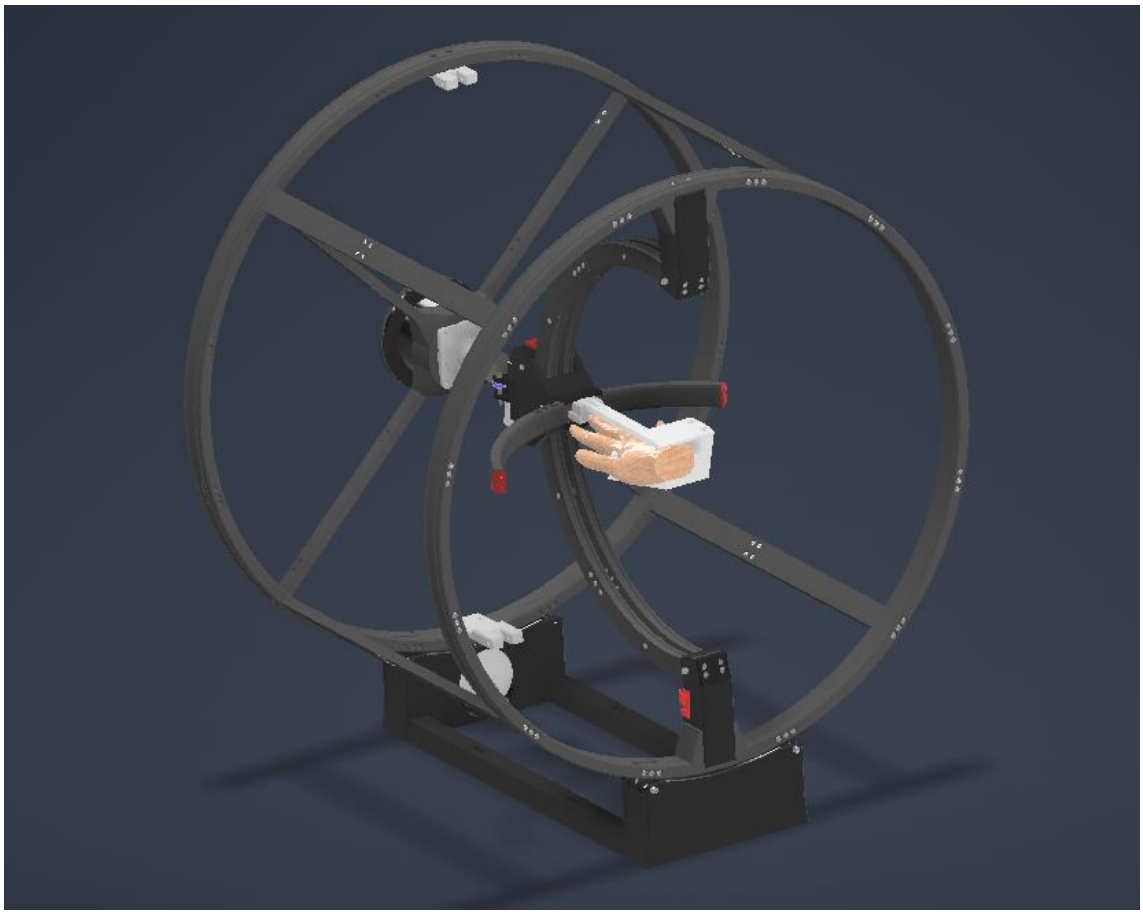


19. ábra. A szítárcsák parametrikus tervezése OpenSCAD szoftverben

A GT2-es ékszíjhoz a két szíjtárcsát 60 (hajtó kerék) és 240 (hajtott kerék) fogszámosra terveztem, hogy a nagy áttéttel tudjak nyomatékot nyerni. A szíj hosszát nem kellett kiszámolnom, mert sajátkezűleg tudtam végteleníteni bármekkora hosszúságot. A szereléshez és az előfeszítéshez a motort a görgős bakban való függőleges vájatban terveztem mozgatni és rögzíteni. A kialakítást úgy hoztam létre, ha a GT2 szíj nem lenne megfelelő a hajtás átvitelére itt is tudjam alkalmazni a zsineges mozgatást az erőátvitelhez.

A motorok pozíciójának megtervezése után az alkatrészeket pontosítottam, közéjük rögzítőelemeket és tartóelemeket helyeztem, valamint meghatároztam a szíjfeszítő tárcsák és végálláskapcsolók helyzetét.

Az alkatrészeket 3D nyomtatva terveztem előállítani ezért a túlméreteseket úgy kellet feldarabolnom, hogy a nyomtató asztalra férjenek. A feldarabolt alkatrészeket lépcsős alakban metszettem ketté több helyen is, hogy így 2-3-4 csavarral tudjam őket összeszerelni, ezzel növelve az egész gép stabilitását. (1. sz. melléklet)



20. ábra. A rehabilitációs gép terve

9. A gép alkatrészeinek gyártása és összeszerelése

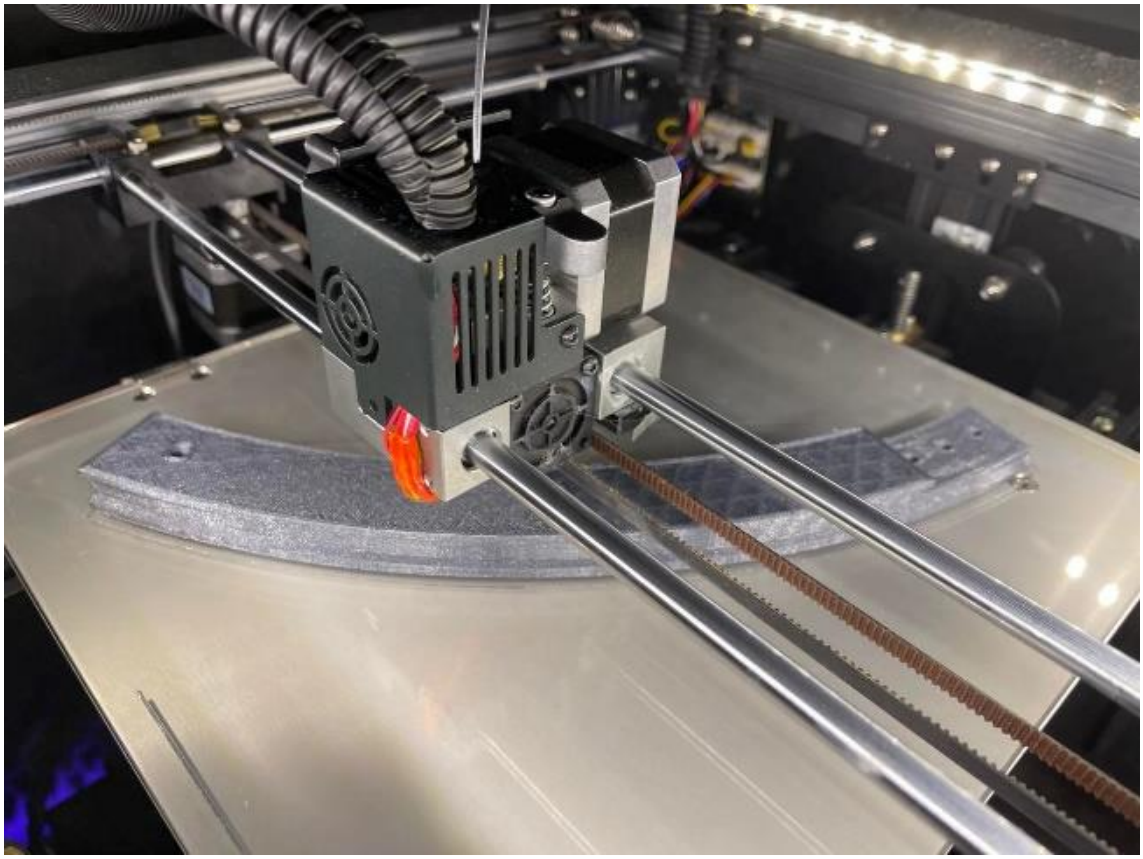
A prototípus kivitelezését 3D nyomtatással terveztem megvalósítani. Ennek az egyik oka, hogy költséghatékonyabb kivitelezést kínált, mintha előre gyártatott alkatrészeket alkalmaztam volna. A másik indok, hogy otthoni körülmények között is le tudtam gyártani ezeket az egyedi alkatrészeket, ezáltal gyorsabb volt a tesztelési- és készre szerelési folyamat.

Az alkatrészek nyomtatására két gépet használtam. Egy módosított Creality Ender 3v2-t és egy Creality Sermoon d1-et. Mind a kettő 3D nyomtató FDM technológia elvén működik. Már a prototípus legyártásához olyan anyagot kerestem, ami megfelel az egészségügyi előírásoknak. Így esett a választásom a Fiberlogy Easy PETG anyagaira. Könnyen nyomtatható, nem deformálódik, nincsenek tapadási gondjai. Mindezek mellett a nyomtatott alkatrészek könnyűek és a rendelkezésemre álló motorokkal könnyen tudom majd mozgatni az így eredményezett súlyokat, de ugyanakkor kellően masszívak is lettek ahhoz, hogy ne deformálódjanak vagy hajoljanak meg az igénybevételek által.

Ennek a PETG-nek a deformálódása 85 °C-nál kezdődik így az ebből készült alkatrészek rendre sterilizálhatóak az egyszerű gőzsterilizálással, ami 80 °C-on már megvalósítható. Bár ez a hőmérsékleti érték megközelíti a határértékét a sterilizálási eljárásnak, a gép végleges verzióját nem ebből az anyagból gyártanám le. Amennyiben 3D nyomtatással készülne a rehabilitációs gép, SLS technológiát alkalmaznék és PEEK anyagot használnék, de ennek a nyomtatási eljárásnak ezzel az anyaggal sokkal magasabb lenne a költsége. Figyelembevéve, hogy az általam elkészített készülék csak egy prototípus, ez az anyag tökéletesen megfelelt az alapvető elvárásoknak és a tesztelések elvégzéséhez.

A 3D nyomtatáshoz szükséges szeletelőprogramban a nyomtatási beállításokat az eddigi tapasztalataimra támaszkodva állítottam be. Ügyeltem a megfelelő falvastagságra és üregességre, hogy az alkatrészek megfelelő mértékben legyenek masszívak, de ugyan akkor könnyűek.

A z tengelyhez tartozó körívalkatrészből és a hozzá tartozó konzolból nyomtattam 1-1 részletet, hogy tesztelni tudjam az illesztést. Első nyomtatásra ez túl szoros lett, ezért a konzol sínjének belső méreteit 0,3 mm-rel bővítettem, hogy a benne futó alkatrész csúszni könnyen, de lötyögni ne tudjon. Ezzel a méretezéssel már megfelelően siklottak egymásba az alkatrészek.

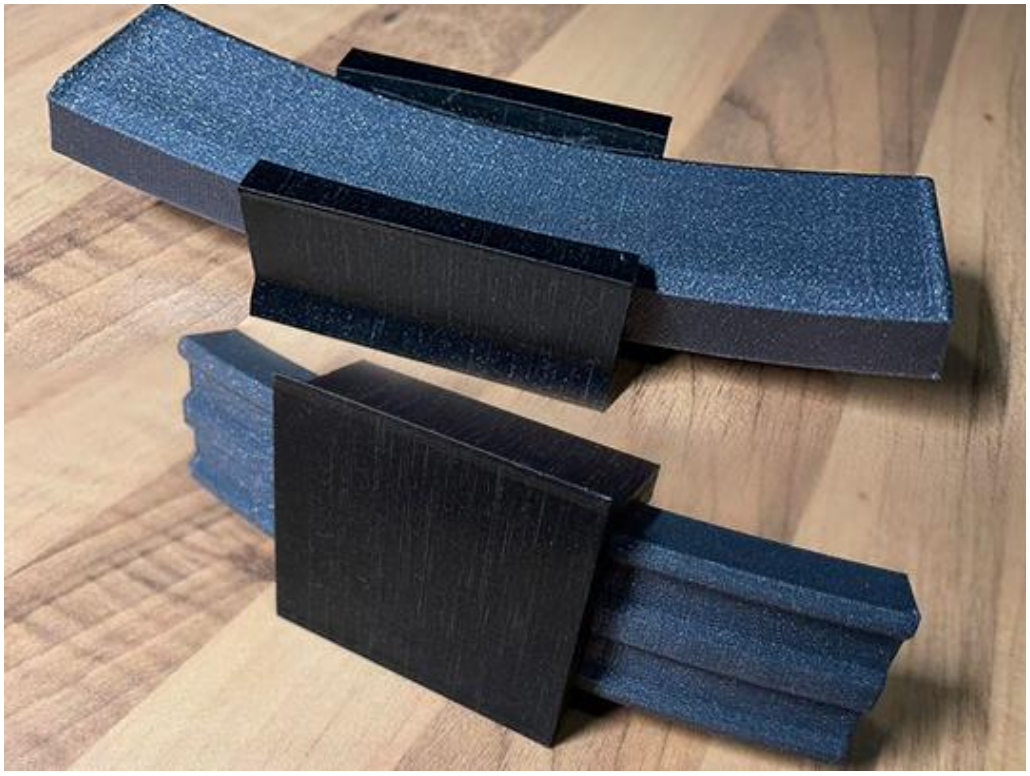


21. ábra. A gép alkatrészeinek nyomtatása Creality Sermoon d1 nyomtatóval

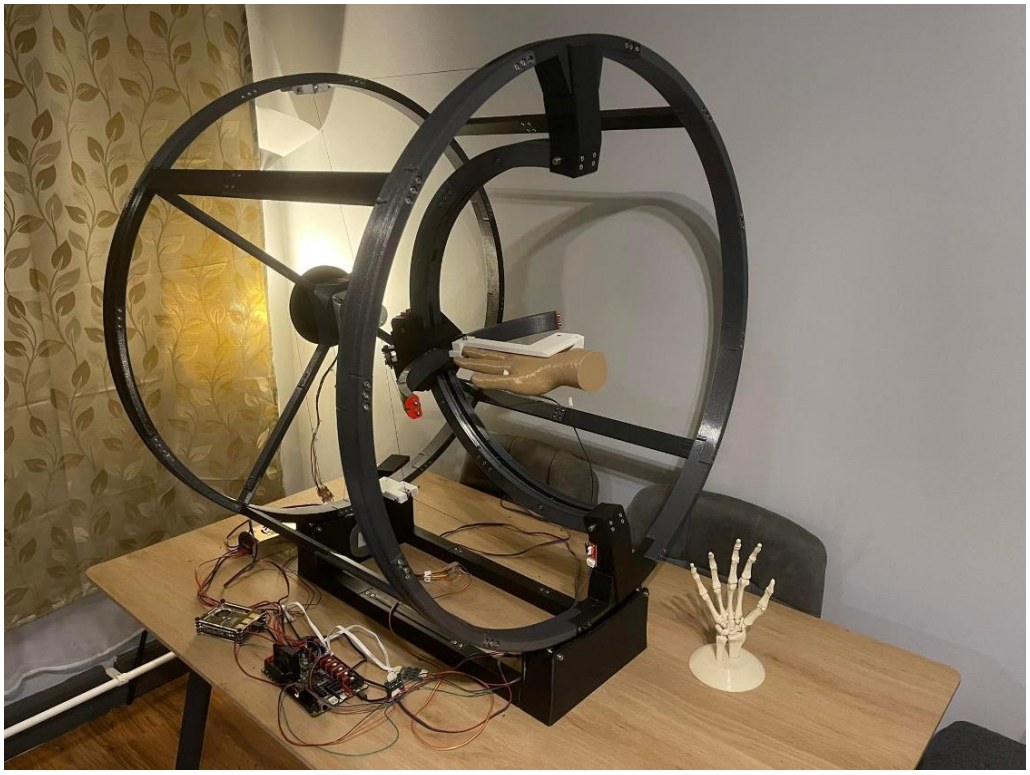
Az x-z konzolt és az y tengely körül forgató, teljeskör alkatrész bakjait Openbuilds CNC műanyag görgőkkel szereltem fel a még simább mozgás érdekében.

Az elemek összeszereléséhez rozsdamentes, belsőkulcsnyílású csavarokat használtam M3-M5 méretben, hozzájuk való rozsdamentes alátétekkel és rozsdamentes önzáró anyákkal. Ezek a kötőelemek megfelelnek az egészségügyi anyaghasználat előírásainak.

A nyomtatott alkatrészek összeszerelésével elkészült a vázszerkezet. Ezután a görgősbakokra felraktam a gépet és beszereltem a léptetőmotorokat. A kinyomtatott zsinegtárcsákat a tengelyekre szereltem. Mozgató zsinegnek horgászboltokban is kapható fonott előke zsinórt használtam. Olyan zsinórt kerestem, amelyiknek 5-8kg terhelés mellett sincs nyúlása és strapabíró. Alkatrészenként két-két zsinórt kötöttem a végükre, a másik felüket pedig ellentétesen a tárcsára csévéltem. Az zsinór megvezetésére 3D nyomtatott görgőket és GT2-6mm T20 fognélküli csapágyas feszítőgörgőket használtam.



22. ábra. Tesztnyomtatás készítése az z-x konzolhoz



23. ábra. Hyperion névre keresztelt rehabilitációs gép prototípusa

10. A vezérlés megépítése és üzembehelyezése

A gép mozgását léptetőmotorokkal végzem, de a léptetőmotorok irányításához egy vezérlő rendszert kellett tervezzek. Több lehetőséget is számításba vettem és egy olyan kétlépcsős vezérlést választottam melyet a gép továbbfejlesztéséhez is tudok majd használni.

Az elképzelés az volt, hogy a léptetőmotorok közvetlenül egy vezérlőre kötve egy általam megírt program segítségével végrehajtják a rehabilitációs mozgatásokat. Ezt számításba véve jutottam arra az opcióra, hogy egy Raspberry Pi 4 B mikrovezérlőre (24. ábra.) kötök USB csatlakozási pontján keresztül 1 db BIGTREETECH Octopus Pro 3D nyomtatáshoz gyártott alaplapot (25. ábra.**Hiba! A hivatkozási forrás nem található.**), amelyre léptetőmotorokat és végálláskapcsolókat csatlakoztatok. Jelen esetben csak a csukló mozgatását kell elvégeznem ezért elég 1db ilyen Octopus Pro alaplap, amivel tudok 8db léptetőmotort és 16db végálláskapcsolót vezérelni. A gép fejlesztési szándékával is számolva választottam a Raspberry Pi-t, mert így akár lehetőségem van 4db ilyen alaplapot csatlakoztatni ehhez a mikrovezérlőhöz és így 32db léptetőmotort alkalmazni egy időben.

24. ábra. A Raspberry Pi 4 8GB-os modellje [22]

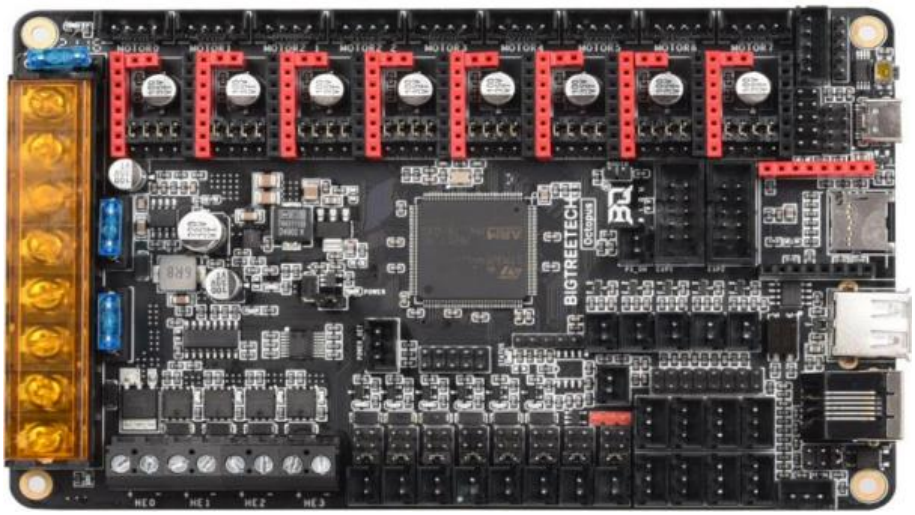
A Raspberry Pi-t évekkel ezelőtt oktatási és hobbi célokra hozták létre. Ez a zsebben elférő kicsi számítógép a megjelenése után nagyon gyorsan terjedt el a hobbi felhasználók körében és mára igen komoly felhasználói bázisa lett.

Nagy előnye, hogy sok LINUX alapú operációs rendszerrel lehet használni és hogy alacsony az áramfelvétele. A Raspberry Pi 4 B-s modellje már egy USB-C végződésű tápegységgel használható. Azaz egy 5.1V/3.0A tápegység el tudja látni alap esetben a működését. A processzora egy 4 magos, 1,5GHz-es ARM Cortex-A72, melyet egy VideoCore VI grafikus kártyával és 8GB RAM-mal együtt raktak az alaplapra. A videós teljesítménye mára már utolérte a mai elvárásokat, ugyanis a 2 db HDMI csatlakozójával 2 külön monitort is ki tud szolgálni, 4k felbontásban 30fps-sel vagy 1 monitort 4k felbontásban 60fps-sel. Bár számomra ennek nincs jelentősége, mert nem használtam ki a gép vezérléséhez, de hasznos lehetőség, ha a rehabilitációs gép későbbi fejlesztését veszem számításba. Beépített túlmelegedési védelemmel van ellátva, ami azt jelenti, hogy ha a beépített processzor üzemelés közben eléri a kritikus hőmérsékleti értéket akkor visszább vesz a teljesítményből, hogy fenn tudja tartani a folyamatos működést. Ez annyit jelent, hogy a magasabb óra jelet - 1,5GHz-et - lentebb skálázza 1,2-1,3 GHz-re, így a rövidtávú de magas teljesítményt leváltja a hosszabban fenntartható, alacsonyabb, de stabilabb teljesítmény. [22]

A Raspberry Pi vezérlővel együtt vásároltam egy microSD memória kártyát, amire előre telepítve van a saját operációs rendszere, de ezt nem használtam. Helyette egy egyszerűsített LINUX alapú Raspbian operációs rendszert installáltam rá, amibe már integrálva van egy webszerver és a Klipper nevű szoftver.

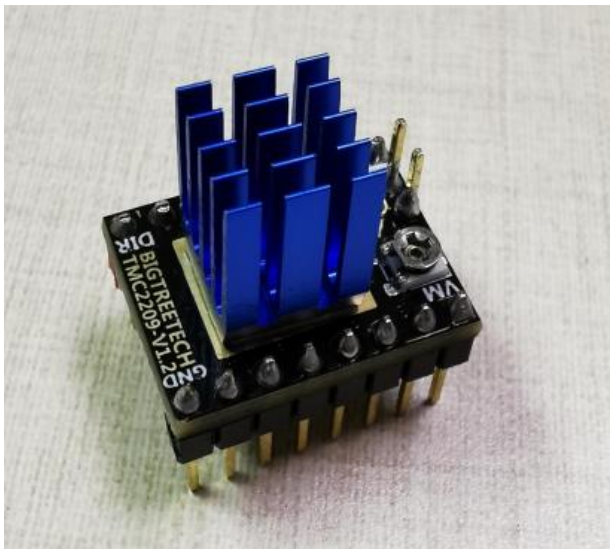
A Pi és a számítógépem kapcsolatát az otthoni hálózaton keresztül, a Kitty nevezetű szoftverrel végeztem el. Ennek segítségével beállítottam, hogy a Pi-t IP címén keresztül is el tudjam érni wifis kapcsolat esetén egy asztali böngészőből. Ezzel a kapcsolódási formával elérem a Klipper kezelőfelületét. Az IP cím alapú elérési metódusnak köszönhetően ugyanerre a hálózatra csatlakoztatott okostelefon vagy tablet segítségével is fel tudok csatlakozni az interfészre.

A BIGTREETECH Octopus Pro tápellátása USB-C-n keresztül lehetséges, vagy külön tápegységgel. USB-C-n keresztül történő áramellátás esetén az alaplap csak konfigurálásra használható és mivel én léptetőmotorokat működtetek ezzel a lappal, ezért egy 24V-os és 360W-os tápegységet szereltem rá.



25. ábra. BIGTREETECH Octopus Pro alaplap [23]

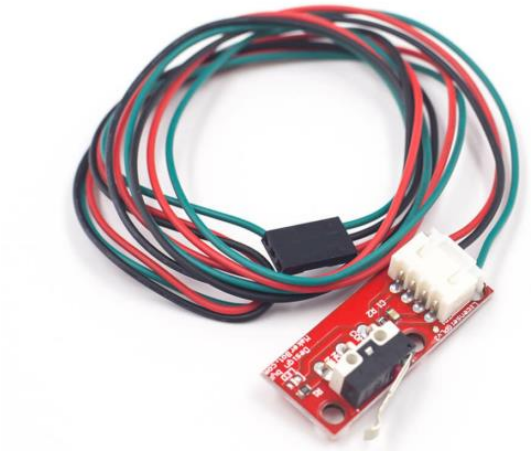
Ezután USB-C kábellel összekötöttem a Pi-t és az Octopus Pro alaplapot. Telepítettem a működéshez szükséges illesztőszoftvereket. Telepítettem a vezérléshez elengedhetetlen Mainsail-t, ami a Pi-n futó webes kezelőfelület, valamint a Moonraker szoftvert, ami kapcsolatot teremt az Octopus-on futó Klipper és a kezelőfelület között. Ez alapvetően egy nyomtatókhoz gyártott szoftver csomag, de mára már kinőte magát robotikai vezérlések alkalmazásához is. A vezérlés beállításához a Raspberry-n, a printer.cfg konfigurációs fájlban kell python programozási nyelven deklarálni, az alaplapra kötött hardvereket és meghívni az ehhez szükséges könyvtárakat.



26. ábra. BIGTREETECH TMC2209 Léptetőmotor meghajtó

A motorok működtetéséhez elengedhetetlen a motorvezérlő chipek alkalmazása. Ezeket a chipeket az Octopus alaplapba kell illeszteni a motor portokon keresztül. A választásom a TMC2209-es vezérlőkre esett. Teljesítményükben nem tudják kiszolgálni a motorokat teljes volumenben, hiszen maximum 2,2 volton 2 ampert tudnak továbbítani egyesével a motoroknak. Ezt a TMC chipeken lévő potenciométerrel állítottam erre a maximum értékre.

Ez azt jelenti, hogy a motorok forgatónyomatéka arányos a vezérlők által biztosított áramerősséggel. A kisebb motorom, a NEMA17-es így közel 100%-on tud dolgozni, jelentős teljesítmény csökkenése nincs. A L31RFD-00N-NN-00 motorok forgató nyomatéka viszont közel a harmadára esik vissza. Mivel a rajtuk lévő tárcsákat túlméretezve számoltam ki, így a gép mozgatásához szükséges nyomatékokat kisebb zsinegtárcsákkal képesek biztosítani. Mivel a TMC2209-es chipeket 100%-on kihasználom, szükséges volt a melegedés miatt egy ventilátort kötnöm az Octopus alaplapra, ami hűti a chipeket.



27. ábra. Végálláskapcsoló [24]

A motorok mozgatásához szükség van motoronként egy referenciapont felvételére. A lépéseinek kezdetén ez a referenciapont lesz a nulla pont, innen indulnak a motorok által mozgatott alkatrészek mozgatása pozitív irányba a meghatározott koordinátákra. Ezen pontok felvételére végálláskapcsolókat használtam. Ezeket az egyszerű kapcsolókat az Octopus Pro alaplapba kellet kötnöm.

11. A printer.cfg konfigurációs fájl létrehozása

Miután kész voltam a gép megépítésével és a vezérlésem összeállításával is következett a vezérlés programsoros megírása. Ehhez a Klipper weboldalán található python programozási nyelven írt makrókat és parancsokat kerestem ki és illesztettem a printer.cfg konfigurációs fájlba. Már korábban említettem, hogy bár ez nem egy nyomtató, de a Klipper szoftver üzemeléséhez ilyen névvel ellátott fájlt kell létrehozni, mert ezt keresi az alap szoftver csomag a működéshez.

Először deklarálnom kellett a mikrokontrollert, a Raspberry-m egyedi azonosító linkjével. Implementáltam az alap működéshez szükséges g-code parancsokat, hogy később a konzolba való beíráskor tudjam használni őket. Ezek a parancsok a PAUSE, mint megállítás, RESUME, mint folytatás és CANCEL_PRINT, ami teljesen megszakítja és leállítja a futó g-code programot.

Deklarálnom kellett egy nyomtatót, aminél megadtam a CARTESIAN nyomtató típust, ami annyit jelent, hogy egy egyszerű nyomtatót hoztam létre Descartes-féle, derékszögű mozgási elven működő nyomtatót. Ennek a gépem létrehozásánál nincs jelentősége, de a működéshez meg kellett adni egy típust. A nyomtató motorjainak meghatároztam a MAX_ACCEL: 50 paranccsal, hogy maximum 50mm/s^2 lehet a motorok gyorsulási értéke. Valamint a maximális sebességet MAX-VELOCITY: 100 értékre állítottam, hogy maximum 100 egységnyi sebességgel mozoghatnak a motorok.

A motorok deklarálását a vezérlőchipek lábainak helyzetével hivatkoztam meg. Minden TMC2209 chip lábát függvényekkel kötöttem össze a hozzájuk tartozó motor csatlakozási pontjaival. Miután az összerendelések készen voltak, beállítottam a motorok forgási útját, ami azt jelenti, hogy egy fordulat alatt, azaz 200 teljes lépés alatt hány mm-t mozognak a hozzájuk tartozó alkatrészek. Ezeket kezdetben a tárcsáik kerületével számoltam ki, mivel zsinóros mozgatást alkalmazok, ezért annyi mm-t mozdul az alkatrész amennyi mm-t le- vagy felteker a zsinegtárcsa. Ugyan így tettem a GT2 szíjhajtásos motornál is. A motorok beállításánál határoztam meg a maximálisan megtehető utak hosszát is, hogy ne menjenek ütközésig az alkatrészek.

Minden motorhoz tartozó végálláskapcsolót is deklaráltam, azaz meghivatkoztam a motorok függvényeiben, a végálláskapcsolók lábainak Octopus alaplapra való kapcsolódási pontjaival. (2. sz. melléklet)

12. Tesztmozgások és léptetőmotor kalibrálások

Az alap vezérlési rendszer felállítása után a konzolon keresztül g-code-ok beírásával alap mozgatásokat kezdtem tesztelni. Minden motor első indulásakor kötelező, hogy a hozzátársított végálláskapcsolónak ütközzön kétszer, ezzel felvéve a tengely referenciapontját.

Első körben a referenciapont felvételén kellet finomítanom, mert elsőre mikor nekiütközött nem húzta el kellően az ütköztetett alkatrészeket a végálláskapcsolóktól, így azok beragadási hibát jeleztek vissza. A printer.cfg fájlban a visszahúzási értékeken kellett növeljek, mert bár nyúlásmentes zsinórt használok, az alkatrészek a zsinór megfeszülésével egy kicsit összébb mozdultak így a visszahúzott alkatrész nem tudott rögtön akkora távot visszamozdulni, hogy leálljon a végálláskapcsolóról. Miután ezt kalibráltam a referenciapontjaimat fel tudtam venni az x és a z tengely mentén.

Az y tengelynél a GT2 szíjjal olyan hibára futottam, hogy a már említett zsinór megfeszülés következtében a két szíjtárcsa közelebb került egymáshoz, hozzávetőlegesen 1-2mm-rel, ami azt eredményezte, hogy a szíj nem tudott a szíjtárcsa fogaiba kapaszkodni. Ezzel már a tervezés során kalkuláltam, így egyszerűen át tudtam alakítani ezt a mozgatást is zsinóros formátumra. Ezután tökéletesen működött a végálláskapcsoló elérése és a referenciapont felvétele az y tengelynél is.

A következő az alkatrészek mozgásának kalibrálása volt. G-code vezérléssel 100mm-rel mozgattam a motorokat a G1 X+100 Y+100 Z+100 kóddal és lejegyeztem az eltéréseket. Ezeket az értékeket a motorok eltérő működése és a már korábban taglalt zsinegek megfeszülése miatt nem lehetett kiszámolni, ezért kellett manuálisan kalibrálnom ezeket a mozgási hosszokat, majd a printer.cfg-ben az adott motorhoz tartozó ROTATION_DISTANCE kódrészlet után írnom.

A motorok forgási iránya számomra pont megfelelő volt, de ha a forgás másik irányára lett volna szükségem, akkor a motorok deklarálásánál a DIR_PIN: utáni bekötési lábhivatkozás negálásával a forgatási irányt az ellenkezőjére tudtam volna módosítani.

Ezek után a konzolon keresztül teszteltem a mozgatási sebességeket g-code-ok beírásával. A G1 X+100 Y+100 Z+100 F500 parancsot teszteltem. Az F értéket mindig változtatva a kéz mozgatásához szükséges kényelmes, lassú sebességre. Végül az F300-F500 sebességeket találtam az ideálisnak, de ezek természetesen tovább csökkenthetőek.

13. Alap csuklómozgatási program megírása programmal

A rehabilitációs mozgatásokat g-code-ok bevitelével valósítottam meg. A Klipper kezelőfelületen lehetőség van soronkénti g-code bevitelre, de ezzel összetett mozgásokat levezérelni a páciens mozgáshatáraihoz képest nem lehetséges. Minden páciens más és más korlátokkal rendelkezik.

Ezért egy olyan program létrehozásán gondolkodtam, amivel a beteg korlátainak megfelelően lehet g-code mozgatási programokat generálni. A java programozási nyelvet valamilyen szinten már ismertem ezért ezen fejlesztési környezetben terveztem megírni a g-code generátor programomat.

Telepítettem a java programíráshoz szükséges fejlesztői környezetet és nekiláttam a tervezéshez.

Minden páciens egyedi azonosítót kap így későbbi rehabilitációs alkalmak során elegendő ugyan azt a fájlt az Octopusnak továbbítani és elindítani. A fájl létrehozása során a program bekérdezi a páciensazonosítót és a csuklója tengelyenként megengedett maximális szögelfordulást pozitív és negatív irányokban, valamint 1-10-es skálán a mozgatási sebességet. Ismerve a körívpályák sugarát, ezen szögeket arányosan mm-be átszámolt szakaszokra váltottam.

A csukló egy - megközelítőleg - félgömb felszínét járja végig a középsőujj végével. Ezen felszínen kalkulálja ki a java nyelven írt programom a betáplált intervallumon belül a mozgási pályát.

A fájl létrehozása során ellenőrzi a program, hogy a fájl létrehozható-e, nem ütközik-e hibába az elérési út és elnevezési azonosságok alapján. Amikor létrehozta az üres fájlt az egyedi azonosítóval, a generált kód exportálásra kerül a fájlba. (3. sz. melléklet)

A g-code-ot a G28 parancs kezdi, ez minden motort nullpont felvételére irányít. Ezután a G91 kód következik, ami azt jelenti, hogy növekményes méretmegadással történik a tengelyek léptetése. Tehát nem a nullponthoz képest léptet a program, hanem az aktuális pozícióhoz képest. Ezek után következik a léptetések felsorolása G1 utasítással, azaz mozgatás egyenes vonal mentén, adott előtolási sebességgel. Az előtolási sebességet 1-10es skálán lehet megadni a programban, ami annyiszor F50-értéket jelent amekkora értéket megadtunk. A generált fájl (4. sz. melléklet) a Klipperbe importálható és így indítható el a rehabilitációs tornáztatás.

14. Összefoglalás

Projektem során megismerkedtem a kéz anatómiájával és elsajátítottam a stroke betegség ismereteit. Megterveztem és megépítettem egy olyan kéz rehabilitációs gép prototípusát, amely alapköve lehet egy újtípusú rehabilitációs gép kifejlesztésének.

A projekt megvalósítását kutatásaimra támaszkodva hajtottam végre, ügyelve az egészségügyi alkalmazhatóságra, sterilizálhatóságra. Elemeztem a csukló mozgási mechanizmusait és szögeit, ezeket implementáltam a készülékem alapelveibe. A gép képes a csuklót mindhárom meghatározott tengelye mentén mozgatni g-code-ok segítségével. A kitűzött célomhoz mérten sikeresen megvalósítottam a vezérlés kivitelezését. A páciensek szükségleteinek megfelelően kifejlesztettem egy olyan programot, amely képes g-code-okat generálni, amelyeknek lefuttatása pontosan mozgatja a kezet a csukló tengelyei mentén. A gép működése által kibocsátott hangok a várthoz képest jelentősen halkabbak, így egy rehabilitációs kezelést tekintve egyáltalán nem számít zavaró tényezőnek.

Munkámat a kitűzött feladatok szempontjából eredményesnek értékelem. A célomat elértem és megvalósítottam. A kezet körül ívelő alkatrészek a Szaturnusz gyűrűire emlékeztetnek, ezért nevét a Szaturnusz egyik holdjáról a Hyperionról kapta.

A géppel kapcsolatos jövőbeni terveim között szerepel a Hyperion továbbfejlesztése, valamint egy rehabilitációs gépekre kiírt kutatás-fejlesztéses pályázat megcélzása. Fejlesztési céljaim között szerepel a gép erősebb anyagból való megépítése, hogy stabilabb és masszívabb legyen a szerkezet. Minden alkatrészem feldarabolva került előállításra, ám fontos lenne, hogy ezek az elemek 1 munkadarabból álljanak. Az y tengelynél a forgatási mechanizmust leváltanám egy, a GT2-nél magasabb bordázott szíjra, hogy stabilabb legyen a nyomatékátvitel. A gépet a továbbiakban tervezem kiegészíteni az ujjak rehabilitációs mozgatásával. Ezeknek az elemeknek kellően elegendő helyet hagytam a tervezés során. Az ujjakat az utolsó ujjpercet rögzítve sínekben fogom mozgatni, rugó és zsineg segítségével.

A komolyabb rehabilitációs eredmények elérése érdekében egy virtuális valóság szemüveggel tervezem a gépet szinkronba hozni. Számos kutatás alátámasztja, hogy az idegek, izmok és velük egyidejűleg a neuronhálózatok stimulálása adja a leghatékonyabb eredményeket a stroke által lebénult betegek kezelése során.

15. Summary

The goals of my thesis are the planning and building of a rehabilitation machine for those who had stroke. The machine shall provide support for the stuck and paralyzed wrist to perform the rehabilitational exercises. In this broad topic my focus is on the movement of wrist joint.

I started my thesis writing process by researching the necessary articles and documentations.

I have looked up the side-effects and symptoms of stroke, the used methods of rehabilitation and basic anatomical knowledge of the hand, mainly its range of motion and medical conditions.

During my own work I have determined the general movement range of the wrist. I have used this information to plan the machine. To help myself with the design process I have 3D scanned my own arm so I can use it as a template. During the this process I have determined the most crucial dimensions and made a jig to hold the arm. I also the designed the parts necessary for the movement of the arm. By summarising my amassed knowledge and research in the topic I have started the building of machine.

I used 3D printers to fabricate the parts of the machine. I used PETG as material as it is suitable to the medical requirements. To make the parts fit my 3d printers I have used a CAD modelling software to cut them up in smaller parts. I continued my thesis work after printing all the parts by assembling the electrical components.

For the controlling of the machine, I choose a Raspberry Pi and a motherboard for 3D printers. I have configured the necessary operating system and software for the Raspberry Pi and made sure to test and review the firmware parameters.

Last but not least I have developed software in Java to generate the required movements for the rehabilitation. This software generates g-code files that contain the necessary movements according the maximum angles already determined. Using this method each patient can receive a personalised and unique treatment. By saving the g-code files the treatment is easily repeatable at a later date.

16. Irodalomjegyzék

[1] Dr. Farkas Eszter: A stroke betegség során fellépő agyi sérülések okai 2016.

[2] Dermolex, https://dermolex.com/hu/blog/sztrok (megtekintve: 2022.09.30.)

[3] www.braintrainingcentre.com
https://www.braintrainingcentre.com.au/stroke (megtekintve: 2022.11.06.)

[4] www.saebo.com
https://www.saebo.com/blog/saebo-hand-rehabilitation-device-right/
(megtekintve: 2022.11.6.)

[5] Denisz Kovács, Steps 07.04.2020.
https://stepsbudapest.com/hu/robottechnologia-es-a-stroke/ (megtekintve: 2022.09.30.)

[6] Dr. Donáth Tibor: Anatómiaélettan, Medicina Könyvkiadó Rt. Budapest, 2005.

[7] Prof. Dr. Renner Antal: A csukló régió anatómiája, www.semmelweis.hu, 2019.
https://semmelweis.hu/traumatologia/files/2019/12/14_A-CSUKLÓ-RÉGIÓ-ANATÓMIÁJA.pdf (megtekintés:2022.04.30.)

[8] Ismeretlen, www.igus.hu, https://www.igus.hu/info/3d-printing-processes (megtekintés: 2022.11.03.)

[9] Dezső Renáta, www.freedee.hu, 4. október 2022. https://www.freedee.hu/3d-nyomtatas-az-egeszsegugyben/ (megtekintés:2022.11.05.)

[10] Dr. habil. Kalácska Gábor, www.quattroplast.hu, 8. január 2009.
https://quattroplast.hu/sites/default/files/attachments/a-szakterulet-kulonlegessegei-alapfogalmak.pdf (megtekintés:2022.11.05.)

[11] TME Electronic Components, Léptető motorok működése, TME Hungary Kft., 2020.09.08.
https://www.tme.eu/hu/news/libraryarticles/page/41861/Leptetomotorok-tipusok-es-alkalmazasok/
(megtekintés:2022.03.23.)

[12] ElectroRules Blog: The Complete Guide to Stepper Motor, 26 augusztus 2021. https://www.electrorules.com/the-complete-guide-to-stepper-motor/. (megtekintés:2022.05.23.)

[13] Zsuffa Attila: Villamos hajtások és mozgásvezérlők, Q-Tech Léptetőmotor 10. 2006. https://q-tech.hu/wp-content/uploads/2018/02/Villamos-hajtasok-es-mozgasvezerlok-4.pdf. (megtekintés:2022.05.23.)

[14] www.all3dp.com, https://all3dp.com/2/what-s-a-stepper-motor-driver-why-do-i-need-it/ (megtekintve:2022.11.06. 6.)

[15] Ismeretlen www.javatpoint.com, https://www.javatpoint.com/programming-language. (megtekintés:2022.11.06.)

[16] Fred Racapé, www.agilenative.com, 21 január 2017. https://www.agilenative.com/2017/01/hello-world/. (megtekintés:2022.11.06.)

[17] https://hu.gadget-info.com, 2019. https://hu.gadget-info.com/difference-between-syntax. (megtekintés:2022.11.06.)

[18] Caxtool, https://www.caxtool.hu/nema17_stepper_motor_42byghw811_48mm_long25a_720mm_vezetekkel_42331

[19] Ismeretlen, https://joy-it.net/files/files/Produkte/NEMA17-06/NEMA17-06_Datasheet.pdf (megtekintés:2022.09.05.)

[20] Adolf Frischherz: Fémtechnológiai táblázatok Budapest, B+L Lap- és Könyvkiadó Kft. 1997, 2021.

[21] Ismeretlen, https://ahs-antriebstechnik.de/images/PDFDateien/Service/L3-GB-0107.pdf (megtekintés:2022.09.05.)

[22] Pi-shop www.pi-shop.hu, https://www.pi-shop.hu/raspberry-pi-4-model-b-8gb (megtekintés:2022.10.02.)

[23] Ismeretlen, www.3djake.com, https://c-3d.niceshops.com/upload/file/BIGTREETECH_Octopus_EN.pdf [megtekintés: 2022.10.28.)

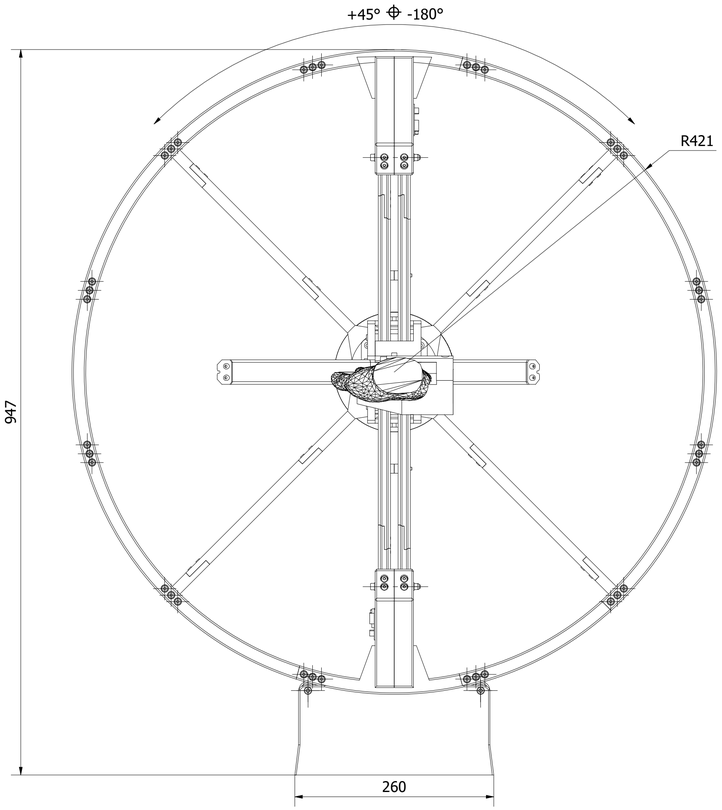
[24] Caxtool, https://www.caxtool.hu/vegallaskapcsolo_a_70cm_vezetekkel_43551 (megtekintés:2022.10.29.)

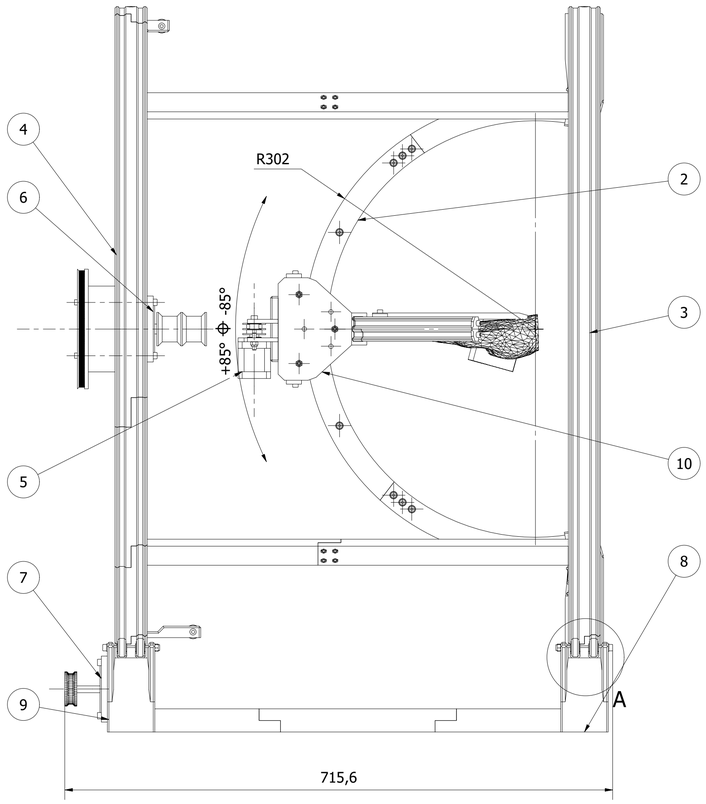
[25] www.tyromotion.com, (megtekintés:2022.11.06.)

[26] 3djake, https://www.3djake.hu/bigtreetech/stepper-motor-driver?sai=9875&gclid=Cj0KCQiA4aacBhCUARIsAI55maEaOuUuUjeP2qwpcb7_Kv8foTMBbT3ChVB-5NZOs-R8HVm3kksyzA0aAhKaEALw_wcB (megtekintés:2022.10.29.)

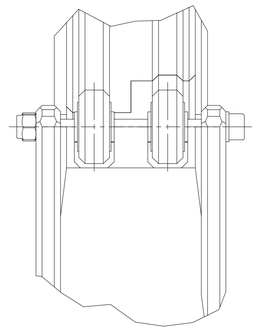
17. Mellékletek listája

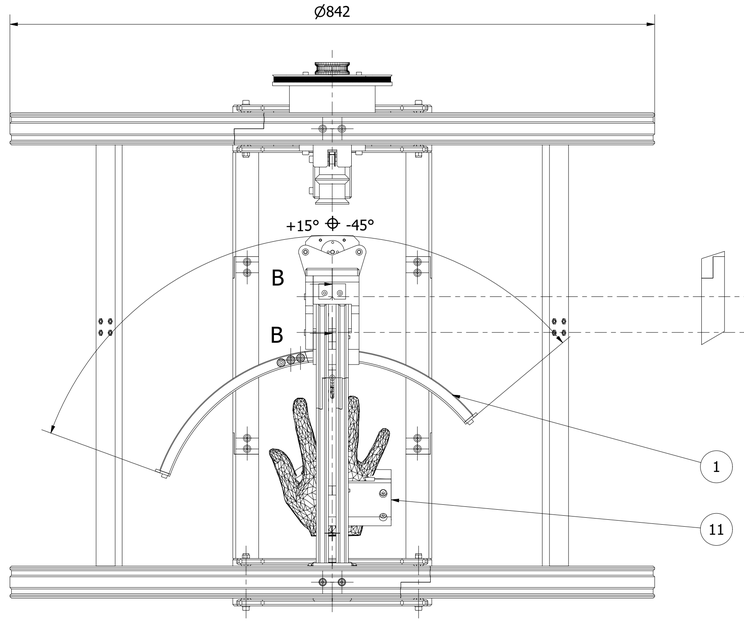
1. sz. melléklet: A rehabilitációs gép összeállítási rajza
2. sz. melléklet: A printer.cfg konfigurációs fájl
3. sz. melléklet: A java nyelven megírt g-code generátor
4. sz. melléklet: G-code generátorral létrehozott példaprogram



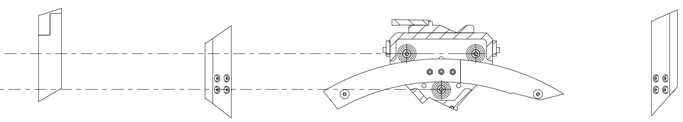


A (1 : 1)





B-B (1 : 4)



Darabjegyzék		
Tételszám	Darabszám	Tétel megnevezése
1	1	Z ív
2	1	X ív
3	1	Y1 gyűrű
4	1	Y2_gyűrű
5	1	Z léptetőmotor
6	1	X léptetőmotor
7	1	Y léptetőmotor
8	1	Első talp
9	1	Hátsó talp
10	1	X-Z konzol
11	1	Kézbefogó konzol

Tervezte	Neptun kód		Lapméret	Dátum	
Turner Máté	GE3O5R	1. sz. melléklet	A1	2022.12.15.	
Óbudai Egyetem - Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar		Hyperion - Kézrehabilitációs gép			
		Méretarány: 1:4	Verzió 1.0	Sheet 1 / 1	

2. sz. melléklet

A printer.cfg konfigurációs fájl

```
[mcu]
serial:/dev/serial/by-id/usb-Klipper_stm32f429xx_34003E001451303439373431-if00
#   The serial port to connect to the MCU. If unsure (or if it
#   changes) see the "Where's my serial port?" section of the FAQ.
#   This parameter must be provided when using a serial port.
baud: 250000
#   The baud rate to use. The default is 250000.
restart_method: command
#   This controls the mechanism the host will use to reset the
#   micro-controller. The choices are 'arduino', 'cheetah', 'rpi_usb',
#   and 'command'. The 'arduino' method (toggle DTR) is common on
#   Arduino boards and clones. The 'cheetah' method is a special
#   method needed for some Fysetc Cheetah boards. The 'rpi_usb' method
#   is useful on Raspberry Pi boards with micro-controllers powered
#   over USB - it briefly disables power to all USB ports to
#   accomplish a micro-controller reset. The 'command' method involves
#   sending a Klipper command to the micro-controller so that it can
#   reset itself. The default is 'arduino' if the micro-controller
#   communicates over a serial port, 'command' otherwise.
[display_status]

[force_move]
enable_force_move: True

[virtual_sdcard]
path:/home/Lechadias/printer_data/gcodes
#   The path of the local directory on the host machine to look for
#   g-code files. This is a read-only directory (sdcard file writes
#   are not supported). One may point this to OctoPrint's upload
```

```
#   directory (generally ~/.octoprint/uploads/ ). This parameter must
#   be provided.
#on_error_gcode:
#   A list of G-Code commands to execute when an error is reported.

[pause_resume]

[gcode_macro PAUSE]
description: Pause the actual running print
rename_existing: PAUSE_BASE
gcode:
  PAUSE_BASE
  _TOOLHEAD_PARK_PAUSE_CANCEL

[gcode_macro PAUSE]
description: Pause the actual running print
rename_existing: PAUSE_BASE
# change this if you need more or less extrusion
variable_extrude: 1.0
gcode:
  ##### read E from pause macro #####
  {% set E = printer["gcode_macro PAUSE"].extrude|float %}
  ##### set park positon for x and y #####
  # default is your max posion from your printer.cfg
  {% set x_park = printer.toolhead.axis_maximum.x|float - 5.0 %}
  {% set y_park = printer.toolhead.axis_maximum.y|float - 5.0 %}
  ##### calculate save lift position #####
  {% set max_z = printer.toolhead.axis_maximum.z|float %}
  {% set act_z = printer.toolhead.position.z|float %}
  {% if act_z < (max_z - 2.0) %}
     {% set z_safe = 2.0 %}
  {% else %}
```

```
    {% set z_safe = max_z - act_z %}
  {% endif %}
  ##### end of definitions #####
  PAUSE_BASE
  G91
  {% if printer.extruder.can_extrude|lower == 'true' %}
   G1 E-{E} F2100
  {% else %}
   {action_respond_info("Extruder not hot enough")}
  {% endif %}
  {% if "xyz" in printer.toolhead.homed_axes %}
   G1 Z{z_safe} F900
   G90
   G1 X{x_park} Y{y_park} F6000
  {% else %}
   {action_respond_info("Printer not homed")}
  {% endif %}

[gcode_macro RESUME]
description: Resume the actual running print
rename_existing: RESUME_BASE
gcode:
  ##### read E from pause macro #####
  {% set E = printer["gcode_macro PAUSE"].extrude|float %}
  #### get VELOCITY parameter if specified ####
  {% if 'VELOCITY' in params|upper %}
   {% set get_params = ('VELOCITY=' + params.VELOCITY)  %}
  {%else %}
   {% set get_params = "" %}
  {% endif %}
  ##### end of definitions #####
  {% if printer.extruder.can_extrude|lower == 'true' %}
```

```
   G91
   G1 E{E} F2100
  {% else %}
   {action_respond_info("Extruder not hot enough")}
  {% endif %}
  RESUME_BASE {get_params}

[gcode_macro CANCEL_PRINT]
description: Cancel the actual running print
rename_existing: CANCEL_PRINT_BASE
variable_park: True
gcode:
  ## Move head and retract only if not already in the pause state and park set to true
  {% if printer.pause_resume.is_paused|lower == 'false' and park|lower == 'true'%}
   _TOOLHEAD_PARK_PAUSE_CANCEL
  {% endif %}
  TURN_OFF_HEATERS
  CANCEL_PRINT_BASE

[gcode_macro _TOOLHEAD_PARK_PAUSE_CANCEL]
description: Helper: park toolhead used in PAUSE and CANCEL_PRINT
variable_extrude: 1.0
gcode:
  ##### set park positon for x and y #####
  # default is your max posion from your printer.cfg
  {% set x_park = printer.toolhead.axis_maximum.x|float - 5.0 %}
  {% set y_park = printer.toolhead.axis_maximum.y|float - 5.0 %}
  {% set z_park_delta = 2.0 %}
  ##### calculate save lift position #####
  {% set max_z = printer.toolhead.axis_maximum.z|float %}
  {% set act_z = printer.toolhead.position.z|float %}
  {% if act_z < (max_z - z_park_delta) %}
```

```
    {% set z_safe = z_park_delta %}
  {% else %}
    {% set z_safe = max_z - act_z %}
  {% endif %}
  ##### end of definitions #####
  {% if printer.extruder.can_extrude|lower == 'true' %}
    M83
    G1 E-{extrude} F2100
    {% if printer.gcode_move.absolute_extrude |lower == 'true' %} M82 {% endif %}
  {% else %}
    {action_respond_info("Extruder not hot enough")}
  {% endif %}
  {% if "xyz" in printer.toolhead.homed_axes %}
    G91
    G1 Z{z_safe} F900
    G90
    G1 X{x_park} Y{y_park} F6000
    {% if printer.gcode_move.absolute_coordinates|lower == 'false' %} G91 {% endif %}
  {% else %}
    {action_respond_info("Printer not homed")}
  {% endif %}

[printer]
kinematics: cartesian
max_velocity: 100
max_accel: 50
#max_z_velocity: 5
#max_z_accel: 100



[stepper_z] # valójában ez a z tengyely deklarációja
step_pin: PF13
```

```
#   Step GPIO pin (triggered high). This parameter must be provided.
dir_pin: PF12
#   Direction GPIO pin (high indicates positive direction). This
#   parameter must be provided.
enable_pin: !PF14
#   Enable pin (default is enable high; use ! to indicate enable
#   low). If this parameter is not provided then the stepper motor
#   driver must always be enabled.
rotation_distance: 63
#   Distance (in mm) that the axis travels with one full rotation of
#   the stepper motor (or final gear if gear_ratio is specified).
#   This parameter must be provided.
microsteps: 16
#   The number of microsteps the stepper motor driver uses. This
#   parameter must be provided.
full_steps_per_rotation: 200
#   The number of full steps for one rotation of the stepper motor.
#   Set this to 200 for a 1.8 degree stepper motor or set to 400 for a
#   0.9 degree motor. The default is 200.
#gear_ratio:
#   The gear ratio if the stepper motor is connected to the axis via a
#   gearbox. For example, one may specify "5:1" if a 5 to 1 gearbox is
#   in use. If the axis has multiple gearboxes one may specify a comma
#   separated list of gear ratios (for example, "57:11, 2:1"). If a
#   gear_ratio is specified then rotation_distance specifies the
#   distance the axis travels for one full rotation of the final gear.
#   The default is to not use a gear ratio.
#step_pulse_duration:
#   The minimum time between the step pulse signal edge and the
#   following "unstep" signal edge. This is also used to set the
#   minimum time between a step pulse and a direction change signal.
#   The default is 0.000000100 (100ns) for TMC steppers that are
```

```
#   configured in UART or SPI mode, and the default is 0.000002 (which
#   is 2us) for all other steppers.
endstop_pin: !PG6
#   Endstop switch detection pin. If this endstop pin is on a
#   different mcu than the stepper motor then it enables "multi-mcu
#   homing". This parameter must be provided for the X, Y, and Z
#   steppers on cartesian style printers.
position_min: 0
#   Minimum valid distance (in mm) the user may command the stepper to
#   move to.  The default is 0mm.
position_endstop: 0
#   Location of the endstop (in mm). This parameter must be provided
#   for the X, Y, and Z steppers on cartesian style printers.
position_max:315
#   Maximum valid distance (in mm) the user may command the stepper to
#   move to. This parameter must be provided for the X, Y, and Z
#   steppers on cartesian style printers.
homing_speed: 5.0
#   Maximum velocity (in mm/s) of the stepper when homing. The default
#   is 5mm/s.
homing_retract_dist: 20.0
#   Distance to backoff (in mm) before homing a second time during
#   homing. Set this to zero to disable the second home. The default
#   is 5mm.
#homing_retract_speed:
#   Speed to use on the retract move after homing in case this should
#   be different from the homing speed, which is the default for this
#   parameter
second_homing_speed:5
#   Velocity (in mm/s) of the stepper when performing the second home.
#   The default is homing_speed/2.
#homing_positive_dir:
```

```
#   If true, homing will cause the stepper to move in a positive
#   direction (away from zero); if false, home towards zero. It is
#   better to use the default than to specify this parameter. The
#   default is true if position_endstop is near position_max and false
#   if near position_min.

[stepper_x] # ez az x tengely deklarációja
step_pin: PG0
dir_pin: PG1
enable_pin: !PF15
microsteps: 16
rotation_distance: 63
endstop_pin: ^!PG12
position_endstop: 0
position_max: 896
homing_retract_dist: 20.0
homing_speed: 5

second_homing_speed:5

[stepper_y] # ez az y tengely deklarációja
step_pin: PF11
dir_pin: PG3
enable_pin: !PG5
microsteps: 16
rotation_distance: 677
endstop_pin: !PG9
position_endstop: 0
position_max: 1650
homing_retract_dist: 10.0
homing_speed: 10
```

```
[tmc2209 stepper_x]
uart_pin: PC4
diag_pin: PG6
#tx_pin:
#select_pins:
interpolate: True
run_current: 0.8
hold_current: 0.7
sense_resistor: 0.110
stealthchop_threshold: 0
#   See the "tmc2208" section for the definition of these parameters.
#uart_address:
#   The address of the TMC2209 chip for UART messages (an integer
#   between 0 and 3). This is typically used when multiple TMC2209
#   chips are connected to the same UART pin. The default is zero.
#driver_IHOLDDELAY: 8
#driver_TPOWERDOWN: 20
#driver_TBL: 2
#driver_TOFF: 3
#driver_HEND: 0
#driver_HSTRT: 5
#driver_PWM_AUTOGRAD: True
#driver_PWM_AUTOSCALE: True
#driver_PWM_LIM: 12
#driver_PWM_REG: 8
#driver_PWM_FREQ: 1
#driver_PWM_GRAD: 14
#driver_PWM_OFS: 36
#driver_SGTHRS: 0
#   Set the given register during the configuration of the TMC2209
#   chip. This may be used to set custom motor parameters. The
#   defaults for each parameter are next to the parameter name in the
```

```
#   above list.
#diag_pin:
#   The micro-controller pin attached to the DIAG line of the TMC2209
#   chip. The pin is normally prefaced with "^" to enable a pullup.
#   Setting this creates a "tmc2209_stepper_x:virtual_endstop" virtual
#   pin which may be used as the stepper's endstop_pin. Doing this
#   enables "sensorless homing". (Be sure to also set driver_SGTHRS to
#   an appropriate sensitivity value.) The default is to not enable
#   sensorless homing.

[tmc2209 stepper_y]
uart_pin: PD11
diag_pin: PG9
run_current: 0.800
stealthchop_threshold: 999999

[tmc2209 stepper_z]
uart_pin: PC6
diag_pin: PG10
run_current: 0.800
stealthchop_threshold: 999999
```

3. sz. melléklet

A java nyelven megírt g-code generátor

```
import java.io.FileWriter;
import java.util.Optional;
import java.util.Scanner;   // konzol beolvasás
import java.io.File;        // file ki-be olvasás és írás
import java.io.IOException; // hiba jelzése fájlba írás eseteén
import java.util.UUID;

public class Main {
    public static float szogbeolvasas(Scanner ertekbe,float max,float min,String tengely, String irany){
        float szog;
        boolean asd=false;

        System.out.println("Adja meg az "+tengely+" tengely menti "+irany+" szög értékét!");
        do {
            if(asd)
                System.out.println("Nem helyes intervallum!");
            szog = ertekbe.nextFloat();

            asd=true;

        }while(szog>max || szog<min);
        return szog;
    }
    public static void main(String[] args) {
```

```
        Scanner be=new Scanner(System.in);
        float x_pozitiv, x_negativ, y_pozitiv, y_negativ, z_pozitiv, z_negativ;
        float z_min=15, z_max=45, x_min=85, x_max=85, y_min=45, y_max=180;

        System.out.println("Kérem a páciens nevét és azonosítóját");
        String paciensazonosito=be.nextLine();

        System.out.println("Adja meg a sebbességet 1-10ig terjedő skálán! 1=F50
sebesség");
        float sebesség=be.nextFloat();



        z_pozitiv=szogbeolvasas(be, z_max,0,"Z", "pozitív");
        z_negativ=szogbeolvasas(be, z_min,0,"Z", "negativ");
        x_pozitiv=szogbeolvasas(be, x_max,0,"X", "pozitív");
        x_negativ=szogbeolvasas(be, x_min,0,"X", "negativ");
        y_pozitiv=szogbeolvasas(be, y_max,0,"Y", "pozitív");
        y_negativ=szogbeolvasas(be, y_min,0,"Y", "negativ");




      //System.out.println(x_pozitiv + " " + " "+ z_negativ);

      float z_szakasz=210, z_szakasz_egyseg=z_szakasz/(z_max+z_min);
      float x_szakasz=896, x_szakasz_egyseg=x_szakasz/(x_max+x_min);
      float y_szakasz=1650, y_szakasz_egyseg=y_szakasz/(y_max+y_min);

      float zMaxmm=z_szakasz_egyseg*z_pozitiv;
```

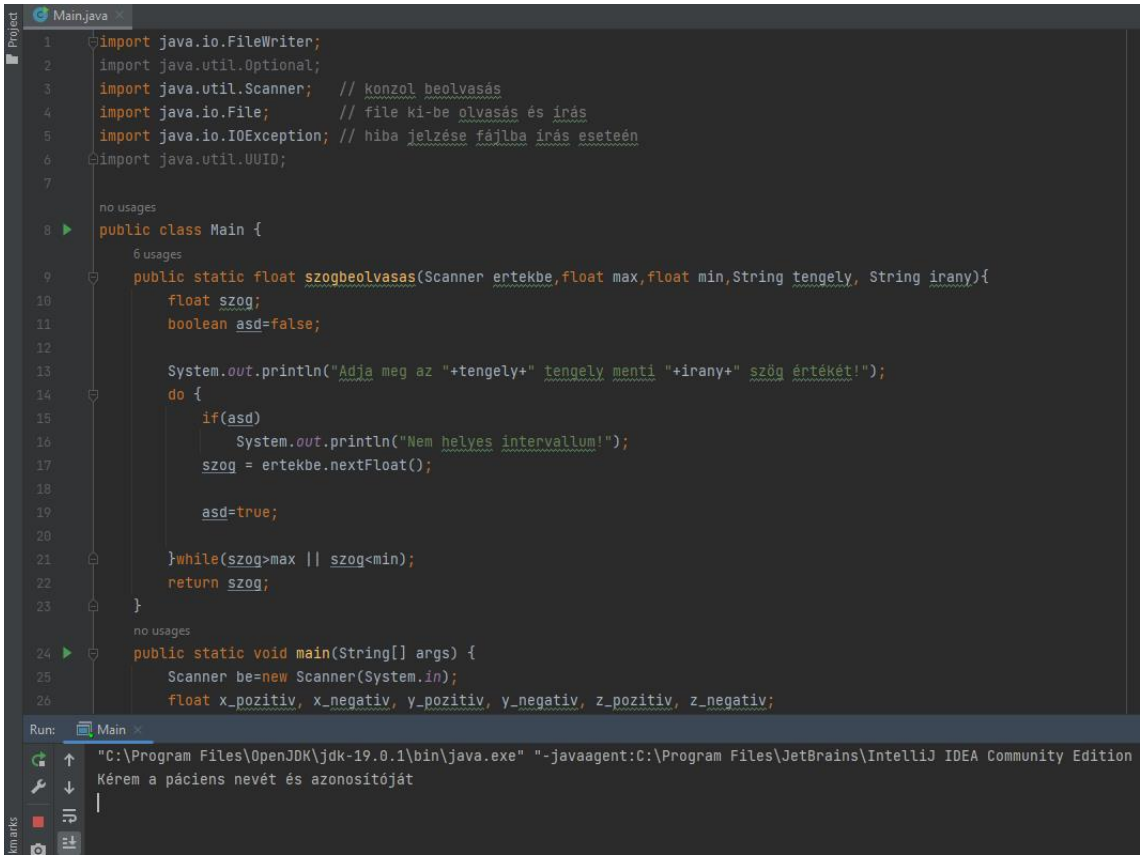
```
    float zMinmm=z_szakasz_egyseg*z_negativ;
    float xMaxmm=x_szakasz_egyseg*x_pozitiv;
    float xMinmm=x_szakasz_egyseg*x_negativ;
    float yMaxmm=y_szakasz_egyseg*y_pozitiv;
    float yMinmm=y_szakasz_egyseg*y_negativ;
    float zSzummm=zMaxmm+zMinmm;
    float xSzummm=xMaxmm+xMinmm;
    float ySzummm=yMaxmm+yMinmm;

   // System.out.println("G28\nG91\nG1 Z+"+ zMinmm + " X+" + xMinmm + " Y+" +
yMinmm);

    String fileeleresiut="E:\\gcodegenerator\\" + paciensazonosito +".txt";

    try {
       File gcodegenerator = new File(fileeleresiut); //létrehoz egy egyedi ures txt-t

       if(gcodegenerator.createNewFile()) {
          System.out.println("Fájl létrehozva: " + gcodegenerator.getName() );
       }else {
          System.out.println("A fájl már létezik!");
       }

    }catch (IOException hiba){
       System.out.println("Hiba történt");
       hiba.printStackTrace(); // kiírja melyik fájlban pontosan mi a hiba, ha nem tudja
olvasni vagy beleírni, hozzáférési hibák
    }
```

```
        try {
            FileWriter beleiro = new FileWriter(fileeleresiut); //txt-be kiírás
            /*beleiro.write("G28\nG91\nG1 Z+"+ z_max + " X+" + x_min + " Y+" + y_min
                    + "\n G1 X"


                        );*/ // nem kell most, alapból a gép középértékekre áll
            beleiro.write("G28\nG91\nG1 Z-"+zMinmm + " F"+ sebesség*50
                    + "\nG1 X+" + xMaxmm + " F"+ sebesség*50
                    + "\nG1 Z+" + zSzummm + " F"+ sebesség*50
                    + "\nG1 X-" + xSzummm + " F"+ sebesség*50
                    + "\nG1 Z-" + zSzummm + " F"+ sebesség*50
                    + "\nG1 X+" + xMinmm + " F"+ sebesség*50
                    + "\nG1 Y-" + yMinmm + " F"+ sebesség*50
                    + "\nG1 Y+" + ySzummm + " F"+ sebesség*50
            ); //alap kerület bejárás a csuklóval


            beleiro.close();
            System.out.println("Sikeres g-code generálás");



        }catch (IOException hiba){
            System.out.println("Hiba történt");
            hiba.printStackTrace(); // kiírja melyik fájlban pontosan mi a hiba, ha nem tudja olvasni vagy beleírni, hozzáférési hibák
        }

    }
}
```

4. sz. melléklet

G-code generátorral létrehozott példaprogram

A konzolon keresztül történő adatbekérés a páciens azonosítójával kezdődik:



Be- és kimeneti értékek a konzolban:

Kérem a páciens nevét és azonosítóját!

Paciens_2022

Adja meg a sebességet 1-10ig terjedő skálán! 1=F50 sebesség

4

Adja meg az Z tengely menti pozitív szög értékét!

30

Adja meg az Z tengely menti negatív szög értékét!

15

Adja meg az X tengely menti pozitív szög értékét!

50

Adja meg az X tengely menti negatív szög értékét!

35

Adja meg az Y tengely menti pozitív szög értékét!

40

Adja meg az Y tengely menti negatív szög értékét!

10

Fájl létrehozva: Paciens_2022.txt

Sikeres g-code generálás

Generált g-code fájl:

G28

G91

G1 Z-52.5 F200.0

G1 X+263.52942 F200.0

G1 Z+157.5 F200.0

G1 X-448.0 F200.0

G1 Z-157.5 F200.0

G1 X+184.4706 F200.0

G1 Y-73.333336 F200.0

G1 Y+366.6667 F200.0

Óbudai Egyetem
Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar
Gépészeti és Technológiai Intézet

SZINOPSZIS

Név, Neptun kód: Turner Máté, GE3O5R

Szakdolgozat címe: Rehabilitációs gép tervezése és megvalósítása

Szakdolgozatom célkitűzése stroke betegségen átesett emberek számára egy kéz rehabilitációs gép megtervezése és megvalósítása, amely a bénult és görcsbeállt kezükön végez rehabilitációs tornagyakorlatokat. Feladatom a csukló mozgatására összpontosul.

Munkámat a készülékem létrehozásához szükséges szakirodalom felkutatásával kezdtem. Ismertettem a stroke betegség ismérveit, rehabilitációs kezeléseit, valamint a kéz alapvető anatómiáját, mozgástartományait és betegségeit.

A saját munkám során a csukló mozgástartományait határoztam meg, hogy köré egy gépet tudjak tervezni. Meghatároztam a csukló mozgatásához szükséges léptetőmotorokat és a tengelyeikre helyezett maximális méretű tárcsák által a kifejtett minimális nyomtatékukat.

3D szkenneltettem a jobb kezem, hogy a gép tervezését ez a modell köré építsem fel. Meghatároztam a legfontosabb méreteit és megterveztem a kéz befogását. Meghatároztam a kezet mozgató elemeket és a mozgási pályáikat. Az eddigiek ismeretében befejeztem a tervezési munkálataimat és nekifogtam a gép gyártásához.

Az alkatrészeket 3D nyomtatással gyártottam le, PETG alapanyagból, mely megfelel az egészségügyi követelményeknek. Az alkatrészeket tervező program segítségével több részre daraboltam, hogy a 3D nyomtatóim nyomtatási területére beférjenek. A legyártott alkatrészekből a gépet összeszereltem és a vezérlés megvalósításával folytattam munkámat.

A rehabilitációs gép vezérlésének egy Raspberry Pi 4 B modellt választottam, egy BIGTREETECH Octopus Pro alaplap együttesével, TMC2209-es léptetőmotor vezérlőchipekkel. Konfiguráltam egy működő keretrendszert a Raspberry-hez, majd elvégeztem a szükséges teszteléseket és utókalibrálásokat.

Végezetül kifejlesztettem java nyelven egy rehabilitációs tornáztatási fájlokat generáló szoftvert, amely g-code-okat generál a beviteli maximális szögértékeknek megfelelően. Ezzel páciensekre szabott, egyedi rehabilitációs mozgatási parancsokat tárolva, melyek egy későbbi kezelés alkalmával is használhatók.

Óbuda University
Donát Bánki Faculty of Mechanical and Safety Engineering
Institute of Mechanical Engineering and Technology

SUMMARY

Name: Máté Turner, GE3O5R

Title of diploma work: Design of a rehabilitation machine

The goals of my thesis are the planning and building of a rehabilitation machine for those who had stroke. The machine shall provide support for the stuck and paralyzed wrist to perform the rehabilitational exercises. In this broad topic my focus is on the movement of wrist joint.

I started my thesis writing process by researching the necessary articles and documentations. I have looked up the side-effects and symptoms of stroke, the used methods of rehabilitation and basic anatomical knowledge of the hand, mainly its range of motion and medical conditions.

During my own work I have determined the general movement range of the wrist. I have used this information to plan the machine. To help myself with the design process I have 3D scanned my own arm so I can use it as a template. During the this process I have determined the most crucial dimensions and made a jig to hold the arm. I also the designed the parts necessary for the movement of the arm. By summarising my amassed knowledge and research in the topic I have started the building of machine.

I used 3D printers to fabricate the parts of the machine. I used PETG as material as it is suitable to the medical requirements. To make the parts fit my 3d printers I have used a CAD modelling software to cut them up in smaller parts. I continued my thesis work after printing all the parts by assembling the electrical components.

For the controlling of the machine, I choose a Raspberry Pi and a motherboard for 3D printers. I have configured the necessary operating system and software for the Raspberry Pi and made sure to test and review the firmware parameters.

Last but not least I have developed software in Java to generate the required movements for the rehabilitation. This software generates g-code files that contain the necessary movements according the maximum angles already determined. Using this method each patient can receive a personalised and unique treatment. By saving the g-code files the treatment is easily repeatable at a later date.