

Regula Gergely

regula@iit.bme.hu

Lantos Béla

lantos@iit.bme.hu

BELTÉRI AUTONÓM NÉGYROTOROS HELIKOPTER SZABÁLYOZÓ RENDSZERÉNEK KIFEJLESZTÉSE ÉS HARDWARE-IN-THE-LOOP TESZTELÉSE

Absztrakt

A cikk egy négyrotoros helikopter kinematikai modellezésétől a szabályozó algoritmusának kifejlesztéséig felmerülő elméleti és gyakorlati problémákat tárgyalja. Részletesen bemutat egy új alacsony szintű szabályozási algoritmust. Ezen kívül bemutatja a szabályozás kiegészítő komponenseit, az állapotbecslést és egy pályakövető algoritmust. A helikopterről felállított modell figyelembe veszi a légellenállást, a rotorok giroszkopikus hatását, valamint a helikopter rotorjait meghajtó BLDC motorok dinamikáját. A szabályozó algoritmus a visszalépéses technikán (backstepping) alapul, és képes a helikoptert stabilizálni realisztikus zavarhatások esetén. A kiterjesztett Kalman-szűrőkre alapozott állapotbecslő egy külső kamera-alapú látórendszer és egy fedélzeti inerciális mérőegység (IMU) mérési eredményei alapján becsüli a nem mért mennyiségeket és csökkenti a mérési hibák hatását. A cikk ezen kívül bemutatja a szabályozó rendszer fejlesztésének környezetét, a valós időben elvégzett tesztek eredményeit, valamint kitér a fejlesztés során szerzett tapasztalatokra.

The objects of the article are the kinematic modeling and control design methods of a quadrotor helicopter. The article focuses on both theoretic and practical aspects. A new low-level control algorithm is introduced. In addition, the supplementary components of the control system are described in detail as well. The helicopter's dynamic model takes into account the aerodynamic friction, the gyroscopic effect of the rotors and also the BLDC motor dynamics. The control algorithm is based on the back stepping approach and is capable of stabilising the model even in case of realistic noises. The state estimator consists of extended Kalman filters and estimates nonmeasured signals and also suppresses measurement noises that are provided by an external camera based vision system and an onboard inertial measurement unit (IMU). Software environment and test methods are presented with attention to the common errors. The results of the first flight tests are also included in the article.

Kulcsszavak: UAV, beltéri négyrotoros helikopter, állapotbecslés, kiterjesztett Kalman-szűrő, hardware-in-the-loop tesztelés ~ UAV, quadrotor helicopter, state estimation, extended Kalman filtering, implementation, hardware-in-the-loop test

Bevezetés

Az utóbbi években a pilótanélküli légi járművek (UAV-k) iránt egyre inkább növekszik az érdeklődés mind a kutatók, mind pedig a járműveket alkalmazni készülőkhöz részéről. Mivel az UAV-knek nem csak katonai, hanem számos civil alkalmazási lehetősége is kínálkozik, egyre jelentősebb fejlesztések folynak ezen a területen. A fejlesztés rövid távú célja, hogy egy olyan kisméretű négyrotoros helikoptert fejlesszünk ki, amely képes önállóan végrehajtani egy komplex missziót, amely egy fel- és leszállási fázisból, valamint egy célterületen tartott megfigyelési, ill. mérésvégzési feladatból áll.

A fejlesztés 2006 tavaszán kezdődött a BME Irányítástechnika és Informatika Tanszék és az MTA SZTAKI Rendszer- és Irányításméleti Kutatólaboratórium együttműködésében. Jelenleg egy olyan beltéri négyrotoros helikopter építésén dolgozunk, amely a későbbiekben alkalmas lesz különböző fejlett irányítási, pályatervezési, ill. képfeldolgozási algoritmusok tesztelésére, valamint pályatervezési, pozícionálási és szenzorfüziós módszerek kipróbálására.

A fejlesztés a következő irányokban indult: irányítási algoritmusok fejlesztése, rendszerterv készítése, elektronikai, mechanikai és aerodinamikai tervezés. A helikopter első prototípusa [1] és [2]-ben jelent meg. 2008-ban sikeres hardware-in-the-loop szimulációkat hajtottunk végre, amelyek igazolták, hogy a fedélzetre szánt MPC555 típusú processzor, amely magas szinten MATLAB Simulinkben programozható, valóban képes lesz a backstepping alapú szabályozást és a kiterjesztett Kalman-szűrőkre épülő állapotbecslést valós időben végrehajtani [3].

A cikk a helikopteren folyó szabályozástechnikai problémákra fókuszál, bemutatja a saját fejlesztésű visszalépéses technikán (backstepping) alapuló szabályozási algoritmust, valamint a fedélzeti irányítás további komponenseit. Ezek között szerepel egy állapotbecslő, amely két kiterjesztett Kalman-szűrőt tartalmaz, valamint egy pályakövető algoritmus is. A rendszer bemutatását követően (1. fejezet) a helikopter és a rotorok modellezése következik (2. fejezet). A 3., 4. és 5. fejezetben a szabályozási rendszer komponensei következnek. A 6. fejezet bemutatja a szimulációs eredményeket, míg a 7. fejezet az algoritmusok megvalósítási kérdéseit tárgyalja. A befejező szakasz röviden összefoglalja az elért eredményeket és a fejlesztés aktuális állapotát.

1. A rendszer felépítése

A helikopter szabályozása pontos pozíció- és orientációmérést igényel. A mérési adatok elsődleges forrása a fedélzetre szerelt Crossbow MNAV100CA inerciális mérőegység (IMU), amely többek között háromdimenziós gyorsulást és szögsebességet mér. A szenzor felépítéséből következően a mérések zajjal és biasszal terhelték. Emiatt integrálás során pontatlanná válnak a pozíció és orientáció becsült értékei. A mérési hibák hatásának csökkentése érdekében kiterjesztett Kalman-szűrőket alkalmazunk az integrálás helyett. Azonban elengedhetetlen, hogy abszolút méréseket is végezzünk. Kültéri járművek esetén megoldás lehet a GPS, ill. A differenciális GPS alkalmazása. Beltéri járművek esetén pedig elterjedt módszer a látórendszerek használata.

A fentiek figyelembevételével a rendszer felépítése az alábbi. A helikopteren egy fedélzeti számítógép van elhelyezve, amely feldolgozza a mérési eredményeket, ill. beavatkozó jeleket számít. A propellerek kefe nélküli egyenáramú (BLDC) motorokra vannak felszerelve. A motorszabályozók és az IMU a központi processzorhoz CAN-buszon keresztül kapcsolódik. Ezen kívül egy szórt spektrumú CDM rádiós csatorna teremt kapcsolatot a helikopter központi egysége és a földi állomás között. A földi állomás küldi a parancsokat és a referenciapályát a helikopternek, valamint ezen a csatornán keresztül érkeznek a pozíció- és orientációmérési eredmények a fedélzeti számítógéphez. A földi állomás felé státuszinformációkat küld a helikopter. A földi állomás hálózaton keresztül csatlakozik a képfeldolgozást végző számítógéphez. A képérzékelő egy kereskedelmi, nagyfelbontású webkamera. További részletek a szenzorrendszerről [4]-ben találhatóak.

2. A négyrotoros helikopter dinamikájának modellezése

2.1. A helikopter mozgásegyenletei

A mozgásegyenletek levezetése során feltételezzük, hogy a Földhöz rögzített K_E koordináta-rendszer inerciarendszernek tekinthető. A helikopter tömegközéppontjához rögzített K_H koordináta-rendszert K_E -hoz képest a $\xi = (x \ y \ z)^T$ pozíciója és RPY szögekben megadott $\eta = (\Phi \ \Theta \ \Psi)^T$ orientációja jellemzi. Az orientációt az R_r mátrix írja le az alábbi módon:

$$R_r = \begin{bmatrix} C_\Psi C_\Theta & C_\Psi S_\Theta S_\Phi - S_\Psi C_\Phi & C_\Psi S_\Theta C_\Phi + S_\Psi S_\Phi \\ S_\Psi C_\Theta & S_\Psi S_\Theta S_\Phi + C_\Psi C_\Phi & S_\Psi S_\Theta C_\Phi - C_\Psi S_\Phi \\ -S_\Theta & C_\Theta S_\Phi & C_\Theta C_\Phi \end{bmatrix}$$

ahol S_x és C_y a $\sin(x)$ és $\cos(y)$ robotikában szokásos jelölése.

A kapcsolat $\dot{\xi}$, $\dot{\eta}$, ill. a sebesség (v) és szögsebesség (ω) között

$$\dot{\xi} = R_r v$$

$$\omega = R_r \dot{\eta}$$

ahol R_r , ill. az inverzének alakja a következő:

$$R_r = \begin{bmatrix} 1 & 0 & -S_\Theta \\ 0 & C_\Phi & C_\Theta S_\Phi \\ 0 & -S_\Phi & C_\Theta C_\Phi \end{bmatrix} \quad \text{és} \quad R_r^{-1} = \begin{bmatrix} 1 & S_\Phi T_\Theta & C_\Phi T_\Theta \\ 0 & C_\Phi & -S_\Phi \\ 0 & S_\Phi / C_\Theta & C_\Phi / C_\Theta \end{bmatrix}$$

ω idő szerint deriváltja

$$\varepsilon = \dot{\omega} = R_r \ddot{\eta} + \dot{R}_r \dot{\eta}$$

Newton törvényeit alkalmazva a rendszerre, a helikopter translációs és rotációs mozgását leíró egyenletek a K_H koordináta-rendszerben az alábbiak:

$$\sum F_{ext} = m\dot{v} + \omega \times (mv)$$

$$\sum T_{ext} = I\dot{\omega} + \omega \times (I\omega)$$

ahol I a helikopter inerciáját jelöli, és feltételezzük, hogy diagonális, azaz $I = \text{diag}(I_x \ I_y \ I_z)$. A két egyenlet bal oldala a helikopterre ható összes külső erőt és forgatónyomatékat jelöli. Ezek a hatások részben a rotorok forgásából adódnak (F és T), részben pedig a gravitáció (F_g) a giroszkopikus hatás (T_g) és a légellenállás (F_a és T_a) eredményeképpen jönnek létre. Összefoglalva

$$\begin{aligned}\sum F_{ext} &= F + F_g + F_a \\ \sum T_{ext} &= T + T_g + T_a\end{aligned}$$

A helikopteren négy beavatkozó szerv található (négy BLDC motor), amely mindegyike felfelé ható erő kifejtésére képes. Ezek a rotorok szögsebességével négyzetesen arányosak ($f_i = b\Omega_i^2$). A BLDC motorok mindegyikéhez külön szabályozó tartozik, amelyek a kívánt szögsebességre pörgetik a rotorokat. A keletkezett nyomatékok és a felhajtóerő az alábbi módon számítható:

$$T = \begin{pmatrix} lb(\Omega_4^2 - \Omega_2^2) \\ lb(\Omega_3^2 - \Omega_1^2) \\ d(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{pmatrix}$$

$$f = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$$

ahol l , b és d a helikopterre, ill. a rotorokra jellemző konstans. A felhajtóerő vektoros formában felírható $F = (0 \ 0 \ f)^T$ módon. A gravitációs erő negatív z -irányban hat, azaz $F_g = -mR_i^T (0 \ 0 \ g)^T = -mR_i^T G$. A giroszkopikus hatás modellezhető a következőképp:

$$T_g = -(\omega \times k)I_r(\Omega_2 + \Omega_4 - \Omega_1 - \Omega_3) = -\omega \times (I_r\Omega_r)$$

ahol I_r a rotor inerciája, k pedig a harmadik standard egységvektor.

Alacsony sebességeken a légellenállás lineáris formulával közelíthető, vagyis $F_a = -K_i v$ és $T_a = -K_r \omega$.

A fenti egyenletekből megkaphatjuk a helikopter mozgásegyenleteit:

$$F = mR_i^T \ddot{\xi} + K_i R_i^T \dot{\xi} + mR_i^T G$$

$$T = IR_r \ddot{\eta} + I \left(\frac{\partial R_r}{\partial \Phi} \dot{\Phi} + \frac{\partial R_r}{\partial \Theta} \dot{\Theta} \right) \dot{\eta} + K_r R_r \dot{\eta} + (R_r \dot{\eta}) \times (IR_r \dot{\eta} + I_r \Omega_r)$$

2.2. Egyszerűsített mozgásegyenletek

Bizonyos ésszerű egyszerűsítésekkel, ill. elhanyagolásokkal jelentősen egyszerűsödnek a mozgásegyenletek. Így a szabályozási algoritmus komplexitása is csökkenthető anélkül, hogy látványosan romlana a minősége.

Mivel a helikopter várhatóan viszonylag lassú mozgásokat végez, nem okoz nagy hibát, ha a légellenállás hatását elhanyagoljuk, mivel K_i és K_r elemei nullaközelieliek. A következő elhanyagolás is az alacsony sebességgel áll kapcsolatban. Alacsony laterális sebességek esetén ugyanis R_r egységmátrixszal közelíthető. Ugyanez a megfontolás az R_i mátrixra nem igaz. A mozgásegyenletek így az alábbi módon közelíthetők:

$$F = mR_i^T \ddot{\xi} + mR_i^T G$$

$$T = I\ddot{\eta} + \dot{\eta} \times (I\dot{\eta} + I_r\Omega_r)$$

Ezek az egyenletek a [3]-ban és [5]-ben található egyenletekre vezetnek.

2.3. Rotordinamika

A négy motor dinamikája az alábbi jól ismert egyenletekkel írható le ($k = 1, \dots, 4$):

$$L\dot{i}_k = u_{m,k} - Ri_k - k_e\Omega_k$$

$$I_r\dot{\Omega}_k = k_m i_k - k_r\Omega_k^2 - k_s$$

ahol k_e , k_m és k_s jelöli a motorok back EMF, motornyomaték és súrlódási állandóját. Ha a motorok induktivitása elhanyagolható, akkor a két egyenlet összevonható:

$$\dot{\Omega}_k = -k_{\Omega,0} - k_{\Omega,1}\Omega_k - k_{\Omega,2}\Omega_k^2 + k_u u_{m,k}$$

3. A helikopter irányítási algoritmus

3.1. Kapcsolódó kutatómunkák

Számos módszer létezik négyrotoros helikopter irányítására. Ezek között található lineáris [6], nemlineáris és soft computing technikára épülő is [7]. A nemlineáris módszerek közül a leggyakrabban tárgyalt a backstepping alapú, bár létezik csúszó szabályozás, ill. feedback linearizáción alapú technika is [8].

A fent említett munkák nem csak a szabályozási algoritmusban különböznek egymástól, hanem a helikopter dinamikai modelljei is eltérnek egymástól. Többen elhanyagolják a rotordinamikát arra hivatkozva, hogy a tranziensek jóval gyorsabbak, mint a helikopteré. Megint mások a légellenállás hatását hanyagolják el az előbbieken említett okok miatt.

A mostani munka legszorosabban [3]-hoz és [9]-hez kapcsolódik szorosan, hiszen a helikopter modellje azonos a [9]-ben leírttal, azonban törekszünk a jobb hibatűrő képességre.

3.2. Backstepping algoritmus négyrotoros helikopterhez

Az algoritmus részben az [5]-ben lártak, részben pedig a saját kutatásunk [3] eredménye. Az itt bemutatott algoritmus ötvözni igyekszik a két módszer előnyeit, azaz hogy a komplex helikoptermodellt képes irányítani, miközben a realisztikus zavarhatásokat is megfelelően kezeli, ahogyan azt a [3] alapján készült korábbi algoritmusunk is tette.

Ehhez először a mozgásegyenletet át kell írni a következő alakra:

$$\ddot{\xi} = f_\xi + g_\xi u_\xi$$

$$\ddot{\eta} = f_\eta + g_\eta u_\eta$$

$$\dot{\Omega}_k = f_{m_k} + g_{m_k} u_{m_k}$$

ahol f_ξ , g_ξ és u_ξ jelentése

$$f_\xi = -G - \frac{1}{m} R_t^T K_t R_t^T \dot{\xi}$$

$$g_\xi = \frac{1}{m} \text{diag}(r_{t,3})$$

$$u_\xi = (f \quad f \quad f)^T$$

Itt $r_{t,3}$ az R_t mátrix harmadik oszlopa, és ezekből építünk fel egy diagonális mátrixot. Ezt azért tehetjük meg, mert az F vektor egyetlen nemnulla elemet tartalmaz. A következő egyenletben f_η , g_η és u_η azonos az alábbiakkal:

$$f_\eta = -(IR_r)^{-1} \left[I \left(\frac{\partial R_r}{\partial \Phi} \dot{\Phi} + \frac{\partial R_r}{\partial \Theta} \dot{\Theta} \right) \dot{\eta} + K_r R_r \dot{\eta} + (R_r \dot{\eta}) \times (IR_r \dot{\eta} + I_r \Omega_r) \right]$$

$$g_\eta = (IR_r)^{-1}$$

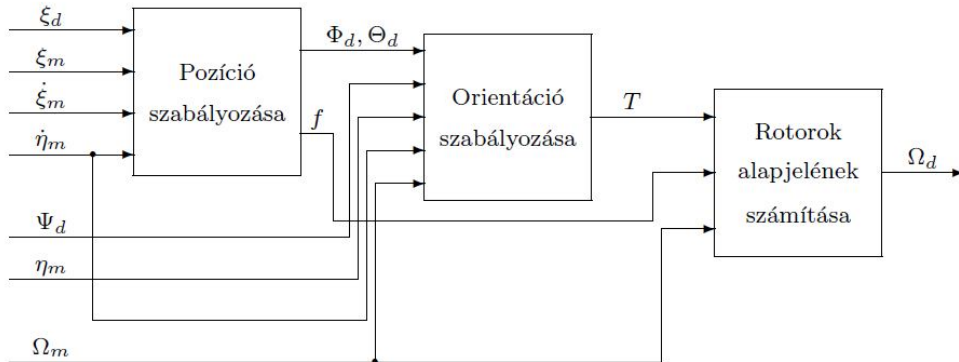
$$u_\eta = T$$

Végül pedig f_{m_k} és g_{m_k} jelöli a következőket:

$$f_{m_k} = -k_{\Omega,0} - k_{\Omega,1} \Omega_k - k_{\Omega,2} \Omega_k^2$$

$$g_{m_k} = k_u$$

Mivel a négyrotoros helikopter alulaktuált rendszer, azaz nagyobb a rendszer szabadságfokainak száma, mint a beavatkozó jelek száma, ezért a szabályozás koncepciója az, hogy a helikoptert egy adott térbeli pontból egy következő pontba juttassa, előírt z-tengelyhez képesti szögben. A referenciajelek tehát az $(x_d \ y_d \ z_d \ \Psi_d)$ négyesből állnak. A szabályozás a helikoptert mindig vízszintes helyzetben stabilizálja. A szabályozás szempontjából a rendszer három alegységre bontható, ahogyan azt az előbbiekben láthattuk. Először a translációs mozgást szabályozzuk, amely megadja a helikopter két hiányzó referenciajelét az orientációt szabályozó alrendszernek. A harmadik alegység felel a motorvezérlők referenciajelének előállításáért. A szabályozó felépítése az 1. ábrán látható. A d és m indexek a kívánt és a mért mennyiségeket jelölik.



1. ábra. A szabályozó felépítése

3.2.1. A pozíció szabályozása

A backstepping algoritmust részletesen itt mutatjuk be. Definiáljuk először a követési hibát: $q_{\xi_1} = \xi_d - \xi$. Ljapunov direkt módszerét alkalmazva definiáljunk egy olyan pozitív definit függvényt, amelynek az idő szerinti deriváltja negatív definit:

$$V(q_{\xi_1}) = \frac{1}{2} q_{\xi_1}^T q_{\xi_1}$$

$$\dot{V}(q_{\xi_1}) = q_{\xi_1}^T \dot{q}_{\xi_1} = q_{\xi_1}^T (\dot{\xi}_d - \dot{\xi})$$

Ha szabadon megválaszthatnák $\dot{\xi}$ -t úgy, hogy $\dot{\xi} = \dot{\xi}_d + A_{\xi_1} q_{\xi_1}$ legyen, akkor

$$\dot{V}(q_{\xi_1}) = -q_{\xi_1}^T A_{\xi_1} q_{\xi_1} < 0$$

negatív definit lenne, ha A_{ξ_1} -et pozitív definitnek választanánk. Ha azonban definiálunk egy virtuális hibát, q_{ξ_2} -t és egy módosított Ljapunov-függvényt a következő módon:

$$q_{\xi_2} = \dot{\xi} - \dot{\xi}_d - A_{\xi_1} q_{\xi_1} = -\dot{q}_{\xi_1} - A_{\xi_1} q_{\xi_1}$$

$$V(q_{\xi_1}, q_{\xi_2}) = \frac{1}{2} (q_{\xi_1}^T q_{\xi_1} + q_{\xi_2}^T q_{\xi_2})$$

akkor ezek segítségével meghatározhatjuk a szükséges felhajtóerőt a módosított Ljapunov-függvény idő szerinti deriváltjából.

$$\begin{aligned} \dot{V}(q_{\xi_1}, q_{\xi_2}) &= q_{\xi_1}^T \dot{q}_{\xi_1} + q_{\xi_2}^T \dot{q}_{\xi_2} = q_{\xi_1}^T (-q_{\xi_2} - A_{\xi_1} q_{\xi_1}) + q_{\xi_2}^T [\dot{\xi} - \dot{\xi}_d - A_{\xi_1} (-q_{\xi_2} - A_{\xi_1} q_{\xi_1})] = \\ &= -q_{\xi_1}^T A_{\xi_1} q_{\xi_1} - q_{\xi_2}^T q_{\xi_2} + q_{\xi_2}^T (f_{\xi} + g_{\xi} u_{\xi}) - q_{\xi_2}^T [\dot{\xi}_d - A_{\xi_1} (q_{\xi_2} + A_{\xi_1} q_{\xi_1})] \end{aligned}$$

Innen u_{ξ} kifejezhető:

$$\begin{aligned} u_{\xi} &= g_{\xi}^{-1} [q_{\xi_1} - f_{\xi} + \dot{\xi}_d - A_{\xi_1} (q_{\xi_2} + A_{\xi_1} q_{\xi_1}) - A_{\xi_2} q_{\xi_2}] = \\ &= g_{\xi}^{-1} [\dot{\xi}_d - f_{\xi} + (I_3 + A_{\xi_2} A_{\xi_1}) q_{\xi_1} + (A_{\xi_2} + A_{\xi_1}) \dot{q}_{\xi_1}] \end{aligned}$$

ahol I_3 egy 3×3 -as egységmátrix. Hasonlóan [5]-höz $\dot{\xi}_d$ -t elhanyagolhatnánk, azonban a szabályozás célja egy térbeli pálya követése lesz folyamatos mozgással, így ennek a tagnak fontos szerepe lesz a későbbiekben. Ha A_{ξ_2} -t is pozitív definitnek választjuk, akkor a Ljapunov-függvény deriváltja valóban negatív definit lesz:

$$\dot{V}(q_{\xi_1}, q_{\xi_2}) = -q_{\xi_1}^T A_{\xi_1} q_{\xi_1} - q_{\xi_2}^T A_{\xi_2} q_{\xi_2} < 0$$

A fenti szabályozójelet alkalmazva a helikopter pozíciójának szabályozására a következő összefüggéshez vezet:

$$\ddot{\xi} = \ddot{\xi}_d + (I_3 + A_{\xi_2} A_{\xi_1}) \dot{q}_{\xi_1} + (A_{\xi_2} + A_{\xi_1}) \dot{q}_{\xi_1}$$

ami ekvivalens a

$$0 = (I_3 + A_{\xi_2} A_{\xi_1}) \dot{q}_{\xi_1} + (A_{\xi_2} + A_{\xi_1}) \dot{q}_{\xi_1} + \ddot{q}_{\xi_1}$$

kifejezéssel. Ha A_{ξ_2} -t és A_{ξ_1} -et diagonálisnak vesszük fel, akkor a karakterisztikus egyenletek

$$s^2 + (a_{\xi_2} a_{\xi_1} + 1)s + (a_{\xi_2} + a_{\xi_1}) = 0$$

alakúak lesznek, ami garantálja a stabilitást. Ez azt jelenti, hogy ha a számított f_{ξ} és g_{ξ} értékek közel vannak a valódihoz, a hibák exponenciálisan 0-hoz tartanak.

Ahhoz, hogy a szükséges felhajtóerőn kívül az orientációs alrendszer két hiányzó referenciajelét meghatározzuk, a $g_{\xi} u_{\xi}$ szorzatban a tagok közül bizonyosakat fel kell cserélni úgy, hogy $g_{\xi} u_{\xi} = \tilde{g}_{\xi} \tilde{u}_{\xi}$ legyen, és \tilde{g}_{ξ} invertálható legyen. Ha tehát \tilde{g}_{ξ} -t és \tilde{u}_{ξ} -t így definiáljuk:

$$\tilde{g}_{\xi} = \frac{1}{m} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & C_{\Phi} C_{\Theta} \end{bmatrix}$$

$$\tilde{u}_{\xi} = \begin{pmatrix} C_{\Psi} S_{\Theta} C_{\Phi} + S_{\Psi} S_{\Phi} \\ S_{\Psi} S_{\Theta} C_{\Phi} - C_{\Psi} S_{\Phi} \\ f \end{pmatrix} = \begin{pmatrix} u_{\xi_x} \\ u_{\xi_y} \\ f \end{pmatrix}$$

akkor a módosított \tilde{u}_{ξ} vektor elemeiből a hiányzó referenciajelek számíthatók

$$S_{\Phi_d} = S_{\Psi} u_{\xi_x} - C_{\Psi} u_{\xi_y}$$

$$S_{\Theta_d} = \frac{C_{\Psi} u_{\xi_x} + S_{\Psi} u_{\xi_y}}{C_{\Phi}}$$

segítségével. Azért tekinthetők ezek referenciajeleknek, mert ha a helikopter a kívánt pozíció felé halad, ezek a mennyiségek 0-hoz tartanak. Megfordítva, ha a helikopter követi a megfelelő irányt, és az adott felhajtóerő hat rá, a célpozícióba fog jutni.

3.2.2. Orientáció szabályozása

A tervezés menete hasonló az előbbi pontban leírtakhoz, így itt csak a főbb lépések szerepelnek. Az orientáció hibáját az előbbihez hasonlóan definiáljuk:

$$q_{\eta_1} = \eta_d - \eta$$

A virtuális követési hiba pedig

$$q_{\eta_2} = \dot{\eta} - \dot{\eta}_d - A_{\eta_1} q_{\eta_1} = -\dot{q}_{\eta_1} - A_{\eta_1} q_{\eta_1}$$

Az előbbi lépéseket követve a végeredmény $T = u_{\eta}$, ahol

$$u_{\eta} = g_{\eta}^{-1} [\ddot{\eta}_d - f_{\eta} + (I_3 + A_{\eta_2} A_{\eta_1}) q_{\eta_1} + (A_{\eta_2} + A_{\eta_1}) \dot{q}_{\eta_1}]$$

3.2.3. A pozíció és orientáció egyszerűsített szabályozása

A 2.2-ben alkalmazott megfontolásokkal élve f_{ξ} -ben, f_{η} -ban és g_{η} -ban egyszerűbb szabályozó algoritmusra jutunk. Az egyszerűsítések a következők:

$$f_{\xi} = -G$$

$$f_{\eta} = -I^{-1} [\dot{\eta} \times (I\dot{\eta} + I_r \Omega_r)]$$

$$g_{\eta} = I^{-1}$$

További lehetőség, hogy a referenciajelek második deriváltját 0-val közelítjük, amely következtében az algoritmus tovább egyszerűsödik.

3.2.4. Rotorszabályozás

Kisebb eltérés az előbbiekhöz képest, hogy a rotor dinamikáját leíró differenciálegyenletben nem második, hanem első idő szerinti derivált szerepel. Ez azt jelenti, hogy nincs szükség q_{m_2}

bevezetésére, és a Ljapunov-függvény $V(q_{m_1}) = \frac{1}{2} q_{m_1}^T q_{m_1}$ -re egyszerűsödik. A hibadinamika szempontjából azonban mégis érdemes q_{m_1} idő szerinti deriváltját is beiktatni. A referenciajel számítása így a következőre módosul:

$$u_m = g_m^{-1} [\dot{\Omega}_d - f_m + (I_4 + A_{m_2} A_{m_1}) q_{m_1} + (A_{m_2} + A_{m_1}) \dot{q}_{m_1}]$$

A motorok számának megfelelően az u_m vektor négy elemből áll.

Érdemes megemlíteni, hogy mivel a felhajtóerő és a nyomatékok a rotorok szögsebességnégyzetének lineáris kombinációi, a referenciajelek az alábbi vektor elemeinek négyzetgyökeiként állnak elő:

$$\begin{pmatrix} \Omega_{1_d}^2 \\ \Omega_{2_d}^2 \\ \Omega_{3_d}^2 \\ \Omega_{4_d}^2 \end{pmatrix} = \begin{bmatrix} 0 & -(2lb)^{-1} & -(4d)^{-1} & (4d)^{-1} \\ -(2lb)^{-1} & 0 & (4d)^{-1} & (4d)^{-1} \\ 0 & (2lb)^{-1} & -(4d)^{-1} & (4d)^{-1} \\ (2lb)^{-1} & 0 & (4d)^{-1} & (4d)^{-1} \end{bmatrix} \begin{pmatrix} T_{3 \times 1} \\ f \end{pmatrix}$$

A Ljapunov-függvény deriváltja pedig

$$\dot{V}(q_{m_1}) = q_{m_1}^T \dot{q}_{m_1} = q_{m_1}^T [(I + A_{m_2} A_{m_1}) \dot{q}_{m_1} + (A_{m_2} + A_{m_1}) \dot{q}_{m_1}] = -q_{m_1}^T (I + A_{m_2} + A_{m_1})^{-1} (I + A_{m_2} A_{m_1}) \dot{q}_{m_1} < 0$$

negatív definit, amennyiben A_{m_2} és A_{m_1} mátrixot pozitív definitnek választjuk.

3.2.5. A szabályozó paramétereinek hangolása

A szabályozó paramétereit alacsony fokszámú karakterisztikus egyenletek, ill. pozitív definit diagonális mátrixok elemeivel állnak kapcsolatban, így megválasztásuk egyszerű. Ráadásul közvetlen a kapcsolatuk a szabályozás gyorsaságával. Figyelembe kell azonban venni, hogy a nagy beavatkozójelek telítést okozhatnak az aktuátorokban.

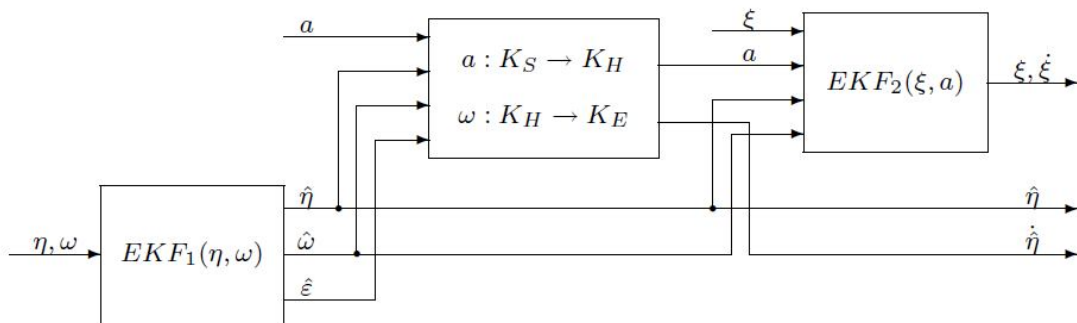
4. Állapotbecslés

Az előbbi fejezetben bemutatott algoritmus az 1. táblázatban látható mennyiségek rendelkezésre állását igényli. A táblázatban *-gal jelölt mennyiségek közvetlenül mérhetők, azonban a pozíció és orientáció mérési gyakorisága nem elegendő. Ráadásul a hosszabb időtartamú mérések azt mutatják, hogy a szenzorok hibái, különösen a bias értéke időben változik. Az említettek miatt tehát szükséges állapotbecslőt alkalmazni a rendszerben.

Mennyiség	Jelentés
ξ	Pozíció*
$\dot{\xi}$	Sebesség
η	Orientáció*
$\dot{\eta}$	Szögsebesség*
Ω	Rotorok szögsebessége*

1. táblázat. A szabályozáshoz szükséges mérendő jelek

Az állapotbecslő két kiterjesztett Kalman-szűrőt tartalmaz, amelyek egymással hierarchikus összeköttetésben állnak. Külön szűrő felelős a pozícióval és az orientációval kapcsolatos jelek becsléséért (EKF₁ és EKF₂). Az állapotbecslő felépítése a 2. ábrán látható. A közepen elhelyezkedő blokk transzformálja a gyorsulás értékét a szenzor koordináta-rendszeréből a helikopterébe.



2. ábra. Az állapotbecslő felépítése

Az állapotbecslés részletes bemutatása előtt érdemes meggondolni, hogy a helikopterre szerelt szenzorokhoz rögzített koordináta-rendszer nem azonos a helikopter koordináta-rendszerével. A két koordináta-rendszer egymáshoz képesti elhelyezkedését az A_s ortonormált mátrix és a p_s vektor írja le.

4.1. A helikopter orientációjának és szögsebességének becslése

A fent említetteknek megfelelően az IMU kimenete három komponensből áll: a valódi értékből és a hozzáadódó véletlen zajból és biasból. A szenzor koordináta-rendszerében kifejezve ez

$$\omega_m = \omega_s = \omega_{s,0} + \omega_{s,b} + \omega_{s,n}$$

alakú, ahonnan a szögsebesség a helikopter koordináta-rendszerében felírva

$$\omega = A_s \omega_{s,0} = A_s (\omega_s - \omega_{s,b} - \omega_{s,n})$$

A Földhöz rögzített inerciarendszerben felírt szögsebesség pedig a

$$\dot{\eta} = R_r^{-1} \omega = R_r^{-1} A_s (\omega_s - \omega_{s,b} - \omega_{s,n})$$

kifejezéssel egyenlő. Mivel a feltételezések szerint a bias lassan változik, az idő szerinti deriváltja 0-hoz közelebb. A változást egy képzeletbeli zavarással lehet modellezni (lásd még [10]-ben):

$$\dot{\omega}_{s,b} = \omega_{s,b,n}$$

További feltételezés, hogy az orientáció mérése biast nem tartalmaz, vagyis

$$\eta_m = \eta + \eta_n$$

Ha a mintavételi időt T_s -sel jelöljük, akkor az Euler-féle diszkrétidejű áttérést alkalmazva a következő differenciaegyenletre jutunk:

$$x_{1,k+1} = x_{1,k} + T_s R_{r,k} A_s (-x_{2,k} + u_k + w_{1,k})$$

$$x_{2,k+1} = x_{2,k} + T_s w_{2,k}$$

$$y_k = x_{1,k} + z_k$$

ahol a jelölések a következők:

$$\begin{array}{lll} x_1 = \eta & x_2 = \omega_{s,b} & x = (x_1 \quad x_2)^T \\ w_1 = -\omega_{s,n} & w_2 = \omega_{s,b,n} & w = (w_1 \quad w_2)^T \\ u = \omega_m & y = \eta_m & z = \eta_n \end{array}$$

A fenti egyenletek a következő általános alakra hozhatók:

$$x_{k+1} = f(x_k, u_k, w_k)$$

$$y_k = g(x_k, z_k)$$

ahonnan a kiterjesztett Kalman-szűrő algoritmus a következő jelölésekkel meghatározható:

$$\begin{array}{ll} R_{w,k-1} = E[w_{k-1} w_{k-1}^T] & R_{z,k} = E[z_k z_k^T] \\ A_{k-1} = \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial x} & B_{w,k-1} = \frac{\partial f(\hat{x}_{k-1}, u_{k-1}, 0)}{\partial w} \\ C_k = \frac{\partial g(\bar{x}_k, 0)}{\partial x} & C_{z,k} = \frac{\partial g(\bar{x}_k, 0)}{\partial z} \end{array}$$

Itt \hat{x}_k a k-adik becsült értéke az állapotváltozóknak, \bar{x}_k az előzetes becslés. A kiterjesztett Kalman-szűrő algoritmus pedig a következő:

1. predikció

$$\bar{x}_k = f(\bar{x}_{k-1}, u_{k-1}, 0)$$

$$M_k = A_{k-1} \Sigma_{k-1} A_{k-1}^T + B_{v,k-1} R_{v,k-1} B_{v,k-1}^T$$

2. mérési adatok frissítése

$$\begin{aligned} S_k &= C_k M_k C_k^T + C_{z,k} R_{z,k} C_{z,k}^T \\ G_k &= M_k C_k^T S_k^{-1} \\ \bar{x}_k &= \bar{x}_k + G_k (y_k - g(\bar{x}_k, 0)) \\ \Sigma_k &= M_k - G_k S_k G_k^T \end{aligned}$$

4.2. A helikopter Pozíciójának és sebességének becslése

A módszer hasonló az előbbihez. Első lépésként a szenzor által mért jel összetevőit kell megállapítani. Az előbbihez képest újdonság egy tag, a gravitáció hatása:

$$a_s = a_{s,0} + a_{s,b} + a_{s,n} - A_s^T R_t^T G$$

Mivel a helikopterhez rögzített koordináta-rendszer nem inerciarendszer, K_H és K_S között a kapcsolatot

$$a_{s,0} = A_s^T (a + \varepsilon \times p_s + \omega \times (\omega \times p_s))$$

írja le. Innen a K_H -beli gyorsulás kifejezhető:

$$a = A_s a_s - \varepsilon \times p_s - \omega \times (\omega \times p_s) + R_t^T G - A_s a_{s,b} - A_s a_{s,n}$$

A fenti kifejezés első része felfogható a gyorsulásmérő kimenetének transzformáltjaként:

$$a_t = A_s a_s - \varepsilon \times p_s - \omega \times (\omega \times p_s) + R_t^T G$$

Ezt a műveletet végzi el a 2. ábrán a középső blokk, és magyarázatot ad a két kiterjesztett Kalman-szűrő hierarchikus elrendezésére.

A mozgó koordináta-rendszerben történő differenciálási szabály ($\ddot{\xi} = \dot{v} + \eta \times v$) alkalmazásával, és az előbbi pontban alkalmazott módszerek segítségével juthatunk az állapotbecslő differenciálegyenletéhez:

$$\begin{aligned} \dot{v} &= -\omega \times v - A_s a_{s,b} + a_t + A_s a_{s,n} \\ \dot{a}_{s,b} &= a_{s,b,n} \\ \dot{\xi} &= R_t v + v_{\xi,n} \\ \xi_m &= \xi + \xi_n \end{aligned}$$

A diszkrétidejű áttérés és a kiterjesztett Kalman-szűrő algoritmusá innen azonos az előbbi pontban leírtakkal.

5. Pályakövetés

5.1. A pályakövetési algoritmus

A tervezés egyik alapvető célja, hogy a helikopter egy térbeli pályán a lehető legkisebb hibával végig tudjon haladni. Ez azt jelenti, hogy egy adott térbeli pontot bizonyos pontossággal meg kell közelíteni mind pozícióban, mind pedig a függőleges tengely körüli előírt orientációban. A követési hibán kívül azonban lényeges szempont, hogy a helikopter mozgása folyamatos legyen. Másképp megfogalmazva a helikopter lehetőség szerint ne lassuljon le látványosan, miközben a célpozíciót megközelíti, hanem haladjon tovább a következő sarokpont felé.

A fent megfogalmazott célok teljesítése érdekében feltételeket fogalmazhatunk meg, amelyek közül akár ha egy is teljesül, a helikopter továbbléphet a következő sarokpont felé.

A pozícióra két feltételt írunk elő:

$$\sum_{j=1}^3 \left(\xi_{d_j}^{(i+1)} - \xi_j \right)^2 < (\Delta \xi_0)^2$$

$$\sum_{j=1}^3 \left(\xi_{d_j}^{(i+1)} - \xi_j \right)^2 < \lambda \sum_{j=1}^3 \left(\xi_{d_j}^{(i+1)} - \xi_{d_j}^{(i)} \right)^2$$

Itt ξ a pillanatnyi pozícióvektor, $\Delta \xi_0$ egy előre definiált konstans térbeli távolság, $\xi_{d_j}^{(i)}$ és $\xi_{d_j}^{(i+1)}$ a követendő pálya két egymást követő sarokpontja. Az első feltétel biztosítja, hogy a helikopter valóban a definiált pálya közelében mozog, míg a második a helikopter folyamatos mozgásban tartásáért felel. A tapasztalat szerint a $\lambda = (3/4)^2$ megfelelő kompromisszum, ha a sarokpontok közel helyezkednek el egymáshoz. Érdeemes megjegyezni, hogy akadályok elkerülésére a módszer korlátozottan alkalmas, azonban az algoritmus nem ezt tűzte ki célul.

Az orientáció követése valamelyest eltér az előbbtől, hiszen bizonyos esetekben fontos lehet, hogy a helikopter milyen orientációban követi a pályát. Emiatt a helikopter orientációjának abszolút hibáját egy korlát ($\Delta \Psi_0$) alá kell szorítani:

$$\left| \Psi_d^{(i+1)} - \Psi \right| < \Delta \Psi_0$$

További lehetséges finomítás, hogy az elfordulás iránya is figyelembe vehető, ha $\Psi_d^{(i+1)}$ -et megfelelően választjuk meg. Egy lehetséges megoldás, ha Ψ értékét nem korlátozzuk a $[0; 2\pi)$ intervallumra.

5.2. A pálya simaságának biztosítása

Ha egymástól távol helyezkedik el két sarokpont, akkor a továbblépéskor keletkező hirtelen megnövekedett hibajelek telítésbe vihetik a motorokat, ill. a szabályozási algoritmusban futásidejő hibák keletkezhetnek, így veszélybe kerülhet a jármű és a környezetének biztonsága.

A referencijelel emiatt érdemes szűrni. A szűrési algoritmust a következő átviteli függvény mintavételezett megvalósítása adja:

$$W(s) = \frac{\gamma |p_s|^{n+1}}{(s + \gamma \cdot p_s)(s + p_s)^n}$$

Érdeemes a paramétereket úgy beállítani, hogy p_s negatív valós, γ pozitív legyen. Ha $n = 5$ és $\gamma = 1/3$, akkor a szabályozójelek simák maradnak, és a szűrő tranzienseit leginkább $\gamma \cdot p_s$ határozza meg.

6. Szimulációs eredmények

Mivel a repülő járművek komplex rendszerek, és hibás működés esetén könnyen kárt tehetnek önmagukban és a környezetükben, a leendő irányítórendszert érdemes alaposan letesztelni. Másrészt ellenőrizni kell, hogy a célhardver képes-e megfelelően ellátni feladatát valós időben. Az említettek miatt hardware-in-the-loop szimulációban teszteltük az irányítórendszert.

A helikopter és a BLDC motorok paramétereit a tervezett méretek alapján határoztuk meg. Az értékeket a 2. táblázat foglalja össze.

Paraméter	Érték
l	0,23 m
b	$1,759 \cdot 10^{-5}$ kg m
d	$1,759 \cdot 10^{-6}$ kg m ²
I_x, I_y	$1,32 \cdot 10^{-2}$ kg m ²
I_z	$2,33 \cdot 10^{-2}$ kg m ²
I_r	$5,626 \cdot 10^{-5}$ kg m ²
m	1,4 kg
K_t	diag([0.1 0.1 0.15]) Ns/m
K_r	diag([0.1 0.1 0.15]) Nm s
$k_{\Omega,0}$	94,37 1/s ²
$k_{\Omega,1}$	3,02 1/s
$k_{\Omega,2}$	0,005
k_u	139,44 V/s ²

2. táblázat. A helikopterre jellemző fizikai paraméterek

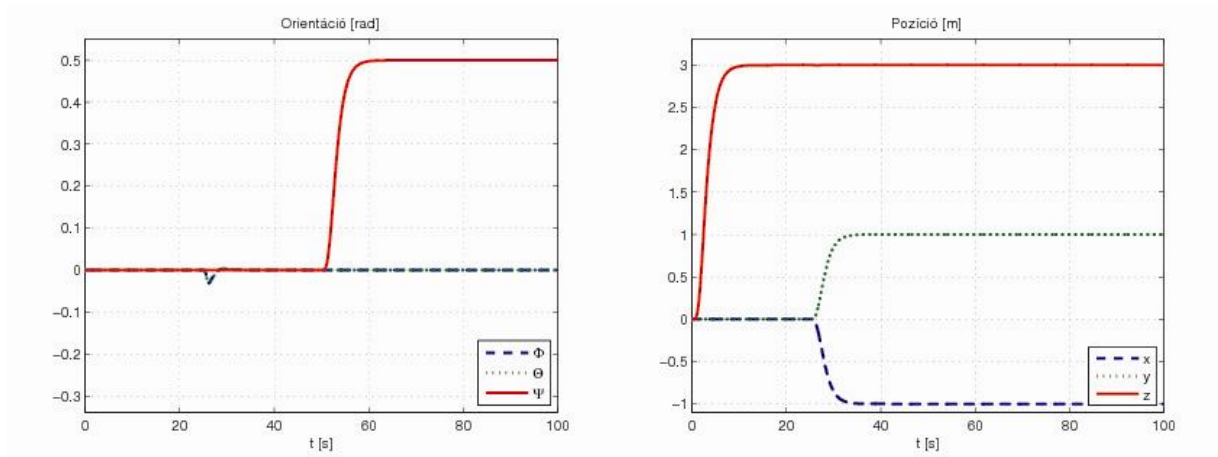
6.1. Simulinkben végrehajtott tesztek

A szabályozás komponenseit fokozatosan beillesztve a rendszerbe Simulinkben teszteltük az algoritmust. A legegyszerűbb esetben mindössze a helikopter modelljét és a szabályozót tartalmazta az összeállítás. A méréseket zajmentesnek tekintettük, és a feladat egyetlen térbeli pont elérése volt megadott orientációban. A referenciajelek az 5. fejezetben leírtaknak megfelelően voltak simítva. A tesztelés mellett a szabályozó paramétereinek behangolása volt az elsődleges cél ekkor (3. táblázat). A mintavételi időt $T_s = 10ms$ értékre állítottuk az IMU legnagyobb működési sebességével összhangban.

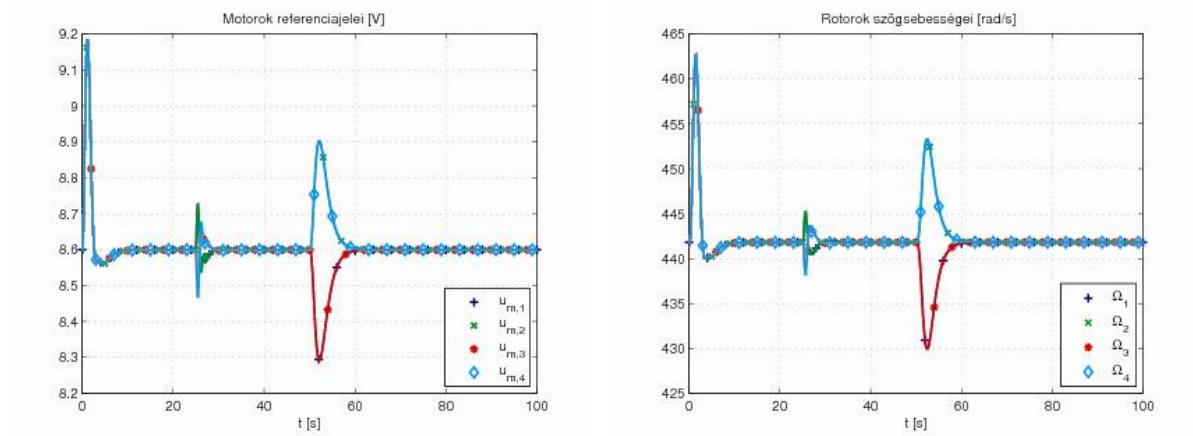
Paraméter	Érték
A_{ξ_1}	diag(2 1.6 1.6)
A_{ξ_2}	diag(1.2 1.2 1.2)
A_{η_1}	diag(12 12 12)
A_{η_2}	diag(8 8 8)
A_{m_1}	diag(0.04 0.04 0.04 0.04)
A_{m_2}	diag(0.016 0.016 0.016 0.016)

3. táblázat. A szabályozáshoz szükséges mérendő jelek

A 3. és 4. ábrán egy egyszerű szimuláció eredményei láthatók, sorrendben a helikopter pozíciója és orientációja, valamint a motorok referenciajelei és a rotorok szögsebességei.

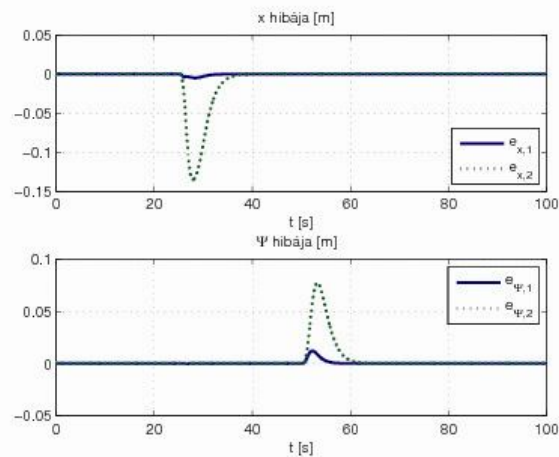


3. ábra. A helikopter pozíciója és orientációja repülés közben



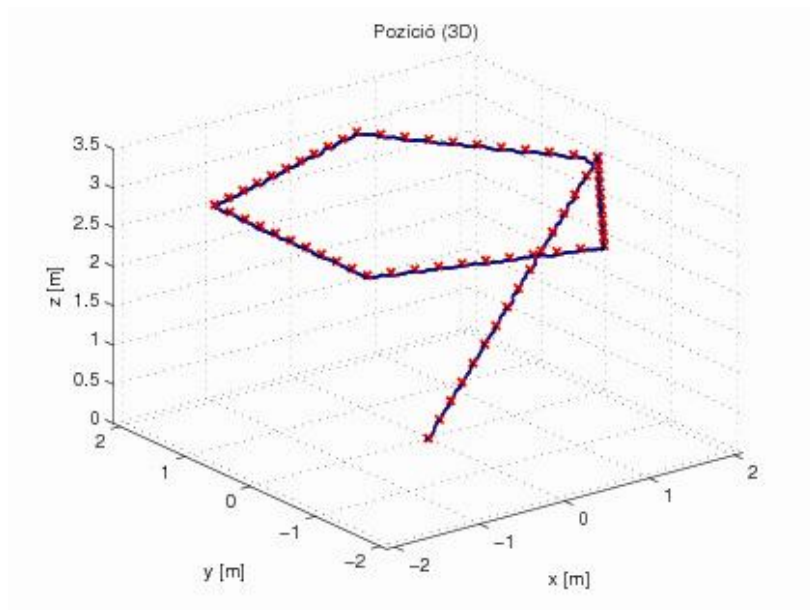
4. ábra. A motorok referenciajelei és szögsebességei

Az eredményeket összehasonlítottuk az egyszerűsített algoritmussal kapott eredményekkel, és a tapasztalat szerint az egyszerűbb változat csupán kevéssel teljesített rosszabbul. Ez igazolja, hogy a helikopter a repülés során a feltételezéseknek megfelelően viselkedett (alacsony sebesség, kis szögelfordulások). Összehasonlításként az 5. ábrán a pozíció x-tengely irányú komponensének hibatranziense látható, ahol a pontozott vonal az egyszerűsített algoritmushoz tartozik.

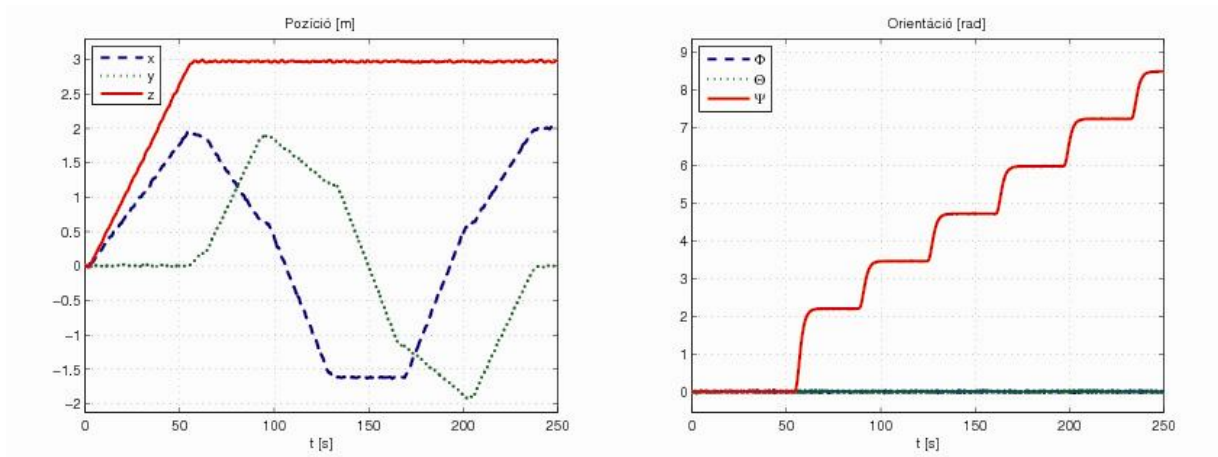


5. ábra. A hibatranziensek a komplex és az egyszerűsített szabályozó esetén (folytonos vonal – komplex, szaggatott vonal - egyszerűsített)

Végül az összes komponens rendszerbe integrálásával a szabályozás egy komplex pálya követésére is alkalmassá válik. A 6. és 7. ábrán egy ilyen feladat végrehajtása látható. A kiterjesztett Kalman-szűrők kezdeti beállításait a 4. táblázat tartalmazza. A beállításokat a valós szenzorok tanulmányozása alapján állítottuk be (bővebb irodalom erről [11]-ben található).



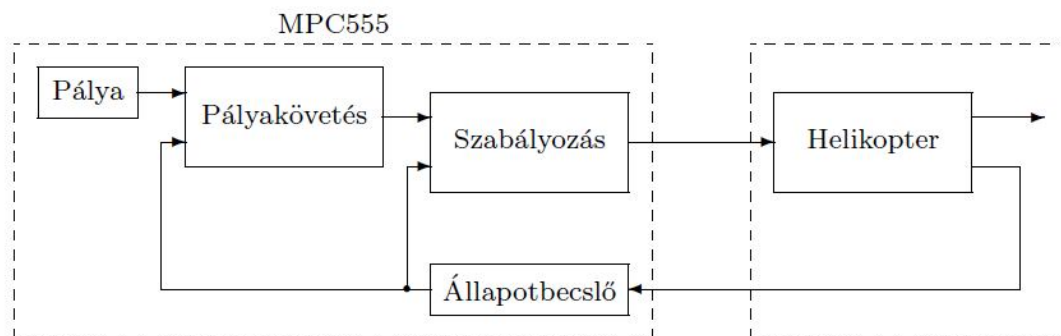
6. ábra. Komplex pályakövetési feladat



7. ábra. A helikopter pozíciója és orientációja komplex pályakövetés esetén

6.2. Valós idejű tesztek

A valós idejű tesztek egy dSPACE DS1103 egység segítségével hajtottuk végre hardware-in-the-loop technikával. A helikopter és a szenzorrendszer működését ezen az egységen emuláltuk, míg a szabályozási feladatokat az MPC555 mikrokontroller látta el, amely a végső összeállításban a fedélzeti elektronika központi egysége. A kommunikáció a két egység között CAN-buszon történik ugyanúgy, ahogy a valódi helikopteren is. A 8. ábrán látható a valós idejű tesztek megvalósításának sematikus vázlata.



8. ábra. A hardware-in-the-loop tesztek sematikus vázlata

6.3. A valódi helikopteren végzett kísérletek

A teljes rendszer megvalósítása során egyszerűbb repülési módokban tesztrepüléseket végzünk. A térbeli pozíció- és orientációmérés működésbe lépése előtt korlátozott ideig a helikopterre szerelt szenzorok segítségével tudjuk becsülni a jármű orientációját. A gondos kalibrációnak köszönhetően két perc alatt mindössze $5\text{-}10^\circ$ -os hibát okoz az IMU jeleinek integrálása.

Ezt alapul véve tesztekét végeztünk az orientáció stabilizálhatóságára. A szabályozás alapja LQ állapot-visszacsatolás, amelyet munkapont környezetében linearizált rendszeren valósítottunk meg. A linearizálás a helikopter vízszintes helyzetben lebegése körül történt. Az állapotegyenlet alakja:

$$A = \frac{\partial f(x,u)}{\partial x} \Big|_{x_0} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

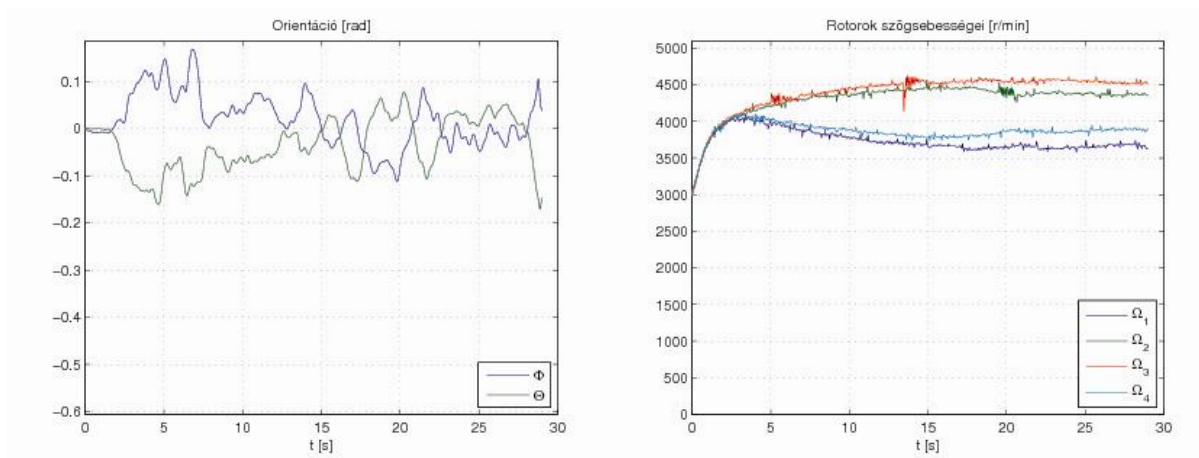
$$B = \frac{\partial f(x,u)}{\partial u} = \begin{bmatrix} 0 & -lb/I_x & 0 & lb/I_x \\ 0 & 0 & 0 & 0 \\ -lb/I_y & 0 & lb/I_y & 0 \\ 0 & 0 & 0 & 0 \\ -d/K_z & d/K_z & -d/K_z & d/K_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \frac{\partial g(x,u)}{\partial x} \Big|_{x_0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \frac{\partial g(x,u)}{\partial u} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

A súlyozómátrix $Q = \text{diag}(1 \ 100 \ 1 \ 100 \ 1 \ 100)$ értékre lett beállítva, a visszacsatolást pedig a fenti rendszer diszkrét idejű állapotegyenletéhez terveztük.

A motorok közti eltérések kiküszöbölésére az LQ visszacsatolást kiegészítettük egy integráló szabályozással, amely a helikopter stabilitására pozitív hatással volt. A következő ábrán egy tesztrepülés eredményei láthatók, ahol a szabályozás a helikoptert vízszintesen stabilizálta. Biztonsági okokból a tesztelés során a teljes felhajtóerő a helikoptert nem volt képes a levegőben tartani, azonban csekély külső beavatkozásra volt csupán szükség.



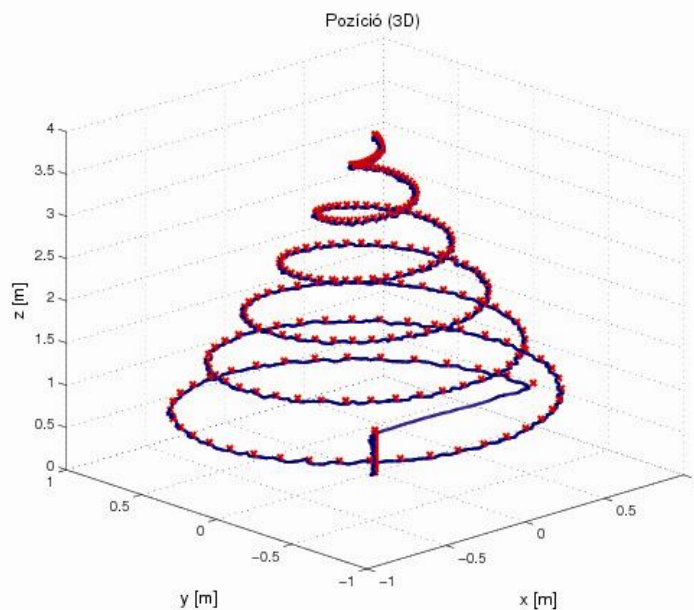
9. ábra. A helikopter orientációja és a rotorok fordulatszáma a tesztrepülés közben

7. A szabályozási algoritmus megvalósítása a célprocesszoron

7.1. Szoftverkörnyezet

Az irányítási algoritmus egy Freescale MPC555 típusú processzoron fut, amely egy Phytec által gyártott alaplapon van elhelyezve. A processzor egy 40Mhz órajelű PowerPC architektúrájú mikrokontroller, amely egy 64 bites lebegőpontos FPU-t is tartalmaz. A szabályozási algoritmust MATLAB/Simulink segítségével terveztük. A hardver támogatottságának köszönhetően automatikusan lehet futtatható kódot tölteni a processzorra. A fordítási és letöltési procedúra a Real-Time Workshop, a Real-Time Workshop Embedded Coder, Embedded Target for Motorola MPC555, valamint a MetroWerks CodeWarrior segítségével történik.

A hardware-in-the-loop szimulációkat a már említett dSPACE DS1103 egység segíti. Az egységet leginkább az autóiparban alkalmazzák gyors prototípustervezésre. Ehhez is tartozik Simulink blokk-könyvtár, azaz a futtatható kódot ugyancsak MATLAB/Simulink segítségével állítjuk elő. A szimulációk során az adatok monitorozását és mentését egy grafikus program segítségével, a ControlDeskkel végezhetjük. A gyűjtött adatokat később MATLAB segítségével tudjuk feldolgozni. A 10. ábrán egy ilyen valós idejű teszt eredményeit láthatjuk.



10. ábra. Komplex pályakövetési feladat szimulációja valós időben

7.2. Kommunikáció

A fejlesztés kezdeti szakaszában bizonyos egységekkel, mint az IMU-val, soros vonalon terveztük a kommunikációt. A tapasztalat azonban azt mutatta, hogy a Simulinkből fordított kód nem volt megfelelő, hiszen jelentős (20ms – 30ms) késleltetést vitt a rendszerbe. Ezért amellet döntöttünk, hogy az MPC555 processzorral minden egyéb egység CAN-buszon kommunikál, és kiegészítő egységek végzik el a konverziót. Gyakorlati okokból azonban ennek következményeként bizonyos jeleket csonkítani kell.

További lényeges szempont, hogy a CAN-buszon az MPC555-nek címzett csomagokat a lehető leghamarabb érdemes feldolgozni, hogy a motorok referenciajeleit a lehető leghamarabb számolni lehessen. Emiatt a kommunikáció mintavételi ideje lényegesen kisebb a szabályozás mintavételi idejénél.

7.3. A fejlesztés során felmerülő szoftverrel kapcsolatos problémák

Több kisebb-nagyobb jelentőségű probléma nehezítette a fejlesztést. A legfontosabb ezek közül a mintavételi idővel volt kapcsolatos. A célprocesszor ugyanis nem képes a teljes szabályozási algoritmust az eredetileg tervezett 10ms-os mintavételi idővel végrehajtani a hardverkorlátok miatt. A mintavételi időt így 30ms értékre állítottuk, azonban ez várhatóan nem okoz stabilitási problémákat.

A könnyen kiküszöbölhető hibák közé tartozik, hogy ha a fordítandó blokkésma ún. beágyazott S-függvényeket tartalmaz C nyelven megírva, akkor ha ezek tartalmaznak globális változókat, akkor azok fájlanként eltérő nevéeknek kell lenniük, ellenkező

esetben a program futása megjósolhatatlan, mivel a fordítás során az egyik felülírja a másikat.

Egy következő problémával akkor szembesülhet a fejlesztő, ha bizonyos blokkok egymástól eltérő mintavételi idővel működnek. Diszkrét deriválást végző blokkok alkalmazása esetén kritikus hibák fordulhatnak elő, ugyanis ezeknek nem adható meg külön, milyen mintavételi idővel működjenek, vagyis a használatuk kerülendő.

8. Az eredmények összefoglalása

A cikkben egy autonóm működésű négyrotoros helikopter dinamikai modellezését és fedélzeti irányítási algoritmusának kifejlesztését mutattuk be. Részletesen bemutattuk a szabályozás komponenseit, amely egy saját fejlesztésű backstepping elvű szabályozásra épül. Kiegészítő komponensként kiterjesztett Kalman-szűrőkből álló állapotbecslőt tartalmaz, amely a helikopterre szerelt inerciális mérőegység és a BME IIT Tanszéken kifejlesztett külső kameraalapú pozícionálórendszer méréseit dolgozza fel.

A szabályozási algoritmus Freescale MPC555 alapú rendszeren valósítottuk meg, és dSPACE DS1103 segítségével hardware-in-the-loop szimulációkat hajtottunk végre. A helikoptert és a szenzorrendszert ez utóbbi egységen emuláltuk.

A szabályozási rendszer komponensei között CAN-busz, ill. rádiófrekvenciás kapcsolat biztosítja az összeköttetést.

A pályakövetés sikeres végrehajtása mutatja a szabályozási algoritmus hatékonyságát.

Köszönetnyilvánítás

A cikkben bemutatott munka háttérét az Országos Tudományos Kutatási Alapprogramok OTKA K 71762 jelű támogatása biztosítja.

Irodalomjegyzék

- [1] A. Soumelidis, P. Gáspár, P. Bauer, B. Lantos, and Z. Prohászka: Design of an Embedded Microcomputer Based Mini Quadrotor UAV, Proceeding of the European Control Conference ECC'07, pp. 2236–2241, 2007.
- [2] A. Soumelidis, P. Gáspár, G. Regula, and B. Lantos, Control of an Experimental Mini Quad-Rotor UAV, Proceeding of the 16th Mediterranean Conference on Control and Automation, pp. 1252–1257, 2008.
- [3] L. Kis, G. Regula, and B. Lantos, Design and Hardware-in-the-Loop Test of the Embedded Control System of an Indoor Quadrotor Helicopter, Proceeding of the

- 6th Workshop on Intelligent Solutions in Embedded Systems WISES'08, pp. 35–44, 2008.
- [4] L. Kis and B. Lantos, Szenzorfüzió alkalmazása beltéri autonóm négyrotoros helikopteren, Hadmérnök, 2009. december.
 - [5] S. Bouabdallah and R. Siegwart, Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor, Proceeding of the IEEE International Conference on Robotics and Automation, pp. 2247–2252, 2005.
 - [6] S. Bouabdallah, A. Noth, and R. Siegwart, PID vs LQ control techniques applied to an indoor micro quadrotor, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems IROS'04, vol. 3, 2004.
 - [7] C. Coza and C. Macnab, A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization, Proceeding of the Annual meeting of the North American Fuzzy Information Processing Society NAFIPS'06, pp. 454–458, 2006.
 - [8] A. Das, K. Subbarao, and F. Lewis, Dynamic Inversion of Quadrotor with Zero-Dynamics Stabilization, Proceeding of the 17th IEEE Conference on Control Applications, pp. 1189–1194, 2008.
 - [9] T. Madani and A. Benallegue, Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique, Proceedings of the 45th IEEE Conference on Decision and Control, pp. 1515–1520, 2006.
 - [10] B. Lantos, State Estimation Based on Image Processing and Inertial Sensors for Indoor Autonomous Systems, BME IIT, Research Report (in Hungarian), 2007.
 - [11] L. Kis, Z. Prohászka, and G. Regula, Implementation and Theoretical Improvement of the Control System of an Autonomous Indoor Quadrotor Helicopter, Proceeding of the 17th International Workshop on Robotics in Alpine-Adria-Danube Region RAAD'08, 2008.