

M Ű E G Y E T E M 1 7 8 2

Az ATR visszaverőfóliára történő alkalmazhatóságának vizsgálata

Szerző: Turák Bence

Konzulens: Dr. Égető Csaba

Tudományos Diákköri Konferencia

Budapest, 2017

TARTALOM

1	Bevezetés	3
2	Mérés	4
2.1	Vizsgálati elrendezések.....	4
2.1.1	A fólia irányvonalban való mozgatása.....	4
2.1.2	A fólia irányvonalra merőleges mozgatása	4
2.1.3	A fólia függőleges tengelye körüli forgatása	5
2.2	Alkalmazott műszerek és eszközök	5
2.2.1	A mérőállomás	5
2.2.2	Interferométer	7
2.2.3	Lineáris vezető.....	8
2.2.4	Visszaverőfóliák.....	8
2.2.5	Az ULYXES rendszer	9
2.3	A vizsgálatok elvégzése.....	9
2.3.1	A szög- és távmérés pontosságának vizsgálata.....	10
2.3.2	A szög- és távolságmérés pontosságának függése a fóliára érkező irányvonal beesési szögétől.....	11
3	Feldolgozás.....	13
3.1	A mozdulatlanság ellenőrzése	13
3.2	A mérőállomás távmérési pontossága	14
3.3	A mérőállomás szögmérési pontossága	16
3.4	A szög- és távolságmérés pontosságának függése a fóliára érkező irányvonal beesési szögétől.....	19
4	Eredmények	20
4.1	A mozdulatlanság ellenőrzése	20
4.2	A meghatározott pontossági mérőszámok	21
4.2.1	A mérőállomás távmérőjét jellemző pontossági mérőszámok.....	21
4.2.2	A mérőállomás szögmérőjét jellemző pontossági mérőszámok.....	23
4.2.3	A mérőállomást jellemző pontossági mérőszámok függése a fóliára érkező irányvonal beesésszögétől.....	26
4.3	Az eredmények célszerű felhasználása (ajánlás).....	33

4.3.1	A kapott eredmények közötti inkonzisztencia feloldása.....	33
4.3.2	Különböző koordináta meghatározási módszerek (pontkapcsolások).....	36
5	Összefoglalás	40
6	Irodalomjegyzék	41
7	Köszönetnyilvánítás.....	42
8	Melléletek.....	43

1 BEVEZETÉS

Mozgás- és deformációvizsgálati rendszerek tervezésekor, ahhoz, hogy a felmerülő pontossági igényeket kielégíthessük, a lehető legpontosabb képpel kell rendelkezünk a műszerek pontossági mérőszámairól. A mérések pontossági tervezéséhez felhasználható a műszer specifikációi között leírt a priori középhiba, az általunk előzetesen meghatározott középhiba. A vizsgálat elvégzését követően számított a posteriori középhiba alkalmas a méréseink ellenőrzésére.

Dolgozatom célja ezen pontossági mérőszámok meghatározása, mind szögmérés, mind távolságmérés tekintetében, és ezek összevetése a műszergyártók által meghatározott pontossági mérőszámokkal. Tekintve, hogy bár az automatizált mérőrendszerek egyre elérhetőbbé válnak, esetenként nem mellőzhetjük a kézi irányzást, így a méréseket elvégeztük ATR (Automatic Target Recognition – automatikus célfelismerő) használatával és kézi irányzással is.

A cél, hogy a kapott eredményeket bármilyen mozgás- és deformációvizsgálati rendszer esetén alkalmazni lehessen, ezért a méréseket visszaverőfóliára végeztük. Bár a visszaverő prizmával (lehet kör-, vagy akár a miniprizma is) pontosabb mérések végezhetőek, ezek nem alkalmazhatók abban az esetben, ha a szerkezetet tönkremenetelig terhelik terhelik. Ahhoz, hogy a lehető legteljesebb képet kaphassuk a műszer pontossági mérőszámairól, a méréseket több távolságon, különböző méretű fóliákra is elvégeztük.

2 MÉRÉS

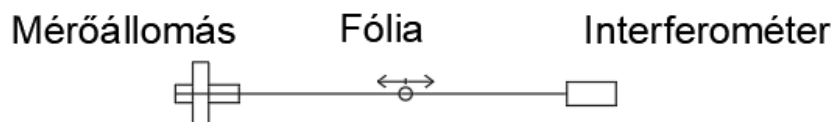
A vizsgálat tárgya egy megfelelően kiválasztott mérőállomás, amellyel méréseket végeztem a visszaverőfóliákra. (A kiválasztás szempontjait a 2.2.1 szakaszban részletezem.) A vizsgálat három részből áll, amelyekben a műszer szögmérési-, távmérési pontosságát határozom meg, valamint ezek függését a fóliára eső irányvonal beesési szögétől. Egy műszer pontosságának a meghatározásához szükséges, a műszerrel mért mennyiséget legalább egy nagyságrenddel pontosabban ismernünk. Ezt a mérési pontosságot egy interferométer (2.2.2 szakasz) segítségével elérhetjük.

2.1 VIZSGÁLATI ELRENDEZÉSEK

1. A fóliát az irányvonalban mozgattam (távmérés pontossága) (2.1.1 szakasz)
2. A fóliát irányvonalra merőlegesen mozgattam (szögmérés pontossága) (2.1.2 szakasz)
3. A fóliát forgattam a függőleges tengely körül (előbbieket függése a fóliára eső irányvonal beesési szögétől) (2.1.3 szakasz)

2.1.1 A fólia irányvonalban való mozgatása

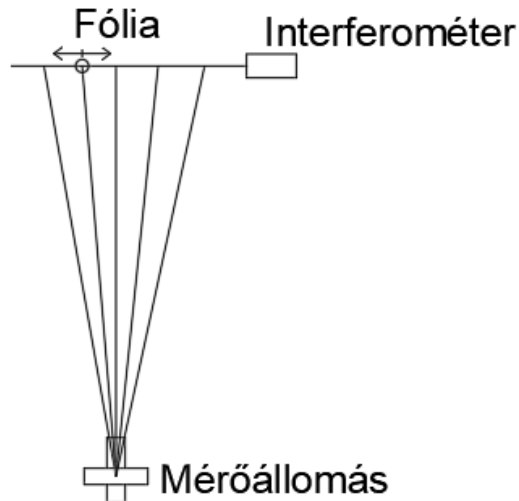
A fóliát az irányvonalban szakaszosan mozgattam (1. ábra), és az interferométerrel (2.2.2 szakasz) mértem az elmozdulásokat. Az egyes pontokban mérőállomással megmértem a távolságokat és az irányértékeket. Az interferométerrel mért távolságokat a feldolgozás során (3.2 fejezet) összehasonlítom a mérőállomással mért értékekkel.



1. ábra: Mérési elrendezés a távmérés pontosságának vizsgálatakor

2.1.2 A fólia irányvonalra merőleges mozgatása

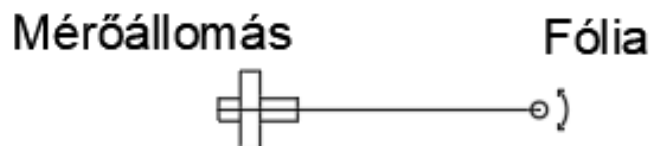
A fóliát az irányvonalra merőlegesen szakaszosan mozgattam (2. ábra), és a 2.1.1 szakaszban leírtaknak megfelelően, az interferométerrel (2.2.2 szakasz) mértem az elmozdulásokat. Az egyes pontokban mérőállomással megmértem a távolságokat és az irányértékeket. A feldolgozás során (3.3 fejezet) a pontok közötti távolságot közvetlenül összehasonlíthatom, az interferométerrel mért távolságokkal.



2. ábra: Mérési elrendezés a szögmérés pontosságának vizsgálatakor

2.1.3 A fólia függőleges tengelye körüli forgatása

A fóliát a függőleges tengelye körül szakaszosan elforgattam ismert szögértékkel (3. ábra), és mérőállomással mértem a távolságot és irányértéket. Ezeket a méréseket a feldolgozás (3.4 fejezet) során egymással hasonlítom össze.



3. ábra: Mérési elrendezés a beesési szög vizsgálatakor

2.2 ALKALMAZOTT MŰSZEREK ÉS ESZKÖZÖK

2.2.1 A mérőállomás

A vizsgált műszer kiválasztásánál fontos volt a minél nagyobb pontosság, valamint, hogy rendelkezzen az automatizált mérőrendszerekhez szükséges beépített motorokkal (robot mérőállomás), és külső számítógépről is vezérelhessük (a vezérléshez szükséges API-t a gyártó szabaddá tegye), továbbá képes legyen az automatikus irányzásra (ATR). Ezért a választás egy Leica TPS1201+ mérőállomásra esett. Ez a műszer kielégíti a fent leírt igényeket.

Accuracy

Type	std. dev. Hz, V, ISO 17123-3		Display least count	
	["]	[mgon]	["]	[mgon]
1201+	1	0.3	0.1	0.1
1202+	2	0.6	0.1	0.1
1203+	3	1.0	0.1	0.5
1205+	5	1.5	0.1	0.5

4. ábra: A Leica TPS1200+ mérőállomások szögmérési pontossága. [Leica TPS1200+ User Manual]

Accuracy



Accuracy refers to measurements to standard prisms.

EDM measuring mode	std. dev. ISO 17123-4, standard prism	std. dev. ISO 17123-4, tape	Measurement time, typical [s]
Standard	1 mm + 1.5 ppm	5 mm + 2 ppm	2.4
Fast	3 mm + 1.5 ppm	5 mm + 2 ppm	0.8
Tracking	3 mm + 1.5 ppm	5 mm + 2 ppm	< 0.15

5. ábra: A Leica TPS1200+ mérőállomások távmérési pontossága. [Leica TPS1200+ User Manual]

A mérőállomás szögmérési pontossága $\pm 1''$ (4. ábra), távolságmérési pontossága visszaverőfóliára $\pm(5\text{mm}+2\text{ppm})$ (5. ábra). Az ATR hatótávolsága visszaverőfóliára 55m (60mm*60mm) (6. ábra).

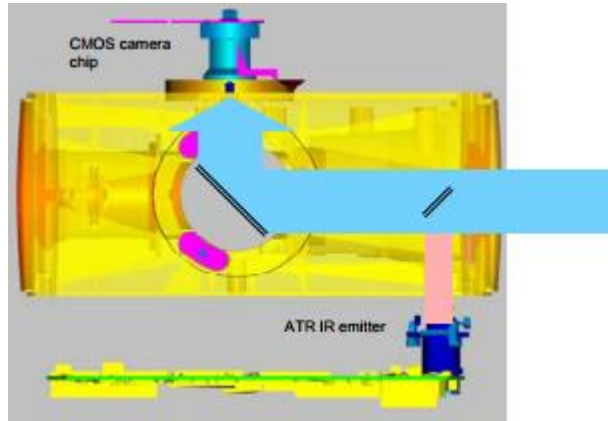
Range ATR/LOCK

Reflector	Range ATR mode		Range Lock mode	
	[m]	[ft]	[m]	[ft]
Standard prism (GPR1)	1000	3300	800	2600
360° prism (GRZ4, GRZ122)	800	2600	600	2000
360° Mini prism (GRZ101)	350	1150	300	1000
Mini prism (GMP101)	500	1600	400	1300
Reflector tape 60 mm x 60 mm	55	175	not qualified	
Machine Automation power prism (MPR122)	600	2000	500	1600
 For Machine Control purposes only!				
 The maximum range can be restricted by poorer conditions, for example rain.				

6. ábra: A Leica TPS1200+ mérőállomások ATR hatótávolsága. [Leica TPS1200+ User Manual]

Az eredmények értelmezéséhez szükséges szót ejteni, az ATR modul működési elvéről. Az automatikus irányzást a mérőállomás saját kibocsájtott elektromágneses hullám segítségével végzi el. Ennek a jelnek a visszaverődését egy nagyfelbontású CMOS kamera érzékeli és ez alapján ismeri fel és irányozza meg a prizmat/fóliát (7. ábra). Ha az aktuális irány és a jel iránya közötti különbség kisebb, mint $16''$, akkor a műszer nem fordul rá a jelre, a visszavert elektromágneses jel alapján kiszámolja a kettő közötti különbséget, és a leolvasott szögértéket mind horizontálisan, mind vertikálisan megjavítja ezekkel az értékekkel. Ha az eltérés nagyobb,

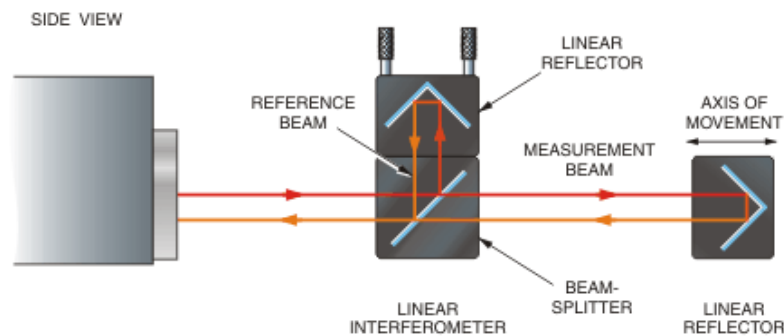
mint 16”, de a jel még látható a műszer távcsövében, a műszer a motorok segítségével ráfordul a jelre, majd az előbb említett módon megjavítja a leolvasott szögértékeket. Ha nem látható jel a műszer távcsövében, a mérőállomás a motorok forgatásával a jel keresésébe kezd [Leica TPS1200+ White Paper].



7. ábra: Az ATR modul sematikus ábrája [Leica TPS1200+ White Paper]

2.2.2 Interferométer

Ezen műszer a fény interferenciajelenségét kihasználva nagy pontosságú távolságok mérésére alkalmas. A műszer ismert frekvenciájú fényt bocsajt ki, ami egy féligáteresztő tükrön halad át, így a fény két nyalábra oszlik. A két fénynyaláb különböző nagyságú utat jár be, így mikor, a visszaverődés után újra egyesülnek, a fáziskülönbségből adódó interferenciából a műszer kiszámítja az elmozdulás mértékét (8. ábra).



8. ábra: Az interferométer működési elve. [Renishaw user guide]

Tekintve, hogy az interferométer az elektromágneses hullám frekvenciáját veszi alapul a számításokhoz, a pontos távméréshez szükséges a környezeti paraméterek ismerete, melyek befolyásolják az elektromágneses hullám frekvenciáját. Az általam alkalmazott Renishaw XL-80 interferométer, rendelkezik saját környezeti egységgel, amely méri ezen paramétereket (hőmérséklet, légnyomás, páratartalom), és korrekcióba veszi a számításokhoz.

Az előzőekben leírt technológiát használva igen nagy pontosságot tudunk elérni a távolságmérésben. Ez a Renishaw XL-80 műszer estében $\pm 0.5\text{ppm}$ (9. ábra), amit, ha 1m távolságra számítunk $\pm 0,5\mu\text{m}$.

Linear measurement		
	Metric	Imperial
Standard range	0-80 m	0-3200 in
Accuracy (with XC compensator)	$\pm 0.5\text{ ppm}^*$	
Resolution	0.001 µm	0.1 µin
Maximum velocity	240 m/min (4 m/s)	9400 in/min (160 in/s)
Note: The accuracy values do not include the errors associated with the normalisation of the readings to a material temperature of 20 °C .		
* $k=2$ (95% confidence) EA-4/02 , ISO		

9. ábra: Renishaw XL-80 interferométer specifikációk lineáris mérés esetén. [Renishaw user guide]

2.2.3 Lineáris vezető

A fólia precíz mozgatásához egy lineáris vezetőt használok, melyet egy korábbi hallgató tervezett, és készített a diplomamunkájához. Az állványzat lehetővé teszi a fólia finom mozgatását, és az interferométer prizmai is könnyen elhelyezhetők rajta (10. ábra). A lineáris vezetőt két műszerállvány segítségével lehet felállítani. Az állványzat tervei és műszaki paraméterei [Piri, 2015] elérhetők Piri Dávid diplomamunkájában.

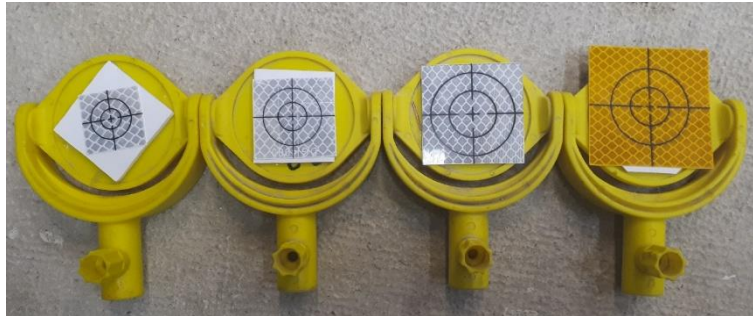


10. ábra: A fólia mozgatásához használt lineáris vezető.

2.2.4 Visszaverőfóliák

A méréseket többfajta visszaverőfóliára is elvégeztem, hogy minél átfogóbb képet kapjunk a műszer pontossági mérőszámairól. Az általam használt fóliák méretei: 30mm*30mm,

40mm*40mm, 50mm*50mm és 60mm*60mm (11. és 12. ábra). A fóliákat egy-egy jeltárcsára ragasztottam és így rögzítettem a lineáris vezetőre, vagy a forgatáskor egy műszertalpra.



11. ábra: A szög- és távmérő pontosságának vizsgálatánál használt fóliák



12. ábra: A forgatáshoz használt fóliák

2.2.5 Az ULYXES rendszer

Az automatizált mérési rendszerekhez szükség van, egy, a műszereket vezérlő külső számítógépre és szoftverre (keretrendszerre), amely a műszerek vezérlését végzi. A méréseimhez az UYLXES (Ultimate Y X Estimation System) [Siki] nyíltforráskódú keretrendszert használtam, amely Python nyelven íródott. Ennek nagy előnye, hogy platformfüggetlen, és népszerű programozási nyelv révén, sok nyíltforráskódú megoldás elérhető az interneten.

2.3 A VIZSGÁLATOK ELVÉGZÉSE

A vizsgálatokat az Általános és Felsőgeodézia Tanszék komparátortermében, illetve az onnan nyíló folyosón végeztem. A mérésekhez az ULYXES rendszert, és az előre megírt filemaker.py és robot.py scripteket használtam.

Minden mérést elvégeztem 10m, 30m és 50m-en, és a 2.2.4 szakaszban leírtak szerint 4 fajta fóliára, hogy lehetőségem legyen a hibákat mindegyik fóliára, a távolság függvényében meghatározni. 50m távolságban a 30mm*30mm-es fóliát az ATR már nem látja, így ezt a vizsgálatot nem tudtam elvégezni.

2.3.1 A szög- és távmérés pontosságának vizsgálata

Ennek a résznek az elején leírtak szerint, a három különböző vizsgáltot három különböző mérési elrendezésben végeztem el. Ezek közül a szögmérés és távmérés középhibáinak a meghatározásához ugyanazokat az eszközöket és műszereket használtam, és a mérés elvégzése is ugyanúgy történt mindkét (szög- és távmérés) esetben, így ezeket egy pontban fogom tárgyalni.

A mérőállomást a komparátor teremben lévő folyosóra nyíló ajtóval szemben található (103-as) pilléren helyeztem el (13. ábra). A pillér alkalmazásával az állványelcsavarodásból keletkező hibákat minimalizáltam. A mérések közben mértem a levegő hőmérsékletét, páratartalmát, és a légnyomást, amiket a mérőállomás korrekcióként figyelembe vesz a távmérés során.

A méréseket két távcsőállásban végeztem. Minden mérés előtt rámértem a komparátorteremben található 608-as prizmára, majd kézi irányzással és ATR-rel mértem a fóliára. A komparátorteremben lévő prizma referencia irányként szolgál. Ezeket a méréseket vizsgálva könnyen észre lehet venni, ha a műszer a pilléren elmozdulna/elfordulna. A mozdulatlanság vizsgálatát a 3.1 fejezetben leírtak szerint végeztem el.



13. ábra: Mérőállomás pilléren, számítógépről vezérelve

A folyosón állítottam fel a lineáris vezetőt az interferométerrel (14. ábra). A lineáris vezetőt két műszerállványra állítottam, a sínen lévő kocsit egy fúró segítségével mozgattam. Az interferométer prizmáit mágnesek segítségével rögzítettem az állványzatra.



14. ábra: Lineáris vezető és az interferométer összeállítva

Az állványzat összeállításakor nagy figyelmet kellett fordítani arra, hogy az elrendezés, a 2.1.1 és 2.1.2 szakaszokban leírtaknak minél jobban megfeleljen, azaz a lineáris vezető az irányvonalba essen, illetve merőleges legyen rá. Továbbá ügyelni kell rá, hogy a lineáris vezető vízszintes legyen, és a mérőállomás irányvonala is vízszintes legyen.

A fóliát 20cm-es tartományon mozgattam oda és vissza, 5cm-es közőket hagyva, azaz egy körben 10-szer mértem egy fóliára (0, 5, 10, 15, 20, 20, 15, 10, 5, 0 cm). Egy fóliára 5 körben mértem rá, így összesen egy távolságon, egy fóliára 50 mérést kaptam ATR-rel és anélkül is. Minden egyes állásban (a 20cm-es mozgatott tartományon belül: 10 helyen, ld. előbb) külön .geo állományt hoztam létre, így egy állományban 6 mérés (komparátor-termi prizma, fólia ATR-rel és fólia kézi irányzással két távcsőállásban: 3x2 db) található, így egy körben 10 db .geo állomány készült. Az interferométer körönként egy .rtl állományt hozott létre.

Minden egyes vizsgálatot **megelőzően** 10-szer rámértem a fóliára ATR-rel és kézi irányzással is, hogy a feldolgozás előtt egy közelítő értéket kaphassunk a középhibára (irány- és távmérés).

A magassági szögmérőre nem végeztem vizsgálatokat, feltételezem, hogy a magassági(zenit) szögméréseket az iránymérésekkel megegyező hiba terheli. Ezt a feltételezést a mérési elrendezés miatt is megtehettem, hiszen a vizsgálatokat vízszintes irányvonal mellett végeztem.

2.3.2 A szög- és távolságmérés pontosságának függése a fóliára érkező irányvonal beesési szögétől

A vizsgálat elvégzésekor a 15. ábrának megfelelően a fóliát egy jeltárcsára ragasztottam, amit egy műszerállványra helyeztem. A forgatás 10° -onként történt, és mivel 10m-en az ATR 50° -os, 30m-en 30° -os, 50m-en pedig már 10° -os beesési szög esetén sem látta a fóliát, így a

vizsgált tartomány 10m-en -40° -tól 40° -ig, 30m-en -20° -tól 20° -ig terjed, 50m távolságon a vizsgálatot nem tudtam elvégezni.

A forgatáshoz egy általam készített szögmérőt használtam, ennek segítségével tudtam a kellő mértékben elfordítani a jeltárcsát és vele a fóliát. A fóliára minden pozícióban 10 mérést végeztem, ATR-rel és kézi irányzással is.



15. ábra: A fólia forgatásához használt szögmérő, és állvány.

3 FELDOLGOZÁS

A nagy mennyiségű mérésre tekintettel úgy döntöttem, hogy a feldolgozást automatizáltan végzem el. Ehhez a Matlab/Octave programozási nyelvet találtam a legcélravezetőbbnek, az egyszerű mátrix-kezelési lehetőségek miatt. A scripteket Matlab szoftverrel futtattam. Ebben a részben, csak a feldolgozáshoz szükséges matematikai modellt írom le, a Matlab forráskódokat a mellékletekben teszem elérhetővé.

A feldolgozás alapvetően mind a három féle vizsgálat esetében két részre bontható. Az első rész az adatok kinyerése és mátrixba rendezése a .geo és .rtl állományokból, a kollimáció és index hiba számítása, valamint a mozdulatlanság ellenőrzése. A második az interferométerrel mért távolságok átalakítása olyan mennyiségre, amit összehasonlíthatunk a mérőállomással mért adatokkal, valamint ezen adatok összehasonlítása. A fólia forgatása esetén a mérőállomással mért mennyiségeket egymással hasonlítjuk össze. Az adatok kiolvasása az állományokból mindegyik vizsgálat esetén megegyezik. A .geo fájlokból a ReadGeo osztály (2. forráskód), a .rtl fájlokból a ReadRtl (3. forráskód) osztály segítségével olvasom ki az adatokat. Mindkét osztálynak az őszosztálya a Read (1. forráskód) osztály.

A mérések tényleges feldolgozását a kollimáció- (1. képlet) és indexhibák (2. képlet) számításával kezdtem, valamint a két távcsőállásban mért távolságokat átlagoltam (3. képlet). A képletek a hibákkal megjavított irány- és zenitszögeket adják meg. A kollimáció- és indexhibákat a GeoTools osztály measureFrom2Faces() statikus metódusával (4. forráskód) számítottam.

$$l = \frac{l_I + (l_{II} \pm 180^\circ)}{2} \quad (1. \text{ képlet})$$

$$z = z_I + \frac{360^\circ - (z_I + z_{II})}{2} \quad (2. \text{ képlet})$$

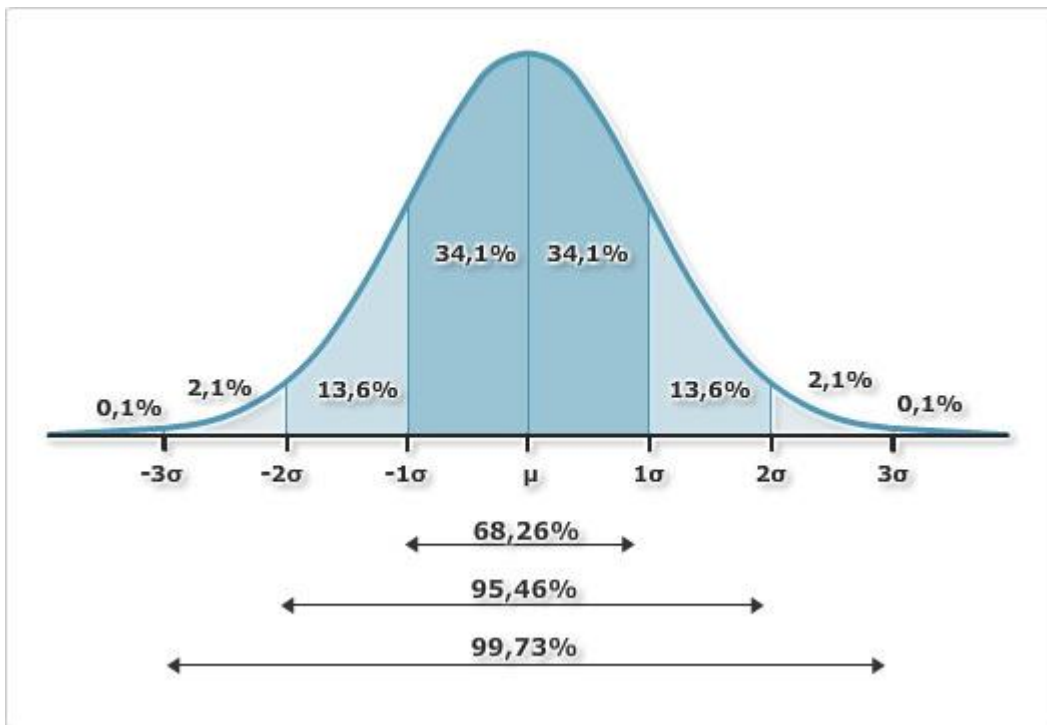
$$t = \frac{t_I + t_{II}}{2} \quad (3. \text{ képlet})$$

Ezután a méréseket csoportosítottam, aszerint, hogy prizmára való referencia mérés, ATR nélküli, vagy ATR-rel való fóliára mérés volt, majd a méréseket mátrixba rendeztem. Ehhez a GeoTools osztály grouping() és struct2matrix() statikus metódusait (4. forráskód) alkalmaztam.

3.1 A MOZDULATLANSÁG ELLENŐRZÉSE

A mozdulatlanság ellenőrzésére a durva hibák kimutatására használható 3-sigma szabályt alkalmaztam. Ennél a módszernél feltételezzük, hogy a méréseink normális eloszlást követnek. A valószínűségi változó várható értékét a mintaközéppel becsüljük. Ekkor annak a valószínűsége, hogy egy véletlen hibával terhelt mérési eredmény a várható érték $\pm 1,2,3\sigma$ tartományba esik, rendre 68,26%, 95,44%, 99,73% (16. ábra), vagyis, ha a várható értéktől

való eltérés nagyobb, mint 3σ , akkor a mérést feltételezhetően durva hiba terheli, azaz a műszer elmozdult [Krauter, 2002].



16. ábra: A 3-sigma szabály [Tasi, 2012]

A mozdulatlanság ellenőrzésének eredményeit a 4.1 fejezet tartalmazza.

3.2 A MÉRŐÁLLOMÁS TÁVMÉRÉSI PONTOSSÁGA

A feldolgozás során célunk, hogy a mérőállomás méréseit összehasonlítsuk az interferométer nagyságrendekkel pontosabb méréseivel. Mivel a mérőállomás távmérési pontossága meg se közelíti az interferométer pontosságát, így az interferométer méréseit hibátlannak feltételezem. Ezzel a feltételezéssel élve a mérőállomás távméréseinek valódi hibáit a 4. képlet segítségével kaphatjuk, ahol L a mérési eredmény, Λ a hibátlan érték [Detrekői, 1991].

$$\varepsilon = L - \Lambda \quad (4. \text{ képlet})$$

Ehhez szükséges a mérőállomás (5. képlet) és az interferométer (6. képlet) távméréseit egymással összehasonlítható mennyiségekre a 17. ábrának megfelelően átszámítanunk.

$$t_{M,i,red} = t_{M,i} - t_{M,0} \quad (5. \text{ képlet})$$

$$t_{I,i,red} = (t_{I,i} - t_{I,0}) * \frac{\cos(\beta)}{\cos(\alpha_i)} \quad (6. \text{ képlet})$$

ahol

t_M : mérőállomással mért távolságok

t_i : interferométerrel mért távolságok

α_i : az irányvonalától való eltérésből származó, a mérőállomással mért törésszögek (ideális esetben 0°)

β : a nullmérés iránya és a fólia mozgatus iránya által bezárt szög (ideális esetben 0°)

A mérőállomással mért redukált távolságot, az abszolút távolságok nullméréshez viszniótt távolságaként kaphatjuk. Az interferométerrel mért redukált távméréseket, az irányvonalától való eltérésből származó hibákkal korrigálva számíthatjuk. Utóbbinak két összetevője a horizontális és vertikális javítás. A korrekciók számításakor nem veszem figyelembe az ATR és a kézi irányzás közötti különbséget, minden esetben az ATR-rel mért irányokat használom.

Az α szög egyszerűen, körönként számítható a nullmérés és a többi mérés irány/zenit-szögének (horizontális és vertikális javítás) különbségeként (7. képlet).

$$\alpha_i = \delta_i - \delta_0 \quad (7. \text{ képlet})$$

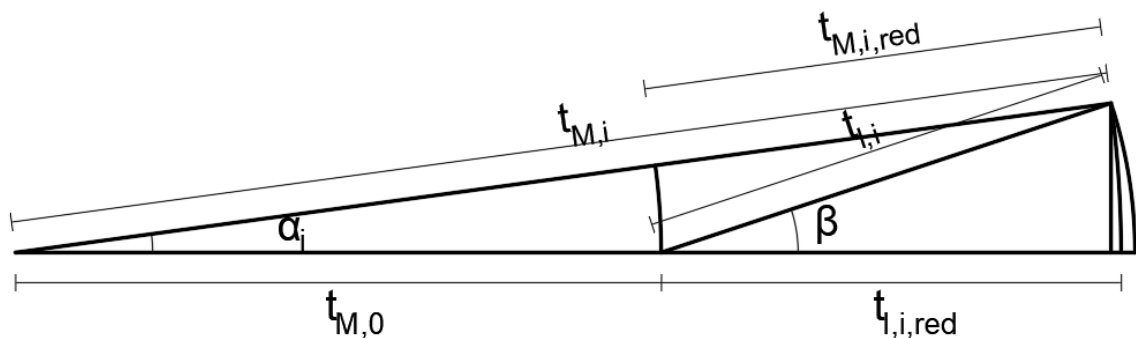
β szög a mozgatus iránya és a nullmérés iránya által bezárt szög (horizontális és vertikális értelemben is) (8. képlet)

$$\beta = \delta_{lin} - \delta_0 \quad (8. \text{ képlet})$$

ahol

δ_i : irányszöggként értelmezett irányértékek (tájékozási szög 0)

δ_{lin} : a lineáris vezető irányszöge



17. ábra: A mérési eredmények redukálása

Ezek alapján a mérőállomás távméréseinek valódi hibája Detrekői képletét (4. képlet) felhasználva (9. képlet):

$$\varepsilon_i = t_{M,i,red} - t_{I,i,red} \quad (9. \text{ képlet})$$

Mind az öt mérési körben számított hibákból kiszámíthatjuk a távmérő középhibáját. Ez a valódi hibák négyzetének várható értékének négyzetgyöke (10. képlet) [Detrekői, 1991]:

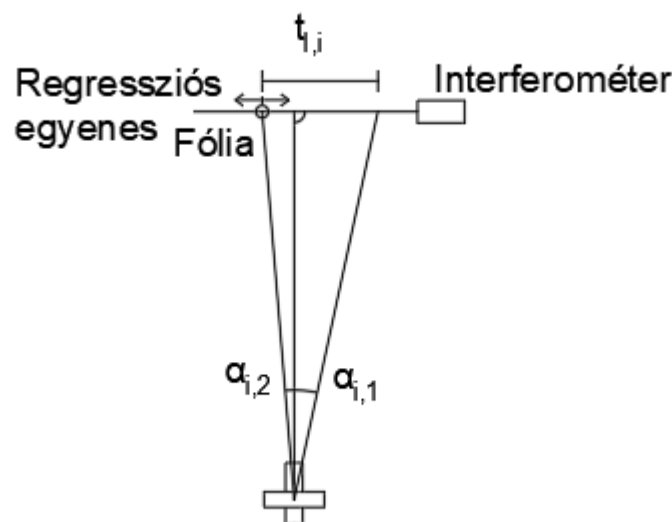
$$\mu = \sqrt{\frac{1}{k} \sum_{i=1}^k \varepsilon_i^2} \quad (10. \text{ képlet})$$

Mindezen számításokat a `direction_process.m` script (5. forráskód) segítségével végeztem el.

3.3 A MÉRŐÁLLOMÁS SZÖGMÉRÉSI PONTOSSÁGA

A szögmérő pontosság vizsgálatához három féle módszert próbáltam alkalmazni. Az első két módszerben az interferométer által mért távolságokat számítottam át törésszögekre, melyek közvetlenül összehasonlíthatók a mérőállomással mért szögekkel. Ám ezek egyike sem alkalmazható, mert hiába tekintem az interferométer méréseit hibátlannak, az átszámításhoz szükségesek a mérőállomás távmérései, melyek a hibák terjedése miatt, még akkor is túl nagy hibákat okoznak a szögekben, ha az általam meghatározott, a vártnál jóval kisebb távmérési középhibákat alkalmazom.

Az első módszerben az általam mért pontokra a legkisebb négyzetek módszerével (5. kiegyenlítési csoport) térbeli regressziós egyenest illesztettem, és ezen egyenesen mozgatott visszaverőfólia interferométerrel mért elmozdulásaiból számítottam a törésszögeket (18. ábra). A módszer nem alkalmas a törésszögek kellő pontosságú meghatározására, ugyanis a meghatározott regressziós egyenes paramétereinek középhibája meghaladja a kívánt értéket, a hibák terjedésével törésszögek hibája túl nagy lesz.



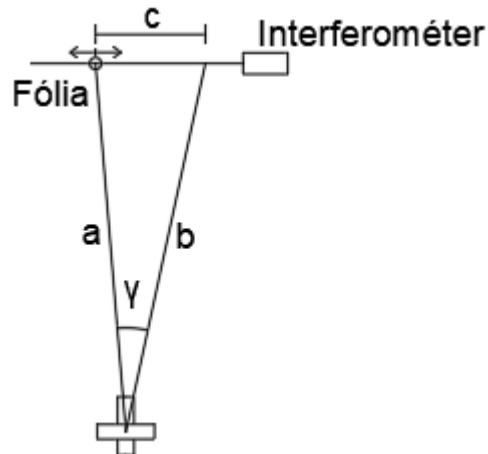
18. ábra: Törésszögek számítása regressziós egyenes segítségével

A második módszer a számítás jóval egyszerűbb az előző módszernél, a törésszögeket közvetlenül számítja a koszinusz-tétel (19. ábra) alapján (11. képlet). A törésszögeket γ -val, az

oldalakat a, b és c-vel jelölve a 12. képlet segítségével számíthatjuk, azonban ezen módszerrel számított középhibák sem teszik lehetővé az értékek összehasonlítását, ezért egy újabb módszert kellett választanom a pontosság vizsgálatára.

$$c^2 = a^2 + b^2 - 2 * a * b * \cos(\gamma) \quad (11. \text{ képlet})$$

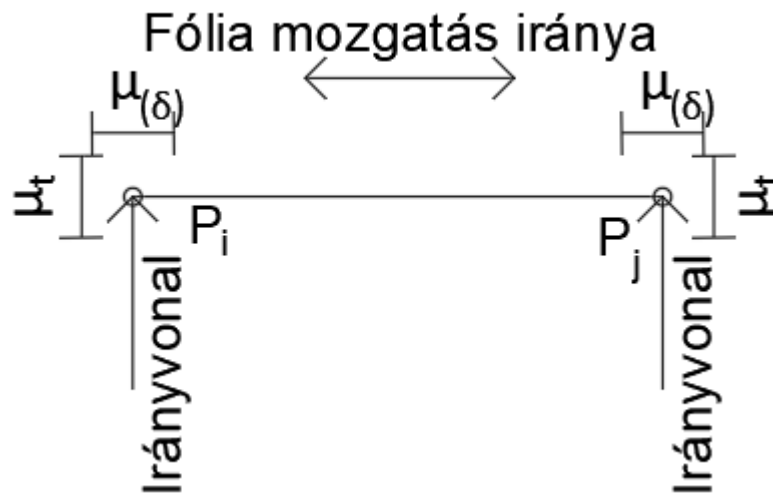
$$\gamma = \arccos\left(\frac{a^2 + b^2 - c^2}{2 * a * b}\right) \quad (12. \text{ képlet})$$



19. ábra: Törésszögek számítása koszinusz-tétel használatával

A harmadik, általam alkalmazott módszer nem az interferométer által mért távolságokat számítja át törésszögekre, hanem a mérőállomás által mért pontok koordinátáit számítja (GeoTools osztály polarPoint() metódus) és a pontok közötti távolságot (GeoTools osztály pointPointDist() metódus) (4. forráskód), hasonlítja össze az interferométerrel mért távolságokkal. Hibaterjedést tekintve ez a módszer bizonyult a legalkalmasabbnak az összes közül. Hátránya, hogy az eredményt nem másodperc, hanem mm/m egységben kapjuk, amiből csak közvetetten számíthatjuk át szög egységre a középhibákat.

A pontok koordinátáit a szögmérés hibáján kívül a távmérés hibája is terheli. Ez számunkra nem kívánatos, ugyanis ezt az előző szakaszban leírtak szerint már meghatároztuk, itt tisztán a szögmérő pontosságát kívánjuk meghatározni. Azonban, ha feltételezzük, hogy a méréseink iránya párhuzamos (ezt a kis iránykülönbségek miatt megtehetjük), és a fólia mozgatásának iránya merőleges az irányvonalra (20. ábra), akkor a hibaterjedés törvénye alapján, a távmérő hibája nem okoz hibát a két pont közötti távolságban. Ezt a GeoTools osztály pointPointDist() statikus metódusával (4. forráskód) számítottam.



20. ábra: A pontok közötti távolságot terhelő nem kívánt hibahatások

Az interferométer méréseit ebben az esetben is redukálni kell, ugyanis a mérőállomással vízszintes szögeket mértünk, a lineáris vezetőt viszont csak megközelítően tudtam vízszintesen felállítani. Az interferométer méréseit a 13. képlet segítségével a 21. ábra alapján redukálhatjuk vízszintesre.

$$t_{I,i,red} = t_{I,i} * \cos(\alpha) = t_{I,i} * \sin(z) \quad (13. \text{ képlet})$$

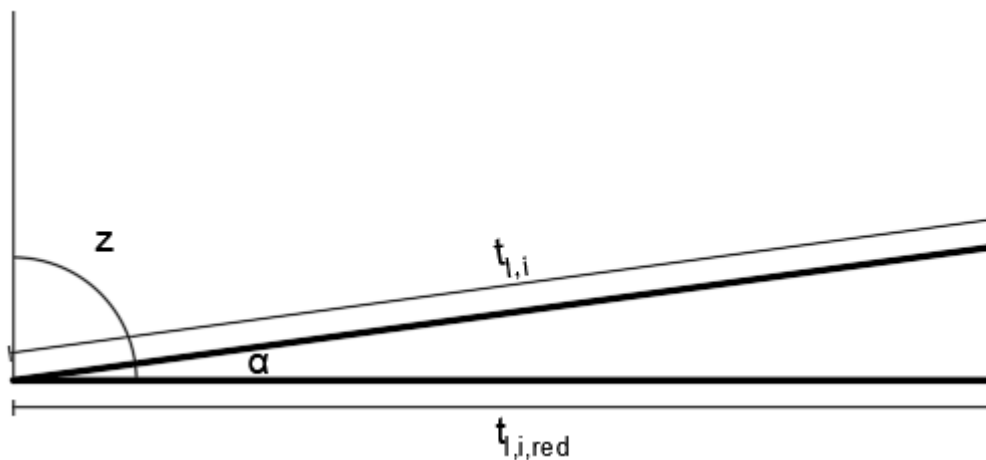
ahol

$t_{I,i,red}$: interferométerrel mért redukált távolság

$t_{I,i}$: interferométerrel mért redukálatlan távolság

α : a lineáris vezető vízszintessel bezárt szöge (magassági szög) (ideális esetben 0°)

z : a lineáris vezető függőlegessel bezárt szöge (zenitszög) (ideális esetben 90°)



21. ábra: Az interferométer méréseinek vízszintesre redukálása

Itt is feltételezzük, hogy az interferométer mérései hibátlanok, így a valódi hibákat a 4. képlet segítségével kaphatjuk, melyekből a középhibákat a 10. képlettel kaphatjuk mm egységben, amit a távolságra fajlagosítva szög értékeket kaphatunk.

A szögmérő pontosságának meghatározásához szükséges számításokat a cross_process.m script (6. forráskód) segítségével végeztem el.

3.4 A SZÖG- ÉS TÁVOLSÁGMÉRÉS PONTOSSÁGÁNAK FÜGGÉSE A FÓLIÁRA ÉRKEZŐ IRÁNYVONAL BEESÉSI SZÖGÉTŐL

A méréseket a beesési szög vizsgálatával zártam. Ehhez a fóliát 10°-onként forgattam, és minden ATR-rel mérhető pozícióban 10-10 mérést végeztem ATR-rel és anélkül is. Minden 10-es sorozat előtt rámértem a komparátor-termi prizmára (referencia irány). Az eddigiekhez hasonlóan a műszert a 103-as pilléren állítottam fel, és mérések a folyosón elhelyezett fóliára történtek.

10m távolságon mindegyik fóliát -40° és 40° között forgattam, az ATR ezt tette lehetővé. 30m-en ez a tartomány mindegyik fólia esetében -20° és 20° közé esett. 50m-en az ATR használata már csak akkor lehetséges, ha az irányvonal megközelítően merőleges a fóliára, még 10° esetén se lehetséges a mérést elvégezni, így ezen a távolságon ezt a fajta vizsgálatot nem tudtam elvégezni.

A vizsgálat arra terjed ki, hogy forgatás esetén számíthatunk-e a mérések várható értékének (átlag) (14. képlet) megváltozására, illetve a mérések szórása (középhiba) (15. képlet) függ-e a fóliára érkező irányvonal beesési szögétől, mind távmérés, mind szögmérés estében.

$$\bar{a} = \frac{1}{k} \sum_{i=1}^k L_i \quad (14. \text{ képlet})$$

$$(\mu) = \sigma = \sqrt{\frac{1}{k-1} \sum_{i=1}^k (L_i - \bar{a})^2} \quad (15. \text{ képlet})$$

4 EREDMÉNYEK

A feldolgozáskor kapott eredményeket a feldolgozásnak megfelelően több részre bontattam. A műszer mozdulatlanságára kapott eredményeket az 4.1 fejezetben, majd a műszert jellemző középhibákat az 4.2 fejezetben ismertetem. A táblázatokban és az ábrákon az állványzat felállításának tökéletlenségéből adódó korrigálatlan, és a korrigált középhibákat is ábrázolom.

4.1 A MOZDULATLANSÁG ELLENŐRZÉSE

A mozdulatlanság ellenőrzést a 3.1 fejezet alapján végeztem el. Az elmozdulás ellenőrzéséhez szükséges szórásnak, a műszergyártó által megadott a priori középhibát vettem fel, ami 1" (2.2.1 szakasz). Ezt megerősíti a 4.3.1 szakaszban leírt 500 elemű mintából számított középhiba prizmára. Így a megengedett legnagyobb eltérés a várható értéktől 3". A mozdulatlanság ellenőrzésre a méréseket a 2.3.1 szakaszban leírtak szerint végeztem el. Az átlagtól való eltéréseket az 1., 2. és 3. táblázatok tartalmazzák.

A prizmára mérések legnagyobb eltérése a várható értéktől (a távmérő pontosságának vizsgálatakor) ["]				
távolság	30mm	40mm	50mm	60mm
10m	1.59	0.93	1.73	0.85
30m	0.79	0.94	0.98	0.91
50m	-	1.56	0.86	0.84

1. táblázat: A komparátor-termi prizmára mérések legnagyobb eltérése a várható értéktől (a távmérő pontosságának vizsgálatakor)

A prizmára mérések legnagyobb eltérése a várható értéktől (a szögmérő pontosságának vizsgálatakor) ["]				
távolság	30mm	40mm	50mm	60mm
10m	1.16	0.99	2.37	0.89
30m	1.09	1.62	0.92	0.91
50m	-	1.54	1.06	0.78

2. táblázat: A komparátor-termi prizmára mérések legnagyobb eltérése a várható értéktől (a szögmérő pontosságának vizsgálatakor)

A prizmára mérések legnagyobb eltérése a várható értéktől (a visszaverőfólia forgatásakor) ["]				
	30mm	40mm	50mm	60mm
10m	0.68	1.03	0.80	0.64
30m	0.84	1.03	0.60	0.96
50m	-	-	-	-

3. táblázat: A komparátor-termi prizmára mérések legnagyobb eltérése a várható értéktől (a visszaverőfólia forgatásakor)

Látható, hogy a 3-sigma szabály alapján egyik vizsgálat esetében se tapasztalunk elmozdulásokat.

4.2 A MEGHATÁROZOTT PONTOSSÁGI MÉRŐSZÁMOK

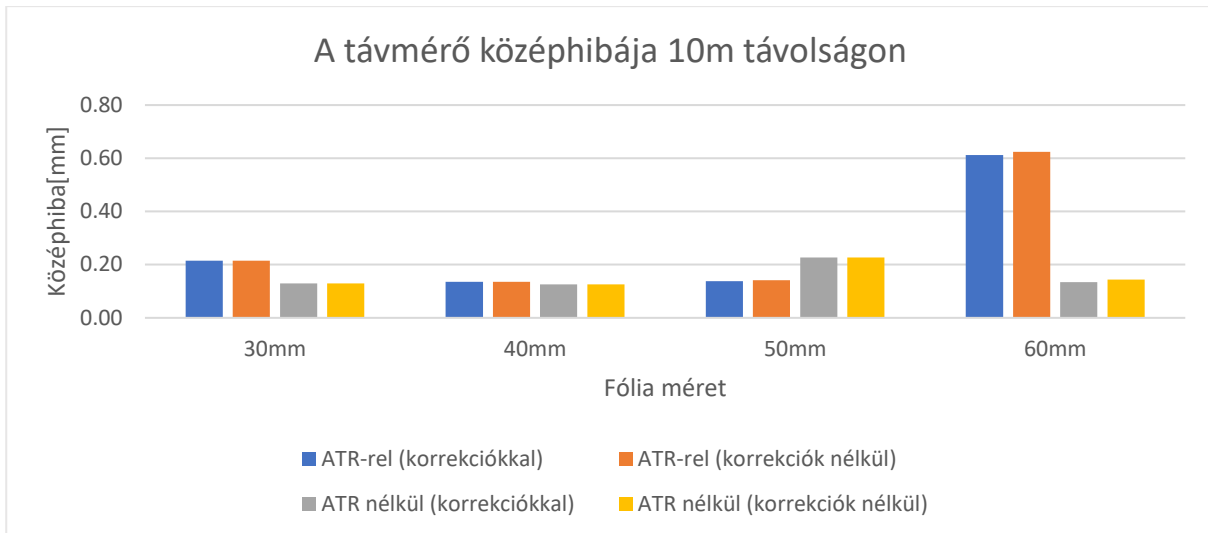
4.2.1 A mérőállomás távmérőjét jellemző pontossági mérőszámok

Az alábbiakban a 3.2 fejezetben leírtak szerint számított pontossági mérőszámokat mutatom be. Az általam meghatározott középhibákat a 4. táblázat tartalmazza és a könnyű áttekinthetőség kedvéért grafikusán is ábrázolom.

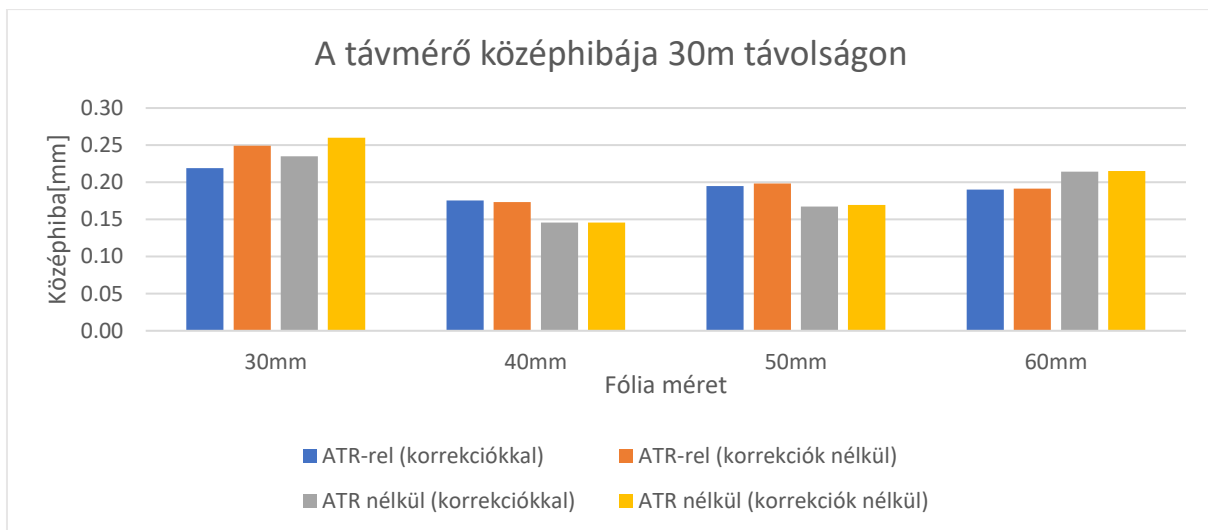
Távmérő középhibája a sorozatmérésekből [mm]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	0.13	0.11	0.07	0.09	0.08	0.09	0.12	0.10
30m	0.16	0.10	0.08	0.17	0.12	0.07	0.13	0.09
50m	-	-	0.24	0.13	0.07	0.10	0.09	0.13
Távmérő középhibája a fólia mozgatásából (korrekciókkal) [mm]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	0.21	0.13	0.14	0.13	0.14	0.23	0.61	0.13
30m	0.22	0.23	0.18	0.15	0.19	0.17	0.19	0.21
50m	-	-	0.31	0.22	0.19	0.19	0.18	0.17
Távmérő középhibája a fólia mozgatásából (korrekciók nélkül)[mm]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	0.21	0.13	0.13	0.13	0.14	0.23	0.62	0.14
30m	0.25	0.26	0.17	0.15	0.20	0.17	0.19	0.22
50m	-	-	0.31	0.22	0.19	0.19	0.18	0.18

4. táblázat: A távmérőt jellemző pontossági mérőszámok az előzetes sorozatmérésekből, valamint a fólia mozgatásával a korrekciók figyelembevételével és anélkül

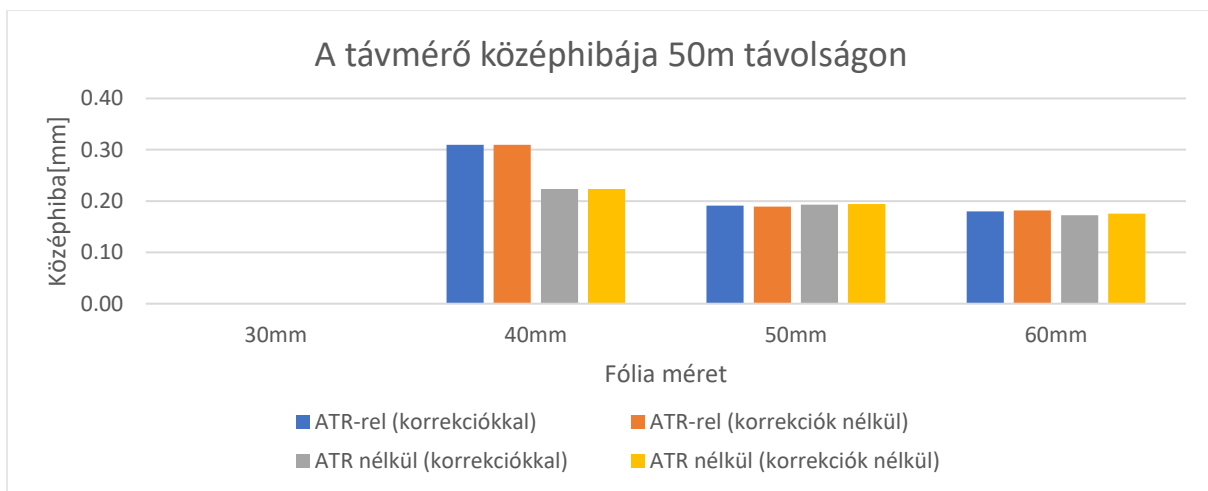
A középhibákat egyenként 45 valódi hiba (50 mérés, melyből 5 nullmérés) alapján számítottam. Látható, hogy a kapott eredmények összhangban vannak egymással, egyedül a 10m-en mért 60mm-es fólia esetében, ATR használatakor tapasztalunk inkonzisztenciát. Bár ez az érték is arra mutat, hogy a műszer sokkal pontosabb, mint amit a műszergyártó állít a műszerről, az általam mért eredményektől nagyban eltér. Az eredményeket távolság szerint a 22., 23. és 24. ábrákon szemléltetem.



22. ábra: A távmérő középhibája 10m távolságon



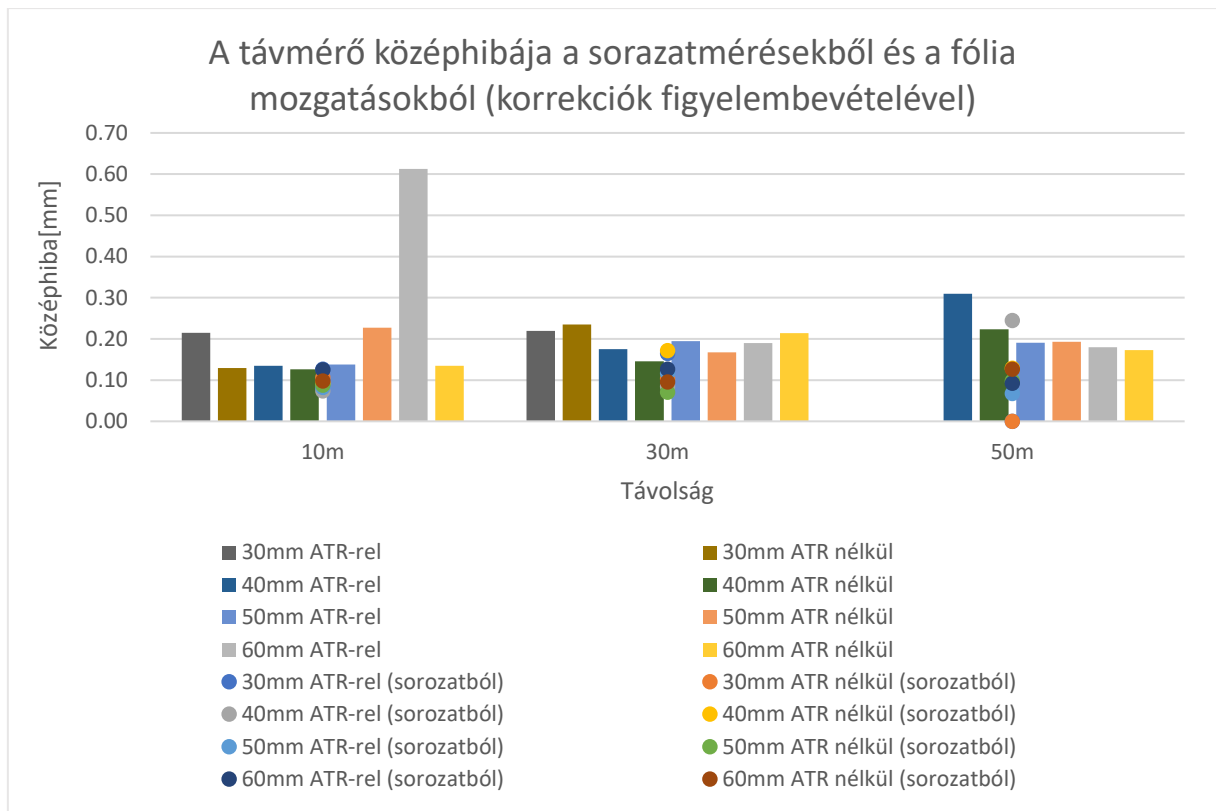
23. ábra: A távmérő középhibája 30m távolságon



24. ábra: A távmérő középhibája 50m távolságon

A fenti ábrákon jól látható, hogy távmérések középhibája azonos távolságokon összhangban vannak, és bár minimális mértékben, de az 50m-es távolságon enyhe fordított arányosság figyelhető meg a pontosság és a visszaverőfólia mérete között.

A 25. ábra csak a korrigált és a sorozatmérésekből számított középhibákat ábrázolom. Ezek szintén jó összhangban vannak egymással, bár a fólia mozgatásával kapott értékek valamivel magasabbak. Azt is észre lehet venni (a várakozásunknak megfelelően), hogy a távolság növekedésével, ugyan minimálisan, de a középhibák is megnőnek.



25. ábra: A távmérő középhibája a sorozatmérésekből és a fólia mozgatásokból (korrekciók figyelembevételével)

4.2.2 A mérőállomás szögmérőjét jellemző pontossági mérőszámok

Ezeket az eredményeket az előző szakaszhoz hasonlóan táblázatban, és diagram segítségével is ábrázolom. A számított középhibák, mint ahogy a 3.3 fejezetben említettem, mm/m egységben kaptam meg.

A 5. táblázatban láthatók a szögmérő méréseinek középhibái, ha ezeket szögekre átszámítjuk, a 6. táblázatban láthatjuk, hogy a számításaim szerint a szögmérő pontossága nem éri el a műszergyártó által meghatározott a priori középhibát, sőt, néhol a sokszorosát is meghaladja. Ezt nem csak a fólia mozgatásából, hanem a sorozatmérésből számított középhibák is alátámasztják. Egyetlen eset van, mikor a középhiba a gyártó által megadott érték alatt van.

Szögmérő középhibája a fólia mozgatásából (korrekciókkal)[mm/m]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	0.02	0.01	0.02	0.01	0.02	0.01	0.01	0.01
30m	0.06	0.01	0.04	0.01	0.03	0.01	0.03	0.01
50m	-	-	0.04	0.01	0.02	0.01	0.02	0.01

Szögmérő középhibája a fólia mozgatásából (korrekciók nélkül)[mm/m]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	0.02	0.01	0.02	0.01	0.02	0.01	0.01	0.01
30m	0.06	0.01	0.04	0.01	0.03	0.01	0.03	0.01
50m	-	-	0.04	0.01	0.02	0.01	0.02	0.01

5. táblázat: A szögmérőt jellemző pontossági mérőszámok mm/m egységben

Az ATR nélküli mérések középhibája sokkal alacsonyabb, és a 26. ábrán azt is észre lehet venni, hogy ezek a mérések ilyen kis távolsági tartományon nem, vagy csak minimálisan függenek a távolságtól, és a fólia méretétől.

Szögmérő középhibája a sorozatmérésekből["]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	1.51	2.24	0.88	2.65	1.35	2.47	1.95	3.03
30m	1.91	2.18	2.41	1.39	1.47	2.07	1.63	2.46
50m	-	-	4.35	2.71	2.93	1.90	1.46	2.63

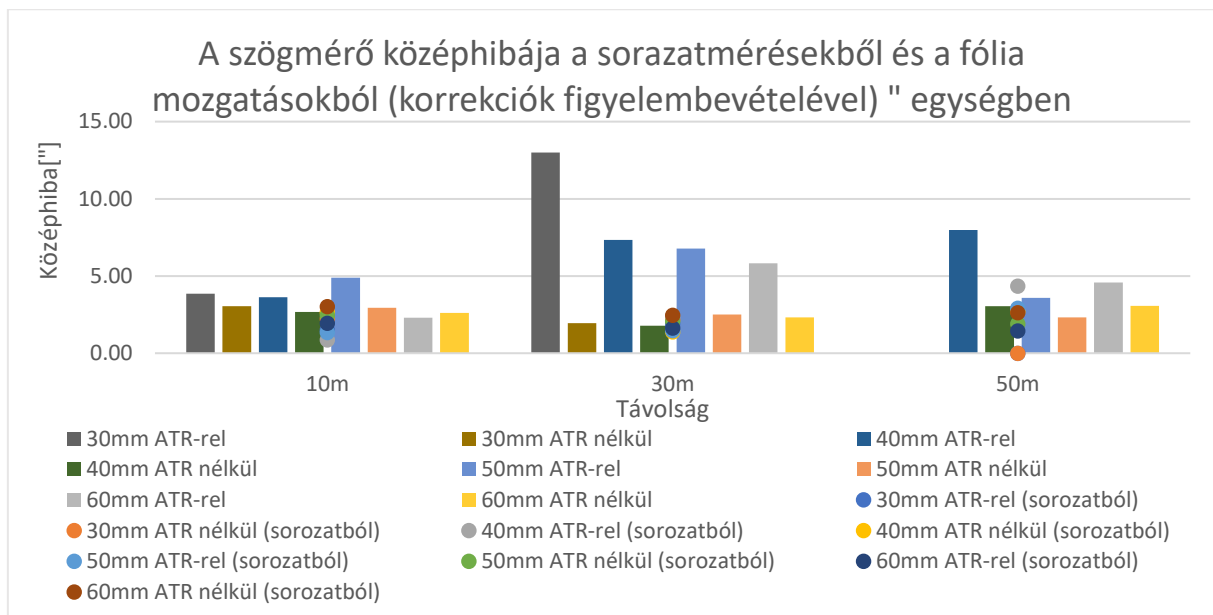
Szögmérő középhibája a fólia mozgatásából (korrekciókkal)["]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	3.86	3.05	3.62	2.66	4.89	2.94	2.31	2.60
30m	13.00	1.94	7.35	1.78	6.77	2.52	5.83	2.31
50m	-	-	7.98	3.05	3.59	2.32	4.58	3.07

Szögmérő középhibája a fólia mozgatásából (korrekciók nélkül)["]								
távolság	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	3.86	3.05	3.63	2.66	4.88	2.94	2.31	2.60
30m	13.00	1.95	7.35	1.78	6.77	2.52	5.83	2.31
50m	-	-	7.99	3.07	3.59	2.32	4.58	3.07

6. táblázat: A szögmérőt jellemző pontossági mérőszámok " egységben

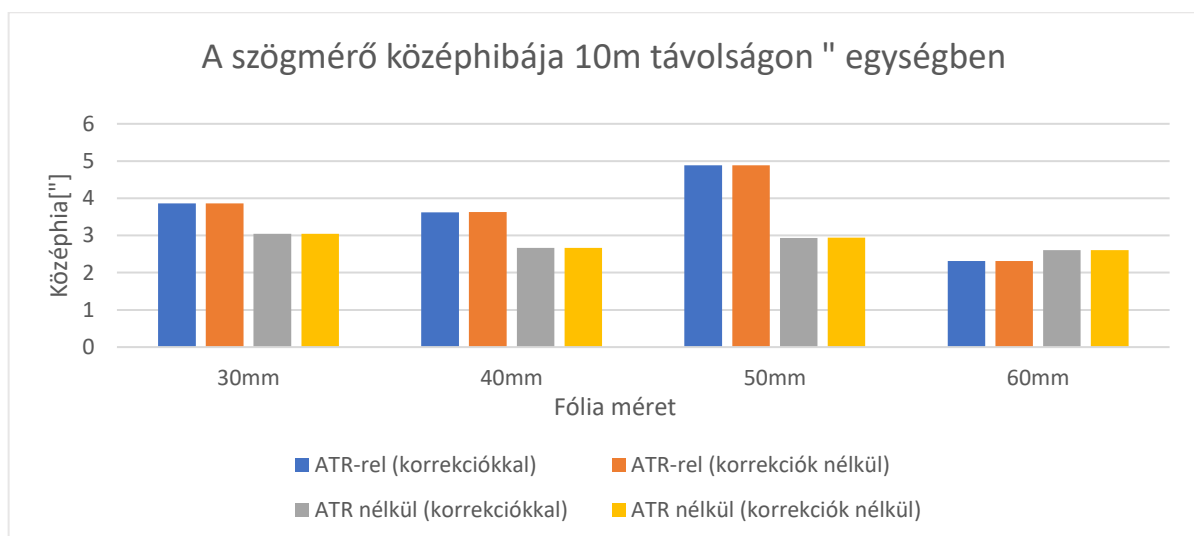
Az ATR-es mérések pontossága erősen függ a fólia méretétől. A középhiba fordítottan arányos a fólia méretével, és bár a távolság esetében nem láthatunk szabályos összefüggést, látható, hogy szintén függ tőle. A legnagyobb középhibák 30m-es távolságon adódnak. A 30mm-es fóliára ezen a távolságon már 13" középhiba adódik. A legkisebb középhibákat 10m-

en sikerült kimutatnom, ezek értéke 2" és 5" között van. 50m távolságon hasonlóak a 30m-en meghatározottakhoz.

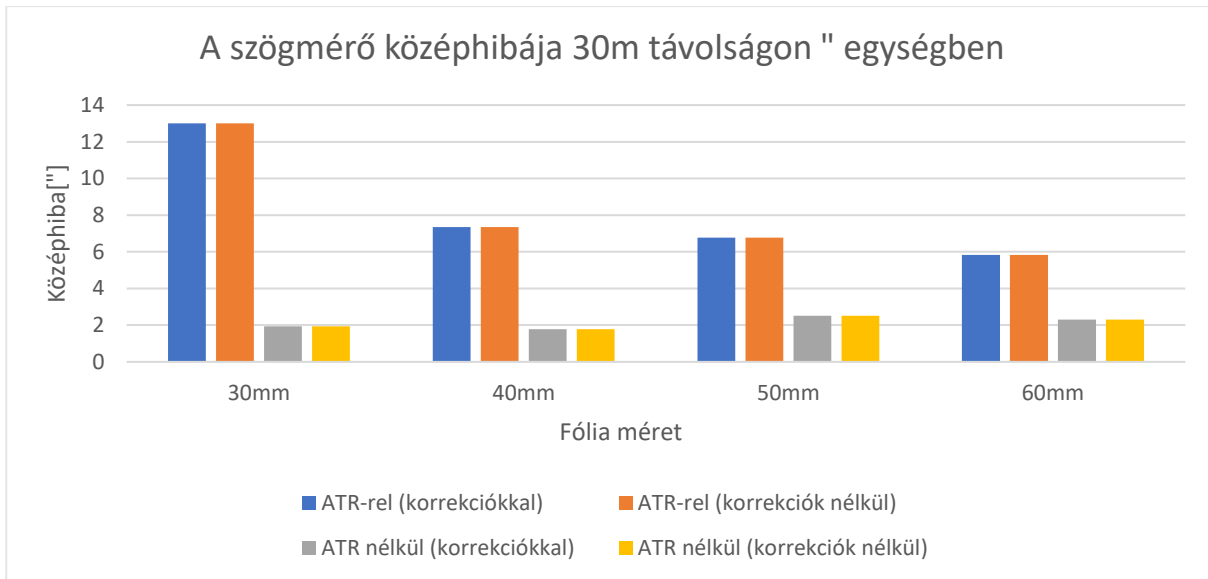


26. ábra: A szögmérő középhibája a sorozatmérésekből és a fólia mozgatókából (korrekciók figyelembevételével)

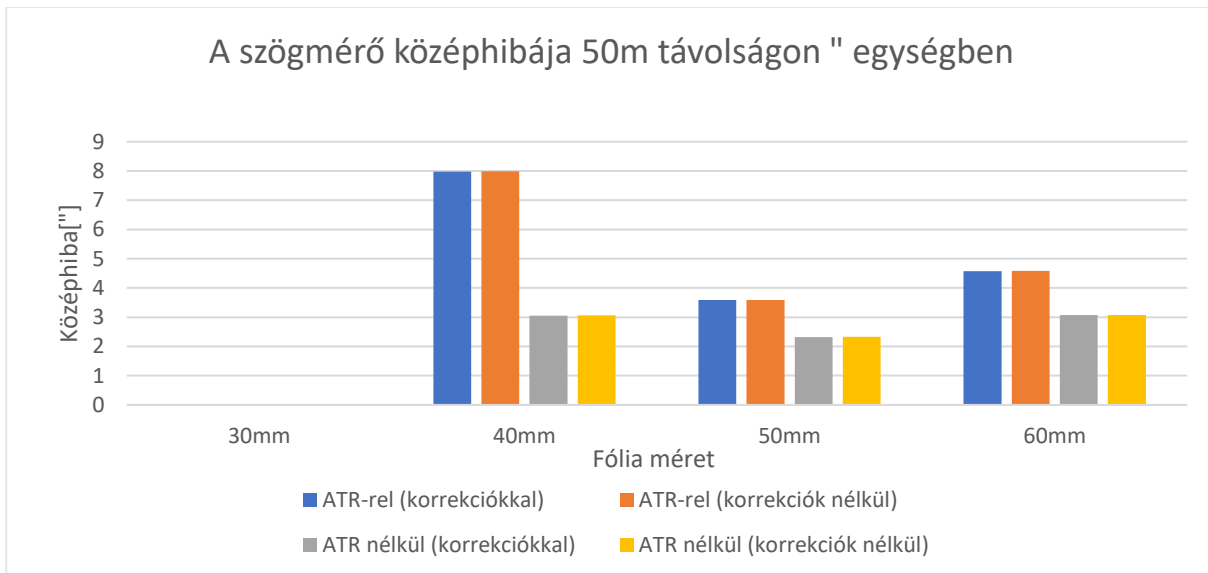
Az ATR-es mérések pontossága jól láthatóan függ a fólia méretétől. 10m-es távolságon 2"-5" (27. ábra), 30m-en 5,5"-13" (28. ábra), 50m-en 3,5"-8" (29. ábra) tartományon belülre esnek a középhibák. Ez azt mutatja, hogy 50m-en pontosabban mért a műszer, mint 30m-en, de nem szabad elfelejteni, hogy 50m-en a 30mm-es visszavérőfóliát ATR-rel már nem tudtam megmérni, aminek középhibáját így nem tartalmazza a 3,5"-8"-es tartomány. Leginkább 30m-es távolságon látható, hogy a nagyobb fólia pontosabb mérést tesz lehetővé az ATR-es mérések esetén, de egy-két kiugró értéktől eltekintve, a többi távolságon is észre lehet venni a pontosság függését a fólia méretétől.



27. ábra: A szögmérő középhibája 10m távolságon



28. ábra: A szögmérő középhibája 30m távolságon



29. ábra: A szögmérő középhibája 50m távolságon

Összeségében látható, hogy a kézi irányzással történt mérések középhibája csaknem minden esetben az ATR-es mérések alatt van.

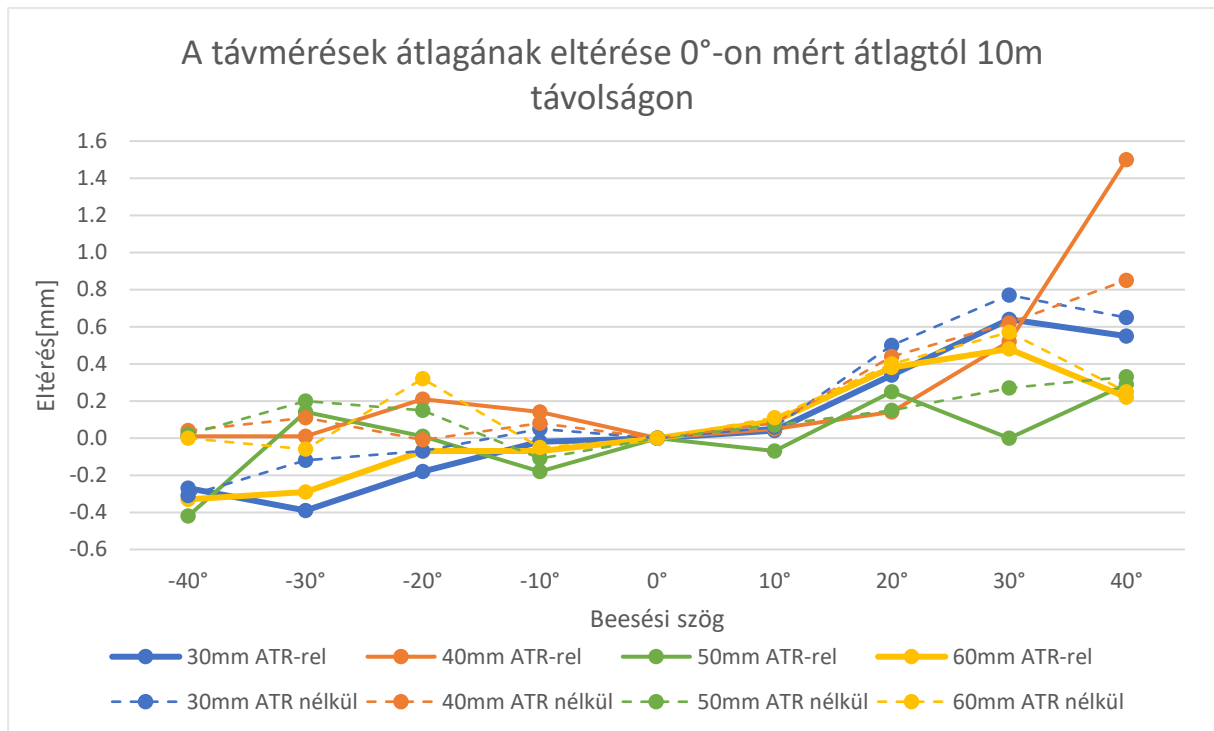
4.2.3 A mérőállomást jellemző pontossági mérőszámok függése a fóliára érkező irányvonal beesésiszögétől

Legvégül a mérőállomást jellemző pontossági mérőszámok függését jellemezném a fóliára érkező irányvonal beesési szögétől. A 3.4 fejezetben leírtak szerint, ezt a vizsgálatot 50m távolságon nem tudtam elvégezni. A táv-és szögmérésre kapott várható értékeket és szórásokat a beesési szög függvényében a 7. táblázat tartalmazza.

		Átlag (távolság)[mm]																	
távolság	pozíció	-40°		-30°		-20°		-10°		0°		10°		20°		30°		40°	
		ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	fólia	-0.27	-0.31	-0.39	-0.12	-0.18	-0.07	-0.02	0.05	0.00	0.00	0.04	0.06	0.34	0.50	0.64	0.77	0.55	0.65
	30mm	0.01	0.04	0.01	0.11	0.21	-0.01	0.14	0.08	0.00	0.00	0.05	0.08	0.14	0.44	0.52	0.62	1.50	0.85
	40mm	-0.42	0.02	0.14	0.20	0.01	0.15	-0.18	-0.11	0.00	-0.07	0.07	0.00	0.25	0.15	0.00	0.27	0.29	0.33
	50mm	-0.33	0.00	-0.29	-0.06	-0.07	0.32	-0.07	-0.05	0.00	0.00	0.09	0.11	0.38	0.40	0.48	0.57	0.22	0.25
	60mm	-	-	-	-	0.18	-0.04	0.41	0.00	0.00	0.00	0.22	-0.09	0.31	0.29	-	-	-	-
	40mm	-	-	-	-	0.74	0.70	0.37	0.37	0.00	0.00	0.18	0.16	0.30	0.52	-	-	-	-
30m	fólia	-	-	-	-	0.46	0.45	0.23	0.23	0.00	0.23	0.11	0.15	0.15	0.07	-	-	-	-
	30mm	-	-	-	-	0.09	0.10	-0.19	-0.08	0.00	-0.35	-0.12	-0.12	-0.10	-	-	-	-	
	40mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	50mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Középhehiba (távolság)[mm]																			
távolság	pozíció	-40°		-30°		-20°		-10°		0°		10°		20°		30°		40°	
		ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	fólia	0.15	0.11	0.11	0.10	0.12	0.12	0.13	0.08	0.11	0.11	0.12	0.09	0.15	0.09	0.13	0.09	0.09	0.14
	30mm	0.14	0.13	0.12	0.12	0.13	0.05	0.09	0.10	0.12	0.09	0.10	0.05	0.14	0.09	0.14	0.13	0.19	0.13
	40mm	0.09	0.14	0.23	0.15	0.08	0.11	0.08	0.08	0.07	0.12	0.17	0.08	0.13	0.12	0.07	0.10	0.05	0.12
	50mm	0.12	0.12	0.08	0.11	0.19	0.15	0.12	0.11	0.09	0.07	0.10	0.09	0.16	0.10	0.12	0.08	0.14	0.14
	60mm	-	-	-	-	0.14	0.07	0.14	0.09	0.08	0.10	0.25	0.14	0.12	0.12	-	-	-	-
	40mm	-	-	-	-	0.11	0.11	0.07	0.10	0.08	0.06	0.12	0.08	0.09	0.07	-	-	-	-
30m	fólia	-	-	-	-	0.08	0.11	0.11	0.05	0.11	0.08	0.10	0.13	0.07	-	-	-	-	
	30mm	-	-	-	-	0.12	0.90	0.09	0.11	0.07	0.14	0.08	0.12	0.06	0.07	-	-	-	
	40mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	50mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Átlag (irányszög)[°]																			
távolság	pozíció	-40°		-30°		-20°		-10°		0°		10°		20°		30°		40°	
		ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	fólia	-0.62	4.33	-3.25	3.20	-6.73	3.81	-7.62	-0.87	0.00	-0.92	-1.43	-1.38	-2.02	-1.94	-4.28	-2.20	54.19	
	30mm	25.66	7.65	16.41	5.65	17.09	5.17	14.47	2.67	0.00	7.08	-2.01	3.52	-4.06	59.62	-3.83	55.55	54.28	
	40mm	-6.35	2.61	-1.52	0.86	1.46	-0.04	2.02	0.38	0.00	-6.24	-0.17	-4.03	-0.90	-1.88	-2.98	-0.14	-5.20	
	50mm	12.88	4.50	18.40	3.73	15.10	1.05	7.55	1.39	0.00	5.13	-3.62	7.65	-5.40	9.90	-4.36	13.57	-7.75	
	60mm	-	-	-	-	6.76	-0.01	7.62	1.14	0.00	0.60	0.78	6.21	0.21	-	-	-	-	
	40mm	-	-	-	-	0.81	-0.98	6.66	-1.13	0.00	5.55	-2.45	1.39	-2.86	-	-	-	-	
30m	fólia	-	-	-	-	2.71	0.07	3.35	-0.35	0.00	-0.98	0.45	0.77	-1.23	-	-	-	-	
	30mm	-	-	-	-	-4.97	0.29	-2.96	-0.68	0.00	1.08	-1.46	-3.91	-2.18	-	-	-	-	
	40mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	50mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Középhehiba (irányszög)[°]																			
távolság	pozíció	-40°		-30°		-20°		-10°		0°		10°		20°		30°		40°	
		ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	fólia	1.62	1.72	1.04	1.14	0.89	1.65	0.79	1.61	0.69	1.57	1.27	0.53	1.64	0.32	1.72	1.18	1.54	
	30mm	0.77	2.52	0.61	1.84	0.80	1.86	0.81	1.88	0.86	1.36	0.90	1.67	0.49	2.28	0.51	2.21	0.93	
	40mm	0.63	1.36	0.50	2.11	0.58	1.78	0.84	1.51	0.35	1.13	0.30	1.59	0.64	1.69	0.45	1.49	0.47	
	50mm	0.69	1.59	1.03	2.04	1.10	1.47	0.83	1.81	0.78	1.18	0.90	1.16	0.91	1.72	0.72	1.82	0.66	
	60mm	-	-	-	-	1.48	1.40	1.81	1.36	0.53	1.60	2.30	1.41	2.41	1.62	-	-	-	
	40mm	-	-	-	-	1.14	3.26	0.72	4.47	1.14	5.93	1.50	3.64	0.82	4.02	-	-	-	
30m	fólia	-	-	-	-	1.01	2.52	0.51	1.82	1.22	1.98	1.03	2.22	0.85	2.13	-	-	-	
	30mm	-	-	-	-	3.49	1.23	1.47	1.95	0.82	1.73	2.08	2.10	1.61	2.43	-	-	-	
	40mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	50mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
	60mm	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

7. táblázat: A táv- és szögmérésekre kapott várható értékek

A táblázatban a várható értékek estében nem a tényleges értéket tüntettem fel, hanem a 0°-on (0°-os beesésnél) kapott várható értéktől való **eltérésüket**. Ezért a 0°-on kapott várható értékek 0 (önmaguktól való eltérése). Ezen szemléltetés mellett azért döntöttem, mert az adatok sokaságát mind táblázatban, mind diagrammokon könnyebb így átlátni, továbbá az adatok értelmezése is az egymással való összehasonlítással történik.



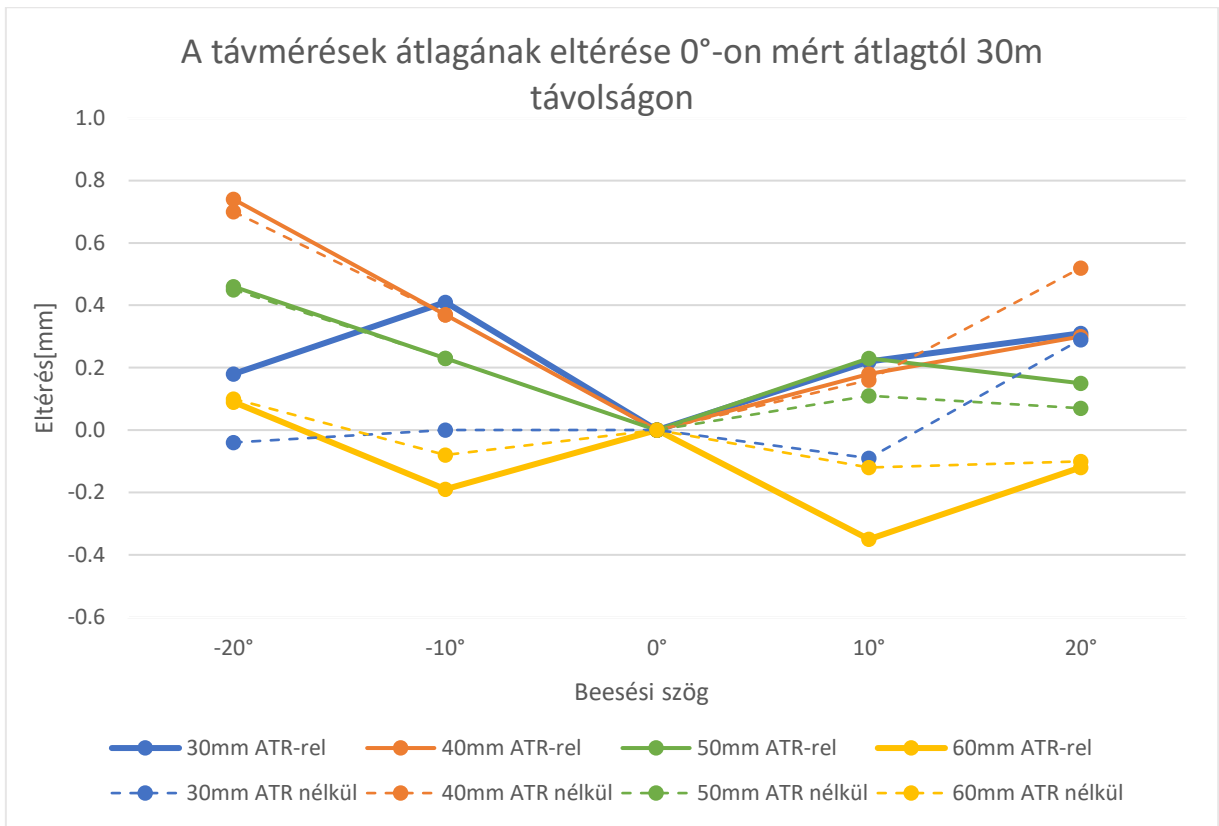
30. ábra: A távmérések átlagának eltérése 0°-on mért átlagtól 10m távolságon

A 30. ábra görbéin egy középpontosan szimmetrikus jelleg látható. A 0-tól való eltérések a 0,6mm-t is elérhetik, ez a $\pm 0,2$ mm-es számított középhibát figyelembevéve (4.2.1-ben), már jelentős. Ez arra utal, hogy az irányzott pont nem esik egybe a forgatás tengelyével. Lehetséges okok:

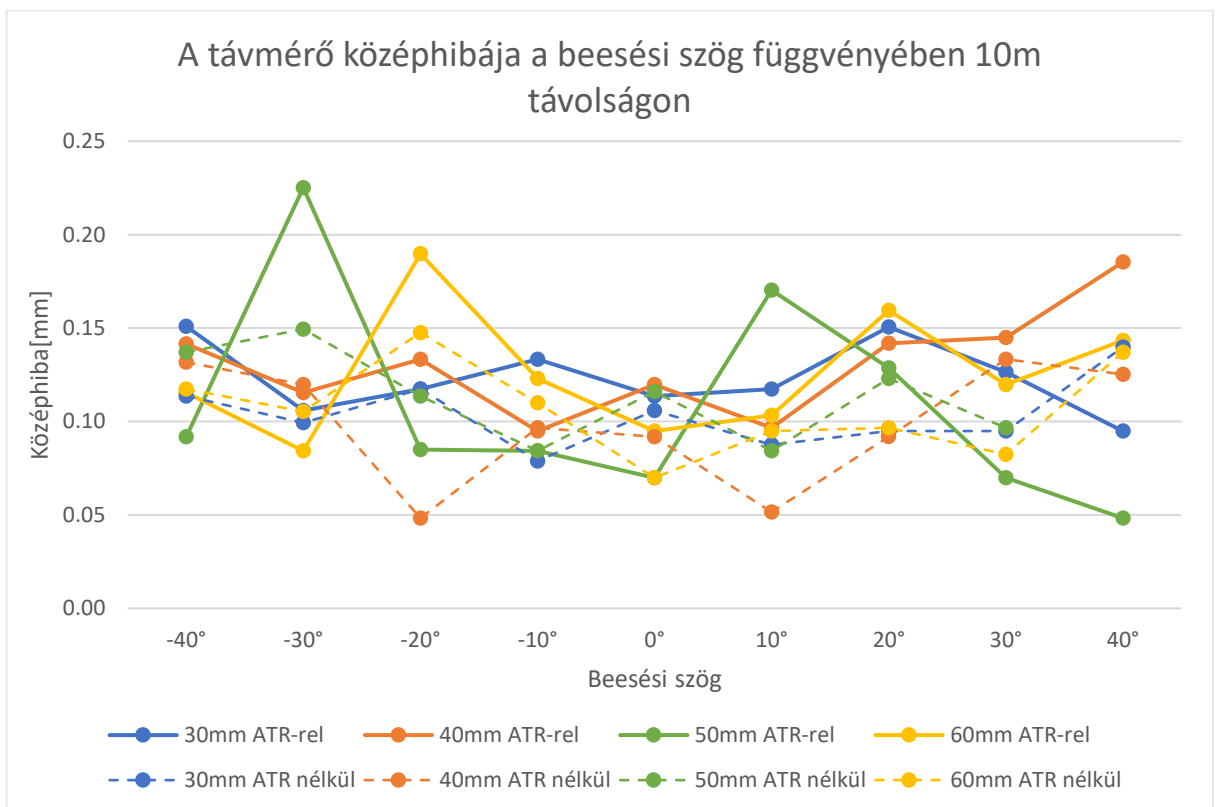
- a jeltárcsa nem középpontosan forog
- a fólia nem tökéletesen jeltárcsa közepére lett ragasztva
- az irányzás nem a fólia közepére történt, megbízhatósága nem tökéletes (ATR-rel és anélkül is)

Ha ugyanezen vizsgálat 30m-en számított értékeit nézzük (31. ábra), akkor nem láthatjuk az előbb említett szimmetriát, az értékek nem mutatnak szabályos eltéréseket. Ennek oka lehet, hogy a vizsgált forgatási tartomány jóval kisebb a 10m-en alkalmazott tartománynál. Így a két ábrát figyelembe véve nem vonhatunk messze menő következtetéseket, vagyis a vizsgálat alapján nem mutatható ki, hogy a távméréseket a fólia forgatásával bármiféle szabályos hiba terhelné. A gyanút, hogy a távméréseinket a beesési szög függvényében

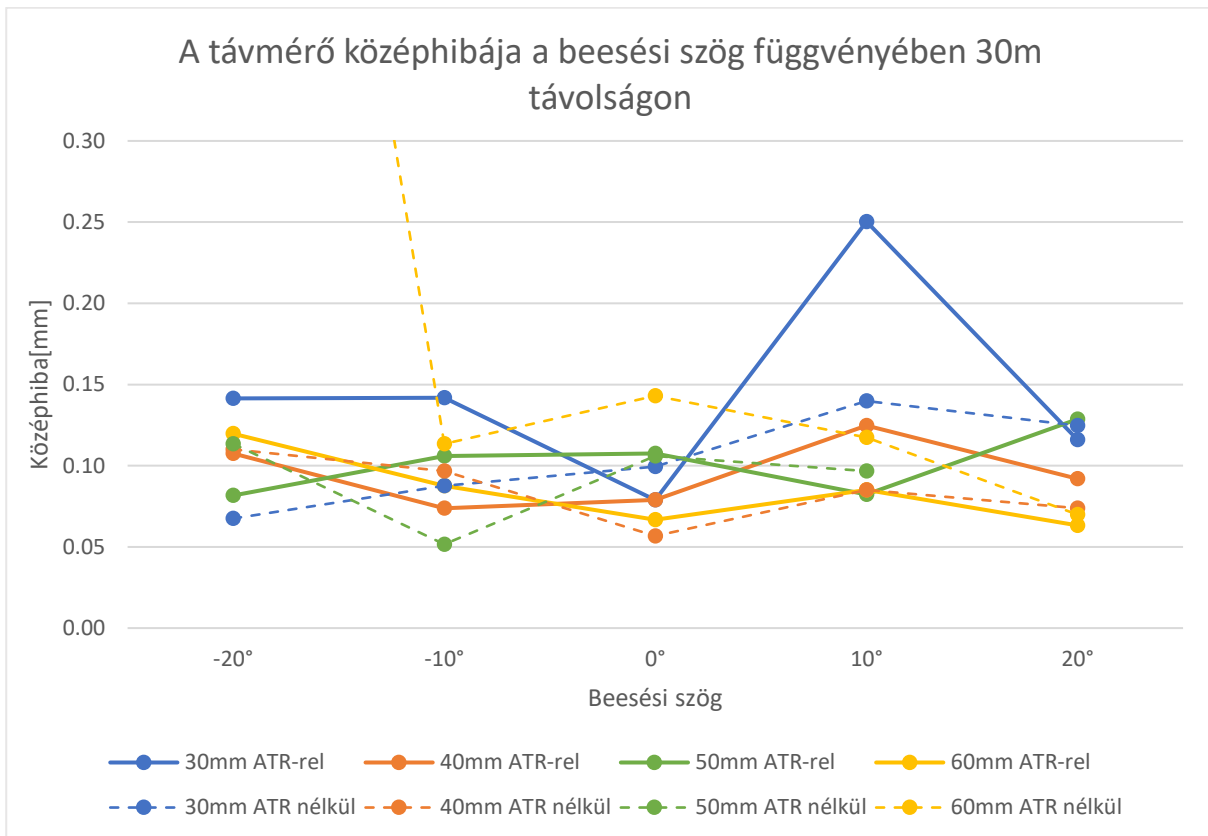
szabályos hiba terheli, fennmarad, de a 30m-en számított eredmények ezt nem támasztják alá. Ennek kimutatására újabb vizsgálatokat kell végezni.



31. ábra: A távmérések átlagának eltérése 0°-on mért átlagtól 30m távolságon



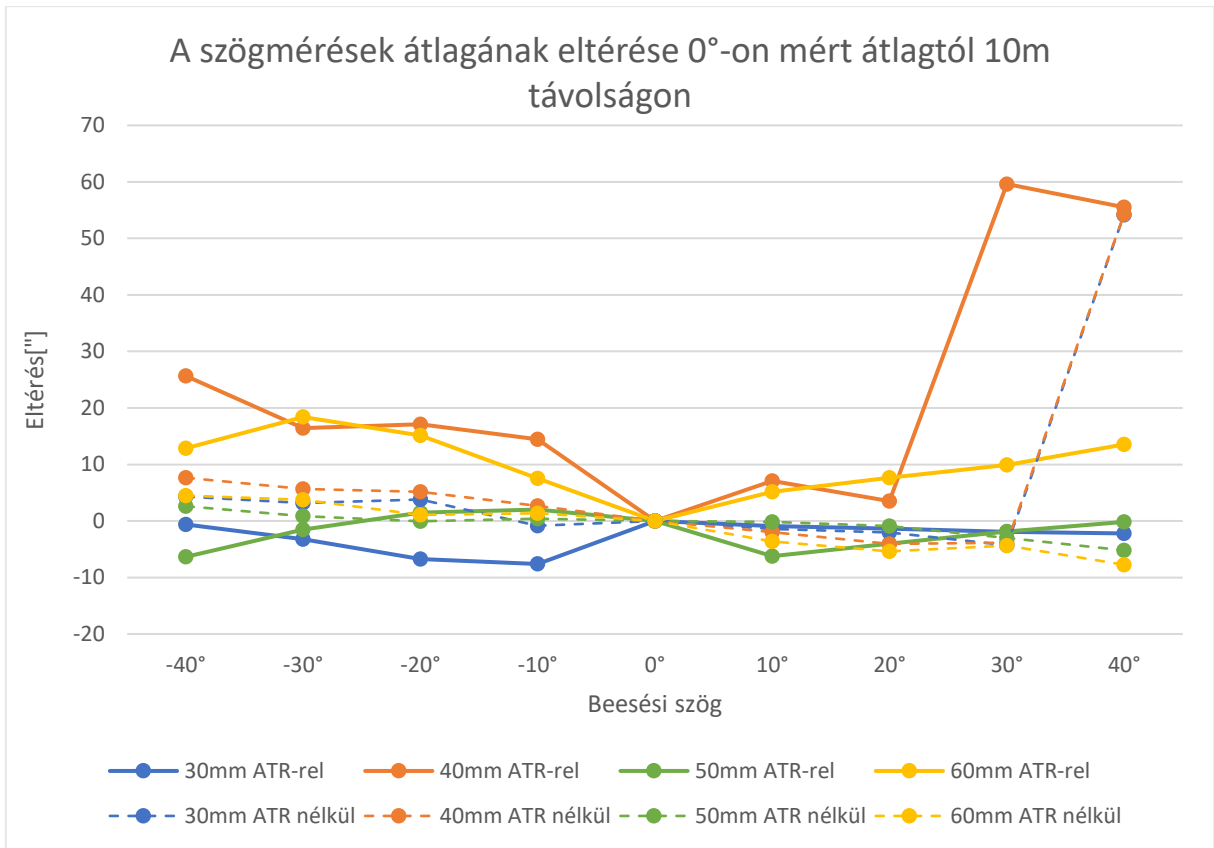
32. ábra: A távmérő középhibája a beesési szög függvényében 10m távolságon



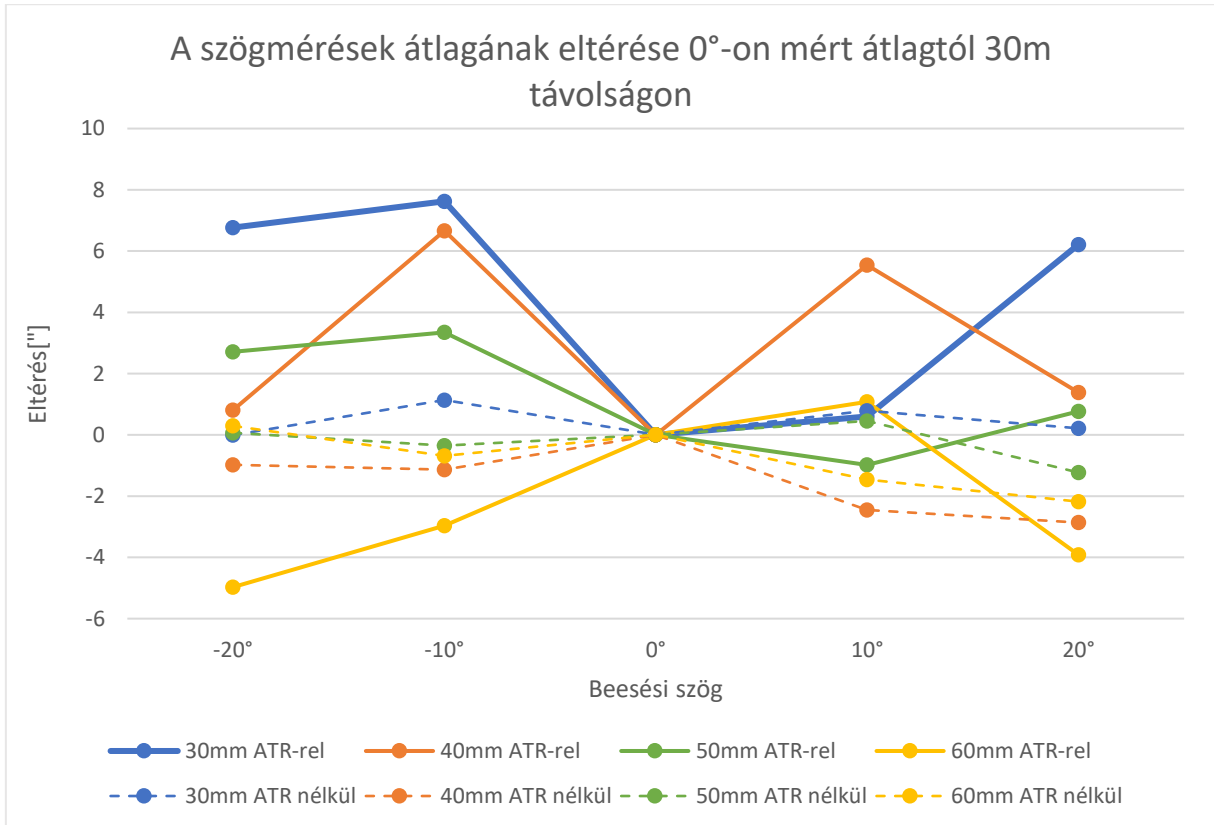
33. ábra: A távmérő középhibája a beesési szög függvényében 30m távolságon

A távmérések a beesési szögtől függő középhibáit 10m-en a 32. ábra, 30m-en a 33. ábra tartalmazza. A 10m-en számított középhibák esetén, 0°-os beesési szögnél látható egyfajta minimum, de ennek mértéke gyakorlati szempontból elhanyagolható, és a 30m-en számított eredmények ezt sem támasztják alá. Az ábrán azt is láthatjuk, hogy az ATR-es és ATR nélküli mérések középhibái nem mutatnak jelentős eltéréseket, és összhangban vannak a 4.2.1 szakaszban leírtakkal, vagyis távmérés esetén, az általam meghatározott középhibák nem mutatnak szignifikáns eltéréseket.

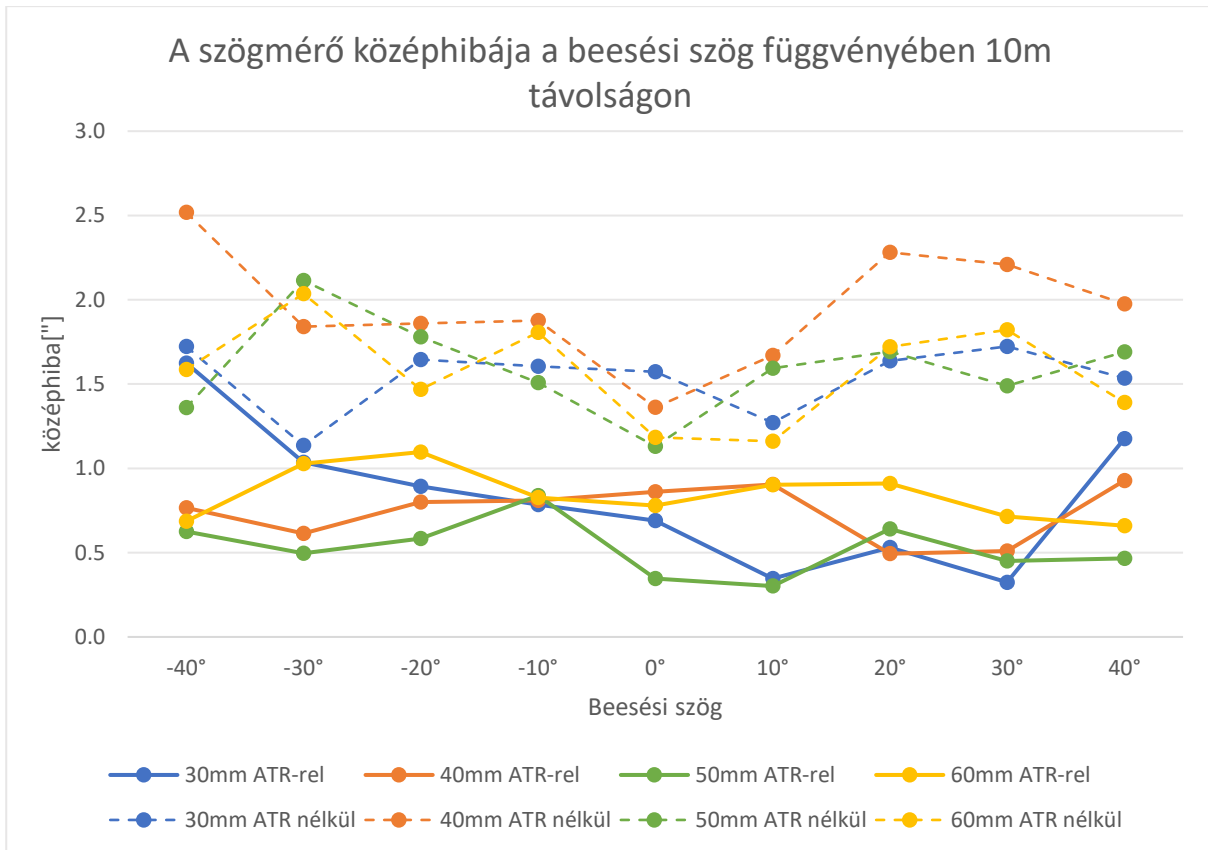
A szögmérések várható értékében jelentős eltérések mutatkoznak (34. és 35. ábra), különösképpen az ATR-es mérések estében. Ennek ellenére a középhibák az ATR-es mérések estében kedvezőbbek (36. és 37. ábra). Ez azt mutatja, hogy az ATR-es mérések pontossága nagyobb, ám a kézi irányzások megbízhatóbbak, így a kézi irányzás pontosan a fólia közepére történt nagy szórással (emberi tényező), míg az ATR-es mérések szórása kicsi, de nem a fólia közepére történtek, így az ATR-rel irányzott pont a forgatáskor elmozdul, a középpont helyben marad.



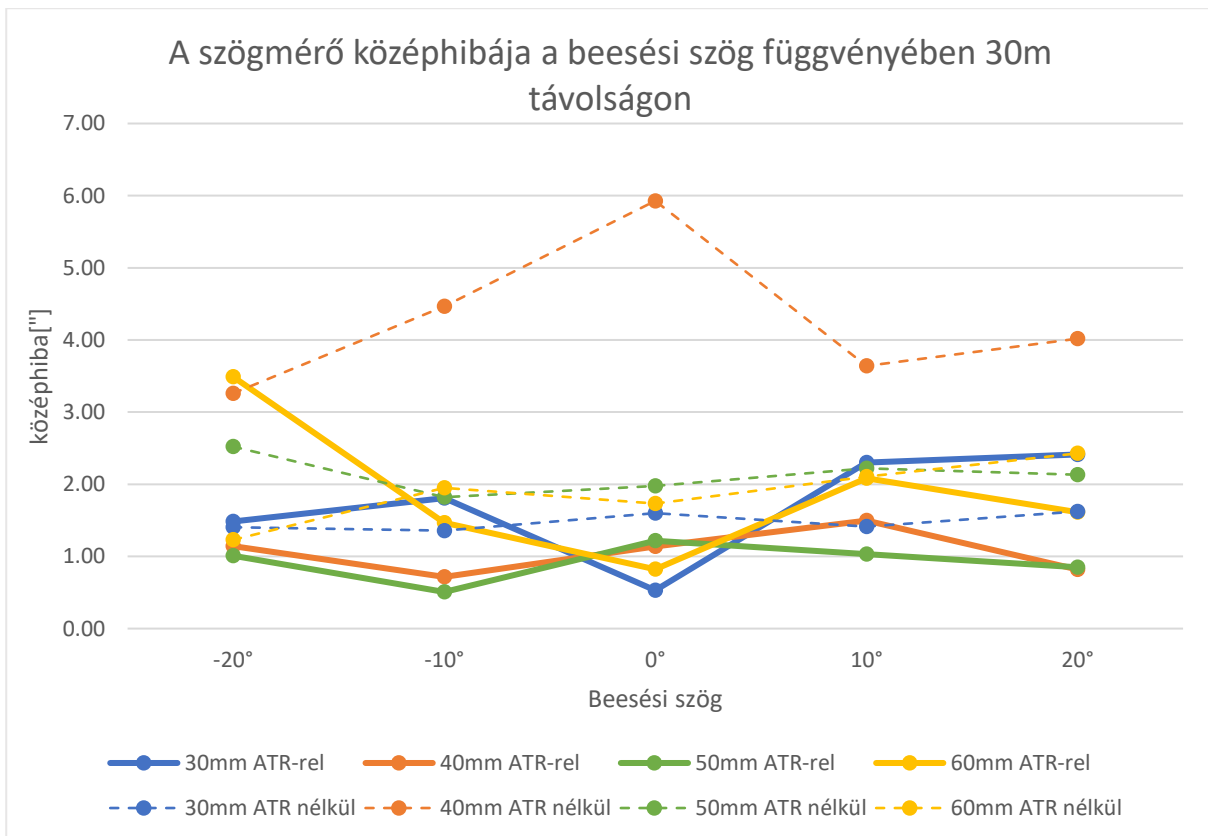
34. ábra: A szögmérések átlagának eltérése 0°-on mért átlagtól 10m távolságon



35. ábra: A szögmérések átlagának eltérése 0°-on mért átlagtól 30m távolságon



36. ábra: A szögmérő középhibája a beesési szög függvényében 10m távolságon



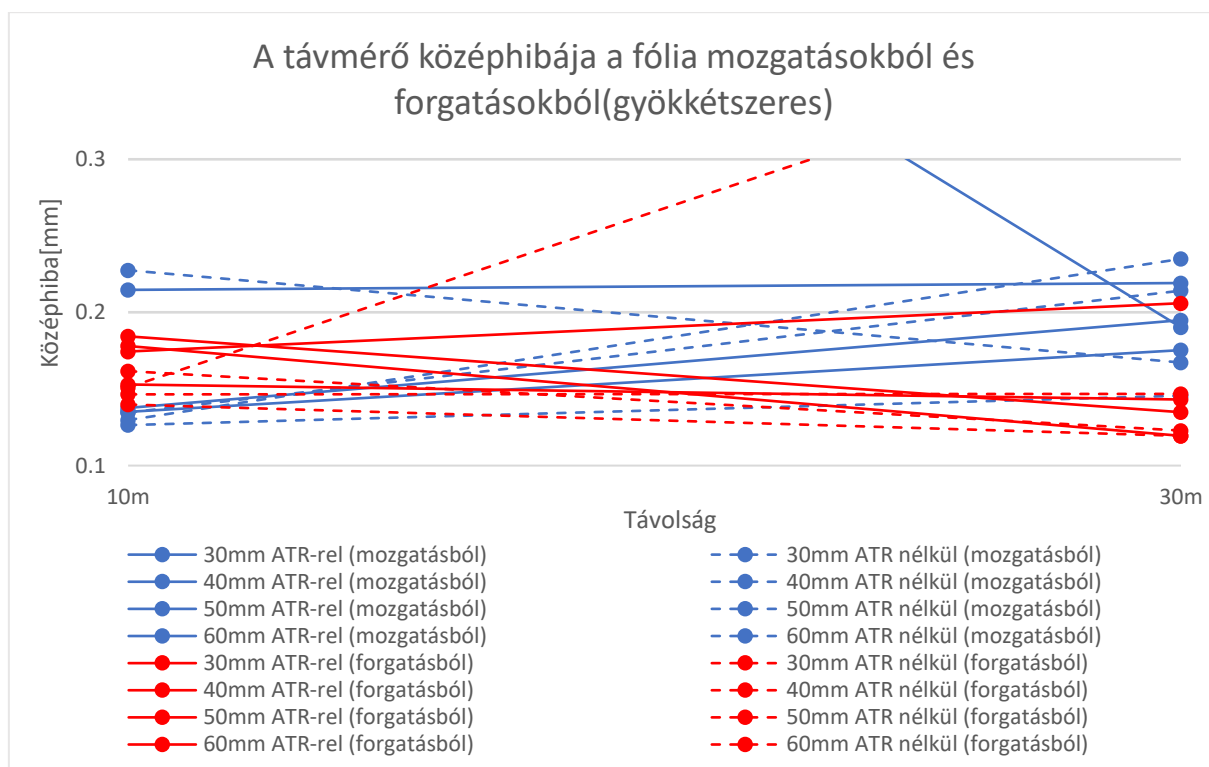
37. ábra: A szögmérő középhibája a beesési szög függvényében 30m távolságon

4.3 AZ EREDMÉNYEK CÉLSZERŰ FELHASZNÁLÁSA (AJÁNLÁS)

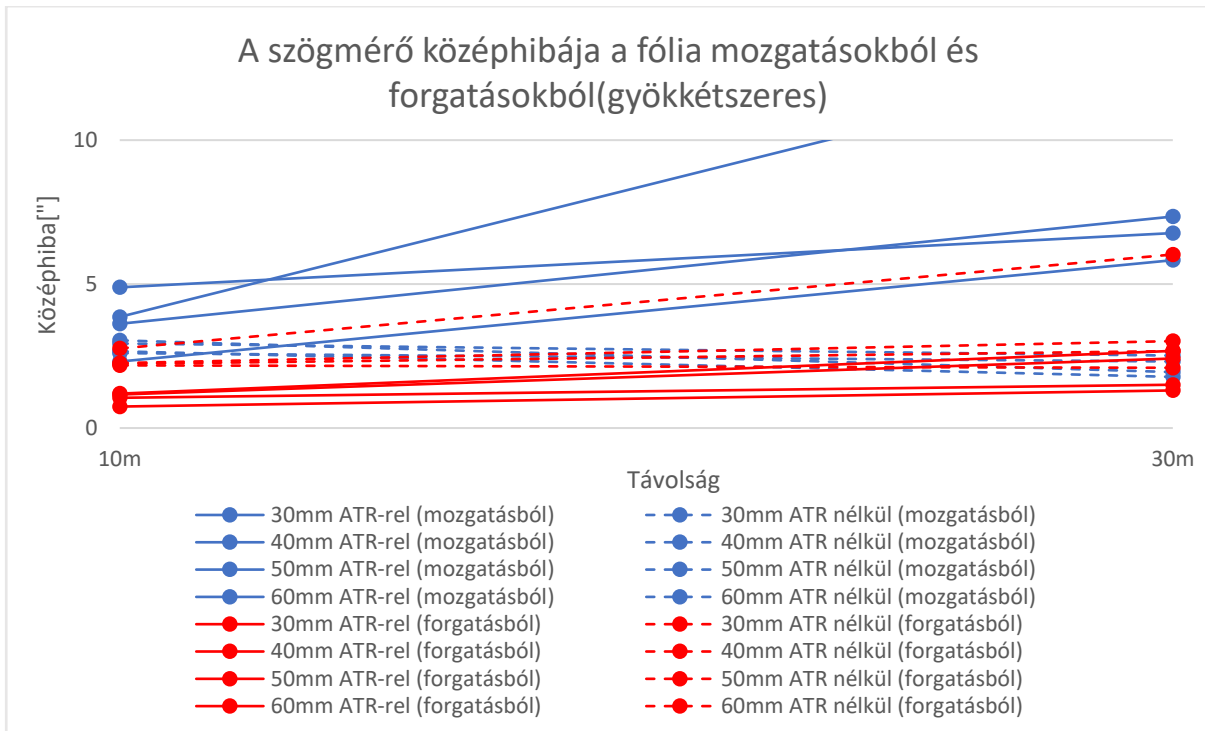
A számítások során némely esetben inkonzisztens eredmények születtek. Ezekre az ellentmondásokra, ebben a fejezetben adok magyarázatokat, továbbá a kapott eredmények alapján a különböző koordinátaszámítási módszereket (pontkapcsolások) veszem sorra, hogy melyiket a legcélszerűbb alkalmazni.

4.3.1 A kapott eredmények közötti inkonzisztencia feloldása

A legnagyobb eltéréseket a fólia eltoláskor és forgatáskor meghatározott középhibák között láthatjuk (táv mérés és szögmérés esetén egyaránt). Az egyszerű szórás/középhiba számításokor (forgatás) egyetlen távolságra és irányértékre határoztuk meg a középhibát. Ezzel szemben a fólia eltolásakor, a távolság különbségre, és törésszögekre határoztuk meg a középhibákat. Ez jelentős eltérés, ugyanis a hibaterjedés törvénye szerint a távolság különbség, és a törésszög középhibája gyökkétszerese az egyetlen távolság és irányérték középhibájának. Ha ennek figyelembevételével ábrázoljuk a táv- és szögmérések középhibáit (38. és 39. ábra), akkor láthatjuk, hogy középhibákban már nem található akkora eltérés. Egy-két kiugró értéket figyelmen kívül hagyva, a középhibák jó összhangban vannak egymással. Az ábrákon az áttekinthetőség kedvéért szándékosan csak két szintet használtam, a különböző fóliák közötti különbséget nem kívánom kiemelni.



38. ábra: A távmérő középhibája a fólia mozgatóból és forgatásból (gyökkétszeres érték)



39. ábra: A szögmérő középhibája a fólia mozgatókészségek és forgatásokból (gyökkétszeres érték)

Mindezek alapján összeállítottam az általam javasolt középhibákat mind távmérés, mind szögmérés esetében. Azonban fontos megjegyezni, hogy ezek csak visszaverőfóliára érvényesek, valamint meghatározásukhoz szükséges mérések pillérrel történtek, így a műszer mozdulatlanúsága biztosított volt. A mérések tervezésekor ezeket a tényezőket figyelembe kell venni.

A távolságkülönbségek középhibáját így $\pm 0,3\text{mm}$ -re javaslom felvenni mind az ATR-es, mind a kézi irányzással végzett mérésekhez. Ez az érték kb. 30%-kal meghaladja az általam relevánsnak tartott $\pm 0,23\text{mm}$ -es legmagasabb értéket, melyet 30m-en a 30mm-es fóliára mértem ATR nélkül. Az általam irrelevánsnak tartott értékeket 50m-en 40mm-es, illetve 10m-en 60mm-es fólián mértem. Ezek értéke $\pm 0,31\text{mm}$ és $\pm 0,61\text{mm}$. Mivel az általam vizsgált tartományban távmérések értéke csak minimálisan függ a távolságtól, és a fólia méretét sem találok befolyásoló tényezőnek, nem tartom szükségesnek minden távolságra és fóliaméretre különböző középhiba alkalmazását.

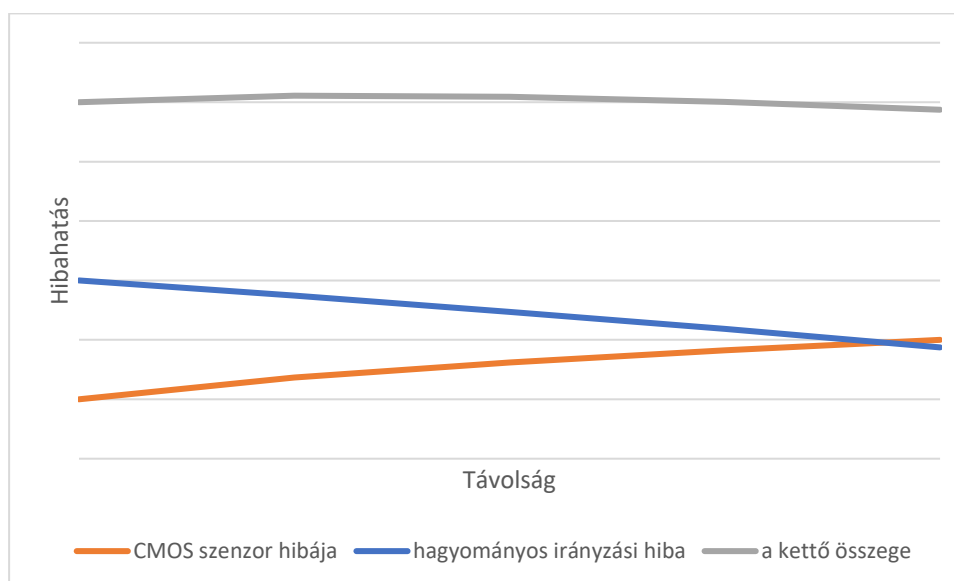
A távolságkülönbségekkel ellentétben, a törésszögek középhibája erősen függ mind a távolságtól, mind a fólia mérettől, mind pedig az ATR alkalmazásától. A kapott eredmények között igen nehéz bármiféle szabályt találni. Mindenesetre, a gyártó által meghatározott a priori középhiba a fent leírt körülmények között nem elérhető. Ezt alátámasztandó elvégeztem egy kétszer 500 mérésből álló mintavételt egy 60mm-es fóliára és egy Leica miniprizmára, melyeket a mérőállomástól kb. 26m-re helyeztem el. Ennek a mérésnek az eredményeit a 8. táblázat tartalmazza.

	Középhiba		
	távolság[mm]	irányérték["]	zenitszög["]
fólia(60mm)	0.12	3.3	5.0
prizma(mini)	0.11	1.2	0.9

8. táblázat: 500 elemű mintából számított középhibák fóliára (60mm) és prizma (Leica mini) ATR használatával

Itt is fontos megjegyezni, hogy ezek a középhibák egyetlen irányra vonatkoznak, nem törésszögre, ezért ezek megszorozandók gyökjettővel, mielőtt összehasonlítanánk őket a fólia mozgatósból kapott középhibákkal.

Továbbá a távolság függvényében várt arányos összefüggés nem mutatható ki az ATR-es mérések esetében, sőt középhibákra illesztett görbének 30m körül maximuma van, majd el kezd csökkenni (26. ábra). Feltételezésem szerint ennek oka, hogy az ATR-es irányzás alapvetően két összetevőből áll. Az egyik a kézi irányzáskor is fellépő irányzási hiba. Ez a távolság növelésével csökken. Az ATR működéséből adódóan fellép egy a CMOS kamera által készített kép készítése, illetve feldolgozása során fellépő hiba. Ez a hiba a távolsággal nő. A két hiba nem azonos mértékű és a kettő egymásra halmozódásából adódik az általunk tapasztalt görbe (40. ábra). Szintén fontos megjegyezni, hogy ez csupán feltételezés. Ennek bizonyítása a jelenlegi adatokkal nem lehetséges. Ehhez több távolságon el kell végezni az eddigi méréseket.



40. ábra: A szögmérés középhibáiban tapasztalható anomália magyarázata

Ennek tükrében a 6. táblázatban látható értékekről elmondhatjuk, hogy nagyságrendileg megfelelnek a valóságnak. Az általam javasolt, némiképp szubjektíven meghatározott középhibákat a 9. táblázat tartalmazza.

Az általam javasolt, szubjektíven meghatározott középhibák törésszögekre ["]								
	30mm		40mm		50mm		60mm	
	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül	ATR	ATR nélkül
10m	5	4	5	4	5	4	4	4
30m	13	4	8	4	8	4	6	4
50m	-	4	9	4	9	4	6	4

9. táblázat: Az általam javasolt, szubjektíven meghatározott középhibák törésszögekre

Az 50m-en 30mm-es fóliára nem végeztem méréseket, de kézi irányzással lehetséges a mérés elvégzése. Mivel az ATR-nélküli mérések középbáinak egyike sem közelítette meg $\pm 4''$ -es értéket, így ezt a középhibát alkalmazhatjuk ebben az esetben is.

Nem meglepő módon láthatjuk, hogy ATR használatakor a szögmérés pontossága (vizsgálataink alapján) nagy mértékben függ, a visszaverőfólia méretétől, vagyis minél nagyobb pontosság elérése érdekében érdemes minél nagyobb fóliát alkalmazni. Nagy pontosság eléréséhez nem elegendő a kis fóliák alkalmazása. Azonban szélső pontosság eléréséhez nem alkalmas a visszaverőfólia, így, ha a pontossági igények nagyobbak (leginkább szögmérés tekintetében), akkor a mérésekhez prizmat kell alkalmaznunk.

4.3.2 Különböző koordináta meghatározási módszerek (pontkapcsolások)

Földmérő körökben az optikai távmérők elterjedése óta vita zajlik a poláris pont, és az előmetszés pontosságáról. Némelyek úgy vélik, hogy a távmérés pontossága nem éri el a kívánt mértéket, így az előmetszés pontosabb a poláris pont meghatározásánál, mások a polárispont módszerét részesítik előnyben. Ennek a kérdésnek a nyomában, az általam meghatározott középhibákkal, a hibaterjedés törvényét használva meghatároztam a két módszer pontosságát. Természetesen ez a vizsgálat nem ad minden eshetőségre választ, csak a visszaverőfóliák alkalmazásakor, illetve az ehhez hasonló esetekben nyújt támpontot.

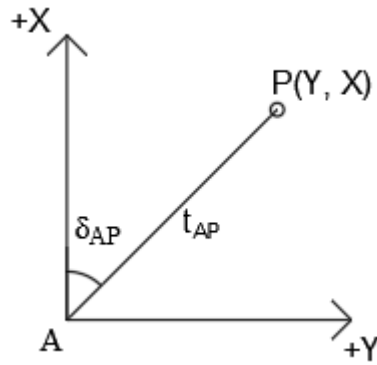
A módszereket minden esetben 2 dimenzióban vizsgálom, a magassági mérésektől eltekintek.

4.3.2.1 Poláris pont

A módszer a lehető legegyszerűbb. Ismert koordinátájú és tájékozású állomással mérünk a meghatározandó pontra irányértéket és távolságot. Ezen mennyiségek alapján I. geodéziai fő-feladattal meghatározzuk a pont koordinátáit (41. ábra).

A pontok Y és X koordinátáinak hibáját, a hibaterjedés törvényével, a következők szerint számítottam (Detrekői, 1991).

Feltételezem, hogy az álláspont koordinátáit nem terheli hiba és a tájékozási szög 0° , az ismert pont az origóban helyezkedik el. Ekkor a pont Y és X koordinátái a 16. és 17. képlettel számíthatók.



41. ábra: Polárispont számítás

$$Y_P = t_{AP} * \sin(\delta_{AP}) \quad (16. \text{ képlet})$$

$$X_P = t_{AP} * \cos(\delta_{AP}) \quad (17. \text{ képlet})$$

A hibaterjedéshez szükséges parciális deriváltak:

$$\frac{\partial Y_P}{\partial t_{AP}} = \sin(\delta_{AP}) \quad (18. \text{ képlet})$$

$$\frac{\partial Y_P}{\partial \delta_{AP}} = t_{AP} * \cos(\delta_{AP}) \quad (19. \text{ képlet})$$

$$\frac{\partial X_P}{\partial t_{AP}} = \cos(\delta_{AP}) \quad (20. \text{ képlet})$$

$$\frac{\partial X_P}{\partial \delta_{AP}} = -t_{AP} * \sin(\delta_{AP}) \quad (21. \text{ képlet})$$

A 18.-21. parciális deriváltakat mátrixba rendezve:

$$F^* = \begin{bmatrix} \frac{\partial Y_P}{\partial t_{AP}} & \frac{\partial Y_P}{\partial \delta_{AP}} \\ \frac{\partial X_P}{\partial t_{AP}} & \frac{\partial X_P}{\partial \delta_{AP}} \end{bmatrix} \quad (22. \text{ képlet})$$

A keresett kovariancia mátrix:

$$M_{YX} = F^* * M_{t\delta} * F \quad (23. \text{ képlet})$$

, ahol

$$M_{t\delta} = \begin{bmatrix} \mu_t^2 & 0 \\ 0 & \mu_\delta^2 \end{bmatrix} \quad (24. \text{ képlet})$$

A pontok jellemzésére a ponthibát alkalmazom. Ez az M_{YX} kovarianciamátrix spurjának a gyökeként határozható meg (Detrekői, 1991).

$$P = \sqrt{Sp M_{YX}} = \sqrt{\mu_Y^2 + \mu_X^2} \quad (25. \text{ képlet})$$

Az alkalmazott középhibák:

$$\mu_t = 0,3mm$$

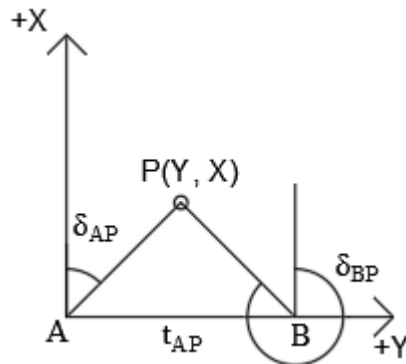
$\mu_\delta = 5", 8", 9"$ rendre 10m, 30m és 50m távolságon,

illetve a gyártó által meghatározott a priori szögmérési hibával is kiszámítom a pontot jellemző pontossági mérőszámokat. A pontok ponthibáját a 10. táblázat tartalmazza.

4.3.2.2 Előmetszés

Ez a módszer azon alapszik, hogy ismert koordinátájú pontokról irányértékeket mérünk az ismeretlen pontra. Így az AP és BP egyenesek metszéspontjaként megkaphatjuk P pont koordinátáit (42. ábra).

A számításokkor feltételezem, hogy az álláspontok koordinátái hibátlanok, és a tájékozási szögek értéke 0° , az A pont az origóban, a B pont az y tengelyen helyezkedik el.



42. ábra: Előmetszés számítás

A pontok Y és X koordinátája:

$$Y_P = \frac{t}{\tan(\delta_{AP}) - \tan(\delta_{BP})} \quad (26. \text{ képlet})$$

$$X_P = \frac{t \cdot \tan(\delta_{AP})}{\tan(\delta_{AP}) - \tan(\delta_{BP})} \quad (27. \text{ képlet})$$

A 26. és 27. képletek alapján a parciális deriváltak:

$$\frac{\partial Y_P}{\partial \delta_{AP}} = - \frac{t}{\cos^2(\delta_{AP}) \cdot \left(\frac{\sin(\delta_{AP})}{\cos(\delta_{AP})} - \frac{\sin(\delta_{BP})}{\cos(\delta_{BP})} \right)^2} \quad (28. \text{ képlet})$$

$$\frac{\partial Y_P}{\partial \delta_{BP}} = \frac{t}{\cos^2(\delta_{BP}) \cdot \left(\frac{\sin(\delta_{AP})}{\cos(\delta_{AP})} - \frac{\sin(\delta_{BP})}{\cos(\delta_{BP})} \right)^2} \quad (29. \text{ képlet})$$

$$\frac{\partial X_P}{\partial \delta_{AP}} = \frac{t}{\cos^2(\delta_{AP}) * \cos^2(\tan(\delta_{AP})) * \left(\frac{\sin(\delta_{AP})}{\cos(\delta_{AP})} - \frac{\sin(\delta_{BP})}{\cos(\delta_{BP})}\right) - \frac{t * \sin(\tan(\delta_{AP}))}{\cos^2(\delta_{AP}) * \cos(\tan(\delta_{AP})) * \left(\frac{\sin(\delta_{AP})}{\cos(\delta_{AP})} - \frac{\sin(\delta_{BP})}{\cos(\delta_{BP})}\right)^2}} \quad (30. \text{ képlet})$$

$$\frac{\partial X_P}{\partial \delta_{BP}} = \frac{t * \sin(\tan(\delta_{AP}))}{\cos^2(\delta_{BP}) * \cos(\tan(\delta_{AP})) * \left(\frac{\sin(\delta_{AP})}{\cos(\delta_{AP})} - \frac{\sin(\delta_{BP})}{\cos(\delta_{BP})}\right)^2} \quad (31. \text{ képlet})$$

A parciális deriváltakat (28.-31. képlet) mátrixba rendezve (22. képlet), és az előző pontban használt középhibákat használva a 23. képlettel megkaphatjuk a pont koordinátáinak kovarianciamátrixát. A ponthibákat 25. képlettel számolhatjuk. A kapott eredményeket a 10. táblázatban láthatjuk.

	Ponthiba[mm]			
	Az általam meghatározott középhibákkal		A műszergyártó által meghatározott a priori középhibával	
távolság	poláris pont	előmetszés	poláris pont	előmetszés
10m	0.39	0.40	0.30	0.08
30m	1.20	1.90	0.33	0.24
50m	2.20	3.56	0.39	0.40

10. táblázat: Poláris ponttal és előmetszéssel meghatározott pont ponthibája

A távmérő középhibáját mindkét esetben $\pm 0,3\text{mm}$ -re vettem fel. A kapott eredmények szerint, ha a méréseket $1''$ -es szögmérési hibával el tudjuk végezni, akkor az előmetszés pontosabb eredményeket ad a vizsgált távolságokon, ám láthatjuk, hogy 50-en már alig van különbség a két módszer között. Ha az általam meghatározott középhibákkal számolunk, 10m felett mindig pontosabb lesz a poláris pont számítás. 10m-en a két módszer majdnem azonos, ám ennél közelebbi méréseket csak kivételes esetekben végzünk.

5 ÖSSZEFOGLALÁS

Vizsgálataim kiterjedtek a szög- és távmérő pontosságának vizsgálatára visszaverőfólia alkalmazásakor, valamint ezek a fóliára eső irányvonal beesési szögétől. A kapott eredmények több kérdést is megválaszoltak, némely esetben meglepő eredményeket kapva. Azonban továbbra is vannak megválaszolatlan kérdések, illetve a vizsgálat elvégzésekor felmerültek további kérdések. A fólia forgatása, bár alátámasztotta a fóliamozgatásból számított középhibákat, nem mutatott egyértelmű összefüggéseket, a mérési eredmények trendszerű változásáról, illetve a középhibákról sem.

A szögmérési középhibák távolságtól való függésére nem kaptunk egyértelmű eredményeket. Az összefüggés létezik, ám az, hogy ez milyen jellegű, és hogy milyen fizikai tényezőktől függ, az általam elvégzett mérésekből nem deríthető ki. Ennek meghatározására további méréseket szükséges végezni.

A poláris pont, előmetszés pontosságának vizsgálata által kapott eredmények nem okoztak túlnagy meglepetést. Ellenben érdekes kérdés, hogy a további pontkapcsolási módszerek milyen pontosságot tesznek lehetővé, tekintve, hogy a távmérési középhibája alacsonyabb, a szögmérése pedig magasabb a vártnál.

6 IRODALOMJEGYZÉK

Leica Geosystems (2012): Leica TPS1200+ User Manual

Leica Geosystems (2007): Leica TPS1200+ White Paper

Renishaw: Renishaw XL-80 user guide

Piri D (2015): Mérőállomás célkövető üzemmódjának pontossági vizsgálata

Siki Z: ULYXES (<https://github.com/zsiki/ulyxes>)

Detrekői Á (1991): Kiegyenlítő számítások

Krauter A (2002): Geodézia

Tasi Miklós (2012): Vállalatirányítási rendszerek
(http://www.tankonyvtar.hu/hu/tartalom/tamop412A/2010-0017_19_valliranyitasi_rendszerek/ch03s05.html)

7 KÖSZÖNETNYILVÁNÍTÁS

Első sorban szeretném megköszönni Dr. Égető Csabának a segítségét, akinek tanácsai és szakmai tapasztalata nagyon sokat segített a mérések tervezésében és elvégzésében, a szakirodalom felkutatásában, a számítások elvégzésében, valamint, hogy a szabadidejét nem sajnálva, a leghetlenebb időpontokban is lehetővé tette a mérések elvégzését.

Továbbá szeretném megköszönni a stáb tagjainak; Kegyes Lillának, Maksa Reginának és Molnár Andreának a segítséget, akik sokszor a nyári, hétvégi és éjszakai időpontokat is vállalták, hogy elvégezhessük a méréseket. Nélkülük ez a dolgozat nem készülhetett volna el.

Végül, de nem utolsó sorban szeretnék köszönetet mondani Dr. Tóth Gyulának, aki kiegyenlítés világában adott sok segítséget. Ez a dolgozatban nem jelenik meg teljes egészében, mivel a regressziós egyenessel kapcsolatos terveim füstbe mentek, de ezekre a próbálgatásokra szükség volt a végeredmény kialakulásához.

8 MELLÉKLETEK

```
classdef Read
    %base class for read file
    properties
        file;
        fID;
        rows;
        measures;
    end
    methods
        %constructor
        function this = Read(file)
            this.fID = fopen(file);
            rows = {};
            while ~feof(this.fID)
                rows = [rows;{fgetl(this.fID)}];
            end
            this.rows = rows;
            fclose(this.fID);
        end%end of constructor
    end
end
```

1. forráskód: read.m (Read osztály)

```
classdef ReadGeo < Read

    methods
        function this = ReadGeo(file)
            this = this@Read(file);

            [r, ~] = size(this.rows);
            for i = 1:r

                this = readRow(this, this.rows(i,1));
            end
        end

        function this = readRow(this, row)

            pattern = '\{[0-9]{1,3} [a-zA-z0-9\. \[:-]*\}';
            blockPattern = '\{(?<key>[0-9]{1,3}) (?<value>[a-zA-z0-9\. \[:-]*\})\}';

            blocks = regexp(char(row), pattern, 'match');
            [~, c] = size(blocks);
            for i = 1:c
                block = regexp(char(blocks(1,i)), blockPattern, 'names');
                switch str2num(block.key)
                    case 7
                        %direction angle
                        measure.direction = str2double(block.value);
                    case 8
```

```

        %zenith angle
        measure.zenith = str2double(block.value);
    case 9
        %distance
        measure.distance = str2double(block.value);
    case 5
        %point ID
        measure.ID = char(block.value);
    otherwise
        %disp(class(str2num(block.value)));
    end
end

end

if (exist('measure', 'var'))
    this.measures = [this.measures; measure];
end

end%end of readRow

function measures = getMeasuresInMatrix(this)
    measures = [];
    [r, ~] = size(this.measures);
    for i = 1:r
        measures = [measures; this.measures.distance,
this.measures.direction, this.measures.zenith];
    end
end%end of getMeasuresInMatrix
end
end
end

```

2. forráskód: readgeo.m (ReadGeo osztály)

```

classdef ReadRtl < Read
    properties
        targets;
        runs;
    end
    methods
        function this = ReadRtl(file)
            this = this@Read(file);

            %read targets
            [r, ~] = size(this.rows);
            for i = 1:r
                %disp(this.rows(i,1));
                if (strcmp(char(this.rows(i,1)), 'Targets :'))
                    %read targets
                    this = readTargets(this, this.rows(i+1,1));

                elseif (strcmp(char(this.rows(i,1)), 'RUNS::'))
                    %number of runs
                    runPattern = 'Run-count:(?<runs>[0-9]*)$';
                    runs = regexp(char(this.rows(i+1,1)), runPattern,
'names');

                    this.runs = str2num(runs.runs);
                elseif (strcmp(char(this.rows(i,1)), 'Run Target Data:'))

                    [~, c] = size(this.targets);

```

```

        for j = i+1:(i + c*this.runs)
            this = readRow(this, this.rows(j,1));
        end
    end
end
end

function this = readTargets(this, row)
    targetsCell = strsplit(char(row), ' ');

    [r, ~] = size(targetsCell);
    for i = 1:r
        targetsVec(i,:) = str2double(targetsCell(i,:));
    end

    this.targets = targetsVec;
end

function this = readRow(this, row)
    pattern = '(?<runs>[0-9]+) [ ]+(?<target>[0-9]+) [ ]+(?<dev>[0-9-
.]+)$';

    datas = regexp(char(row), pattern, 'names');

    datas.dists = this.targets(str2num(datas.target)) +
str2double(datas.dev)/1000;

    this.measures = [this.measures;datas];
end
end
end

```

3. forráskód: readrtl.m (ReadRtl osztály)

```

classdef GeoTools
    methods(Static)

        function [gamma, gammaM] = cosines(dists, M)
            syms f(a, b, c) F(a, b, c) a b c;
            f(a, b, c) = acos((a^2 + b^2 - c^2)/(2*a*b));

            gamma = double(f(dists(1), dists(2), dists(3)));

            F(a, b, c) = [diff(f, a),diff(f, b),diff(f, c)];

            gammaM = double(F(dists(1), dists(2), dists(3))*M*F(dists(1),
dists(2), dists(3)).');
        end

        function [val, M] = minus(val1, val2, M1, M2)
            syms f(x,y) x y;

            f(x,y) = x - y;

            val = double(f(val1, val2));
        end
    end
end

```

```

F = [diff(f, x), diff(f, y)];

M = double(F(val1, val2)*blkdiag(M1,M2)*transpose(F(val1,
val2)));
end
function [aTan, atanM] = calcAtan(dist1, dist2, M1, M2)
syms Fatan(x, y) x y;

Fatan(x, y) = atan(x/y);
M = blkdiag(M1, M2);

F = [diff(Fatan, x), diff(Fatan, y)];

aTan = double(Fatan(dist1, dist2));

atanM = double(F(dist1, dist2)*M*transpose(F(dist1, dist2)));

end

function [dist, distM] = pointPointDist(point1, point2, pointM1,
pointM2)

syms F(x1, y1, z1, x2, y2, z2) Fd(x1, y1, z1, x2, y2, z2) x1 y1
z1 x2 y2 z2;

M = blkdiag(pointM1, pointM2);

Fd(x1, y1, z1, x2, y2, z2) = ((x1 - x2)^2 + (y1 - y2)^2 + (z1 -
z2)^2)^(1/2);

F = [diff(Fd, x1), diff(Fd, y1), diff(Fd, z1), diff(Fd, x2),
diff(Fd, y2), diff(Fd, z2)];

dist = double(Fd(point1(1), point1(2), point1(3), point2(1),
point2(2), point2(3)));

distM = double(F(point1(1), point1(2), point1(3), point2(1),
point2(2), point2(3))*M*transpose(F(point1(1), point1(2), point1(3),
point2(1), point2(2), point2(3))));
end

function [outParams, outM] = lineParamsAngles2Vector(inParams,
inputM)

x = inParams(1);
y = inParams(2);
z = inParams(3);

syms Fa(phi, lambda) Fb(phi, lambda) Fc(phi, lambda) phi lambda;

a = 1;
b = tan(inParams(4));
c = sqrt(a^2 + b^2)*tan(inParams(5));

v = [a;b;c];

Fa(phi, lambda) = 1/norm(v);

```

```

Fb(phi, lambda) = tan(phi)/norm(v);
Fc(phi, lambda) = sqrt(1^2 + (tan(phi))^2)*tan(lambda)/norm(v);

F = [
    diff(Fa, phi), diff(Fa, lambda);
    diff(Fb, phi), diff(Fb, lambda);
    diff(Fc, phi), diff(Fc, lambda)
];
outParams = double([x; y; z; Fa(inParams(4), inParams(5));
Fb(inParams(4), inParams(5)); Fc(inParams(4), inParams(5))]);

tempoutM = double(F(inParams(4),
inParams(5)))*inputM(4:5,4:5)*transpose(double(F(inParams(4),
inParams(5))));
outM = inputM(1:3,1:3);
outM(4,4) = tempoutM(1,1);
outM(4,5) = tempoutM(1,2);
outM(4,6) = tempoutM(1,3);
outM(5,4) = tempoutM(2,1);
outM(5,5) = tempoutM(2,2);
outM(5,6) = tempoutM(2,3);
outM(6,4) = tempoutM(3,1);
outM(6,5) = tempoutM(3,2);
outM(6,6) = tempoutM(3,3);
end

function [point, outputM] = origoLinePerp(params, inputM) %origo-
line perpendicular

syms FX(X,Y,Z,A,B,C) FY(X,Y,Z,A,B,C) FZ(X,Y,Z,A,B,C) X Y Z A B
C;

FX(X,Y,Z,A,B,C) = X+A*(A*X+B*Y+C*Z)/(A^2+B^2+C^2);
FY(X,Y,Z,A,B,C) = Y+B*(A*X+B*Y+C*Z)/(A^2+B^2+C^2);
FZ(X,Y,Z,A,B,C) = Z+C*(A*X+B*Y+C*Z)/(A^2+B^2+C^2);

F(X,Y,Z,A,B,C) = [
    diff(FX, X),diff(FX, Y),diff(FX, Z),diff(FX, A),diff(FX,
B),diff(FX, C);
    diff(FY, X),diff(FY, Y),diff(FY, Z),diff(FY, A),diff(FX,
B),diff(FY, C);
    diff(FZ, X),diff(FZ, Y),diff(FZ, Z),diff(FZ, A),diff(FZ,
B),diff(FZ, C)
];

point =
[double(FX(params(1),params(2),params(3),params(4),params(5),params(6)));do
uble(FY(params(1),params(2),params(3),params(4),params(5),params(6)));doubl
e(FZ(params(1),params(2),params(3),params(4),params(5),params(6)))]];

outputM =
double(F(params(1),params(2),params(3),params(4),params(5),params(6)))*inpu
tM*double(transpose(F(params(1),params(2),params(3),params(4),params(5),par
ams(6))));
end
function deg = rad2deg(rad)
deg = rad*180/pi;
end

```



```

function rad = deg2rad(deg)
    rad = deg * pi/180;
end

function deg = DMS2deg(DMS)
    deg = DMS(:,1) + DMS(:,2)*60 + DMS(:,3)*3600;
end

function rad = DMS2rad(DMS)
    deg = DMS(:,1) + DMS(:,2)*60 + DMS(:,3)*3600;
    rad = deg*pi/180;
end

function DMS = deg2DMS(deg)
    DMS = [floor(deg) floor((deg-floor(deg))*60) deg*3600-
(floor(deg)*3600+floor((deg-floor(deg))*60)*60)];
end

function DMS = rad2DMS(rad)
    deg = rad*180/pi;
    DMS = [floor(deg) floor((deg-floor(deg))*60) deg*3600-
(floor(deg)*3600+floor((deg-floor(deg))*60)*60)];
end

function outputMeasures = grouping(inputMeasures)
    %grouping by point ID
    [r, ~] = size(inputMeasures);
    for i = 1:r
        if (~exist('outputMeasures', 'var'))
            outputMeasures.(inputMeasures(i).ID).distance =
inputMeasures(i).distance;
            outputMeasures.(inputMeasures(i).ID).direction =
inputMeasures(i).direction;
            outputMeasures.(inputMeasures(i).ID).zenith =
inputMeasures(i).zenith;
        else
            if (~isfield(outputMeasures ,inputMeasures(i).ID))
                outputMeasures.(inputMeasures(i).ID).distance =
inputMeasures(i).distance;
                outputMeasures.(inputMeasures(i).ID).direction =
inputMeasures(i).direction;
                outputMeasures.(inputMeasures(i).ID).zenith =
inputMeasures(i).zenith;
            else
                temp.distance = inputMeasures(i).distance;
                temp.direction = inputMeasures(i).direction;
                temp.zenith = inputMeasures(i).zenith;

                outputMeasures.(inputMeasures(i).ID) =
[outputMeasures.(inputMeasures(i).ID); temp];
            end
        end
    end
end%end of grouping

function matrix = struct2matrix(struct)
    %convert structure to matrix [distance, direction, zenith];

    matrix = [];

```

```

[r, ~] = size(struct);
for i = 1:r
    matrix = [matrix; struct(i).distance, struct(i).direction,
struct(i).zenith];
end
end%end of struct2matrix

function outMeasures = measureFrom2Faces(measures)
%measures struct {ID, direction, zenith, distance}

outMeasures = [];
[r, ~] = size(measures);
readyIndexes = [];

for i = 1:r
    facesIndexes = [];
    exist = 0;
    [r, ~] = size(readyIndexes);
    for j = 1:r
        if (strcmp(measures(i).ID,
measures(readyIndexes(j)).ID))
            exist = 1;
        end
    end

    if (exist == 0)
        facesIndexes = [i];

        [r, ~] = size(measures);
        for j = (i+1):r
            if (strcmp(measures(i).ID, measures(j).ID))
                facesIndexes = [facesIndexes j];
            end
        end

    end

    [~, c] = size(facesIndexes);
    if (c == 1)
        %only one face
        outMeasures = [outMeasures; measures(i)];
        readyIndexes = [readyIndexes; i];
    elseif (c == 2)
        %2 faces
        %calculate collimation and index fault
        %and average distance

        firstFace = measures(facesIndexes(1));
        secondFace = measures(facesIndexes(2));

        %collimation fault

        if (firstFace.direction > 2*pi)
            firstFace.direction = firstFace.direction - 2*pi;
        end

        if (secondFace.direction > 2*pi)
            secondFace.direction = secondFace.direction - 2*pi;
        end
    end
end

```

```

        calculatedMeasures.direction = (firstFace.direction +
secondFace.direction)/2;

        %index fault

        fullCircle = firstFace.zenith + secondFace.zenith;

        calculatedMeasures.zenith = firstFace.zenith + (2*pi -
fullCircle)/2;

        %average distance

        calculatedMeasures.distance = (firstFace.distance +
secondFace.distance)/2;

        calculatedMeasures.ID = firstFace.ID;
        %disp(calculatedMeasures);
        outMeasures = [outMeasures; calculatedMeasures];
        readyIndexes = [readyIndexes; facesIndexes(1);
facesIndexes(2)];

        elseif (c >= 2)
            %more than 2 faces ???
            for j = 1:c
                outMeasures = [outMeasures;
measures(facesIndexes(j))];
                readyIndexes = [readyIndexes; facesIndexes(j)];
            end
        end
    end%end if measureFrom2Faces

function [coords, outputM] = polarPoint(measure, inputM)
    %format shortE;
    syms F X Y Z t d z;

    X(t, d) = t*cos(d);
    Y(t, d) = t*sin(d);
    Z(t, z) = t*cos(z);

    F(t, d, z) = [
        diff(X(t, d), t), diff(X(t, d), d), 0;
        diff(Y(t, d), t), diff(Y(t, d), d), 0;
        diff(Z(t, z), t), 0, diff(Z(t, z), z)];

    [r, ~] = size(measure);

    coords = [double(X(measure(1), measure(2))),double(Y(measure(1),
measure(2))),double(Z(measure(1), measure(3)))]);
    outputM = double(F(measure(1), measure(2),
measure(3))*inputM*F(measure(1), measure(2), measure(3)).');
    end%end of polarPoint

function [d, z, dM, zM] = directionFrom2Points(p1, p2, M1, M2)

```

```

    %p1, p2 vectors [Y, X, Z]
    syms f(y1, x1, z1, y2, x2, z2) g(y1, x1, z1, y2, x2, z2) x1 y1
z1 x2 y2 z2;

    if p1(1) <= p2(1) && p1(2) <= p2(2)

        f(y1, x1, z1, y2, x2, z2) = atan((y2-y1)/(x2-x1));

    elseif p1(1) <= p2(1) && p1(2) > p2(2)
        f(y1, x1, z1, y2, x2, z2) = pi - atan((y2-y1)/(x1-x2));

    elseif p1(1) > p2(1) && p1(2) > p2(2)
        f(y1, x1, z1, y2, x2, z2) = pi + atan((y1-y2)/(x1-x2));
    else
        f(y1, x1, z1, y2, x2, z2) = 2*pi - atan((y2-y1)/(x1-x2));
    end

    g(y1, x1, z1, y2, x2, z2) = atan(sqrt((x2-x1)^2 + (y2-
y1)^2)/(z2-z1));

    F = [diff(f, x1), diff(f, y1), diff(f, x2), diff(f, y2), 0, 0];

    G = [diff(g, x1), diff(g, y1), diff(g, z1), diff(g, x2), diff(g,
y2), diff(g, z2)];

    d = double(f(p1(1),p1(2),p1(3),p2(1),p2(2),p2(3)));

    z = double(g(p1(1),p1(2),p1(3),p2(1),p2(2),p2(3)));

    dM =
double(F(p1(1),p1(2),p1(3),p2(1),p2(2),p2(3))*blkdiag(M1,M2)*F(p1(1),p1(2),
p1(3),p2(1),p2(2),p2(3)).');

    zM =
double(G(p1(1),p1(2),p1(3),p2(1),p2(2),p2(3))*blkdiag(M1,M2)*G(p1(1),p1(2),
p1(3),p2(1),p2(2),p2(3)).');
    end
end
end
end

```

4. forráskód: geotools.m (GeoTools osztály)

```

format longG;clear;

dir = 'meresek';
distance = '50m';%{'10m', '30m', '50m'};
orientation = 'direction';%{'cross', 'direction'};
tapeSize = '60mm';%{'30mm', '40mm', '50mm', '60mm'};
rounds = 5;
way = 2;
pos = {'tape00', 'tape05', 'tape10', 'tape15', 'tape20'};

%deviations from the manual
aprioriAngle = 1/3600*(pi/180);
syms aprioriDistance t
aprioriDistance(t) = 0.005+2*t/1000000;

```

```

tapefile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', dista
nce, '_', orientation, '_', tapeSize, '_sorozat', '.geo');
dev = ReadGeo(tapefile);

%disp(dev.measures(1));
dev = GeoTools.measureFrom2Faces (dev.measures);
%disp(dev);

pr.all = [];
watr.all = [];
woatr.all = [];

seriesFile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', dista
nce, '_', orientation, '_', tapeSize, '_sorozat.geo');

temp = ReadGeo(seriesFile);

temp = GeoTools.grouping(temp.measures);

pr.series = GeoTools.struct2matrix(temp.pr);
watr.series = GeoTools.struct2matrix(temp.tape_watr);
woatr.series = GeoTools.struct2matrix(temp.tape_woatr);

for i = 1:rounds
    pr.(strcat('r_', num2str(i))) = [];
    watr.(strcat('r_', num2str(i))) = [];
    woatr.(strcat('r_', num2str(i))) = [];
    [~, c] = size(pos);
    for k = 1:way
        if k == 1
            interval = 1:c;
        else
            interval = c:-1:1;
        end
        for j = interval
            tape = char(pos(j));
            tapeFile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', num2s
tr(i), '/', distance, '_', orientation, '_', tapeSize, '_', tape, '_', num2str(k), '.g
eo');

                if exist(tapeFile, 'file')

                    temp = ReadGeo(tapeFile);

                    temp = GeoTools.measureFrom2Faces (temp.measures);

                    temp = GeoTools.grouping(temp);

                    pr.(strcat('r_', num2str(i))) = [pr.(strcat('r_',
num2str(i));GeoTools.struct2matrix(temp.pr)];
                    watr.(strcat('r_', num2str(i))) = [watr.(strcat('r_',
num2str(i));GeoTools.struct2matrix(temp.tape_watr)];
                    woatr.(strcat('r_', num2str(i))) = [woatr.(strcat('r_',
num2str(i));GeoTools.struct2matrix(temp.tape_woatr)];

```

```

        distsFile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', num2s
tr(i), '/', distance, '_', orientation, '_', tapeSize, '_', num2str(i), '.rtl');
        interferoDists.(strcat('r_', num2str(i))) =
ReadRtl(distsFile);
        else
            disp(strcat(tapeFile, ' does not exist'));

            pr.(strcat('r_', num2str(i))) = [pr.(strcat('r_',
num2str(i)));0 0 0];
            watr.(strcat('r_', num2str(i))) = [watr.(strcat('r_',
num2str(i)));0 0 0];
            woatr.(strcat('r_', num2str(i))) = [woatr.(strcat('r_',
num2str(i)));0 0 0];

            end
        end
    end
    pr.all = [pr.all;pr.(strcat('r_', num2str(i)))]];
    watr.all = [watr.all;watr.(strcat('r_', num2str(i)))]];
    woatr.all = [woatr.all;woatr.(strcat('r_', num2str(i)))]];
end

%displacement test

[r, c] = size(pr.all(:,2));

pr.directions = [];
for i = 1:r
    if pr.all(i,2) ~= 0
        pr.directions = [pr.directions; pr.all(i,2)];
    end
end

pr.average = mean(pr.directions(:)); %prism average

pr.diffs = pr.directions(:) - pr.average; %diffs from average

maxDiff = max(abs(pr.diffs))*180/pi*3600; %max difference in prism measures
in seconds

%deviations from series

aposteriori.watrDist = std(watr.series(:,1));
aposteriori.watrDir = std(watr.series(:,2));
aposteriori.watrZen = std(watr.series(:,3));
aposteriori.woatrDist = std(woatr.series(:,1));
aposteriori.woatrDir = std(woatr.series(:,2));
aposteriori.woatrZen = std(woatr.series(:,3));

aposterioriM = diag([aposteriori.watrDist^2, aposteriori.watrDir^2,
aposteriori.watrZen^2]);

watr_coords.all = [];
woatr_coords.all = [];

```

```

for i = 1:5
    [r, c] = size(watr.(strcat('r_', num2str(i))));

    for j = 1:r
        [watr_coords.(strcat('r_', num2str(i)))(j,:),
watr_coordsM.(strcat('r_', num2str(i))(:, :, j)] =
GeoTools.polarPoint(watr.(strcat('r_', num2str(i)))(j,:), aposterioriM);
        [woatr_coords.(strcat('r_', num2str(i)))(j,:),
woatr_coordsM.(strcat('r_', num2str(i))(:, :, j)] =
GeoTools.polarPoint(woatr.(strcat('r_', num2str(i)))(j,:), aposterioriM);
    end
    watr_coords.all = [watr_coords.all;watr_coords.(strcat('r_',
num2str(i)))]];
    woatr_coords.all = [woatr_coords.all;woatr_coords.(strcat('r_',
num2str(i)))]];
end

[r, ~] = size(watr_coords.r_1);
watr_dists = [];
woatr_dists = [];
for i = 1:r
    for j = i:r
        if i ~= j
            watr_dists(i,j) =
GeoTools.pointPointDist(woatr_coords.all(i,:), woatr_coords.all(j,:),
zeros(3,3), zeros(3,3));
            woatr_dists(i,j) =
GeoTools.pointPointDist(watr_coords.all(i,:), watr_coords.all(j,:),
zeros(3,3), zeros(3,3));
        end
    end
end

[vec, tempr] = max(watr_dists);
[val, c] = max(vec);
r = tempr(c);

point1 = [watr_coords.all(r,2), watr_coords.all(r,1),
watr_coords.all(r,3)];
point2 = [watr_coords.all(c,2), watr_coords.all(c,1),
watr_coords.all(c,3)];

[intD, intZ] = GeoTools.directionFrom2Points(point2, point1, zeros(3,3),
zeros(3,3));

dirAngleDis = std(watr.all(:, 2));%direction angle dispersion
dirAngleMean = mean(watr.all(:, 2));%direction angle mean(average)

zenAngleDis = std(watr.all(:, 3));%zenith angle dispersion
zenAngleMean = mean(watr.all(:, 3));%zen angle mean(average)

[r, ~] = size(interferoDists.r_1.measures);

for i = 1:r

```

```

        intDists(i) = (interferoDists.r_1.measures(i).dists-
interferoDists.r_1.measures(1).dists)/1000;%interferometer dists
end

watr_distDiffsWCor = [];
woatr_distDiffsWCor = [];
watr_distDiffsWoCor = [];
woatr_distDiffsWoCor = [];

horCorr = [];
verCorr = [];
for i = 1:5
    watr_zero(i) = watr.(strcat('r_', num2str(i)))(1,1);
    woatr_zero(i) = woatr.(strcat('r_', num2str(i)))(1,1);

    alphaD(:) = watr.(strcat('r_', num2str(i)))(:,2) - watr.(strcat('r_',
num2str(i)))(1,2);
    alphaZ(:) = watr.(strcat('r_', num2str(i)))(:,3) - watr.(strcat('r_',
num2str(i)))(1,3);

    betaD = intD - watr.(strcat('r_', num2str(i)))(1,2);
    betaZ = intZ - watr.(strcat('r_', num2str(i)))(1,3);
    horCorr(i,:) = abs(cos(betaD)./cos(alphaD(:)));
    verCorr(i,:) = abs(cos(betaZ)./cos(alphaZ(:)));

    [r, ~] = size(watr.(strcat('r_', num2str(i))));

    wCorIntDists.(strcat('r_', num2str(i))) = [];
    woCorIntDists.(strcat('r_', num2str(i))) = [];
    watr_beDist.(strcat('r_', num2str(i))) = [];
    woatr_beDist.(strcat('r_', num2str(i))) = [];

    for j = 2:r
        if(watr.(strcat('r_', num2str(i)))(j,1) ~= 0)
            wCorIntDists.(strcat('r_', num2str(i))) =
[wCorIntDists.(strcat('r_', num2str(i)))] + (interferoDists.(strcat('r_',
num2str(i))).measures(j).dists - interferoDists.(strcat('r_',
num2str(i))).measures(1).dists)*horCorr(j)*verCorr(j)];%corrected
interferometer dists
            woCorIntDists.(strcat('r_', num2str(i))) =
[woCorIntDists.(strcat('r_', num2str(i)))] + (interferoDists.(strcat('r_',
num2str(i))).measures(j).dists - interferoDists.(strcat('r_',
num2str(i))).measures(1).dists)];%without correction interferometer dists
            watr_beDist.(strcat('r_', num2str(i))) =
[watr_beDist.(strcat('r_', num2str(i)))] + (watr.(strcat('r_',
num2str(i)))(j,1) - watr_zero(i))*1000];
            woatr_beDist.(strcat('r_', num2str(i))) =
[woatr_beDist.(strcat('r_', num2str(i)))] + (woatr.(strcat('r_',
num2str(i)))(j,1) - woatr_zero(i))*1000];
        end
    end

    [~, c] = size(watr_beDist.(strcat('r_', num2str(i))));

    watr_distDiffsWCor = [watr_distDiffsWCor; watr_beDist.(strcat('r_',
num2str(i)))(:) - wCorIntDists.(strcat('r_', num2str(i)))]];
    woatr_distDiffsWCor = [woatr_distDiffsWCor; woatr_beDist.(strcat('r_',
num2str(i)))(:) - woCorIntDists.(strcat('r_', num2str(i)))]];

```



```

    watr_distDiffsWoCor = [watr_distDiffsWoCor; watr_beDist.(strcat('r_',
num2str(i)))(:) - woCorIntDists.(strcat('r_', num2str(i)))];
    woatr_distDiffsWoCor = [woatr_distDiffsWoCor;
woatr_beDist.(strcat('r_', num2str(i)))(:) - woCorIntDists.(strcat('r_',
num2str(i)))];

end

watr_mu_wCor = sqrt(mean(watr_distDiffsWCor.^2));
woatr_mu_wCor = sqrt(mean(woatr_distDiffsWCor.^2));
watr_mu_woCor = sqrt(mean(watr_distDiffsWoCor.^2));
woatr_mu_woCor = sqrt(mean(woatr_distDiffsWoCor.^2));

filename =
strcat('eredmenyek/', distance, '_', orientation, '_', tapeSize, '.mat');
save(filename);

```

5. forráskód: *direction_process.m* (A távmérő pontosságának vizsgálata)

```

format longG;clear;

dir = 'meresek';
distance = '10m';%{'10m', '30m', '50m'};
orientation = 'cross';%{'cross', 'direction'};
tapeSize = '30mm';%{'30mm', '40mm', '50mm', '60mm'};
rounds = 5;
way = 2;
pos = {'tape00', 'tape05', 'tape10', 'tape15', 'tape20'};

%deviations from the manual
aprioriAngle = 1/3600*(pi/180);
syms aprioriDistance t
aprioriDistance(t) = 0.005+2*t/1000000;

tapefile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', dista
nce, '_', orientation, '_', tapeSize, '_sorozat', '.geo');
dev = ReadGeo(tapefile);

%disp(dev.measures(1));
dev = GeoTools.measureFrom2Faces(dev.measures);
%disp(dev);

pr.all = [];
watr.all = [];
woatr.all = [];

seriesFile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', dista
nce, '_', orientation, '_', tapeSize, '_sorozat.geo');

temp = ReadGeo(seriesFile);

temp = GeoTools.grouping(temp.measures);

pr.series = GeoTools.struct2matrix(temp.pr);
watr.series = GeoTools.struct2matrix(temp.tape_watr);
woatr.series = GeoTools.struct2matrix(temp.tape_woatr);

```

```

for i = 1:rounds
    pr.(strcat('r_', num2str(i))) = [];
    watr.(strcat('r_', num2str(i))) = [];
    woatr.(strcat('r_', num2str(i))) = [];
    [~, c] = size(pos);
    for k = 1:way
        if k == 1
            interval = 1:c;
        else
            interval = c:-1:1;
        end
        for j = interval
            tape = char(pos(j));
            tapeFile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', num2s
tr(i), '/', distance, '_', orientation, '_', tapeSize, '_', tape, '_', num2str(k), '.g
eo');

            if exist(tapeFile, 'file')

                temp = ReadGeo(tapeFile);

                temp = GeoTools.measureFrom2Faces(temp.measures);

                temp = GeoTools.grouping(temp);

                pr.(strcat('r_', num2str(i))) = [pr.(strcat('r_',
num2str(i)));GeoTools.struct2matrix(temp.pr)];
                watr.(strcat('r_', num2str(i))) = [watr.(strcat('r_',
num2str(i)));GeoTools.struct2matrix(temp.tape_watr)];
                woatr.(strcat('r_', num2str(i))) = [woatr.(strcat('r_',
num2str(i)));GeoTools.struct2matrix(temp.tape_woatr)];

                distsFile =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', num2s
tr(i), '/', distance, '_', orientation, '_', tapeSize, '_', num2str(i), '.rtl');
                interferoDists.(strcat('r_', num2str(i))) =
ReadRtl(distsFile);
            else
                disp(strcat(tapeFile, ' does not exist'));

                pr.(strcat('r_', num2str(i))) = [pr.(strcat('r_',
num2str(i));0 0 0)];
                watr.(strcat('r_', num2str(i))) = [watr.(strcat('r_',
num2str(i));0 0 0)];
                woatr.(strcat('r_', num2str(i))) = [woatr.(strcat('r_',
num2str(i));0 0 0)];

            end
        end
    end
    pr.all = [pr.all;pr.(strcat('r_', num2str(i)))]];
    watr.all = [watr.all;watr.(strcat('r_', num2str(i)))]];
    woatr.all = [woatr.all;woatr.(strcat('r_', num2str(i)))]];
end

%displacement test

[r, c] = size(pr.all(:,2));

```

```

pr.directions = [];
for i = 1:r
    if pr.all(i,2) ~= 0
        pr.directions = [pr.directions; pr.all(i,2)];
    end
end

pr.average = mean(pr.directions(:)); %prism average

pr.diffs = pr.directions(:) - pr.average; %diffs from average

maxDiff = max(abs(pr.diffs))*180/pi*3600; %max difference in prism measures
in seconds

%deviations from series

aposteriori.watrDist = std(watr.series(:,1));
aposteriori.watrDir = std(watr.series(:,2));
aposteriori.watrZen = std(watr.series(:,3));
aposteriori.woatrDist = std(woatr.series(:,1));
aposteriori.woatrDir = std(woatr.series(:,2));
aposteriori.woatrZen = std(woatr.series(:,3));

aposterioriM = diag([aposteriori.watrDist^2, aposteriori.watrDir^2,
aposteriori.watrZen^2]);

watr_coords.all = [];
woatr_coords.all = [];
for i = 1:5
    [r, c] = size(watr.(strcat('r_', num2str(i))));

    for j = 1:r
        [watr_coords.(strcat('r_', num2str(i)))(j,:),
watr_coordsM.(strcat('r_', num2str(i)))(:,:,j)] =
GeoTools.polarPoint(watr.(strcat('r_', num2str(i)))(j,:), aposterioriM);
        [woatr_coords.(strcat('r_', num2str(i)))(j,:),
woatr_coordsM.(strcat('r_', num2str(i)))(:,:,j)] =
GeoTools.polarPoint(woatr.(strcat('r_', num2str(i)))(j,:), aposterioriM);
    end
    watr_coords.all = [watr_coords.all;watr_coords.(strcat('r_',
num2str(i)))];
    woatr_coords.all = [woatr_coords.all;woatr_coords.(strcat('r_',
num2str(i)))];
end

[r, ~] = size(watr_coords.r_1);
watr_dists = [];
woatr_dists = [];
for i = 1:r
    for j = i:r
        if i ~= j
            watr_dists(i,j) =
GeoTools.pointPointDist(woatr_coords.all(i,:), woatr_coords.all(j,:),
zeros(3,3), zeros(3,3));
        end
    end
end

```

```

        woatr_dists(i,j) =
GeoTools.pointPointDist(watr_coords.all(i,:), watr_coords.all(j,:),
zeros(3,3), zeros(3,3));
        end
    end
end

[vec, tempr] = max(watr_dists);
[val, c] = max(vec);
r = tempr(c);

point1 = [watr_coords.all(r,2), watr_coords.all(r,1),
watr_coords.all(r,3)];
point2 = [watr_coords.all(c,2), watr_coords.all(c,1),
watr_coords.all(c,3)];

[intD, intZ] = GeoTools.directionFrom2Points(point2, point1, zeros(3,3),
zeros(3,3));

dirAngleDis = std(watr.all(:, 2));%direction angle dispersion
dirAngleMean = mean(watr.all(:, 2));%direction angle mean(average)

corr = abs(sin(intZ)); %correction

zenAngleDis = std(watr.all(:, 3));%zenith angle dispersion
zenAngleMean = mean(watr.all(:, 3));%zen angle mean(average)

[r, ~] = size(interferoDists.r_1.measures);

for i = 1:r
    intDists(i) = (interferoDists.r_1.measures(i).dists-
interferoDists.r_1.measures(1).dists)/1000;%interferometer dists
end

watr_distDiffsWCor = [];
woatr_distDiffsWCor = [];
watr_distDiffsWoCor = [];
woatr_distDiffsWoCor = [];

for i = 1:5
    watr_zero(i,:) = watr_coords.(strcat('r_', num2str(i)))(1,:);
    woatr_zero(i,:) = woatr_coords.(strcat('r_', num2str(i)))(1,:);

    [r, ~] = size(watr_coords.(strcat('r_', num2str(i))));

    wCorIntDists.(strcat('r_', num2str(i))) = [];
    woCorIntDists.(strcat('r_', num2str(i))) = [];

    watr_beDist.(strcat('r_', num2str(i))) = [];
    watr_beDistM.(strcat('r_', num2str(i))) = [];
    woatr_beDist.(strcat('r_', num2str(i))) = [];
    woatr_beDistM.(strcat('r_', num2str(i))) = [];
end

```

```

for j = 2:r
    if(watr.(strcat('r_', num2str(i)))(j,1) ~= 0)
        wCorIntDists.(strcat('r_', num2str(i))) =
[wCorIntDists.(strcat('r_', num2str(i))); (interferoDists.(strcat('r_',
num2str(i))).measures(j).dists - interferoDists.(strcat('r_',
num2str(i))).measures(1).dists)*corr];%corrected interferometer dists
        woCorIntDists.(strcat('r_', num2str(i))) =
[woCorIntDists.(strcat('r_', num2str(i))); (interferoDists.(strcat('r_',
num2str(i))).measures(j).dists - interferoDists.(strcat('r_',
num2str(i))).measures(1).dists)];%without correction interferometer dists
        [temp_watr, temp_watrM] =
GeoTools.pointPointDist(watr_zero(i,:), watr_coords.(strcat('r_',
num2str(i)))(j,:), watr_coordsM.(strcat('r_', num2str(i)))(:,:,1),
watr_coordsM.(strcat('r_', num2str(i)))(:,:,j));
        [temp_woatr, temp_woatrM] =
GeoTools.pointPointDist(woatr_zero(i,:), woatr_coords.(strcat('r_',
num2str(i)))(j,:), woatr_coordsM.(strcat('r_', num2str(i)))(:,:,1),
woatr_coordsM.(strcat('r_', num2str(i)))(:,:,j));
        watr_beDist.(strcat('r_', num2str(i))) =
[watr_beDist.(strcat('r_', num2str(i))); temp_watr*1000];
        watr_beDistM.(strcat('r_', num2str(i))) =
[watr_beDistM.(strcat('r_', num2str(i))); temp_woatrM];

        woatr_beDist.(strcat('r_', num2str(i))) =
[woatr_beDist.(strcat('r_', num2str(i))); temp_woatr*1000];
        woatr_beDistM.(strcat('r_', num2str(i))) =
[woatr_beDistM.(strcat('r_', num2str(i))); temp_woatrM];
    end
end

[~, c] = size(watr_beDist.(strcat('r_', num2str(i))));

    watr_distDiffsWCor = [watr_distDiffsWCor; watr_beDist.(strcat('r_',
num2str(i)))(:) - wCorIntDists.(strcat('r_', num2str(i)))];
    woatr_distDiffsWCor = [woatr_distDiffsWCor; woatr_beDist.(strcat('r_',
num2str(i)))(:) - wCorIntDists.(strcat('r_', num2str(i)))];
    watr_distDiffsWoCor = [watr_distDiffsWoCor; watr_beDist.(strcat('r_',
num2str(i)))(:) - woCorIntDists.(strcat('r_', num2str(i)))];
    woatr_distDiffsWoCor = [woatr_distDiffsWoCor;
woatr_beDist.(strcat('r_', num2str(i)))(:) - woCorIntDists.(strcat('r_',
num2str(i)))];

end

watr_mu_wCor = sqrt(mean(watr_distDiffsWCor.^2));
woatr_mu_wCor = sqrt(mean(woatr_distDiffsWCor.^2));
watr_mu_woCor = sqrt(mean(watr_distDiffsWoCor.^2));
woatr_mu_woCor = sqrt(mean(woatr_distDiffsWoCor.^2));

filename =
strcat('eredmenyek/', distance, '_', orientation, '_', tapeSize, '.mat');
save(filename);

```

6. forráskód: *cross_process.m* (A szögmérő pontosságának vizsgálata)

```

format longG;clear;

dir = 'meresek';
distance = '30m';%{'10m', '30m', '50m'};
orientation = 'rotation';%{'cross', 'direction', 'rotation'};

```

```

tapeSize = '30mm';%{'30mm','40mm','50mm','60mm'};
%pos = {'-40', '-30', '-20', '-10', '00', '10', '20', '30', '40'};%10m
pos = {'-20', '-10', '00', '10', '20'};%30m

%deviations from the manual
aprioriAngle = 1/3600*(pi/180);
syms aprioriDistance t
aprioriDistance(t) = 0.005+2*t/1000000;

pr.all = [];
watr.all = [];
woatr.all = [];

[r, c] = size(pos);

for i = 1:c
    pr.(strcat('p_', num2str(i))) = [];
    watr.(strcat('p_', num2str(i))) = [];
    woatr.(strcat('p_', num2str(i))) = [];
    tape = char(pos(i));
    tapeFileWatr =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', dista
nce, '_', orientation, '_watr_', tapeSize, '_', tape, '.geo');
    tapeFileWoaatr =
strcat(dir, '/', distance, '/', distance, '_', orientation, '_', tapeSize, '/', dista
nce, '_', orientation, '_woatr_', tapeSize, '_', tape, '.geo');
    if exist(tapeFileWatr, 'file')
        tempWatr = ReadGeo(tapeFileWatr);
        tempWoaatr = ReadGeo(tapeFileWoaatr);

        tempWatr = GeoTools.measureFrom2Faces(tempWatr.measures);
        tempWoaatr = GeoTools.measureFrom2Faces(tempWoaatr.measures);

        tempWatr = GeoTools.grouping(tempWatr);
        tempWoaatr = GeoTools.grouping(tempWoaatr);

        pr.(strcat('p_', num2str(i))) = [pr.(strcat('p_',
num2str(i)));GeoTools.struct2matrix(tempWatr.pr)];
        pr.(strcat('p_', num2str(i))) = [pr.(strcat('p_',
num2str(i)));GeoTools.struct2matrix(tempWoaatr.pr)];
        watr.(strcat('p_', num2str(i))) = [watr.(strcat('p_',
num2str(i)));GeoTools.struct2matrix(tempWatr.tape)];
        woatr.(strcat('p_', num2str(i))) = [woatr.(strcat('p_',
num2str(i)));GeoTools.struct2matrix(tempWoaatr.tape)];

    else
        disp(strcat(tapeFile, ' does not exist'));
        pr.(strcat('p_', num2str(i))) = [pr.(strcat('p_', num2str(i)));0 0
0];
        watr.(strcat('p_', num2str(i))) = [watr.(strcat('p_',
num2str(i)));0 0 0];
        woatr.(strcat('p_', num2str(i))) = [woatr.(strcat('p_',
num2str(i)));0 0 0];

    end
pr.all = [pr.all;pr.(strcat('p_', num2str(i)))]];
watr.all = [watr.all;watr.(strcat('p_', num2str(i)))]];
woatr.all = [woatr.all;woatr.(strcat('p_', num2str(i)))]];

```

```

end

%displacement test

[r, c] = size(pr.all(:,2));

pr.directions = [];
for i = 1:r
    if pr.all(i,2) ~= 0
        pr.directions = [pr.directions; pr.all(i,2)];
    end
end

pr.average = mean(pr.directions(:)); %prism average

pr.diffs = pr.directions(:) - pr.average; %diffs from average

maxDiff = max(abs(pr.diffs))*180/pi*3600; %max difference in prism measures
in seconds

[r, c] = size(pos);

for i = 1:c
    a_dist_watr(i) = mean(watr.(strcat('p_', num2str(i)))(:,1));
    a_dist_woatr(i) = mean(woatr.(strcat('p_', num2str(i)))(:,1));
    a_angle_watr(i) = mean(watr.(strcat('p_', num2str(i)))(:,2));
    a_angle_woatr(i) = mean(woatr.(strcat('p_', num2str(i)))(:,2));
    mu_dist_watr(i) = std(watr.(strcat('p_', num2str(i)))(:,1));
    mu_dist_woatr(i) = std(woatr.(strcat('p_', num2str(i)))(:,1));
    mu_angle_watr(i) = std(watr.(strcat('p_', num2str(i)))(:,2));
    mu_angle_woatr(i) = std(woatr.(strcat('p_', num2str(i)))(:,2));
end

filename =
strcat('eredmenyek/', distance, '_', orientation, '_', tapeSize, '.mat');
save(filename);

```

7. forráskód: rotation_process.m (A szög- és távmérő pontosságának vizsgálata a beesési szög függvényében)