

DRÓNOKON ALKALMAZOTT OPTIKAI ÁRAMLÁS ALAPÚ LEGYEZÉSI SZÖGSEBESSÉGBECSLÉS ÉS HIBADETEKTÁLÁS

Rövidítések

OF	-	Optical Flow
FAST	-	Features from Accelerated Segment Test
SURF	-	Speeded-Up Robust Features
SIFT	-	Scale-Invariant Feature Transform
ORB	-	Oriented FAST and Rotated BRIEF
FPS	-	Frames Per Second
BRIEF	-	Binary Robust Independent Elementary Features
ROC	-	Receiver Operating Characteristic
FA	-	False Alarm
MD	-	Missed Detection
TP	-	True Positive
FP	-	False Positive
TN	-	True Negative
FN	-	False Negative
<i>Tol1</i>	-	Reziduál küszöbértéke
<i>Tol2</i>	-	Hibaszámláló küszöbértéke

Bevezetés

Napjainkban a drónok gyors elterjedése és méretnövekedése megköveteli a redundáns rendszerek használatát. A szögsebesség-szenzorok jelentős szerepet töltenek be a drónok irányításában, ezért meghibásodásuk gyors észlelése fontos feladat a személy- és vagyonbiztonság érdekében. A legtöbb drón különböző funkciók ellátása érdekében (pld. felvétel készítése, videószemüveg) fel van szerelve kamerával, amely célszerűvé teszi annak használatát a szenzorhibák detektálására.

A tanulmányban egy kutatást mutatunk be, amelynek célja volt egy hexakopterre erősített, gimballal stabilizált, lefelé néző kamera képe alapján becsülni a legyezési szögsebességet az optikai áramlás segítségével, majd ezen szögsebesség felhasználásával valós repülésben mért szögsebesség adatokon mesterségesen generált hibákat detektálni.

Ehhez először levezetésre kerülnek az egyenletek, amelyek a lyukkamera-modell segítségével kapcsolatot teremtenek az optikai áramlás és a légi járműünk sebességei között. Ezeket az egyenleteket korábban nem tömegközéppontba helyezték, előre, ferdén lefele néző kamerához vezették le merevszárnyas pilóta nélküli légi járművekre. Elsőként ezen egyenletek jelenlegi problémára történő aktualizálása, majd az optikai áramlás meghatározása kerül bemutatásra. Az optikai áramlás számításához az egymást követő képeken jellemzőpontokat (feature point) kell találnunk és összepárosítanunk, viszont mivel közelítő eljárásokat használunk, ezért fontos a hibás párosítások kiszűrése, amelyre szintén az optikai áramlást hívjuk segítségül. A meghatározott optikai áramlás és a levezetett egyenletek segítségével becsülhető a legyezési szögsebesség.

Ismertetésre kerülnek a vizsgált szenzorhibák, a hibadetekciós eljárások, annak hangolása és az alkalmazás eredményei. A valós repülési adatokon ezen hibák mesterségesen előállításra kerülnek. A detektáláshoz összehasonlításra kerül a kamerakép alapján számolt legyezési szögsebesség és a mért legyezési szögsebesség. Utóbbi két külön szenzor adatainak használatával, amelyből egy tartalmazza a generált hibákat.

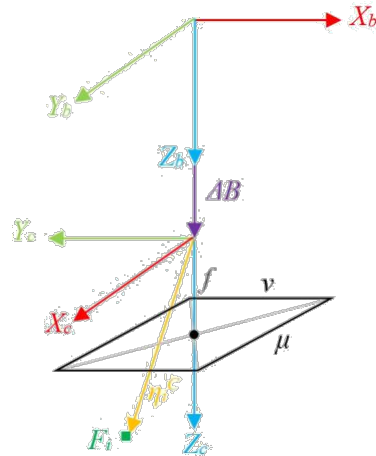
Végül bemutatásra kerül egy számlálót és határértékeket használó hibadetekciós eljárás, ahol minden időpillanatban kiértékelésre kerül az összehasonlítás, és ha ennek eredménye egy bizonyos határérték felett van, akkor egy számláló értékét növeljük, ha a határérték alatt van, akkor pedig csökkentjük egy előre meghatározott értékkel. A hibás szenzort a számláló egy bizonyos értéken való átlépése jelzi számunkra, és innentől kezdve a vizsgált szenzorunkat hibásnak tekintjük.

Optikai áramlás egyenletei a kamera koordináta-rendszerében

Ebben a fejezetben bemutatásra kerül a kapcsolat a kamera képe és a környezetben érzékelt jellemzőpontok között. Az itt ismertetett egyenleteket Joseph J. Kehoe és társai [1] cikkükben mutatták be, ahol az egyenletrendszer megoldásához különböző módszereket is leírtak. Ezt felhasználva [2] Kun Szabolcs és Bauer Péter bevezettek egy új kétlépéses lineáris megoldót, amellyel egy későbbi munkájukban [3] megmutatták, hogy a módszer alkalmas szenzorhiba detekcióra is. A kísérlet fő célja az volt, hogy megvizsgálja, ezen egyenletek alkalmazhatóak-e valós repülési adatok és valós kameraképek felhasználásával szögsebességszenzor hibadetekcióra.

Koordináta-rendszerek

Ahhoz, hogy megértsük, hogy a járműünk hogyan mozog a környezetünkben érzékelt jellemzőpontokhoz képest, először szükséges a koordináta-rendszerek meghatározása. A felhasznált koordináta-rendszereket az 1. ábra mutatja. Három fő koordináta-rendszert használunk:



1. ábra: Koordináta rendszerek
(saját szerkesztés)

Föld koordináta-rendszer (Earth): Más néven North-East-Down koordináta-rendszer. Ezen koordináta-rendszer a föld felszínének egy pontjához van kötve, amelynek x_e tengelye a geodetikus észak (N) felé mutat (a geodetikus észak az ellipszoid legészakibb pontja), y_e tengelye a geodetikus kelet (E) felé mutat, z_e tengelye pedig az érintősík merőlegese, a Föld belseje felé mutat (D).

Test koordináta-rendszer (Body): A test koordináta-rendszer a légi járművünk tömegközéppontjához (súlypont) van kötve, amelynek x_b tengelye a jármű orra felé néz, y_b tengelye a jármű jobb oldala felé, a z_b tengelye pedig lefelé mutat.

Kamera koordináta-rendszer (Camera): A kamerához kötött koordináta-rendszer. Ebben tudjuk értelmezni a kamera képe által szolgáltatott információkat, például ebben a rendszerben tudjuk számolni az optikai áramlást. Az x_c tengelye esetünkben a jármű jobb oldalának irányába mutat, y_c tengelye a jármű hátulja felé, z_c tengelye pedig lefelé mutat. Pozíciója a testrendszerben a ΔB vektorral van leírva, amely mutatja, hogy a kamera jó közelítéssel a súlypont alatt lefelé, függőlegesen helyezkedik el ($\Delta B = [0.0285 \quad -0.0115 \quad 0.175] \text{ [m]}$).

Transzformációk

Mivel célunk, hogy a későbbiekben az egyenletek egyszerűbb formáját kapjuk, ennek okán a kamera koordináta-rendszerben írjuk fel az egyenleteinket.

Ehhez szükséges a testrendszerben értelmezett sebességvektorainak, szögsebességvektorainak, illetve a ΔB vektorának kamera koordináta-rendszerbe történő transzformálása. Ezt a két koordináta-rendszer között általánosan a Z-Y-X kötött forgatási sorrendű Euler szögekkel való transzformációval tudjuk megtenni. Euler szögeknek az azimut szöget (Ψ), bólintási szöget (Θ) és a bedöntési szöget (Φ) nevezzük. Ez három egymás utáni elforgatást jelent, amelynek sorrendje kötött.

Az első forgatás a z tengely körül történik az azimut szöggel, a második az y körül a bólintási szöggel, majd az utolsó forgatás x tengely körül a bedöntési szöggel a következők szerint:

$$T_{\Psi} = \begin{bmatrix} \cos(\Psi) & \sin(\Psi) & 0 \\ -\sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T_{\Theta} = \begin{bmatrix} \cos(\Theta) & 0 & -\sin(\Theta) \\ 0 & 1 & 0 \\ \sin(\Theta) & 0 & \cos(\Theta) \end{bmatrix} \quad T_{\Phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Phi) & \sin(\Phi) \\ 0 & -\sin(\Phi) & \cos(\Phi) \end{bmatrix}$$

A szögsebességeinket és sebességeinket a test koordináta-rendszerben mérjük. Ahhoz, hogy kamerarendszerbe áttranszformáljuk, először transzformálnunk kell a föld koordináta-rendszerbe, melyet az alábbi transzformációs mátrixszal tudunk megtenni:

$$T_{EB} = (T_{BE})^T = (T_{\Phi} T_{\Theta} T_{\Psi})^T$$

Ezután a Föld koordináta-rendszerből tudunk a kamerarendszerbe transzformálni. Mivel a használt kameránk egy gimballal stabilizált, folyamatosan függőlegesen lefelé néző kamera, így a Föld-kamera transzformáció során csak a Φ és Θ szögekkel szükséges transzformálnunk, hogy a Ψ elforgatás hatása megmaradjon. Mivel a kamerarendszerünk x_c tengelye a jármű jobb oldala felé néz, az y_c tengelye pedig hátrafelé, ez a Föld koordináta-rendszerhez képest egy z tengely körüli 90 fokos elforgatásnak felel meg. Így az Euler szögekkel való transzformáció után még szükséges a T_{90° rotációs mátrixal való szorzás is.

$$T_{90^\circ} = \begin{bmatrix} \cos(90^\circ) & \sin(90^\circ) & 0 \\ -\sin(90^\circ) & \cos(90^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_{CE} = T_{90^\circ} \cdot T_{\Psi}$$

Így az alábbi transzformációs mátrix adódik:

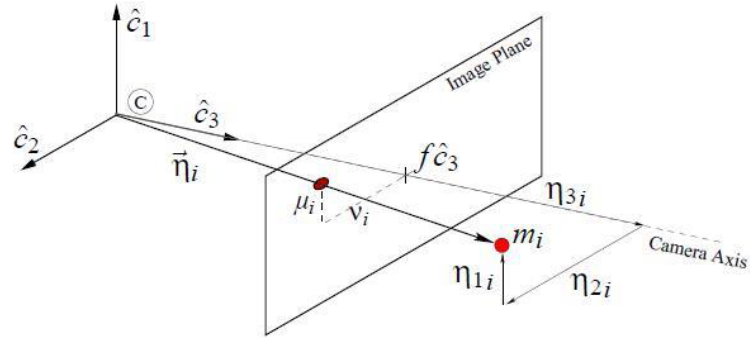
$$T_{CB} = T_{CE} * T_{EB}$$

A \underline{v}_b sebességvektor, a $\underline{\omega}_b$ szögsebességvektor és a ΔB pozíció vektor kamerarendszerbeli képe az alábbi módon számolhatóak:

$$\underline{v}_c = T_{CB} * \underline{v}_b, \quad \underline{\omega}_c = T_{CB} * \underline{\omega}_b, \quad \Delta C = T_{CB} * \Delta B$$

Lyukkamera-modell

Egy digitális kamera alapvető funkciója, hogy a háromdimenziós világunkról egy kétdimenziós képet csináljon. A legegyszerűbb kamera modell a lyukkamera (camera obscura). A lyukkamera-modell egy, a térben elhelyezkedő pont (későbbiekben jellemzőpont) koordinátái és a kamera képsíkjára vetített képe között határozza meg a matematikai összefüggést.



2. ábra: m pont leképezése a kamera képsíkjára [1]
(saját szerkesztés)

A kamera koordináta-rendszerének origója a kamera vetítési középpontjában van, azaz a fókuszpontban található. A kamera képsíkja merőleges a kamera optikai tengelyére. Az origó és a képsíkunk távolsága a fókusz távolság, a főtengely mentén mérve.[4]. A kamerarendszerben a jellemzőpontunk helyzetét az $(\eta_i = [\eta_{1i} \ \eta_{2i} \ \eta_{3i}]^T)$ vektor írja le. Ha ezt a pontot levetítjük a képsíkunkra, annak a vetületét a $(\mu_i \ \nu_i)$ koordináta páros fogja leírni. Ezt a leképezést az alábbi összefüggés írja le.[3]:

$$\mu_i = f \frac{\eta_{1i}}{\eta_{3i}} \quad (2.1)$$

$$\nu_i = f \frac{\eta_{2i}}{\eta_{3i}} \quad (2.2)$$

Képpont dinamikája

Maga az optikai áramlás a jellemzőpontnak a képsíkon értelmezett sebességeként értelmezhető. Tehát ha a korábbi lyukkamera-modell egyenletét idő szerint deriváljuk, azzal lényegében meghatározzuk az optikai áramlást. A deriválás elvégzése után:

$$\begin{bmatrix} \dot{\mu}_i \\ \dot{\nu}_i \end{bmatrix} = \frac{f}{\eta_{3i}^2} \begin{bmatrix} \eta_{3i} & 0 & -\eta_{1i} \\ 0 & \eta_{3i} & -\eta_{2i} \end{bmatrix} \dot{\eta}_i^C \quad (2.3)$$

egyenletrendszer adódik [1], ahol $(\dot{\mu}_i \ \dot{\nu}_i)$ koordináta páros írja le az optikai áramlást, $\dot{\eta}_i^C$ pedig a jellemzőpontunk sebessége a kamera koordináta-rendszerben kifejezve. A $(\dot{\mu}_i \ \dot{\nu}_i)$ koordináta páros értékei meghatározhatók a kamera által szolgáltatott képi adatokból, amennyiben két egymást követő képen követni tudjuk a jellemzőpontunkat. Ebben az esetben a képsíkon egyértelműen meghatározható a pozíciója, és numerikus algoritmusok segítségével jól becsülhető az optikai áramlás.

Az i -ik jellemzőpontunk sebessége meghatározható úgy, hogy a ΔC vektor és a η_i^C vektor összegének a sebességekkel és szögsebességekkel vett vektori szorzatát vesszük.[3]

$$\dot{\eta}_i^C = \left[-I_3 \mid [\Delta^C \times] + [\eta_i^C \times] \right] \begin{bmatrix} V^C \\ \omega^C \end{bmatrix} \quad (2.4)$$

A negatív előjelű 3×3 -as egységmátrix ($-I_3$) abból adódik, hogy amilyen irányba mozgunk a kamerával, minden jellemzőpont azzal ellentétes irányba fog mozogni a kamera képsíkján. Tehát például, amikor a drónunkkal jobbra forgunk, akkor a jellemzőpontjaink balra fognak mozogni.

Egyenletek

Ha a 2.4 egyenletet behelyettesítjük a 2.3 egyenletbe, és kifejtjük, akkor az alábbi egyenletrendszert kapjuk:

$$\begin{aligned} \dot{\mu}_i &= \frac{\eta_{2i} f r^C}{\eta_{3i}} - \frac{f u^C}{\eta_{3i}} - f q^C + \frac{\eta_{1i} f w^C}{\eta_{3i}^2} - \frac{f q^C z^C}{\eta_{3i}} + \frac{f r^C y^C}{\eta_{3i}} - \frac{\eta_{1i}^2 f q^C}{\eta_{3i}^2} + \frac{\eta_{1i} \eta_{2i} f p^C}{\eta_{3i}^2} \\ &\quad + \frac{\eta_{1i} f p^C y^C}{\eta_{3i}^2} - \frac{\eta_{1i} f q^C x^C}{\eta_{3i}^2} \\ \dot{\nu}_i &= f p^C - \frac{f v^C}{\eta_{3i}} - \frac{\eta_{1i} f r^C}{\eta_{3i}} + \frac{\eta_{2i} f w^C}{\eta_{3i}^2} + \frac{f p^C z^C}{\eta_{3i}} - \frac{f r^C x^C}{\eta_{3i}} + \frac{\eta_{2i}^2 f p^C}{\eta_{3i}^2} - \frac{\eta_{1i} \eta_{2i} f q^C}{\eta_{3i}^2} \\ &\quad + \frac{\eta_{2i} f p^C y^C}{\eta_{3i}^2} - \frac{\eta_{2i} f q^C x^C}{\eta_{3i}^2} \end{aligned} \quad (2.5)$$

ahol $(p^C \ q^C \ r^C)$ és $(u^C \ v^C \ w^C)$ rendre a drón kamerarendszerbe transzformált szögsebesség és sebesség komponensei $(x^C \ y^C \ z^C)$ pedig a kamerarendszer testrendszerben lévő pozícióját leíró vektor koordinátái, amelyet áttranszformáltunk kamerarendszerbe. Továbbá $(\eta_{1i} \ \eta_{2i} \ \eta_{3i})$ a jellemzőpontunk kamerarendszerbeli helyzetét leíró vektor koordinátái, $(\mu_i \ \nu_i)$ a jellemzőpontunk kamera képsíkján értelmezett koordinátái és a $(\dot{\mu}_i \ \dot{\nu}_i)$ a jellemzőpontunk sebessége a kamera képsíkján.

Ez az egyenletrendszer $6+3N$ ismeretlen tartalmaz, amelynek megoldására $2N$ egyenlet szolgál. Azonban, ha a lyukkamera-modellből kifejezzük a η_{1i} és η_{2i} jellemzőpont pozíció koordinátáit, akkor csökkenthetjük az ismeretleneink számát.[3]

$$\eta_{1i} = \frac{\mu_i f}{\eta_{3i}} \quad (2.6)$$

$$\eta_{2i} = \frac{\nu_i f}{\eta_{3i}} \quad (2.7)$$

$$\begin{aligned}
\dot{\mu}_i &= r^C \left(\nu_i + \frac{f y^C}{\eta_{3i}} \right) - q^C \left(f + \frac{\mu_i^2}{f} + \frac{f z^C}{\eta_{3i}} + \frac{\mu_i x^C}{\eta_{3i}} \right) \\
&\quad + p^C \left(\frac{\mu_i y^C}{\eta_{3i}} + \frac{\mu_i \nu_i}{f} \right) + \frac{\mu_i w^C}{\eta_{3i}} - \frac{f u^C}{\eta_{3i}} \\
\dot{\nu}_i &= p^C \left(f + \frac{\nu_i^2}{f} + \frac{f z^C}{\eta_{3i}} + \frac{\nu_i y^C}{\eta_{3i}} \right) - r^C \left(\mu_i + \frac{f x^C}{\eta_{3i}} \right) \\
&\quad - q^C \left(\frac{\nu_i x^C}{\eta_{3i}} + \frac{\mu_i \nu_i}{f} \right) - \frac{f v^C}{\eta_{3i}} + \frac{\nu_i w^C}{\eta_{3i}}
\end{aligned} \tag{2.8}$$

Ezek után további feltételezésekkel/elhanyagolásokkal élünk. Jelen vizsgálatunkban feltételezzük, hogy a drón a repülés során közelítőleg állandó magasságon repül és mivel a kamera koordináta-rendszer z_c tengelye a NED koordináta-rendszer *down* tengelyével egybeesik – mert a gimbal folyamatosan függőleges helyzetben tartja – a w^C sebesség komponens zérusnak tekinthető ($w^C = 0$).

Az u^C és v^C komponensek nem zérusok, hiszen a drónunk nem csak előre (u^C) tud haladni, hanem oldalazó mozgást (v^C) is tud végezni. Ez egy fontos különbség a merevszárnyas drón és a multikopter között (merevszárnyas drónnál $v^C = 0$ feltételezéssel élhetünk). Továbbá feltételezzük, hogy a legyezési szögsebességen (r^C) kívül a másik kettő szögsebesség értékünk zérusértékű ($p^C = 0$, $q^C = 0$). Ezen felül, mivel a kameránk közvetlenül a súlypont alá van elhelyezve függőleges irányban, azaz z_c koordináta-tengely mentén, az egyenleteinkben szereplő x^C és y^C értékek zérusnak tekinthetők (lásd ΔB vektor).

Ezeket beépítve az egyenletrendszerünkbe:

$$\dot{\mu}_i = \nu_i r^C - \frac{f u^C}{\eta_{3i}} \tag{2.9}$$

$$\dot{\nu}_i = -\mu_i r^C - \frac{f v^C}{\eta_{3i}} \tag{2.10}$$

Az így kapott egyenletrendszer már csak $3+N$ ismeretlent tartalmaz $2N$ egyenletre, így ez az egyenletrendszer már három darab jellemzőpontra számolt sikeres optikai áramlás esetén eredményre vezet.

Az egyenletrendszer megoldása

A (2.9) és (2.10) által alkotott egyenletrendszer egy nemlineáris egyenletrendszer, ahol az ismeretleneink r^C , u^C , v^C , η_{3i} . Ha figyelembe vesszük, hogy $u^C = kv^C$, akkor az alábbi egyenlet írható fel:

$$\frac{f u^C}{\eta_{3i}} = k \frac{f v^C}{\eta_{3i}} \tag{2.11}$$

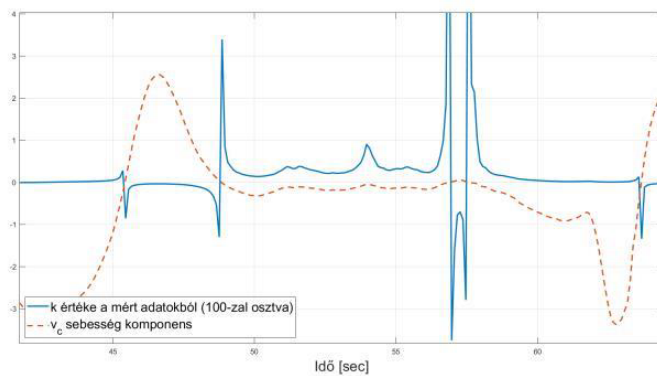
Ha a (2.11) egyenlet bal oldalát kifejezzük a (2.10) egyenletből és behelyettesítjük a (2.11) egyenletbe, majd ezt behelyettesítjük a (2.9) egyenletbe, akkor a következő összefüggést kapjuk:

$$\dot{\mu}_i = \nu_i r^C + k \mu_i r^C + k \dot{\nu}_i \tag{2.12}$$

Ez egy bilineáris egyenletrendszer, amelyben már az η_{3i} ismeretlen nem szerepel. Ez az egyenlet egy iterációs módszer segítségével megoldható. Az első iterációban k -t nullának tekintjük, majd a legyezési szögsebességet (r^C) kiszámoljuk a legkisebb négyzetek módszerével. Az így kapott r^C értéket behelyettesítjük a (2.11) egyenletbe és a legkisebb négyzetek módszerével kiszámoljuk a k értékét. A kapott k értéket visszahelyettesítve a (2.11) egyenletbe újra számolunk egy r^C -t, amelyet visszahelyettesítve egy újabb k értéket számolunk, és így tovább, folytatva az iterálást. Egy fedélzeti alkalmazásban természetesen konvergencia kritérium használata elengedhetetlen a fölösleges számítások elkerülése végett, jelen esetben offline számolva 100 iteráció lett végigszámolva, amely elegendőnek bizonyult.

Szingularitásvizsgálat

A (2.11) egyenletben ($k = \frac{u^C}{v^C}$) szingularitáshoz vezethet, hogyha v^C nullával egyenlő, vagy nullához közeli értéket vesz föl. A 3. ábra a mért adatokból számolt k értéket mutatja, amelyben a v^C -nek zérus átmenetei nagy kilengéseket okoznak k értékében.

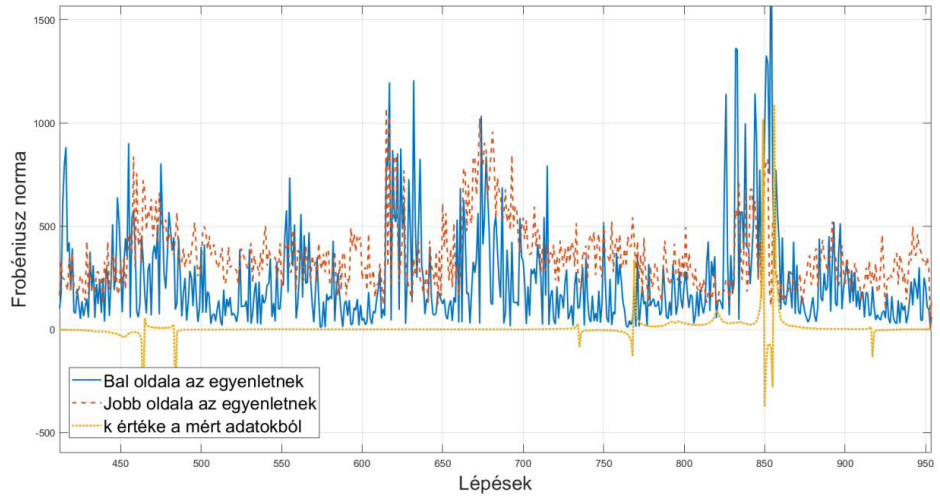


3. ábra: Mért szenzoradatokból számolt k érték
(saját szerkesztés)

A (2.10) egyenletben, hogyha v^C értéke zérus, abban az esetben ($\mu_i r^C + v_i = 0$) írható. Ha ezt behelyettesítjük a (2.12) átrendezett egyenletébe, akkor láthatóvá válik, hogy a (2.13) egyenlet mindkét oldalán zérus vektorok kell, hogy álljanak.

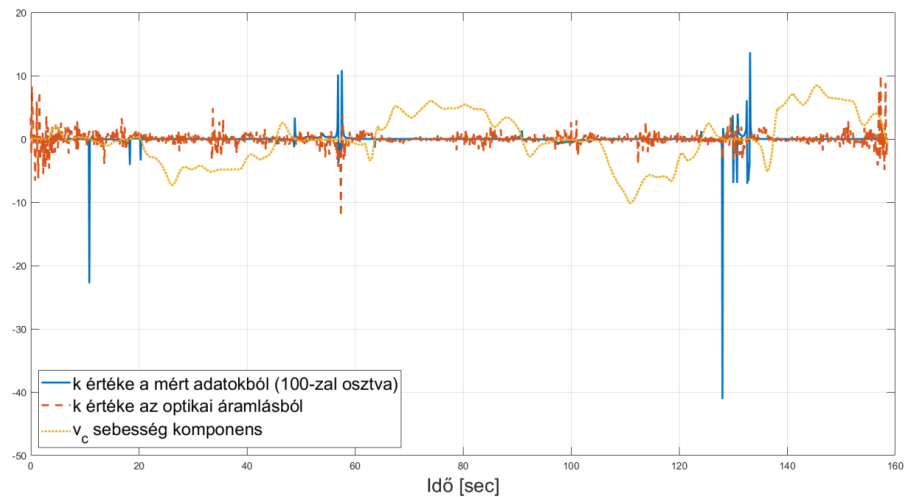
$$\dot{\mu}_i - v_i r^C = k (\mu_i r^C + v_i) \quad (2.13)$$

Ha kiszámoljuk mindkét oldal Frobeniusz-normáját, akkor eredményül azt kapjuk, hogy ezek nem lesznek zérus vektorok. Ez többnyire betudható a különböző zajoknak, amelyek adódhatnak a különböző pixelhibákból és pontatlan gimbszögekből. A 4. ábra mutatja a (2.13) egyenlet két oldalának Frobeniusz normáit. Ezen zajok miatt akkor sincsenek nulla közelében, ha v^C nulla, vagy közel nulla.



4. ábra: Frobenius normák a repülés egy szakaszában
(saját szerkesztés)

Köszönhetően ezeknek a zajoknak, a számításainkban szingularitás nem lépett fel. Ahogyan azt az 5. ábra mutatja, az iterációs módszer által számolt k értékeket a mért adatokból számolt k értékekkel közösen ábrázolva. Itt fontos megjegyezni, hogy utóbbi értéke 100-zal osztva van a jobb láthatóság végett. Ebből fakadóan szingularitásdetekcióra nem volt szükség.

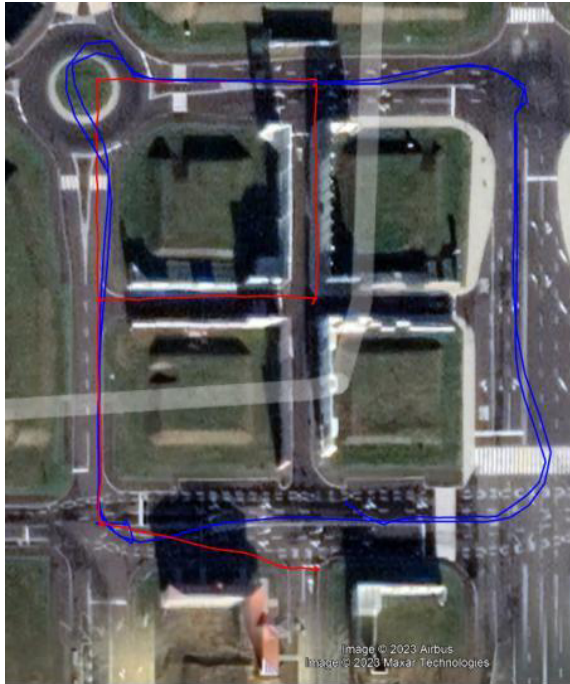


5. ábra: Optikai áramlásból és mért adatokból számolt k értékek
(saját szerkesztés)

Az optikai áramlás meghatározása

A cél a (2.12) egyenletben szereplő μ_i , v_i , μ_i , v_i értékek meghatározása. Bauer és Kun [3] munkájuk során Matlab-ban, merevszárnyas pilóta nélküli repülőgéppel szimulált repülési adatokat exportáltak Unreal-Carla környezetbe, amelyben több különböző közeget szimulálva (városi, erdő) kameraképeket készítettek, és alkalmazták a fentebb említett kétlépéses lineáris megoldót. Kutatásuk során sok problémát okoztak a hibásan párosított pontok, ezért saját adatsorunkban ezek kiszűrésére a megfordított optikai áramlást használtuk fel.

Repülési adatok



6. ábra: A felhasznált repülések útvonalai ZalaZONE Smart City felett
(saját szerkesztés)

Olyan valós kamera felvételeivel és valós szenzor adataival dolgoztunk, amelyeket két különböző repülés (Repülés1 és Repülés2) során készültek egy DJI M600-as típusú multikopterrel.[5] Az első repülés (Repülés1) gyorsabban ($\sim 6\text{m/s}$), egy nagy négyzet alakú útvonalat (kék útvonal) jár be hosszabb időtartamú ($\sim 6\text{sec}$) forgásokkal, mialatt a kamera látómezejében egy mozgó járművet követ. A második repülés (Repülés2) egy kicsit lassabban ($\sim 1\text{m/s}$), egy kis négyzet alakú utat (piros útvonal) jár be, rövidebb forgásokkal ($\sim 4\text{sec}$) és a kamera látómezejében nincsen mozgó jármű.

Oriented FAST and Rotated BRIEF

A jellemzőpontok detektálására és leírására az OpenCV könyvtárban implementált ORB (Oriented FAST and Rotated BRIEF) függvényt alkalmaztam. Ennek alternatívája lehetne a SIFT vagy SURF detekciós eljárás, de mivel ezek szabadalmaztatott eljárások, ezért az ORB detekcióra esett a választás. A Bauer és Kun [3] munkájában alkalmazott goodFeaturesToTrack (Shi-Tomasi sarokdetektor) ezeken a felvételeken már nem volt használható, mivel a 10FPS-sel készült valós képeken már nem volt képes elegendő számú jellemzőpontot találni. Az ORB függvény FAST sarokdetekciós eljárással egy többszintű képpiramis minden szintjén megkeresi a jellemzőpontokat, majd egy BRIEF (Binary Robust Independent Elementary Features) leíró használatával elkészíti a jellemzőpontunk leírását. A jellemzőpontok megtalálására egyszintű képpiramis lett alkalmazva, a jellemzőpontok 600-ban lettek maximalizálva.



7. ábra: ORB detektor jellemzőpontjai
(saját szerkesztés)



8. ábra: Shi-Tomasi sarokdetektor jellemzőpontjai
(saját szerkesztés)

A FAST algoritmus működésének lényege, hogy egy adott p pixel környezetében egy 16 pixelből álló 3 pixel sugarú kört vizsgál. A 16 pixel mindegyikét besorolja az alábbi két osztály egyikébe: világosabb, mint p plusz, vagy egy bizonyos határérték (t), sötétebb, mint p , vagy egy bizonyos határérték. Amennyiben legalább N pixele sötétebb, mint a $(p-t)$, vagy N pixele világosabb, mint $(p+t)$ akkor azt a pontot sarokpontként detektálja.

A BRIEF leíró a megtalált képpontokat egy bináris vektorba konvertálja, amely csak nullákat és egyeseket tartalmaz. Ennek nagy előnye, hogy míg a SURF és SIFT algoritmusok lebegőpontos számokat használnak, addig a BRIEF leírás kettes számrendszerbeli reprezentációt használ, amivel sok memóriát és feldolgozási időt megspórol.

Lucas és Kanade módszere

Miután meghatároztuk a jellemzőpontjainkat, ezeknek a pontoknak a mozgását szeretnénk követni az egymást követő képeken. Az optikai áramlás egy mozgásdetekciós eljárás, ami minden jellemzőponthoz meghatároz egy olyan vektort, amely arányos a pont mozgásával az egymást követő képeken.

Az OpenCV könyvtárban implementált OpticalFlowPyrLK függvénye 9x9-es vizsgálóablak mérettel, maximum 7 szintű képpiramissal és maximum 10 iterációval volt alkalmazva.



9. ábra: Kiszűrt jellemzőpontok
(saját szerkesztés)

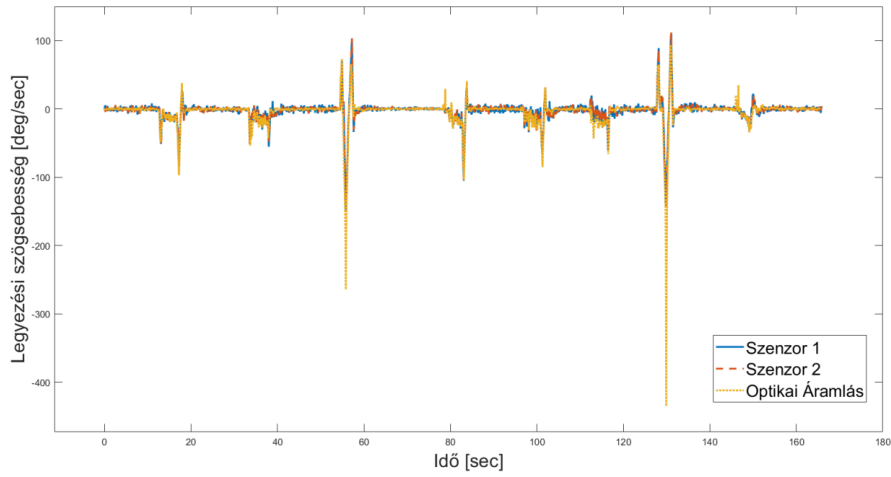
E módszer segítségével meg tudjuk határozni, hogy az ORB által a (k)-ik képen detektált pontok a (k+1)-ik képen hol találhatóak. Vannak azonban esetek, amikor ezek a párosítások hibásan történnek meg, mert változik a pixelintenzitás, vagy túl nagyok a mozgások. E probléma feloldására a (k+1)-ik képen párosított jellemzőpontokat visszaszámoljuk Lucas és Kanade módszerével a (k)-ik képre. Ezt követően az ORB által és a megfordított optikai áramlás által számolt jellemzőpontok koordinátái között egy Euklideszi távolságot számolunk. Erre az Euklideszi távolságra bevezethetünk egy határértéket, amit, ha átlép, akkor az adott pontot már ne használjuk fel a későbbi számításainkban. A 9. ábrán kék színnel láthatóak a kiszűrt pontok.

Miután meghatároztuk a pontok helyét a (k+1)-ik képen, és a rossz párosításokat is kiszűrtük, az optikai áramlás sebessége numerikus deriválás segítségével számolható [3]:

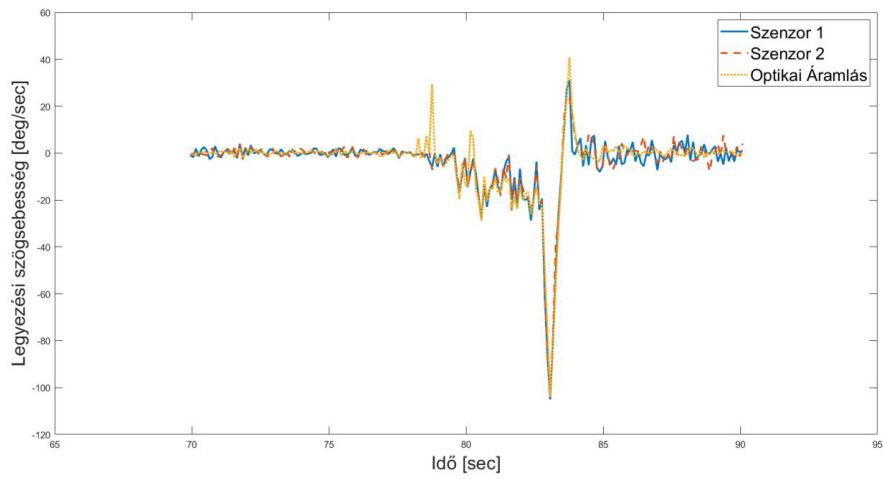
$$\dot{\mu}_i \approx \frac{\mu_i(k+1) - \mu_i(k)}{\Delta t}, \quad \dot{\nu}_i \approx \frac{\nu_i(k+1) - \nu_i(k)}{\Delta t}$$

ahol Δt a két egymást követő képkocka között eltelt időt jelöli.

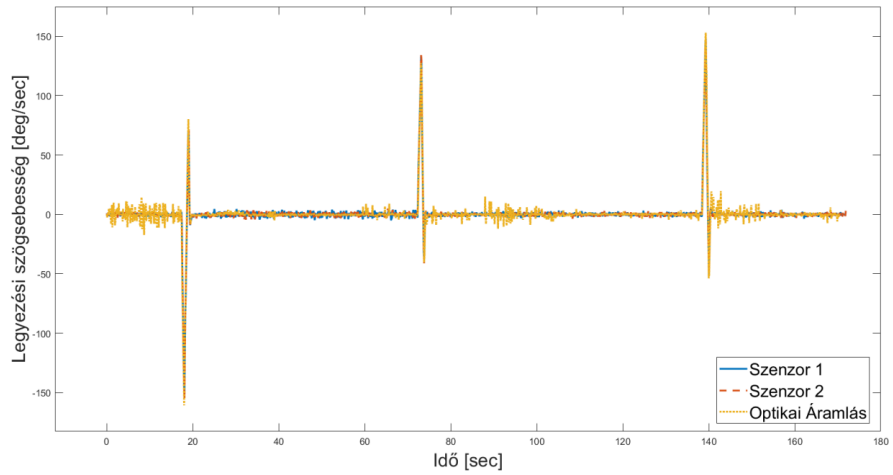
Ezzel a (2.12) egyenletben szereplő a $\dot{\mu}_i$, $\dot{\nu}_i$, μ_i , ν_i értékek meghatározásra kerültek. Amennyiben ezeket behelyettesítjük a (2.12) egyenletbe, és megoldjuk a fent bemutatott módon, az alábbi ábrákon látható eredményeket kapjuk.



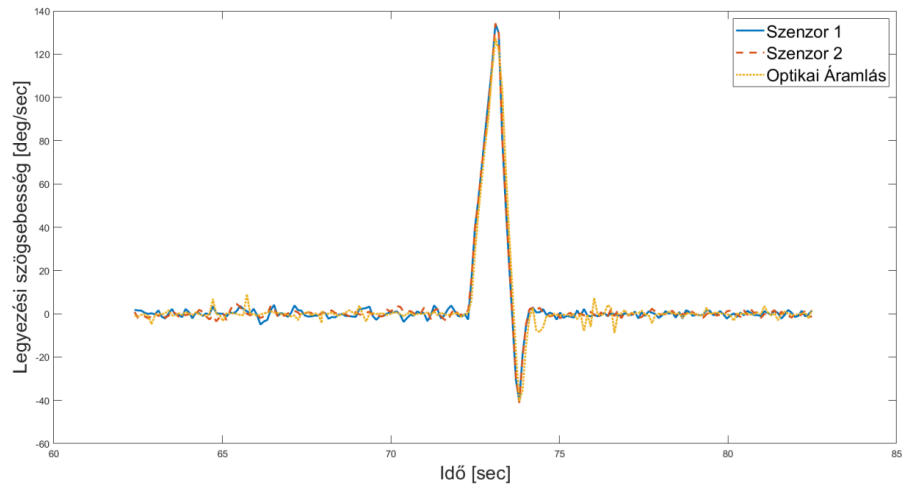
10. ábra: Az 1-es repülés mért és becsült szögsebességei
(saját szerkesztés)



11. ábra: Az 1-es repülés mért és becsült szögsebességei 70s és 90s közötti időtartományban
(saját szerkesztés)



12. ábra: A 2-es repülés mért és becsült szögsebességei
(saját szerkesztés)



13. ábra: A 2-es repülés mért és becsült szögsebességei 62s és 84s közötti időtartományban
(saját szerkesztés)

Ez a módszer korrekt becsléseket adott a legyezési szögsebességekre, de a 12. ábrán látható, hogy a két nagy sebességű forgásnál (58s és 110s) jelentősen túlbecsüli a szögsebességeket, viszont ez kiküszöbölhető, ha ismeretes, mekkora maximális szögsebesség értékekre van korlátozva a drónunk.

Szenzorhiba-detekció

Szenzormeghibásodások fajtái

Guillermo Heredia és társai [6] munkájukban részletesen bemutatják egy autonóm modellhelikopter jellemző szenzormeghibásodásait. Erre és [3] munkára alapozva az alábbi négyféle szenzorhibát vizsgáljuk:

1. Szenzor teljes meghibásodása: A hiba beálltának időpillanatától kezdődően zérus kimenetet ad a szenzor.
2. Szenzor konstans értéken történő beragadása: A szenzor kimenetén a hiba beálltának időpillanatától fogva, a hiba beálltának időpillanatában érvényes konstans érték fog csak megjelenni.
3. Additív típusú szenzormeghibásodás: A szenzor által mért értékekre egy konstans érték szuperponálódik.
4. Multiplikatív típusú szenzormeghibásodás: A szenzor kimenetén a mért értékek $100+P\%$ -a fog megjelenni.

Hibadetekciós eljárás

A DJI M600 drón három, egymástól független szögsebességszenzorról van felszerelve, amelyből kettőt (Szenzor1 és Szenzor2) használunk. A fentebb felsorolt hibákat Szenzor2 adatain mesterségesen állítjuk elő, Szenzor1 adatait érintetlenül hagyjuk. A hangolás és a tesztelés során Repülés1 és Repülés2 adatait használjuk. Ahmet Arda Ozdemir és társai [7] munkájuk során egy számlálót és határértékeket használó hibadetekciós eljárást használnak szélturbinák részegységei meghibásodásának előrejelzésére, amelyben a módszer alapja, hogy az optikai áramlásból becsült legyezési szögsebesség r_{OF} és a szenzorok által mért szögsebesség különbségét képezzük az alábbi reziduált kapva.

$$e_{1/2} = r_{szenzor1/2} - r_{OF} \quad (4.1)$$

Ha ez az eltérés nagyobb, mint egy bizonyos határérték ($To1$), akkor egy számláló értékét növeljük egy előre meghatározott értékkel (U), ha kisebb, csökkentjük egy előre meghatározott értékkel (D). Ezeknek az értékeknek nem kell szükségszerűen azonosnak lenniük. Végül erre a számlálóra is be-vezetünk egy határértéket ($To2$) és ha a számláló átlépi ezt az értéket, a szenzort hibásnak tekintjük.

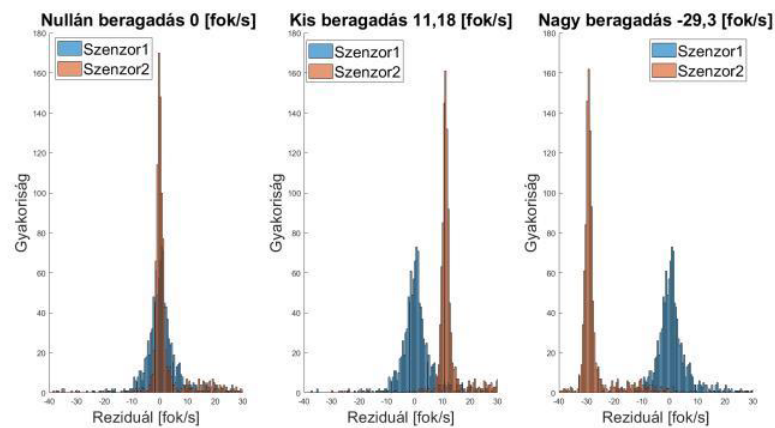
Határértékek hangolása

A $To1$ határérték adja meg a hibadetektálás érzékenységét, azaz, hogy mekkora eltérést tudunk maximálisan detektálni, a $To2$ pedig segíti a zajokból eredő téves riasztások elnyomását. Ezért előnyös a fel/le számlálót és határértékeket alkalmazó megoldás, mert ennek segítségével akár kisebb hibákat is észlelhetünk, mint egy egyszerű küszöbértékkel.

A határértékek hangolásánál nekünk egy kompromisszumot kell kötnünk, mivel a reziduálon lévő zajok téves riasztásokat (false alarm (FA)) okozhatnak, ha a zajok $Tol1$ küszöbértéknél nagyobbak, azaz hibát jelzünk akkor is, amikor valójában nincs szenzorhiba. Másik kedvezőtlen eset, amikor a szenzorhibánk mértéke kisebb, mint $Tol1$. Ebben az esetben tévesztést (missed detection (MD)) fog okozni, azaz nem jelzünk hibát, amikor a szenzorunk hibás. Tehát lényegében $Tol1$ meghatározása egy olyan optimalizálási feladat, melyben a téves riasztás és tévesztés arányát kell gondosan megválasztani. Gróf Tamás és Bauer Péter [8] cikkükben egy hasonló optimalizálási feladatot oldanak meg.

Ez alapján definiálunk egy valószínűséget ($P(\text{data} > Tol1 | \text{no error})$), mely azt fejezi ki, hogy a reziduálunk hány alkalommal lépi át a $Tol1$ küszöbértéket, mikor hibamentes a szenzorunk (FA), illetve definiálunk egy másik valószínűséget $P(\text{data} < Tol1 | \text{error})$, amikor azt vizsgáljuk, hogy a reziduálunk hányszor nem lépi át a $Tol1$ értéket egy hibás szenzoradatból generált különbség esetén (MD).

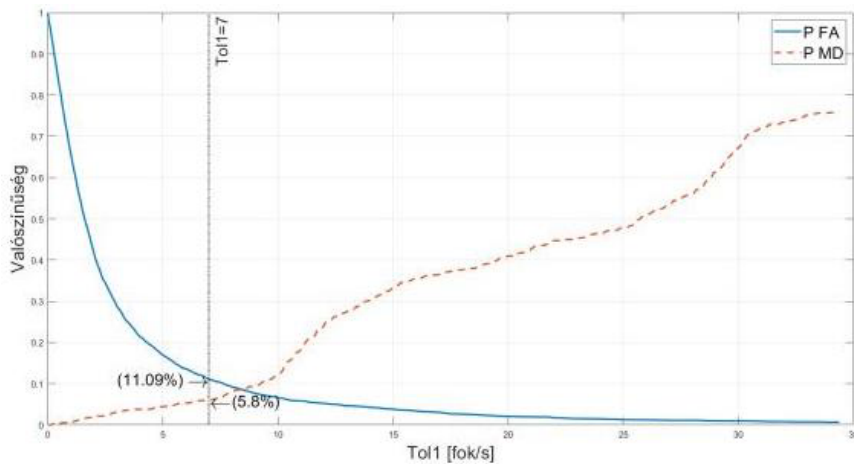
A $P(\text{data} < Tol1 | \text{error})$ számolásához a szenzoradatokon egy kis és egy nagy konstans értéken történő beragadás típusú szenzorhibát idéztem elő mindkét repülésre külön-külön. A kisértékű beragadáshoz repülés1: $11.18^\circ/\text{s}$ és repülés2: $-9^\circ/\text{s}$, a nagyobb értékű beragadáshoz repülés1: $-29.3^\circ/\text{s}$ és repülés2: $23.45^\circ/\text{s}$ értékeken lett megakasztva a szögsebesség.



14. ábra: Repülés1 reziduál hisztogramjai különböző hibák esetén (Szenzor1 nem tartalmaz hibát, szenzor2 hibával terhelt) (saját szerkesztés)

Teljes szenzormeghibásodás (nulla értéken történő megakadás) a hangolás során nem lett figyelembe véve, mivel a reziduálja nem megkülönböztethető egy hibamentes szenzor reziduáljától. Ez jól látható 14. ábrán, ahol a teljes szenzormeghibásodás reziduáljának a hisztogramja közel teljes fedésbe van a hibátlan szenzor reziduáljának hisztogramjával. Amikor nincsen forgás, a nagyon kis értéken történő megakadások észlelését a mozgás hiánya megakadályozhatja, ezzel szemben ekkor a zajok képesek téves riasztásokat előidézni. Ennek a kiküszöbölésére csak akkor számoljuk a $P(\text{data} < Tol1 | \text{error})$ és $P(\text{data} > Tol1 | \text{no error})$ valószínűségeket, amikor a mért jelünk $Tol3 = 4^\circ/\text{s}$ feletti értéket vesz föl.

Ezt az értéket úgy határoztuk meg, hogy a 14. ábrán látható görbéket különböző $To13$ értékekre kiszámoltuk, majd figyelembe véve a reziduálon lévő zajok mértékét, $To13 = 4^\circ/s$ értéket választottuk.



15. ábra: A valószínűségek különböző küszöbértékek mellett
(saját szerkesztés)

A $P(data > To1|no\ error)$ valószínűséget úgy határozzuk meg, hogy összeszámoljuk, mindkét repülésen hányszor haladja meg a $To1$ értékét a hibamentes reziduál, majd elosztjuk az összes időlépés számával. $P(data < To1|error)$ valószínűségeket úgy tudjuk meghatározni, hogy összeszámoljuk, hányszor nem lépik át a fentebb hibákat magukba foglaló reziduálok a $To1$ értékét, majd elosztjuk az összes időlépéssel, amelyekben a hibák fennálltak. Ezt megismételve különböző $To1$ értékekre $0^\circ/s$ és $34^\circ/s$ között a 15. ábrának megfelelő görbéket kapjuk. Ez az ábra szemlélteti, hogy a határérték növelésével a téves riasztások aránya csökken, míg a tévesztések aránya nő. Esetünkben a két arány körülbelül $8.37^\circ/s$ -nál egyenlő, ennél kicsit alacsonyabb, $To1 = 7^\circ/s$ határértéket választottunk, mert így a téves riasztások aránya csökkenthető.

A $To2$ határérték fogja eldönteni számunkra, hogy az adott szenzorunk meghibásodott, vagy sem. A döntéseinket 4 különböző csoportban osztályozhatjuk:

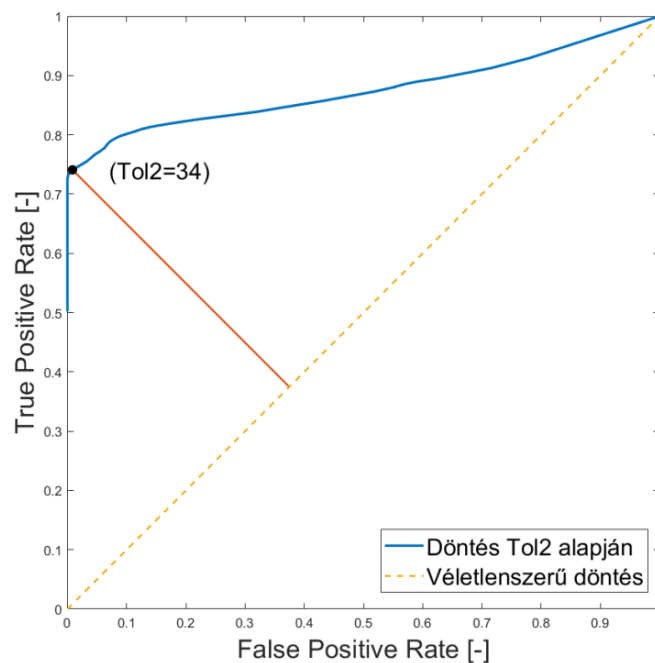
1. True Positive (TP): a szenzor hibás és hibásnak ítéljük.
2. False Positive (FP): a szenzor nem hibás és hibásnak értékeljük.
3. True Negative (TN): a szenzor nem hibás és nem hibásnak is osztályozzuk.
4. False Negative (FN): a szenzorunk hibás és nem hibásnak minősítjük.
 - a. $To2$ értékét a Receiver Operating Characteristic (ROC) görbe segítségével került meghatározásra [8]. Az ROC görbe a hamis pozitív arányának (FPR) függvényében ábrázolja a valós pozitív arányát (TPR) különböző küszöbértékek mellett.
 - b. False Positive észlelések aránya a hibamentes jel fel/le számlálóján a $To2$ határt meghaladó értékek összege osztva a False Positive és a True Negative észlelések számának összegével.

$$FPR = \frac{FP}{FP + TN} \quad (4.2)$$

A True Positive döntések arányának meghatározásához az eredeti jelünkön 25 különböző időpillanatban konstans értéken történő beragadás-típusú szenzorhibát idéztünk elő, mindkét repülésen külön-külön. Mind az 50 különböző esetben megszámláljuk, hogy a számláló hányszor lépi át a küszöbértéket, majd elosztjuk a True Positive és a False Negative észlelések számának összegével.

$$TPR = \frac{TP}{TP + FN} \quad (4.3)$$

Mind az FPR, mind a TPR a korábban már meghatározott $Tol1 = 7^\circ/s$ és $Tol3 = 4^\circ/s$ továbbá, $U = 1$ $D = 1$ paraméterekkel lett meghatározva. Ha a (4.2) és (4.3) egyenleteknek megfelelő számolásokat elvégezzük 0-tól 100-ig terjedő $Tol2$ értékre, a 4.3 függvényben szerinti ROC görbét kapjuk eredményül.



16. ábra: ROC görbe az optimális Tol2 érték meghatározásához
(saját szerkesztés)

Az ideális esetet a 16. ábra bal felső sarkában lévő (0,1) pont jelölné, amely azt jelentené, hogy a hibás szenzort mindig sikerült felismerni, és egyszer sem ítéltük hibásnak a hibátlanul működő szenzort. Azt szeretnénk, hogy a hibadetekciós eljárásunk a lehető legjobban közelítse meg ezt a pontot, azaz a True Positive döntések aránya a lehető legmagasabb, míg a False Positive döntések aránya a lehető legalacsonyabb legyen.

Ezt az optimális értéket meghatározhatjuk úgy, hogyha az átlós vonaltól (amely a véletlenszerű döntést reprezentálja) a legnagyobb mérőleges távolságra lévő pontot megkeressük. Ez esetünkben a $ToI2 = 34$ érték, ahol False Positive döntések aránya ($F P R = 0.0085$) és True Positive döntések aránya ($T P R = 0.7408$).

A hibadetektálás eredményei

Az előzőekben meghatározott paraméterekkel ($ToI1 = 7^\circ/s$ $ToI2 = 34$ $ToI3 = 4^\circ/s$ $U = 1$ $D = 1$) felhangolt hibadetektációs tesztelése két különböző repülésen két különböző meghibásodási időpillanatban történt. Egyik, amikor a szögsebesség közel konstans értéket vesz fel (konstans), másik a forgások kezdetén a szögsebességben történő nagyobb változásoknál (változó).

Hiba típusa	Repülés1 Változó	Repülés1 Konstans	Repülés2 Változó	Repülés2 Konstans
Szenzor teljes meghibásodása	2.17s	5.367s	67.4377s	69.5392s
Konstans értéken beragadás	13.795s	5.367s	67.6409s	69.5392s
Additív típusú hiba	55.677s	59.677s	X	X
Multiplikatív típusú hiba	51.267s	55.267s	67.5392s	69.5392s

1. táblázat: Hibadetektálási időtartamok

Az 1. táblázat foglalja össze a hiba beálltának időpontjától a hibadetektálásáig terjedő időtartamokat sikeres hibadetektáció esetén, X jelöli a hibás szenzor detektálásának sikertelenségét.

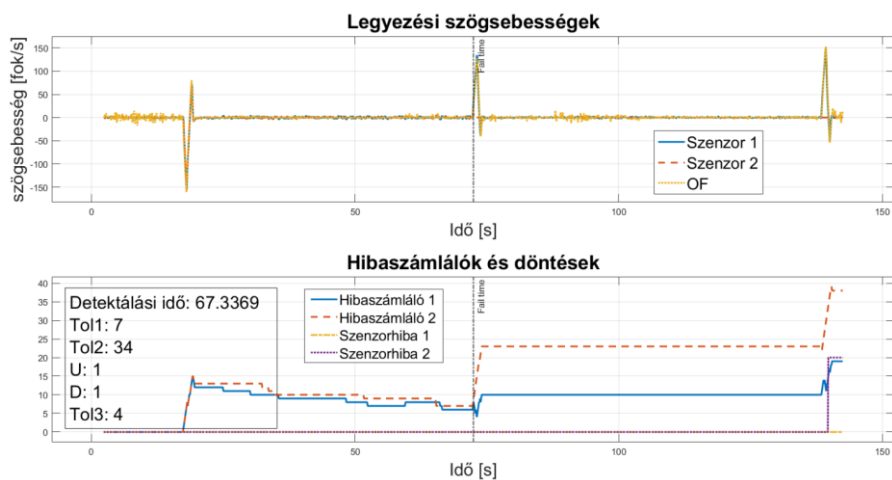
Additív típusú szenzormeghibásodás esetén ez a módszer ilyen hangolással $A = 2.1^\circ/s$ és ennél nagyobb hozzáadott konstans értékeket képes detektálni az 1-es repülés esetében.

Multiplikatív típusú szenzormeghibásodás esetén $P=8\%$ és nagyobb százalékok esetén képes detektálni mindkét repülés esetén. Az 1. táblázatban szereplő értékek a legkisebb A és P értékekhez tartoznak, amelyeket a módszer képes volt detektálni. Természetesen nagyobb eltérések esetén ezek a detektálási időtartamok fokozatosan rendre csökkennek.

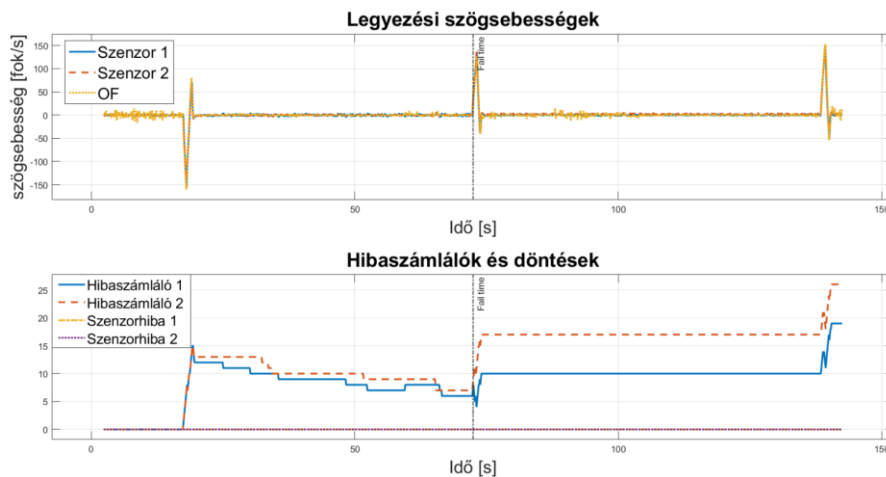
Additív típusú szenzormeghibásodás esetén a 2-es repülés hibás szenzorának detekciója nem volt sikeres, mert a hibaszámláló nem érte el $ToI2$ értékét a rendelkezésre álló időtartam alatt a rövid időtartamú forgások és a 10Hz-es mintavételezés miatt. A nagyon hosszú detektálási időtartamokat a rövid forgások okozzák, és így ez a módszer csak a következő forgásnál fogja tudni detektálni a hibás szenzort. Ez jól látható 18. ábrán, ahol a két forgás között közel 60 másodperc telik el. A 17. ábrán pedig a leggyorsabb detektálás látható.



17. ábra: Repülés1 teljes szenzormeghibásodással (saját szerkesztés)



18. ábra: Repülés2 teljes szenzormeghibásodással (saját szerkesztés)



19. ábra: Repülés2 additív szenzormeghibásodással
(saját szerkesztés)

Konklúzió

Kísérleteink igazolták, hogy a szögsebességszenzorok hibái hatékonyan detektálhatók és korrigálhatók a drónok kameráinak segítségével. Az optikai áramlás és a szögsebességek kapcsolatát leíró egyenletek bevezetése után az egyenletrendszert aktualizáltuk egy gimballal stabilizált függőlegesen lefelé néző kamera esetére. Az egyenletrendszer megoldását követően bemutattuk a szükséges képfeldolgozási eljárást. A szenzorhibák detektálására egy hibaszámlálót és küszöbértékeket használó módszert alkalmaztunk, amelynek hangolása a téves riasztások és a tévesztések rátájának figyelembevételével, illetve ROC görbe használatával történt. A valós repülési adatokban mesterségesen hibákat generáltunk, és ezeket a szenzoradatokat az optikai áramlásból becsült szögsebesség értékkel összehasonlítva detektáltuk. A kidolgozott módszer a legtöbb vizsgált szenzorhibát képes volt észlelni. Additív hiba esetén $2.1^\circ/s$ felett, multiplikatív hibáknál 8% feletti értékekre, utóbbiakat hosszabb időtartamra. Módszerünk a rövid időtartamú, gyors forgások esetén még nem bizonyult elég hatékonynak, a nagyon kis értéken történő beragadásokat, $2.1^\circ/s$ alatt nem volt képes detektálni. Jelen munkában csak a legyezési szögsebesség becslésével foglalkoztunk, azonban a hatékonyabb hibadetektálás érdekében a jövőben a többi szögsebesség-komponens becslésével tervezzük a módszert kiegészíteni egy merevszárnyas pilóta nélküli légi járművön, nem tömegközéppontba helyezett ferdén lefelé néző kamera képe és valós repülési adatok felhasználásával. A kis értéken történő beragadások detektálása a szögsebességekből kumulált szögelfordulás számításával és a kameraképekből a szögelfordulás meghatározásával válhat lehetségessé.

Felhasznált irodalom:

- [1] KEHOE, Joseph et al.: State Estimation Using Optical Flow from Parallax-Weighted Feature Tracking. AIAA Guidance, Navigation, and Control Conference and Exhibit. doi: 10.2514/6.2006-6721. <https://arc.aiaa.org/doi/abs/10.2514/6.2006-6721> (Letöltés ideje: 2023. 11. 27.)
- [2] KUN Szabolcs – BAUER Peter: Improvements in Optical Flow-based Aircraft Partial State Estimation. 2021 17th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA), 2021. szept., pp. 1-4. doi: 10.1109/CNNA49188.2021.9610822 (Letöltés ideje: 2023. 11. 27.)
- [3] BAUER Peter – KUN Szabolcs: Optical flow-based angular rate sensor fault detection on UAVs*. IFAC-PapersOnLine 55.14 (2022). 11th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2022, pp. 46-51. ISSN: 2405-8963. doi: 10.1016/j.ifacol.2022.07.581. <https://www.sciencedirect.com/science/article/pii/S2405896322009922> (Letöltés ideje: 2023. 11. 27.)
- [4] CZÚNI László – KATÓ Zoltán: Számítógépes látás. In: KATÓ Zoltán – CZÚNI László: Typotex. Egyetemi tananyag, 2011. <http://nbn.urn.hu/N2L?urn:nbn:hu-108199> (Letöltés ideje: 2023. 11. 27.)
- [5] BAUER Peter et. al.: Encounter Risk Evaluation with a Forerunner UAV. Remote Sensing 15.6 (2023). ISSN: 2072-4292. doi: 10.3390/rs15061512. <https://www.mdpi.com/2072-4292/15/6/1512> (Letöltés ideje: 2023. 11. 27.)
- [6] HEREDIA, G. et. al.: Detection of Sensor Faults in Autonomous Helicopters. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ápr., pp. 2229–2234. doi: 10.1109/ROBOT.2005.1570444 (Letöltés ideje: 2023. 11. 27.)
- [7] OZDEMIR, Ahmet Arda – SEILER, Peter – BALAS, Gary J.: Wind Turbine Fault Detection Using Counter-Based Residual Thresholding. IFAC Proceedings 2011/1. 18th IFAC World Congress, pp. 8289-8294. ISSN: 1474-6670. <https://doi.org/10.3182/20110828-6-IT-1002.01758>. <https://www.sciencedirect.com/science/article/pii/S1474667016449426> (Letöltés ideje: 2023. 11. 27.)
- [8] GROF, Tamas – BAUER, Peter: Voting-based Fault Detection for Aircraft Position Measurements with Dissimilar Observations. This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. The research presented in this paper was funded by the Higher Education Institutional Excellence Program. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 690811 and the Japan New Energy and Industrial Technology Development Organization under grant agreement No. 062600 as a part of the EU/Japan joint research project entitled 'Validation of Integrated Safety-enhanced Intelligent flight control (VISION)'. IFAC-PapersOnLine 2020/2., 21st IFAC World Congress, pp. 14724-14729. ISSN: 2405-8963. doi: 10.1016/j.ifacol.2020.12.1841. <https://www.sciencedirect.com/science/article/pii/S2405896320324551> (Letöltés ideje: 2023. 11. 27.)