



Budapesti Műszaki és Gazdaságtudományi Egyetem

Tudományos Diákköri Konferencia

Geodézia és térinformatika szekció

2014

Ciklustöbbértelműség értékek és ionoszférikus késleltetés becslése abszolút GNSS mérések alapján

KÉSZÍTETTE

Ambrus Bence

Földmérő- és térinformatikai mérnök MSc hallgató

Budapesti Műszaki és Gazdaságtudományi Egyetem

(I. évfolyam)

KONZULENS

dr. Takács Bence

Egyetemi docens

Budapesti Műszaki és Gazdaságtudományi Egyetem

Általános- és Felsőgeodézia Tanszék

Bevezetés	1
1. A GNSS mérések egyenletei	2
2. A Kálmán-szűrő és működése	3
2.1 A szűrési eljárás ismertetése.....	4
2.2 A Kálmán-szűrő alkalmazása GNSS mérésekre	5
3. Mérések Kálmán-szűrővel történő feldolgozása	6
3.1 Az észlelési és navigációs RINEX fájlok feldolgozása	7
3.2 A mérésekhez tartozó magassági szög, azimut és Klobuchar féle ionoszféra késleltetés számítása	8
3.3 Kálmán-szűrő alkalmazása a mérésekre.....	10
3.4 Eredmények és értékelés.....	11
4. Összefoglalás, további lehetőségek	15
5. Matlab Programkódok	16
5.1 Az észlelési és navigációs RINEX fájlok feldolgozása	16
5.2 A mérésekhez tartozó magassági szög, azimut és Klobuchar féle ionoszféra késleltetés számítása	20
5.3 Kálmán-szűrő alkalmazása a mérésekre.....	23
6. Felhasznált irodalom	25

ÁBRAJEGYZÉK

3.1. ábra: N1 és N2 értékei. Vastag színes vonallal a szűrt eredmények, fekete vonallal a nyers eredmények látható.....	11
3.2. ábra: A nyers és szűrt p' értékek mindkét műholdnál.....	12
3.3. ábra: Az ionoszféra késleltetések és a magassági szög értékei a GALILEO-19 műhold esetében.	13
3.4. ábra: Az ionoszféra késleltetések és a magassági szög értékei a GPS-01 műhold esetében	13
3.5. ábra: Variancia-értékek alakulása a szűrés folyamán (GPS-01). Az ábrák mellett a maximum és a minimum van feltüntetve.....	14
3.6. ábra: Variancia-értékek a GALILEO-19 esetében. Az ábrák mellett a maximum és a minimum érték van feltüntetve.....	14

ABSZTRAKT

Amíg a navigációs célú pozicionálást a Globális Navigációs Műholdrendszerek (GNSS) műholdjai által sugárzott jelek kód-méréssel történő feldolgozása is lehetővé teszi, addig a geodéziai igényű helymeghatározási pontosság eléréséhez a fázismérések használata szükséges. A fázismérések feldolgozása során felmerülő komoly probléma, hogy nem ismerjük a mérések kezdeti időpontjában a műhold-vevő távolságot lefedő egész ciklusok számát, ezáltal nem tudjuk megfelelő pontossággal elvégezni a helymeghatározást.

Annak érdekében, hogy méréseinkből a valóságnak megfelelő és a geodéziai pontossági igényeket kielégítő pozíciókat kaphassunk, szükséges, hogy a kezdeti ciklusok számát egész számként rögzítsük, vagyis feloldjuk a ciklustöbbsértelműséget. Ennek a viszonylag számítás-igényes folyamatnak az időigénye csökkenthető, amennyiben megfelelő kezdőértéket tudunk használni, tehát minél jobb az előzetes becslésünk. A kezdőérték becslésének egy módja a kód-mérési adatok feldolgozása. Mivel azonban az abszolút helymeghatározás esetén a kód-méréseket terhelő zaj igencsak nagymértékű, így célszerű valamilyen szűrési eljárás alkalmazása az adatok feldolgozása során.

Dolgozatomban GPS és GALILEO műholdak adatait felhasználva, Kálmán-szűrő segítségével végeztük el a kód- és fázismérések feldolgozását, amelyek során vizsgálatunk fő tárgya az volt, hogy milyen pontossággal becsülhetők meg a ciklustöbbsértelműség értékek. A mérések feldolgozása során további eredményként az ionoszféra által a jelterjedésre gyakorolt késleltető hatás mértékét is megállapítottuk, valamint az így kapott értékeket összevetettük a Klobuchar-féle ionoszféra modellből számítható értékekkel.

BEVEZETÉS

A modern világ hétköznapijai által támasztott polgári célú navigációs és helymeghatározási igények leghatékonyabb kielégítési módja a Globális Navigációs Műholdrendszerek (GNSS - *Global Navigation Satellite System*) használata. A jelenleg teljesen kiépített státuszban lévő amerikai GPS-rendszer, valamint az orosz fejlesztésű GLONASS-rendszer mellett több országnak/régióknak is folyamatban van a saját maga által kialakított és karbantartott rendszer létrehozása (pl. Európa - Galileo, Kína - Compass/BeiDou).

A navigációs pontossági igényű (méteres - többméteres) helymeghatározás mellett azonban fellép a geodéziai igényű pontmeghatározás szükségessége is, ami legalább cm-es, de inkább mm-es helymeghatározási pontosságot követel meg (különleges esetekben mm alatti, ld. kéregmozgás vizsgálatok). A GNSS műholdak által sugárzott adatokat felhasználva, csakis az eredeti vivőhullám visszaállításával és feldolgozásával érhető el ilyen pontosság. Ezen fázismérések esetében azonban követelmény, hogy ismerjük a vevő bekapcsolásakor (továbbá az esetleges jelvesztések után történő újracsatlakozáskor, vagyis ciklusugráskor) a műhold-vevő távolságot lefedő egész hullámok számát, vagyis feloldjuk a ciklustöbbsértelműséget. Ennek elvégzésére manapság több technika is ismert, melyek közös ismertetőjele, hogy valamilyen előzetes becsléssel kapott értékből indulnak ki és ezekből jutnak el a többsértelműség megoldására, amely egy műholdakként (és ciklusugrásokként) egy egész szám. Elképzelhető tehát, hogy a végleges eredményhez minél közelebb álló kiindulási értéket adunk meg, a számítás teljesítmény- és időigénye annál jobban lecsökken. Ilyen kezdőérték számítható a kód- és fázismérések előzetes feldolgozásából. Mivel azonban a mérési zaj meglehetősen nagymértékű ezek esetében, így célszerű lehet valamilyen szűrési eljárás, mint pl. a Kálmán-szűrő alkalmazása. Az így kapott eredményekből várhatóan a ciklustöbbsértelműség egész értékének (fix megoldás) pontosabb közelítése állítható elő.

A továbbiakban a GNSS méréseknél alkalmazott mérési egyenletek általános alakját, valamint a Kálmán-szűrőt és annak a méréseken történő alkalmazását ismertetjük. Ezután kitérünk az így kapott eredmények értékelésére és a konklúziók levonására.

1. A GNSS MÉRÉSEK EGYENLETEI

A GNSS mérések alapvető típusai a kódérések, valamint a fázismérések. A kódérések mérési egyenleteinek, a P_i^k pszeudotávolságok és a fázismérések mérési L_i^k egyenleteinek egy általános felírása i vevő és k műhold között az alábbiak szerint tehető meg [Schaer, 1999]:

$$P_i^k = \rho_i^k + c(\Delta t_i - \Delta t^k) + \Delta \rho_{i,trop}^k + \Delta \rho_{i,ion}^k + c(b^k + b_i) + \varepsilon \quad (1.1)$$

$$L_i^k = \rho_i^k + c(\Delta t_i - \Delta t^k) + \Delta \rho_{i,trop}^k - \Delta \rho_{i,ion}^k + \lambda B_i^k + \varepsilon \quad (1.2)$$

ahol a jelölések a következők

$\Delta t_i, \Delta t^k$ a vevő, illetve a műhold órája által mutatott idő eltérése a GPS időrendszerben mért időtől,

$\Delta \rho_{i,trop}^k$ a troposzféra által okozott jelkésleltetés hatása,

$\Delta \rho_{i,ion}^k$ az ionoszféra által okozott jelkésleltetés hatása,

b^k, b_i a műholdnál és a vevőnél fellépő hardverkésés

λ a vivőhullám hullámhossza

B_i^k konstans eltérés ciklusegységben kifejezve, mely tartalmazza a kezdeti N_i^k ciklustöbbséget és a hardverkéséseket

ε a mérések maradék ellentmondásai.

Általános esetben a P_i^k konstans eltérés értékét minden egyes műholdra, vevőre és vivőfrekvenciára meg kell állapítani. Amennyiben a fázisméréseink nem folyamatosak, szakadások lépnek fel bennük (pl. a műhold kitakaró objektum mögé kerül, majd ismét előtűnik), úgy minden egyes ciklusugrásra egy új eltérés paraméter bevezetése szükséges. [Schaer, 1999]

A továbbiakban nem az (1.1) és (1.2) egyenleteket fogjuk felhasználni, hanem azok egyszerűsített alakjait, amelyek a következőképpen adhatók meg. Ezekben az egyenletekben a maradék ellentmondásokat elhanyagoljuk [Schaer, 1999]:

$$P_{i,1}^k = \rho_i'^k + I_i^k + c(b^{k,1} + b_{i,1}) \quad (1.3)$$

$$L_{i,1}^k = \rho_i'^k - I_i^k + \lambda_1 B_{i,1}^k$$

$$P_{i,2}^k = \rho_i'^k + \xi I_i^k + c(b^{k,2} + b_{i,2}) \quad (1.4)$$

$$L_{i,2}^k = \rho_i'^k - \xi I_i^k + \lambda_2 B_{i,2}^k$$

ahol

$I_{i,1}^k, I_{i,2}^k$	mindkét frekvencián végzett fázismérések eredményei,
$P_{i,1}^k, P_{i,2}^k$	mindkét frekvencián végzett kódérések eredményei,
ρ_i^k	a órahibákat és a troposzféra késleltető hatását tartalmazó geometriai távolság,
$\xi = \frac{\nu_1^2}{\nu_2^2}$	a vivőjelek frekvenciáinak egymáshoz képesti aránya,
I_i^k	az ionoszféra késleltető hatása az L1 frekvencián,
$B_{i,1}^k, B_{i,2}^k$	mindkét frekvenciához tartozó eltolás paraméter (tartalmazza a ciklustöbbsértékelést és a hardverkésés értékeket).

Az általunk végzett számítások során a (1.3) és (1.4) egyenletekben nem vesszük figyelembe a vevőnél és a műholdnál fellépő hardverkéséseket, ugyanis a BUTE permanens állomásra (amelyre a mérési adataink vonatkoznak) nem állt rendelkezésre ilyen adat. Ennek megfelelően az egyenletekben lévő $B_{i,1}^k, B_{i,2}^k$, paraméterek helyett $N_{i,1}^k, N_{i,2}^k$ ciklustöbbsértékelés értékeit használunk, valamint a $c(b^{k,1} + b_{i,1})$ és $c(b^{k,2} + b_{i,2})$ értékeket elhanyagoljuk.

2. A KÁLMÁN-SZŰRŐ ÉS MŰKÖDÉSE

Méréseink feldolgozása során gyakran előfordul olyan eset, hogy egy ismeretlenhalmaz elemeit az ismeretlenek számánál jóval több mérésből szeretnék meghatározni. Ebben az esetben általában kiegyenlítést szoktunk végezni, hogy a meghatározni kívánt paraméterek legoptimálisabb becslését kaphassuk. Abban az esetben, ha méréseink folyamatosan érkeznek (pl. 1 mp-es frekvenciával), felmerülhet az igény arra, hogy a kiegyenlítést ne az összes mérést felhasználva, mintegy egyszerre, a mérések befejezése után végezzük el, hanem valamilyen módon egy kezdeti becsléserőrtet korrigáljunk a beérkező méréseink által szolgáltatott információ felhasználásával. Ennek megoldására a Kálmán-szűrő lehetőséget nyújt.

A szűrő által megadott algoritmus általános esetben valamilyen változó rendszer t_i időpontbeli állapotáról ad becslést, miközben figyelembe veszi az előző $t_0, \dots, t_{i-1}$ állapotbeli méréseket, valamint a méréseket terhelő zajokat és bizonytalanságokat. Maga a szűrési eljárás két lépésben működik: először a rendszer állapotegyenletét felhasználva becslést végzünk az ismeretlenek új, t_i időpontbeli értékeire, felhasználva az ismeretlenek t_{i-1} időpontbeli állapotát (ez

viszont már tartalmazza az eddigi összes $t_0, \dots, t_{i-1}$ időpontbeli állapotok méréseit), valamint megbecsüljük az ismeretlenekhez tartozó t_i időpontbeli kovariancia mátrix értékeit is. A szűrés második lépésében az i időpontbeli mérés és a Kálmán-féle nyereségmátrix felhasználásával korrigáljuk a becslésünket, ezáltal megkapva a t_i időpontra vonatkozó ismeretlenek és kovarianciájuk értékét. Mind a becslést, mind a korrekciót rekurzív módon számítjuk, csakis az előző időpontbeli ismeretlen és kovariancia értékeket, valamint az adott időpontbeli méréseket vesszük figyelembe. Ennek értelmében nem szükséges eltárolni az összes beérkezett mérést és számított eredményt, hanem mintegy görgetve a számítást, mindig a legoptimálisabb becslést kaphatjuk. A szűrő ezen tulajdonsága miatt gyorsan változó, dinamikus rendszerek valósídejű (on-the-fly) feldolgozása is hatékonyan elvégezhető.

Az eljárást E. Rudolf Kálmán, magyar származású amerikai villamosmérnök dolgozta ki. Mára az algoritmusnak számos változata létezik, mint pl. a bővített Kálmán-szűrő, a Kálmán-Bucy módszer, a minimális szórásnégyzetű modell, a hibrid Kálmán-szűrő stb., melyeket a mérnöki gyakorlat és mérésfeldolgozás rengeteg területén alkalmaznak.

2.1 A SZŰRÉSI ELJÁRÁS ISMERTETÉSE

Az egyenletek felírásánál több szerző többféle jelölése is elterjedt, mi a *Gilbert Strang & Kai Borre – Linear algebra, Geodesy and GPS* című könyv jelöléseit fogjuk a továbbiakban alkalmazni.

Amennyiben rendelkezésünkre áll az ismeretlen paraméterek valamilyen előzetes becslése $k-1$ időpontban, továbbá a hozzájuk tartozó $P_{k-1|k-1}$ kovariancia mátrix $k-1$ időpontra vonatkozó értéke, valamint rendelkezünk a mérések hibáit jellemző Σ_e és a rendszer állapotváltozásához tartozó hibákat jellemző Σ_ϵ kovariancia mátrixokkal, akkor a szűrés a következő lépéseket végrehajtva végezhető el [Strang & Borre, 1998]:

1. Kiszámítjuk az ismeretlenek értékeinek, valamint a hozzájuk tartozó kovariancia mátrix becslését, az $\hat{x}_{k|k-1}$ vektort és a $P_{k|k-1}$ mátrixot:

$$\hat{x}_{k|k-1} = F_{k-1} x_{k-1|k-1} \quad (2.1)$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T \quad (2.2)$$

ahol F_{k-1} a $k-1$ időpillanatban bekövetkező állapotváltozást leíró mátrix.

2. Meghatározzuk a Kálmán-féle nyereségmátrixot, a becsült kovariancia mátrixot és a rendszert jellemző A_k együttthatómátrixot (alakmátrixot) felhasználva:

$$K_k = P_{k|k-1} A_k^T (A_k P_{k|k-1} A_k^T + \Sigma_{e,k})^{-1} \quad (2.3)$$

3. A nyereségmátrixot felhasználva számítjuk az ismeretlenek és a kovariancia mátrix korrigált értékét:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (b_k - A_k \hat{x}_{k|k-1}) \quad (2.4)$$

$$P_{k|k} = (I - K_k A_k) P_{k|k-1} \quad (2.5)$$

ahol b_k mérések vektora, I pedig az egységmátrixot jelöli.

Érdemes észrevenni, hogy a (2.4) egyenlet zárójelben lévő tagja tulajdonképpen nem más, mint k -adik időpontbeli mérési eredmények és a $k-1$ -edik időponthoz tartozó ismeretlenekkel számolt becsült mérései eredmények közötti eltérés. A nyereségmátrix ezt az eltérést súlyozza és a korrekció során ezzel az értékkel módosul a becsült paraméterek vektora.

2.2 A KÁLMÁN-SZŰRŐ ALKALMAZÁSA GNSS MÉRÉSEKRE

Mivel az (1.3) és (1.4) egyenletrendszerek baloldalán lévő pszeudotávolság és fázismérés eredményeket meglehetősen nagy mérési zaj terheli, ezért célszerűnek tűnhet ezeknek a méréseknek a Kálmán-szűrővel történő feldolgozása. Az egyenletek mátrixos alakja egy adott vevő esetén, egyetlen epochában a következőképpen írható fel:

$$\begin{bmatrix} P_1 \\ L_1 \\ P_2 \\ L_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & \lambda_1 & 0 \\ 1 & \xi & 0 & 0 \\ 1 & -\xi & 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \rho' \\ I \\ N_1 \\ N_2 \end{bmatrix} + \text{mérési zaj} \quad (2.6)$$

Ahhoz hogy a számítást elkezdhessük, szükségünk van a méréseket jellemző Σ_e és az állapotváltozást jellemző Σ_ε kovariancia mátrixokra. Az ezekben lévő elemek értékeit [Strang & Borre, 1998] alapján az alábbiaknak vettük fel (az értékek méterben értendők).

$$\Sigma_e = \begin{bmatrix} 0.3^2 & & & \\ & 0.05^2 & & \\ & & 0.3^2 & \\ & & & 0.05^2 \end{bmatrix} \quad \Sigma_\varepsilon = \begin{bmatrix} 100 & & & \\ & 10 & & \\ & & 0 & \\ & & & 0 \end{bmatrix} \quad (2.7)$$

Látható, hogy az N_1 és N_2 értékek varianciája 0, értékük definíció szerint nem változik, amennyiben nem következik be ciklusugrás a mérések során.

A kezdeti $\hat{x}_0$ paraméterek becslését úgy állíthatjuk elő a legkönnyebben, ha az első epochához tartozó mérések lineáris egyenletrendszerét egyszerűen megoldjuk és az ismeretlenek így kapott értékeit tekintjük kiindulási adatoknak. A számításhoz szükséges kezdeti kovariancia mátrixot nekünk kell felvenni, ez esetünkben a

$$P_{0|0} = \begin{bmatrix} 10 & & & \\ & 10 & & \\ & & 10 & \\ & & & 10 \end{bmatrix} \quad (2.8)$$

mátrix megadását jelentette, a kezdeti mérések zajosságára számítva. A számításainkban a vizsgált rendszert statikusnak feltételezzük, tehát $\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1}$, tehát $F_k = I$. Ezeket az értékeket sorba behelyettesítve a (2.1) - (2.5) egyenletekbe, $t = 0 \dots k$ időpontokban megkapjuk az ismeretlenek és kovariancia mátrixuk becsült értékét, az időponthoz tartozó nyereségmátrixot, majd ezeket felhasználva az adott időponthoz tartozó korrigált becsléseket.

3. MÉRÉSEK KÁLMÁN-SZŰRŐVEL TÖRTÉNŐ FELDOLGOZÁSA

A Kálmán-szűrő 2.2 fejezetében ismertetett alkalmazását a Budapesti Műszaki és Gazdaságtudományi Egyetem K épületén lévő permanens állomás (BUTE) 2014. 04. 29-ére vonatkozó adatsorán hajtottuk végre, GALILEO és GPS műholdakra egyaránt. A műholdak kiválasztása során fontos szempont volt, hogy a mérések folyamatosan történjenek, ne legyenek szakadások az adatsorban, ez ugyanis szembe ment volna a feldolgozás eddig vázolt elméleti megfontolásaival.

Az adatok feldolgozását Matlab programozási környezetben végeztük el, melynek során parancssorból futtatható függvényeket és eljárásokat készítettünk az egyes feldolgozási lépésekhez. Ezek sorrendje a következő volt:

- RINEX (Receiver Independent EXchange format – vevő független adatsere formátum) v3.01 típusú észlelési fájl feldolgozása, a műholdakhoz tartozó kód- és fázismérési eredmények kinyerése a fájlból.

- Az adott mérési periódushoz tartozó navigációs fájl feldolgozása, a műholdakhoz tartozó navigációs adatok kinyerése a fájlból.
- Az egyes észlelt műholdakhoz tartozó azimut és magassági szög értékek meghatározása az észlelési epochában.
- A mérésekben okozott ionoszférikus késleltetés kiszámítása a Klobuchar-féle modell felhasználásával.
- A kiválasztott műholdak adatsorának feldolgozása Kálmán-szűréssel.
- Eredmények grafikus és numerikus megjelenítése.

A továbbiakban ezen lépések ismertetésére térünk, bemutatva a lépések végrehajtásához írt Matlab kódrészleteket is.

3.1 AZ ÉSZLELÉSI ÉS NAVIGÁCIÓS RINEX FÁJLOK FELDOLGOZÁSA

Ebben a fejezetben a BUTE permanens állomás 2014. 04. 29-én végzett észleléseit és a napra vonatkozó navigációs adatokat tartalmazó RINEX fájl Matlabbal történő feldolgozását ismertetjük. A feldolgozás célja, hogy az ASCII formátumú RINEX-ből egy olyan Matlab változót tartalmazó .mat kiterjesztésű fájlt hozzunk létre, amely táblázatszerűen tartalmazza a mérések tárgyául szolgáló műhold rendszerét és számát, a mérési epochát és az egyes mérési eredményeket. A navigációs fájlból szintúgy egy .mat kiterjesztésű eredményfájlt hozunk létre, amely az egyes műholdakhoz tartozó navigációs üzenetek adatait tartalmazza a vonatkoztatási epochával együtt.

1. Legelső lépésként a bementi és a kimeneti fájl megadása szükséges, valamint azon helymeghatározó rendszer kódja, amely méréseit szeretnénk a RINEX fájlból kinyerni.
2. Ezután a program először a bemeneti fájl fejlécét (header) olvassa be, ahonnan egyrészt elmenti az álláspont közelítő koordinátáit, valamint, hogy az általunk megadott rendszerhez az adott fájlban milyen észlelések tartoznak. Az észlelési típusokat egy fejléc változóba elmenti, hogy a későbbi feldolgozás során a sorrendjük visszakereshető legyen.

3. A harmadik lépésben a fájl fejléc utáni fő részét soronként végigpásztázva, a program azonosítja az egyes epochákat jelző sorokat, kiolvassa belőlük az epochák értékét, valamint a követő sorok által tartalmazott észlelési adatokat.
4. Az észlelési adatokat vektorba elmentve, azok folyamatosan összefűzésével előállítja az eredményfájlt, mely tartalmazza az álláspont WGS84 rendszerben értendő térbeli derékszögű koordinátáit, az észleléseket tartalmazó tömböt, valamint az értelmezéshez szükséges fejlécet.

A fenti lépéseket végrehajtó Matlab kód az 5.1 fejezet első részében látható.

A navigációs fájl feldolgozása során a fejléc rész kihagyható, mivel nem szolgáltat olyan információt, amely jelen feladatunkhoz szükséges lenne. Ezután egyszerű dolgunk van, hiszen minden műhold esetén ugyanannyi sor adat tartalmazza a pályaelemeket és a korrekciókat, ezek könnyedén összefűzhetők egyetlen tömbbe, majd kimenthetők az általunk megadott .mat kiterjesztésű fájlba. A feldolgozáshoz írt Matlab kód az 5.1 fejezet második részében látható.

3.2 A MÉRÉSEKHEZ TARTOZÓ MAGASSÁGI SZÖG, AZIMUT ÉS KLOBUCHAR FÉLE IONOSZFÉRA KÉSLELTETÉS SZÁMÍTÁSA

Az előző fejezetben előállított, észleléseket tartalmazó tömb soraihoz, vagyis az egyes észlelésekhez a továbbiakban szeretnénk meghatározni a műhold adott epochára vonatkozó magassági szögét, azimutját és a Klobuchar-féle modellből számítható ionoszféra általi késleltetést. Ezek mindegyikét egy-egy Matlab függvény létrehozásával számítottuk ki, úgy, hogy a függvényeket az összes észlelésre meghívtuk.

A magassági szög és az azimut számításához a függvénynek a következő bemeneti adatokat kellett megadni:

x, y, z és φ, λ, h	az álláspont térbeli derékszögű és ellipszoidi koordinátái (WGS84 rendszerben),
$t_{\text{észlelés}}$	az észlelési epochát tartalmazó vektort, melynek első eleme a GPS-hét száma, második eleme pedig a héten belül eltelt másodpercek száma
<i>navdata</i>	a navigációs adatokat tartalmazó tömb, mely célszerűen csak az adott műholdhoz tartozó adatokat tartalmazza (ezen szűrés elvégzése előzetesen megtörténik).

A függvény a navdata változóban megadott tömbben megkeresi azt a sort, amely időben a legközelebb áll az észlelési epochához (de még megelőzi azt), majd az ahhoz tartozó navigációs adatokból kiszámolja a műhold észlelési epochára vonatkozó térbeli derékszögű koordinátáit. Ezeket átváltva topocentrikus rendszerbe végül előállítja a magassági szög és az azimut értékét. A függvényhez tartozó Matlab kód az 5.2 fejezetben látható.

A magassági szögek és a azimutok ismeretében kiszámíthatók az ionoszféra késleltetések a Klobuchar-féle modell segítségével. A modellparaméterek értékeit előzetesen egy külső ionoszféra fájlból beolvassuk, melyek napi bontásban a CODE (Center for Orbit Determination in Europe) FTP szerveréről (<ftp://ftp.unibe.ch/aiub/CODE/>) tölthetők le (amennyiben a navigációs RINEX fájl nem tartalmazza őket). Pontosabb számítások esetén mindenképp célszerű a CODE-féle ionoszféra paramétereket használni, ezek ugyanis jobb meghatározást tesznek lehetővé, mint a műholdak által sugárzott navigációs üzenetben lévő paraméterek.

Az ionoszféra késleltetését számító függvény esetében a következő argumentumokat szükséges megadni a számításhoz:

φ_0, λ_0	az álláspont ellipszoidi koordinátái (WGS84 rendszerben),
A, h	a műholdhoz tartozó azimut és magassági szög érték,
t_{nap}	a napon belül eltelt másodpercek száma,
α, β	a Klobuchar-modell paraméterei.

A számítás során csak az L1 frekvenciára vonatkozó késleltetés értékét határozzuk meg, ugyanis a szűrés eljárásból is ezt az értéket fogjuk megkapni. A függvény felépítése az alábbiakban látható:

3.3 KÁLMÁN-SZŰRŐ ALKALMAZÁSA A MÉRÉSEKRE

A mérések szűréséhez alapvetően a *3.1 fejezet* részben létrehozott, méréseket tartalmazó tömböt használjuk fel, a *2.2 fejezet*ben ismertetett módon. A program bemeneti adataként szükséges megadni a vizsgálandó műhold PRN számát, a számolás kezdő epochájának értékét és a használandó mérési típusokat. Ezek esetünkben az L1 és L2 frekvencián végzett kód- és fázismérések. Kiindulási paramétereknél bevitelre kerülnek a használt konstans értékek is, a fénysebesség, a vivőhullámok hullámhosszai, frekvenciái és ezek aránya. A számítás ezek után a következőképpen történik:

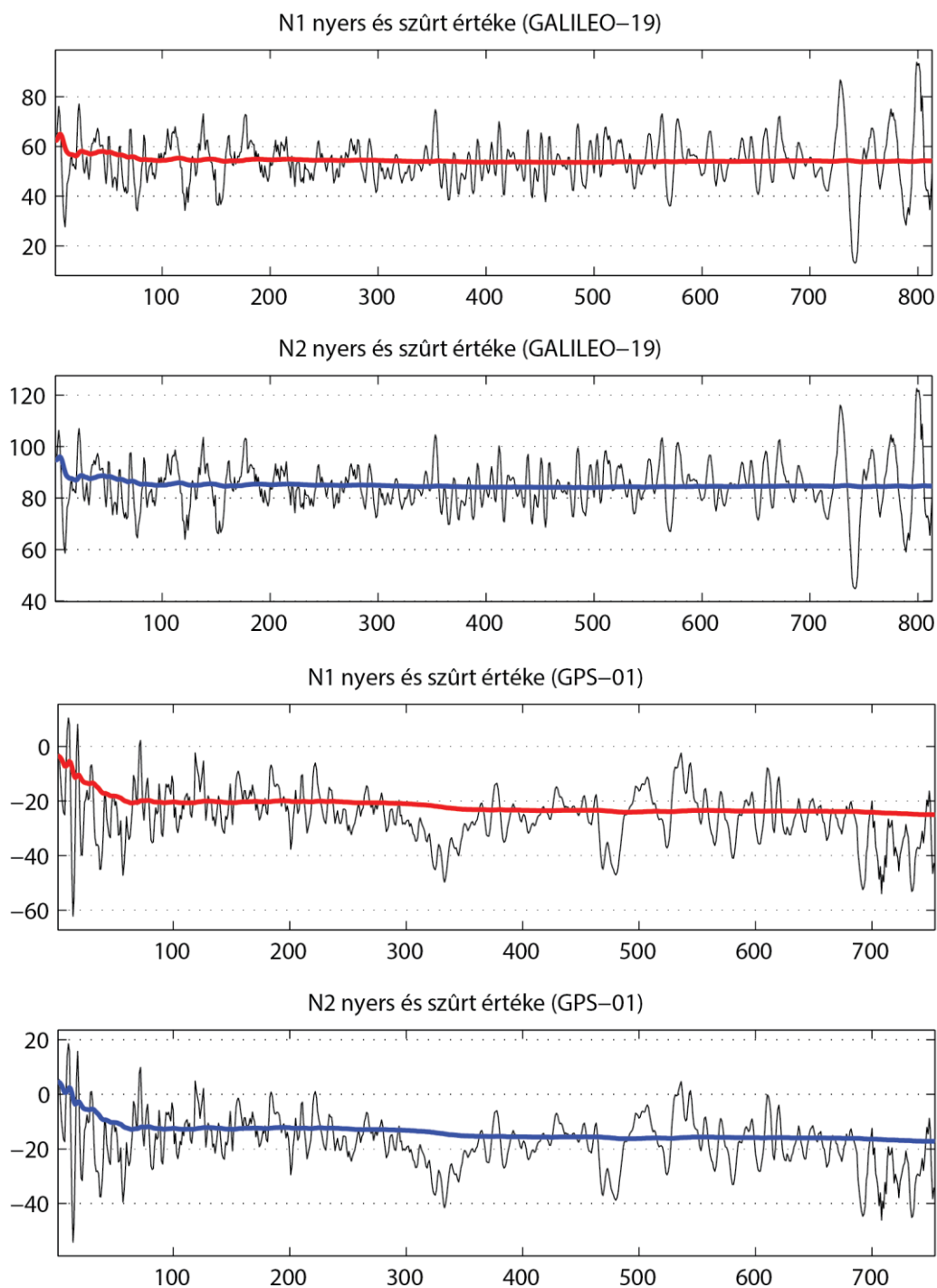
1. Az észleléseket tartalmazó Matlab változó betöltése.
2. A mérések közül a bemeneti adatoknál megadott PRN számú műhold adatainak leválogatása.
3. A mérésekből a bemeneti adatok között megadott mérési típusok kiválasztása.
4. Az észlelések kiszűrjük azokat az epochákat, ahol egyrészt vagy nem történt mérés az általunk használni kívánt mérési típusból (kód- és fázismérések a különböző frekvenciákon) vagy a navigációs adatok elavultsága miatt a műhold magassági szöge, azimutja illetve az ionoszféra késleltetés nem volt kiszámolható. A két észlelés között eltelt idő alapján az adatsorban lévő szakadások detektálása is megtörténik és ezek alapján a számításhoz tartozó befejező epocha meghatározzuk.
5. A fázismérések értékeit felszorozzuk a megfelelő hullámhosszal, hogy távolság jellegű értéket kapjunk.
6. A szűrés végrehajtása a *2.2 fejezet*nek megfelelően. (A szűrés során minden epochában kiszámítjuk a paraméterek „nyers” értékeit is, vagyis egyszerűen megoldjuk a (2.6) egyenletrendszert.)

Az következőkben a Kálmán-szűrést végrehajtó Matlab kódot ismertetjük:

A szűréseket egy GPS és egy GALILEO holdra végeztük el, mindkét rendszer esetében négy mért mennyiséget (két frekvencián végzett kód- és fázismérések eredményeit) felhasználva. A fenti algoritmus a GALILEO adatokon történt szűrést mutatja. Az eljárás lényegében csak az egyik vivőhullám frekvenciájának értékének megadásában (E5a frekvencia esetében, ugyanis $E1 = L1$), valamint a megadott mérési típusok kiválasztásában különbözik a GPS műholdon használt algoritmustól.

3.4 EREDMÉNYEK ÉS ÉRTÉKELÉS

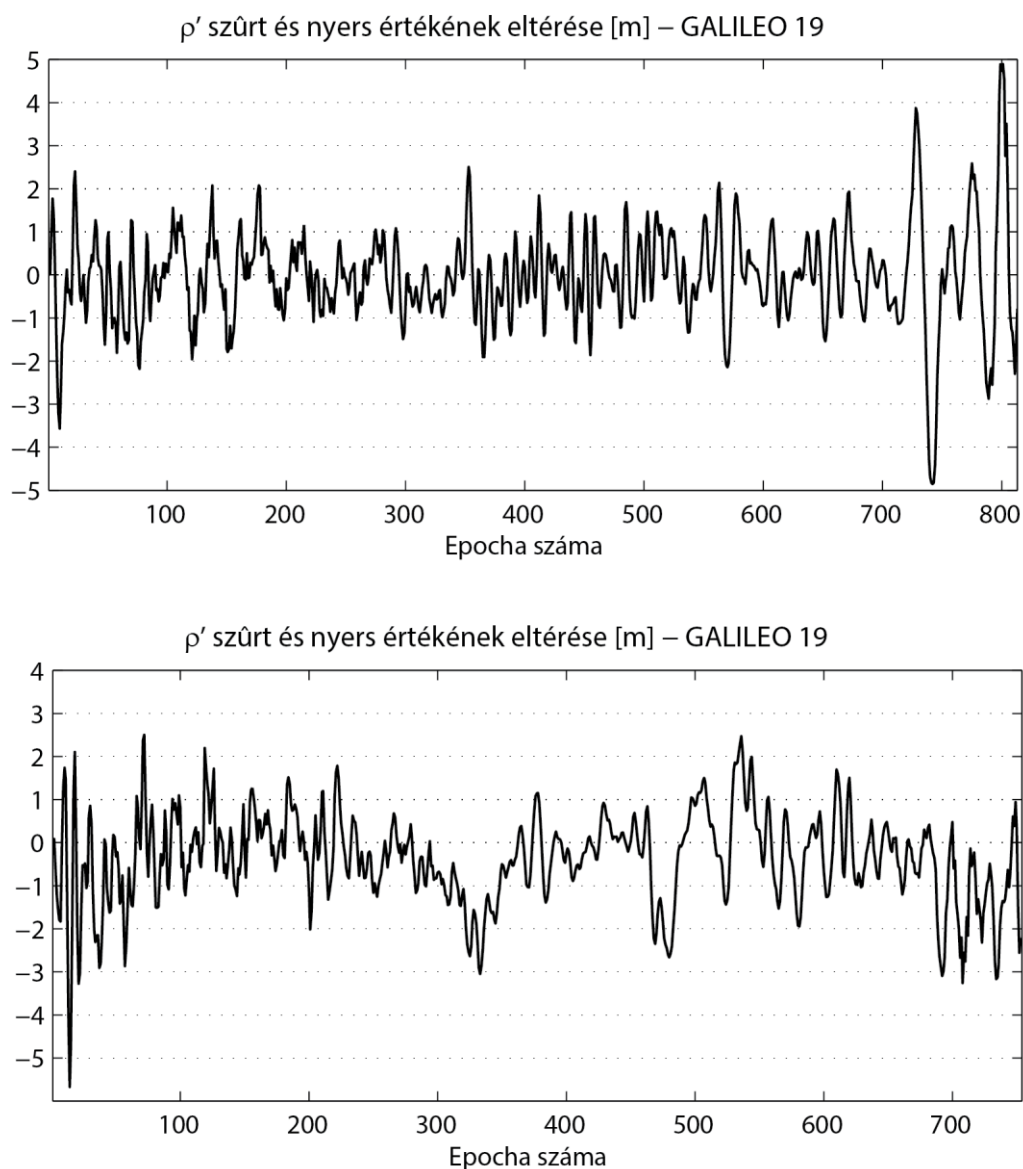
A számítás végrehajtása után az eredményként kapott paraméter tömbből, valamint a kovarianciákat tartalmazó tömbből kiolvashatjuk a paraméterek Kálmán-szűrő által szolgáltatott értékeit és a hozzájuk tartozó kovarianciákat. Az alábbi ábrákon először az N1 és N2 nyers, valamint Kálmán-szűrővel meghatározott értékeit mutatjuk meg, mind a GALILEO, mind a GPS műholdra végzett mérésekből.



3.1. ábra: N1 és N2 értékei. Vastag színes vonallal a szűrt eredmények, fekete vonallal a nyers eredmények látható

A fenti ábrákon az x tengely mentén a mérési epocha sorszáma, az y tengelyen pedig a ciklusegység látható. Az ábra elejét megfigyelve észrevehető, hogy a kezdeti zajosság után, kb. 100 mérési epochát meghaladva a szűrt értékek változása lecsökken, kvázi konstans értékre állnak be. A mérések zajossága inentől kezdve tehát már nem érezhető számottevő mértékben. Az értékek validációjára sajnos jelen körülmények között nem volt lehetőségünk, ugyanis a dolgozat befejezéséig nem álltak rendelkezésre a szükséges ellenőrző adatok a BUTE állomásra. Mindenesetre jó feltételezés abból kiindulni, hogy a valóságtól való (esetleges nagyfokú) eltérés az 1. fejezetben ismertetett elhanyagolásoknak köszönhető.

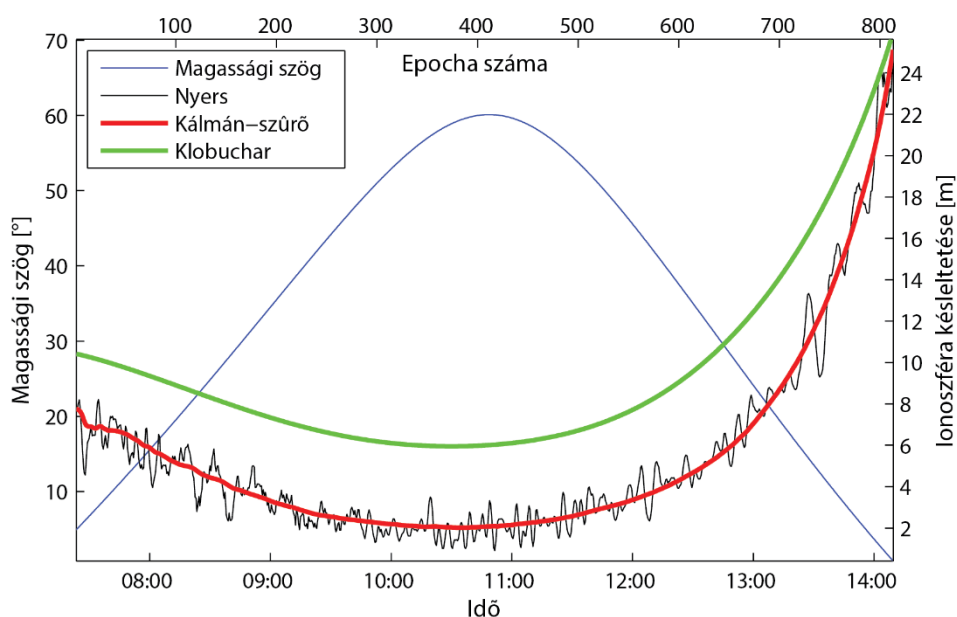
A következő két ábrán ρ' szűrt és nyers értékeinek eltérést mutatjuk be mindkét műhold mérései alapján.



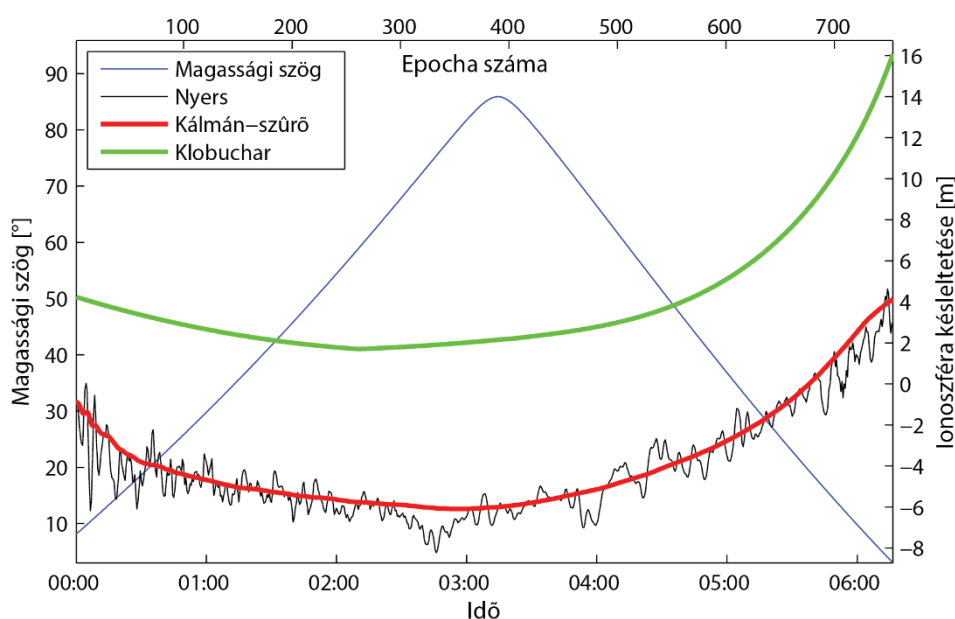
3.2. ábra: A nyers és szűrt ρ' értékek mindkét műholdnál

Az ábrákról szembetűnik, hogy az eltérés átlagértéke a 0-hoz közel esik mindkét esetben, tehát a szűrő megpróbálja a lehető legjobban közelíteni a zajmentes mérési adatot. Az eltérések minden esetben -5 m és +5 m között vannak. Az ionoszféra késleltetést tekintve szintén ábrázoltuk a nyert és a Kálmán-szűrővel kapott értékeket, valamint a Klobuchar-modell felhasználásával számolt ionoszféra késleltetést is, továbbá feltüntettük a műhold magassági szögét.

A kapott eredmények alapján az látható, hogy a szűrt mérési eredményekből számított késleltetés, valamint a Klobuchar modell által adott késleltetés között egy kvázi konstans eltolás érték a különbség. Csakúgy, mint az N értékek esetében, itt is jó okunk van feltételezni, hogy az 1. fejezetben tett elhanyagolások okozhatják az eltérést.



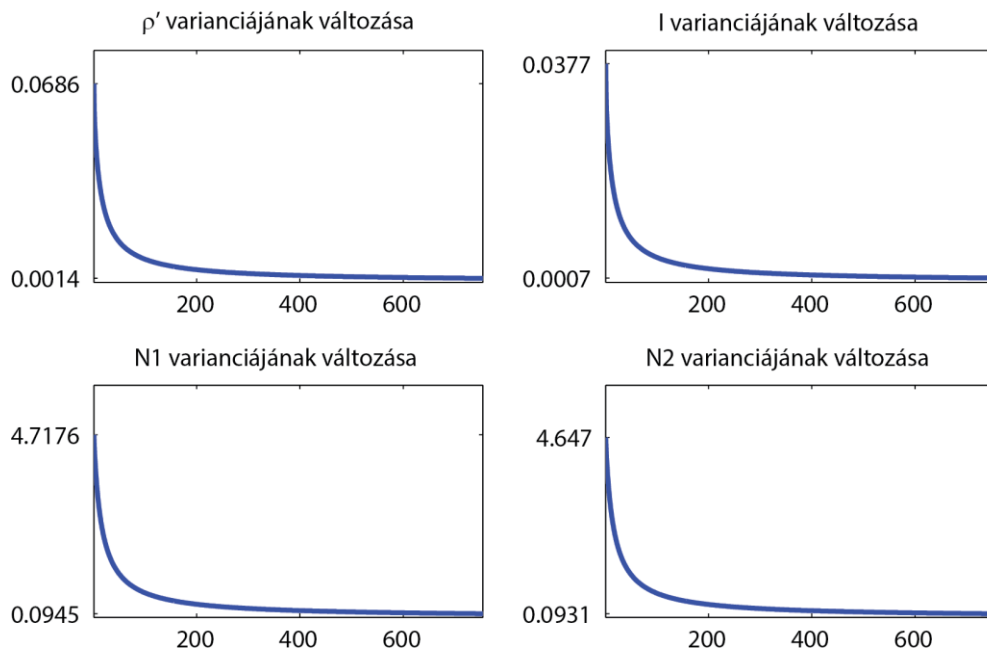
3.3. ábra: Az ionoszféra késleltetések és a magassági szög értékei a GALILEO-19 műhold esetében.



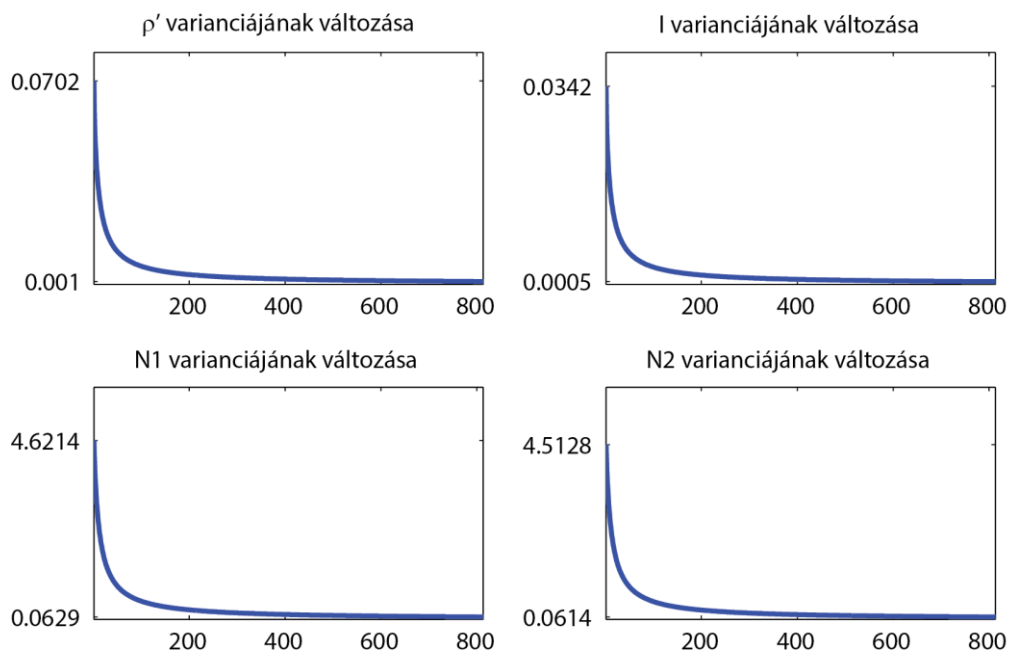
3.4. ábra: Az ionoszféra késleltetések és a magassági szög értékei a GPS-01 műhold esetében

Az ábrák alapján a GPS-műhold esetében nagyobb eltérés mutatkozik a Klobuchar-modell értékei és a szűrt értékek között. Az értékek közötti jobb összhang megteremtése valószínűsíthetően az (1.3) és (1.4) egyenletekben alkalmazott elhanyagolások kiküszöbölésével (pl. a hadverkésések figyelembevételével) lehetséges.

Végezetül közöljük a paraméterek kovariancia mátrixának alakulását a szűrés folyamata alatt.



3.5. ábra: Variancia-értékek alakulása a szűrés folyamán (GPS-01). Az ábrák mellett a maximum és a minimum van feltüntetve.



3.6. ábra: Variancia-értékek a GALILEO-19 esetében. Az ábrák mellett a maximum és a minimum érték van feltüntetve.

Látható, hogy a paraméterek varianciája a szűrés alatt folyamatosan, gyakorlatilag exponenciális mértékben csökkent. Mintegy 150 epocháig a javulás jelentős, 150 és 800 között azonban már ellaposodik a görbe, a mérések ilyen mértékű növelésével már nem tudunk számottevő javulást elérni a varianciák értékében.

4. ÖSSZEFOGLALÁS, TOVÁBBI LEHETŐSÉGEK

Dolgozatomban a BUTE permanens állomás 2014. 04. 29-én végzett méréseinek feldolgozását végeztem el. Az állomás észleléseit, valamint a műholdak navigációs üzenetét tartalmazó RINEX fájlokból kiolvastam az egyes műholdakra vonatkozó mérési adatokat, továbbá a műholdakhoz tartozó navigációs üzenetek adatait. A navigációs adatokból az egyes mérési epochákhoz kiszámítottam a műhold magassági szögét, azimutját, amelyekből megállapítottam az ionoszférikus késleltetés Klobuchar-modell szerinti értékét. Mindezek után a műholdakra vonatkozó zajos észleléseket Kálmán-szűrő felhasználásával dolgoztam fel, ezáltal a zaj hatásától mentes közelítést kapva a meghatározandó paraméterekre (az órahiba és a troposzféra hatását tartalmazó geometriai távolság, ionoszférikus késleltetés, N1 és N2 ciklustöbblettelműség érték). A feldolgozást a mérési periódus során látott műholdak közül egy GPS és egy GALILEO műholdra végeztem el.

A TDK munkám során sajnos nem volt lehetőségem a mérési adatok szűrése után nyert eredmények teljes körű ellenőrzésére és igazolására, így a jövőbeli legfontosabb teendők között ez áll a prioritási lista tetején. Ennek legfőbb kiindulási iránya a műholdakra és a feldolgozó állomásra vonatkozó hardverkésés értékek beszerzése és figyelembevétele a mérési egyenletek felírása esetében. Ugyanakkor más lehetőségek is mutatkoznak az adatok szűrővel történő pontosabb kinyerésére, mint pl. a mérések lineáris kombinációinak kihasználása, másfajta szűrőtípus implementálása a számításba stb.

Véleményünk szerint a GALILEO műholdak irányába is érdemes lehet kitekintést tenni, ugyanis a polgári felhasználásra alkalmas mérési típusok nagyobb száma miatt (több vivőhullám és moduláció) akár jobb eredményeket is elérhetünk, hogyha a mérési egyenletrendszerünket kibővítjük. Jelenleg a rendszer nem üzemel teljes kiépítettséggel, de a feljuttatott és mérésre alkalmas műholdak száma a közeljövőben minden valószínűség szerint növekedni fog, így abszolút érdemes lehet megvizsgálni a rendszer lehetőségeit, előnyeit és hátrányait a mérésfeldolgozás szemszögéből.

5. MATLAB PROGRAMKÓDOK

5.1 AZ ÉSZLELÉSI ÉS NAVIGÁCIÓS RINEX FÁJLOK FELDOLGOZÁSA

Észleléseket tartalmazó RINEX állomány beolvasása és feldolgozása:

```
%%% BEMENETI ADATOK -----

inputfile = '02071190.14o';
system = 'E';
fileout = 'teszt.mat';

%%% BEMENETI ADATOK VÉGE -----

% fájl megnyitása
fid = fopen(inputfile, 'r'); % Információk kinyerése a HEADER-ből
eoh = 0;
while ~eoh
    sor = fgetl(fid);
    % az álláspont közelítő koordinátáinak beolvasása
    if ~isempty(strfind(sor, 'APPROX POSITION XYZ'));
        approx = textscan(sor, '%f %f %f %*s');
        xyz(1) = approx{1}; xyz(2) = approx{2}; xyz(3) = approx{3};
    end
    % a mérések típusának beolvasása
    if strcmp(sor(1), system) && ~isempty(strfind(sor, 'OBS TYPES'))
        obst = strsplit(sor);
        oszlopszam = str2num(obst{2});
        % max. 13 mérési típus fér el egy sorban, ha ennél több van,
        % akkor a köv. sort is be kell olvasni
        if oszlopszam > 13
            obst2 = strsplit(fgetl(fid));
            l = 16;
            for k = 2:length(obst2)
                obst{l} = obst2{k};
                l = l + 1;
            end
        end
    end
    % header vége
    if ~isempty(strfind(sor, 'END OF HEADER'));
        eoh = 1;
    end
end

% Fejléc előállítás
fejlec = [{'PRN' 'EV' 'HO' 'NAP' 'ORA' 'PERC' 'MP'} obst{3:3+oszlopszam-1}];

sorok = 1; % sorok változó inicializálása
% Epocha adatok és észlelési adatok kinyerése
while ~feof(fid)
    sor = fgetl(fid);
```

```

% epochát jelző sor azonosítása
if strcmp(sor(1), '>')
    epochasor = strsplit(sor);
    % nem "egészséges" epochák kiszűrése
    if epochasor{8} == '0'
        epoch = textscan(sor, '> %f %f %f %f %f %f')
        ev = epoch{1}; ho = epoch{2}; nap = epoch{3}; ora = epoch{4};...
        perc = epoch{5}; mp = epoch{6};
        % ha az epocha nem megfelelő, kihagyjuk
    else
        skipsor = epochasor{8};
        for i = 1:skipsor
            fgetl(fid);
        end
    end
    ep_adatok = [ev ho nap ora perc mp];
end

% észleléseket tartalmazó sor azonosítása
if sor(1) == system
    prn = str2num(sor(2:3));
    l = 4;
    m = l + 13;
    % a mérési típusoknak megfelelő számú oszlop beolvasása
    for k = 1:oszlopszam
        % üres sor esetén megszakítás
        if l > length(sor)
            break
        end
        % ha a sorhossz nem megfelelő, 1-gyel csökkentjük
        if m > length(sor)
            m = m - 1;
        end
        % a beolvasott string számmá alakítása
        ertek = str2num(sor(l:m));
        % ha az adott oszlop üres, akkor NaN-rel töltjük fel
        if isempty(ertek)
            meres(k) = NaN;
        else
            meres(k) = ertek;
        end
        l = m + 3;
        m = l + 13;
    end
    % az adatsor összefűzése
    obs_adatok(sorok,:) = [prn ep_adatok meres];
    sorok = sorok + 1;
end

end

% az outputfájl mentése és a fájlok bezárása
save(fileout, 'xyz', 'fejlec', 'obs_adatok');
fclose(fid);
clear all;

```

Navigációs adatokat tartalmazó RINEX állomány beolvasása és feldolgozása:

```
%%% BEMENETI ADATOK -----

inputfile = 'brdm1190.14p';
system = 'G';
outputfile = 'brdm1190_GPS_NDATA.mat';

%%% BEMENETI ADATOK VÉGE -----

% fájl megnyitása
fid = fopen('brdm1190.14p');

% header beolvasása és kihagyása
eoh = 0;
while ~eoh
    line = fgetl(fid);
    if ~isempty(strfind(line, 'END OF HEADER'))
        eoh = 1;
    end
end

% Ephemeris adatok beolvasása
i = 1; % sorok számának inicializálása
while ~feof(fid)
    line = fgetl(fid);
    % navigációs adatokat tartalmazó sor azonosítása
    if ~isempty(line) && line(1) == system
        data = textscan(line, [system '%f %f %f %f %f %f %f %f %f %f']);
        prn = data{1}; % műhold száma
        year = data{2}; % epocha adatok
        month = data{3};
        day = data{4};
        hour = data{5};
        minute = data{6};
        second = data{7};
        af0 = data{8}; % műhold órahiba paraméterei
        af1 = data{9};
        af2 = data{10};

        line = fgetl(fid);
        data = textscan(line, '%f %f %f %f');
        IODE = data{1}; % Issue of Data, Ephemeris adat
        crs = data{2}; % radiális korr. szin.
        deltan = data{3}; % középmozgás korr.
        M0 = data{4}; % középánómália a ref. epochában

        line = fgetl(fid);
        data = textscan(line, '%f %f %f %f');
        cuc = data{1}; % pályamenti korr. cos.
        ecc = data{2}; % excentricitás
        cus = data{3}; % pályamenti korr. szin.
        roota = data{4}; % fél nagytengely gyöke
        line = fgetl(fid);
        data = textscan(line, '%f %f %f %f');
        toe = data{1}; % ephemeris vonatkoztatási epochája
        cic = data{2}; % pályasíkra merőleges korr. cos.
        Omega0 = data{3}; % felszálló csomó hossza a ref. epochában
        cis = data{4};
```

```

line = fgetl(fid);
data = textscan(line, '%f %f %f %f');
i0 = data{1};          % inklináció értéke a ref. epochában
crc = data{2};          % radiális korr. cos.
omega = data{3};        % perigeum argumentuma
Omegadot = data{4};    % felszálló csomó hosszának driftje

line = fgetl(fid);
data = textscan(line, '%f %f %f %f');
idot = data{1};         % inklináció megváltozása
codes = data{2};        % L2 kódok száma
weekno = data{3};       % hét sorszáma
L2flag = data{4};       % L2 P adat

line = fgetl(fid);
data = textscan(line, '%f %f %f %f');
svaccur = data{1};      % SV megbízhatóság
svhealth = data{2};
tgd = data{3};          % csoportkésés
iodc = data{4};         % Issue of Data, Clock adat

line = fgetl(fid);
data = textscan(line, '%f %f %f %f');
tom = data{1};          % az üzenet időpontja (GPS hét másodperceiben)
fit = data{2};          % a pályaadatok illesztésének intervalluma

% az adatok összefűzése egyetlen sorba
eph(i,1) = prn; eph(i,2:7) = [year month day hour minute second];
eph(i,8) = af2; eph(i,9) = M0; eph(i,10) = roota; eph(i,11) =
deltan;
eph(i,12) = ecc; eph(i,13) = omega; eph(i,14) = cuc; eph(i,15) = cus
eph(i,16) = crc; eph(i,17) = crs; eph(i,18) = i0; eph(i,19) = idot;
eph(i,20) = cic; eph(i,21) = cis; eph(i,22) = Omega0;
eph(i,23) = Omegadot; eph(i,24) = toe; eph(i,25) = af0;
eph(i,26) = af1; eph(i,27) = weekno;

i = i + 1;
end
end

fclose(fid);

% Kimeneti fájl létrehozása
save(outputfile, 'eph');
clear all;

```

5.2 A MÉRÉSEKHEZ TARTOZÓ MAGASSÁGI SZÖG, AZIMUT ÉS KLOBUCHAR FÉLE IONOSZFÉRA KÉSLELTETÉS SZÁMÍTÁSA

A magassági szög és az azimut kiszámítása:

```
function mag_az = msz_az(xyz, plh, epocha, navdata)

% A számításhoz szükséges konstansok megadása
GM = 3.986005e14;
Omegae_dot = 7.2921151467e-5;

% xyz és plh vektorok komponensekre bontása
st_x = xyz(1); st_y = xyz(2); st_z = xyz(3);
phi_0 = plh(1); lambda_0 = plh(2); h_0 = plh(3);

% A topocentrikus kr.-be történő áttéréshez szükséges forg. mátrix
R_topo = [-sin(phi_0)*cos(lambda_0)   -sin(phi_0)*sin(lambda_0)
           cos(phi_0);
           -sin(lambda_0)              cos(lambda_0)
           0;
           cos(phi_0)*cos(lambda_0)   cos(phi_0)*sin(lambda_0)
           sin(phi_0)];
% Az ephemerisz adatok és az észlelési epocha között eltelt idő meghatározása
WN = epocha(1); % GPS hét száma
tow = epocha(2); % héten belül eltelt másodpercek
navdata_WN = navdata(navdata(:,27) == WN,:); % a héthez tartozó nav. adatok
dif = tow-navdata_WN(:,24); % az észlelési és a referencia epocha különbsége
mindif = min(dif(dif >= 0)); % a különbségek minimuma
% ha nincs vagy két óránál régebbi a nav. adat, akkor nem használjuk fel
if isempty(mindif) || mindif > 2*3600
    mag_az = [NaN(1,2)];
    return
end

% A legközelebbi navigációs adatsor kiválasztása
index = find(dif == mindif);
ephemeris = navdata_WN(index(1),8:27);
tk = mindif;

% A navigációs üzenet elemeinek definiálása
af2 = ephemeris(1);
M0 = ephemeris(2);
roota = ephemeris(3);
deltan = ephemeris(4);
ecc = ephemeris(5);
omega = ephemeris(6);
cuc = ephemeris(7);
cus = ephemeris(8);
crc = ephemeris(9);
crs = ephemeris(10);
i0 = ephemeris(11);
idot = ephemeris(12);
cic = ephemeris(13);
cis = ephemeris(14);
Omega0 = ephemeris(15);
```

```

Omegadot = ephemeris(16);
toe = ephemeris(17);
af0 = ephemeris(18);
af1 = ephemeris(19);

% A műhold pályasíkban értelmezett koordinátáinak kiszámítása
A = roota * roota; % pályaellipszis nagytengelye
n0 = sqrt(GM/A^3); % középmozgás referencia értéke
n = n0 + deltan; % középmozgás tényleges értéke
M = M0 + n * tk; % középánomália értéke

% excentrikus anomália számítása iterációval
E = M;
for k = 1:10
    E_old = E;
    E = M + ecc * sin(E);
    dE = E-E_old;
    if abs(dE) < 1.e-12
        break
    end
end

v = atan2(sqrt(1-ecc^2)*sin(E), cos(E)-ecc); % a valódi anomália értéke
phi = v + omega; % a műhold helyzete a pályasíkban
u = phi + cuc*cos(2*phi)+cus*sin(2*phi); % jav. szöghelyzet
a pályasíkban
r = A*(1-ecc*cos(E)) + crc*cos(2*phi)+crs*sin(2*phi); % jav. geocentrikus
táv.
i = i0 + idot*tk + cic*cos(2*phi)+cis*sin(2*phi); % javított inkliná-
ció
Omega = Omega0 + (Omegadot-Omegae_dot)*tk - Omegae_dot*toe; % jav. felsz.
csomó
x1 = cos(u)*r; % pályasíkbeli geocent. koordináták
y1 = sin(u)*r;

% A pályasíkbeli koordináták átszámítása WGS84 rendszerbe
satp(1) = x1*cos(Omega)-y1*cos(i)*sin(Omega);
satp(2) = x1*sin(Omega)+y1*cos(i)*cos(Omega);
satp(3) = y1*sin(i);

% A műhold és a vevő közti koordináta különbségek számítása
deltax = satp(1) - st_x;
deltay = satp(2) - st_y;
deltaz = satp(3) - st_z;
tavolsag = sqrt(deltax^2 + deltay^2 + deltaz^2);

% Áttérés topocentrikus koordinátákra
satp_topo = R_topo * [deltax; deltay; deltaz];

% A műhold magassági szögének és azimutjának kiszámítása
mag_szog = rad2deg(asin(satp_topo(3)/tavolsag));
az = rad2deg(atan2(satp_topo(2), satp_topo(1)));
if az < 0
    azimut = az + 360;
else azimut = az;
end

mag_az = [mag_szog azimut];

end

```

A Klobuchar-féle ionoszférikus késleltetés kiszámítása:

```
function iono_keses = klobuchar(phi_0, lambda_0, azimut, mag_szog, t_nap,
alfa, beta)

% Fénysebesség
c = 299792458;

% Azimut és mag. szög értéke félkör egységben
azimut_felkor = azimut/180;
mag_szog_felkor = mag_szog/180;

% A vevő és az ionoszférikus pont alapfel. távolsága
pszi = 0.0137/(mag_szog_felkor + 0.11) - 0.022;

% Az ionoszférikus pont földrajzi szélessége
phi_ip = phi_0 + pszi*180 * cosd(azimut);
if phi_ip > 75
    phi_ip = 75;
end

% Az ionoszférikus pont földrajzi hosszúsága
lambda_ip = lambda_0 + pszi*180*sind(azimut)/cosd(phi_ip);

% Az ionoszférikus pont geomágneses szélessége
phi_ip_m = (deg2rad(phi_ip) + deg2rad(11.6) * cosd(lambda_ip - 291))/pi;

% Az ionoszférikus pontban a helyi idő meghatározása
t_s = 43200*lambda_ip/180 + t_nap;

% A Klobuchar-modell amplitúdójának meghat.
A_2 = alfa(1) + alfa(2)*phi_ip_m + alfa(3)*phi_ip_m^2 + alfa(4)*phi_ip_m^3;
if A_2 < 0
    A_2 = 0;
end

% A periódus meghatározása
A_4 = beta(1) + beta(2)*phi_ip_m + beta(3)*phi_ip_m^2 + beta(4)*phi_ip_m^3;
if A_4 < 72000
    A_4 = 72000;
end

A_1 = 5e-9;
A_3 = 50400;

% A késleltetés számításához szükséges fázis
x = 2*pi*(t_s - A_3)/A_4;
if abs(x) > pi/2
    deltaT_z = 5e-9;
else
    deltaT_z = A_1 + A_2*cos(x);
end
```

5.3 KÁLMÁN-SZŰRŐ ALKALMAZÁSA A MÉRÉSEKRE

```
clear all; clc; close all;
%%% BEMENETI ADATOK -----
% Vizsgált műhold számának megadása
prn = 19;
ef = 1;
tipus = {'C1C' 'L1C' 'C5Q' 'L5Q'}; % használandó észlelési típusok

% Kiindulási paraméterek megadása
c0 = 299792458; % fénysebesség
f1 = 154*10.23e6; % E1 frekvencia értéke
f2 = 1176.45e6; % E5a frekvencia értéke
lambda1 = c0/f1; % E1 vivőhullám hullámhossza
lambda2 = c0/f2; % E5a --
beta = (f1/f2)^2;
%%% BEMENETI ADATOK VÉGE -----

% az észlelések betöltése
load('GALILEO_teljes.mat');

% észlelési típusok oszlopainak megkeresése
indexek = [];
for ind = 1:length(tipus)
    index = find(strcmp(tipus{ind},fejlec));
    indexek = [indexek index];
end

% a választott műhold adatainak leválogatása
data = obs_data(obs_data(:,1) == prn,:);
% azon adatok kiszűrése, ahol a nav. adat 2 óránál régebbi
data(isnan(data(:,24)),:) = [];
% azon adatok kiszűrése, ahol nem történt észlelés a vizsgált típusból
holNaN = find(sum(isnan(data(:,indexek))));
data(holNaN,:) = [];

% Ugrások detektálása a mérési állományban
ugras = diff(data(:,5)*3600+data(:,6)*60+data(:,7));
ugras_index = find(ugras > 30);

% Végző epocha meghatározása az ugrások fv.-ében
ef_k = find((ef < ugras_index));
if isempty(ef_k)
    el = size(data,1);
else
    el = ugras_index(ef_k(1));
end
el = el-1;

range = data(:,indexek);
range(:,2) = range(:,2)*lambda1;
range(:,4) = range(:,4)*lambda2;

% Koefficiens mátrix megadása
A = [1 1 0 0;
     1 -1 lambda1 0;
     1 beta 0 0;
     1 -beta 0 lambda2];
```

```

% Az észlelések kovarianciája
Sigma_e = diag([0.1 0.000025 0.1 0.000025]);
% Az állapotváltozás kovarianciája
Sigma_eps = diag([100 10 0 0]);
% Kezdeti értékek
Sigma_plus = 10*eye(4,4);
x_minus = A\range(ef,:).';

% Változók inicializálása
x = [];
P = [];
x_nofil = [];

% Szűrési eljárás végrehajtása
for k = ef:el
    % nyers megoldás
    x_nf = A\range(k,:).';
    % Becsült kovariancia
    Sigma_minus = Sigma_plus + Sigma_eps;
    % Nyereségmátrix
    K = Sigma_minus * A.' * inv(A * Sigma_minus * A.' + Sigma_e);
    % paraméterek korrigált értéke
    x_plus = x_minus + K*(range(k,:).' - A*x_minus);
    % paramétervektor elmentése
    x = [x x_plus];
    % Kovariancia korrigált értéke
    Sigma_plus = (eye(4) - K * A) * Sigma_minus;
    % Korrigált paraméterek lesznek az új becslések
    x_minus = x_plus;
    % LKN megoldások összefűzése
    x_nofil = [x_nofil x_nf];
    % Kovariancia mátrixok összefűzése
    P = [P Sigma_plus];
end

```

6. FELHASZNÁLT IRODALOM

Ádám J, Bányai L, Borza T, Busics Gy, Kenyeres A, Krauter A, Takács B (2004): Műholdas helymeghatározása. Műegyetemi Kiadó, Budapest.

Ádám J, Rózsa Sz, Takács B: GNSS elmélete és alkalmazása. Oktatási segédlet, BME, Budapest.

European GNSS (Galileo) Open Service – Signal In Space Interface Control Document. (2010) Issue 1.1.

Gurtner W, Estey L (2009): The receiver independent exchange format version 3.01.

Schaer S (1999): Mapping and predicting the Earth's ionosphere using the Global Positioning System. PHD értekezés, Berni Egyetem, Csillagászati Intézet, 19-21

Strang G, Borre K (1998): Linear algebra, geodesy and GPS. Wellesley-Cambridge Press, Wellesley USA, 511, 543-545, 548-552, 562-565.

Takács B (2006): GPS-mérések feldolgozása. Számítási segédlet, BME, Budapest.