

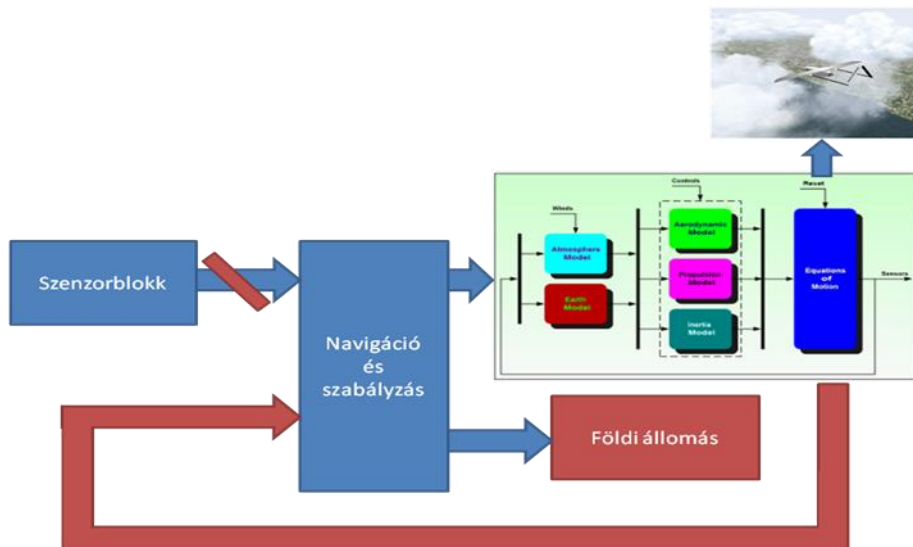
HIL SZIMULÁCIÓ ROBOTPILOTA FEJLESZTÉSÉBEN

Bevezető

Pilóta nélküli robotrepülőgéppel végzett kutatás és fejlesztés elengedhetetlen része a tesztelés. Az időjárási feltételek miatt a tesztelést csak szűk feltételek mellett lehet végrehajtani, illetve közben a teljes repülőeszköz, a robotpilóta és a hordozott teher fokozott veszélynek van kitéve.

Tisztán szoftveres szimulációval (pl. Matlab-Simulink) tesztelni lehet ugyan egyes szabályozási elveket, megvalósításokat, de a szimuláció a fedélzeti szoftver illetve hardver hibákra nem mutat rá. Az ideális tesztelést a HIL (Hardware in the loop) szimuláció jelenti, ahol a robotpanel (hardver) össze van kötve a szoftveres szimulációval, így a rendszer válasza a valóságos roboteszköztől érkeznek.

A szoftveres modell fogja kiszámítani a repülés összes jellemzőjét (pozíció, orientáció, repülési adatok stb.), majd az adatokat meghatározott mintavételezési frekvenciával továbbítja a robotpanelnek, amelyet az feldolgoz (1. ábra). A robot nem a saját fedélzeti szenzorai jelét használja a szűrési, navigációs és szabályozási algoritmusokhoz, hanem a modell által szimulált értékeket.



1. ábra. HIL szimuláció

A robotpanel miután kiszámította a beavatkozó jeleket (kormánykitérések, gázállás) visszaküldi azokat a modell számára, amely frissíti a szimulációt.

Megfelelően beállított szimuláció esetén rekonstruálni lehet valós repülések mérési adatait, amivel validálni lehet a rendszert. A valóságnak megfelelő rendszer esetén a szimulációt fel lehet használni szabályzókörok hangolásához, veszélyes helyzetek, manőverek előzetes teszteléséhez, illetve különböző, szélsőséges időjárási körülmények között végrehajtott repülések vizsgálatához.

Robotrepülőgép tesztelési folyamata

Egy robotrepülőgép fejlesztése és tesztelése hosszú folyamat. Első lépésben az üres repülőgépet manuális módon kell berepülni, megismerni a repülésének jellemzőit (pl. fel és leszálló sebesség, utazó sebesség, átesési sebesség, manőverező képesség stb.). Amint a jellemzők adottak, ezekből illetve az elérendő célokat figyelembe véve ki kell választani megfelelő szabályzási és navigációs eljárásokat. Szoftveres szimulációkat kell végezni az egyes szabályzókörökre, ki kell számítani az optimális szabályzási paramétereket, majd a robotpilóta vezérlőjén implementálni kell azokat.

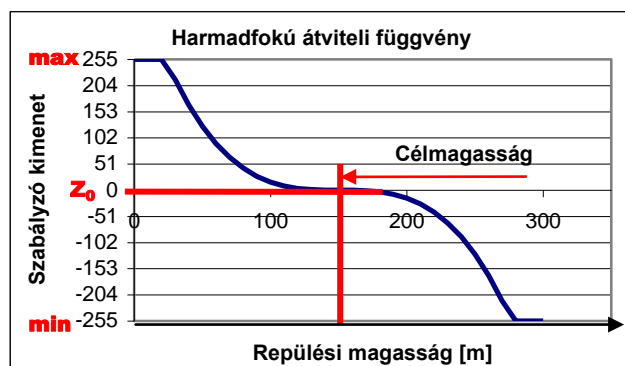
Autonóm üzemmódban a repülőgép kormányfelületeit mozgató aktuátorokat és a sebességszabályzót (a teljes repülőgépet) a robotpilóta irányítja. A szabályzókörök független hangolásakor a robotpilóta mindig csak az aktuális tesztelendő kormányfelületet tudja mozgatni. A félautonóm üzemmódot felhasználva a robotpilóta csak az adott kormányfelülettel rendelkezik. Alapbeállításnak az előzetes szimulációkból származó erősítési tényezőket ajánlott felhasználni.

Rendkívüli fontossággal bír, hogy a szabályzóköröket (magasság-, sebesség- és iránytartás valamint navigáció) és a giroszkóp erősítési tényezői egyenként, egymástól függetlenül legyenek kezdetben beállítva. Érdemes a sebesség és magasságtartás függvényeket hosszú (800-1000m) egyenesekben beállítani, figyelve arra, hogy fordulóban se romoljanak el a repülési értékek. Ha a szabályzókörök tökéletesen működnek függetlenül, akkor kerülhet sor az egyesített teszttükre. Ha jól vannak megválasztva a paraméterek, akkor a robotpilóta adott sebességgel a célértéknek megadott magasságban tartja a gépet a kijelölt útvonalon. Ennek ellenére ekkor még a robotpilóta beállítása nem fejeződik be. Az egymástól függetlenül beállított szabályzók együttes alkalmazása ugyanis egymásra hatásokat eredményez, amik a rendszer további, úgynevezett finomhangolását teszik szükségessé.

Megfelelő szabályzási elv kiválasztása

A robotrepülőgépen alkalmazni kívánt szabályzóknak kiválasztása során több szempontot kell figyelembe venni. Sok érv szól a klasszikus PID szabályzók mellett. Ilyen elvű szabályzásokat számos robotpilóta egység alkalmaz [1, 2]. Ezek a rendszerek azonban vagy nagyon bonyolultak (MP 2028), vagy erős korlátozással alkalmazhatók a repülőgép nemlineáris szabályzási feladatainak megvalósítása során.

Az AERObot rendszerében egy speciális harmadfokú átviteli függvénnyel jellemezhető szabályzórendszer került alkalmazásra. Ennek előnye részben az,



2. ábra. a magasságszabályozás átviteli függvényének értelmezése

hogy nemlineáris rendszerek szabályzására is alkalmas, másrésztől több éves tapasztalattal rendelkezünk a szabályzó gyakorlati alkalmazása terén.

A szabályzóban alkalmazott átviteli függvények értelmezése a 2. ábrán, a magasságszabályozás kapcsán a következők szerint értelmezhető. A diagramon meg lett jelölve a célmagasság ($X_c=150$ m), valamint az ahhoz tartozó neutrális kimeneti érték ($Z_0=0$). Amennyiben a célmagasság értéke magasabb a pillanatnyilag mért magasságnál, akkor a kimeneti jel értéke magasabb lesz, mint a neutrális kimeneti érték. Abban az esetben, ha a célmagasság kisebb, mint a pillanatnyilag mért magasság, akkor a kimeneti jel értéke alacsonyabb lesz a neutrális kimeneti értéknél. Mivel a függvény értékei a célmagasságtól eltérve a \pm végtelenbe tartanak, a gyakorlati alkalmazhatóság érdekében a kimenő értékét egy előre definiált maximum és minimum értékek közé kell korlátozni. Az 1. ábrán látható függvény olyan kormány szabályozásához alkalmazható, amely szimmetrikus kitérésű. Ilyen kormány lehet a magassági- az oldal- vagy a csűrő-kormányfelület. A hajtómű teljesítményének vagy az ívelőlapok szabályozásának a függvényt úgy kell módosítani, hogy annak minimális kimenőjele legyen nulla ($Z_{\min}=0$) és maximális kimenőjele legyen $Z_{\max}=\max$. Ebben az esetben a függvény „lapos” szakasza legyen a szabályozni kívánt értékhez tartozó kimenőjel szintje. Például, ha a motorteljesítmény szabályozása a cél, akkor a függvény inflexiós pontjához tartozó kimenőjel (Z_0) szintje éppen akkora motorteljesítményhez kell, hogy tartozzon, ami az adott szabályozási szinthez szükséges.

Az ismertetett szabályzó közvetlenül alkalmas a repülőgép sebesség és magasság szabályozására és kiegészítéssel az iránytartásra.

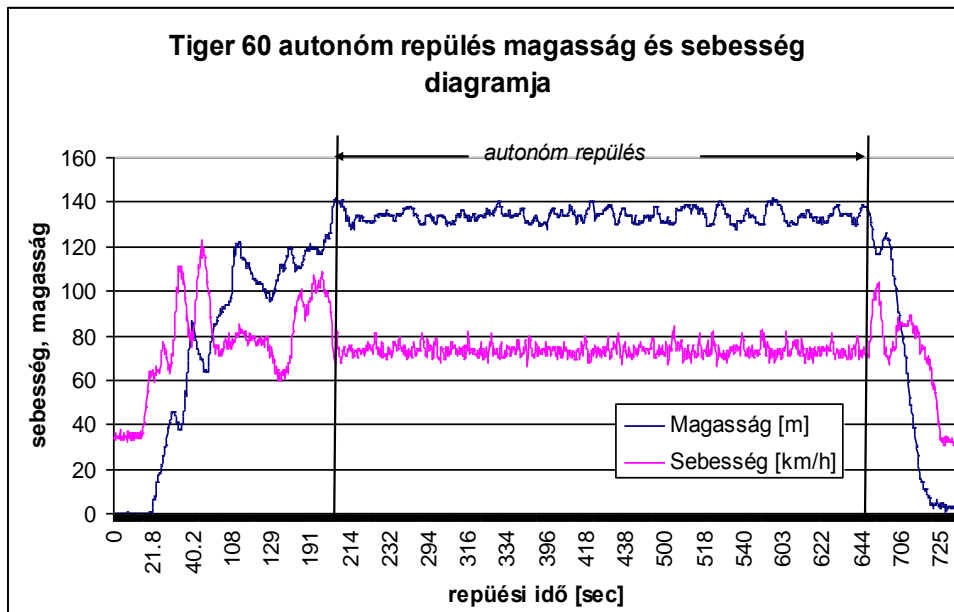
A különálló szabályzókat szimulációval egyszerű vizsgálni különböző peremfeltételek mellett, de az egész rendszert, a szabályzókat egymásra hatását csak oly módon lehet vizsgálni, ha rendelkezésre áll a repülőgép valamint a szabályzás és navigáció teljes matematikai modellje, amely tartalmazza az aerodinamikai elemeket is.

Az előzetes számításokat, a paraméterek hangolását az ideális matematikai modellen célszerű végrehajtani, majd a robot hardveren implementálva lehet azon is tesztelni.

Szabályzás és navigáció éles tesztelése

Az Óbudai Egyetem Neumann János Informatikai Karán fejlesztett Tiger60 robotrepülőgép (AERObot robotpilótával) fejlesztése és tesztelése az említett klasszikus módon történt. A repülőgép kis hibával képes a beállított célértékek (150m starthely feletti magasság, 80 km/h repülési sebesség) mellett képes autonóm útvonalrepülésre. Az 3. ábrán megfigyelhető a repülőgép ofszet hibája mind a repülési sebesség, mind pedig a repülési magasság esetében.

A robot hardveren azonban rengeteg olyan potenciális, előre nem látható hibalehetőség lehet (elektronikai, szoftveres stb.), amelyet valószínűleg nem ír le a matematikai modell. A robot éles tesztje alatt, repülés közben egy ilyen hiba végzetes kimenetelűvé válhat.

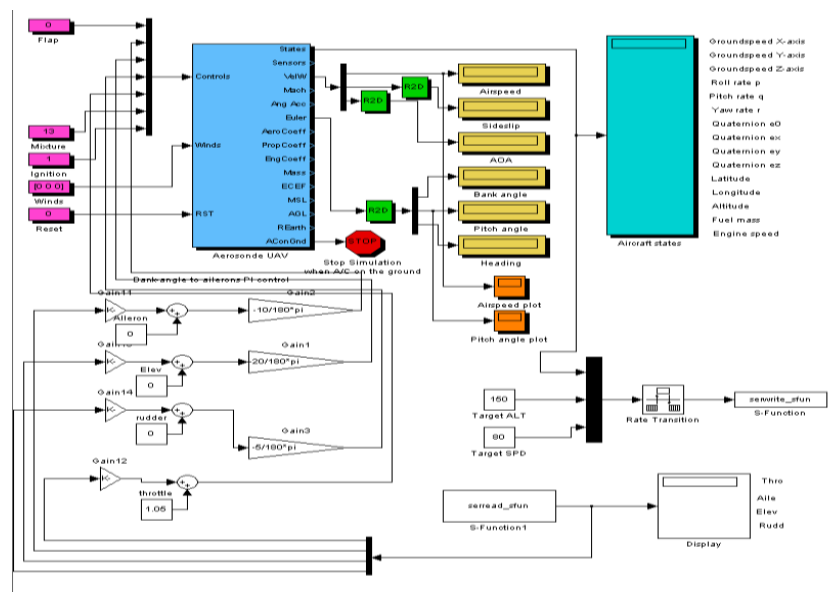


3. ábra. ofszet hiba a szabályozott szakaszon

HIL szimuláció

HIL szimulációval éles repülési teszttel közel azonos állapotot lehet létrehozni. A robotpilóta ugyan azt csinálja, mint éles helyzetben, nincs róla tudomása, hogy valójában a bemeneti adatait nem a saját szenzorairól, hanem egy szoftveres szimulátorról kapja.

A számítógépen létrehozott matematikai modell fogja (4. ábra) kiszámítani a repülés összes jellemzőjét (pozíció, orientáció, sebességek, magasságok stb.) az adott bemeneti paramétereket figyelembe véve, majd az adatokat meghatározott



4. ábra. Matlab modell

mintavételezési frekvenciával továbbítja a robotpanelnek, melyet az feldolgoz. Nem a saját fedélzeti szenzorok jelét használja a szűrés, navigációs és szabályzási algoritmusokhoz, hanem a modell által előállítottat.

A modell kimenetét (pozíció, orientáció, repülési sebesség és magasság stb.) a szimulációs program (pl. matlab) soros porton kiküldi adott frekvenciával a robotnak. Ez a frekvencia a robot

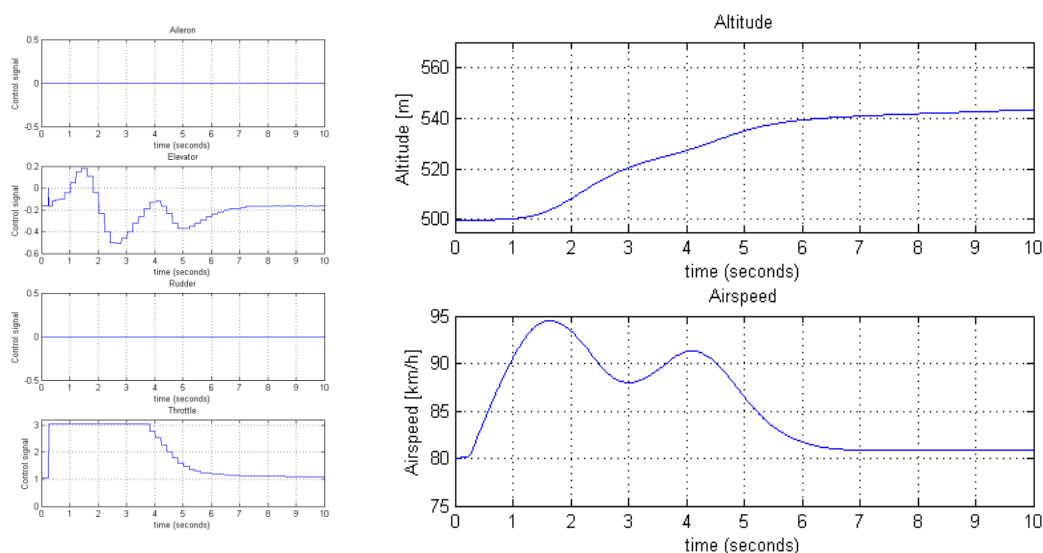
szabályzóköreinek a frissítési frekvenciájával kell, hogy egyezzen (diszkrétizált szabályzás a folyamatos idejű modellen). Mivel a szimuláció általában a valós idejű végrehajtásnál lassabb, ezért a robot programját célszerű adatvezérelté tenni. Ez azt jelenti, hogy az időzítéseket nem a saját belső órájához igazítja, hanem a szimulációs programtól beérkezett üzenetekhez illeszti.

A szimulált értékek lehetnek ideálisak, vagy (generált) zajjal terheltek. Amennyiben ideális értékekkel számol a rendszer, a robot belső szűrő algoritmusait ki lehet kerülni. Ellenkező esetben a HIL szimuláció alkalmas a robot rendszer fedélzeti szűréseinek kontrollált vizsgálatára különböző bemenetek és helyzetek esetén.

A robot a kapott értékek és a navigációs célértékek alapján kiszámítja a beavatkozó jeleket, amiket nem csupán az aktuátoroknak küld el, hanem a visszaküldi a szimulációs programnak is. Ezek a vezérlő értékek lesznek a modell bemenetei, amelyek alapján a modell kiszámolja a következő időpillanatnak megfelelő állapotot.

A HIL szimuláció validálására elkészült a Tiger 60 modellje a Matlab AeroSim blockset [3] segítségével. Mivel az említett robotrepülőgépről és az alkalmazott szabályzási megoldásairól rengeteg tapasztalat és mérési adat áll a rendelkezésünkre, ezért a szimulációs eredményeket és értékeket össze tudtuk hasonlítani a valós mért eredményekkel. Az alkalmazott blockset rendelkezik egy olyan interfésszel, amely segítségével a szimuláció repülési értékei grafikus formában ábrázolhatók a FlightGear szoftveren keresztül.

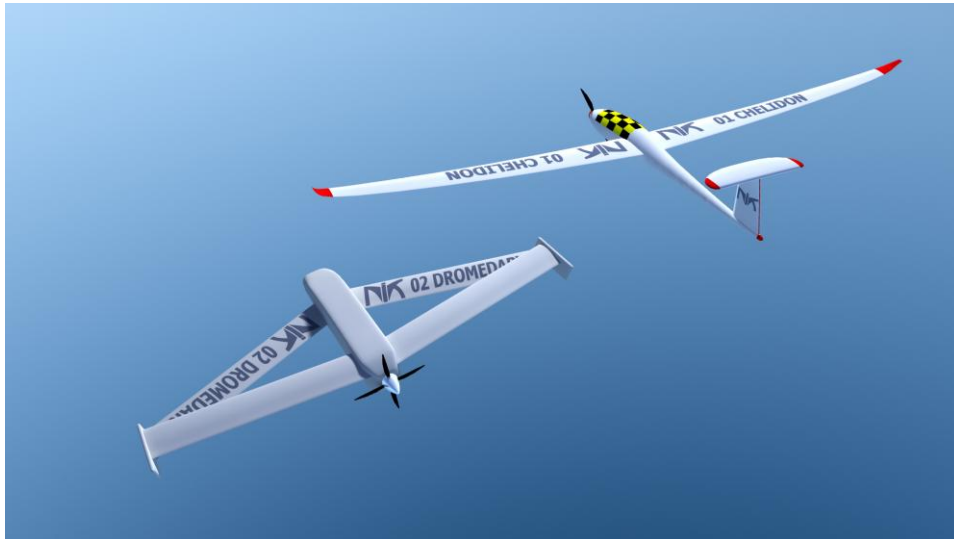
A szimulációs a valós tesztekkel egyező eredményt adott. A szabályzókörok diszkrét jellege és a szabályozott értékek jellemző offset hibája jól megfigyelhető a szimulált repülés diagramján (6 a/b ábra).



6 /a,b ábra: beavatkozó jelek, valamint repülési magasság és sebesség (magasság szabályzás, emelkedés 500m-ről 550m-re)

Továbbfejlesztés

A kidolgozott HIL szimuláció alkalmas lehet a következő, jelenleg fejlesztés alatt álló, a Tiger 60-nál jóval nagyobb méretű (4 m feletti szárnyfeszítávolságú) robotrepülőgépek (Chelidon és Dromedary, 7. ábra) valós repülés előtti tesztelésére, finomhangolására. Ehhez szükséges a modell finomítása, amely magában foglalja a repülőgépekben alkalmazni szándékozott hajtáslánc és sárkányszerkezet további vizsgálatát, és matematikai leírásának pontosítását.



7. ábra: Chelidon, Dromedary UAV látványterv

IRODALOMJEGYZÉK

- [1] MP2028g Installation and Operation, MicroPilot 1998
- [2] Chris ANDERSON: ArduPilot 2.x manual, 2009
- [3] AEROSIM BLOCKSET, Unmanned Dynamics, LLC, 2002