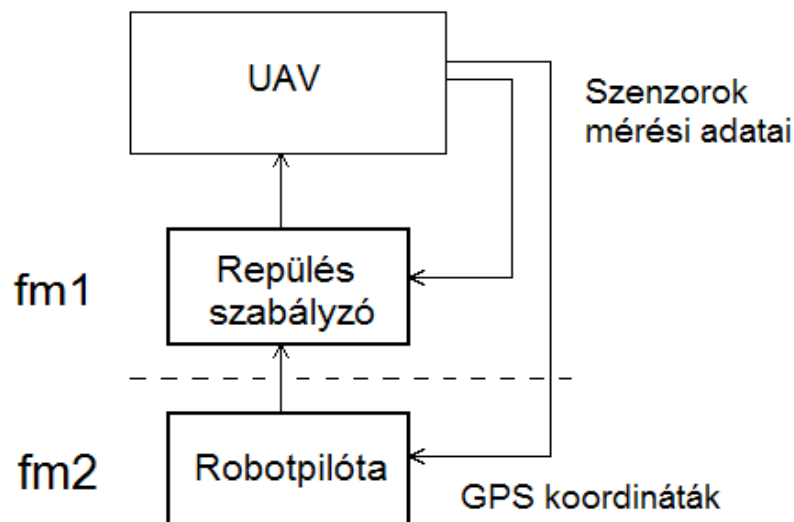


## MIKRO MÉRETŰ PILÓTA NÉLKÜLI REPÜLŐK REPÜLÉSBIZTONSÁGI KÉRDÉSEI – FEDÉLZETI ADATFÚZIÓ

### BEVEZETÉS

A pilóta nélküli repülő (UAV-k) alkalmazásának és elterjedésének feltétele a hibatűrő működés. Cikkemben, mint azt korábbi írásaimban is tettem, most is a kis méretű (mikro) UAV-k automatizálását biztosító **robot, repülésbiztonsági kérdéseit** tárgyalom. Korábban áttekintettem a központi vezérlő struktúrákat és azok redundáns, hibatűrő kialakításainak lehetőségét [1.], valamint foglalkoztam az UAV-k elektromos tápellátás biztonságának kérdésével is [2.]. Nyilvánvaló, hogy az előbb említett funkciókat ellátó berendezések működésképtelenné válása az UAV zuhanásához vezet, ezért mind a hardver kialakításban, mind pedig az egyes építő elemekben futó szoftverrel kapcsolatban precíz elvárásokat kell megfogalmaznunk.

Hibatlan tápellátás esetén, valamint ha az UAV fedélzetén hibátlan, működőképes központi vezérlő üzemel még nem jelent garanciát a biztonságos repülésre, ezek csak a szükséges, de nem elégséges feltétel kategóriába tartoznak. Természetesen a sárkánytest, valamint a beavatkozó szervek, hajtóművek kielégítő működése is elengedhetetlen tényező. A repülési paramétereket, a sárkánytest helyzetét, pozícióját szenzorok sokasága méri. A szenzoroktól érkező adatok a központi vezérlőbe kerülnek, mely adatok az alapjellel (kívánt pillanatnyi értékkel) összevetésre kerülnek és a rendszer jellemzőinek figyelembevételével kerül kiszámításra a beavatkozó jel. Az adott pillanatban elvárt repülési adatok (jelen esetben az UAV robotpilóta utasításai), mint az adott pillanatban aktuális alapjel, a szenzoroktól érkező jelek, valamint az ezeket feldolgozó digitális jelfeldolgozó áramkör egy többszörösen visszacsatolt rendszert alkotnak (1. ábra).



1. ábra Mikro UAV, mint többszörösen visszacsatolt rendszer

A szenzoroktól érkező mérési eredmények szolgáltatják a visszacsatolást. Abban az esetben, ha hibás a visszacsatolt jel, akkor a központi vezérlő jelfeldolgozó áramköre hibás rendelkező

jelet fog előállítani, vagyis a beavatkozó szervekhez hibás adatok fognak érkezni: így, egy látszólag nem olyan „lényeges” mérési eredmény hibája is, az UAV zuhanását eredményezheti.

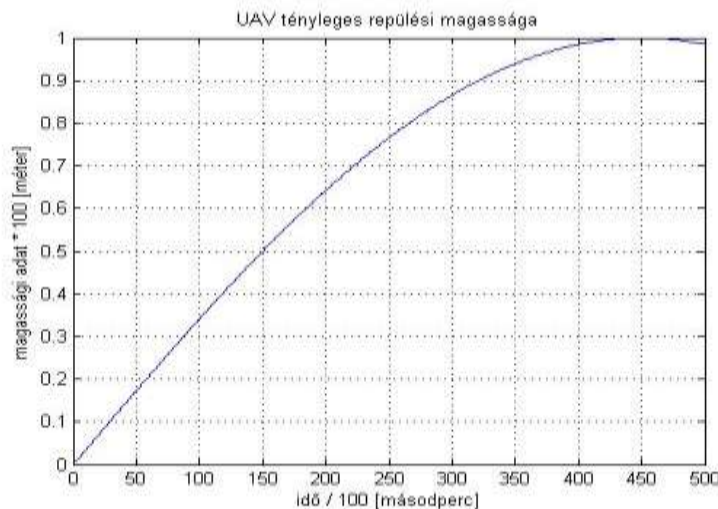
## SZENZORADATOK ELLENŐRZÉSE, ADATFÚZIÓS MODELLEK

### Zajszűrés, hihetőség-vizsgálat

Az egyes szenzorok mérési eredményei véletlen zajjal terheltek. A véletlen zajok szűrésére hatékony módszer az integrálás. A mérési adatok integrálhatóságának feltétele az, hogy a digitális jelfeldolgozó áramkör –jelen esetben a visszacsatolt szabályzó kör- mintavételi frekvenciájának a többszöröse legyen az a frekvencia, amellyel a mérési adatok a szenzortól érkeznek. Aktív szenzorok esetén előfordul, hogy a szenzoradat előszűrését a szenzorban megvalósítja a gyártó.

Egy szenzor pillanatnyi mérési eredménye ellenőrizhető az előző mérési eredmények alapján végzett hihetőség vizsgálatával, vagy az előzőleg mért adatokból a következő mérési eredmény jóslásával, predikciójával.

Egy elképzelt repülési szimuláció szerint, melyet MATLAB környezetben készítettem az UAV emelkedése a következőképpen néz ki:



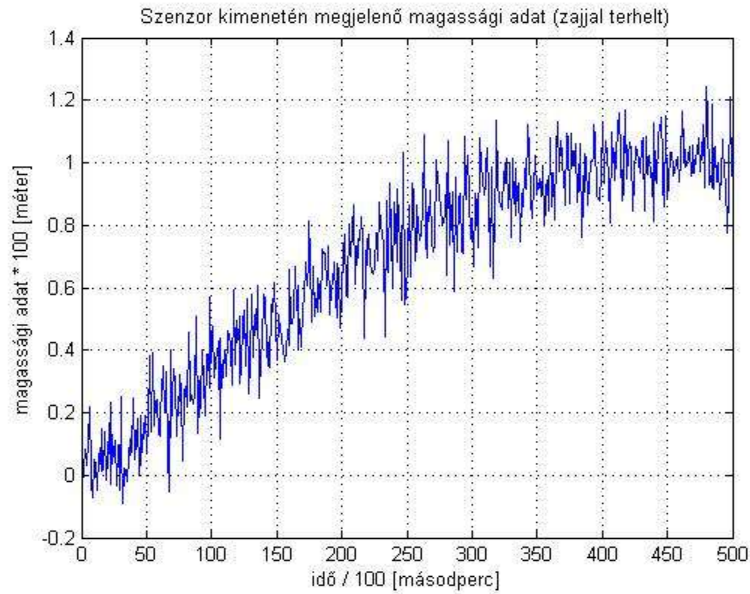
2. ábra Mikro UAV repülési magassága {MATLAB szimuláció}

A szenzor, a 2. ábrán látható „tényleges” magasságot, a szenzorra jellemző mérési hibával és zajjal terhelt tudja mérni. A zajjal terhelt magassági adat áll rendelkezésre a továbbiakban az UAV fedélzetén.

A zajt a MATLAB szimulációban a következőképp lehet előállítani:

$$\text{noise} = \text{nvar} * \text{randn}(1, N); \quad \{1.\}$$

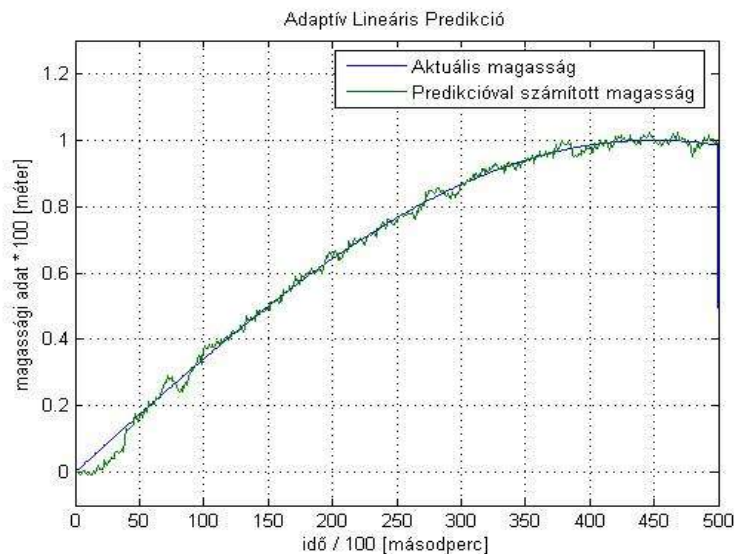
ahol a zajminták egy N elemszámú vektorba kapnak helyet. A *nvar* konstans, jelen példában 0,1 értéket kapott (noise variance). A **randn** függvény a MATLAB beépített függvénye, mely normál eloszlású véletlen számokat generál. A mesterségesen előállított zajt additív zajként értelmezzük és azt feltételezzük, hogy ezzel a zajjal terhelt jelennek meg a magasság adatok a magasságmérő szenzor kimenetén:



3. ábra Mikro UAV repülési magassága zajjal terhelt  
{MATLAB szimuláció}

A zajjal terhelt mérési adatok a szimuláció szerint a magasságmérő szenzor kimenetén jelenik meg. Ezt az adatsort valós időben kell feldolgoznunk és lehetőség szerint szűrni kell (el kell nyomni) a zajt.

Szűrés után látható, hogy a mérési eredményről a zaj részben eltávolítható:



4. ábra Mikro UAV repülési tényleges magassága és  
zajsztűrt szenzor adat {MATLAB szimuláció}

A predikció során tehát az adott jel jövőbeli pillanatértékét jósoljuk meg, az adott jelre vonatkozó korábban szerzett tapasztalat alapján és a jel korábbi értékeinek ismeretében, vagyis a jövőbeli minta értéket a múltbeli minták lineáris kombinációjával állítjuk elő. Gyakorlatban egy szenzortól érkező jel hihetőség vizsgálatát úgy tudjuk elvégezni, hogy a szenzortól előzőleg érkezett adatok alapján (legyen  $p$  számú adat) jósoljuk a következő mérési eredményt, majd a mérési eredmény megérkezésekor összevetjük azt a megjósolttal. Abban az

esetben, ha a jósolt és a tényleges mérési eredmény egymástól való eltérése egy hibahatáron belüli, akkor az eredményt jó valószínűséggel mondhatjuk hibátlannak. Mérési eredmény jóslása történhet úgynevezett lineáris predikcióval [3.], ahol az  $s(n)$  mérési eredményt jósoljuk az előzőekben már megismert mérési adatokból:

$$\tilde{s}(n) = \sum_{i=1}^p \alpha_i \cdot s(n-i) \quad \{2.\}$$

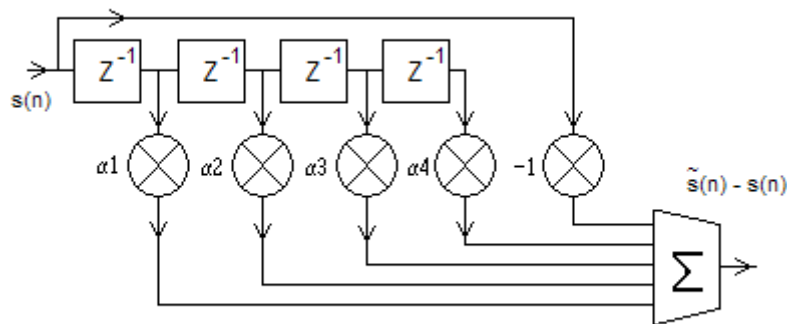
A fenti összefüggés szerint a jóslást,  $p$  számú, már a megérkezett mérési eredményből végezzük, mely alapján a lineáris predikció fokszámát  $p$ -nek mondjuk.

A lineáris predikciós összefüggésben  $\alpha_i$  az adott jelre, jelenségre vonatkozó korábban szerzett tapasztalat (előismeret)  $i$ -edik eleme.

A jelenségre vonatkozó előismeretet a predikció fokszámával egyező  $p$  számú elemből álló oszlopvektorba foglalhatjuk össze:

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} \quad \{3.\}$$

Az {2.} összefüggésben megadott predikciós modell FIR (Finite Impulse Response) digitális szűrőstruktúrával realizálható, melynek a kimenetén megjelenő  $n$ -edik becsült minta értékéből le kell vonnunk a tényleges  $n$ -edik mérési eredmény mintát. Természetesen, ha fordítva járunk el (tényleges mérési eredmény – becsült mérési eredmény különbség képzése), az is ugyan olyan jól használható eredményt ad, hiszen jelen esetben csakis az eltérés abszolút értéke érdekes a továbbiakban. A következő ábrán egy  $p=4$  fokszámú lineáris predikciós modellel működő mérési eredmény hihetőségét vizsgáló realizációt láthatunk:



5. ábra Negyedfokú predikciós hihetőség-vizsgáló

A fenti ábrán a  $Z^{-1}$  komponensek memóriáláncot alkotnak, melyekben az eltárolt adat a mintavételi frekvencia ütemében lép át a következő tagba. A DSP realizáció a  $Z^{-1}$  esetben tehát egy számábrázolási pontosságnak megfelelő méretű, „együtemnyi” késleltetéssel működő tároló elem. Az  $\alpha$  szorzó konstansok jelentése pedig nem más, mint a FIR szűrő szorzó konstansai, jelen értelmezés szerint a lineáris predikció együtthatói. Az  $\alpha$  szorzó konstansokat úgy is felfoghatjuk (FIR szűrő szemlélet szerinti megközelítésben), mint a  $H(p)$  átviteli függvényű szűrő dirac impulzus (mintavételes esetben Kronecker delta) bemenő jelre adott válaszának mintáit, vagyis az úgynevezett súlyfüggvény mintáit.

A fentiekből tehát az következik, hogy a jelenleg alkalmazott lineáris predikciós becslő jelfeldolgozó áramkör  $\alpha_i$  predikciós együtthatókból képzett vektor {3.} Fourier

transzformáltja adja a predikciós kör szűrő részének frekvencia tartománybeli leírását (csillapítását és fázistolását a frekvencia függvényében).

A predikciós áramkör tervezésénél a feladat tehát az, hogy az alkalmazott szenzor, valamint a mérendő fizikai jellemző pontos ismeretében meghatározzuk az  $\alpha$  predikciós együtthatókat. Természetesen a mérendő fizikai jellemzőt pontosan nem ismerhetjük, így a tervezés egy tipikus realizáción valósulhat meg [5.].

A predikciós áramkör a működése során annál pontosabb előrejelzést ad, minél jobban sikerült a tervezés során a predikciós együtthatók meghatározása, valamint a predikciós szűrő fokszáma is kihatással van a jósolt eredményre.

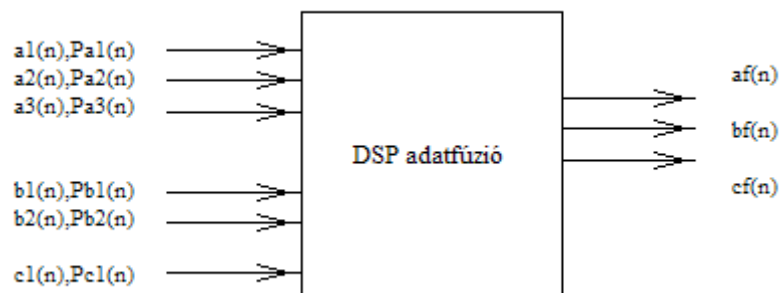
A fentiekből látszik, hogy a predikcióval előállított, becsült  $s(n)$  eredmény valamilyen hibával terhelt lesz, és ez a hiba valamilyen valószínűséggel marad egy bizonyos hibahatáron belül. A tényleges mérésből származó  $s(n)$  adatot tehát egy kiegészítő számmal kell megjelölni, mely megjelölést végző szám a predikciós áramkörből származó, az adott szenzor méréshez tartozó  $\tilde{s}(n) - s(n)$  -ből, mint hibajelből képezhető. Ezt az adatot a továbbiakban az adatfúziós áramkör azon része használja fel, mely az előfeldolgozott adatokat összesíti, fuzionálja.

### Adatfúzió

Az adatfúzió redundáns mérési adatok alapján végezhető el, vagyis bizonyos repülésbiztonságot jelentő fizikai jellemzők mérését végző szenzorokból az UAV fedélzetére többet kell telepíteni.

Az adatfúzió alapját jelenti továbbá az egyes szenzorok által biztosított, első ránézésre fölöslegesen megjelenő, redundáns többletadatok felhasználása. Az aktuális „X-Y” koordináta adatok biztosítása mellett a fedélzeti helyzet meghatározó eszköz (például GPS vevő), magassági adatokat, sőt haladási sebesség adatokat is szolgáltat. Abban az esetben, ha ezen adatokat összevetjük az elsődleges sebesség-, és magasságmérő szenzor adataival, akkor ez jó ellenőrzési támpont. A különböző szenzoroktól érkező, azonos fizikai jellemzők adataira vonatkozó nagy eltérés esetén valószínűség vizsgálattal dönthetünk arról, hogy mely adatot fogadjuk el valószínűnek, érvényesnek.

Az előfeldolgozott, szűrt szenzoradatok fuzionálása hihetőség vizsgálaton alapulhat. Azt a szenzor mérési eredményt, melynek hihetőségét minősítő számértéke jobb, mint egy másik, vele azonos paramétert szolgáltatóé, azt nagyobb súllyal kell figyelembe venni. A mérési adatok esetén ha az adatfúziót végző áramkör nagy eltérést tapasztal, akkor a rosszabb hihetőségű adatot el kell dobni és a továbbiakban a jobb jellemzőkkel bírót kell felhasználni.



6. ábra UAV fedélzeti adatfúziós modell

A fenti adatfúziós modellben az azonos fizikai jellemzőket azonos betűvel jelöltem, míg a betűknél előforduló indexszám a mérési adat sorszámot (például szenzor sorszámot) jelöli. Az „ $n$ ” pedig minden esetben mintasorszámot jelöl. Minden szenzortól a megérkezett  $n$ -edik adat a modell szerint előfeldolgozáson már átesett, így a mérési adathoz egy digitális szám formájában egy hihetőségi mérőszám tartozik. Például a  $PaI(n)$  jelentése az, hogy az első „ $a$ ” mennyiséget mérő szenzortól érkező  $n$ -edik adat  $P$  valószínűséggel tekinthető egy előre definiált hibahatáron belülnek.

A kimeneti adatok már az adatfúzió eredményeképpen előálló fúziós adatok. A 6. ábrán láthatunk egy példát, mely szerint a „ $c$ ” fizikai mennyiség csak egy szenzortól származtatott, a kimeneten viszont ennek a fizikai mennyiségnek is előállhat egy nagyobb megbízhatóságúnak tekinthető komponense. Ez úgy lehetséges, hogy az egyes fizikai mennyiségekből származtathatók (kalkulálhatóak) más fizikai mennyiségek is, jelen esetben a „ $c$ ” mennyiség. E szerint lehetőség nyílik a  $cI(n)$  adatfúzióra a többi fizikai jellemzőből származtatott „ $c$ ”-vel.

A 6. ábrán felvázolt fúziós modell további realizálhatósági kérdéseket is felvet:

- Miként kezelhető az az eset, ha egy szenzoron nem végzünk hihetőségi vizsgálatot, vagy azt a szenzor belsejébe integrált jelprocesszor már megtette, így a  $P(n)$  adat nem áll rendelkezésünkre?
- A DSP adatfúziós modell, -mint szinkron szekvenciális digitális hálózat- bemenetére nem szinkronozottan érkeznek a mérési adatok, sőt az egyes szenzorok mintavételi frekvenciája is eltérő.

A fent nevezett problémák vonatkozásában a fúziós modell változatlanul használható az alábbiak szerint:

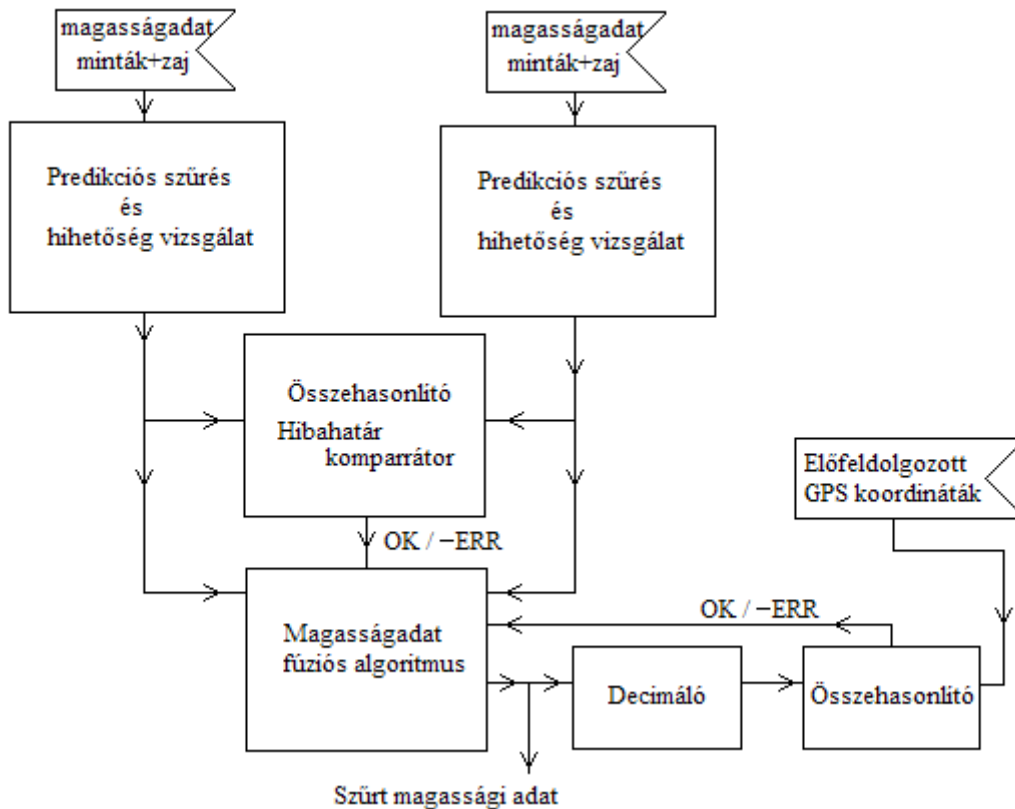
Ha nem áll rendelkezésünkre az adott  $n$ -edik eredményhez tartozó  $P(n)$ , akkor a szenzor megbízhatósága és a gyártó által garantált pontossági adata alapján a fúziós áramkör tervezésekor nekünk kell a  $P(n)$ -nek egy konstans értéket adni, ami ekkor már természetesen  $n$ -től független.

Az adatfúziós áramkör bemenetére érkező adatok szinkronizálása a gyakorlatban szintén egyszerűen megoldható feladat. Az adatok fogadására egy-egy memóriaterületen kialakított üzenet buffer-re van szükségünk, így az egymáshoz képest aszinkron módon megérkező mérési eredmények a felhasználáskor (buffer kiolvasáskor) válnak szinkronozottá. Ez a módszer az egymástól jelentősebben eltérő mintavételi frekvenciával dolgozó szenzor adatok decimációját is megoldja. Valójában a ritkábban érkező adatok interpolálási feladatát is meg lehet így kerülni, de ekkor célszerűbb egy becslési algoritmus beiktatása.

## UAV szenzor adatfúziós példa

A továbbiakban tételezzük fel, hogy egy mikro UAV fedélzetére kettő, ugyanazon fizikai paramétert mérő szenzort telepítettünk, például magasságmérőt. Ugyanezen UAV fedélzetén GPS vevő is helyet kapott, mely szintén szolgáltat magassági adatot.

Ekkor a fedélzeti adatfúzió jelfeldolgozó algoritmusának folyamata a következő lesz:



7. ábra UAV fedélzeti magassági adatfúziós modell

Az algoritmus részlet három bemenettel rendelkezik és egy kimenete (szűrt magassági adat) jelölt. Természetesen az egyes „jelfolyamok” is felhasználhatók kimenetként, hiszen ezek memóriában átmenetileg tárolt adatok. A GPS adatok és a magasság mérők által szolgáltatott adatok összehasonlítás eredményeképpen előálló OK/-ERR jel elsősorban tájékoztató adatként szolgál és mindenképpen előző ütemből származik. Ha ez nem így lenne, az ábra szerint késleltetésmentes hurok keletkezne, ami a digitális jelfeldolgozásban nem realizálható. A magasságmérő és a GPS adatok elképzelhető, hogy más-más mintavételi frekvenciával állnak rendelkezésre, ennek feloldására (feltételezve azt, hogy a sűrűbb mintavétellel dolgozik a magasságmérő, mint a GPS vevő) került a folyamatba a „Decimáló” modul.

## ÖSSZEGZÉS, KÖVETKEZTETÉSEK

A mikroméretű UAV-k esetén a méretkorlátok és a szállítható maximális tömeg szűk korlátja szűk tervezési szabadságot biztosít az UAV-t építők, üzemeltetők számára. Egy biztos, hogy a biztonságról lemondani semmi áron sem szabad, hiszen egy néhány kilogramm tömegű, 50 – 100 km/h sebességgel repülő UAV jelentős anyagi károkat okozhat, a lezuhanó, irányíthatatlan eszköz pedig emberéletet is veszélyeztethet, kioltathat. Magyarországon ezt egy sajnálatos példa is alátámasztja [4.].

A biztonságos működés alapfeltétele a hibátlan sárkánytesten, hajtóművön kívül a hibátlan (hibatűrő) központi vezérlő, valamint az elektromos eszközök stabil elektromos tápellátása. A repülési adatokat mérő szenzorok meghibásodása, vagy zavarjel hatása miatt hibás repülési adatok érzékelése a visszacsatolt rendszer hibás működését eredményezi. A szenzoradatok szabályzó körbe vezetését megelőzően azokat zajtól szűrni kell, továbbá ellenőrizni kell azok hihetőségét. A hibás mérési adatokat a rendszerből ki kell zárni. A szenzor adatok szűrésére, az egyes szenzorok adatainak összevetésére a mikro UAV fedélzetén adatfúziós központot kell létrehozni. Az adatfúziós központ realizálása történhet úgy, hogy a redundáns központi vezérlőben –mint digitális jelfeldolgozó áramkörben- ezt a feladatot egy processz, vagy timer ütemezett task látja el, de történhet úgy is, hogy a központi vezérlő és a szenzorok közé beépítésre kerül egy mikrovezérlő, vagy DSP áramkör, ami kizárólag ezzel a feladattal foglalkozik. Az UAV adatfúziós központnak feladata továbbá az egyes szenzor meghibásodások detektálása, majd ezt az eseményt a helyzet súlyosságának függvényében a központi vezérlőnek jeleznie kell.

A központi vezérlő áramkör visszacsatolt repülés szabályzó körébe csakis szűrt, ellenőrzött és nagy valószínűséggel hibamentes (elfogadható hibahatáron belüli) adat kerülhet.

### FELHASZNÁLT IRODALOM

- [1.] WÜHRL Tibor: Mikro UAV-k vezérlő redundanciája, Fél évszázad forgószárnyakon a Magyar Katonai repülésben Szolnok, 2005.
- [2.] WÜHRL Tibor: Mikro méretű pilóta nélküli repülők repülésbiztonsági kérdései – elektromos tápellátás biztonsága, Szolnok, 2006.
- [3.] GORDOS G. TAKÁCS Gy.: Digitális beszédfeldolgozás (Műszaki Könyvkiadó, Budapest, 1983)
- [4.] 2006. május 13. Ócsény, nemzetközi modellbemutató  
{„Szabadlábbon a halált okozó repülőgépmoделlt irányító..” MN 2006. május 15.}
- [5.] MATLAB Ver. 7.0.1