

Országos Tudományos Diákköri Konferencia

Drónokon alkalmazott optikai áramlás alapú legyezési szögsebességbecslés és hibadetektálás

Szerző:

Jevuczó Gábor



Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedés- és Járműirányítási Tanszék

Konzulens:

Dr. Bauer Péter

2024. május 2.

Tartalomjegyzék

1. Bevezető	1
2. Optikai áramlás egyenletei a kamera koordináta-rendszerében	3
2.1. Koordináta-rendszerek	3
2.2. Transzformációk	4
2.3. Lyukkamera modell	5
2.4. Képpont dinamikája	6
2.5. Egyenletek	7
2.6. Egyenletrendszer megoldása	9
2.7. Szingularitás vizsgálat	9
3. Optikai áramlás meghatározása	11
3.1. Repülési adatok	11
3.2. Oriented FAST and Rotated BRIEF	12
3.3. Lucas és Kanade módszere	13
4. Szenzor hibadetekció	16
4.1. Szenzor meghibásodások fajtái:	16
4.2. Hibadetekciós eljárás	16
4.3. Határértékek hangolása	17
4.4. A hibadetektálás eredményei	21
5. Összefoglalás	23

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
Tol1	-	Reziduál küszöbértéke
Tol2	-	Hibaszámláló küszöbértéke

1. fejezet

Bevezető

Napjainkban a drónok gyors elterjedése és méret növekedése 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 végett (pl.: felvétel készítése, videószemüveg) fel van szerelve kamerával, mely célszerűvé teszi annak használatát a szenzorhibák detektálására.

Kutatásom 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 kerültek az egyenletek, melyek a lyukkamera modell segítségével kapcsolatot teremtenek az optikai áramlás és a légijárművünk sebességei között. Ezeket az egyenletek korábban nem tömegközéppontba helyezett, előre, ferdén lefele néző kamerához vezették le merevszárnyas pilóta nélküli légijárművekre. Ezen egyenletek jelenlegi problémára történő aktualizálását mutatom be a dolgozat első részében.

A dolgozat második részében bemutatásra kerül az optikai áramlás meghatározása. 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, melyre szintén az optikai áramlást hívjuk segítségül. A meghatározott optikai áramlás és az előző részben levezetett egyenletek segítségével becsülhető a legyezési szögsebesség.

A dolgozat utolsó részében ismertetem a vizsgált szenzorhibákat, a hibadetekciós eljárást, annak hangolását és az alkalmazás eredményeit. 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 melyből egy tartalmazza a generált hibákat. Továbbá bemutatásra kerül egy számlálókát és határértékeket használó hibadetekciós eljárás. 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.

2. fejezet

Optikai áramlás egyenletei a kamera koordináta-rendszerében

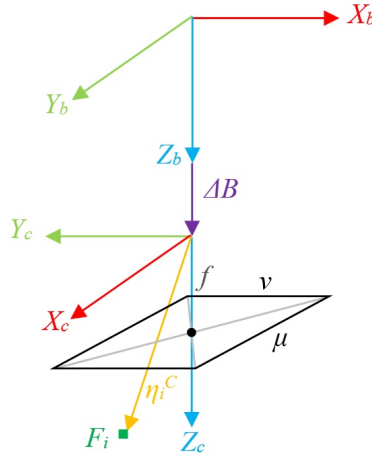
Ebben a fejezetben leírom a kapcsolatot a kamera képe és a környezetben érzékelt jellemzőpontok között. Az itt ismertetett egyenletek Joseph J. Kehoe és társai [1] cikkükben mutatták be ahol az egyenletrendszer megoldásához különböző módszereket is leírtak. Ezt felhasználva [2] Kun Szabolcs és Bauer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

2.1. Koordináta-rendszerek

Ahhoz, hogy megértsük, hogy a járművünk hogyan mozog a környezetünkben érzékelt jellemző pontokhoz képest, először szükséges a koordinátarendszerek meghatározása. A felhasznált koordinátarendszereket az 2.1 ábra mutatja. Esetünkben három fő koordinátarendszert használunk:

- Föld koordináta-rendszer (Earth):

Más néven North-East-Down koordinátarendszer. Ezen koordinátarendszer a föld felszínének egy pontjához van kötve melynek 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).



2.1. ábra. Koordináta-rendszerek.

- Test koordináta-rendszer (Body):

A test koordinátarendszer a légi járművünk tömegközéppontjához (súlypont) van kötve, melynek x_b tengelye a jármű orra felé néz, y_b tengelye a jármű jobb oldala felé mutat, a z_b tengelye pedig lefelé mutat.

- Kamera koordináta-rendszer (Camera):

A kamerához kötött koordinátarendszer. 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 jármű hátulja felé, z_c tengelye pedig lefelé mutat. Pozíciója a test rendszerben a ΔB vektorral van leírva, amely mutatja, hogy a kamera jó közelítéssel a súlypont alatt lefelé, függőlegesen helyezkedik el ($\Delta B = [0.0285 \ -0.0115 \ 0.175] \ [m]$).

2.2. Transzformációk

Mivel célunk, hogy a későbbiekben az egyenletek egyszerűbb formáját kapjuk ennek okán a kamerarendszerben írjuk fel az egyenleteinket. Ehhez szükséges a testrendszerben értelmezett sebességvektorainak, szögsebességvektorainak illetve a ΔB vektorának kamerarendszerbe történő transzformálása. Ezt a két koordinátarendszer 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 melynek 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.

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

A szögsebességeinket és sebességeinket a test koordináta rendszerben mérjük. Ahhoz, hogy kamerarendszerbe áttanszformáljuk először transzformálnunk kell a föld koordinátarendszerbe melyet az alábbi transzformációs mátrixal tudunk megtenni:

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

Majd a föld koordinátarendszertől tudunk a kamerarendszerbe transzformálni. Viszont a kameránk egy gimballal stabilizált kamera amely folyamatosan függőlegesen lefele tartja a kameránkat, ezért a föld kamera transzformáció során csak a Φ és Θ szögekkel szükséges transzformálnunk, hogy a Ψ elforgatás hatása megmaradjon. De mivel a kamerarendszerünk x_c tengelye a jármű jobb oldala felé néz, illetve az y_c tengelye pedig hátrafelé néz, ez a föld koordinátarendszerhez 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.

$$T_{CB} = T_{CE} \cdot T_{EB}$$

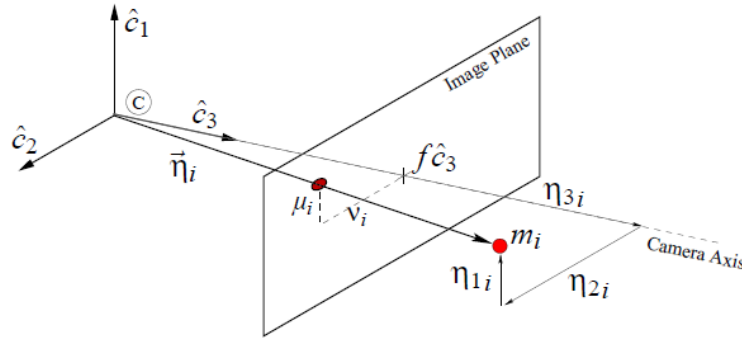
A $\underline{v}_b$ sebességvektor, a $\underline{\omega}_b$ szögsebességvektor és a ΔB pozíció vektor kamerarendszerbeli képe az alábbi módon számolhatóak:

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

2.3. 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 pontnak (későbbiekben jellemzőpont) a

koordinátái és a kamera képsíkjára vetített képe között határozza meg a matematikai összefüggést.



2.2. ábra. m pont leképezése a kamera képsíkjára. [1]

A kamera koordináta rendszerének az 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 kamera rendszerben 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]:

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

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

2.4. 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 illetve $\dot{\eta}_i^C$ a jellemző pontunk sebessége a kamera koordinátarendszerben kifejezve. A $(\dot{\mu}_i \ \dot{\nu}_i)$ koordináta páros értékei meghatározhatóak 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 -edik jellemző pontunk sebessége meghatározható úgy, hogy a Δ^C vektor és a η_i^C vektor összegének a sebességekkel és szögsebességekkel vett vektori szorzatát vesszük [3].

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

A negatív előjelű 3x3-as egység mátrix ($-I_3$) abból adódik, hogy amilyen irányba mozgunk a kamerával minden jellemzőpont azzal ellentétes irányba fog mozogni a kamera képsíkján. Tehát például amikor a drónunkal jobbra forgunk akkor a jellemzőpontjaink balra fognak mozogni.

2.5. Egyenletek

Ha a (2.4) egyenletet behelyettesítjük a (2.3) egyenletbe és kifejtjük akkor az alábbi egyenletrendszeret 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, $(x^C \ y^C \ z^C)$ a kamerarendszer testrendszerben lévő pozícióját leíró vektor koordinátái melyet átttranszformáltunk 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 $6+3N$ ismeretlent tartalmaz melynek megoldására $2N$ egyenlet szolgál. Viszont amennyiben mi a lyukkamera modellből kifejezzük a η_{1i} és η_{2i} jellemzőpont pozíció koordinátáit akkor ezzel csökkenthetjük az ismeretleneink számát [3].

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

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

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

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

Az u^C és v^C komponensek nem zérusok hisz 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 multicopter között (merevszárnyas drónnál $v^C = 0$ feltételezéssel élhetünk). Továbbá feltételezzük, hogy a legyezési szögsebességen (r^C) kívül a másik kettő szögsebesség értékünk zérus értékű ($p^C = 0$, $q^C = 0$). Továbbá mivel a kameránk közvetlenül a súlypont alá van elhelyezve függőleges irányban, azaz z_c koordináta tengely mentén ezért az egyenleteinkben szereplő x^C és y^C értékek zérusnak vehetők (lásd ΔB vektor).

Ezeket az elhanyagolásokat beépítve az egyenletrendszerünkbe:

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

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

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.

2.6. 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} . Viszont ha figyelembe vesszük, hogy $u^c = kv^c$ akkor az alábbi egyenlet írható fel:

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

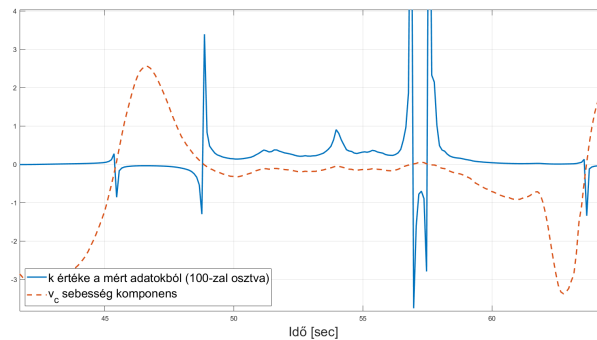
Ha kifejezzük $\frac{f v^C}{\eta_{3i}}$ a (2.10) egyenletből és behelyettesítjük (2.11) egyenletbe majd a kapott eredményt behelyettesítjük a (2.9) egyenletbe akkor az alábbi összefüggést kapjuk:

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

Ez egy bilineáris egyenletrendszer melyben már az η_{3i} ismeretlen nem szerepel. Ez az egyenletet 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 melyet 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 off-line számolva, 100 iteráció lett végig számolva mely elegendőnek bizonyult.

2.7. Szingularitás vizsgálat

A (2.11) egyenletben ($k = \frac{u^C}{v^C}$) szingularitáshoz vezethet, hogyha v^C nullával egyenlő vagy nullához közeli értéket vesz föl. Az 2.3 ábrán a mért adatokból számolt k értékét mutatja melyben a v^C -nek zérus átmenetei nagy kilengéseket okoznak k értékében.

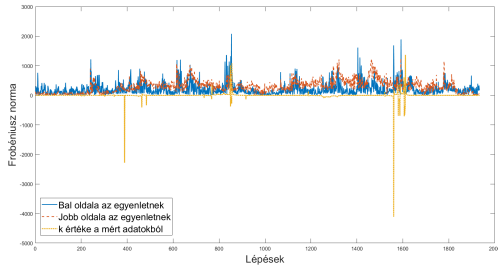


2.3. ábra. Mért szenzor adatokból számolt k érték.

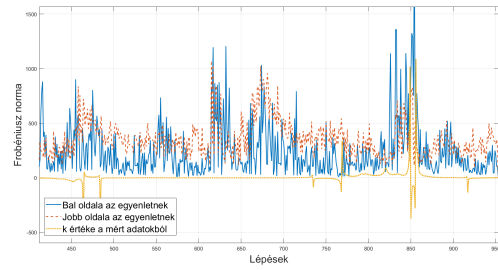
Dolgozatom során fontos szempont volt ezen szingularitások detektálása. A (2.10) egyenletben, hogyha v^C értéke zérus abban az esetben ($\mu_i r^C + \dot{\nu}_i = 0$) írható. Ha ezt behelyettesítjük a (2.12) átrendezett egyenletébe akkor láthatóvá válik, hogy az (2.13) egyenlet mindkét oldalán zérus vektorok kell, hogy álljanak.

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

Ha kiszámoljuk mindkét oldal Frobénusz-normáját akkor eredményül azt kapjuk, hogy ezek nem lesznek zérus vektorok. Ez többnyire betudható a különböző zajoknak melyek adódhatnak a különböző pixel hibákból és pontatlan gimbal szögekből. A 2.4 ábra mutatja a (2.13) egyenlet két oldalának Frobénusz normáit. Ezek a zajok miatt akkor sincsenek nulla közelében, ha a v^C nulla, vagy közel nulla.

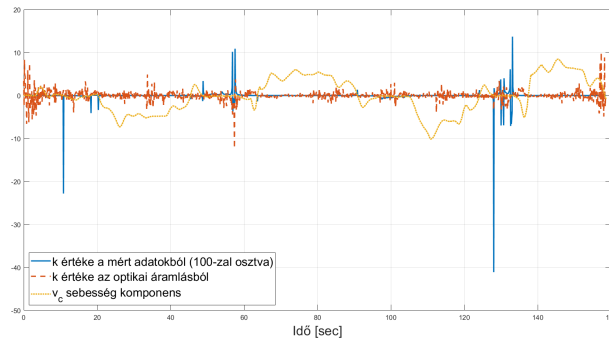


(a) A normák értéke a teljes repülés alatt.



(b) 2.4a ábra nagyított képe

2.4. ábra. A (2.13) egyenlet két oldalának Frobénusz normája.



2.5. ábra. Optikai áramlásból és mért adatokból számolt k értékek.

Köszönhetően ezeknek a zajoknak a számításainkban szingularitás nem lépett fel. Aho-
gyan azt a 2.5 ábra mutatja, az iterációs módszer által számolt k értékeket a mért ada-
tokból számolt k értékekkel közösen ábrázolva. Itt fontos megjegyezni hogy utóbbi értéke
100-al osztva van a jobb láthatóság végett. Ebből fakadóan munkám során szingularitás
detekcióra nem volt szükség.

Optikai áramlás meghatározása

3.1. Repülési adatok



11

multikopterrel. 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.

3.2. 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 [3]-ban alkalmazott *goodFeaturesToTrack* (Shi-Tomasi sarokdetektor) ezeken a felvételeken már nem volt használható mivel a 10FPS-el készült valós képeken már nem volt képes elegendő számú jellemző pontot találni (lásd 3.2 ábra). Az ORB függvény FAST sarok detekciós eljárással egy többszintű képpiramis minden szintjén megkeresi a jellemző pontokat majd egy BRIEF (Binary Robust Independent Elementary Features) leíró használatával elkészíti a jellemző pontunk leírását. A jellemző pontok megtalálására egy szintű képpiramist használtam illetve 600-ban maximalizáltam a jellemző pontok számát.



3.2. ábra. *Shi-Tomasi* sarokdetektor jellemzőpontjai.



3.3. ábra. *ORB* detektor jellemzőpontjai.

FAST algoritmus működésének a 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 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 0-kat és 1-eseket tartalmaz. Ennek nagy előnye, hogy míg a SURF és SIFT algoritmusok lebegőpontos számokat használnak addig a BRIEF leírás kettes számrendszerbeli reprezentációt használ ezzel sok memóriát és feldolgozási időt megspórol.

3.3. 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, melynek segítségével ez a feladat megvalósítható. Fő feladata, hogy minden jellemzőponthoz meghatározzon egy olyan vektort mely arányos a pont mozgásával az egymást követő képeken.

OpenCV könyvtárban implementált *OpticalFlowPyrLK* függvényt használtam az alábbi paraméterekkel. A vizsgáló ablak mérete 9x9-es, maximum 7 szintű képpiramist használjon, és az iterációk számát 10-ben maximáltam.



3.4. ábra. Kiszűrt jellemzőpontok.

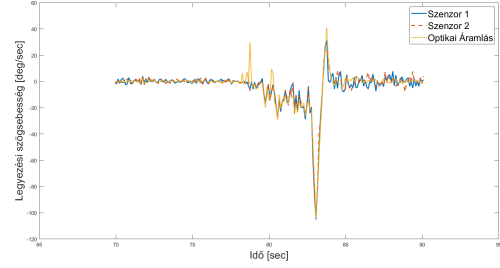
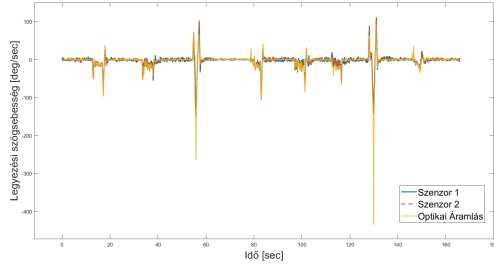
Ennek a módszernek a segítségével megtudjuk határozni, hogy az *ORB* által a (k) -adik képen detektált pontok a $(k + 1)$ -ik képen hol találhatóak. Viszont vannak esetek amikor ezek a párosítások hibásan történnek meg mert változik a pixel intenzitás vagy esetleg túl nagyok a mozgások. Ezen probléma 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, hogyha ezt az értéket átlépi abban az esetben ezt a pontot már ne használjuk fel a későbbi számításainkban. A 3.4 ábrán kék színnel láthatóak a kiszűrt pontok.

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

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

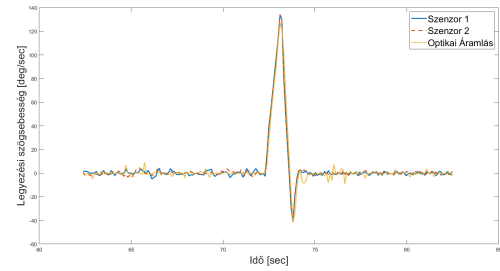
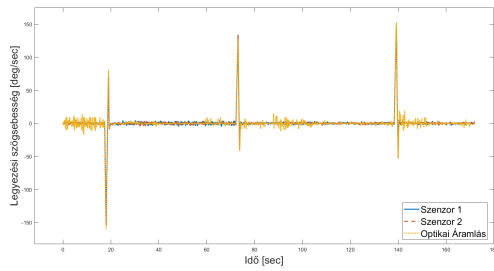
ahol Δt a két egymást követő képkocka között eltelt időt jelöli.

Ezzel a (2.12) egyenletben szereplő $\dot{\mu}_i$, $\dot{\nu}_i$, μ_i , ν_i értékek meghatározása kerültek. Amennyiben ezeket behelyettesítjük a (2.12) egyenletbe és megoldjuk a 2.6 részben bemutatott módon akkor az alábbi eredményeket kapjuk:



(a) 1-es repülés mért és becsült szögsebességei (b) 3.6a ábra 70s és 90s közötti nagyított képe

3.5. ábra. 1-es repülés szögsebességeinek összehasonlítása



(a) 2-es repülés mért és becsült szögsebességei (b) 3.6a ábra 62s és 84s közötti nagyított képe

3.6. ábra. 2-es repülés szögsebességeinek összehasonlítása

Ez a módszer szép becsléseket adott a legyezési szögsebességekre de a 3.6a ábrán látható, hogy a két nagy sebességű forgásnál (58s és 110s) jelentősen túlbecsüli a szögsebességeket, ellenben ez kiküszöbölhető ha ismeretes mekkora maximális szögsebesség értékere van korlátozva a drónunk.

4. fejezet

Szenzor hibadetekció

Az utolsó fejezetben ismertetem a vizsgált szenzorhibákat, a hibadetekciós eljárást és ezek alkalmazásának eredményeit.

4.1. Szenzor meghibásodások fajtái:

Guillermo Heredia és társai [6] munkájukban részletesen bemutatják egy autonóm modellhelikopter jellemző szenzor meghibásodásait, és [3] munkája alapján 4 fajta szenzor hibát vizsgáltam:

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 szenzor meghibásodás: A szenzor által mért értékekre egy konstans érték A szuperponálódik.
4. Multiplikatív típusú szenzor meghibásodás: A szenzor kimenetén a mért értékek $100+P\%$ -a fog megjelenni.

4.2. Hibadetekciós eljárás

A DJI M600 drón három darab egymástól független szögsebesség szenzorral van felszerelve, melyből én csak két darab szenzor (Szenzor1 és Szenzor2) adatait használom fel a munkám során. A fentebb felsorolt hibákat mesterségesen előállítottam a Szenzor2 adatain ezzel reprezentálva a hibás szenzort. Fontos szempont, hogy a hibadetekciós eljárás

egy hibátlanul működő szenzort ne érzékeljen hibásnak ezért a szenzor1 adatait meghagytam eredeti állapotukban. Továbbá a hangolás és a tesztelés során is két különböző repülés (repülés1 és repülés2) adatait használtam fel melyek szögsebesség értékei a 3.5 és 3.6 ábrákon láthatóak. Ahmet Arda Ozdemir és társai [7] munkájuk során egy számlálót és határértékeket használó hibadetekciós eljárást használnak szélturbinák részegységei meghibásodásának előrejelzésére melyben a módszer egy részletes leírása is található.

Lényegében a módszer alapja, hogy az optikai áramlásból becsült legyezési szögsebesség r_{OF} és a szenzorok által mért szögsebesség különbségét képezzük az alábbi reziduált kapva.

$$e_{1/2} = r_{sensor1/2} - r_{OF} \quad (4.1)$$

Ha ez az eltérés nagyobb mint egy bizonyos határérték ($Tol1$) akkor egy számláló értékét növeljük egy előre meghatározott értékkel (U), viszont ha kisebb, mint a határérték abban az esetben csökkentjük az értékét egy előre meghatározott értékkel (D). Ezeknek az értékeknek nem kell szükségszerűen azonosnak lenniük. Majd erre a számlálóra is bevezetünk egy határértéket $Tol2$ és abban az esetben ha a számláló átlépi ezt az értéket a szenzort hibásnak tekintjük.

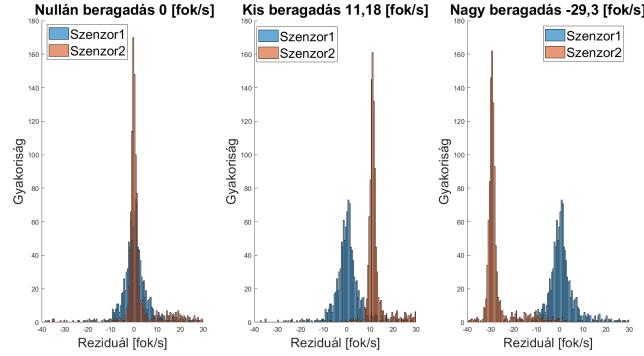
4.3. Határértékek hangolása

A $Tol1$ határérték adja meg a hibadetektálás érzékenységét, hogy mekkora eltérést tudunk maximálisan detektálni, a $Tol2$ 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, hogyha a zajok $Tol1$ küszöbértéknél nagyobbak. Azaz hibát jelzünk amikor valójában nincs is szenzorhiba. Másik kedvezőtlen eset, amikor a szenzorhibánk mértéke kisebb mint $Tol1$ abban az esetben tévesztést (missed detecion (MD)) fog nekünk okozni. Azaz nem jelzünk hibát holott a szenzorunk hibás. Tehát lényegében nekünk a $Tol1$ 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.

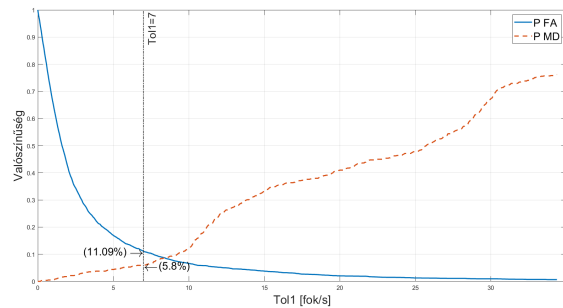
Ennek alapján definiálunk egy valószínűséget ($P(data > Tol1|no\ error)$) mely azt fejezi ki, hogy a reziduálunk hány alkalommal lépi át a $Tol1$ küszöbértéket akkor amikor hibamentes szenzorunk (FA). Illetve definiálunk egy másik valószínűséget $P(data < Tol1|error)$ amikor azt vizsgáljuk, hogy a reziduálunk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

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



4.1. ábra. Repülés1 reziduál hisztogramjai különböző hibák esetén. (Szenzor1 nem tartalmaz hibát, szenzor2 hibával terhelt)

Teljes szenzormeghibásodást (nulla értéken történő megakadás) a hangolás során nem vettem figyelembe mivel a reziduálja nem meglülönbőztethető egy hibamentes szenzor reziduáljától. Ez jól látható a 4.1 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 risztásokat előidézni. Ennek a kiküszöbölésére csak akkor számoljuk a $P(data < Tol1|error)$ és $(P(data > Tol1|no error))$ valószínűségeket amikor a mért jelünk $Tol3 = 4^\circ/s$ feletti értéket vesz föl. Ezt az értéket kísérletileg határoztam meg nevezetesen, hogy a 4.2 ábrán látható görbéket különböző $Tol3$ értékekre kiszámoltam majd figyelembevéve a reziduálon lévő zajok mértékét (lásd 4.1 ábra) $Tol3 = 4^\circ/s$ értéket választottam.



4.2. ábra. A valószínűségek különböző küszöbértékek mellett.

A $(P(data > Tol1|no\ error))$ valószínűséget úgy határozzuk meg, hogy összeszámoljuk mindkét repülésen hányszor haladja meg a Tol1 értékét a hibamentes reziduál majd elosztjuk az összes időlépés számával. $P(data < Tol1|error)$ valószínűségeket úgy tudjuk meghatározni, hogy összeszámoljuk hányszor nem lépik át, a fentebbi hibákat magukba foglaló reziduálok, a Tol1 értékét majd elosztjuk az összes időlépéssel melyekben a hibák fennálltak. Ezt megismételve különböző Tol1 értékekre $0^\circ/s$ és $34^\circ/s$ között a 4.2 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 de $Tol1 = 7^\circ/s$ határértéket választottam alacsonyabb tévesztés és magasabb téves riasztás aránnyal mert egy jól megválasztott fel/le számlálóval a téves riasztások aránya csökkenthető.

A Tol2 határérték fogja eldönteni számunkra, hogy az adott szenzorunk meghibásodott vagy sem. A döntéseinket 4 különböző csoportba osztályozhatjuk:

1. True Positive (TP): Amikor a szenzorunk hibás és hibásnak ítéljük.
2. False Positive (FP): Amikor a szenzorunk nem hibás és hibásnak értékeljük.
3. True Negative (TN): Amikor a szenzorunk nem hibás és nem hibásnak is osztályozzuk.
4. False Negative (FN): Amikor a szenzorunk hibás és nem hibásnak minősítjük.

A Tol2 értékét a Receiver Operating Characteristic (ROC) görbe segítségével határozom meg [8]. Az ROC görbe a hamis pozitív arányának (FPR) függvényében ábrázolja a valós pozitív arányát (TPR) különböző küszbértékek mellett.

A False Positive észlelések arányát úgy határoztam meg, hogy a hibamentes jel fel/le számlálóján megszámláltam mindkét repülés esetén hány alkalommal haladta meg Tol2 értékét majd elosztottam a False Positive és a True Negative észlelések számának összegével.

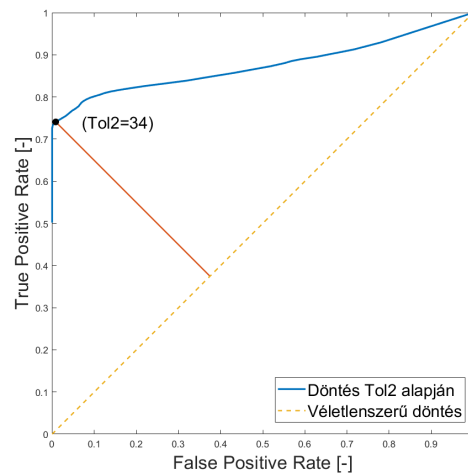
$$FPR = \frac{FP}{FP + TN} \quad (4.2)$$

A True Positive döntések arányának a meghatározásához az eredeti jelünkön 25 különböző időpillanatban konstans értéken történő beragadás típusú szenzor hibát idéztem elő, mindkét repülésen külön-külön. Mind az 50 különböző esetben megszámloljuk, 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 és 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 abban az esetben a 4.3 ábrán látható ROC görbét kapjuk eredményül.



4.3. ábra. ROC görbe az optimális $Tol2$ érték meghatározásához.

Az ideális esetet az 4.3 ábra bal felső sarkában lévő (0,1) pont jelölné mely azt jelentené, hogy a hibás szenzort mindig sikerült felismerni és egyszer sem ítéltük hibásnak a hibátlanul működő szenzorunkat. 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 (mely a véletlenszerű döntést reprezentálja) a legnagyobb merőleges távolságra lévő pontot megkeressük. Ez esetünkben a $Tol2 = 34$ értéket határozta meg, ahol False Positive döntések aránya ($FPR = 0.0085$) és True Positive döntések aránya ($TPR = 0.7408$).

4.4. A hibadetektálás eredményei

Az előzőekben meghatározott paraméterekkel ($Tol1 = 7^\circ/s$ $Tol2 = 34$ $Tol3 = 4^\circ/s$ $U = 1$ $D = 1$) felahangolt hibadetekciós eljárást két különböző repülésen teszteltem.

Meghibásodás időpillanatának két különböző esetet is választottam. Amikor a szögsebesség közel konstans értéket vesz föl (konstans), illetve a forgások kezdetén mielőtt nagy változás játszódik le a szögsebesség értékében (változó).

4.1. táblázat. Hibadetektálási időtartamok.

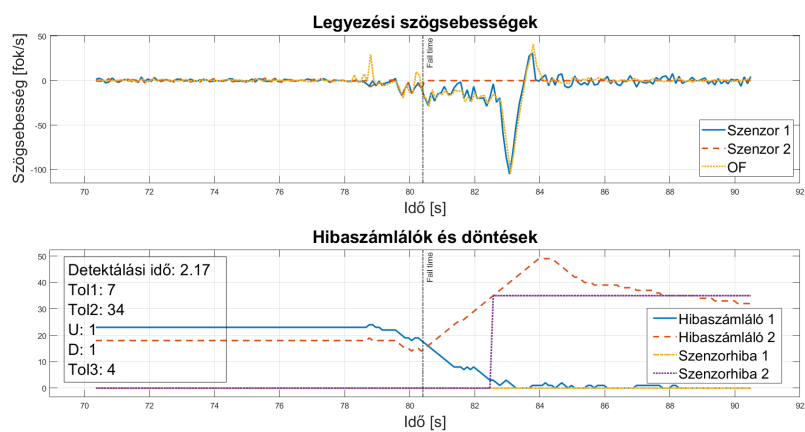
Hiba Típusa	Repülés1 Változó	Repülés1 Konstans	Repülés2 Változó	Repülés2 Konstans
Szenzor teljes meghibásodása	2.17s	5.367s	67.4377s	69.5392s
Konstans értéken beragadás	13.795s	5.367s	67.6409s	69.5392s
Additív típusú hiba	55.677s	59.677s	X	X
Multiplikatív típusú hiba	51.267s	55.267s	67.5392s	69.5392s

A 4.1 táblázat foglalja össze a hiba beálltának időpontjától a hibadetektálásáig terjedő időtartamokat sikeres hibadetekció esetén és X jelöli amikor nem sikerült a hibás szenzort detektálni.

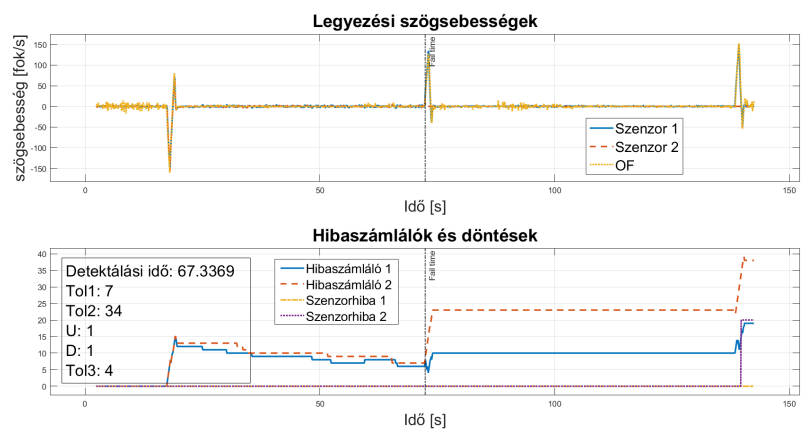
Additív típusú szenzor meghibásodás esetén ez a módszer ilyen hangolással, $A = 2.1^\circ/s$ és ennél nagyobb hozzáadott konstans értékeket képes detektálni az 1-es repülés esetében.

Multiplikatív típusú szenzor meghibásodás esetén $P = 8\%$ és nagyobb százalékok esetén képes detektálni mindkét repülés esetén. A 4.1 táblázatban szereplő értékek a legkisebb A és P értékekhez tartoznak melyeket a módszer képes volt detektálni. Természetesen nagyobb eltérések esetén ezek a detektálási időtartamok fokozatosan rendre csökkenek.

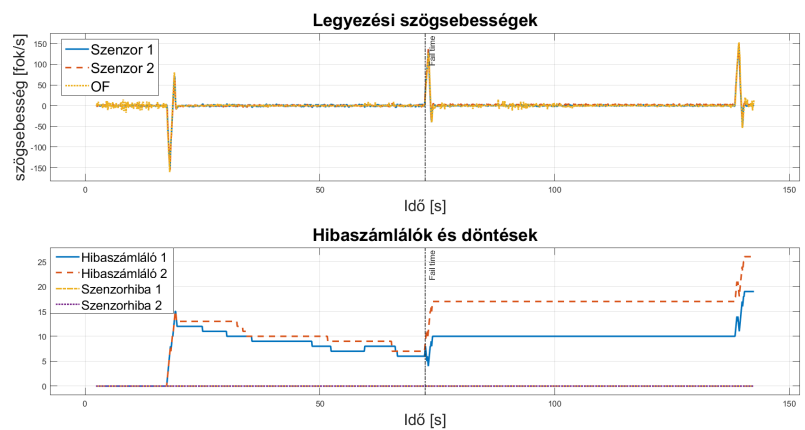
Additív típusú szenzor meghibásodás esetén a 2-es repülés hibás szenzorát nem tudtam detektálni mert a hibaszámláló nem érte el $Tol2$ értékét a rendelkezésre álló időtartam alatt (lásd 4.6 ábra) nevezetesen a rövid időtartamú forgások és a 10Hz-es mintavételezés okán. Hasonló eset áll fenn a többi típusú meghibásodás esetén a 2-es repülésnél. 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 a 4.5 ábrán ahol is a két forgás között közel 60 másodperc telik el. A 4.4 ábrán pedig a leggyorsabb detektálás látható.



4.4. ábra. Repülés1 teljes szenzor meghibásodással.



4.5. ábra. Repülés2 teljes szenzor meghibásodással.



4.6. ábra. Repülés2 additív típusú szenzor meghibásodással.

5. fejezet

Összefoglalás

TDK dolgozatomban egy optikai áramlás alapú szenzorhiba detekciót mutattam be valós repülési adatok felhasználásával. Az optikai áramlás és a szögsebességek kapcsolatát leíró egyenletek bevezetése után az egyenletrendszert aktualizáltam egy gimballal stabilizált függőlegesen lefelé néző kamera esetére. Az egyenletrendszer megoldásának ismertetését követően a megoldáshoz szükséges képfeldolgozást prezentáltam. A szenzorhibák detektálására egy hibaszámlálót és küszöbértékeket használó módszert alkalmaztam, melynek hangolása a téves riasztások és a tévesztések rátájának figyelembevételével történt illetve ROC görbe használatával. A valós repülési adatokban mesterségesen hibákat generáltam és ezeket a szenzor adatokat az optikai áramlásból becsült szögsebesség értékkel összehasonlítva detektáltam. A kidolgozott módszer a legtöbb vizsgált szenzorhibát képes volt észlelni additív hiba esetén $2.1^\circ/s$ felett, multiplikatív hiba esetén pedig 8% feletti értékek esetén de elég hosszú idő elteltével. Ez főként a forgások nagyon rövid időtartamának tudható be. A módszer hátránya, hogy nagyon kis értéken történő beragadásokat, $2.1^\circ/s$ alatt nem képes detektálni. Ebben a dolgozatban csak a legyezési szögsebességet becsültem, a későbbiekben a többi szögsebesség komponens becslése és hibadetektálása is cél, merevszárnyas pilóta nélküli légi járműveken, 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 kamera képekből a szögelfordulás meghatározásával válhat lehetségessé.

Irodalom

- [1] Joseph Kehoe és tsai. „State Estimation Using Optical Flow from Parallax-Weighted Feature Tracking”. *AIAA Guidance, Navigation, and Control Conference and Exhibit*. DOI: 10.2514/6.2006-6721. eprint: <https://arc.aiaa.org/doi/pdf/10.2514/6.2006-6721>. URL: <https://arc.aiaa.org/doi/abs/10.2514/6.2006-6721>.
- [2] Szabolcs Kun és Peter Bauer. „Improvements in Optical Flow-based Aircraft Partial State Estimation”. *2021 17th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*. 2021. szept., 1–4. old. DOI: 10.1109/CNNA49188.2021.9610822.
- [3] Peter Bauer és Szabolcs Kun. „Optical flow-based angular rate sensor fault detection on UAVs*”. *IFAC-PapersOnLine* 55.14 (2022). 11th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2022, 46–51. old. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2022.07.581>. URL: <https://www.sciencedirect.com/science/article/pii/S2405896322009922>.
- [4] Czúni László Kató Zoltán. *Számítógépes látás [elektronikus dok.] : egyetemi tananyag / írta Kató Zoltán, Czúni László*. Typotex, 2011. URL: <http://nbn.urn.hu/N2L?urn:nbn:hu-108199>.
- [5] Péter Bauer és tsai. „Encounter Risk Evaluation with a Forerunner UAV”. *Remote Sensing* 15.6 (2023). ISSN: 2072-4292. DOI: 10.3390/rs15061512. URL: <https://www.mdpi.com/2072-4292/15/6/1512>.
- [6] G. Heredia és tsai. „Detection of Sensor Faults in Autonomous Helicopters”. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. 2005. ápr., 2229–2234. old. DOI: 10.1109/ROBOT.2005.1570444.
- [7] Ahmet Arda Ozdemir, Peter Seiler és Gary J. Balas. „Wind Turbine Fault Detection Using Counter-Based Residual Thresholding”. *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, 8289–8294. old. ISSN: 1474-6670. DOI: <https://doi.org/10.3182/20110828-6-IT-1002.01758>. URL: <https://www.sciencedirect.com/science/article/pii/S1474667016449426>.

- [8] Tamas Grof és Peter Bauer. „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* 53.2 (2020). 21st IFAC World Congress, 14724–14729. old. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2020.12.1841>. URL: <https://www.sciencedirect.com/science/article/pii/S2405896320324551>.

Ábrák jegyzéke

2.1. Koordináta-rendszerek.	4
2.2. m pont leképezése a kamera képsíkjára. [1]	6
2.3. Mért szenzor adatokból számolt k érték.	9
2.4. A (2.13) egyenlet két oldalának Frobeniusz normája.	10
2.5. Optikai áramlásból és mért adatokból számolt k értékek.	10
3.1. A felhasznált repülések útvonalai ZalaZONE Smart City felett.	11
3.2. <i>Shi-Tomasi</i> sarokdetektor jellemzőpontjai.	12
3.3. <i>ORB</i> detektor jellemzőpontjai.	13
3.4. Kiszűrt jellemzőpontok.	14
3.5. 1-es repülés szögsebességeinek összehasonlítása	15
3.6. 2-es repülés szögsebességeinek összehasonlítása	15
4.1. Repülés1 reziduál hisztogramjai különböző hibák esetén. (Szenzor1 nem tartalmaz hibát, szenzor2 hibával terhelt)	18
4.2. A valószínűségek különböző küszöbértékek mellett.	18
4.3. ROC görbe az optimális $Tol2$ érték meghatározásához.	20
4.4. Repülés1 teljes szenzor meghibásodással.	22
4.5. Repülés2 teljes szenzor meghibásodással.	22
4.6. Repülés2 additív típusú szenzor meghibásodással.	22

Táblázatok jegyzéke

4.1. Hibadetektálási időtartamok.	21
---	----