

AUTONÓM 3D NAVIGÁCIÓS STÍLUSOK

Az utóbbi évtizedben a robotirányítási algoritmusok rohamos fejlődésnek indultak. Számos olyan munka jelent meg, amely dinamikus vagy ismeretlen környezetben történő navigálással foglalkozik. (Lammens et al., 1993; Arkin, 1989; Brooks, 1991). Cikkünkben egy olyan algoritmust mutatunk be, amely egy, a fejlődési irányvonalak között népszerű neuro-fuzzy megközelítésben oldja meg egy három dimenzióban mozgó, az akadályok elkerülésével a célt megközelítő robot navigációjának problémáját.

A háromdimenziós navigációs algoritmus tervezése során két szempontot tartottunk szem előtt. Az egyik: kiválasztani egy széles körben elterjedt, jól használható navigációs módszert, ami az irányítási modellünk alapjául szolgál majd. A másik pedig, hogy olyan algoritmust tervezzünk, amely képes különböző egyszerűbb vezetési stílusok megvalósítására és tanulására is, de az algoritmus számítási komplexitása ne legyen túl nagy.

A potenciál alapú navigáció¹ (Latombe, 1991) széles körben elterjedt módszer, elsősorban egyszerűségének és valósidejű alkalmazhatóságának köszönhetően (Bronstein and Koren, 1989). Számos gyakorlati navigációs vagy robotirányítási algoritmus épül a potenciál alapú módszerre vagy ennek valamilyen változatára, különösen olyan esetekben, ha a fő cél az akadályok elkerülése (Bronstein and Koren, 1991), (Kubota et al., 1999). A modell alapgondolata, hogy virtuális vonzó és taszító erők hatnak a robotra. A mozgás irányát és sebességét a virtuális erők eredőjéből határozzuk meg. A potenciál alapú vezetés előnyei mellett meg kell említenünk, hogy egyszerűségéből adódóan korlátai is vannak. Szűk folyosóban vagy közel egyenletes elosztású akadályok között a PBG-vel vezetett robot viselkedése nem kielégítő. Szükség van egy általánosabb modellre, ezért vezetjük be a vektortér alapú vezetést.

A kidolgozott algoritmus hatékonyságát elsősorban az alkalmazott neuro-fuzzy technika biztosítja, amely képes mind a közönséges PBG, mind a vektortér alapú általánosítás közelítő megvalósítására. Ez a megközelítés közismert előnyei mellett azt is megkönnyíti, hogy korábbi szituációk alapján szerzett tudást is fel lehessen használni az ismeretlen környezetben történő navigálás során. (Fukuda and Kobayashi, 1999; Piaggio and Zaccaria, 1997; Kubota et al., 1999). A robot eltanulhatja továbbá más robotok vezetési stílusát, illetve ember által is tanítható.

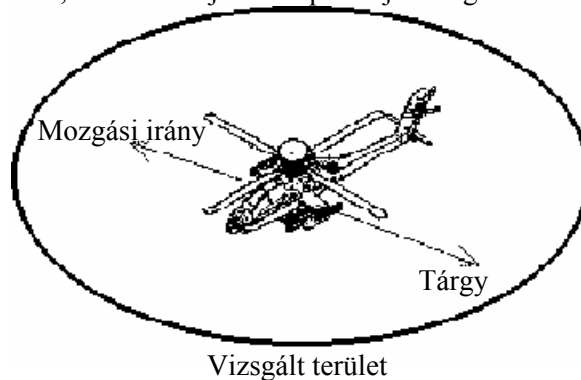
¹ PBG — Potential Based Guiding.

A fuzzy logika alkalmazásának egyik előnye, hogy képes modellezni és később reprodukálni szakképzett kezelők cselekvéseit a konkrét matematikai modell hiányában is. Hátránya viszont, hogy nincs általánosan alkalmazható tervezési eljárás optimalitás, redukálhatóság és egyéb szempontokból. A létrehozott algoritmusok — függetlenül attól, hogy szakképzett kezelők hozták létre vagy tanulás során keletkeztek — tartalmazhatnak redundáns, gyengén befolyásoló vagy akár felesleges részeket. Hasonló a helyzet a neurális hálózatok esetében, a jó közelítés eléréséhez szükséges szintek és összeköttetések számát egyes megközelítések jócskán túlbecsülik. Ez a szükséges számítási kapacitást jelentősen megnöveli. Ugyanakkor a fuzzy közelítő eljárások elméletileg szintén exponenciális számításidő- és tárigénnyel rendelkeznek (Kóczy and Hirota, 1990). Ezért ismertetünk egy komplexitás-csökkentő eljárást is, amely a lehetőségeken belül optimálisan összetömöríti a létrehozott modellt.

Fontos megemlíteni, hogy a fuzzy szabálybázis tervezése során szem előtt tartott két cél ellentmond egymásnak: minél jobb közelítést szeretnénk elérni, míg a számítási komplexitást csökkenteni kívánjuk. A tervezés fő problémája e két ellentmondó cél között a megfelelő kompromisszum megtalálása.

A ROBOTIRÁNYÍTÁS DEFINÍCIÓJA

Ebben a részben röviden bevezetjük a vezetési stílus fogalmát. Robotunk, esetünkben egy helikopter, látókörében észlelni képes a különböző akadályokat (1. ábra). A helikopter látóköre érzékelő irányokból tevődik össze. Az érzékelő rendszer szolgáltatja az adott irányban lévő legközelebbi tárgy távolságát a helikoptertől. Ezeket a távolságinformációkat kamerák képeinek feldolgozásából vagy távolságmérő szenzorok segítségével nyerhetjük. A szenzorok nem csak egy síkban vannak, hanem a teljes teret próbálják meg lefedni.



1. ábra. A PBG alap gondolata

A vezetés célja, hogy meghatározzuk a mozgási irányt a látókörből nyert információ és a megvalósított stílus alapján. Többfajta vezetési stílust valósíthatunk meg, tekintsünk ehhez két szélsőséges példát.

Egy veszélyes anyagot szállító jármű alapszabálya lehet a következő: „minél messzebb minden objektumtól”. Ugyanakkor egy lopakodva közlekedő jármű esetén hasznos lehet a „maradjunk közel az objektumokhoz” szabály. Megjegyezzük, hogy a robotirányítástól nem követeljük meg a probléma-megoldási képességet, például egy labirintusból való kitaláláshoz már többletinformációkra van szükség.

POTENCIÁL ALAPÚ VEZETÉSI MODELL

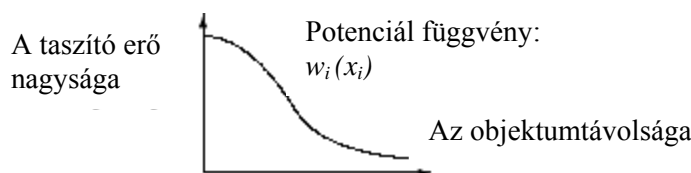
A potenciál alapú vezetési modell alapötlete, hogy a robotot az őt körülvevő tárgyak taszítják vagy vonzzák. Minden tárgy tulajdonképpen egy virtuális erőt hoz létre, és ezek hatnak a robotra. Ezen erőket összegezve kapjuk az akadályok elkerülése szempontjából optimális mozgásirányt, amelyet a megközelítendő célpont irányának megfelelően egy kiértékelő algoritmus szerint módosíthatunk, vagy a célpont irányában egy külön vonzó erőt vehetünk fel. A virtuális erők számítását minél gyorsabban kell elvégezni a folytonos és reakcióképes vezetés érdekében.

A taszító erő nagysága általában fordítottan arányos a tárgy és a robot közti távolsággal, azaz²:

$$|y| = \frac{1}{(\text{távolság})^a} \quad (1)$$

ahol: a — egy állandó (általában egynél nagyobb).

Az így kapott függvények minden stílus esetében jellemzően nemlineárisak.



2. ábra. Egy általános potenciálfüggvény

Számos alkalmazásban ugyanazt a képletet alkalmazzák az összes szenzorirányban³, ami szimmetrikus potenciálfelületet eredményez. A különböző vezetési stílusok létrehozásához természetesen aszimmetrikus potenciálfelületre is szükség lehet. Erre egyszerű példa a közlekedés „szabálya”: a járművek jobbra tartanak, tehát a robotnak a jobboldalt lévő objektumokhoz közel, míg a baloldalt lévő objektumoktól lehetőleg távol kell maradnia.

² Bronstein and Koren, 1989.

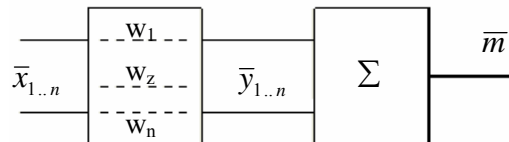
³ Bronstein and Koren, 1991.

A legtöbb esetben az objektumok akadályok, ezért taszító erővel hatnak a robotra. Léteznek olyan esetek, amikor a vonzóerő a kívánatos. Ilyen eset, ha a robotnak egy falat kell követnie (pl. ha folyosókon használjuk a robotot), hiszen ha a robot túlságosan eltávolodott a faltól, akkor a falnak vonzania kell a robotot. Negatív szakaszokat is tartalmazó potenciálfüggvények segítségével ezek a stílusok is megvalósíthatók.

Vizsgáljuk meg részletesebben az irányító algoritmust. A robot látóköre vonalakkal áll. Az irányokat jellemezhetjük az azonos irányú egységvektorokkal. Ezeket jelölje: \bar{e}_i , a megfelelő irányban mért távolságokat pedig x_i . Az irányító eljárás két lépésből áll: Az első lépésben az egyes irányokból ható virtuális erők nagyságát számoljuk ki minden egyes irányra:

$$\bar{y}_i = \bar{e}_i \cdot w_i(x_i) \quad (2)$$

ahol: $w_i(x_i)$ — az i -edik potenciálfüggvény, $i = 1 \dots n$ és n az érzékelő irányok száma.



3. ábra. Potenciál alapú vezetés blokkvázlata

Az algoritmus számításigénye kritikus a valós idejű alkalmazhatóságra nézve, ezért a megvalósítás során közelítő módszereket alkalmazunk. Mint már láttuk, a potenciálfüggvények többnyire erősen nemlineárisak, ezért érdemes lágyszámítási (soft computing) eszközöket használni.

A POTENCIÁL ALAPÚ VEZETÉSI MODELL FUZZY KÖZELÍTÉSE

Ebben a részben a PSG⁴ fuzzy eljárást mutatjuk be, amit a PBG közelítésére használunk és későbbi algoritmusok alapjául is szolgál. Az általunk választott szabálybázis különleges abban az értelemben, hogy a szabályok száma nem nő exponenciálisan a bemenetek számának növelésével. Ezt az biztosítja, hogy a potenciál felület egy négydimenziós felület lesz, függetlenül a szenzorok számától.

A 3. ábra alapján a fuzzy szabálybázis bemenetei a távolságok, az algoritmus kimenete pedig a kívánt mozgásirányt meghatározó y_z vektor. *Az algoritmus leírása:*

⁴ PSG — product-sum-gravity

- *antecedens halmazok*: Az antecedens fuzzy halmazokat $A_{i,j} : \mu_{A_{i,j}}(x_i), i = 1 \dots n, j = 1 \dots m, x_i \in X_i$ adja meg, ezek minden X_i halmaz felett Ruspini-partíciót alkotnak;
- *konzekvens*: A konzekvens halmazok $B_j : \mu_{B_j}(y) = \delta(b_j)$ alakúak, ahol Y_j a j -edik kimeneti halmaz;
- *megfigyelés*: Az x_i bemeneti változókat szingleton bemeneti A_i^* fuzzy halmazokba fuzzifikáljuk a következőképpen: $\mu_{A_i^*}(x_i) = \delta(x_i)$;
- *szabályok*: Az antecedens halmazok összes lehetséges kombinációja által kapott szabályok száma exponenciálisan nő az antecedens halmazok és a bemenetek számának növelésével (Kóczy and Hirota, 1997). A mi esetünkben viszont a potenciálfüggvény egy 4-dimenziós felület, amelyen a kereszthatásoktól eltekintünk. A szabályokat egyszerű, ha A_i akkor B_i alakban adhatjuk meg. Azaz a szabályok száma $n \cdot m$;
- *következtetés*: A fuzzy következtetés product-sum-gravity (Mizumoto, 1990) eljárásán alapul;
- *product*: Ebben a lépésben a kimenetet a megfigyelés illeszkedése függvényében súlyozzuk, ahol az illeszkedés mértéke: $\mu_{A_{i,j}}(x_i)$;
- *sum-gravity*: Ez a súlypont⁵ defuzzifikációs technikán alapul. A konzekvens halmazokat megfelelően súlyozva és összeadva kapjuk a kimeneti vektort:

$$y_i = \frac{\sum \mu_{A_{i,j}} b_{i,j}}{\sum \mu_{A_{i,j}}} \quad (3)$$

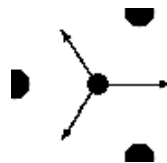
Az antecedens halmazok Ruspini-partíciót alkotnak, ezért (3) nevezője 1.

A POTENCIÁL ALAPÚ MÓDSZER KORLÁTAI

A Potenciál alapú vezetési modell alap gondolata, hogy az objektumok vonzzák vagy taszítják a robotot, nem elég hatékony. Például, ha egy akadály pontosan előttünk jelenik meg, sokkal jobb eredményt kapnánk az egyszerű taszításnál, ha az irányítás például azt mutatná, merről lehet megkerülni.

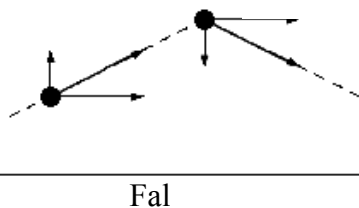
A potenciál alapú vezetés során a robot „beragadhat”. Mint azt a 4. ábrán láthatjuk, a robotot körülvevő tárgyak szimmetrikusan helyezkednek el. Ebben az esetben az eredő virtuális erő nulla lesz, ezáltal a robot mozdulatlanul fog állni. A probléma feloldására különböző megoldások képzelhetők el, például választhatunk egy lehetséges irányt valamilyen kritérium szerint vagy akár véletlenszerűen.

⁵ COG — Center of Gravity



4. ábra. Beragadás

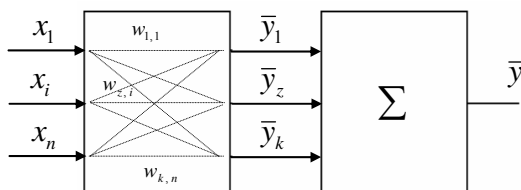
Egy másik tipikus problémás helyzet látható az 5. ábrán. A robotnak a fallal párhuzamosan kellene mozognia, tehát a virtuális erők eredőjének is ilyen irányúnak kellene lennie. A PBG nem képes a megfelelő vektor előállítására.



5. ábra. A PBG-modell „pattogása”

A POTENCIÁLTÉR ÁLTALÁNOSÍTÁSA VEKTORTÉRRE

Az előző rész alapján hasznosnak tűnhet, ha minden egyes irányban mért távolság nem csak a taszítás (virtuális erő) nagyságát, hanem az irányát is befolyásolná. Ebből a célból kiterjesztettük az előbb bemutatott potenciál alapú modellt. Ha az egyes bemenetek más kimenetekre is hatással lehetnek, akkor — az alábbi blokkvázlat szerint — egy vektortér alapú modellhez jutunk.

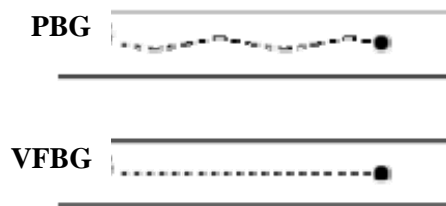


6. ábra. A vektortér alapú vezetés blokkvázlata

Az általánosított struktúrában az összes bemenet hatással van az összes kimenetre. Tehát strukturális szinten a PBG-től csak az összeköttetések számában különböznek (6. ábra). Az ilyen általánosabb modellt vektortér alapú vezetésnek⁶ nevezzük.

⁶ VFBG (Vector Field Based Guiding) — vektortér alapú vezetés.
162

Mint a 7. ábrán megfigyelhetjük, a vektortér alapú vezetés kiküszöböli a PBG bizonytalan, oszcilláló mozgását, mert képes a falakkal párhuzamos vektort eredményezni.



7. ábra. A PBG és VFBG útvonala egy szűk folyosón

A VEKTORTÉR ALAPÚ VEZETÉS NEURÁLIS HÁLÓVAL TÖRTÉNŐ MEGVALÓSÍTÁSA

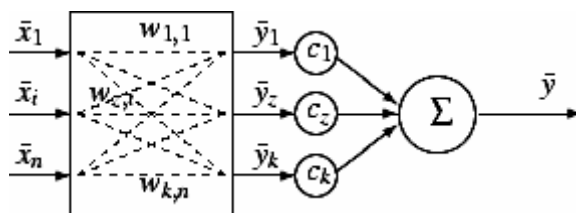
Ebben a részben bemutatjuk az általánosított neurális hálót, melynek segítségével egyszerűen modellezhetjük a VFBG-t. A 6. ábra egy kétszintű általánosított neurális hálót ábrázol. A háló olyan értelemben általánosított, hogy nem a neuronokban vannak az átviteli függvények, hanem az összeköttetéseken. Egy neuron kimenete:

$$\bar{y}_z = \bar{e}_z \sum w_{z,i}(x_i) \quad (5)$$

A második blokk egy harmadikhoz kapcsolódik, mely a neuronok kimenetét összegzi, létrehozva az algoritmus kimenetét:

$$\bar{y} = \sum_{z=1}^k \bar{y}_z = \sum_{z=1}^k \left(\bar{e}_z \sum_{i=1}^n w_{z,i}(x_i) \right) \quad (6)$$

Szükség lehet arra, hogy egyes bemenetek hatása erőteljesebben érződjön. Például az előrefelé irányuló szenzorok fontosabbak, mint a hátsók. Ezzel az általános alakkal könnyen megtehetjük ezt a változtatást, például a megfelelő \bar{y}_z vektorok c_z állandókkal történő megszorzásával. Az így kapott hibrid neurális háló (8. ábra) egy nemlineáris és egy lineáris réteget tartalmaz.



8. ábra. Általánosított neurális háló

A hibrid háló kimenete:

$$\bar{y} = \sum_{z=1}^k \left(c_z \cdot \bar{e}_z \cdot \sum_{i=1}^n w_{z,i}(x_i) \right) \quad (7)$$

Alkalmazzuk az ismertett fuzzy közelítést a $w_{z,i}(x_i)$ átviteli függvényekre. Így egy neuro-fuzzy rendszerhez (Brown and Harris, 1994) jutunk, melynek kimenete:

$$\bar{y}_z = \sum_{z=1}^k c_z \sum_{i=1}^n \sum_{j=1}^m \mu_{A_{i,j}}(x_i) \cdot b_{z,i,j} \quad (8)$$

AZ ÁLTALÁNOSÍTOTT HÁLÓ SZÁMÍTÁSI KOMPLEXITÁSA

Az algoritmus (8) számításigényét a szorzások számával jellemezzük, az összeadások számításigényét elhanyagoljuk.

1. *Állítás:* (8) számításigénye:

$$P_c = n \cdot k \cdot m + P_\mu \quad (9)$$

ahol: P_μ — a tagsági függvények kiszámításához szükséges számítási kapacitás

2. *Állítás:* (8) átalakítható az alábbi formába:

$$\bar{y}_z = \sum_{z=1}^{k^r} c_z \sum_{i=1}^n \sum_{j=1}^{m^r} \mu_{A_{i,j}^*}(x_i) b_{z,i,j}^r \quad (10),$$

ahol: r — mint redukált.

Mivel (10) számításigénye nyilván nem több, mint a (8) egyenleté, az átalakítással csökkenteni tudtuk a szükséges kapacitást.

BIZONYÍTÁS: A (8) transzformálásához alkalmazzunk egy, a szinguláris értékfelbontáson alapuló eljárást⁷. Ez garantáltan optimális eredményre vezet.

PÉLDÁK

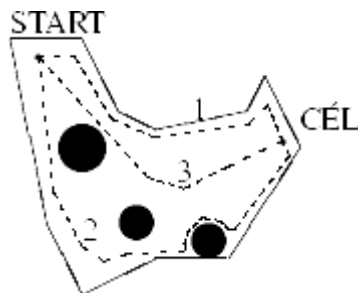
Az ismertett neuro-fuzzy eljárás hatékonyságának bemutatásához különböző PBG alapú vezetési stílusokat választottunk a szakirodalomból, és ezeket egy „tanár” robotban alkalmaztuk. Egy egyszerű tanítási algoritmust valósítottunk

⁷ Yam et al., 1999.

meg a „tanuló” robotban. A tanulás az általánosított neurális háló alapján történt, az erősen nemlineáris függvényeket lineárisokkal közelítettük. Az emberi irányítás alapstílusai a következők voltak:

- a lehető leginkább a bal oldalon haladni;
- a jobb oldalon haladni;
- mindentől a lehető legtávolabb haladni.

A 9. ábra három különbözőképpen tanított „tanuló” robotot mutat be egy számukra ismeretlen terepen.



9. ábra. Különböző stílusok

A háromdimenziós működés vizsgálata érdekében egy háromdimenziós kezelőfelületet hoztunk létre, melynek segítségével az algoritmus működése nyomonkövethető és tesztelhető.

ÖSSZEFOGLALÁS

Cikkünkben a potenciál alapú vezetés egy lehetséges továbbfejlesztését mutattuk be. A kibővített modell kiküszöböli a PBG erősen oszcilláló mozgását. Egy egyszerűsített neuro-fuzzy modellt is mutattunk az általánosított modell megvalósításához. A neurális hálós és fuzzy logikai alkalmazások mind nagy számítási igényekkel rendelkeznek. Ezért ismertettünk egy komplexitáscsökkentő eljárást a neuro-fuzzy algoritmushoz. Példákat mutattunk be a hatékonyság igazolására.

FELHASZNÁLT IRODALOM

- [1] Arkin, R. C.: Motor schema-based mobile robot navigation. *International Journal of Robotics Research* 8, 1989. p. 92-112.
- [2] Bronstein, J. – Y. Koren: Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man and Cybernetics* 19, 1989. p. 1179-1187.
- [3] Bronstein, J. – Y. Koren: The vector field histogram — fast obstacle avoidance for mobile robots. *IEEE Jour. Robotics and Automation* 7, 1991. p. 278-288.
- [4] Brooks, R. A.: *Intelligence without reason*. Massachusetts Institute of Technology, A.I. Memo No. 1293, 1991.

- [5] Brown, M. – C. Harris: Neurofuzzy Adaptive Modeling and Control. Prentice-Hall. New York, 1994.
- [6] Fukuda, T. – F. Kobayashi: Adaptation, learning and evolution for intelligent systems. IEEE Int. Fuzzy Systems Conf. Proc. 3, 1999. p. 1203-1210.
- [7] Kóczy, L.T. – K. Hirota: Fuzzy interference by compact rules. Proc. of Int. Conf. on FL & NN, 1990.
- [8] Kóczy, L.T. – K. Hirota: Size reduction by interpolation in fuzzy rule bases. IEEE Tr. S.M.C. 27, 1997. p. 14-25.
- [9] Kubota, N. – T. Morioka – F. Kojima – T. Fukuda: Sensory network for fuzzy controller of a mobile robot. IEEE Int. Fuzzy Systems Conf. Proc. 2, 1999. p. 986-991.
- [10] Lammens, J. – H. Hexmoor – S. Shapiro: Of elephants and men. In Proceedings of the NATO-ASI on the Biology and Technology of Intelligent Autonomous Agents, 1993.
- [11] Latombe, J.: Robot Motion Planning. Kluwer Academic Publishers, 1991.
- [12] Mizumoto, M.: Fuzzy controls by product-sum-gravity method. In: Advancement of Fuzzy Theory and Systems in China and Japan (Liu and Mizomoto, Eds.). Chap. 1.1-1.4, International Academic Publishers, 1990.
- [13] Piaggio, M. – R. Zaccaria: An autonomous system for vehicle navigating in a partially or totally unknown environment. Proc. Int. Workshop on Mechatronical Computer Systems for Perception and Action, MCPA, 1997.
- [14] Yam, Y.– P. Baranyi – C. T. Yang: Reduction of fuzzy rule base via singular value decomposition. IEEE Trans. on Fuzzy Systems 7, 1999. p. 120-132.