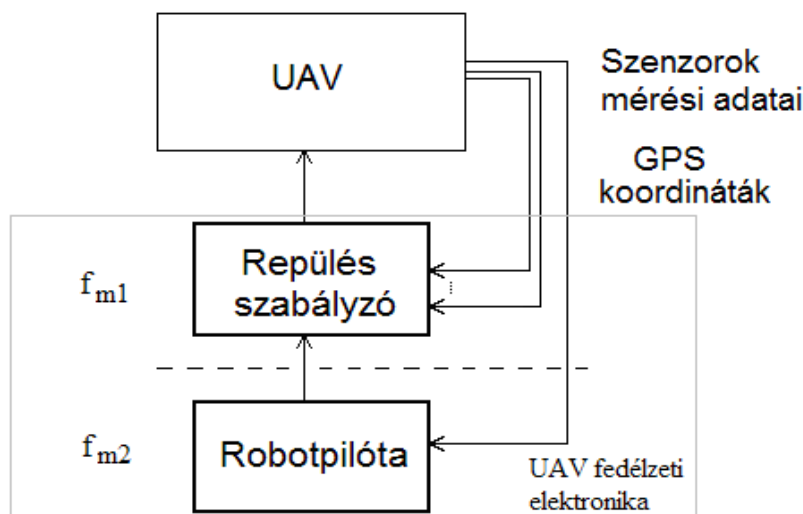


KISMÉRETŰ MODULÁRIS UAV FEDÉLZETI ROBOTIKA

BEVEZETÉS

A pilóta nélküli repülő (UAV-k) alkalmazásának jelenleg több akadálya van. Bár számos megoldás létezik [1], a széles körű elterjedés korlátját a repülés biztonság hiánya jelenti. Az UAV-k felhasználása és felhasználhatósága tekintetében számos tanulmány és publikáció látott napvilágot [2], [3]. A repülésbiztonság komplex fogalom. Egy **kisméretű UAV esetén**¹ a megbízható működést befolyásolja a sárkánytest, a hajtómű, az egyes beavatkozó szervek és a fedélzeti elektronika és annak tápáram ellátását biztosító áramkörök megbízhatósága. 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. A repülés automatizálásáért felelős fedélzeti elektronikát (robotikát) funkciók szerint két blokkra bonthatjuk szét. A repülés stabilitásért, kívánt repülési magasság eléréséért és tartásáért, valamint a fordulások lebonyolításáért, az állásszögek tartásáért a repülés szabályozó körök a felelősek. A repülés szabályozó körök úgynevezett követő szabályozást végeznek. A szabályozók számára az alapjelet pedig a robotpilóta modul állítja elő. Egy ilyen struktúrát az 1. ábrán láthatunk.



1. ábra. A kisméretű UAV fedélzeti elektronika modellje

¹ Néhány kg felszálló tömegű UAV-k;

A szenzoroktól érkező mérési eredmények szolgáltatják a visszacsatolást. A 1. ábrán látható egyszerűsített rendszermodell értelmében a repülés stabilitásáról a repülés szabályzó körök gondoskodnak, mely digitális jelfeldolgozó áramkörök (DSP) esetén a működéshez f_{m1} mintavételi frekvenciát feltételeznek. A stabilis repülés feltételeinek megteremtését követően a repülés szabályzó körök alapjelére kihatással levő „robotpilótát” definiálom, mely a pillanatnyi döntéseket a rögzített feladat, és a szenzor mérési adatok (elsősorban pozíció koordináták) alapján hozza meg. A robotpilóta kialakítása szintén digitális jelfeldolgozó áramkörökkel valósulhat meg, melynek mintavételi frekvenciája (f_{m2}) eltérhet a repülés szabályzó kör mintavételi frekvenciájától (f_{m1}). A különböző mintavételi frekvenciával működő jelfeldolgozó áramkörök szinkronizálása **decimálással**, vagy **interpolálással** oldható meg.

A gyakorlati megvalósítás, a repülés szabályozó és a robotpilóta realizálása történhet akár egy közös jelfeldolgozó processzorral, de a feladat logikusan szétosztható két különálló processzorra. **A kisméretű UAV-k esetén természetesen gondolni kell a kis felszálló tömegre²**, mely jelentős **tervezési korlátnak** számít. A fedélzeti robotika kétprocesszoros megoldása (melegtartalékolt esetben ez négy processzort jelent) viszont jelentős többlethasználattal jár, melyek a következők:

- Az elektronika fejlesztése során a kutató-fejlesztői feladatok jól elkülöníthetők;
- Az egyes modulok jól tesztelhetők;
- A robotpilóta és a repülés szabályozó kör közti „külső” kommunikációs csatorna könnyű továbbfejlesztést tesz lehetővé.

FEDÉLZETI ROBOTIKA MODULÁRIS HARDVER KIALAKÍTÁSA

- A repülést szabályozó kör realizálás;
- A robotpilóta áramkör megvalósítás;
- A repülés szabályozó kör és a robotpilóta modul kommunikációs interfészének definíciója.

A **repülést szabályozó kör** kialakítás célszerűen egy jelfeldolgozó processzorra épül, mely bemenő adatként szenzor adatokat fogad (repülési magasság, szöghelyzet, repülési sebesség), míg a kimenetén beavatkozó szerveket (kormányfelületek, hajtómű) vezérel. Kisméretű UAV-k esetén erre a célra megfelel a dsPIC33 család valamely tagja, például dsPIC33FJ256GP710 [6].

A fent említett jelfeldolgozó processzor 16 bites, fix pontos számábrázolással dolgozik, de az aritmetikai és logikai egységhez kapcsolódó akkumulátora (ACC-A és ACC-B) 40 bit méretű. Kifinomult DSP utasításokkal rendelkezik, számábrázolása lehet előjeles/előjel nélküli tört, vagy egész.

A CHIP-be beépített kerekítő és szaturációs logika segítségével a DSP túlcsoordulási hatások jobban kézben tarthatók. Motor vezérlésre (mint beavatkozó szerv vezérlés) a maximum 8 PWM³ csatorna áll rendelkezésre.

² CLASS0 szerint maximum 25kg [5]

³ PWM – Pulse-Width Modulator (Impulzus-szélesség modulátor)

A **robotpilóta áramkör megvalósítás** hardver igénye nagymértékben a működési elvárásoktól, felhasználói specifikációktól függő. A hardver kialakítása akár egy 8 bites mikrovezérlőtől a grafikus felületű térinformatikai adatbázist tartalmazó számítógépig is terjedhet (például PDA). Az előbbi esetben csak a fordulópontok koordinátáinak tárolására és letöltésére van lehetőség.

A robotpilóta a bemenetén pozíció koordinátákat fogad (például egy GPS vevőtől), és a kimenetén alapjeleket szolgáltat, amit a repülés szabályozó áramkör használ fel a működéshez.

A bevezetőben megadott 1. ábra értelmében a repülés szabályozó és a robotpilóta két különálló hardver modulként is megvalósítható, ezért elengedhetetlen az egyes részegységek kommunikációs szabályainak lefektetése. A kommunikáció fizikai közegének célszerű valamilyen régi jól bevált módszer alkalmazni. A csatlakozópontok minimalizálása, valamint az összekötő vezetékrendszer egyszerűsítése érdekében soros szinkron, vagy soros aszinkron kommunikációs interfész használatát javaslom.

A jelenlegi vezérlő modell értelmében két processzor egység kommunikál, vagyis az összeköttetés pont-pont jellegű, így akár két mikrovezérlő CHIP UART porton is összekapcsolható, de a továbbfejleszthetőség érdekében BUS topológia is kialakítható. Több részegység kommunikációjára jól bevált és általánosan elterjedt módszer az RS-485 [7], [8], vagy a CAN⁴ – BUS [9] kialakítás.

FEDÉLZETI ROBOTIKA ALGORITMUS KÉRDÉSEI

Fedélzeti kommunikációs protokoll

Az előzőekben láttuk, hogy a kommunikáció pont-pont relációban valósul meg a robotpilóta és a repülés szabályozó áramkör között, de elképzelhető, hogy erre a kommunikációs csatornára a későbbiekben más áramkörök is kapcsolódnak. A protokollt ennek szellemében célszerű lefektetni, vagyis a protokoll ne jelentsen fejleszthetőségi korlátot.

A kommunikációs protokoll kialakítása célszerűen az OSI⁵ réteg elvekkel összhangban készül. A fizikai réteg fölött az adatkapcsolati réteg található. Itt történik a keretszervezés, a címezés, valamint a hibaellenőrzés és hibakezelés. A címezés a pont-pont relációban elhagyható, sőt értelmetlennek tűnő, de mint már említettem a későbbi továbbfejlesztés esetén ez teremti meg a több eszköz használatának lehetőségét.

Egy lehetséges keretstruktúra (aminek felépítése majdhogynem szokásosnak is mondható, logikájában hasonló több ismert protokoll adatkapcsolati rétegéhez) a következőképpen nézhet ki:

SOF	Cím (forrás)	Cím (cél)	Keret típus	Mező hossz	PDU	CRC	EOF
-----	-----------------	--------------	----------------	---------------	-----	-----	-----

2. ábra. A kisméretű UAV fedélzeti kommunikáció keretszervezése

⁴ CAN – Contoller Area Network

⁵ OSI – Opens System Interconnection

Az SOF⁶ a keret elejét, míg az EOF⁷ a keret végét jelző egység. A keretben megtalálható a küldő eszköz címe (forrás cím), valamint a célcím is. A célcím lehet egyedi, de létezhet a BUS összes eszközét megszólító „broadcast” cím is. A keret típus segítségével megadhatjuk, hogy a küldött üzenet milyen típusú (például: kérés, válasz, nyugtázó stb.).

A felsőbb rétegek protokoll egysége a PDU⁸ mezőben kaphat helyet, mely hosszát a „mező hossz”-ban adjuk meg. A hibás, illetőleg a hibátlan vétel az adás és a vétel helyén is számított kóddal ellenőrizhető. Egyszerűbb esetben hatékony védelemnek bizonyul egy modulo aritmetika alapján akkumulált ellenőrző összeg, de megbízhatóbb, ha a hibavédelemre CRC⁹ kódot alkalmazunk, mely az ellenőrzött mező és az úgynevezett generátor polinom osztása során keletkező maradék. A fenti keretszervezés példában az ellenőrzött mező célszerűen a forráscímtől a PDU-ig tartó (és még azt is magába foglaló) mező.

A kommunikációs protokoll kialakításánál azt is figyelembe kell venni, hogy az egyes eszközök milyen megfontolás szerint tehetnek üzenetet a BUS-ra. Legegyszerűbb a dolgunk, ha a BUS úgynevezett Master – Slave relációban működik és a kommunikációs csatornán csakis egy Master funkciójú eszköz található. Több Master esetén, vagy egyenrangú eszközök létét feltételezve az adatkapcsolati rétegben további eljárások kidolgozása válik szükségessé. A közös közeg esetén ekkor ugyanis ütközések és versenyhelyzetek is kialakulhatnak, tehát ütközéskezelésről (esetleg ütközés elkerülést lehetővé tevő algoritmusról – mint például a telekommunikációból ismert WLAN¹⁰ CSMA/CA¹¹ algoritmus), prioritásokról is gondoskodni kell. Kisméretű UAV-k esetén ez utóbbi problémakör a kisteljesítményű processzor részegységektől jelentős erőforrást igényelne, ezért tehát célszerű az egy Master-es kialakítás mellett dönteni, és Master eszköznek pedig a repülést szabályozó köröket realizáló jelfeldolgozó processzort kinevezni.

Az üzenet tartalmakat a keret PDU mezői tartalmazzák, melyek felépítését az adatkapcsolati rész feletti rétegben kell definiálni (például ASN.1¹² ajánlás szerint).

Repülés szabályozó algoritmusok

A repülés szabályozó algoritmusok jelen esetben olyan digitális jelfeldolgozással megvalósított algoritmusok összessége, melyek memóriában eltárolt alapjelből, valamint szintén a memóriában előállított, megszürt, fuzionált szenzor adatok alapján (mint visszacsatolt jel) állítják elő a rendelkező jelet. A szabályozóköri algoritmusok tervezésével, szimulációjával számos irodalom foglalkozik [11] [12]. A legfontosabb feladatok pontokba szedve a következők:

⁶ SOF – Start Of Frame – keret kezdete

⁷ EOF – End Of Frame – keret vége

⁸ PDU – Protocoll Data Unit – Protokoll adat egység

⁹ CRC – Cyclic Redundancy Check

¹⁰ WLAN – Wireless Local Area Network

¹¹ CSMA/CA – Carrer Sense Multiply Access/Collision Avoidance

¹² ASN.1 – Abstract Syntax Notation number One

- Szenzor adat előszűrés;
- Szenzor adatok hihetőség vizsgálata;
- Szenzor adatok fúziója;
- Szabályozó körök realizálása (rendelkező jelek előállítás);
- Szenzor adatok hiányát kezelő algoritmusok;
- Beavatkozó szervek meghibásodása esetén futó vészhelyzeti algoritmusok.

A szenzor adatok előszűrése, szenzor adatok hihetőségének vizsgálata, az adatok fúziója a repülés biztonságát növelő tényezők, hiszen a rendszerbe így kis valószínűséggel kerülhet be hibás adat, ami alapján az algoritmus a döntéseit meghozza. A gondos szenzor adat feldolgozás esetén – ugyan kis valószínűséggel – is előfordulhat bizonyos létfontosságú adatok hiánya. Az ilyen hibák kezelésére teoretikusan előre definiált problémamegoldás jelenthet kulcsot. Az algoritmusokat célszerű valamely szabványos leírónyelvvel megadni, mely például lehet az SDL¹³ [13].

A magassági adat vesztést kezelő algoritmus például a következő lehet (két redundáns magasságmérő szenzort feltételezünk a fedélzeten) [14]:

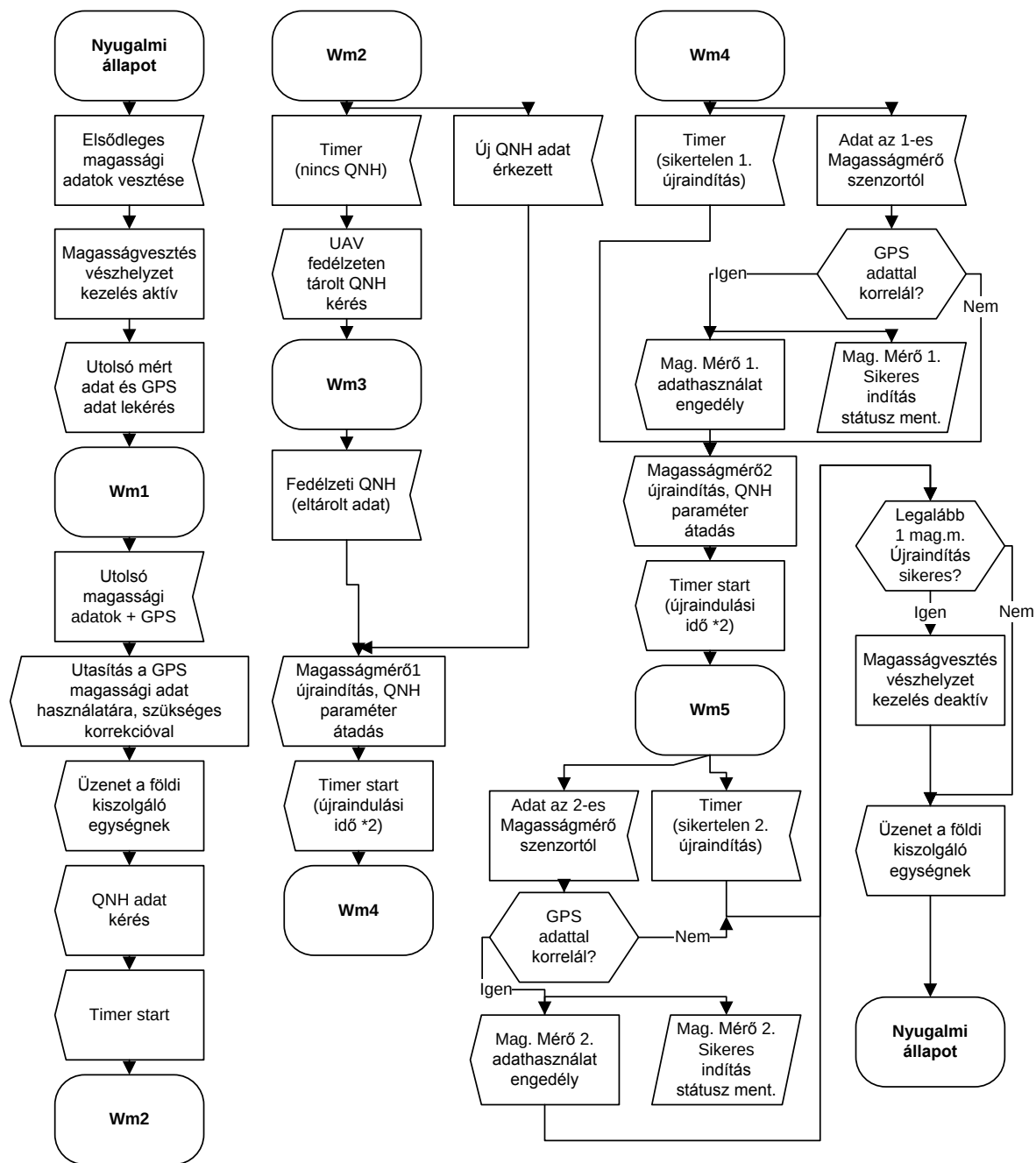
Az elsődleges magassági adatok vesztése esetén a másodlagos magassági adatok felhasználása válik szükségessé. A másodlagos magassági adatok pontossága kisebb, valamint ritkább időközönként frissülnek, mint az elsődleges mérési adatok. Ez a gyakoriság érték GPS vevő esetén 100ms – 1s időintervallumba esik.

Másodlagos magassági adat felhasználásáról rádiós csatornán értesíteni kell a földi kiszolgáló egységet és biztonsági okból a vészhelyzet kezelés időtartamára a repülési magasságot egy minimális, - vészhelyzet esetén meghatározott - magasságba kell emelni az UAV-t, ha az egyébként a normál küldetés során kisebb magasságban repülne.

Az elsődleges magassági adatok hiánya esetén elinduló vészhelyzet kezelő algoritmus a másodlagos magassági adatok felhasználás engedélyezését követően inicializálja az egyes számú magasságmérőt, valamint paraméterként átadja a QNH¹⁴ magasság-nyomás korrekciós paramétert. A QNH paraméter lehet az UAV fedélzetén tárolt paraméter, de az adat származhat a földi irányítótól is (például rádiós csatornán a kérés hatására felküldött érték). A kettes számú szenzor újraindítására is kísérletet tesz az algoritmus. Mindkét újraindítást időzítés védetten kezel, azért, hogy teljes szenzor hiba (hardver hiba, ami az újraindítást lehetetlenné teszi) esetén se fordulhasson elő egy átmeneti állapotba (Wm4, Wm5) ragadás. Abban az esetben, ha valamely szenzor elindul (1-es, vagy 2-es elsődleges magasságmérő), akkor a „magasság vesztés kezelés” jelzése deaktiválásra kerül.

¹³ SDL – Specification and Description Language

¹⁴ QNH - Quoted Nominal Height, Quantified Nominal Height



3. ábra A kisméretű UAV magasság adat veszteszt kezelő algoritmus

A beavatkozó szervek meghibásodásának kezelésére a fenti algoritmussal filozófiájában egyező algoritmusokat kell készíteni.

Robotpilóta algoritmusok

A robotpilóta algoritmusok az UAV fedélzeti, repülés szabályozó rendszerei számára állítják elő az alapjelet. Bemenő jelként az aktuális pozíció adatok értelmezhetőek, valamint a tréinformatikai rendszerből származó útvonal tervek.

A fontosabb robotpilóta algoritmusok a következők:

- Szenzor adat előszűrés (GPS);
- Szenzor adatok hihetőség vizsgálata (GPS);
- Szenzor adatok fúziója (GPS);
- Kapcsolat a térinformatikai rendszerrel, egyszerűbb esetben csak útvonalterv (fordulópont és magasságadat) tárolása;
- Alapjel előállítás a fedélzeti szabályozó körök számára;

Természetesen a robotpilóta egység feladata a szabályozó körök számára az alapjel előállítás. Hibás alapjel esetén az UAV eltéved, esetleg lezuhan, ezért a robotpilóta egységbe érkező adatokat itt is gondos előfeldolgozásnak, adatszűrésnek kell alávetni. Több rendelkezésre álló szenzor (például több GPS vevő modul) esetén az adatok fúziója jelentősen növeli a megbízhatóságot.

ÖSSZEGZÉS, KÖVETKEZTETÉSEK

A mikroméretű UAV-k esetén a méretkorlátok és a szállítható maximális tömeg szűk korlátja erősen behatárolt 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, kiolthat. Magyarországon ezt egy sajnálatos tragédia 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 kisméretű UAV fedélzeti robotikai egységének az alap működési funkciókon túl, számos működés biztonságot javító feladatot kell ellátnia. A központi vezérlő áramkör visszacsatolt repülés szabályzó körébe csakis szűrt, ellenőrzött és nagy valószínűséggel hibamentes (elfogadható hibahatáron belüli) adat kerülhet. 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 fedélzeti elektronika hibátlan és megbízható működésének számos további feltétele van. Nem szabad megfeledkeznünk arról, hogy hibás működést nemcsak kizárólag meghibásodás, hanem rejtett hibák bizonyos körülmények közti előbukkanása is okozhatja. Rejtett hiba lehet CHIP hiba is, de nagyobb valószínűséggel fordulhatnak elő szoftverhibák.

A fenti okból az egyes építőelemeket gondosan tesztelni kell. A központi vezérlő áramkört, mint az UAV fedélzeti robotikai áramkörét ezért célszerű két különálló egységként tervezni (repülés szabályzó kör és robotpilóta). A két egység között hibatűrő, könnyen bővíthető, adatátviteli redundanciával és hibafelismeréssel rendelkező kommunikációs protokollt kell kialakítani.

A modulárisan kialakított rendszer jó tesztelhetőséget és jó feladat szétoszthatóságot biztosít a fejlesztők számára.

FELHASZNÁLT IRODALOM

- [1] MOLNÁR ANDRÁS: MP2028 fedélzeti egységgel szerelt pilóta nélküli repülőgépek. Robothadviselés 6 Tudományos Konferencia, 2006. november 22.
- [2] KOVÁCS LÁSZLÓ – VÁNYA LÁSZLÓ: Az Európai Unió polgári célú pilóta nélküli repülőgépeinek kutatás-fejlesztési programja: az UAVNET. Repüléstudományi Közlemények elektronikus különszám, Fél évszázad forgószárnyakon a magyar katonai repülésben c. konferencia kiadványa, Szolnok, 2005. ISSN 1417-0604
- [3] KOVÁCS LÁSZLÓ – VÁNYA LÁSZLÓ: Pilóta nélküli repülőgépek kutatás-fejlesztési tapasztalatai Magyarországon. Hadtudomány, 2007. 2. szám. 50-61. p.
- [4] 2006. május 13. Öcsény, nemzetközi modellbemutató
{„Szabadlábbon a halált okozó repülőgépmodellt irányító..” MN 2006. május 15.}
- [5] IABG - Final Report; CARE Innovative Action Preliminary Study on Integration of Unmanned Aerial Vehicles into Future Air Traffic Management V1.1 7th December 2001. Dept. Airbone Defence;
<http://www.eurocontrol.int/care-innov/gallery/content/public/docs/studies2001/iabg-finalreport.pdf>
- [6] Microchip; dsPIC33F Family Data Sheet, High-Performance, 16-Bit Digital Signal Controllers (DS70165E)
www.microchip.com
- [7] Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems, ANSI/TIA/EIA-485-A-1998, Telecommunications Industry Association, 1998
- [8] MANNY SOLTERO, JING ZHANG, CHRIS COCKRILL: Application Report SLLA070c – June 2002; 422 and 485 Standards Overview and System Configurations <http://focus.ti.com/lit/an/slla070c/slla070c.pdf>
- [9] BOSCH, CAN Specification, Version 2.0; Robert Bosch GmbH 1991
www.semiconductors.bosch.de/pdf/can2spec.pdf
- [10] CAN Data Link Layer: ISO 11898-1 (2003)
- [11] SZABOLCSI RÓBERT: Automatikus repülésszabályzó rendszerek, Zrínyi Miklós Nemzetvédelmi Egyetem, Budapest 2004
- [12] SZABOLCSI RÓBERT: Automatikus repülésszabályozás – a kezdetektől napjainkig, Szolnoki Tudományos Közlemények VII, Szolnok, 2003, HU ISSN 1419-256x, CD
- [13] Specification and Description Language; ITU-T Recommendation Z.100
- [14] WÜHRL TIBOR: Kisméretű pilóta nélküli repülőgépek biztonságtechnikája, PhD értekezés 2007.