

Schuster György¹ – Terpecz Gábor² – Radnai Viktor³

MIKROKONTROLLEREK ALKALMAZÁSA AUTOMATA REPÜLŐ SZERKEZETEK BEN⁴

A járművekben a 80-as évek elejétől alkalmaznak mikrokontrollereket, ez az utóbbi másfél évtizedben elengedhetetlen tartozéka lett járműveinknek a mindennapi használatban - akár a földi, vízi, vagy légi járműveinket tekintjük. Ez az előadás ezeknek az elektronikus eszközöknek a követelményeknek megfelelő kiválasztásával, alkalmazásával, fejlesztési kérdéseivel és tesztelésükkel foglalkozik. A cikk ismerteti egy eljárást, amely kezdeti szakaszban segíti az alkalmazott algoritmusok kipróbálását.

MIKROCONTROLLER APPLICATION ON AUTOMATIC FLYING VEHICLES

Microcontrollers have been applied on vehicles boards since early 80s. Those devices have become essential parts of our vehicles in the last fifteen years, considering cars, ships or airborne machines. This presentation deals with selection, application, development questions and testing of those electronic parts. Moreover this paper shows a method that can help to check applied algorithms in early development phase.

BEVEZETÉS

A vezető nélküli járművek alkalmazására az igény egy jobban növekszik. Ennek több oka van, például az élőerő alkalmazásának csökkentése, illetve ennek a védelme. Az utóbbi időben ezeknek az eszközöknek fokozatosan növekedett az autonómiája, ahogy egyre több számítási intelligenciát lehetett a fedélzetre telepíteni.

Ennek a szintje már elérte azt, hogy egy kisméretű eszköz fedélzetén akkora számítási teljesítmény helyezhetünk el, amely lehetővé teszi egy teljes küldetés vezérlését – beleértve a váratlan és vészhelyzeti szituációkat is.

Ebben a bevezetőben eddig szándékosan nem említettük a légi robotokat, mert ezek a megállapítások minden autonóm járműre igazak. A légi eszközök esete speciális, mivel a ezek három dimenzióban mozognak.

Ez bizonyos szempontból többlet feladatot jelent, de azt is meg kell említenünk, hogy pontos navigáció esetén sokkal kisebb esély van az akadályokkal történő ütközésre, illetve ezek kerülésére, hasonlítsuk csak össze egy városi közlekedésben résztvevő gépkocsival.

A megfelelő fedélzeti vezérlőrendszer kialakítására célszerű eszköz a mikrokontrollerek alkalmazása.

¹ Dr; OE KVK MAI igazgató, schuster.gyorgy@kvk.uni-obuda.hu

² OE KVK MAI oktató, terpecz.gabor@kvk.uni-obuda.hu

³ OE KVK MAI szigorló

⁴ Lektorálta: Prof. Dr. Makkay Imre ny. okl. mk. ezredes, egyetemi tanár, Nemzeti Közszolgálati Egyetem Katonai Repülő Tanszék

Ebben az előadásban azt a fejlesztő munkát szeretnénk bemutatni, amelyet intézetünkben folytatunk ebben a témakörben.

Alkalmazott érzékelők és problémáik

Egy robotrepülőnek lényegében azonos műszerezettségűnek kell lennie, mintegy hagyományos IFR szabályok szerint repülő repülőgépnek, tehát szükségünk van a gép térbeli helyzetére, a repülő eszköz haladási irányú és függőleges sebességére, illetve a vonatkoztatási rendszerben történő dőlésszögekre és ezek változásaira.

Az elsődleges alkalmazott érzékelő a GPS. A GPS által szolgáltatott pozíció jelentős zajjal terhelt, a szolgáltatott pozíciók szórása 2-4m közötti értékeket mutatnak ez egyértelműen nem megfelelő a repülés alighanem minden szakaszán, de a leginkább kritikus a leszállás.

Ezért a GPS jelei igen alapos szűrésen kell átesniük. Továbbá az esetleges vételi kiesések is problémát jelenthetnek.

A nyomásérzékelők alkalmazása kézenfekvő, viszont az utóbbi időkig ezek mérete és kialakítása nem tette lehetővé, hogy kisméretű eszközön alkalmazzunk. Most lehetőség nyílt arra, hogy ezeket az érzékelőket a fedélzetre telepítsük, az egyetlen kérdés, hogy milyen módon tudjuk ezeket kalibrálni.

Ez a probléma azonban megoldható. További megjegyzés, hogy a függőleges sebesség mérésére nem kell külön érzékelőt használni, a magasság érzékelő jelének differenciál hányadosa tökéletesen megfelel. A tengely irányú orientációk célszerű eszközei a műhorizont, a csúszásjelző és az elfordulás jelző. Ezeket az információkat tengelyirányokban elhelyezett gyorsulás-érzékelők és a többi érzékelőkből származó jelek kombinációjából számítható.

További lehetőség a mágneses irányok használata. Ezek az érzékelők szintén zajjal terheltek. Például a gyorsulásérzékelők zajként veszik a motorok és a légcsavar(ok) által keltett rezonanciát. A közvetlen akadályok érzékelésre célszerű érzékelő egy 2D LiDar⁵.

Azért elegendő a 2D, mert nem célunk a térképezés, illetve nem cél a LIDAR által biztosított navigáció, azonban földközelségben az útirányban található, illetve a kikerülési irányban található akadályok felfedezése lényeges. Sajnos ez az egyetlen eszköz, amelynek az ára egyenlőre túl magas.

A fejlesztés során a többször felmerül a kérdés, hogy miért nem használunk ultrahangos érzékelőt. A válasz nagyon egyszerű. Ezek az érzékelők túlságosan függenek a környezeti körülményektől, nem beszélve az áramló levegő által keltett zajokról - különösen helikopter esetén.

Számítási igény

A fedélzeten alkalmazott számítástechnikai eszköz kiválasztásánál két kérdés merül fel, ezek:

1. Képes-e az egység az adott számítási feladatot elvégezni. Például szükséges-e lebegőpontos aritmetika, vagy az alkalmazott szószélesség elegendően nagy?
2. Képes-e az egység az adott feladatokat a real-time követelményeknek megfelelően elvégezni?

Tény, hogy az érzékelőink által szolgáltatott adatok alapvetően egészjellegűek, illetve analóg

⁵ LiDar Laser Imaging Detection and Ranging)



jelek. Ez azt sugallja, hogy elegendő olyan vezérlőt használni, amely úgynevezett fixpontos aritmetikával dolgozik.

Tapasztalataink szerint olyan aritmetikai műveletek esetén, ahol több osztás történik célszerű lebegőpontos aritmetikát használni.

A legfontosabb információt a GPS egységből és a sebesség szenzorból kapjuk. További járulékos és kiegészítő információk a gyorsulás érzékelőkből és a magasságmérőből származnak.

A GPS 2-4m-es szórást mutat tengelyenként. Ezt mindenképpen korrigálni kell, mert, vagy le-
rontjuk a pozicionálás pontosságát - amely nagy magasságban tulajdonképpen megfelel –, vagy
valamilyen szűrő algoritmust választunk.

Lehetőségek:

- klasszikus FIR digitális szűrő;
- klasszikus IIR digitális szűrő;
- Kálmán filter.

A tapasztalatok azt mutatják, hogy a Kálmán filter adja a legjobb eredményt, viszont relatív nagy számítási igénye van.

Célszerűen a minden olyan bemeneti jel, amely zajjal terhelt ezt a szűrőalgoritmust használja. Ez egész egyszerűen programozási kérdés, csak egy algoritmust kell implementálni és tesztelni, ezután csak paraméterezni kell.

A repülőeszköz irányítására fuzzy algoritmusokat használunk, ezen belül is a Mamdani MMG döntést használjuk. A számítási igény nem túl magas [2]. A legnagyobb számítási igényt a súlypontok számítása jelenti. Ez alapvetően négy kimeneti jelet jelent.

A bemeneti jelek:

- GPS x;
- GPS y;
- GPS z;
- gyorsulás x;
- gyorsulás y;
- gyorsulás z;
- gyorsulás horizontális tangenciális;
- mágneses irány (elektronikus iránytű);

amennyiben légnyomásmérésen alapuló műszereket is használunk:

- magasság;
- sebesség.

A felsorolt bemeneti információk közül a GPS adatok elsődleges navigációs információt jelentenek és a feldolgozásuk az úgynevezett kívánt pozíció és a valós pozíció eltérése alapján történik ezért az X, Y és Z irányban 5 elemű fuzzy klasztert tervezünk.

A GPS adatok nem közvetlen kormány akciókat jelent, hanem irány követési információt szolgáltatnak. Ez azt jelenti, hogy nem azt mondja meg például, hogy az UAV oldalkormányát mennyire kell kitéríteni, hanem azt, hogy mekkora iránykorrekcióra van szükség - mondjuk 10

fok eltérés vízszintes irányban.

A sebesség adatok származtathatók a GPS adatokból, ezeket - a már egyszer szűrt adatokat célszerű újra szűrni. Az a tény, hogy a GPS abszolút koordinátákat szolgáltat és az ebből származó sebességi adatok szintén az abszolút koordináta rendszerhez viszonyítottak.

Ha lehetőség van légsebesség mérésre, akkor két sebességi adattal célszerű dolgozni. Az egyik az abszolút sebesség, amely a kívánt abszolút sebesség és a valós abszolút sebesség eltérése. A valós légsebesség alapvetően az aerodinamikai jellemzőkkel kapcsolatos döntésekben játszik szerepet. Ennek széles időben történő repüléskor van fokozott jelentősége.

A kívánt és a valós abszolút sebesség eltéréséből származó motor szabályozás bemeneti szabályrendszere 5 szabályból áll. Ezt a szabályrendszert egy befolyásolja a légsebességből származó 9 szabályos szabályrendszer. A kimeneti szabályok klasztere 5 elemű. A gyorsulásérzékelők adatait igen alaposan kell szűrni, mert adataik egyszeres vagy kétszeres idő szerinti integráláson keresztül további állásszög, dőlésszög és elfordulás információkat biztosítanak.

A fentiek alapján ezeket a jeleket numerikusan integrálni kell. Mivel a mintavételezés elég sűrű első próbálkozásként egyszerű korrektív trapéz eljárást alkalmazunk.

A fentiek alapján a GPS x, y, z , a sebesség abszolút, a légsebesség és a statikus légnyomás továbbá a gyorsulás érzékelők jelei igényelnek fokozott szűrést, erre Kálmán szűrőket célszerű használni (12 darab).

A célszerű mintavételezési idő 10 ms. Ez azt feltételezi, hogy lebegőpontos aritmetikával rendelkező processzort kell választani.

Mikrokontroller választás

A számítási igény magas, viszonylag sok lebegőpontos műveletet kell elvégezni rövid idő alatt. Ezért csak olyan mikrokontroller jöhet szóba, amely lebegőpontos támogatással rendelkezik.

A lehetőségek:

- DSP processzor. Előnye, hogy a mátrix műveleteket rendkívül gyorsan hajtja végre. Hátránya az, hogy általános feladatok elvégzésére nem optimális;
- 32 bites lebegőpontos processzor. Előnye, hogy minden szempontból kielégíti az igényeket, képes általános célú operációs rendszer futtatására. Hátránya, hogy többségük tokozása úgynevezett BGA, amely speciális beültető technológiát kíván. Másik problémájuk, hogy a programozásuk operációs rendszer támogatás nélkül bonyolult;
- beültető. Ilyen eszközök az ARM Cortex Ax magos eszközök. Ezekre a kontrollerekre többféle operációs rendszer implementálható és teljesítményük is elegendő a feladat elvégzésére. Ilyen mikrokontrollerek például az ARM Cortex M3 magos áramkörök.
- 32 bites mikrokontroller lebegőpontos támogatással. Előnyük az egyszerű beültetés, megfelelő lebegőpontos számítási teljesítmény, kis operációs rendszerek megfelelő támogatása. További előny, hogy ezek az eszközök az ipari interfészeket ezek az eszközök támogatják. Például ilyen mikrokontrollerek az ARM Cortex M4 magos áramkörök.

A motor, a szervók és a fedélzeti kiegészítő feladatok ellátására célszerű egy egyszerű PWM

támogatással rendelkező 32 bites kontrollert választani. A könnyebb építés, fejlesztés és hiba-keresés érdekében ez a controller legyen egy külön egység. Célszerű választás ARM Cortex M0, M3 magos eszközök.

Ha a fedélzetre speciális érzékelőket telepítünk, például valamilyen digitális képrögzítő eszköz, akkor célszerű 32 bites lebegőpontos eszköz alkalmazása. Ekkor azonban lehetőség van a kereskedelemben kapható kész kártyák alkalmazására, amelyek az egész egyszerű eszköztől a többprocesszoros eszközökig – viszonylag elérhető áron – beszerezhetők. Amennyiben ezen eszközök számítási teljesítménye kevésnek bizonyul, lehetőség van ezek klaszterbe kötésére [1]

Ha a GPS navigációt kiegészítjük LiDar eszközzel, amely ismert környezetben lehetővé teszi a navigációt, illetve az akadály elkerülést. Ezt az eszközt is önálló feldolgozó egységhez kell kötni, mivel sebessége akkora számítási kapacitást köt le, hogy a többi feladatra nem maradna kellő idő.

Ha az teljes navigációt LiDar-ra bizzuk, akkor az akadály felismerés és a leképezett tér komparálásához mindenképpen nagy teljesítményű eszközre van szükség.

Az alkalmazott számítástechnikai eszköz, a térképezett területtől, a felbontástól és repülőeszköz sebességétől függően változó teljesítményt igényel. Ekkor megfontolásra kerül a klaszter alkalmazása. A járulékos probléma a klaszter real-time használata.

Ezt a problémát úgy tudjuk feloldani, hogy a klaszter fölé egy "karmestert" rendelünk, amely az információk be - illetve kijuttatását ütemezi.

Ha a klaszter teljesítménye elég arra, hogy a mintavételezési időn belül képes eredményt szolgáltatni, akkor a megoldás megfelelő [3].

Fejlesztési és tesztelési módszerek

Első pillantásra a szoftver fejlesztésére számos lehetőség adott, azonban három szempontot tartottunk szem előtt, melyek a következők:

- a fejlesztői környezet legyen ingyenes,
- az alkalmazott fordító és hibakereső programok legyenek kompatibilisek egymással és a PC-n futó programmal,
- legyen elegendően hatékony és támogatott, hogy az adott feladatot el lehessen végezni bennük.

Az első szempont nem is kérdéses, az egyetlen probléma az, hogy létezik-e ilyen környezet. A válasz nagyon egyszerű, a gcc és a gdb adott rocesszorra kifejlesztett változatai rendelkezésre állnak. Ha valaki integrált fejlesztői környezetet szeretne használni, akkor javasolt eszköz az Eclipse. Az alapértelmezetten alkalmazott fejlesztési környezetünk a megfelelő gcc fordítóprogram és az ehhez tartozó gdb. Ez a fejlesztő kollégák túlnyomó többségénél megfelelő.

Következő kérdés, hogy a fordítók miért legyenek kompatibilisek. Számos olyan eljárásunk és algoritmusunk van, melyek PC-n kényelmesen ellenőrizhetők, tesztelhetők, de a cél rendszeren csak nagyon nehezen. Ilyen eljárás például a Kálmán szűrő, illetve az egyes fuzzy döntési eljárások.

A harmadik szempont szintén lényeges. Mivel az alkalmazott fejlesztési eszköznek az adott mikrokontroller számára támogatottnak kell lennie. Tehát képesnek kell lennie az adott – processzor specifikus – bináris kód előállítására, illetve ennek debuggolhatónak kell lennie. A

fuzzy szabályrendszerek generálását egy egyszerű grafikus tervezői felülettel rendelkező szkript állítja elő. A tesztelési eljárások mindegyikét az adott környezetek biztosítják. A funkcionális tesztelésre azonban Radnai Viktor szigorló hallgató munkája alapján egy alapvető funkcionális tesztelési eljárást alkalmazunk.

A módszer lényege, hogy a Flight Gear nyílt forráskódú repülés szimulátor modelljeit használjuk fel oly módon, hogy a vezérlési interfészeket a PC felületeivel kötjük össze, így elsődleges tesztelést tudunk végrehajtani.

Kérdés az alkalmazott modell helyessége. Egyenlőre a fejlesztett UAV modellje még nincs kész, így egy Cessna 172 modelljét használjuk fel.

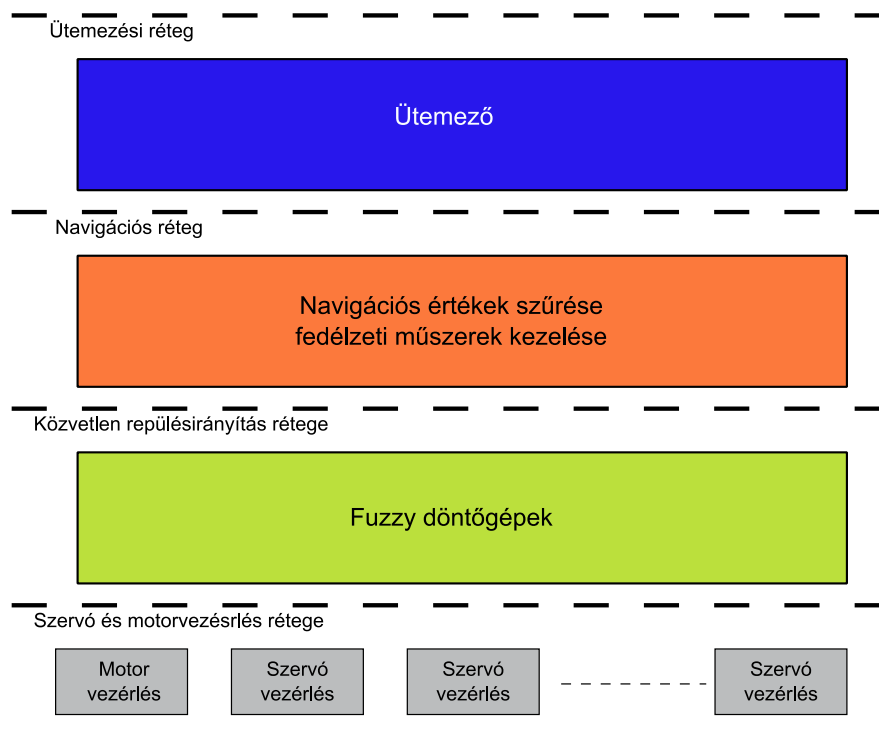
Javasolt hardver architektúra

A hardvert célszerű rétegesen felépíteni, ennek okai:

- minden egyes hardver elemet külön-külön lehet tesztelni, kezdve például a szervó vezérlőket. Miután minden funkciót leellenőriztünk lehet a következő réteget felépíteni. Ezen kívül minden egyes rétegben található eszközök funkcionálisan - kiépített teszt-környezetben – ellenőrizhetők;
- második szempont, hogy a hardveren egyszerre több tervező és építő dolgozhat a megadott specifikációk alapján.

Hátránya a réteges felépítésnek, hogy az összes hardver igény megnövekszik, a hardver megbízásodás valószínűsége is növekszik, továbbá a rétegek közötti kommunikáció is „sebezhető”.

A javasolt felépítés:



1. ábra A navigációs rendszer elvi vázlata

A rétegek:

- a szervó és motorvezérlés rétege. Ebben a rétegben csak és kizárólag a motor fordulatszám szabályozása és a szervók mozgásának elektronikai találhatók;
- a közvetlen repülésirányítás rétege. Itt vannak azok a fuzzy döntőgépek, amelyek biztosítják, hogy a kormányokat a rendszer úgy kezelje, hogy az UAV a megfelelő pályát járja be. Ezen réteg feladata a vészhelyzeti eljárások kezelése is;
- a navigációs rétegben vannak az alap navigációs funkciók, illetve az érzékelők szűrőalgoritmusai is. Itt kerülnek szűrésre a GPS jelei és a légnyomáson alapuló műszerek jelei. Ha van LiDar az ütközés elhárítás döntése is itt történik meg. A réteg másik funkciója, hogy a kívánt pályát és a valós pályát összevesse és parancsot küldjön az alatta lévő rétegnek, hogy az UAV kerüljön vissza a kívánt útvonalra;
- a ütemező réteg feladata a rendszer navigációs rétegének ütemezése.

ÖSSZEFOGLALÁS

A fejlesztés jelen esetben túl van a szervó és motorvezérlő áramkörök tesztelésén. Jelenleg a kormányvezérlési modellel kísérletezünk a Flight Gear használatával.

A szűrőalgoritmusok beállítása is folyamatban van, egy tablet segítségével földfelszínen gyűjtjük a GPS információkat, de az idő javulásával kísérletet szeretnénk végezni egy moovit fedélzetén is.

Legnagyobb problémánk a LiDar rendkívül magas ára, ezért kollégáink egy egyszerű eszköz kifejlesztésébe kezdtek, amely 2D-ben képes kis pontosságú térképezésre elfogadható sebességgel.

FELHASZNÁLT IRODALOM

- [1] Walbrzych: MPI HYBRID CLUSTER. X. Student Science Conference Wroclaw Polytechnica Conference 2012 Poland pp. 48-52.
- [2] J. Tick, J. Fodor: Fuzzy Implications and Inference Processes COMPUTING AND INFORMATICS ISSN: 1335-9150 24: 6 pp. 591-602. 2005
- [3] JÓZSEF Kopják - JÁNOS Kovács: Timed cooperative multitask for tiny real-time embedded systems IEEE 10th Jubilee International Symposium on Applied Machine Intelligence and Informatics, Herl'any, Slovakia, ISBN:978-1-4577-0197-9 pp. 377-382