

# MARG szenzor alapú pozíció becslés periodikus mozgásra

Nagy Balázs

Mechatronika, Optika és Gépészeti Informatika Tanszék  
Budapesti Műszaki és Gazdaságtudományi Egyetem  
Budapest, Magyarország  
nagybalazs@mogi.bme.hu

**Absztrakt** – A kutatás során magnetométer, giroszkóp és gyorsulásérzékelő szenzorok (MARG szenzor) jeleinek fúzióját vizsgáltuk. A cikk fókuszosa a felügyelt tanulás által végzett szenzorfüzió, amely a különböző típusú mért adatok párhuzamosfeldolgozásán alapuló pozíció becslését jelenti periodikus mozgások esetén. A tanuló algoritmusok alkalmazhatóságát egy inga gyakorlati példáján keresztül vizsgáltuk meg

**Kulcsszavak** — MARG, felügyelt tanulás, szenzor-fúzió, pozíció becslés

## I. BEVEZETŐ

Az orientáció és pozíció becslés elengedhetetlen feltétele a legtöbb mechatronikai feladatnak, mint például a robot navigációnak. A navigációs rendszer az autonóm mobil robot fontos része. Számos gyakorlati alkalmazás létezik a problémára [1]. Vannak rendszerek, amelyek globális helymeghatározó rendszert (GPS) [2] használnak. Másrendszerek fedélzeti kamerát [3], lézeres távolságmérő (LIDAR) szenzort [4] vagy egyéb külső megfigyelő rendszert [5] alkalmaznak. A robot fedélzeti számítógépén a navigáláshoz megbízható megoldás szükséges, amely még korlátozott számításikapacitású robotokon is megvalósítható. Léteznek megoldások, amelyek a számítási szempontból költséges algoritmusokat a robot fedélzeti számítógépéről kiszervezik a robot környezetébe [6]. Ebben a koncepcióban nincs szükség nagy számítási kapacitásra a fedélzeten így az energiafogyasztás alacsonyabb szinten tartható a mobil roboton és az üzemidő is növelhető. Az erőforrás kiszervezésen alapuló megoldások hátránya a vezeték nélküli kapcsolat. A kapcsolat megszakadása esetén teljes funkcióvesztés következhet be. A képfeldolgozáson alapuló vagy

3D pontfelhő alapú alkalmazások nagy számítási kapacitást igényelnek az egyidejű térképezés és lokalizáció (SLAM) [7] végrehajtásához. Ezek a módszerek nem alkalmasak korlátozott számítási kapacitású kis robotok számára. Általánosságban elmondható, hogy a legtöbb lokalizációs algoritmus előre meghatározott térképet használ. A térkép elkészítése a kezdetektől fogva kihívást jelent. Egy teljesen automatizált kis méretű robotnak képesnek kell lennie a külső érzékelőktől függetlenül történő működésre, csak fedélzeti érzékelők használatával. Ezen megfontolások mentén egy MARG érzékelő jó választásnak tűnik, melyre a fedélzeten futó navigációt lehet építeni. Madgwick kutatásában egy MARG szenzorjelek feldolgozásán alapuló pozíció becslő algoritmust mutat be. Az alkalmazásban [8] a MARG érzékelőt egy ember lábára szerelték, és egy Arduinot használtak adatgyűjtésre. A mérés után az algoritmus kiszámította az érzékelő útvonalát. A számítást offline módon végezték, nem valós időben, és a tesztet egyedi jellemzőit felhasználták az algoritmus fejlesztése során. A számítás a járás mozgás lépésfrekvenciáján alapul. A frekvenciaelemzés csak periodikus mozgások esetén nyújthat hasznos információt. A gyaloglás során egy másik figyelemre méltó jellemző, hogy a láb minden lépésben a padlóval ütközik. Ez a hatás észlelhető és felhasználható az összegyűjtött hiba kiküszöbölésére egy lépés során. Ez a korrekció megakadályozhatja a pozícióbecslés összesített hibából adódó elcsúszását. Mivel a mért jel a gyorsulás ezért az algoritmusnak kettős integrálást kell végrehajtania a helyzet becsléséhez. A zaj integrálása problémát jelent, ami a pozíció becslés elcsúszását eredményezheti. A hibát további szenzorfüzióval lehet kompenzálni [9]. A bemutatott alkalmazásban a szenzorfüziós algoritmust explicit definiálták. A kérdés az, hogy tanuló algoritmusokkal lehet-e

javítani a becslés pontosságán, esetleg optimalizálni a számítást, hogy korlátozott számítási kapacitással robotok fedélzeti feldolgozó egységén lehessen alkalmazni közel valós idejű alkalmazásokban. Bizonyított, hogy a MARG szenzorok jól használhatók orientációbecslési feladatra [10]. A gyorsulásmérő információt nyújt a gravitációs erőről, a giroszkóp képes reagálni a gyors mozgásváltozásra, a magnetométer pedig megakadályozhatja a hosszú távú elcsúszási hibát. E három érzékelő fúziója orientáció becslési feladatoknál egyesíti az egyes érzékelők előnyeit. A kiegészítő szűrő [11] gyors és számításiszempontból olcsó módszer a MARG szenzor adatai alapján történő orientáció becslésére. A pozíció becslése a szenzorokon fellépő zajok és zavarások miatt nagyobb kihívást jelent. Hatékonyabb szenzorfüziós stratégiák kialakításához megoldást jelenthet géptanuláson alapuló algoritmusok alkalmazása [12]. A gépi tanulási technikák nagyon népszerűek a bonyolult problémák kezelésében, az idősor elemzést is bele értve.

## II. MÓDSZERTAN

Ez a tanulmány egy olyan módszer előállításának lehetőségére összpontosít, amely nem függ egy külső megfigyelőrendszertől, csak fedélzeti érzékelőket és számításokat használ pozíció becsléséhez periodikus mozgást végző alkalmazás esetén. Az algoritmus számításigényének elég kicsinek kell lennie ahhoz, hogy korlátozott számítási kapacitással rendelkező alkalmazások esetén is implementálható legyen. Ezen megfontolások mellett célul tűztük ki, hogy a javasolt algoritmus akár egy Raspberry PI fedélzeti egységen is futtatható legyen.

### A. Elméleti háttér

A szenzorfüzió alapján történő pozíció becslés gyorsulásiadatok felhasználásával azt jelenti, hogy az adatokat integrálni kell. A gyorsulás és a helyzet elméleti összefüggését az (1) egyenlet határozza meg.

$$a(t), v(t) = v_0 + \int_0^t a(\tau)d\tau, r(t) = r_0 + \int_0^t v(\tau)d\tau \quad (1)$$

Ahol  $t$  az idő,  $a(t)$  a gyorsulás,  $v(t)$  a sebesség és  $r(t)$  a pozíció. A gyorsulási adatokat erősen befolyásolja a zaj, és a mérési zaj integrálása, melyek torzulást (shift) okoznak a pozíció számításában és

rontják az előrejelzést. A zaj által érintett kapcsolatokat az alábbi egyenletek határozzák meg: (2) (3) és (4).

$$a(t) + n_a(t) \quad (2)$$

$$v_0 + \int_0^t (a(\tau) + n_a(\tau))d\tau = v(t) + e_v(t) \quad (3)$$

$$r_0 + \int_0^t (v(\tau) + e(\tau))d\tau = r_0 + v_0 t + r(t) + e_r(t) \quad (4)$$

Ahol  $e$  a kalkulált hiba és  $n_a$  a zaj. Ha ismertek a kezdetifeltételek és  $r_0 = 0$  és  $v_0 = 0$  akkor az egyenlet egyszerűsíthető (5).

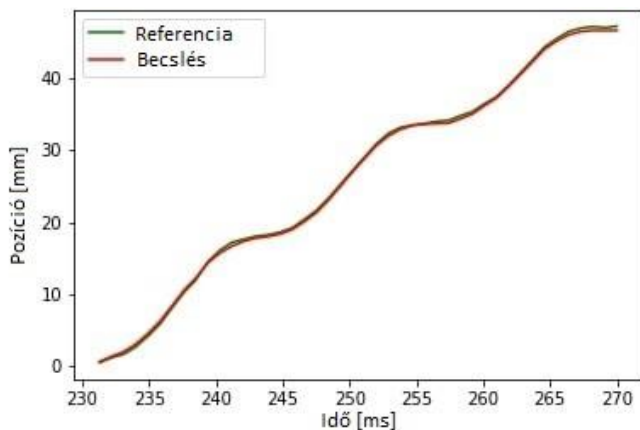
$$\int_0^t \int_0^{\tau_1} (a(\tau_2) + n_a(\tau_2))d\tau_2 d\tau_1 = r(t) + e_r(t) \quad (5)$$

A gyorsulási adatok kettős integrálása megegyezik a pozíció adatokkal, de a zaj kettős integrálásának eredménye egy nem kívánt eltolási hiba ( $e_r$ ) a pozícióbecslésben. A hibakezelés kulcsfontosságú lépés, de a hiba típusa és a hibaparaméterei, például az eloszlás vagy az eltolás, nem minden esetben ismertek. A legtöbb esetben  $e_r(\infty) = \infty$ . A pozíció kiszámításához a szükséges információk szerepelnek a gyorsulásmérésben, de minden esetben zaj terheli azokat. A valós mérések és neurális hálózat betanítása előtt szimulációval vizsgáltuk a lehetséges zaj típusokat. A szimulációk azt mutatták, hogy egy neurális hálózat képes zajszűrőfunkciókat is megtanulni és kezelni a fellépő zajokat.

### B. Szimulációs tesztek

A valós problémára történő alkalmazás előkészítéseképpen szimulált jeleken teszteltük a zajoknak a neurális hálózattanulási képességére gyakorolt hatásait. A szimuláció során két különböző paraméterekkel rendelkező szinuszos hullám modellezte a bemeneteket. A jelekre külön alkalmaztunk zajt a mintavételi időre és az amplitúdóra. Az egyesjelekre szuperponálódó zajok minden esetben Gauss eloszlásúak voltak. A zajjal terhelt jelekből ez után kétféle módszerrel képeztünk kimeneti jelet. Első lépésben csak a szinuszjelek összeadásával, második lépésben az egyik szinuszos jelet numerikusan integráltuk. A kimeneti jelet külön zajjal terheltük. A kimeneti jelen alkalmazott zaj modellezte a mérésrendszer hibáját. Mivel a valós eset során

használt adatok se tökéletesek, számítanunk kell a tanuló adatbázis elkészítése során használt mérőrendszer pontatlanságából adódó hibára. A két bemeneti szinuszciklus két féle zajt vizsgáltunk. A zajt a mintavételi időn és a zajt az amplitúdón. A kimeneti jelen további zaj komponenszt szuperponáltunk az amplitúdóra. A kimenet kialakítását illetően két esetet különböztettünk meg. Összegzés alapút és numerikus integrálás alapút. Az összes lehetséges kombinációt figyelembe véve 16 különböző esetet vizsgáltunk attól függően, hogy a bemeneteken, a kimeneten és a mintavételi időn jelen van-e zaj vagy nincs. A tesztek során kiderült, hogy az egyes bemenetek összegzésével generált kimenet becsléséhez kevés réteg és az egyes rétegekben alkalmazott kevés neuron is elfogadható eredményt hozott. A numerikus integrálás becsléséhez viszont szélesebb háló struktúrára van szükség. A háló megfelelő bővítésével, illetve a hiperparaméterek hangolásával az integrált kimenet is jelleghelyesen becslhető volt. A kibővített tesztelés során használt hálózat predikcióját az összes zajjal terhelt esetben integrálással előállított kimenet esetén az 1. ábra tartalmazza.

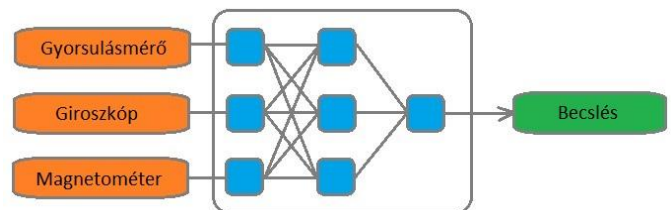


1. ábra: Szimulált integrált kimenet becslése

### C. Szenzorfüzió felügyelt tanulással

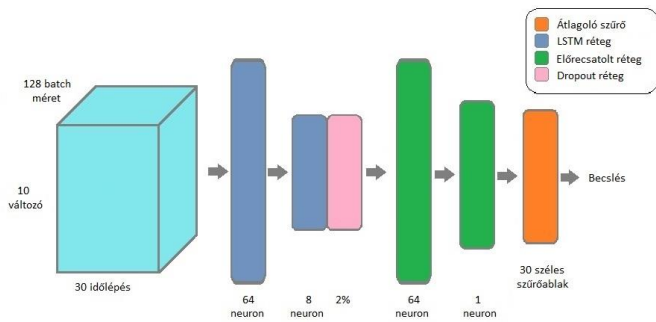
Az előrecsatolt neurális hálózatok univerzális közelítési képessége azt jelenti, hogy egy zárt halmazon definiált bármely folytonos függvényt tetszőleges pontossággal egységesen közelíteni lehet egy háromrétegű előrecsatolt neurálishálózattal [13]. A probléma megoldása lehet egy neurális hálózat, amely megtanulja, hogyan becslje meg a pozíciót a MARG érzékelő adatai alapján. A létező módszerek [10] [11] főleg az orientáció becslést

explicit modellel közelítették meg. Az egyes érzékelők adatait külön szűrik, és a szűrési lépés után az érzékelők adatait egy előre meghatározott sorrend szerint kombinálják. Ezzel a megközelítéssel az a probléma, hogy az érzékelő fúziójának irányai és kapcsolódási pontja előre meghatározottak. Az algoritmus nem rugalmas. Az algoritmus rugalmasabbá tétele és robusztusabb használata miatt a 2. ábrán látható megközelítést javasoljuk.



2. ábra: Javasolt megközelítés

Egy ilyen hálózat betanításához a tanulási folyamat során egy külső kamerarendszer adja meg a referencia adatokat. Ily módon a feladat felügyelt tanulási regressziós feladatként értelmezhető. A szinkronizált mérőrendszer egyszerre naplózza a szenzor helyzetét egy abszolút koordináta-rendszerben és a fedélzeti MARG érzékelő szenzor adatait. A külső kamerarendszer biztosítja a 3 dimenziós pozíció adatokat és a szenzor orientációját kvaternió alakban. A külső kamerarendszer adatait csak a tanulási szakaszban használjuk referenciaként. Miután az algoritmus megtanulta, hogyan lehet becslni a pozíciót a szenzor jelekből, a predikciós állapotban csak az érzékelő adataira van szükség. A szinkronizált mérési adatok feldolgozásához egy sekélyneurális hálózatot használtunk. A megvalósított hálózat egy 64 neuronnal és egy 8 neuronnal rendelkező LSTM réteget, egy 64 neuronnal és egy 1 neuronnal rendelkező előrecsatolt réteget tartalmazott. A mérés során a szenzort egy ingára rögzítettük, így elegendő volt egy kimeneti változót definiálni. A hálózat végén egy átlagoló szűrőt alkalmaztunk a becslés simábbá tételére érdekében. Az LSTM rétegek modellezték az idősor adatait, az előrecsatolt rétegek pedig elkészítették a végső becslést. A hálózat lokális minimumba ragadását kiküszöbölendő az LSTM és az előrecsatolt rétegek közé egy drop out réteget adtunk az architektúrához.



3. ábra: Neurális hálózat struktúrája

### III. MÉRÉSI ÖSSZEÁLLÍTÁS

Mérési tesztetként a szenzort egy ingára rögzítettük. Az ideális inga mozgásának 1 szabadságfoka van. Ez az eset elég egyszerű a különböző tanulási algoritmusok teszteléséhez. Az ingára szerelt felmarkerezett szenzort a 4. ábra szemlélteti.



4. ábra: Mérési összeállítás

A valós mérések során Phidget Spatial 3/3/3 szenzort használtunk [14]. Ez az érzékelő tartalmaz egy 3 tengelyes gyorsulásmérőt, egy 3 tengelyes giroszkópot és egy 3 tengelyes magnetométert. A szenzor specifikus adatokat a 1. táblázat tartalmazza.

1. táblázat: Szenzorspecifikus paraméterek

#### Magnetométer

Felbontás	400 $\mu\text{G}$
Offset hiba	$2^\circ$

#### Giroszkóp

Maximális sebesség	$400^\circ/\text{s}$
Felbontás	$0,02^\circ/\text{s}$
Drift	$4^\circ/\text{perc}$

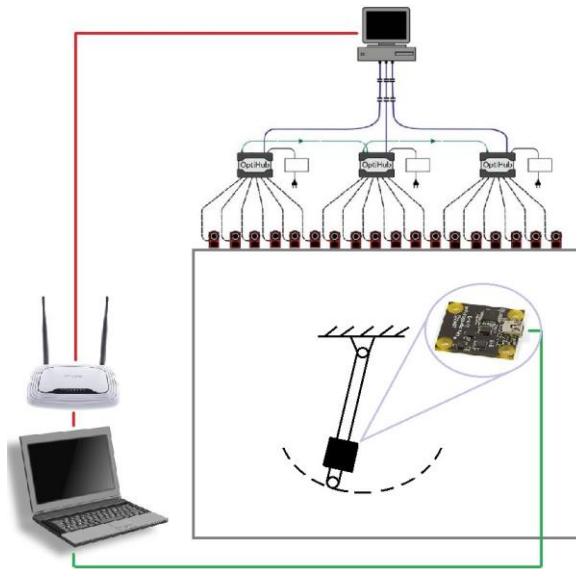
#### Gyorsulásmérő

Felbontás	228 $\mu\text{g}$
Méréstartomány	$\pm 5\text{g}$
Zaj (X, Y tengely)	300 $\mu\text{g}$
Zaj (Z tengely)	500 $\mu\text{g}$

#### Szenzor

Maximális mintavételi sebesség	4 ms/minta
--------------------------------	------------

A mintavételi idők eltérnek a három szenzornál. A magnetométer mintavételi ideje 8 ms, míg a gyorsulásmérő és a giroszkóp mintavételi ideje 4 ms. A mintavételi időt a mérések során 9 ms-ra választottuk, ami a későbbiekben fenntartható ütemezésnek bizonyult. A méréseket egy 18 kamerát tartalmazó mozgáskövető (Motion Capture) laboratórium megfigyelőterében végeztük. A kamerarendszer nyomon követheti a megfigyelőterben található speciális jelölőkkel ellátott különböző testek pozícióját és orientációját. A szenzort 3 marker jelöli. A rendszer frissítési frekvenciája 120 képkocka/másodperc(FPS). Az összes adatot egy Lenovo Y520 laptpra rögzítette. A hálózat tanítása is ezen a laptpon történt egy NVIDIA GeForce GTX 1050 grafikus kártyával. A mérő-rendszer sematikus rajza a 5. ábrán látható



5. ábra: Mérési összeállítás

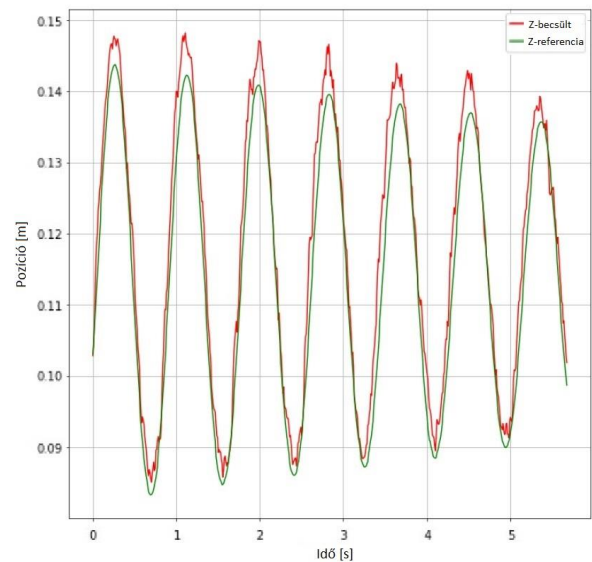
### A. Adat előkészítés

A neurális hálózat tanítása előtt a mérésből származó adatok további előfeldolgozására volt szükség. A tanítás bizonyítottan hatékonyabban működik normalizált adatokon. A mérési összeállításban a külső megfigyelő rendszer a pozíció adatokat saját abszolút koordináta rendszerében rögzítette. A neurális hálózat szempontjából ez függővé teszi a tanulást a megfigyelt téren belüli elhelyezkedésétől. Relatív elmozdulások számolása áthidalhatja ezt a problémát és a vizsgált jelenség fizikai mivoltából adódóan a ciklusonként fellépő elmozdulások jól behatárolhatóak, megkönnyítve a normalizálást. A differencia számítást mind az időbélyegekre, mind a pozíció adatokra elvégeztük. Felügyelt tanítás esetén a címkézetlen adatok szintén nem kívánt hibát vihetnek a rendszerbe. A különböző frissítési frekvenciával rendelkező szenzorok miatt ezt a problémát is kezelni kell a tanítás előtt. Ideális esetben a MARG szenzorból érkező és a mozgáskövető rendszerből érkező adatok szinkronban vannak. A mozgáskövető rendszer protokollja szerint egy nulladrendű tartó biztosítja a legutolsóvalós érték megtartását az újabb mérésig. Ezen megfontolással mindig a MARG szenzorból érkező adatokhoz igazítjuk a szinkronizálást. A MARG szenzoron belül a magnetómer adataira szintén egy nulladrendű tartót alkalmaztunk, mivel ez a szenzoregység kétszer lassabban üzemel. A hibás adatsorok javítására interpolációt is

használhattunk volna, ezt a megoldást azonban elvetettük. Interpoláció be-vezetésével egy irányított számítási lépést vittünk volna a folyamatba, ami a tanuló algoritmust befolyásolja, illetve ezt a lépést egy kellően komplex algoritmus meg is tudja tanulni.

### EREDMÉNYEK

A neurális hálózat a tanulási fázisban a megfigyelő rendszer abszolút koordináta rendszerben mért pozíció adatait is felhasználta. A tanulás során az algoritmus bemenetként a 9 szenzor értéket és a mintavételi időt használta. Elvart kimenetként a külső megfigyelő rendszer pozíció adatait kapta meg, mint címkéket. Az eredmény a 6. ábrán látható.



6. ábra: Predikció eredménye (zöld - referencia jel, piros - becsült jel)

Az algoritmus átlagosan 6,7 mm hibát ért el. A maximális kitérés hiba 21 mm volt. A relatív magas maximális hiba mögött az időbeli késés áll. A jelalakokból látszik, hogy alakra követi a becsült pozíciógrafikonja a referencia jelet, de van benne egy kis késés. Az adatbázis 6324 tanuló mintát, 2996 validációs mintát és 2996 tesztmintát tartalmazott. A tesztmintákat külön mérés során rögzítettük, hogy megakadályozzuk az adatokszivárgását a tanulási szakaszban és biztosítsuk az adatokfüggetlenségét a tesztelés során. A betanított algoritmus számítási igényét illetően egy Raspberry PI 3 B+ célhardverre implementálva is képes feldolgozni a mérés során alkalmazott Phidget Spatial 3/3/3MARG szenzor jeleit. A tesztek alapján kijelenthető, hogy alkalmas

kis számítási kapacitással rendelkező alkalmazásokfedélzeti kiegészítő rendszerének.

## ÖSSZEFOGLALÁS

A tanulmány végeredményeként a javasolt neurális hálózaton alapuló megközelítés megoldást kínál a pozíció előrejelzésére egy MARG szenzor jeleinek feldolgozásával. A neurális hálózat zajos érzékelőadatokon történő használata csökkentheti a zajszintet és kihasználhatja a zaj rejtettstruktúráját is. Fontos szempont volt a hálózat struktúra kialakításánál, hogy a számítási kapacitás a lehető legkisebb legyen. Ez magában foglalja a tanítási fázist is. Ez volt a fő oka, hogy a tesztelés során sekély hálózatot használtunk. A különböző hatások és viselkedésmódok vizsgálata alapján az alap szintű megoldás értékes információkat nyújt a jövőbeli fejlődéshez. Ez a tanulmány nem vizsgálta a hiperparaméterek hangolásának hatását a neurális hálózat fejlesztése során. A tesztek során nem optimalizáltuk az epochok számát, az időbeli lépéseket, a bemeneti jellemzőket és a tanító mintákat. A neurális hálózat teljesítménye javítható ezen paraméterek hangolásával. A hiperparaméterek hangolása alapját képezheti további jövőbeni kutatásoknak. A valós életben végzett tesztesetek tapasztalatai rámutattak, hogy a neurális hálózat megtanulta a mérés beállításának néhány sajátosságát, így az algoritmus nem független a tesztesettől. Az alkalmazás robusztusabbá tétele érdekében ezt a hatást a jövőben büntethetjük a hálózat tanulása során, így a rendszer túltanulása elkerülhető lenne.

## Köszönetnyilvánítás

A jelen publikációban megjelenő kutatások az ITM NKFIÁ által nyújtott TKP2020 IKA támogatásból, az NKFIH által kibocsátott támogatói okirat alapján valósultak meg (projekt azonosító: TKP2020 BME-IKA-MIFM). A projektben használt eszközök beszerzése kapcsán köszönet illeti BudaiCsabát.

## Hivatkozások

[1] G. Szayer, F. Tajti, M. Burdelis and P. Korondi, "A novel potential field method for path planning of mobile robots by adapting animal motion attributes," *Robotics Auton. Syst.*, no. 82, pp. 24-34, 2016.

Recent Innovations in Mechatronics (RIiM) Vol 7. (2020) No 1.  
DOI: 10.17667/riim.2020.1/9.

[2] Y. Kim, J. An and J. Lee, "Robust Navigational System for a Transporter Using {GPS/INS} Fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3346-3354, 2018.

[3] C. H. Yun, Y.-S. Moon and N. Y. Ko, "Vision Based Navigation for Golf Ball Collecting Mobile Robot," *13th International Conference on Control, Automation and Systems (ICCAS 2013)*, 2013.

[4] S. Gatesichapakorn, J. Takamatsu and M. Ruchanurucks, "ROS based Autonomous Mobile Robot Navigation using 2D {LiDAR} and {RGB-D} Camera," *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, 2019.

[5] L. P. a. A. F. a. G. Pereira, R. Mesquita, E. Silva, W. Caminhas and M. Campos, "Robot navigation based on electrostatic field computation," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1459 - 1462, 2006.

[6] B. Nagy and G. Štefan, "Intelligent Space environment for ethorobotics," *RECENT INNOVATIONS IN MECHATRONICS*, vol. 5, pp. 1-5, 2018.

[7] K. Yousif, A. Bab-Hadiashar and R. Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," *Intelligent Industrial Systems*, vol. 1, no. 5, pp. 289-311, 2015.

[8] S. O. H. Madgwick, A. J. L. Harrison and R. Vaidyanathan, "Estimation of {IMU} and {MARG} orientation using a gradient descent algorithm," *2011 IEEE International Conference on Rehabilitation Robotics*, 2011.

[9] G. Zizzo and L. Ren, "Position Tracking During Human Walking Using an," *MDPI Sensors*, vol. 17, p. 2866, 2017.

[10] M. R. Alfonso, A. Frizera and K. F. Coco, "Magnetic, angular rate and gravity sensor systemfusion for orientation estimation," *Studies in HealthTechnology and Informatics*, pp. 261-266, 2015.

[11] J. Wu, Z. Zhou, J. Chen, H. Fourati and R. Li, "Fast Complementary Filter for Attitude Estimation Using Low-Cost (MARG) Sensors," *IEEE Sensors Journal*, pp. 6997 - 7007, 2016.

[12] D. Tang, J. Botzheim and N. Kubota, "Supervised Learning Based Multi-modal Perception for Robot Partners using Smart Phones," *Acta Polytechnica Hungarica*, vol. 11, no. 8, pp. 139-159, 2014.

[13] Z. Zainuddin and S. P. Fard, "Approximation of multivariate  $2\pi$ -periodic functions y multiple  $2\pi$ -periodic approximate identity neural networks based on the universal approximation theorems," in *11th International Conference on Natural Computation (ICNC)*, 2015.

[14] Phidgets Inc., [Online]. Available: <https://www.phidgets.com/?prodid=30>. [Hozzáférés dátuma: 2020.11.26].