

Pozíció meghatározása inerciális szenzorokkal

Position determination with inertial sensors

NAGY Roland¹ (MSc), SZALAI István² (DSc), FÜLÖP Tamás³

Pannon Egyetem, Mérnöki Kar, Mechatronikai Képzési és Kutatási Intézet
H-8900 Zalaegerszeg, Gasparich Márk u. 18/A. C. ép., www.zek.uni-pannon.hu

¹ nagy.roland@mk.uni-pannon.hu; ² szalai@almos.uni-pannon.hu, ³ fulop.tamas@mk.uni-pannon.hu

Abstract

We present the development and testing of an algorithm for relative positioning based on inertial (IMU) sensor data. After a review of possible solutions, the architecture of the implemented algorithm is presented. The program code is developed in Matlab environment, which includes sensor calibration, sensor fusion methods and filtering and correction steps. Testing of the algorithms and verification of the measurement accuracy was performed on a servo motor driven linear unit.

Keywords: inertial sensor, positioning, sensor fusion, data acquisition, AHRS algorithm

Kivonat

Közleményünk egy inerciális (IMU) szenzor adatai alapján relatív pozíció meghatározására képes algoritmus fejlesztését és tesztelését mutatja be. A lehetséges megoldások áttekintése után bemutatjuk a megvalósított algoritmus felépítését. A programkód Matlab környezetben került kialakításra, mely tartalmazza a szenzor kalibrációs, szenzorfüziós módszereket és szűrési, pontosítási lépéseket. Az algoritmusok tesztelését és a mérési pontosság ellenőrzését egy szervomotor hajtású lineáris egységen végeztük.

Kulcsszavak: inerciális szenzor, pozíció meghatározás, szenzorfüzió, adatgyűjtés, AHRS algoritmus

1. BEVEZETÉS

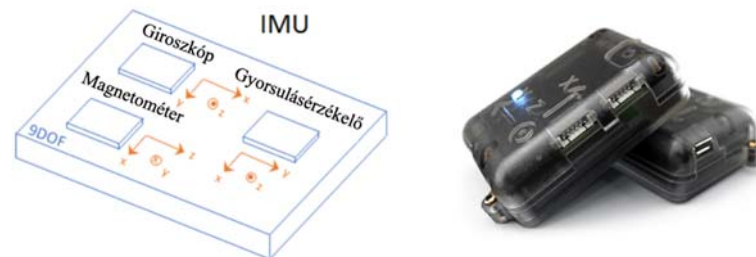
A pozíció meghatározására a mindennapok során számos helyen szükség van. Ez történhet globális helymeghatározó rendszer (GPS) vagy egyéb környezeti rendszerek, mint például kamera vagy LIDAR szenzorok segítségével. Ezek kiemelkedő pontosságú mérést tesznek lehetővé, az alkalmazási terület jellegétől függően lehetőség van akár centiméteres pontosság elérésére is. [1][2] Ezek hátránya lehet azonban a környezeti hatásoktól való nagy mértékű függés, így az eső, a hó, a fényviszonyok változása vagy az esetleges szennyeződések erősen zavarhatják a szenzorok működését. GPS helymeghatározás esetén a pontosság korlátai vagy a beltéri alkalmazások jelenthetnek akadályt.

Az inerciális szenzorok ezen esetekben megfelelő alternatívát jelenthetnek, mivel működésük a környezeti hatásoktól és egyéb külső eszközöktől független. Egy test relatív pozíciója meghatározható annak egyes tengelyek menti gyorsulásai és szögsebességei alapján, mely számítási módszert összefoglalóan inerciális navigációnak neveznek. Az ehhez szükséges szenzorokat, mint 3 tengelyű gyorsulásmérőt, 3 tengelyű giroszkópot és gyakran magnetométert egy tokozásban tartalmazza az inerciális mérőegység (IMU egység). Ezen mérési módszert rendszerint kiegészítő vagy biztonsági eszközként alkalmazzák például autópárhuzamban vagy beltéri rendszerekben, ahol cél egy általános mozgásprofil detektálása, ugyanakkor a mérési pontosság másodlagos.

Ezen egységek jellemzően MEMS típusú szenzorokat tartalmaznak, melyek azonban felépítésükből adódóan hajlamosak a driftre, kimenetüket továbbá zaj terhelheti. A mérés pontosítása egyedileg fejlesztett, különböző adatfeldolgozási és szűrési módszerekkel lehetséges. Munkánk célja ehhez kapcsolódóan egy algoritmus fejlesztése és pontosságának ellenőrzése, amellyel inerciális szenzorok fúziója által számítható a relatív pozíció. Közleményünkben elsőként áttekintjük az alkalmazott módszereket, majd bemutatjuk a fejlesztés lépéseit és a mérési eredményeket.

2. POZÍCIÓ MEGHATÁROZÁSA INERCIÁLIS SZENZORRAL

Munkánk során az X-IO által gyártott, X-IMU típusú szenzoregységet használtuk. Az egység kis méretű, kompakt kialakítású, rendelkezik giroszkóp, gyorsulásmérő, magnetométer és hőmérsékletmérő szenzorokkal. A giroszkóp $2000^\circ/s$, a gyorsulásmérő 8 g méréshatárral rendelkezik, míg mintavételi frekvenciájuk 512 Hz, felbontásuk 16-bit. A magnetométer méréshatára 4 Gauss, mintavételezési frekvenciája 160 Hz, felbontása 12-bit. A hőmérsékletérzékelő alkalmazható az előbbi szenzorok kalibrációjához és korrekciójához, így a drift mértéke csökkenthető. Az IMU egység így összesen 9 tengely mentén képes detektálni a mozgásokat. A nyers mérési adatok SD kártyára kerülnek mentésre, további előny, hogy az egység beépített akkumulátorral rendelkezik, így akár 18 óráig is üzemeltethető külső tápellátás nélkül. Egy általános IMU egységet és az X-IMU szenzor képét az 1. ábra mutatja.



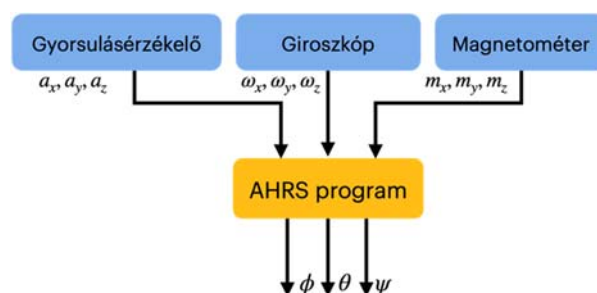
1. ábra. Egy általános IMU egység felépítése és az X-IMU típusú szenzor

Az inerciális navigáció során egy objektum pozíciójának meghatározása alapvetően 3 tengely menti gyorsulás és 3 tengely körüli szögsebesség adatokból történik. A gyorsulások kettős integrálásával számítható az elmozdulás, azonban általános mozgás során a test orientációja folyamatosan változik, az ezekből adódó szögváltozásokat tehát szükséges a gyorsulásadatok integrálása előtt eltávolítani. A fejezet további részében ennek lépéseit mutatom be.

Az algoritmus három fő részre osztható

- Első lépés a relatív vagy abszolút orientáció meghatározása az úgynevezett Attitude and Heading Reference Systems (AHRS) algoritmus segítségével. A megnevezés összefoglalja a szenzorfüzióhoz alkalmazott programkódokat, mely a gyorsulás és szögsebesség adatokból képes a relatív orientáció meghatározására. Amennyiben rendelkezésre áll magnetométer adat, úgy a valódi északi irány számítható, tehát az abszolút orientáció is meghatározható.
- A második lépés a számított orientációk ismeretében a szögelfordulások kompenzálása, vagyis a mért gyorsulásértékek transzformációja az IMU koordinátarendszeréből a globális koordinátarendszerbe. A transzformáció történhet forgatómátrixok vagy kvaterniók segítségével. Ez utóbbi az elterjedtebb, mivel számításigénye alacsonyabb és eredménye megbízhatóbb a forgatómátrix számítási módszernél.
- Az utolsó lépés során a transzformált gyorsulásadatok kétszeres numerikus integrálása segítségével kaphatjuk meg a globális koordinátarendszerben az X, Y és Z tengely menti elmozdulást.

A fenti lépéseket minden adatsorra végrehajtva számítható a teljes mozgáspálya. A leírt folyamatot a 2. ábra szemlélteti.



2. ábra. Orientáció meghatározása szenzorfüzió segítségével

A gyakorlati alkalmazásoknál számos, különböző felépítésű AHRS programkód terjedt el. A legelterjedtebb algoritmusok a komplementer, a Mahony vagy a Kálmán szűrő.

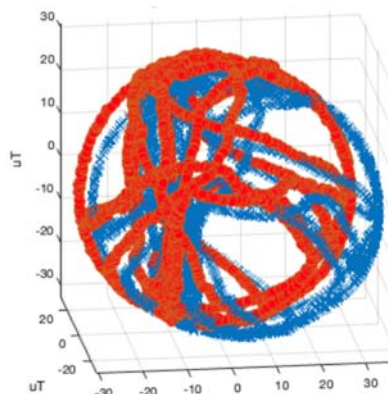
A komplementer szűrő egy összekapcsolt alul- és felüláteresztő szűrőből áll. A giroszkóp pontossága rövid távon jó, viszonyt hosszú távon drift jelentkezik, míg a gyorsulásérzékelők esetén a statikus pontosság kiemelkedő. A komplementer szűrő így a szenzorok hátrányos tulajdonságait megszüntetve biztosítja a szenzorfüziót. A szűrő karakterisztikája a szenzorok súlyozásával állítható be. Az algoritmus előnye az egyszerű felépítés, a kis számításigényből adódóan szinte bárhol alkalmazható, továbbá a relatíve jó pontosság miatt az egyik legelterjedtebben alkalmazott AHRS algoritmus. Kiterjesztett Kálmán szűrő alkalmazásánál szükség van a rendszer modelljére, ez alapján pontosabb becslés adható, nagyobb hatékonysággal szűrhető ki a zaj, azonban telepítése és beállítása nagyságrendekkel bonyolultabb, számításigénye magas, ezért viszonylag ritkán használt AHRS algoritmusként. A Mahony szűrő alapelve a Kálmán és komplementer szűrő összeillesztéséből adódik, az algoritmus megfigyelőt tartalmaz, alkalmas hardveres megvalósításra, a becslés során a zajok kompenzálásra kerülnek, pontossága ugyanakkor a komplementer és Kálmán szűrő között helyezkedik el. [3]

A pozíció számítás pontosságát a választott algoritmus nagymértékben meghatározza, azonban ezzel együtt az optimalizálás időszükséglete is nagyságrendekkel növekedhet. Ezen okból a legtöbb felhasználási módnál elegendő lehet a kisebb számításigényű és gyorsabban paraméterezhető szűrők alkalmazása, mint például a komplementer algoritmus. Az orientáció számításánál meghatározó a zaj és a drift mértéke, míg a numerikus integrálásból adódó pontatlanságok hatása hosszabb időtartam után jelentkezik. Mivel az IMU egységgel végzett mérések jellemzően dinamikus jellegűek, így utóbbi hatása a legtöbb esetben elhanyagolható. Ebből következően kiemelt jelentőségűek az alkalmazott zaj és drift szűrési módszerek hatékonysága. Továbbá ebből adódóan az AHRS algoritmust és az előzetes adatfeldolgozási lépéseket minden esetben az adott alkalmazási területhez szükséges illeszteni és optimalizálni. [4]

3. ALKALMAZOTT ALGORITMUSOK

Jelen fejezetben röviden áttekintjük a pozíció meghatározásához létrehozott algoritmusokat. Az orientáció meghatározásához a komplementer szűrőt választottuk a korábban bemutatott előnyei, rugalmassága folytán. A fejlesztést Matlab környezetben végeztük. A létrehozott programkód tartalmazza az előzetes kalibrációs és adatszűrési lépéseket, majd az AHRS programrész segítségével meghatározhatók az egyes elmozdulások.

A pozíció meghatározása előtt elsőként szükséges a magnetométer kalibrációját elvégezni. A magnetométer az X, Y és Z tengely mentén méri a mágneses tér erősségét, ennek pontossága meghatározza az abszolút orientáció becslés pontosságát. Az IMU egységekben általánosan alkalmazott MEMS szenzorok érzékenyek lehetnek a gyártási hibákra és környezeti hatásokra egyaránt. Egy ideális magnetométer által mért adatokat 3 dimenzióban ábrázolva az értékek egy gömb felületén oszlanak el. A gyártási hibák gyakran állandó mágneses effektusként viselkednek, így az ideális gömb középpontját eltolhatják az origóból, míg a környezeti zajok az ideális gömb formát ellipszoiddá torzíthatják. A korrekció során iterációs lépésekkel szükséges előállítani egy olyan 3x3-as mátrixot és egy 1x3-as vektort, melyekkel az eredeti értékeket szorozva az ábrázolt adatpontok átkerülnek az origóba és az ideális gömb felületére. A fejlesztés során a korrekció előtti és utáni adatpontokat a 3. ábra mutatja kék és piros színnel.



3. ábra. X-IMU magnetométer adatai ábrázolva 3 dimenzióban, kék színnel a még nem kalibrált, piros színnel a kalibrált értékek

A magnetométer korrekciója után a komplementer szűrő következik a programkódban. Az algoritmus a következő iterációs lépésekből épül fel:

- θ_{acc} szög számítása a gyorsulásértékek alapján
- Az abszolút orientáció számítása az alábbi egyenlet alapján:

$$\theta_{absz.szög} = \alpha * (\theta_{absz.szög} + \omega_{gyro} * \Delta t) + (1 - \alpha) * \theta_{acc}$$

ahol α a súlyozási tényező, mely jelen esetben $\alpha = 0,98$, ω_{gyro} a giroszkóp egyes tengelyek körül mért szögsebességei, Δt a mintavételek között eltelt idő. Ennek értelmében a szögsebességből téglalap szabály szerinti numerikus integrálással történik a szögelfordulás számítása.

- Az aktuális orientáció átszámítása kvaternió egységbe
- Az eredeti gyorsulásértékek transzformálása a globális koordináta-rendszerbe kvaternió forgatással

Az algoritmus kimenete a globális koordináta-rendszer X, Y és Z tengely menti gyorsulásértékei, melyeket szükséges az egyes adathibáktól tisztítani a kettős integrálás elvégzése előtt. Elsőként a függőleges, Z tengely mentén mért gravitációs gyorsulás értékét vonjuk ki, mely az integrálás során nagy mértékű, lineárisan növekvő hibát eredményezne. Ezek után az átlagérték kivonásával és lineáris regressziós módszerrel korrigáljuk a nullpont és trendszerű hibákat.

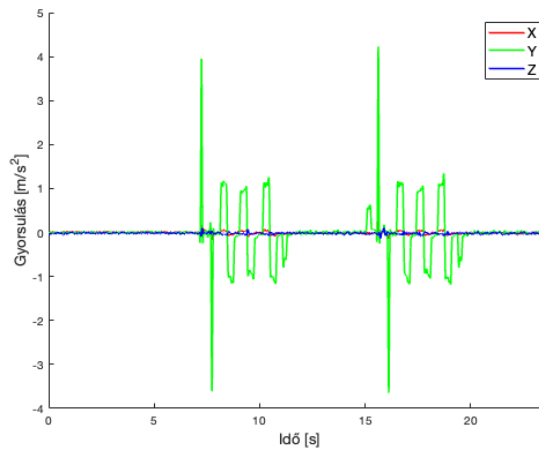
A MEMS szenzorból adódó pontatlanságok és a környezeti hatások következtében a mért jelet folyamatosan változó, kis amplitúdójú zaj terheli, melynek hatása a kettős integrálás következtében nagyságrendekkel növekedne. Ennek érdekében a zaj szűréséhez több módszert alkalmazunk. A hasznos jelek a mérések során $0,5 - 1,5 \text{ m/s}^2$ tartományba esnek, így a $0,1 \text{ m/s}^2$ -nél kisebb értékeket véletlen zajnak tekintjük, ezen jeleket a mérési adatokból eltávolítjuk. Továbbá mivel a lineáris egység viszonylag lassan változó és állandó sebességű mozgásokat biztosít, így az esetleges nagyobb amplitúdójú zajok szűrésére egy 1 Hz levágási frekvenciájú digitális aluláteresztő szűrőt helyeztünk el az algoritmusban. A programkódba építettünk továbbá egy, a statikus helyzetek detektálására alkalmas kódrészletet. Ezzel a tényleges mozgáson kívül eső időtartamok meghatározhatók, ezeken a helyeken a szenzorok értékét zérus értékkel írjuk felül, ezzel tovább csökkentve a zajok és integrálás okozta halmozódó hibákat.

A leírt adattisztítási lépéseket követően az algoritmus téglalap szabály szerint végzi el a kettős numerikus integrálást a gyorsulás adatokon, ezzel meghatározva a sebesség, majd az elmozdulást az egyes tengelyek mentén. Az integrálás lépései között és után további adattisztítási műveleteket nem végzünk, mivel jelen alkalmazásnál, a viszonylag lassú mozgások esetén egyéb szűrők már nagyban torzítanak az adatokat.

4. MÉRÉSI EREDMÉNYEK

A bemutatott algoritmussal több mérést is végeztünk, melyek során vizsgáltuk a felépített program pontosságát, robusztusságát és esetleges pontatlanságait. Az algoritmus iterációs lépések eredményeként, több lépcsőben történő pontosítás után vált véglegessé. Az alábbiakban egy lineáris egységen végzett teszt eredményeit mutatjuk be.

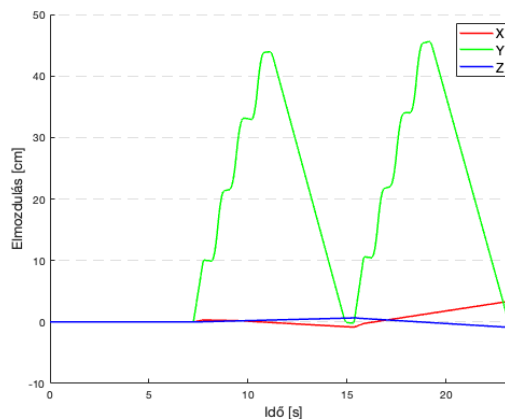
A mozgathoz egy Lucas-Nuelle márkájú EDT 33-3 típusú pozicionáló rendszert használtunk, szinkron szervomotorral és 800/12 típusú egytengelyű, fogasszíj hajtású lineáris egységgel. A vezérléshez és a mozgásprofil meghatározásához a PosiDrive szoftvert használtuk, melyben manuálisan meghatároztuk az elérendő pozíciókat és célsebességeket, míg a szükséges gyorsításokat a szoftver számította. Jelen esetben a mozgás egy lépcsős mozgásprofil alapján valósult meg, 4 darab, egyenként 10 cm-es előre történő mozgást követett egy 40 cm-es távolságú, ellenkező irányú mozgás a kezdeti pozícióba. Az egyes mozgások között állandó, $0,1 \text{ m/s}$ célsebességet határoztunk meg. A leírt mozgást egymást követően kétszer hajtottuk végre, a mérés teljes időtartama így 25 s volt. A globális koordináta-rendszerbe transzformált gyorsulásértékeket a 4. ábra mutatja.



4. ábra X, Y és Z tengely menti gyorsulásértékek az idő függvényében, a globális koordinátarendszerbe való transzformálást követően

A fenti ábrán megfigyelhető, hogy a mozgás a szenzor Y tengelye mentén valósult meg, az X és Z tengely adatait kis mértékű zaj terheli. Az Y tengely kiemelkedő amplitúdói jelzik az egyes célpozíciók közötti gyorsításokat és lassításokat. A Z tengely mentén megfigyelhetők nagyobb amplitúdójú és frekvenciájú zajok, melyek a következő lépésekben szűrésre kerültek.

Az IMU szenzor adatai alapján számított végleges elmozdulásokat az idő függvényében az 5. ábra mutatja. Az Y tengely mentén megfigyelhető a korábban leírt mozgásprofil, míg az X és Z tengely mentén az elmozdulás a mérés időtartama alatt zérus, ebből következően a korábban bemutatott szűrési módszerek megfelelően működtek, a zajok nem okoztak nagy mértékű hibát az integrálás során. A kirajzolt mozgásprofilból továbbá megfigyelhető, hogy a mozgás első két szakaszán a mért elmozdulás 10 cm volt, viszont ezt követően hozzávetőlegesen 12,5 cm elmozdulás látható a valós 10 cm helyett. Ezt követően a kezdő pozícióba való visszaállás során hibajelenség nem lépett fel, a mért adatok alapján a szenzor zérus pozícióba mozgott, megfelelően a beállított mozgásprofilnak. Ezt követően a második mozgás során hasonló eredmények figyelhetők meg, a kezdeti szakaszos mozgás során kis mértékű pontatlanságok léptek fel, azonban a visszaállás mérése megfelelő pontossággal történt. Ebből következően a beépített szűrő algoritmusok megfelelően működtek, azok a statikus időtartamokban is megszüntették a zajokból és mérési pontatlanságból eredő hibákat. A lépcsős profilú szakaszok pontatlansága a gyorsulásértékek amplitúdójának kis mértékű eltéréseiből adódik, mely hiba a mérések során maximum 4%-os eltérést okozott. Ezek mellett az X és Z tengelyek mentén drift jellegű hiba figyelhető meg, mely a szűrés mellett fennmaradó zajok, majd az integrálási műveletekből adódik.



5. ábra Az IMU szenzor adatai alapján meghatározott elmozdulások az idő függvényében

5. ÖSSZEFOGLALÁS

Munkánk során létrehoztunk egy inerciális szenzorok adataiból a relatív pozíció meghatározására alkalmas algoritmust. A fejlesztés során Matlab szoftverben fejlesztettünk egy komplementer szűrő alapú programkódot. A teszteléshez egy lineáris egység által, előre meghatározott mozgásprofil mentén mozgattott IMU mérőegységet használtunk. A feldolgozott mérési eredmények megfelelő pontosságot mutattak. Az

eredményben további szűrési módszerrel nem javítható kis mértékű pontatlanságok figyelhetők meg, ezek feltehetően az inerciális szenzorok és az integrálási műveletek jellegéből származnak.

KÖSZÖNETNYILVÁNÍTÁS

A közlemény a TKP2020-NKA-10 számú projekt keretében a Kulturális és Innovációs Minisztérium Nemzeti Kutatási Fejlesztési és Innovációs Alapból nyújtott támogatásával, a 2020. évi Tématerületi Kiválóság Program pályázati program finanszírozásában valósult meg.

A programot a ZalaZONE Autóipari Próbapálya Zala Kft. támogatta.

IRODALMI HIVATKOZÁSOK

- [1] Harrap R. *An Overview of LIDAR: collection to applications*, Academia, Oslo, 2010.
- [2] Lidar mapping, <https://www.vectornav.com/applications/lidar-mapping> (Utolsó letöltés: 2022. 09.01).
- [3] Tariqul I., Saiful I., Shajid-Ui M., Hossam-E H. *Comparison of complementary and Kalman filter based data fusion for attitude heading reference system*. AIP Conference Proceedings. 2017, 1919(020002).
- [4] Saiful I., Shajid-Ui M., Tariqul I., Syedul A., Hossam-E H. *A Low Cost MEMS and Complementary Filter Based Attitude Heading Reference System (AHRS) for Low Speed Aircraft*. iCEEiCT Conference. 2016.