



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

TUDOMÁNYOS DIÁKKÖRI DOLGOZAT

MOBILTELEFON MOZGÁSÁNAK PONTOSÍTÁSA KÁLMÁN SZŰRŐVEL

Mobilephone trajectory estimation using Kalman filter

Szerző:

Kovács Benedek Marcell

földmérő- és földrendező mérnök
BSc. szak, II. évf.

Konzulens:

dr. habil Molnár Gábor Péter

egyetemi docens

Székesfehérvár, 2022.

TARTALOMJEGYZÉK

1.	Bevezetés	4
1.1	Helymeghatározási feladatok csoportosítása pontosság szerint ..	4
2.	A telefon szenzorjai és az adattárolás	6
2.1	Az adattárolás	6
2.2	Felhasznált szenzorok.....	7
2.2.1	Műholdas helymeghatározó egység.	7
2.2.2	Gyorsulásmérő (Accelerometer)	9
2.3	Jövőben használni kívánt szenzorok.....	10
2.3.1	Szögsebességmérő (Gyroscope).....	10
2.3.2	Mágneses tér mérő (Magnetometer)	11
3.	Módszerek.....	12
3.1	Kálmán Rudolf Emil.....	12
3.1.1	Bevezetés.....	12
3.1.2	Élete és munkássága.....	12
3.1.3	A Kálmán szűrő előnyei.....	13
3.1.4	A Kálmán szűrő alkalmazása	13
3.2	Működésének alapja	13
3.3	Előfeltételek kielégítése.....	14
3.3.1	A MatLab, mint fejlesztői környezet.....	14
3.4	Adatfeldolgozás	15
3.4.1	A mérés leegyszerűsítése	15
3.4.2	A bias kiszűrése.....	16
3.4.3	A telefon órájának anomáliája.....	18
3.4.4	Egyéb feldolgozás	19

3.5	A Kálmán szűrő alkalmazása	21
3.5.1	Bevezetés.....	21
3.5.2	A jóslás.....	22
3.5.3	A frissítés	24
4.	Eredmények	25
4.1	A valóságghű mozgásra.....	25
4.2	A hibaellipszis	28
5.	Diszkusszió	30
5.1	Hasonló munkák.....	30
6.	Köszönetnyilvánítás.....	31
7.	Idézett forrásmunkák	32

1. BEVEZETÉS

Mindannyiunk zsebében egy nagyon komplex mérőműszer található. A mobiltelefonok egyre több beépített érzékelővel rendelkeznek, melyek a készülék és annak környezetének megannyi tulajdonságát képesek ismertetni a felhasználóval. Ezt a technológiai fejlődést szemlélteti például, hogy a piac legújabb termékei már nem egy közeli meteorológiai állomásról kérnek le az időjárást leíró adatokat, hanem azt a beépített légnyomás, hő- és páratartalom mérőikből képesek meghatározni, vagy akár a jövőre vonatkozó következtetéseket becsülni.

Dolgozatomban a telefon helymeghatározásához szükséges lehető legtöbb szenzor figyelembevételére tettem kísérletet, hogy megkapjam arra a kérdésekre a választ: *Mégis mennyire pontosan határozható meg egy átlagos mobil elhelyezkedése, illetve milyen pontosan modellezhető a mozgása?* Igyekeztem olyan érzékelőket felhasználni, amelyek általánosan ma már minden eszközben megtalálhatók, így munkám alapján bárki elvégezheti saját telefonja 'pontosságának' vizsgálatát.

A földmérő és földrendező mérnöki szakon tanulmányaim során megtanultam, hogy egy geodétának különösen fontos, hogy ismerje mérése mennyire volt pontos, mekkora középhibával rendelkezik. Munkámban azt próbáltam elérni, hogy a telefonos helymeghatározás pontatlansága ismertté és javíthatóvá, vagyis csökkenthetővé váljon. Ezt elérve az egyszerűbb navigációs, helymeghatározási feladatokat civilek is elvégezhetik a kezükben lévő mobil segítségével. Ez a felhasználás azért lenne hatalmas mérőföldkő, mert a feladat megoldásában mellőzi a kifejezetten erre a célra fejlesztett, telefonoknál nagyságrendekkel költségesebb eszközöket, és szakembereket.

1.1 Helymeghatározási feladatok csoportosítása pontosság szerint

Az előzőekben említett célkitűzésemet a következő skálán helyeztem el, amely bizonyos feladat helymeghatározáshoz köthető, mérési pontosságának igénye szerint sorbarendezett példákat tartalmaz.



Az ábra különböző feladatok mérési pontosságának igénye szerint van sorba rendezve fentről lefelé haladva, a több 10 méteres mérésektől eljutva az igen nagy precizitást igénylőkhig. Az ábrától jobbra látható jelölés mutatja, hogy kutatásommal a telefont, mint mérési eszközt hol lehetne elhelyezni.

2. A TELEFON SENZORJAI ÉS AZ ADATTÁROLÁS

2.1 Az adattárolás

A telefon szenzorjai által mért adatokat egy SensorData nevű alkalmazás segítségével tudtam felhasználni, ami egy egyszerű szöveges dokumentumba (.txt formátumban) mentette le a mérési idő során keletkezett adatokat. Ezek az állományok, egyszerűen beolvashatók bármilyen programnyelvben, így ideális volt számomra a választott telefonos applikáció.

A text fájl elején a mérés tulajdonságait leíró adatok találhatók. Ezek a következők: az adott szenzor neve, a mérésre felhasznált telefon gyártója és típusa, a szenzor maximális mérési intervalluma és a mérés felbontása.

Tartalmi felépítésük szempontjából egy táblázat következik, melynek oszlopai tabulátorral, sorai pedig sortöréssel vannak elválasztva. Minden sor egy időpillanatra vonatkozó mérések eredményét tartalmazza. Általánosan mindig az első oszlop tartalmazza az adott időpillanat pontos leírását, ez egy olyan szám, amely az 1970. január elseje óta eltelt ezredmásodperceket számolja, így biztosítva egyéni tulajdonságot minden mérési sornak. Néha azonban anomáliaként két külön méréshez (sorhoz), ugyan azon időpillanat-érték társul, ezt a hibát egy későbbi fejezetben figyelembe veszem és javítom. A többi oszlop a szenzorra jellemző mértékegységben, általában a telefon tengelyeinek irányába mért, más-más méréseket tartalmaz.

```

*test_accelerometer_t3 - Jegyzetfőm
Fájl Szerkesztés Megtekintés

Trial: test_accelerometer_t3.txt
Phone/Tablet Model: Xiaomi M2003J15SC

Sensor Information
Name: bmi160_acc ACCELEROMETER
Maximum Range: 78.4532m/s²
Resolution: 0.0012m/s²

Unix Time (ms since Jan 1, 1970) X Y Z
1666688757847 -1.0520000457763672 0.32899999618530273 9.812000274658203
1666688757847 -1.274999976158142 0.15600000321865082 10.187999725341797
1666688757847 -1.152999997138977 -0.003000000026077032 10.21399974822998
1666688757847 -0.875 -0.04899999871850014 10.395999908447266
1666688757847 -0.7649999856948853 -0.006000000052154064 10.748000144958496
1666688757847 -0.7599999904632568 -0.25200000405311584 11.267999649047852
1666688757847 -0.6169999837875366 -0.47699999809265137 11.229000091552734
1666688757847 -0.43700000643730164 -0.5440000295639038 11.279999732971191
1666688757847 -0.33399999141693115 -0.4869999885559082 11.253000259399414
1666688757847 -0.32199999690055847 -0.3580000102519989 11.34000015258789
1666688757847 -0.29600000381469727 -0.2619999945163727 10.98799991607666
1666688757847 -0.21899999678134918 -0.07999999821186066 10.621000289916992

7. sor, 23. oszlop Unix (LF) UTF-8

```

1 Egy adatokat tartalmazó szöveges állomány, melyet a SensorData készített automatikusan.

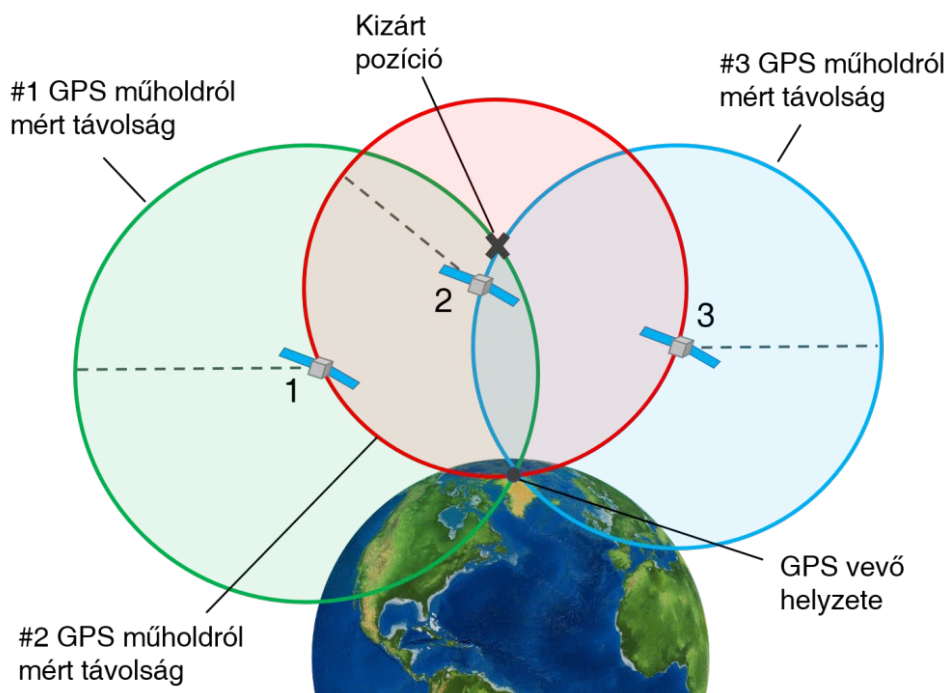
2.2 Felhasznált szenzorok

2.2.1 Műholdas helymeghatározó egység.

A műholdrendszerek segítségével egy mára minden országban ismertté vált helymeghatározáshoz használt összetett technológia fejlődött ki. [1] [2] Alkotó elemei közül kulcsfontosságú a Globális Navigációs Műholdrendszer, röviden GNSS, amely lehetővé teszi a világ bármely pontján a rendszer elérhetőségét. A GNSS rendszerek egyike az amerikai fejlesztésű GPS (Global Positioning System). Ez a műholdrendszer 24 darab, a Földet naponta kétszer megkerülő 6 különböző pályasíkon keringő műholdból áll. A műholdakat a Földről tartják karban, üzemeltetik, melyhez 5 darab monitor, 4 darab feltöltő állomás, és 1 központi vezérlőpont tartozik. A rendszerhez még egy elem hiányzik, a felhasználó egy vevő készüléke, amely képes fogadni a rádióhullámokat. Erre szolgál a telefonban lévő vevő is mely -a többi hasonló eszközre is igaz- 4.9 méteres pontosságot képes elérni. [3] Az egyszerűség kedvéért a későbbiekben összefoglalóan a rendszerre GPS néven hivatkozok, mivel

munkám során én is ezt a rendszert használtam koordináta adatok szerzésére

Maga a helymeghatározás a GPS segítségével időmérésre vezethető vissza, amely a vevő készülék órája és a műholdak higanyionos atomórái közt valósul meg. A műholdak rádióhullámokat bocsátanak ki, melyeknek terjedési sebessége, a Föld légköre által okozott késleltetése és torzítása ismertek, illetve számba vehetők. A két óra közt pontosan meghatározott időeltérés, és a rádióhullám sebessége ismeretében távolság számítható minden műholdra, amely jel-küldési és fogadási távolságra egyaránt megfelelő. Általában 8-10 műhold fogható a nap bármely órájában, a matematikailag szükséges 3-hoz képest ez bőven ad lehetőséget fölös mérések rögzítésére, a helymeghatározás kiegyenlítéssel történő pontosítására. A műholdak pontos helyzetéből és a vevőkészüléktől való távolságukból a geodéziában térbeli hátrametszésként ismert számítással meghatározhatóak a telefon pontos helyzetét leíró koordináták. Ezeket a koordinátákat NMEA szabványos adatformátumként tároljuk.



A GPS rendszer távolságokból meghatározott koordináta számításának szemléltetése [1]

A kapott koordináták azonban nem méterben vannak megadva, így ezt a következő képlettel még át kell váltani úgynevezett É északias és K keleties koordinátákba.

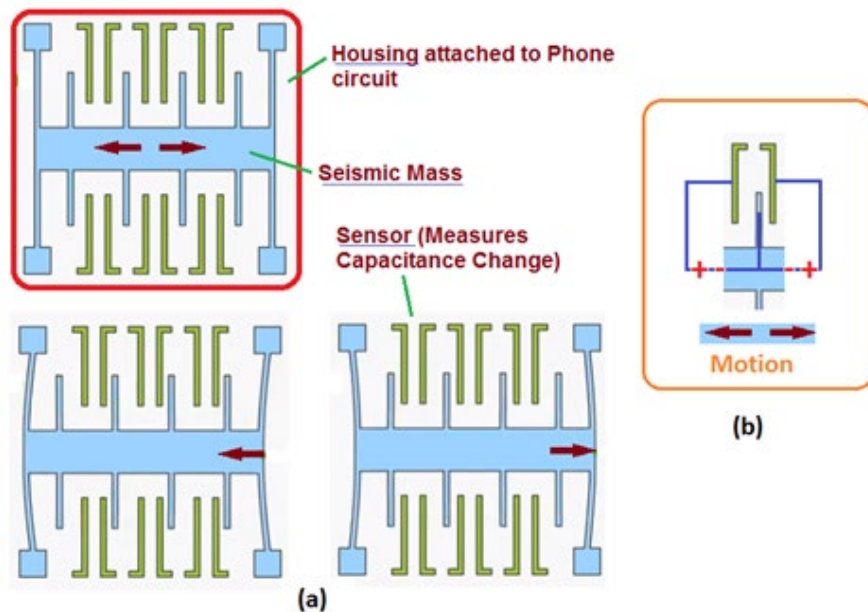
$$\dot{E}_{\text{méter}} = \text{GPS}_{\text{szélesség}} \cdot 111000$$

$$K_{\text{méter}} = (\text{GPS}_{\text{hosszúság}} \cdot 111000) \cdot \cos(\text{GPS}_{\text{szélesség}})$$

Munkámban a méterbe átváltott GPS koordinátarendszert tekintetem a modellezés koordinátarendszerének, így ez ideális, hiszen semmilyen transzformációra nincs szükség a továbbiakban.

2.2.2 Gyorsulásmérő (Accelerometer)

A gyorsulásmérő szenzor egy olyan érzékelő, amely a telefon mozgását, annak saját a telefonhoz kötött, úgynevezett lokális tengelyein megjelenő komponenseit méri.[4] Működésének alapelve, hogy a telefonhoz rögzített ellenállásmérők közt szeizmikusan szabadon mozgó, a tömegvonzás által elmozdított elemek befolyásolják a mért kölcsönhatást. Ezen értékek egymáshoz viszonyított mértékét feldolgozva képes becsléseket végezni a tengelyek irányába. Ez azonban hibákkal terhelt.



MEMS szenzor, gyorsulásmérő a telefonban

[4]

A legszembe ötlőbb gond a Föld gravitációs vonzása, amely gravitációs gyorsulásként, a bolygó középpontja felé, átlagosan 9.81m/s^2 gyorsulással folyamatosan hat minden földi tárgyra. Telefonunk ezt a gyorsulást is érzékeli, s ez alapján detektálja például, hogy a készülékünk 'álló' vagy 'fekvő' pozícióban van-e. Amennyiben ez a tömegvonzási erő nem egyezik meg a mobilnak valamely lokális tengelyével, a többi tengely irányában is meg fog jelenni valamekkora hányada, így hibával terhelve mérésünket. Erre kiemelten oda kell figyelni, hiszen akár kismértékű 'ferdeség', a mérési idő növekedésével arányosan torzítja a mérést.

A másik akadály, hogy az okostelefonokban dinamikus gyorsulásmérők vannak. Amint már korábban is említettem, a telefon koordinátarendszere nem egyezik meg a végleges modellezési koordinátarendszerrel. Ez azonban egy számítás előtti forgatómátrixsal, mérési időpillanatonként pedig a szögsebességmérővel a lokális elfordulást figyelembe véve átszámíthatók a gyorsulások a globális tengelyek irányába.

2.3 Jövőben használni kívánt szenzorok

2.3.1 Szögsebességmérő (Gyroscope)

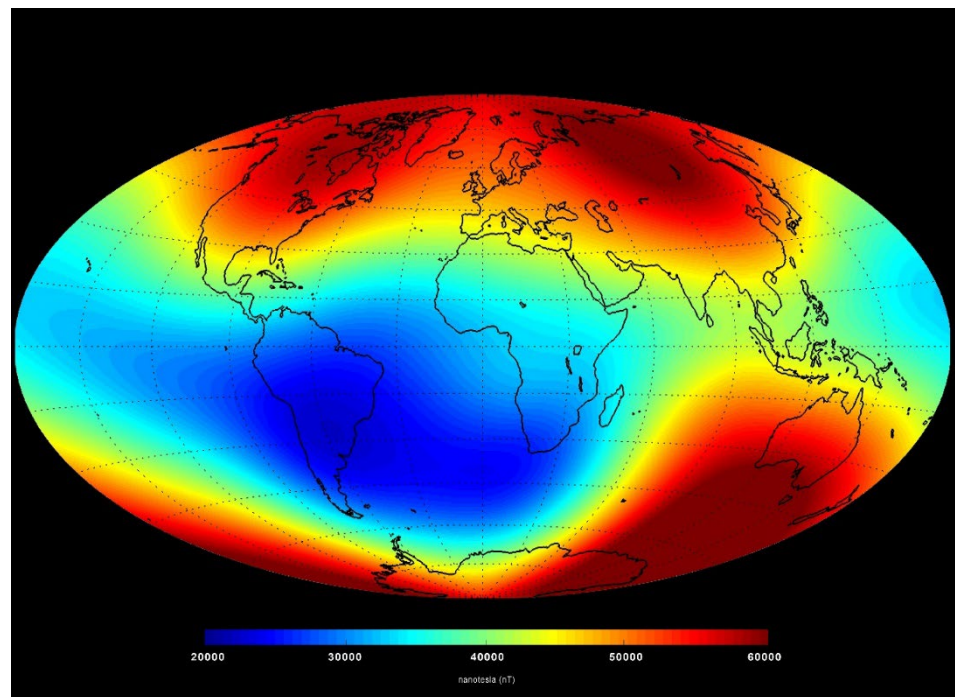
Az ember az elfordulást, a tájolást nehezen érzékeli. Ehhez azonban segítséget nyújt a telefon szögsebességmérője. A magyar elnevezése a 'gyroscope' -nak onnan ered, hogy a már említett készülék három lokális tengelyéhez képest milyen sebességgel fordul el. Mérési mértékegysége rad/s. A telefonban lévő gyroscope-ok, melyek a MEMS rendszer része méretükből és költségükből fakadóan nem tartalmazznak folyamatosan vákuumban pörgő részeket, így a korábban szintén így nevezett eszközökkel ellentétben nem alkalmasak tájékozásra, ezt a szerepet a gyorsulásmérő és a magnetométer tölti be.

Méréseim során nem vettem figyelembe, pontosabban nullának tekintettem az elfordulást, mivel egyenes vonalú közel egyenletesen változó mozgást végeztem. A jövőben szeretném folytatni kutatásomat, és

implementálni a szögsebességmérőt a számítási módszerbe, hiszen így jóval komplexebb mozgásokat is képes lennék reprodukálni.

2.3.2 Mágneses tér mérő (Magnetometer)

A következő szenzor a Föld mágneses mezejének telefonra értendő vonzását a mágneses fluxussűrűség mértékegységében, nanoteslában adja meg a készülék három tengelyére vonatkozóan. A három komponensből egy vektor számítható, mely a Föld mágneses északi pólusa felé mutat. A magnetométer az orientáció, tájolás szempontjából szükséges, hiszen megállapítható, hogy a telefonhoz képes milyen térbeli irányban található észak.



A Föld mágneses mezeje által kifejtett vonzás, melyet a magnetométer képes mérni [5]

Jelentős problémát okoz azonban, hogy a legtöbb tárgy rendelkezik mágneses mezővel, amely kis mértékben, de sokszor és sok irányban, tehát kiszámíthatatlanul torzítják a szenzor pontosságát.

3. MÓDSZEREK

3.1 Kálmán Rudolf Emil

3.1.1 Bevezetés

Az első probléma, ami felmerült, hogy bár minden felhasznált szenzor valamilyen szinten kapcsolódik a helymeghatározáshoz, mindegyik különböző, mértékegységében különböző tulajdonságokat határoz meg. Ezzel a problémával, és ennek megoldásával az úgynevezett sensor fusion-el a Kálmán szűrő foglalkozik, amely az egész dolgozatom központi eleme. A következő fejezetben Kálmán Rudolf rendszerének kialakulásáról, előnyeiről és működéséről, egy későbbi fejezetben pedig a feladatomba való implementációjáról fogok beszélni.

3.1.2 Élete és munkássága

Első sorban a szűrő szellemi atyját, Kálmán Rudolf Emilt ismertetem. [6] Az 1930-ban született, 2016-ban elhunyt magyar-amerikai villamosmérnök és matematikus a műszaki tudományok terén végzett számos kutatást. Kifejezetten érdekelte, hogy a földi tárgyak, folyamatok és mozgások állapotait hogyan lehet meghatározni, megbecsülni. Számos tudományos publikációja ehhez köthető, mint például a sokdimenziós állapotérmodellek axiomatikus elmélete, ezen túl továbbfejlesztette a lineáris dinamikus rendszerek elméletét. A matematika területén a statisztikával foglalkozott, ahol többek közt megalkotta a nemlineáris diszkrét rendszerek állapotterére alkalmazható matematikai eljárást, a Kálmán szűrő elvét.

Az Óbudai Egyetem őt választotta legelső díszdoktorává 2010. június 30-án. Később az Óbudai Egyetemet fenntartó alapítványt is ő róla nevezték el.



Kálmán Rudolf Emil (1930-2016) [6]

3.1.3 A Kálmán szűrő előnyei

A metódus nagyszerűsége több tulajdonságában is megnyilvánul. A mért, hibával terhelt adatokból kiszűrhetővé, figyelembe vehetővé válik a zaj, a bizonytalanság és a pontatlanság. Hiányos információk esetében is rekurzív paraméterbecsléssel megállapítható az adott rendszer időállapota. Működésének részletes leírására később visszatérek.

3.1.4 A Kálmán szűrő alkalmazása

Kezdetben az Apollo-programban alkalmazták az eljárást, az űrhajók, és a Hold-szondák navigációs rendszerének optimalizálására. Navigáción túl a repülésirányításhoz használták a '60-as évektől. Később tovább gondolva rendszermodellezéshez, és az algebrai rendszerelmélet megalkotásához használta fel a szűrőt. A Kálmán szűrő túlnőtte névadóját, hiszen a mai napig alkalmazzák az algoritmust repülőgépeknél, űrhajóknál, robotrepülőgépeknél és egyéb jelfeldolgozó rendszerekben.

3.2 Működésének alapja

[7]Működésének alapja, hogy az érzékelők által kifejezett adatokat a telefon állapotát leíró, egymással függő vagy független kapcsolatban álló állapot-komponenseknek tekintjük. Minden időpillanatban meghatározunk egy jóslást arról, hogy szerintünk mi lesz a mobil következő állapota, melyet az előző idő-pillanatbeli állapotából becsüljük.

Ezt követően a szenzorok éppen mért adatainak figyelembevételével megjavítjuk becslésünket. A szenzorok pontosságára is tekintettel vagyunk, ezt megadva a módszer magától határozza meg az adott állapotkomponensekre adott javítások súlyát. Így a kevésbé pontos szenzorok (pl.: GPS) kisebb mértékben befolyásolják a meghatározott helyzetet.

A beolvasott nyers adatokat elsősorban fel kell dolgozni, a program belső adataiként el kell tárolni. A következő fejezetben bemutatom mely szenzorokat használtam fel, azok mérési eredményeit milyen módon mentettem el.

3.3 Előfeltételek kielégítése

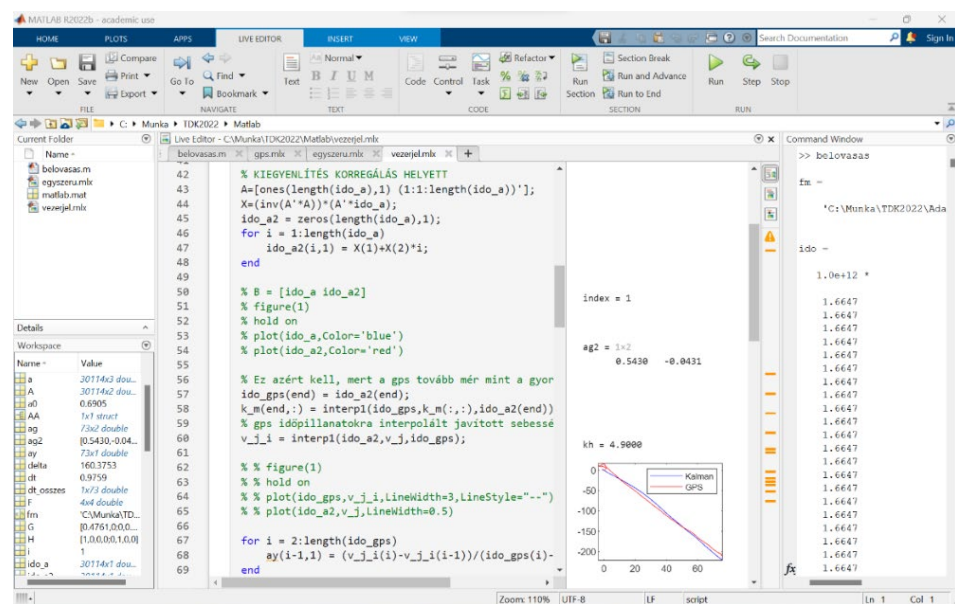
Minden mérésemet ugyanazon telefon használatával végeztem el. A mobil típusa Xiaomi Redmi Note 9, ami teljesen alkalmas volt az adatszerzéshez, hiszen a felhasznált szenzorokkal együtt összesen 11 különböző érzékelővel rendelkezik. Ezen műszerek mérési tartománya, adott idő alatt érzékelhető mérések száma (mérések frekvenciája) és a mérések középhibája is megtalálható a gyártó honlapján, illetve a már említett telefonos applikációban.

A SensorData által eltárolt text formátumú adatokat áthelyeztem számítógépre, ahol a további számításokat végeztem a MatLab nevű alkalmazásban. A következőkben a Kálmán szűrő programba ágyazott megoldását és a problémamegoldások menetét fejtem ki, a MatLab rövid ismertetését követően.

3.3.1 A MatLab, mint fejlesztői környezet

A MathWorks által fejlesztett program egy olyan numerikus számításokra specializált logikával rendelkezik, ami a mátrixműveletek gyors kiszámítását teszi lehetővé. Ezen kívül a felhasználókat helyezi előtérbe azáltal, hogy képes függvények és adatok ábrázolására, algoritmusok implementációjára és felhasználói interfész egyéni igények szerinti kialakítására. Programozási nyelve szerint M-code alapú, amit nagyrészt

felhasználóbarátnak ítélem meg, mivel interaktív sűgóval és egyéni függvények készítésének lehetőségével is rendelkezik. Logikailag a Python programozási nyelvhez hasonlít, aminek alapja a C nyelv, így az egyetemen eltanult programozási tudásomat is felhasználhattam, viszonylag gyorsan tudtam haladni a gyakorlati megvalósítással. A szoftver használatához szükséges hozzáférést, a licenszt az Óbudai Egyetem biztosította.



A MatLab felhasználói környezete, amiben megvalósítottam a számolás gyakorlati részét.

3.4 Adatfeldolgozás

3.4.1 A mérés leegyszerűsítése

Az érthető szemléltetés érdekében -habár a pontosság kárára- munkámban egy leegyszerűsített mozgást, ezzel együtt egy leegyszerűsített mérést és számítást végeztem. A mérést úgy végeztem el, hogy a okostelefonomat egy kerékpárra helyezett telefontartóba helyeztem. A járművel a hosszúsáttatéri kollégium utcájának egy háztömbnyi hosszúságú szakaszán mértem, ahol kvázi egyenesen tudtam haladni, így a mozgásomat egy egyenes vonalú közel egyenletesen változó modellként tudtam számításba venni. A telefont úgy helyeztem el a kerékpáron, hogy annak egyik helyi tengelye (nevezetesen az y) megegyezzen a mozgásom irányával. Így kizárva a gyorsulásmérő másik két tengelyen mért értékeinek- és a

szögsebességmérő szükségességét a mozgásom egyenes vonalúnak tekintésével.

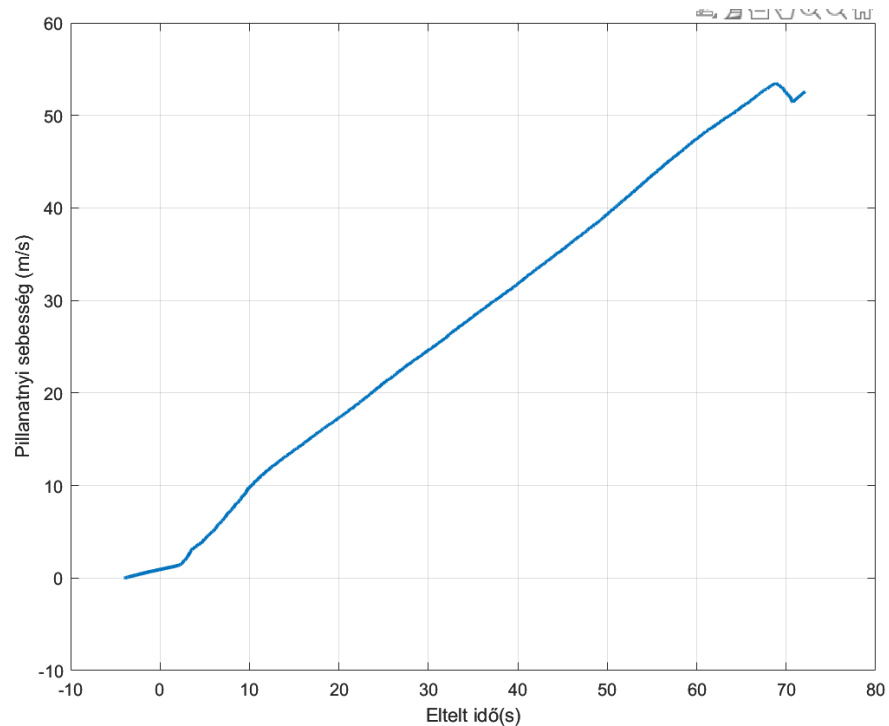
3.4.2 A bias kiszűrése

A nyers adatokat az *importdata()* függvénnyel olvastam be a program rövidtávú memóriájába, majd vektorokba és mátrixokba mentettem el a beolvasott fájl oszlopait. A gyorsulásmérő nyers adatai többféle hibával terheltek, ide tartozik az úgynevezett 'bias', magyarul elferdülés is. A hiba eredete, hogy a mérőkészülék nem lett tökéletesen vízszintesbe állítva, így a mért gyorsulás valamennyivel több a valóságnál. Az ebből számított pillanatnyi sebesség az idő múlásával egyenletesen torzul. A pillanatnyi sebesség a következőképpen számolható:

$$\bar{v}_i = \bar{v}_{i-1} + \bar{a}_{i-1} \cdot (\bar{s}_i - \bar{s}_{i-1})$$

ahol a kezdő sebességet (v_0) nullának tekintjük.

Ez elvégzendő minden komponens minden időpillanatára. A következő, 1) számú ábrán a telefon y irányú pillanatnyi sebességét ábrázolom az eltelt idő szerint, amely jól szemlélteti a jelenséget.



1) A bias-t, vagyis az elferdülést tartalmazó gyorsulásból számított pillanatnyi sebesség

A bias javítása következő egyenlettel számítható:

$$a_0 = \frac{v_n - v_1}{s_n - s_1}$$

ahol n a mérések száma

A mérést több időintervallumra osztva, több bias korrekciós együtthatót számolva pontosabb javítást érhetünk el, azonban az egyszerűség kedvéért, én ugyanazzal az értékkel javítottam az összes gyorsulásértéket. Mivel a bias minden tengely mérését, más mértékben befolyásolja, így más-más javítási együtthatót kell számítani a különböző irányokba. A hibával terhelt sebesség a következő képlettel javítható, az immáron kiszámított a_0 -val.

$$\bar{v}_i = \bar{v}_{i-1} + (\bar{a}_{i-1} - a_0) \cdot (\bar{s}_i - \bar{s}_{i-1})$$

Ügyeljünk arra, hogy a sebességekhez a hozzájuk tartozó javításokat alkalmazzuk.

3.4.3 A telefon órájának anomáliája

A következő akadályt az okozta, hogy a telefon órája, mely minden szenzor mérésének első oszlopában lévő idő meghatározásáért felel, véletlenszerű időközönként rossz, hamis időpillanatot határoz meg. Ennek oka feltételezhetően az lehet, hogy a telefon processzora teljesítményének határán volt, hiszen 2-3 ezredmásodpercenként számtalan adatot kell kezelnie, tárolnia a mérés során. Habár a hiba elhanyagolható volna, ez mégis gondot okoz számunkra. A következőkben a hiba figyelembevételének okát fejtem ki. A mobil GPS vevője közel másodpercenként képes mérni, a 2-3 ezredmásodpercenként adatot tároló gyorsulásmérőhöz képest. Hogy a Kálmán szűrőt alkalmazni lehessen, egységes időközöket kell alkalmazni a mérésekre. A gyorsulásmérő GPS-mérési időnkénti értékét interpolálással becsültem meg. Ennek az *interp1()* parancsnak a feltétele, hogy minden kezdőindexnek egyedi értéket kell felvennie. Ez azonban nem minden esetben teljesült.

A problémát úgy szüntettem meg, hogy az időpillanatok értékét, és indexét pontok koordinátájaként értelmeztem. Ezen pontokra kiegyenítő számítással egy, a pontokra legjobban 'rásimuló' egyenes illeszthető, melynek két paramétere (a_0 – y tengely metszete és a_1 - egyenes meredeksége) alapján írható fel. A paraméterek megismeréséhez szükség van az $\bar{A}$ alakmátrixra, és az $\bar{l}$ tisztatag vektorra. Az alakmátrix oszlopai az egyenes egyenletének paraméterek szerinti deriváltjai, sorai pedig az adott mérés értékeinek (x - időmérés indexe, y – időmérés értéke) behelyettesítésével tölthetők fel. A tisztatagot jelen esetünkben a mért idő értékei fogják feltölteni.

$$\bar{A} = \begin{bmatrix} 1 & i \\ 1 & i+1 \\ \vdots & \vdots \\ 1 & n \end{bmatrix}; \bar{l} = \begin{bmatrix} s_i \\ s_{i+1} \\ \vdots \\ s_n \end{bmatrix}$$

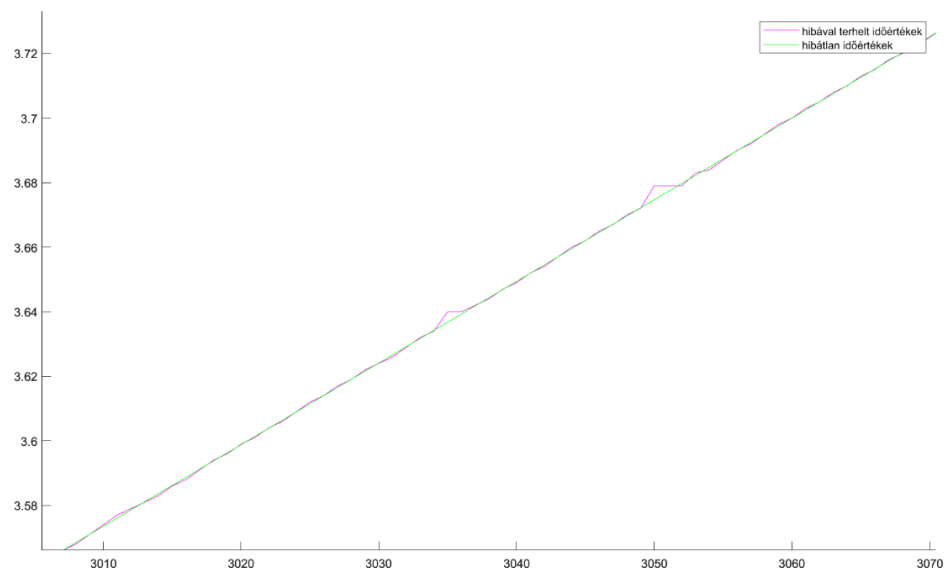
Az egyenes paramétereit egy $\bar{x}$ vektorban kapjuk meg következő képpen:

$$\bar{x} = (\bar{A}^T \cdot \bar{A})^{-1} \cdot (\bar{A}^T \cdot \bar{l})$$

Ezt követően minden addigi gyorsulásmérő-időpillanatra kiszámoltam, hogy a fentebbi paraméterekkel rendelkező egyenes az adott helyen milyen értéket venne fel. Mivel az egyenes a tengelyeken ferdén helyezkedik el, így biztosan nem lesz két azonos értékkel rendelkező időpillanat.

$$\bar{s}_i = \bar{x}(1) + \bar{x}(2) \cdot i$$

A következő, 2) számú ábrán magenta színnel az eredeti telefon órája által mért időpillanatok, zölddel a kiegyenlítővel számolt egyenesen elhelyezkedő időpillanatok láthatóak. Az óra időanomáliái jól megfigyelhetők azokon a helyeken, ahol az egyenestől eltérnek.



2) A telefon órájából származó időanomáliák lilával, illetve a helyettük használt kiegyenlített időpillanatok zölddel vannak jelölve.

3.4.4 Egyéb feldolgozás

Ahhoz, hogy a Kálmán szűrőt alkalmazni tudjuk, még két utolsó adatátalakításra szükség van.

Időközökre értelmezett mérések

Az első teendő a fentebb említett általános időközök megválasztása. A programomban a GPS időpillanatait tekintetem ideálisnak, mivel a gyorsulásmérő könnyebben és pontosabban alkalmazkodik a ritkásabb

mérésekkel rendelkező szenzorhoz, mint fordítva. Ehhez az *interp1()* funkciót használtam a következő szintaktikával:

$$\bar{v}_{GPS \text{ időnkénti}} = \text{interp1}(\bar{s}_{gyorsulásmérő}, \bar{v}, s_{GPS})$$

A program nyelvén ez azt határozza meg, hogy az anomáliamentes gyorsulásmérő-időpillanataiban értelmezzük a bias-mentes pillanatnyi sebességeket, majd a GPS-időpillanatokhoz becsüljük a hozzájuk tartozó sebességértéket. A kapott pillanatnyi sebességek vektorának elemei segítségével megbecsüljük azt a vektort, amely azokat a gyorsulásokat tartalmazza, melyeket mérnie kellett volna egy olyan gyorsulásmérőnek, mely pont a GPS-időpillanatokban mér kizárólag.

$$\bar{a}_{i-1} = \frac{(\bar{v}_i - \bar{v}_{i-1})}{(\bar{s}_i - \bar{s}_{i-1})}$$

A kapott $\bar{a}$ gyorsulás-vektor természetesen egy háromszlopú gyorsulás mátrix lenne, amennyiben a másik két tengely irányába történő mozgásunkat nem nullának tekintenénk.

Forgatómátrix a két rendszer között

Amennyiben az okostelefonunkat vízszintesnek tekintjük, az eszköz lokális- és a modellezésünk globális koordinátarendszere közti átszámítást egy egyszerű forgatómátrix biztosítja, legyen a jele: $\bar{M}$. Felírásához ismernünk kell a δ elforgatási szöveget. Ezt a GPS már méteres mértékegységű értékeinek utolsó tagjaival, és az *atan2()* függvénnyel számíthatjuk ki, a következő kódsorral, ahol n a mérések száma:

$$\delta = \text{atan2}(\text{GPS_X}(n), \text{GPS_Y}(n))$$

Az eredményt radiánban kapjuk meg, így a fent megkapott δ -t még meg kell szorozni $\frac{180^\circ}{\pi}$ -vel. A forgatómátrix a következőképp fog kinézni:

$$\bar{M} = \begin{bmatrix} \cos \delta & -\sin \delta \\ \sin \delta & \cos \delta \end{bmatrix}$$

Az utolsó lépés, hogy elforgassuk a telefon helyi méréseit, ezen esetben a GPS, és az y irányban mért gyorsulást. GPS mérésenként, a két mért koordinátánkat oszlopvektorként értelmezve, a forgatómátrixsal baloldaltól megszorozzuk.

$$GPS_g(1,2) = (\bar{\bar{M}} \cdot [GPS_l(1) \quad GPS_l(2)]^T)^T$$

A gyorsulásmérő 'helyére forgatása' hasonló az előbb felírt képlethez, azonban mivel nullának tekintettem az x-irányú gyorsulást, így az egyenlet jobb oldalán 0 értéket fog felvenni. Amennyiben figyelembe akarjuk venni a haladási irányunkra merőleges gyorsulást, azt a 0 helyén kell megtenni. Ne feledjük, az $\bar{a}_y$ alatt **nem** az eredeti gyorsulás mérő értékei értendők, hanem a GPS-időpillanatokra becsültek.

$$\bar{a}_g = (\bar{\bar{M}} \cdot [0 \quad \bar{a}_y]^T)^T$$

3.5 A Kálmán szűrő alkalmazása

3.5.1 Bevezetés

A Kálmán szűrő két fő szegmensből épül fel: jóslás (predict) és frissítés (update). A következő részben erre a két egységre bontva vezetem végig a szűrő elméletét, a programkódba fejtett példákkal végig kísérve. Az imént említett kettő lépést egy úgynevezett állapotvektorra $\bar{x}_{k|k}$, és annak kovariancia mátrixára $\bar{\bar{P}}_{k|k}$ értelmezzük. Az állapotvektor az az oszlopvektor, amelynek alkotó elemei leírják a vizsgált tárgy állapotának számunkra lényeges tulajdonságait. Ilyen komponensek jelen esetünkben a telefon x és y irányú globális koordinátái, illetve az x és y tengelyek irányába mutató pillanatnyi sebességet leíró vektorok nagysága.

Komplexebb mozgás modellezésének számításához több és több elemmel egészülhet ki az állapotot leíró adatok halmaza. A következőkben néhány példát sorolok fel ezekre. Amennyiben térben vizsgálánk a mozgást, z irányú koordináta, illetve pillanatnyi sebesség is megjelenne. Ha nem egyenes vonalú mozgást is vizsgálni szeretnénk, szükség volna minden időpillanatban újabb $\bar{\bar{M}}$ mátrix felírására, amely ezúttal nem egy egyszerű

síkbeli forgatási mátrix lenne, hanem egy 4 oszlopból és sorból álló kvaterniós forgatómátrix. [8] Minden új $\bar{\bar{M}}$ mátrix felírásához szükség van egy $\bar{\bar{R}}$ mátrixra is, mely a forgató mátrixot transzformálja. Az $\bar{\bar{M}}$ és az $\bar{\bar{R}}$ mátrix is 4-4 együtthatóval helyezhetők bele az állapotvektorba. Az immár 14 tagú oszlopvektor igen komplex mozgás követésére képes, mely tovább pontosítható a magnetométer adataival. Amennyiben munkámat folytathatom, a jövőben mindenképp céljaim közt van ennek a bonyolult rendszernek a megvalósítása MatLab-ban.

3.5.2 A jóslás

A szűrőt minden időpillanatban, jelen esetben minden GPS-mérési alkalommal használnunk kell, ezért érdemes szintaktikailag egy ciklusba foglalni a számítást. A jóslás lényegében azt jelenti, hogy a telefon előző időpillanat-béli $\bar{x}_{k-1|k-1}$ állapotát és $\bar{\bar{P}}_{k-1|k-1}$ kovarianciamátrixát állandónak tekintjük, és az alapján becsüljük meg mi is volna a következő pillanatban az $\hat{x}_{k|k-1}$ állapotvektor, illetve a hozzátartozó $\hat{\bar{P}}_{k|k-1}$ kovarianciamátrix. Ehhez szükség van több nevezetes mátrixra is.

Nevezetes mátrixok

Az $\bar{\bar{F}}_k$ (néhol $\bar{\bar{A}}$ -val jelölt) állapot-átmenet mátrixra, amely két egymástól eltérő időbeli állapot közt határozza meg a komponensek egymástól való függését. Ez a mátrix jelen esetben állandó és a következő képpen írható fel:

$$\bar{\bar{F}}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ahol Δt a két mérési időpillanat közt eltelt idő másodpercben.

Hasonlóan állandó a $\bar{\bar{G}}_k$ (máshol $\bar{\bar{B}}$ -vel jelölt) vezérlőjel-állapot kollineációs mátrix. Ez a mátrix a vezérlőjelnek tekintett mérések állapotvektorra gyakorolt befolyását határozza meg. Szerkezetileg ez esetben 2 oszlopból és 4 sorból áll:

$$\bar{\bar{G}}_k = \begin{bmatrix} \frac{1}{2} \cdot \Delta t^2 & 0 \\ 0 & \frac{1}{2} \cdot \Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}$$

A harmadik változatlan mátrix, a $\bar{\bar{H}}_k$ (vagy $\bar{\bar{C}}$) állapot-mérés mátrix, amelyre a frissítés résznél lesz szükség, azonban az átláthatóság érdekében előrébb vettem ismertetését. Alakja jelenleg 4 oszlopból és 2 sorból áll, funkciója az, hogy az előzetes (jósolt) állapotvektorral összeszorozva, megkaphatjuk a jósolt mérést.

$$\bar{\bar{H}}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Számítás

A legelső állapotot és hozzá tartozó kovarianciamátrixot 0-kal feltöltött vektornak és mátrixnak tekintjük. Az előzetes (jósolt) állapotvektorhoz szükséges: az előző állapot, az adott időpillanatbeli vezérlőjel (esetünkben a gyorsulás, tehát $\bar{u}_k = \bar{a}_g$), a nevezetes mátrixok közül pedig az $\bar{\bar{F}}_k$ és a $\bar{\bar{G}}_k$.

$$\hat{x}_{k|k-1} = \bar{\bar{F}}_k \cdot \bar{x}_{k-1|k-1} + \bar{\bar{G}}_k \cdot \bar{u}_k$$

Az előzetes állapothoz tartozó kovarianciamátrixhoz szükséges ismerni az előző állapot $\bar{\bar{P}}_{k-1|k-1}$ kovarianciamátrixát, az $\bar{\bar{F}}_k$ mátrixot, és a mérések bizonytalanságából megkapható $\bar{\bar{Q}}_k$ kovariancia, avagy zajmátrixot.

$$\bar{\bar{Q}}_k = \left[\bar{\bar{G}}_k \cdot \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \right] \cdot \left[\bar{\bar{G}}_k \cdot \begin{bmatrix} \sigma_x & 0 \\ 0 & \sigma_y \end{bmatrix} \right]^T$$

ahol σ_x, σ_y a nyugalmi állapotban és a globális rendszerben lévő gyorsulásmérő szórása, vagyis azok a mérések átlagtól való átlagos eltérése, amelyeknél még nem történt elmozdulás:

$$\sigma_x = \frac{\sum |\bar{a}_x|}{n}$$

$$\sigma_y = \frac{\sum |\bar{a}_y|}{n}$$

ahol n a mérések száma. Fontos ügyelni arra, hogy a fentebb használt $\bar{a}_x, \bar{a}_y$ és n változók csak arra a mérés időtartamában értelmezendők, amelyeknél még mozdulatlan volt a telefon.

A $\bar{Q}_k$ ismeretében most már felírható a $\hat{P}_{k|k-1}$ mátrix:

$$\hat{P}_{k|k-1} = \bar{F}_k \cdot \bar{P}_{k-1|k-1} \cdot \bar{F}_k^T + \bar{Q}_k$$

3.5.3 A frissítés

A frissítés során megvizsgáljuk az adott időpillanatban számított méréseket, ezeket összevetjük a jósolt értékekkel, majd a $\bar{K}_k$ Kálmán mátrix felírásával, megfelelő súlyozással megjavítjuk az állapotvektort és a kovarianciamátrixát. Ezt követően utóellenőrzést végzünk, majd átlépünk a következő időpillanatba.

A Kálmán mátrix meghatározása

A korábban már említett $\bar{H}_k$ állapot-mérés mátrixot most felhasználva különbségvektort számítunk ki a mért GPS és a jósolt koordináták közt a következő módon:

$$\tilde{y}_k = z_k - \bar{H}_k \cdot \hat{x}_{k|k-1}$$

ahol z_k a GPS méréseink oszlopvektorba rendezve, méterbe átszámítva.

A $\bar{K}_k$ Kálmán mátrix felírásához egy $\bar{S}_k$ segédmátrix szükséges, amely a kovarianciák javítását avagy módosítását definiálja. Kiszámításához ismernünk kell az $\bar{R}_k$ mátrixot, ami a középhibák várható értékeit, vagyis a mérések szórását tartalmazza. Esetünkben a GPS pontosságát, vagyis a 4.9 métert használtam a fő átlóban.

$$\bar{R}_k = \begin{bmatrix} \mu_{GPS}^2 & 0 \\ 0 & \mu_{GPS}^2 \end{bmatrix}$$

$$\bar{\bar{S}}_k = (\bar{\bar{H}}_k \cdot \hat{P}_{k|k-1} \cdot \bar{\bar{H}}_k)^T + \bar{\bar{R}}_k$$

Azt, hogy a mérések milyen súlyokkal befolyásolják az egyes állapotokat, a $\bar{\bar{K}}_k$ Kálmán mátrix adja meg. Kiszámolásához szükséges a segédmátrix inverze, azonban ezt a program kiszámolja helyettünk az *inv()* parancs segítségével.

$$\bar{\bar{K}}_k = \hat{P}_{k|k-1} \cdot \bar{\bar{H}}_k^T \cdot \bar{\bar{S}}_k^{-1}$$

Az új állapot

Az utolsó lépés meghatározni az új állapotot és a kovarianciáját. Ehhez a következő képlet nyújt segítséget:

$$\bar{\bar{x}}_{k|k} = \hat{x}_{k|k-1} + \bar{\bar{K}}_k \cdot \tilde{y}_k$$

$$\bar{\bar{P}}_{k|k} = (\bar{I} - \bar{\bar{K}}_k \cdot \bar{\bar{H}}_k) \cdot \hat{P}_{k|k-1}$$

ahol $\bar{I}$ esetünkben egy 4 oszlopból és sorból álló egységmátrix.

Az újonnan kapott állapotot visszaellenőrizhetjük, hogy mennyiben térünk el attól az állapottól, amit csak az adott pillanatra értelmezett GPS mérésekből számítottunk volna. Ezt az eltérést a további számításokban azonban nem vesszük figyelembe.

$$\tilde{y}_{k|k} = z_k - \bar{\bar{H}}_k \cdot \hat{x}_{k|k}$$

4. EREDMÉNYEK

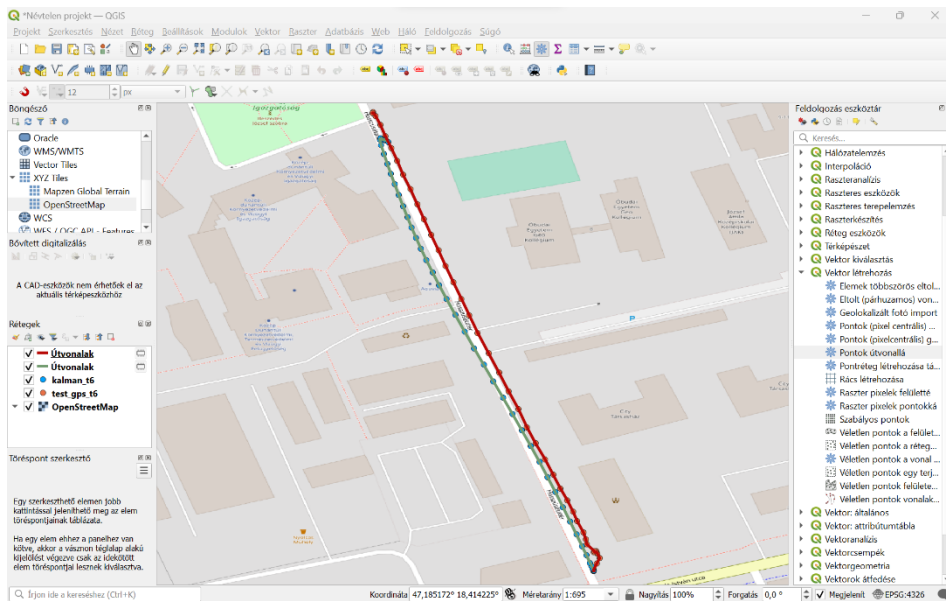
4.1 A valósághű mozgásábra

Ezt a programkódot az utcában végzett azonos tulajdonságú mérések eredményével is feltöltöttem. A következő 3)-as ábrán szemléltetem a különbséget a csak GPS mérés, illetve a Kálmán szűrő által meghatározott bejárt út közt. A MatLab-ban számított állapotvektorokban szereplő koordinátákat a már ismerttetett földrajzi szélesség és hosszúságból északias-keleties koordinátákká átalakító függvény inverzével

viSSZaváltottam gömbi koordinátákká, melyet a QGIS nevű programban jelenítettem meg.

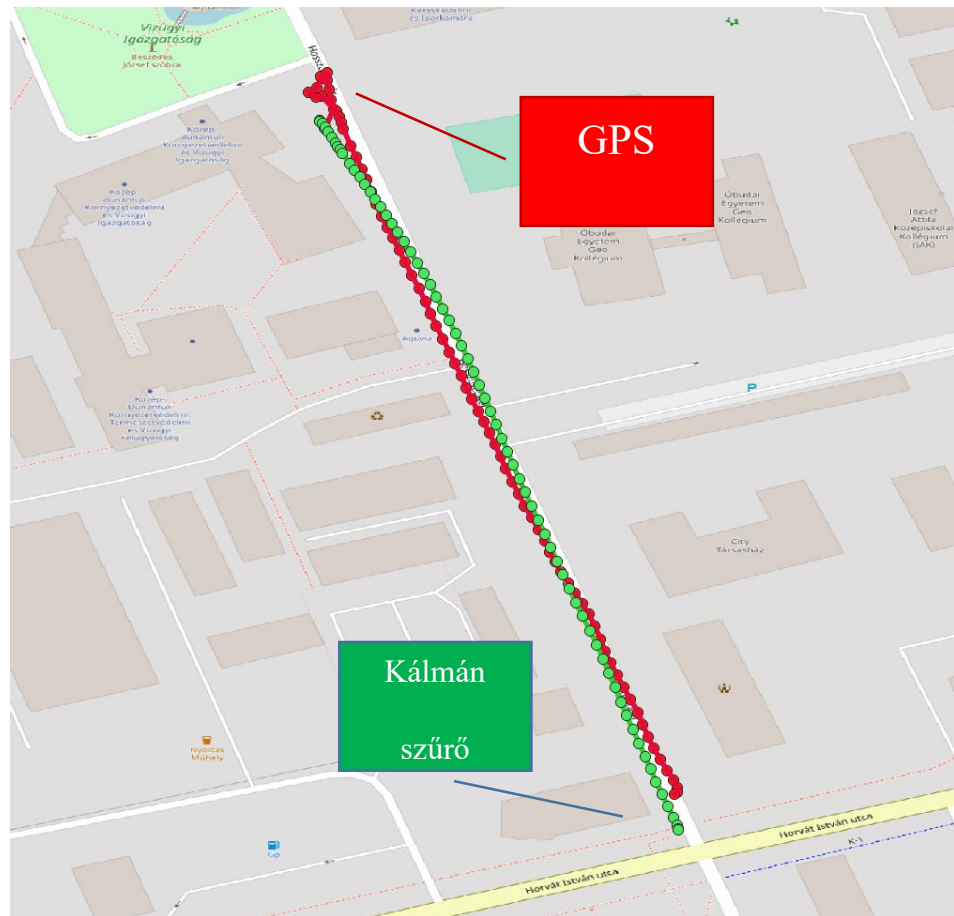
$$GPS_{szélesség} = \frac{\acute{E}_{méter}}{111000}$$

$$GPS_{hosszúság} = \frac{K_{méter}}{111000 \cdot \cos(GPS_{szélesség})}$$



A QGIS program felhasználói felülete

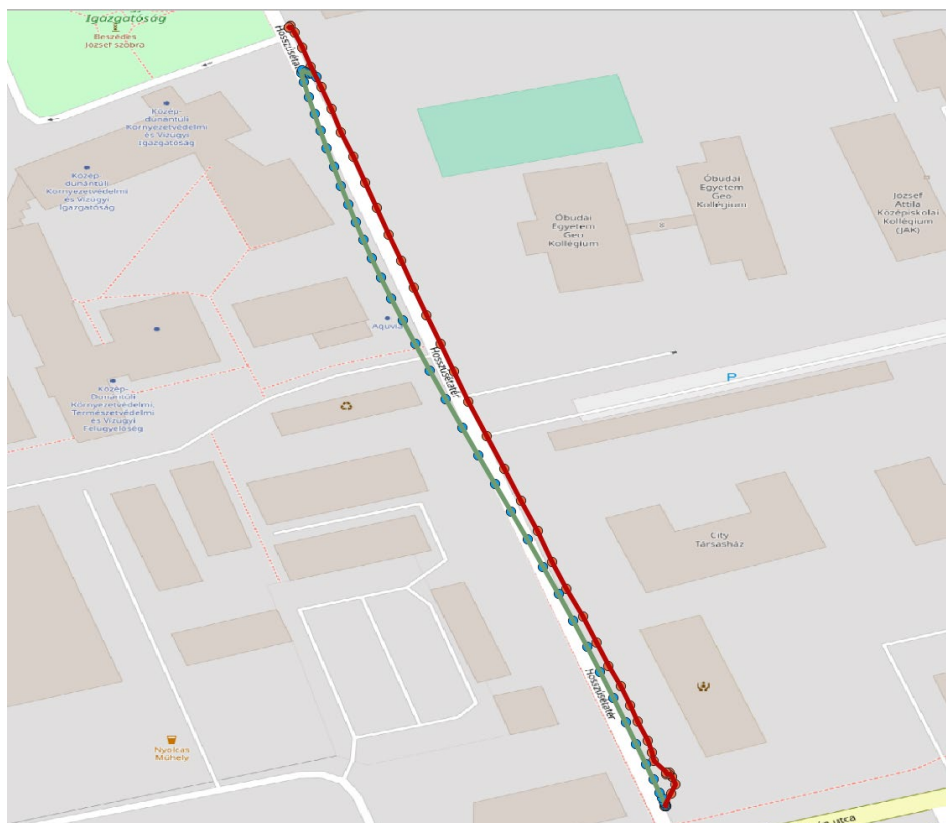
Az OpenStreetmap nevű nyílt térképet használtam alaprétegnek, melyre a GPS által mért, és a programkódom által számított koordinátákat helyeztem el. Vetületnek a WGS84-et választottam, mely a földrajzi szélességeket és hosszúságokat képes ábrázolni.



3) Az Openstreetmap-ben ábrázolt mért és számított bejárt utak

Látszik, hogy a valóban megtett út útvonalát nem követi a mozgásomat modellező zöld egyenes.

Visszafele is végeztem mérést az utcában, és az egyik közülük meglepően jó eredményt adott. Itt a zöld vonalnak megfelelően valóban az út jobb oldalán haladtam, a kollégium előtt elhaladva pedig beljebb tértem az út közepe fele, majd az út szélén álltam meg. A két számítást ugyanazon programkóddal végeztem, így feltételezem, hogy a mérési körülmények nagyban befolyásolják a végeredményt. A valóban bejárt út ismeretében azt a következtetést is levontam, hogy a haladási irányra merőlegesen elkezdett növekedni a hiba, majd korrigálódott. Menetiránynak megfelelően 'lemaradás' látszik, ezt a bias nem elég precíz kiszűrésének tulajdonítom. A kezdeti GPS mérés koordinátáinak tökéletesnek vétele is befolyásolja a pontatlanságot.

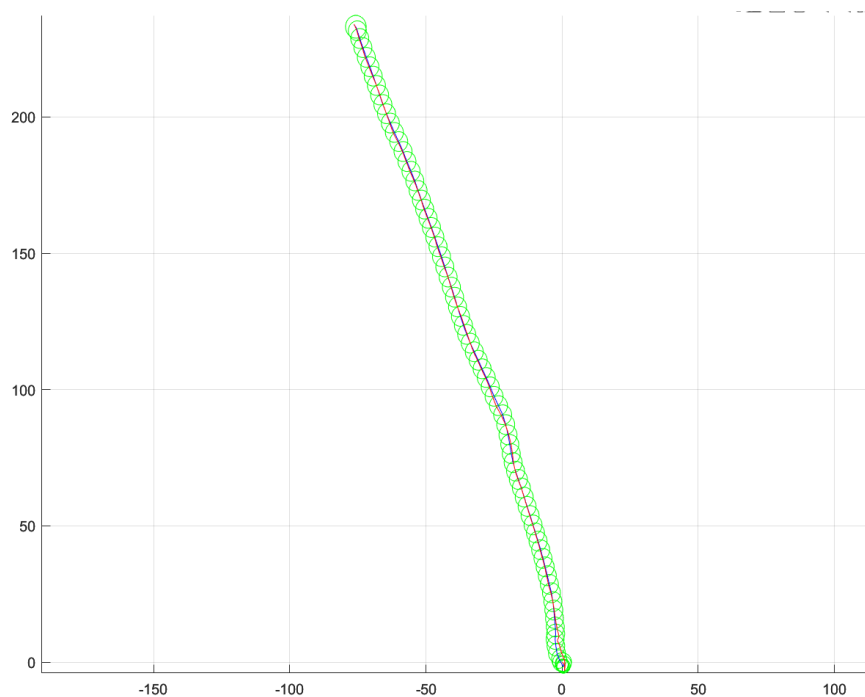


4) Pontosabb számítás ideálisabb mérés esetén. Amely például a telefon vízszintest jobban megközelítő helyzetének köszönhető. (Zöld – Kálmán szűrő, Piros – GPS)

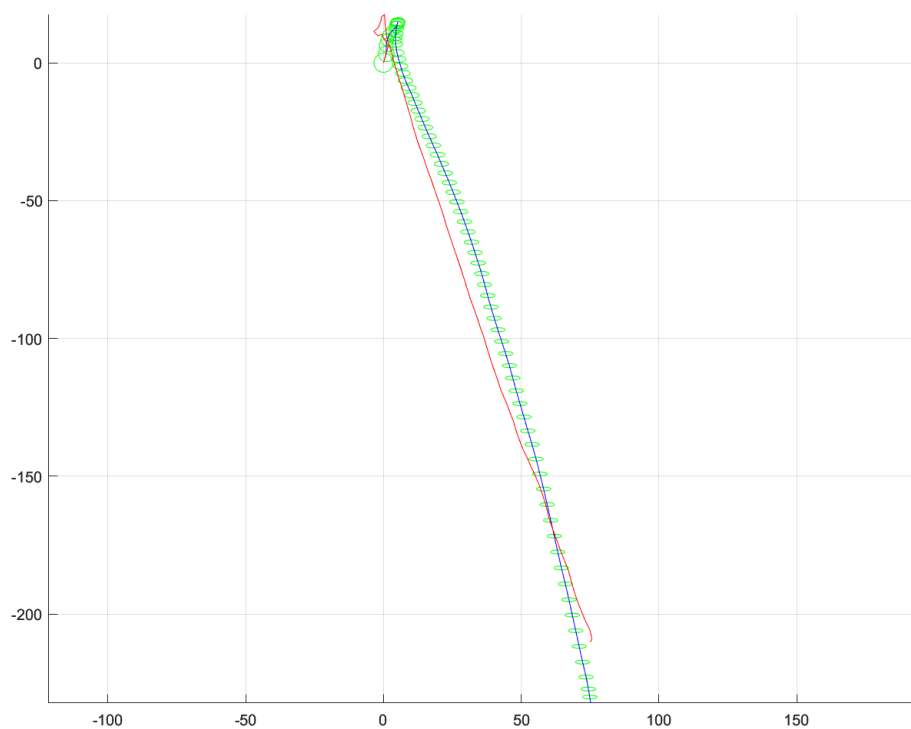
4.2 A hibaellipszis

Az x állapotvektor P kovarianciamátrixát ismerve kifejezhető egy L mátrix, amellyel az adott pillanat Kálmán szűrővel meghatározott koordinátákra illesztett kör az adott számítás hibaellipszisévé torzítható.

Biztosan állíthatjuk, hogy a valódi helyzetünk ezen hibaellipszisen belül helyezkedett el. Látható, hogy számításunk egyre kevésbé megbízható, az ellipszis által lefedett terület egyre nagyobb. A hibaellipszisek az 5) és 6) ábrákon zölddel láthatóak, a korábban bemutatott Kálmán szűrő által számított bejárt úton.



5) Láthatóan bizonytalanabb, pontatlanabb mérés eredményeként a hibaellipszisek nagyok, a valódi bejárt út nagyobb területen helyezkedhet el.



6) Egy láthatóan pontosabb mozgásmodell hibaellipszisei is számottevően kisebbek.

5. DISZKUSSZIÓ

5.1 Hasonló munkák

Napjainkban számos megoldás születik a helymeghatározás egyre pontosabb és hatékonyabb megoldására. [9] Ezen megoldások azonban gyakran valamilyen ideális rendszerben, környezetben képesek csak működni, ezért tartom kiemelkedőnek az alkalmazkodni képes mobiltelefonba épített szenzorok „rendszerét”, amivel én is dolgoztam.

A Kálmán szűrő, habár egy rendkívül hasznos módszer a szenzorok pontosságának javítására, amennyiben nem veszünk figyelembe minden mérést és számítást terhelő hibát, addig csekély javulás vehetünk észre az egyszerű egy-szenzoros mérésekhez képest.

Személyes véleményem szerint a dolgozatom kutatási témáját nem tekintem lezártnak. Ahogy azt már korábban említettem, a Kálmán szűrő tovább bővíthető határok nélkül, ezzel pontosabb, és akár más célra is felhasználható eredményeket elérve. Én személy szerint izgalmasnak tartom ezt a tudományterületet és remélem, tudok a jövőben is ezzel foglalkozni: a bemutatott egyszerűnek nevezhető modellt komplexebbé tenni, későbbi kutatásomban valamennyi mérési hibát felismerni majd kezelni. Továbbá vizsgálnám összetett mozgások modellezésének pontosítását más módszerek alkalmazásával. Ezen túl célom a helymeghatározás és a mozgásmodellezésben elért tanulmányokban mélyíteni tudásomat, a Kálmán szűrő, vagy Kálmán Rudolf egyéb elméleteinek hasznosságát más tudományterben is felfedezni, felhasználni.

Minden nap több órát használjuk okostelefonjainkat és annyi mindenre képesek lennénk használni ezt a kisméretű számítógépet. Remélem dolgozatommal felkeltettem a figyelmét arra, hogy használjuk ki eszközeink felfoghatatlan képességeit.

6. KÖSZÖNETNYILVÁNÍTÁS

Dolgozatom elkészítéséhez sok támogatást kaptam családtagjaimtól, közeli ismerőseimtől, egyetemi szaktársaimtól és tanáraitól. Külön köszönet illeti: Busics György, Lengyel Ágoston, Molnár Gábor Péter, Dr. Nagy Gábor, Pusztai Henrik Antal, Szűcs Napsugár Veronika.

A Kálmán szűrőről és a szenzor fúzióról legtöbbet a MathWorks weboldalon, és az interneten megtalálható ingyenes videósorozatukból tanultam, ahol ezekben a tágabb témákban is elmélyedhettem. [10] [11]

7. IDÉZETT FORRÁSMUNKÁK

- [1] T. Simon, Magyarország és a világűr - Szerepünk a világ űrtevékenységében, MCC Press Kft., 2021.
- [2] „ELTE Informatikai Kar, Térképtudományi és Geoinformatikai Intézet honlapja,” [Online]. Available: <http://lazarus.elte.hu/~climbela/gps3.htm>. [Hozzáférés dátuma: 9 október 2022].
- [3] F. v. D.-P. Enge, "The World's first GPS MOOC and Worldwide Laboratory using Smartphones," in *28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015)*, Tampa Convention Center, Florida, 2015.
- [4] G. Paresh, "Techulator," 25 március 2020. [Online]. Available: <https://www.techulator.com/resources/8930-How-does-smart-phone-accelerometer-work.aspx>. [Accessed 20 július 2022].
- [5] E. S. C. Team, "NASA/Global Climate Change," 23 augusztus 2021. [Online]. Available: https://climate.nasa.gov/internal_resources/2414/. [Accessed 10 október 2022].
- [6] „Számítástechnikai és Automatizálási Kutatóintézet (SZTAKI),” 3 május 2018. [Online]. Available: <https://www.sztaki.hu/tudomany/hirek/kovetkezo-felevtol-elinditjuk-rudolf-e-kalman-distinguished-lecturer-program-ot>.
- [7] A. Becker, "Kalman Filter," 2022. [Online]. Available: <https://www.kalmanfilter.net/kalmanmulti.html>.

- [8] B. K. P. Horn, "Closed-form solution of absolute orientation using unit," Department of Electrical Engineering, University of Hawaii at Manoa, Honolulu, Hawaii, 1986.
- [9] L.-G. Mónika, „National Geographic - Tudomány - Helyzetmeghatározó, műhold nélkül,” 26 November 2022. [Online]. Available:
<https://ng.24.hu/tudomany/2022/11/26/helyzetmeghatarozo-muhold-nelkul/>.
- [10] MATLAB®&Simulink®, „MathWorks - Understanding Kalman Filters,” Simulink®, [Online]. Available:
https://www.mathworks.com/videos/series/understanding-kalman-filters.html?s_eid=PSM_15028.
- [11] MathWorks®, „MathWorks - Estimating Orientation Using Inertial Sensor Fusion and MPU-9250,” [Online]. Available:
https://www.mathworks.com/help/fusion/ug/Estimating-Orientation-Using-Inertial-Sensor-Fusion-and-MPU-9250.html?s_eid=PSM_15028.