Szemcsehalmazon belüli elmozdulások mérésére alkalmazható gyorsulásmérő szenzor alapú mérőrendszer fejlesztése

Tudományos Diákköri Konferencia 2020

Szerzők:

Wágner Árpád BSc IV. évf. *(wgnrar@gmail.com)*

Kovács Gyula MSc II. évf. *(kgyula9307@gmail.com)*

Konzulensek:

Automatizálás és Alkalmazott Informatikai Tanszék:

Kovács László *(kovacs.laszlo@vik.bme.hu)*

Gép- és Terméktervezés Tanszék:

Dr. Tamás Kornél *(tamas.kornel@gt3.bme.hu)*

Hudoba Zoltán *(hudoba.zoltan@gt3.bme.hu)*

Budapest, 2020

Tartalom

1 Köszönetnyilvánítás

Dolgozatom részletezése előtt szeretnék köszönetet mondani mindenkinek, aki bármilyen módon segítette ezen dolgozat elkészültét. Szeretném ezt megköszönni konzulenseimnek, Kovács László, Dr. Tamás Kornél és Hudoba Zoltán tanár uraknak, szerzőtársamnak, Kovács Gyulának, továbbá Dr. Farkas Zsolt József tanár úrnak, akik tanácsaikkal és szakértelmükkel segítették a munkámat. Köszönöm továbbá Illés Lilla Szilvia munkáját, aki a 3D nyomtatáshoz használt modellt készítette, ezen túl köszönöm a Budapesti Műszaki és Gazdaságtudományi Egyetem Ipar 4.0 Technológiai Központjának, hogy a szükséges eszközöket a rendelkezésemre bocsátották. Köszönet illeti továbbá a NAIK Mezőgazdasági Gépesítési Intézetet, akiknél a méréseket végeztük.

2 Bevezetés

Szemcsés anyagok által alkotott szemcsehalmazokkal sok helyen találkozhatunk. Ilyen lehet például egy gyógyszergyárban a legyártott tabletták sokasága egy tárolóedényben, de akár a talaj is. Világos, hogy ilyen szemcsehalmazokon belül lehetnek elmozdulások, például amikor a tárolóedény alját megnyitva annak tartalma „kifolyik", vagy amikor egy mezőgazdasági gép a talajt megmozgatja. A szemcsék mozgásának ismerete sok esetben hasznos információval szolgálhat, azonban felmerül néhány nehézség ezzel kapcsolatban. Ilyen nehézség például, hogy egy szemcsehalmaz belsejébe nem látunk be, így konkrét képet legfeljebb nagyon drága eszközökkel kaphatunk. Ezen problémák kiküszöbölésére lehet alkalmas egy olyan eszköz, amely mind gyorsulást, mind elfordulást (orientációt) képes mérni. A technológia fejlődésének köszönhetően ilyen szenzorok ma már olcsón beszerezhetők és mindenki számára elérhetők. Figyelembe kell azonban venni, hogy ezek az eszközök korántsem tökéletesek, az általuk mért értékekben a mérési eredményeket nagymértékben eltorzító zaj is lehet, éppen ezért néhány kikötéssel kell élnünk a megvalósítás során, ezek a következők:

1. A mozgásnak mind időben, mind távolságban aránylag rövidnek kell lennie.
2. A lehető legpontosabb eredmény érdekében az eszköz összelmozdulása más módon mérhető kell, hogy legyen.

Az elkészült mérőeszközt talajvályúban próbáltuk ki a vele párhuzamosan készülő penetrométerrel együtt.

3 Szakirodalom

A gyorsulásmérésre épülő helymeghatározás természetesen nem egy teljesen új ötlet, más is kísérletezett már hasonló eszközökkel. Ilyen kísérletet végeztek például Yang és kollégái 2017-ben [1], akik egy saját maguk által készített kígyószerű robot mozgását tudták rögzíteni egy IMU segítségével megfelelő hibajavító algoritmusok alkalmazásával. Az ő értekezésükben azonban nem esik szó az orientáció megváltozásáról, ami a mi esetünkben egy kifejezetten fontos tényező.

Ezen túlmenően a munkavégzés közben történő talajelmozdulást is többen igyekeztek már mérni. Tamás 2018-ban [2] sikeresen tudta vizsgálni a talaj, mint halmaz függőleges elmozdulását lézeres profilográf segítségével. Milkevych és kollégái szintén 2018-ban [3] a föld alá adott mélységbe elhelyezett jelölőkockák segítségével vizsgálták a szerszám okozta

elmozdulásokat. Ezzel a méréssel azonban csupán az egyes kockák kezdeti- és végpontjait tudták meghatározni, a megtett útról nem kaptak információt.

4 Célkitűzések

Cél egy olyan mérőrendszer létrehozása, mely segítségével feltérképezhetjük a szemcsehalmazokon belüli elmozdulásokat anélkül, hogy a mérőeszközökre közvetlen rálátásunk lenne. Mindezt úgy, hogy szükség szerint a lehető legkisebb méretűre lehessen építeni a mérőelemeket. Szintén célként fogalmaztuk meg, hogy az eszköz a lehető legolcsóbb legyen, hogy egy-egy méréshez több (akár néhányszor tíz) mérőegységet is használhassunk.

További cél, hogy az elkészülő eszközök alkalmazásra kerülhessenek egy talaj-szerszám kapcsolatot vizsgáló kutatásban, ahol a talajelmozdulás modellezésére felvett diszkrételemes módszerre (DEM) épülő modellezés változóinak konfigurálásának alapjául szolgálhassanak (a kapcsolt mérés eredményeivel együtt).

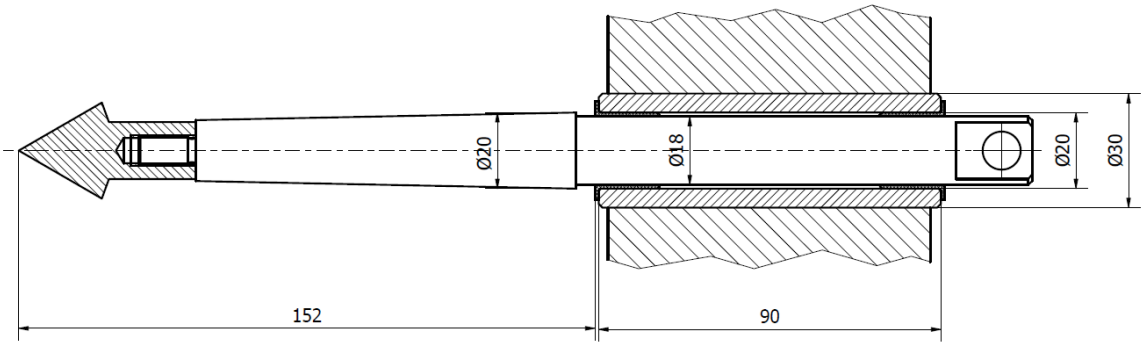
5 A mérőeszköz

5.1 A horizontális penetrométer

Az előbb elírtakból kiderül, hogy az eszköz egy kapcsolt mérésben került alkalmazásra. A másik eszköz nem más, mint egy penetrométer (5.1), ami képes a talaj horizontális penetrációs ellenállásának mérésére. Ezeket a méréseket különböző sebességeknél, a penetrométeren különböző kúpokkal is megtesszük. A várt eredmény itt az, hogy ezen paraméterek megváltozása esetén is a mért jellegeknek hasonlónak kell lenniük.

5.1 Laboratóriumi talajvályú és mérőkocsi

A mérőkocsira szerelt horizontális penetrométer alapját egy mérőtüske jelenti, ami 5.2 ábrán látható. Ennek feladata a rászerelt kúpokra ható penetrációs erő továbbítása a mérőegység felé. A mérőcsúcson mérhető erő és ennek felületének hányadosa adja a penetrációs ellenállás értékét. A kialakításnak köszönhetően változtatható a mérőszár hossza, így különböző vizsgálati beállítások lehetségesek.

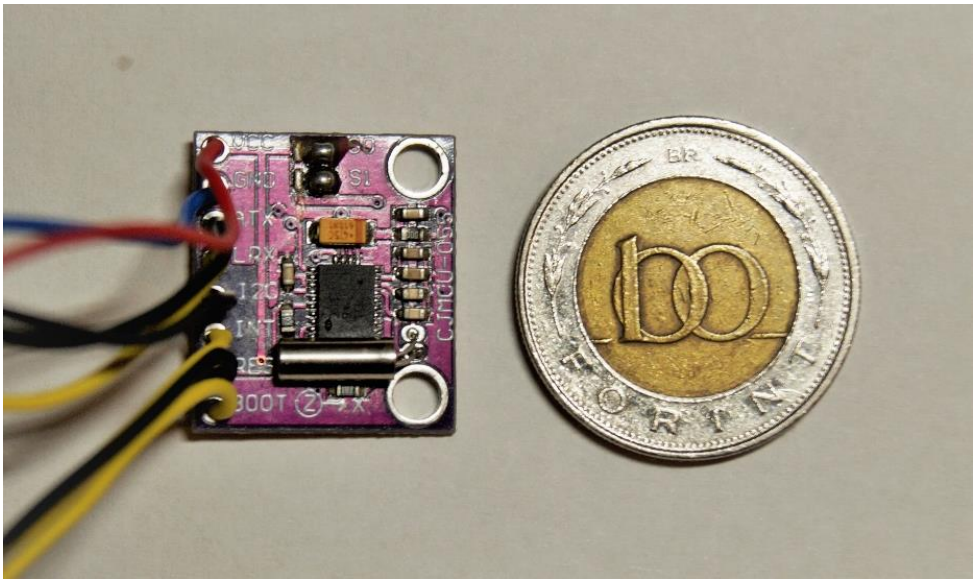


5.2 Megvalósított penetrációs egység

Mind a horizontális penetrométer, mind az elmozdulásmérő mérési eredményei jó alapot szolgáltatnak különböző talajmodellek kalibrációihoz.

5.2 Az elmozdulásmérő és fejlesztése

Az alapötlet az volt, hogy egy gyorsulásmérőt használjunk a hely kiszámításához. Azonban „szimpla” gyorsulásmérőt szinte nem is lehet kapni, majdnem mindegyikben van legalább még egy giroszkóp vagy magnetométer. Ez természetesen nem véletlen, hiszen ezek alkalmazásával nem csak az egyes szenzorértékeket (mint például gyorsulás vagy mágneses térerősség) lehet mérni, hanem szenzorfúzión keresztül megkaphatjuk az eszköz orientációját is jóval pontosabban, mint ahogy az egyes részegységek adataiból külön-külön ki lehetne azt következtetni. Az ilyen egységeket az angol terminológia Inertial Measurement Unit-nak (rövidítve IMU) nevezi. Ezekből számtalan típus található meg a különböző internetes áruházakban, ezek közül a Bosch BNO055 nevű szenzorára épülő panel (5.3) kiválasztása tudatos döntés volt. Ez a szenzor ugyanis nem csak önmagában egy IMU, hanem tartalmaz egy beépített 32 bites mikrokontrollert, melyen a gyárilag telepített szoftver miatt képesek vagyunk közvetlenül a fuzionált adatokat is lekérni az eszköztől (ezért is hívja a gyártója smart sensor-nak). Ilyen fuzionált adatok például az Euler-szögekkel vagy kvaterniókkal reprezentált orientáció. Ezen adatokon túl képesek vagyunk lekérni az egyes részegységek adatait is, így tehát egy szenzor segítségével egyszerre állnak rendelkezésünkre gyorsulás és orientáció adatok is. Ezen kívül nem hagyható figyelmen kívül az sem, hogy az eszközhöz elérhető egy hivatalos könyvtár is, mely nagymértékben megkönnyíti az eszközzel való kommunikációt.

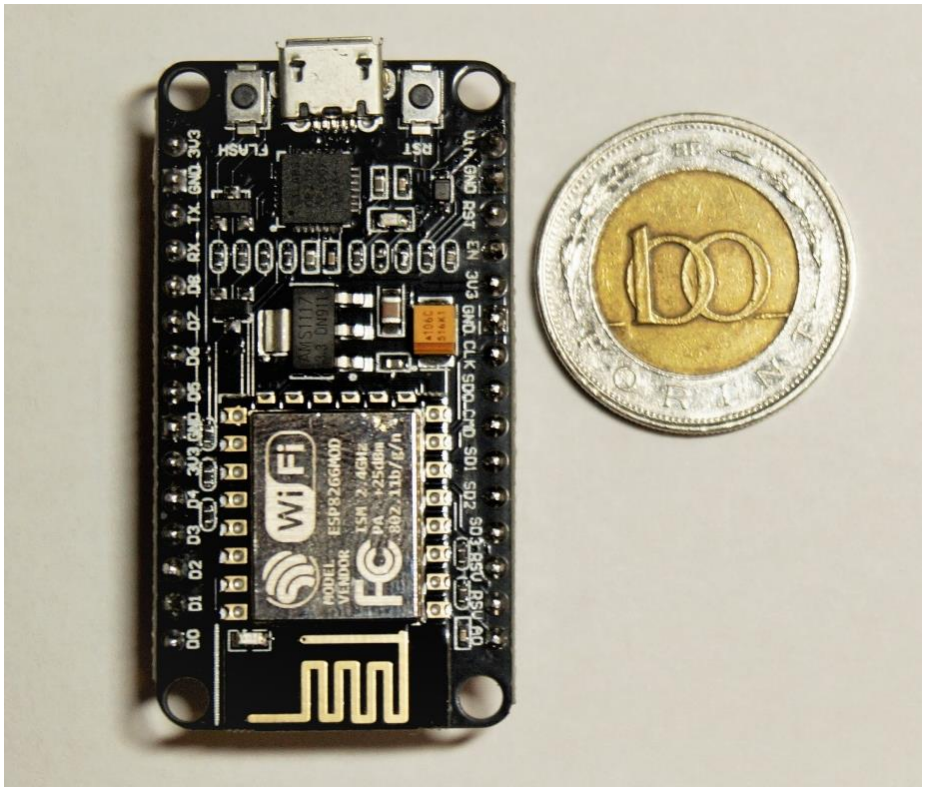


5.3 A Bosch BNO055 breakout panelje

A szenzor mellett természetesen szükségünk van egy vezérlőre is. Ezek közül mi egy ESP8266 alapú lapkát (5.4) választottunk. Ezen eszközöknek nagy előnyük, hogy használhatóak az Arduino keretrendszerhez, melyhez számos könyvtár elérhető (így például az általunk használt

szenzorhoz adott könyvtár is létezik erre a platformra). Ez persze még nem indokolja, hogy valamely Arduino típus helyett ezt használjuk. Ezek az eszközök azonban nemcsak gyorsabbak és nagyobb memóriával rendelkeznek, mint az Arduino lapkák nagyrésze, de beépített WiFi-vel is rendelkeznek, mely nagyban megkönnyíti a mérések vezérlését. Mindemellett olcsóbban is beszerezhetőek.

Mint azt említettem, ezek az ESP-k több formában is léteznek. Mi az első tesztekhez egy nagyobb méretű lapkát választottunk, amelynek a használata sokkal kényelmesebb, mint egy kisméretű „rokonáé". Azonban mivel ilyen széleskörű ez a termékcsalád, a későbbiekben lehetőség nyílhat más lapkák alkalmazására, amelyek segítségével a mérőeszköz mérete csökkenthető.

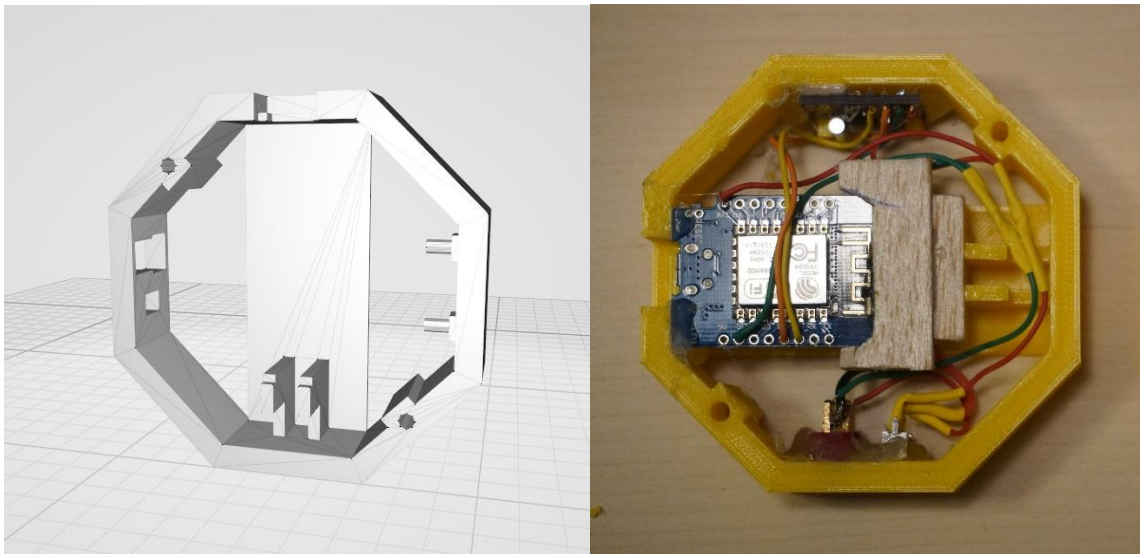


5.4 A méréshez is használt ESP8266 alapú lapka

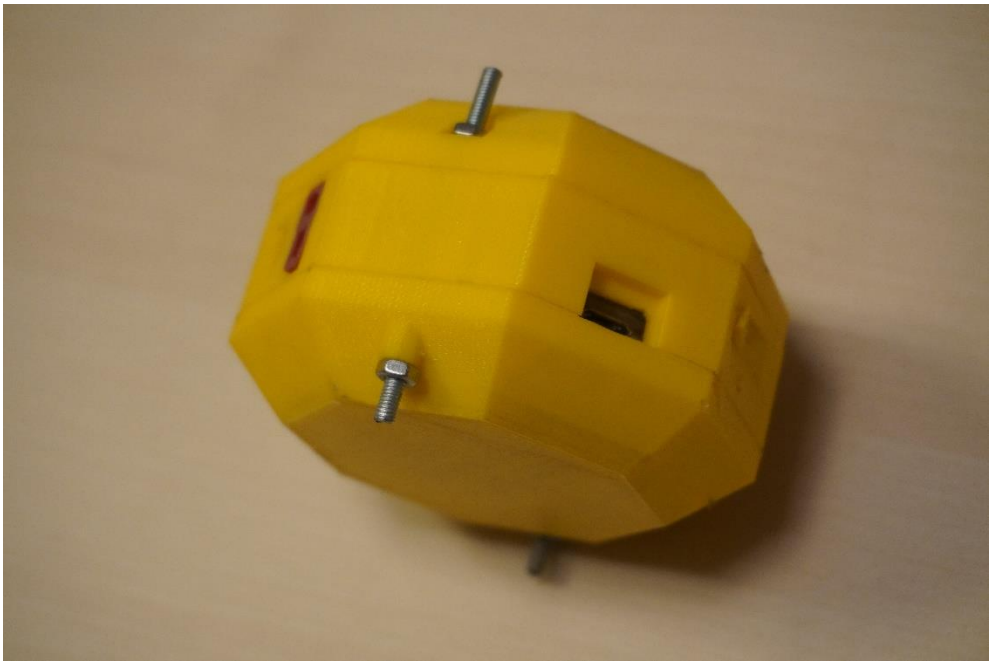
Ez a két eszköz (a szenzor és az ESP8266-alapú MCU (Microcontroller Unit)) nevezhető a mérőegység fő részének. Ezek egymással végzett kommunikációja I2C buszon valósul meg, az MCU és a számítógépen futó mérésvezérlő program kommunikációja pedig WiFi-n vagy USB-n keresztül történhet.

A fő részegységek egy 3D nyomtatással készülő házba (5.5, 5.6) kerültek bele, amiben mellettük egy akkumulátor is helyet kapott. A ház három részből állt össze, melyek közül a középső úgy került kialakításra, hogy abba a fent említett alkatrészek behelyezhetők legyenek.

Ezen kívül helyet kapott rajta egy nyílás, ahova az ESP USB-portja illeszkedik, illetve egy, ahova egy kapcsoló építhető be. A másik két elem adja az eszköz alját és tetejét, az egyes elemeket csavarral lehet egymáshoz rögzíteni (valós körülmények között ezen túl szigetelőszalaggal is körbetekertük a házat, hogy lehetőleg ne jusson be por az elektronikai alkatrészekhez).



5.5 A mérőegység házának középső eleme: terv (b) és a kinyomtatott ház (j) a mérőegység többi alkatrészével



5.6 Az elkészült mérőegység összeszerelve

5.3 Mérésvezérlés, adatfeldolgozás

Amint azt már említettem, a mérőeszköz vezérlése WiFi-n keresztül számítógépről történik. Ehhez természetesen szükséges, hogy a két eszköz egy alhálózaton legyen, melyet biztosíthatunk úgy, hogy ugyanarra a vezetéknélküli hálózatra csatlakozik a kettő, vagy akár úgy is, hogy a számítógépen saját hálózatot indítunk, a mérőeszköz pedig arra csatlakozik.

A mérési szoftverrendszer három részre bontható:

1. **Adatgyűjtő**: Az ESP-n futó program. Kommunikál a mérésvezérlővel és az onnan érkező parancsokat végrehajtja.
2. **Mérésvezérlő**: Feladata kommunikálni a mérőegységekkel, azoknak a felhasználótól érkező parancsokat el kell juttatnia. Az onnan érkező adatokat fájlba tudja menteni, ami az adatfeldolgozás alapjául szolgál.
3. **Adatfeldolgozó**: A begyűjtött adatokat lehet vele javítani, feladata korrigálni a mérőeszköz tökéletlenségéből adódó hibákat (amennyire lehet), ezen kívül célja az adatok értelmezése, tehát a nyers gyorsulás- és orientációadatokból az eszköz által megtett út felvázolása.

Az adatgyűjtő program Arduino környezetben készült, a másik két alkalmazást Python nyelven készítettem, ugyanis itt rengeteg adat- és jelfeldolgozást segítő programkönyvtár elérhető, ami nagyban leegyszerűsíti a fejlesztést.

5.4 Adatgyűjtő program

Amint azt említettem, adatgyűjtő program alatt az ESP-n futó szoftvert értem. Egy mérés során a feladata, hogy amikor épp nem „dolgozik”, figyelje a mérésvezérlőtől bejövő parancsokat és azokat végrehajtsa.

A megfelelő parancs beérkezésekor az adatgyűjtés elkezdődik, az eszköz folyamatosan olvassa ki a hozzá csatolt szenzor adatait és azokat egy erre a célra a memóriában előre lefoglalt területre menti. Ebből az is következik, hogy a vett adatok száma fix, a felesleges részeket a feldolgozás során a felhasználónak kell eltávolítania.
Ez a programrész úgy lett tervezve, hogy a lehető legnagyobb frekvenciával legyen képes az eszköz a mintavételezésre, emiatt eközben nem fogad parancsokat és csak nyers értékeket tárol, azokat a feldolgozás során kell mértékegységekre váltani.

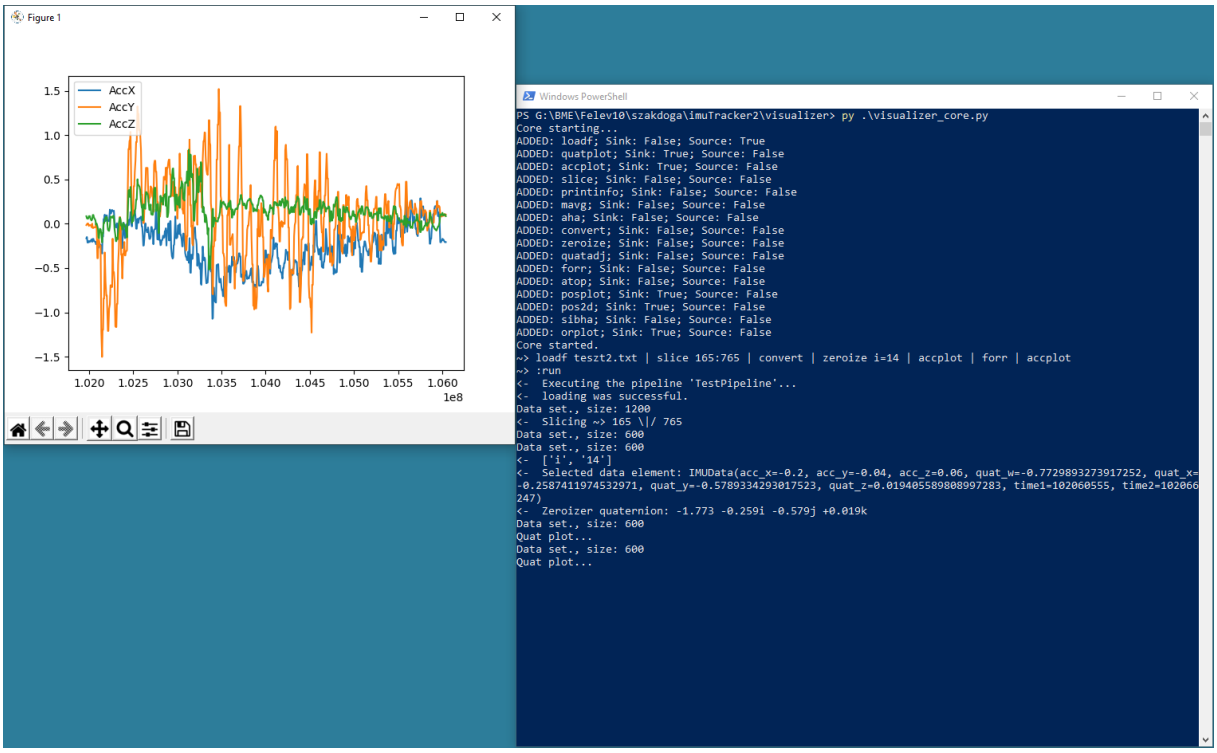
5.5 Mérésvezérlő program

Amint említettem, ennek a különálló programként megvalósított szoftverrendszer-elemnek a feladata, hogy a mérőeszközt vezérelje és vele kommunikáljon. Feladata továbbá az is, hogy az eszköz által visszaküldött válaszokat képes legyen megjeleníteni a felhasználó számára és adott esetben azokat fájlba elmenteni.

Az alább ismertetett adatfeldolgozóhoz hasonlóan ez az alkalmazás is parancssoros felülettel rendelkezik, de úgy lett tervezve, hogy grafikus felületet is lehessen hozzá készíteni.

5.6 Adatfeldolgozó program

A program a Python interpreter segítségével futtatható és parancssoros felülettel (CLI – Command Line Interface) rendelkezik (5.7), de a program úgy lett tervezve, hogy GUI-t (Graphical User Interface – Grafikus felhasználói Interfész) is lehessen készíteni hozzá. Tervezésnél szempont volt az, hogy az adatfeldolgozás lépései és azok sorrendje is rugalmasan változtatható legyen. Ennek érdekében a program az adatfeldolgozást csővezeték-szerűen (pipeline) végzi. Ez azt jelenti, hogy több parancs áll a felhasználó rendelkezésére, melyek mindig a feldolgozás egy-egy lépéséért felelősek. Így külön parancs például egy fájl beolvasása és a beolvasott nyers adatok konvertálása mértékegységekkel ellátott adatokká. Ilyen módon a felhasználó nagy szabadságot kap abban, hogy meghatározza, milyen algoritmusokat használ és azokat milyen sorrendben, ráadásul így egy problémára több algoritmust is kínálhat a program és a felhasználóra lesz bízva, hogy melyiket használja. Ezen túl az algoritmusok repertoárja így bármikor egyszerűen bővíthető marad, ha szükség van rá.



5.7 Az adatfeldolgozó program működés közben (PowerShellből futtatva)

A csővezeték úgy lett tervezve, hogy abból egy programon belül is több legyen belőle létrehozható. Így lehetővé válik az, hogy ha egy mérés során több eszköztől érkeznek az adatok, akkor azok egymástól függetlenül feldolgozhatóak legyenek. Ezen túl megjegyzendő, hogy az adathalmazok pipeline példányokhoz kötődnek, nem parancshoz vagy statikus módon osztályhoz.

Egy pipeline elemei megadhatók egy sorban az egyes lépéseket „|” karakterrel elválasztva, vagy soronként beírva is. Amint a csővezeték terve összeállt, a `:run` parancsot kiadva elindul annak végrehajtása.

Tehát a

```
~> loadf adatok.txt | convert | aha | accplot
~> :run
```

Sorok ekvivalensek a

```
~> loadf adatok.txt
~> convert | aha
~> accplot
~> :run
```

sorokkal.

A parancsok három részre bonthatók: lehetnek

1. **Bemenetek** (source): Olyan parancsok, amelyek adatot visznek be a feldolgozási csővezetékbe. Ilyen lehet például egy fájlból való beolvasás.
2. **Köztes elemek**: Bemenetük és kimenetük is van, ezek azok a parancsok, amelyek magát a feldolgozást végzik. Ilyen lehet például egy eltolás alkalmazása minden adatelemre.
3. **Nyelők** (sink): Olyan parancsok, amelyek a kapott adatot nem módosítják és nem is adják tovább a csővezetéken. Ezekre példa lehet egy megjelenítő vagy fájlba mentő parancs.

   Megjegyzendő, hogy nyelők is állhatnak köztes elemként, amit úgy értelmezünk, hogy az utánuk következő parancs ugyanazt az adatot kapja meg a bemenetén, amit az előtte álló nyelő.

A részletek előtt még szeretnék szót ejteni a feldolgozandó adatokról. A mérőegységtől kilenc adatrészt kapunk, melyek a következők:

1. **Gyorsulás** az X, Y, Z tengely mentén
2. **Orientáció** kvaternió formában W, X, Y, Z tagokkal
3. Két **idő**: az egyik a gyorsulásadatok, a másik a kvaternióadatok beolvasásának időpontja időbélyeg jelleggel

Vegyük észre, hogy ezen adatok sorozata külön-külön egy-egy bejövő jelként értelmezhető.

Az alábbiakban ismertetem a beolvasott jel feldolgozásához szorosan kapcsolódó parancsokat (az előbbiekből kiderülhet, hogy ezek mind a „köztes elem” kategóriába tartoznak). Minden parancs leírásának első sora egy példát mutat arra, hogy a programban milyen néven hivatkozhatunk rájuk.

5.6.1 Konvertálás

`convert`

Egy egyszerű parancs, mely a bemenetén várja a nyers gyorsulás- és orientáció adatokat és visszaadja ezeknek a mértékegységesített értékét. Így a nyers gyorsulásadatokat m/s^2-ban, míg a kvaternióadatokat egy egységkvaternió elemeiként adja vissza.

5.6.2 Slice

`slice 10:200`

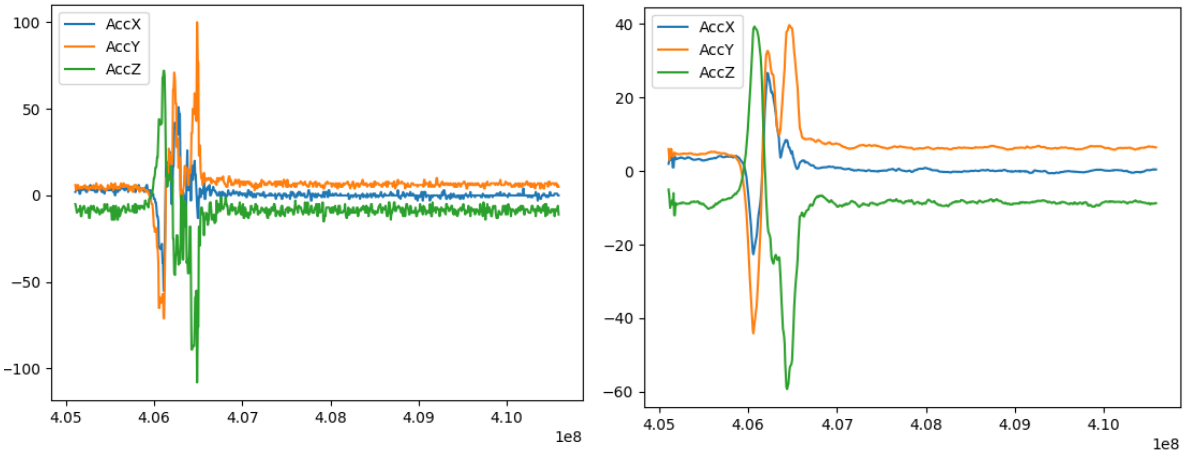
A Python nyelv azonos nevű funkciójával analóg módon a teljes adathalmaz két megadott pont közti részét adja vissza.

Az egyszerűség kedvéért úgy lett a funkció implementálva, hogy ne legyen kötelező megadni mindkét paramétert. Így tehát a `slice :750` és a `slice 20:` is értelmes paracs, melyek csak az adathalmaz végéről, illetve elejéről vágnak le elemeket.

5.6.3 Mozgóátlag

`mavg 2:xyz`

Mozgóátlagot számol a gyorsulásadatok kért tengelyein. A tengelyeken túl az ablakméret is állítható. Ennek segítségével csökkenthető a gyorsulásmérő zaja, azonban figyelnünk kell arra, hogy ezzel együtt a jel amplitúdója is megváltozik.



5.8 Mozgóátlag hatása az adatokra (figyeljük meg az Y tengely skálázását!) Függőleges tengely – nyers jelérték, vízszintes tengely - időlépés

5.6.4 Orientációnullázás

`zeroize i=0`

Egy megadott kezdeti pontot fog „nulla” orientációnak venni, ami azt jelenti, hogy a szenzor azon időpontbeli orientációját állítja be világ-koordinátarendszernek, azaz ekkor világ- és eszközkoordináták tengelyei egybeesnek. Legyen q_v az a kvaternió, amit kiválasztottunk (tehát amit a nulla elforgatásnak szeretnénk tekinteni).

A feladat elvégzéséhez induljunk ki a

$$q_1 = [1 \quad 0 \quad 0 \quad 0]$$

kvaternióból, hiszen ez az, amelyiknél az elforgatás éppen nulla. A feladat ezután az, hogy készítsen a program egy q_z kvaterniót, amelyet a többi q_k számnégyesből kivonva éppen akkor lesz q_1, ha az egyenlő q_v-vel. Innen már ki lehet következtetni, hogy

$$q_z = [(-1 + w_k) \quad x_k \quad y_k \quad z_k] \,|w_k, x_k, y_k, z_k \in q_k$$

5.6.5 Orientáció igazítás

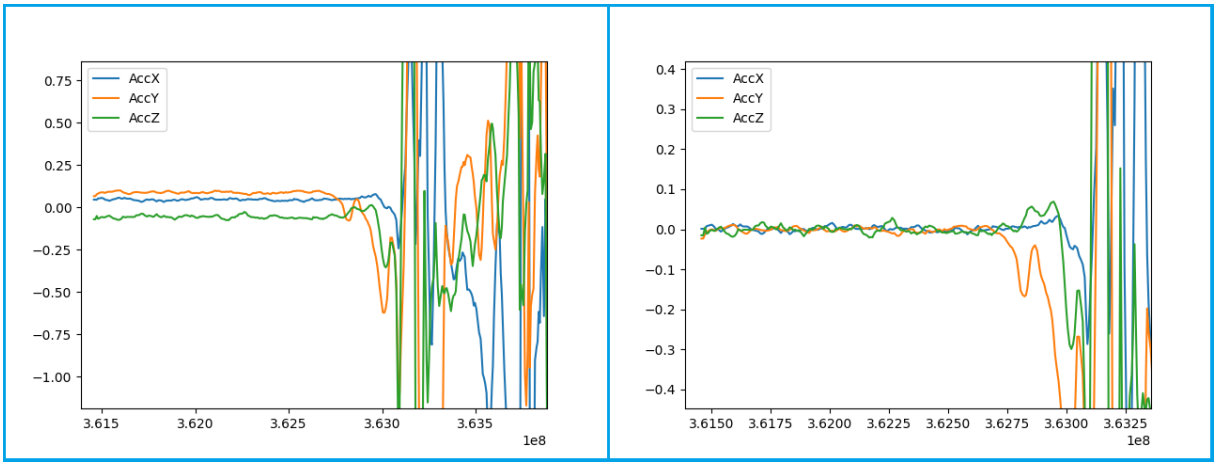
`quatadj`

A mintavételezés mind a gyorsulásadatok, mind a kvaterniók lekérésénél időbe telik. Ez a parancs az adattag saját és az előző adattag kvaterniója alapján becsli, hogy a gyorsulás mintavételezésének pillanatában mi volt az orientáció.

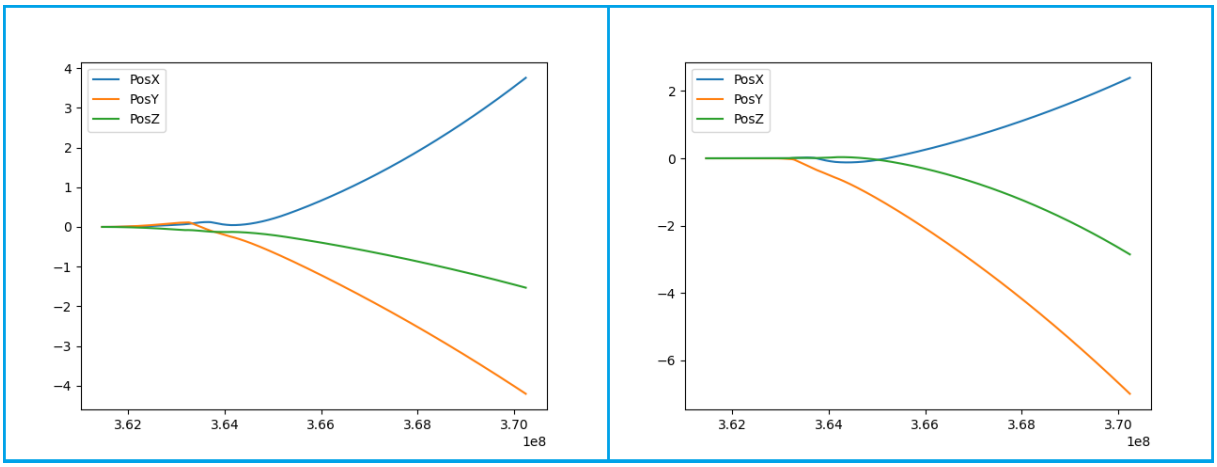
5.6.6 Automatikus eltoláskorrekció

`aha`

A tapasztalat azt mutatja, hogy a mérések elején a mérőeszköz egy ideig nyugalomban marad, azonban ebből az időszakból is érkeznek be adatok. Ezek azonban koránt sem feleslegesek, hiszen segítségükkel megbecsülhetjük, hogy a szenzornak mennyi a nyugalmi helyzeti hibája az egyes tengelyek mentén. Egy mozgóátlag után valahány elem átlagolásával kaphatunk egy aránylag pontos képet erről az értékről. Ezzel az egész jelet eltolva pontosíthatjuk a mérést. Ez az algoritmus is szignifikáns javulást tud okozni a mért értékekben, amint az alábbi ábra mutatja.



5.9 A mért gyorsulásértékek az egyes tengelyeken eltoláskorrekció előtt (b) illetve után (j) Függőleges tengely – nyers jelérték, vízszintes tengely - időlépés



5.10 A gyorsulásadatokból számolt pozíció az egyes tengelyeken eltoláskorrekció előtt (b) illetve után (j). Függőleges tengely –számolt dimenziótlan pozíció, vízszintes tengely - időlépés

5.6.7 Várt helyzetérték alapú eltoláskorrekció

`sibha 0:0.5:1.3`

A pontos eredmények érdekében fontos, hogy az elmozdulást egy másik eszközzel meg tudjuk mérni. Az ilyen referenciához azután hozzáigazíthatjuk a mért értékeket.

Az ötlet az, hogy ha tudjuk azt, hogy a kiindulási helyzethez képest a mérőeszköz a mozgása végéig mennyit mozdult el az egyes (világkoordinátarendszer-beli) tengelyek mentén, akkor tudjuk, hogy helyes eredményt akkor kapunk, ha a kimeneti helyzet épp ezzel megegyező.

Mivel itt a helyzetet ismerjük és a gyorsulásértékeket szeretnénk korrigálni (és az utóbbiból a kétszer integrálás visz át az előbbibe), így a programban ezt az eljárást „Second Integral Based Heigth Adjustion” - nek neveztem (vagy rövidítve SIBHA). Az ötlet hasonlít egy P vagy PI vezérlőre:

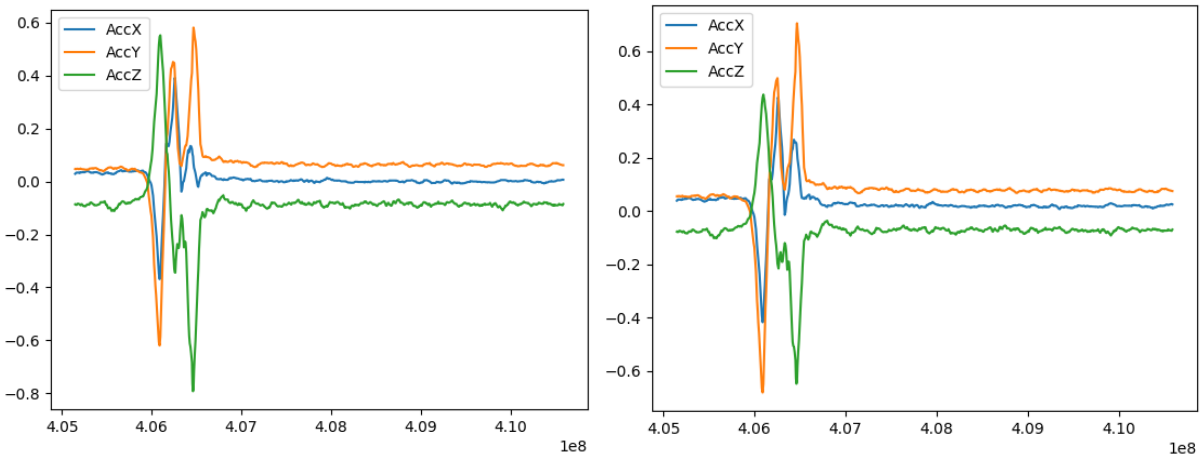
1. Kiszámoljuk a második integrált.
2. Vesszük a második integrál utolsó értékét és megnézzük, mennyivel tér el a várt értéktől. Ez meghatároz egy hibát.
3. Ha a hiba abszolút értéke egy meghatározott tűréshatáron belül van, akkor visszatérünk.
4. A hibát egy konstans értékkel elosztjuk (például 100-zal, de az optimális érték függ a módosítandó adathalmaztól[1]), és azt hozzáadjuk minden gyorsulásértékhez.

[1] Ha ezt az értéket túl nagynak választjuk, az algoritmus futásideje nagymértékben megnő, ha viszont túl kicsinek, akkor előfordulhat, hogy a számolt adat egyre távolodni fog a kívánt értéktől ahelyett, hogy közeledne.

5.6.8 Koordinátarendszer-elforgatás

`forr`

Mivel minden gyorsulásértékhez egy orientáció is kapcsolódik, így megtehetjük azt, hogy a gyorsulásvektorokat elforgassuk a hozzájuk tartozó kvaternióval. Ilyen módon átvihető a mért érték az eszköz koordinátarendszeréből a világ-koordinátarendszerbe.



5.11 Az eredeti (b) illetve az orientációkorrigált adatok (j) Függőleges tengely – nyers jelérték, vízszintes tengely - időlépés

5.6.9 Pozíció számítása gyorsulásadatokból

`atop`

A megtett utat a következőképp kaphatjuk meg:

$$s(\tau) = \iint_{t=0}^{\tau} a(t)\,dt$$

Természetesen az általunk vett jel nem folyamatos, hanem diszkrét, így az integrálás elvégzésére például a trapézszabály alkalmazható, mely a feldolgozásra használt egyik programkönyvtár (numpy) része.

Természetesen alapesetben figyelembe kellene venni a kezdeti sebességet és a kezdeti pozíciót is, azonban nekünk lehetőségünk van arra, hogy a méréseinket úgy végezzük, hogy ezeket az értékeket zérusnak lehessen tekinteni, így ezzel az egyszerűsítéssel élek a mérések során.

6 A szerkezet működésének ellenőrzése

A rendszer működését ellenőrizendő elvégeztem egy egyszerűbb, de a funkciók javarészét lefedő mérést: az asztalomon a mérőegységet megmozgattam nagyjából félkör alakban, aminél tudható, hogy a kezdeti- és végpontok azonosak. A mérés közben figyeltem arra, hogy az eszköz orientációja a mozgás közben változzon. Az eszköz által gyűjtött adatokat elmentettem egy szöveges fájlba.

A kapott adatokat ezután elkezdtem a feldolgozóprogram segítségével feldolgozni. Lépésről lépésre haladva kialakult a helyes pipeline, ami szavakkal leírva a következő:

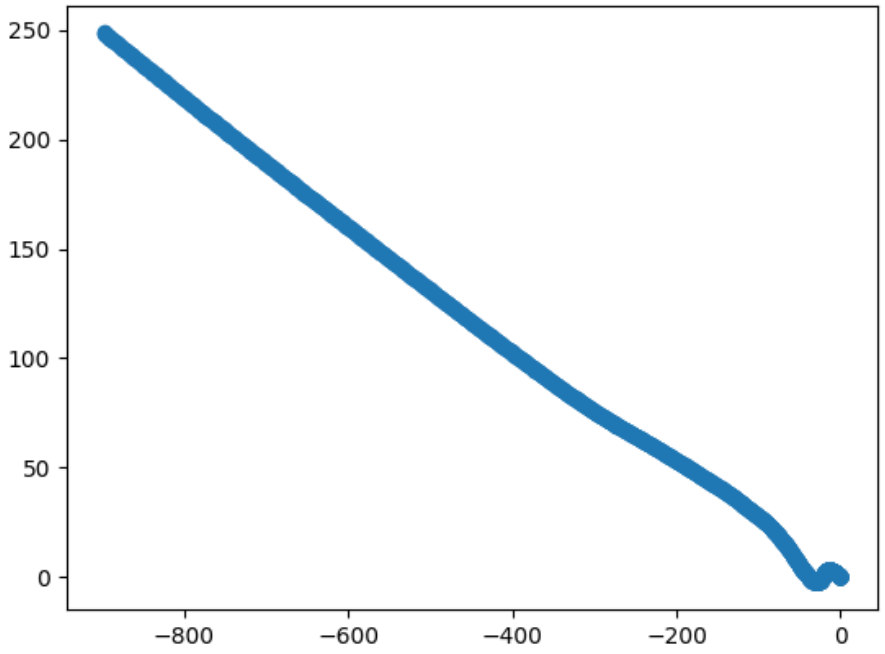
1. Fájl beolvasása.
2. Felesleges adatelemek levágása (a szükséges adatelemek nagyjából a 165-ös és 765-ös indexű adattagok között vannak).
3. Várt helyzetérték alapú eltoláskorrekció: minden tengelyen nagyjából ugyanoda érkeztünk vissza, így minden esetben a végső pozíciónak 0-nak kell lennie.
4. Nyers adatok konvertálása.
5. Orientációnullázás.
6. (Gyorsulásadatok kirajzolása.)
7. Koordinátarendszer-elforgatás.
8. Mozgóátlag 10 hosszú ablakkal minden tengelyre.
9. (Gyorsulásadatok kirajzolása.)
10. Pozíciók számítása a gyorsulásadatokból (kétszeres integrálás).
11. Az X és Y tengelyen a pozíciók kirajzolása.

Ez a folyamat a programban a következő sorral érhető el:

```
loadf teszt2.txt | slice 165:765 | sibha 0:0:0 | convert | zeroize
i=14 | accplot | forr | mavg 10:xyz | accplot | atop | pos2d xy
```

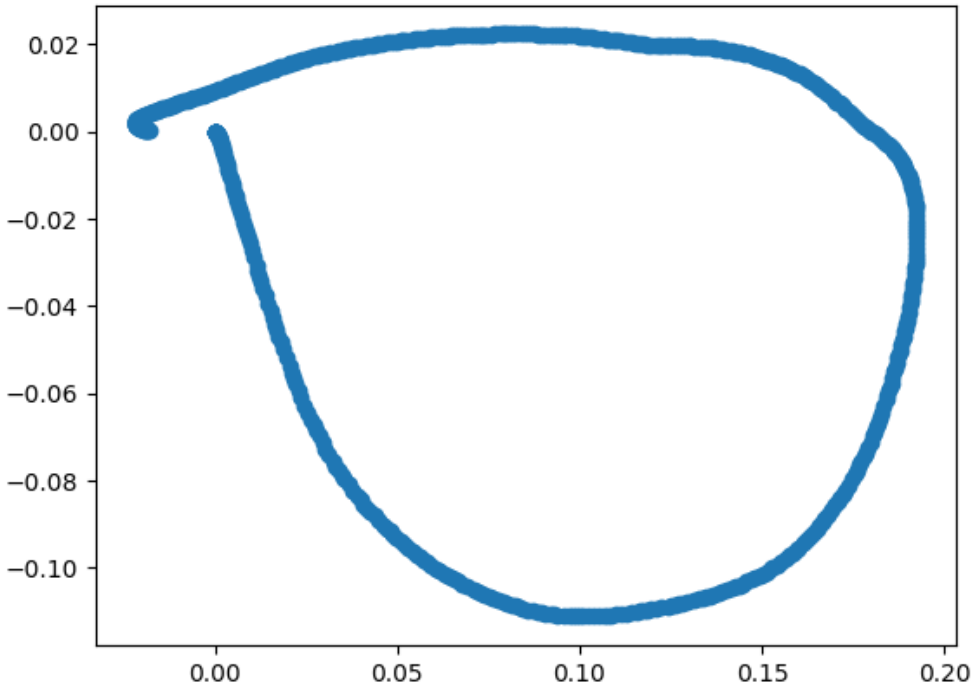
Ezen lépésekkel az alábbi eredményt érhetjük el:

A nyers adatokat közvetlenül pozícióvá alakítva a következő eredményt kapjuk:



6.1A nyers adatok közvetlenül elmoztulássá konvertálva az X-Y tengelyen

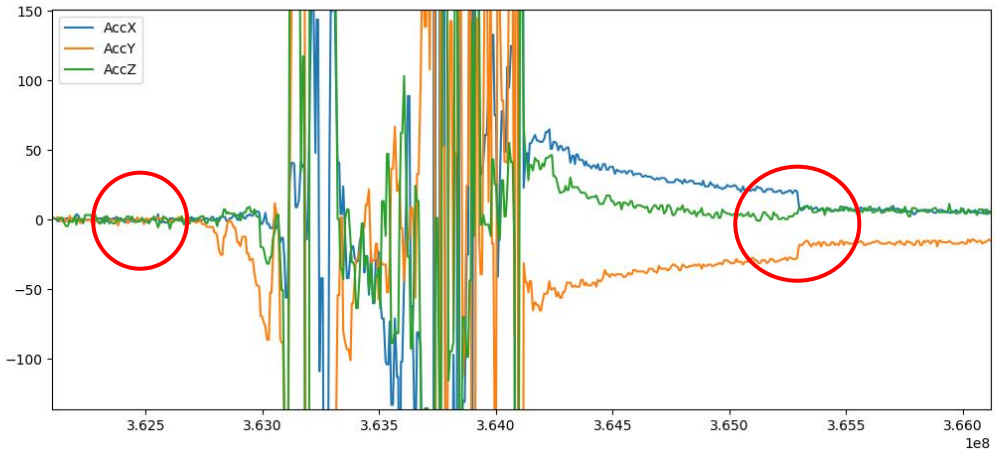
A fent említett korrekciókkal együtt azonban ezt:



6.2Az adatokból korrekció után számolt elmozdulás az X-Y tengelyen méterben

Az eredmény meglátásom szerint kifejezetten jó. Természetesen nem lesz 100%-os, de ezt nem is várhatjuk el. A látható forma elég jól visszaadja azt a formát, amit én az eszközzel „megrajzoltam”.

A tesztmérés után tűnt fel nekem a szenzor egy eddig ismeretlen hibája. A szenzor által nyugalmi pozícióban mutatott értékek ugyanis mások a mérés előtt, illetve után. Ez a hiba a tesztmérések alatt is benne volt a rendszerben, azonban láthatóan nagy mértékben nem zavarta meg a kimeneti értékeket.



6.3 A nyugalombeli hiba mozgatás előtt, illetve után (a mozgás előtti értékek korrigálva lettek) Függőleges tengely – nyers jelérték, vízszintes tengely - időlépés

7 Mérés

A mérőeszközzel végeztünk méréseket Gödöllőn a NAIK Mezőgazdasági Gépesítési Intézet talajvályújában (7.1). Összesen hat mérést végeztünk, minden esetben egy darab mérőeszközzel.

Az minden mérésről elmondható, hogy az eszközt a talajban nagyjából 20 cm mélyre elástuk (7.2), majd betemettük. Az eszköz vezérlését laptopról az erre megírt szoftver segítségével végeztem. Minden mérésnél fixen 1200 adatot rögzítettünk, melyekből általában a teljes adathalmaz nagyjából harmada volt az a rész, amely a tényleges mozgásadatokat tartalmazta. A hat mérésből négy esetben a mérőeszközt igyekeztünk úgy elhelyezni, hogy a penetrométerrel való találkozáskor a két eszköznek a közepe találkozzon. Két esetben a mérőeszközt a középvonaltól 5, illetve 10 centiméterre helyeztük el, hiszen ez is érdekes adatokkal szolgálhat.

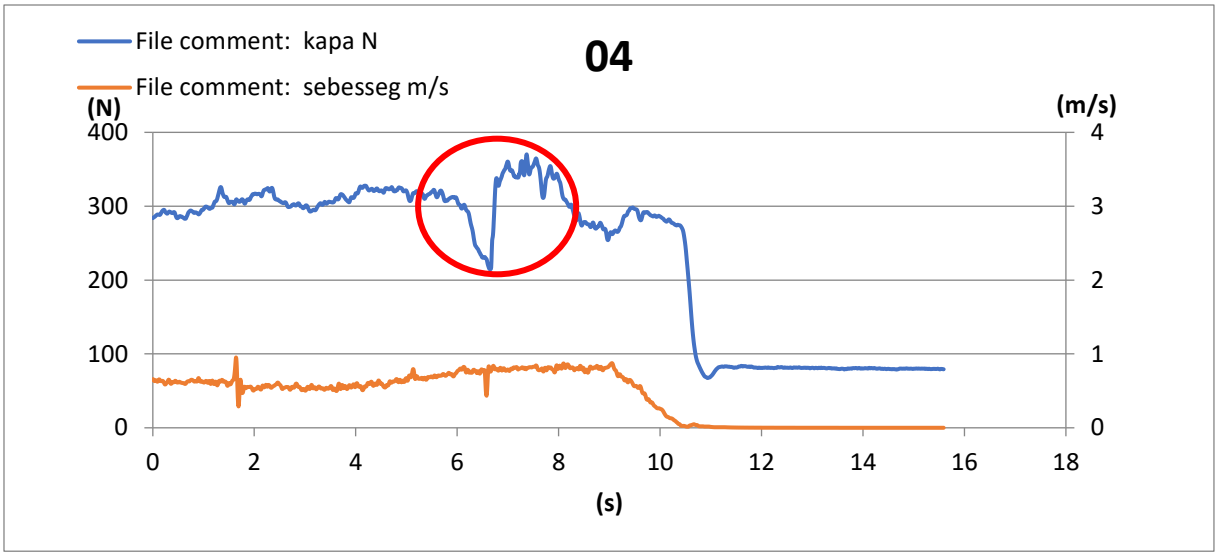
A mérési körülmények a következőképp alakultak:

- Hőmérséklet: 24,5 [°C]
- Páratartalom: 40 [V/V%]
- Talajnedvesség: 10 [V/V%]
- Mintavételezési frekvencia: 100 [Hz]
- Talaj típusa: Homok (93,28%-ban)
- Mérési szakasz hossza: 30 [m]

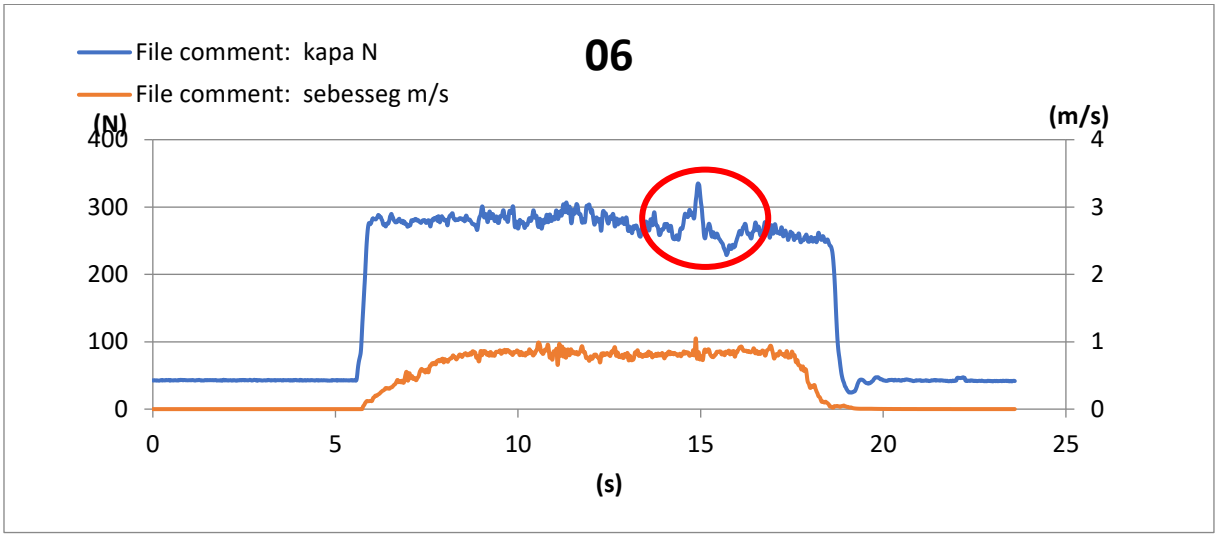
7.1 A méréshez használt eszköz (piros kör jelöli közelítőleg azt a helyet, ahová a mérőeszközt elhelyeztük)

7.2 A mérőegység a neki kiásott gödörben (piros körrel jelölve a 7.1-es képen)

Mérések után a penetrométerről kiolvasott adatokat figyelve feltűnt, hogy a két eszköz találkozásakor az addig egyenletes kiolvasott erő visszaesik. Úgy gondoltuk, hogy ez azért van, mert a beásáskor nem tömörítettük a talajt az eszköz helyén, így az utolsó mérésnél ezt megtettük. Az erőcsökkenés problémája valóban elhárult ezzel, azonban a tömörítés valószínűleg túl erős lett, hiszen ezen esetben a szükséges erő a találkozás környékén megnőtt.



7.3 A kapán mért erő és a gép sebessége a 4. mérésben

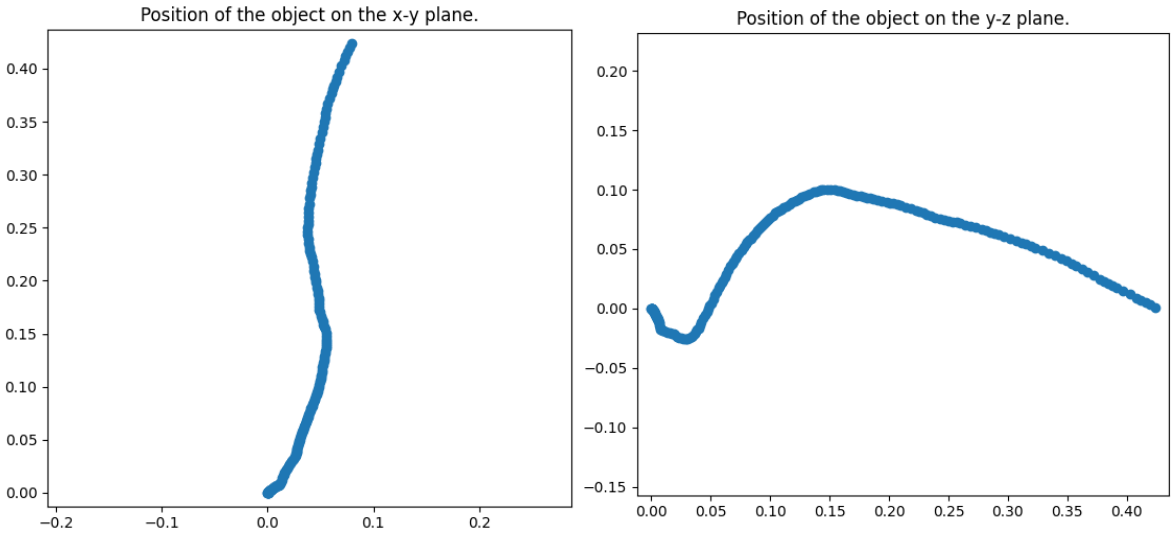


7.4 A kapán mért erő és a gép sebessége a 6. mérésben

A méréseknél természetesen nem csak a penetrométeres adatokat gyűjtöttük, hanem az elmozdulást is vizsgáltuk. Minden mérésnél feljegyeztük a mérőeszköz kezdeti pozícióját és a mérés után lemértük, hogy az egyes tengelyeken mennyi volt az elmozdulás.

7.5 A mérőeszköz elhelyezése a mérések előtt

A 4. mérésnél a mérőeszközt a penetrométer útjának közepére helyeztük el. A mérés végén 9 centimétert mozdult el jobbra, 42,5 centimétert előre, a Z tengely mentén pedig nagyjából ugyanazon a szinten maradt. Ezen adatokhoz való hozzáigazítás után az X-Y síkon való elmozdulásra az alábbi eredményt kaptuk:



7.6 A 4. mérésben mért elmozdulás az X-Y és Y-Z síkon (ahol (0:0) a kiindulási pozíciót jelenti)

A kapott kép nagymértékben hasonlít arra, amit vártunk, így az eredményt bíztatónak nevezhetjük.

Ezen eredmény eléréséhez a programban a következő pipeline-t hoztam létre:

```
loadf meres4.txt | aha | slice 200:390 | convert | zeroize i=10 |
quatplot | forr | sibha 0.08:0.425:0 | accplot | atop | pos2d xy
```

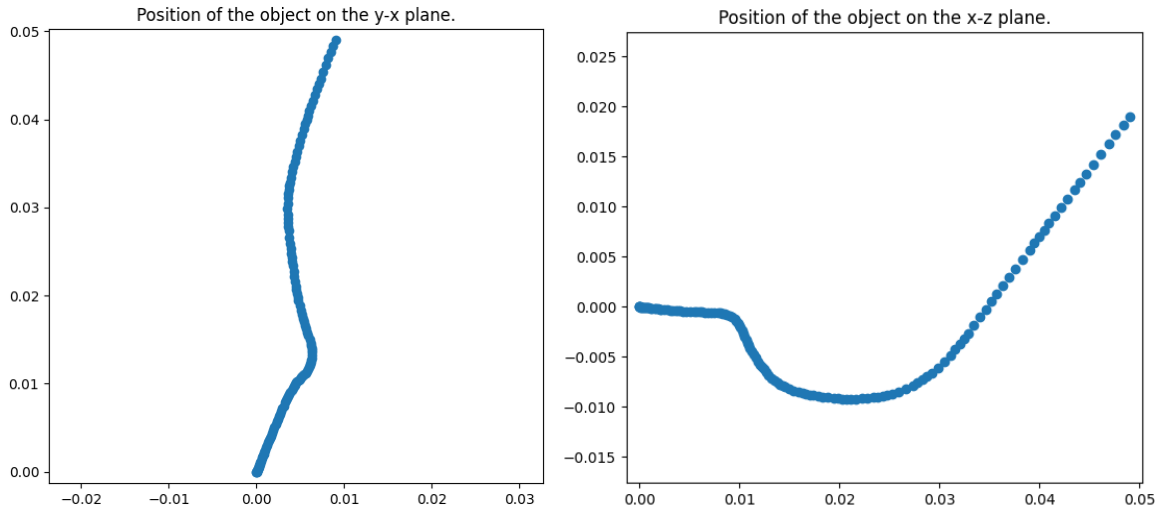
A másik mérés, ahol hasonlóan jónak tűnő adatokat kaptunk, az a hatos számú mérés volt. Ennél a mérésnél az adatgyűjtőt a penetrométer útjától 10 centiméterre jobbra tettük le. Abban különbözött továbbá ez a mérés a többitől, hogy -amint azt már említettem- ebben az esetben a mérőeszköz talajba helyezése után a felette lévő földet tömörítettük, hogy homogénebb környezetben tudjunk mérni.

A mérés végén azt tapasztaltuk, hogy az eszközünk 5 centimétert mozdult előre, 1 centimétert jobbra és kettőt felfelé.

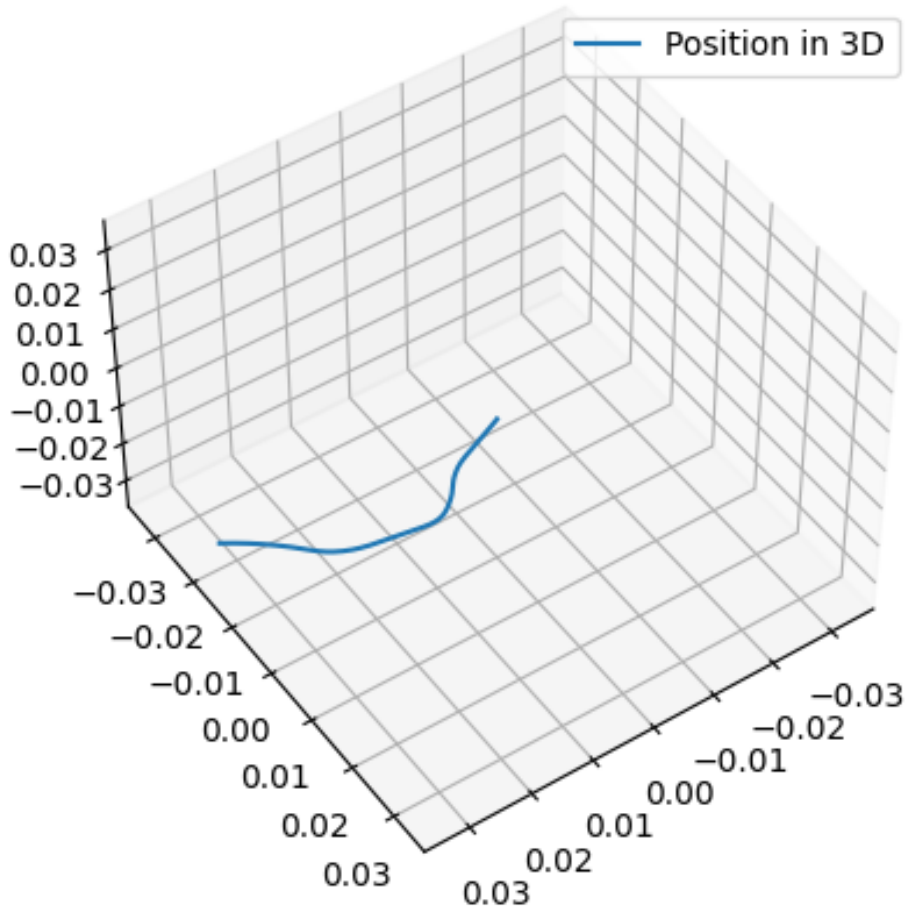
A mérési eredmények pontosításához az alábbi pipeline-t használtam:

```
loadf meres6.txt | mavg 3:xyz | zeroize i=3 | slice 90:225 | convert
| forr | sibha 0.05:0.01:0.02 | accplot | atop | pos2d yx
```

Mellyel az alábbi kép rajzolódott ki a megtett útról (ismét az X-Y síkon, azonban mivel az eszköz kiindulási orientációja más volt, mint a 4-es mérésben, ezt most Y-X síkként tudjuk értelmezni):



7.7 A 6. mérésben mért elmozdulás az Y-X és X-Z síkon (ahol (0:0) a kiindulási pozíciót jelenti)



7.8 A 6. mérésben mért elmozdulás a háromdimenziós térben

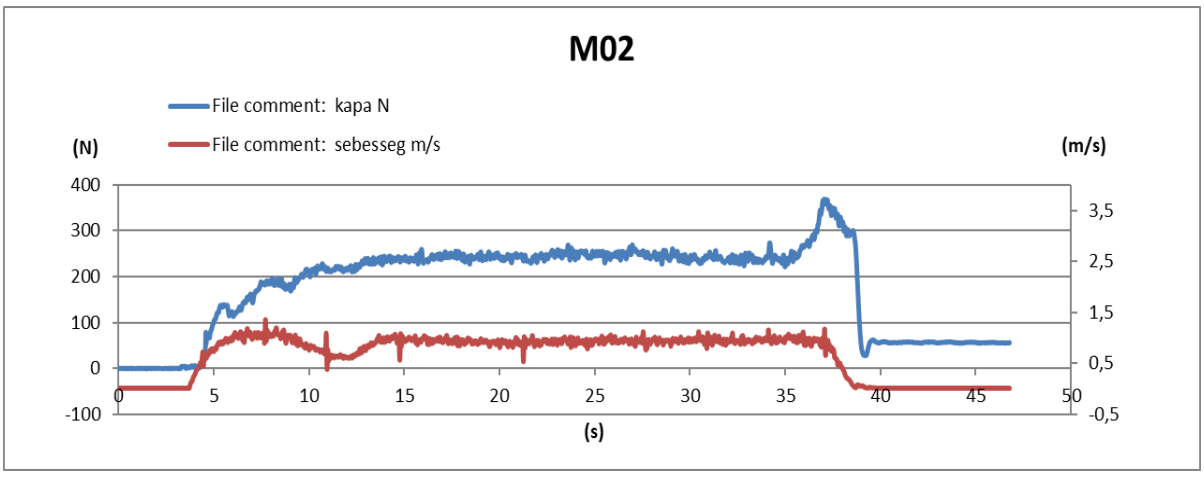
A feldolgozás során nyilvánvalóvá vált, hogy a jelenleg használt algoritmusok nem képesek arra, hogy külső referencia nélkül visszafejtsék a megtett utat, ezért a várt helyzetérték alapú eltoláskorrekció nem megkerülhető. Így a teljes elmozdulás megmérésére minden esetben szükség van.

7.1 A csatolt mérés eredményei

Amint azt már említettem, a párhuzamosan készülő mérésben a talaj horizontális penetrációs ellenállását vizsgáltuk (7.10). A mérések három különböző csúccsal (1. táblázat), három különböző sebességszinten (2. táblázat) történtek. Itt megjegyezném, hogy a talajelmozdulás vizsgálatakor mindvégig ugyanaz, a lenti ábrán 4-es számmal jelzett csúcs volt felhelyezve.

7.9 A penetrométer a vontatógépen az egyik mérés végeztével

Ezeknél a méréseknél a kapott eredmények azt mutatják, hogy a vontatási sebesség jóval nagyobb intervallumban mozoghat, mint a penetrációs ellenállás. A mérés előtt várt eredményeknek megfelelően az derült ki, hogy az egyes csúcsokkal végzett mérések jellegre megegyező tendenciát mutatnak. Ez alól kivételt képez az alábbi ábrán 2-es számmal jelölt csúcs, mely annyiban különbözik a többitől, hogy ennél a csúcs legnagyobb átmérője sem éri el az azt tartó rudazat átmérőjét.



7.10 A kettes számú mérés a négyes mérőkúppal

A mérésnél használt csúcsok jellemzői:

1. táblázat A felhasznált csúcsok jellemzői

Sorszám	Kúpfelület [cm^2]	Legnagyobb átmérő [mm]	Kúpvégi átmérő [mm]
2	2	15,96	15,55
3	3,33	20,60	20,08
4	5	25,23	24,59

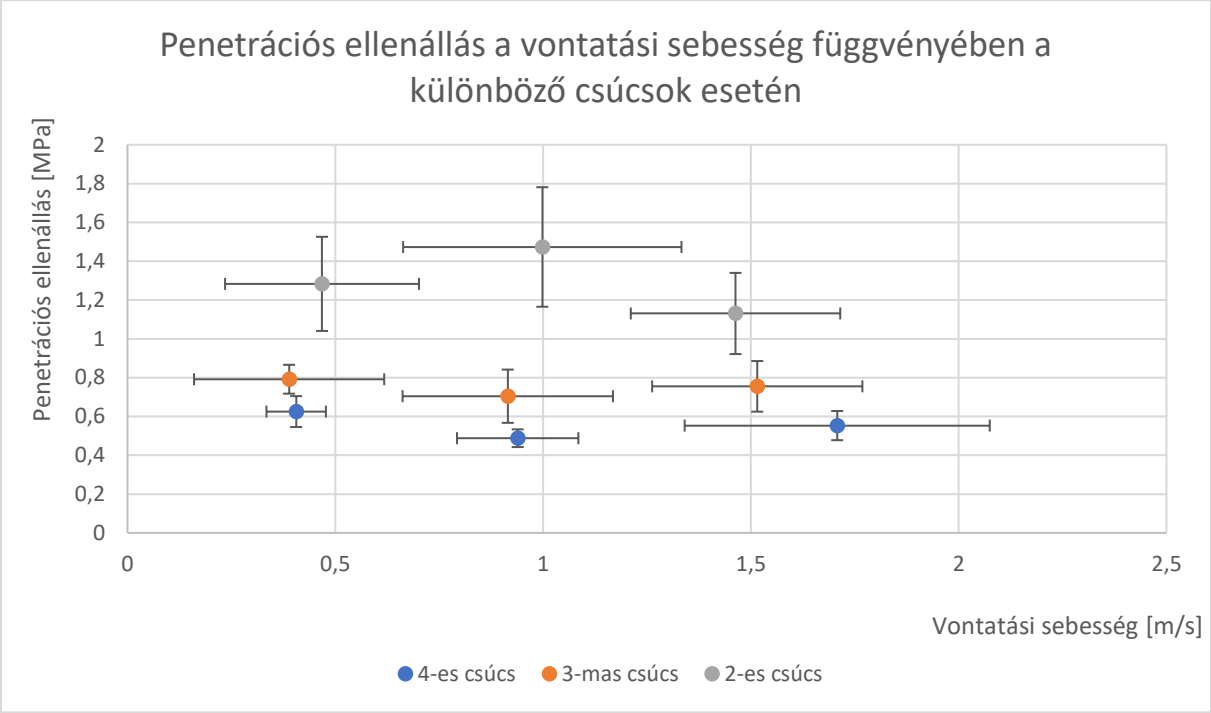
A kúpméreteken kívül a sebességek is előre megszabottak voltak, hogy a mért értékeket egymással össze lehessen vetni. Ezek a sebességszintek a 2. táblázat mutatja.

2. táblázat Mérési sebességértékek

Sorszám	Sebességérték [m/s]
1	0,5
2	1,0
3	1,5

Ezen mérések konklúziója az, hogy a penetrációs ellenállás értékének minimalizálásához létezhet egy adott vontatási sebesség mellett meghatározható optimális kúpméret, ami egy adott vizsgált szemcsés közeg típusára vonatkoztatható.

A mérés eredményeit az alábbi ábra foglalja össze:



7.11 A csatolt mérés eredményei

Figyeljük meg, hogy a vontatási sebesség hibája jóval nagyobb értékek között mozog, mint a penetrációs ellenállásé. Ez a mérés során fellépő terhelésingadozás miatt van, aminek hatására a vontatógép sebessége is ingadozik.

8 Értékelés

Amint említettem, egyelőre elengedhetetlennek tűnik, hogy az elmozdulást a mérés végeztével saját magunk mérjük le. Ettől eltekintve azonban a mért értékek egészen jónak mondhatók, a kapott eredmények összhangban vannak azzal, amit vártunk.

Ez azonban nem jelenti azt, hogy a mérőeszközt ne lenne érdemes tovább fejleszteni. Azt ugyanis sikerült bizonyítanunk, hogy a koncepció működőképes lehet, azonban a pontosságon és a felhasználóbarát kezelésen még lehet fejleszteni.

9 Összefoglaló / további tervek

A többféle további fejlesztési ötletek közül az első, hogy érdemes lehetne a mérőeszközt felruházni azzal a képeséggel, hogy meg tudja határozni a mérés végén a saját pozícióját, hogy ezt ne nehézkes és pontatlan eszközökkel kézzel kelljen megtennünk.

Egy további ötlet az, hogy egy helyett több szenzor alkalmazásával a vett jeleket egymással összehasonlítva azok zaját eredményesebben lehetne csökkenteni.

Ezen túl a kísérleteket más méretű és/vagy formájú házakba épített mérőeszközökkel is érdemes lenne megismételni, hiszen adott esetben ezek egész másképpen mozoghatnak, mint a most használt eszköz. Az ESP lapkák sokfélesége miatt elméletileg a néhányszor néhány centiméteres nagyságig el tudnánk jutni.

További terveink között szerepel, hogy az eszközt a valós körülményeket hűbben tükröző környezetben is kipróbáljuk, például a talajvályú helyett szántóföldön.

10 Irodalomjegyzék

1. Yang, W., Bajenov, A. & Shen, Y. Improving low-cost inertial-measurement-unit (IMU)-based motion tracking accuracy for a biomorphic hyper-redundant snake robot, 2197-3768 (2017).

2. Tamás K.: The role of bond and damping in the discrete element model of soil-sweep interaction, 1537-5110 (2018).

3. Milkevych, Viktor, et al. Modelling approach for soil displacement in tillage using discrete element method. Soil and Tillage Research, 2018, 183: 60-71.