



Navigácia mobilného robota v neznámom prostredí s využitím fuzzy logiky

V príspevku je opísaný reaktívny navigačný algoritmus na navigáciu mobilného robota v neznámom prostredí s prekážkami. Inferenčný mechanizmus využíva bázu fuzzy pravidiel Mamdaniho typu. Zvlášť je opísaný princíp navigácie v značne uzavretých (dutých) priestoroch, ktoré sa v reálnych prostrediach často vyskytujú.

Úvod

Pojem fuzzy logika je od začiatku 90. rokov veľmi populárny. Pojem „Fuzzy Logic“ označuje istú teóriu „nejasných, neostrých množín“. Vo vede a v počítačových procesoroch sa obvykle pracuje iba s dvoma hodnotami: pravda alebo nepravda, resp. 1 alebo 0. Medzistupne miery príslušnosti do danej množiny v tomto prípade neexistujú. Prakticky všetky tradičné metódy spracovania informácií sú založené na binárnom princípe. Tento binárny princíp síce zjednodušuje a uľahčuje aplikáciu matematiky a logiky v počítačovom spracovaní, ale na druhej strane komplikuje možnosti spracovania vágne definovaných veličín.

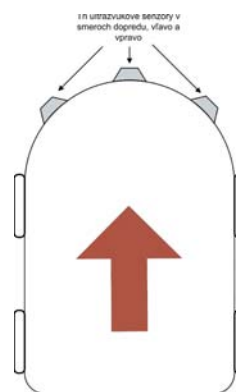
Základná myšlienka fuzzy prístupu spočíva v manipulácii s „neostrými množinami“, teda s množinami, ktorých elementy patria, resp. nepatria graduálne do danej množiny, teda ich príslušnosť do danej množiny je vyjadriteľná spojenou funkciou s oborom definície z množiny reálnych čísel a oborom hodnôt z intervalu $< 0,1 >$. Hodnota 0 značí, že hodnota danej veličiny do množiny určite nepatrí, kým hodnota 1 značí, že do množiny určite patrí. Hodnoty funkcie príslušnosti z intervalu $(0, 1)$ vyjadrujú teda stupeň príslušnosti. Nemajú však nič spoločné s pravdepodobnosťou, či daný prvok do množiny patrí, resp. nepatrí. Pomocou fuzzy logiky možno vágne definované veličiny matematicky vyjadriť a ďalej spracovať.

Lingvistické veličiny výstupného priestoru sú expresívnejšie vo vyjadrovaní vágnosti či presnosti, ako veličiny klasickej dvojhodnotovej či viachodnotovej logiky. Napr. teplota telesa, ktorá je zvyčajne udaná v stupňoch, by mohla byť v prostredí fuzzy množín (lingvistických premenných) vyjadrená ako studené, vlažné, teplé či horúce telo. Fuzzy IF – THEN pravidlá, teda pravidlá, v ktorých je premisná aj konzekventná časť vyjadrená cez fuzzy množiny, sa takto stávajú veľmi komfortným prostriedkom na zobrazenie jednoznačných hodnôt veličín zo vstupného priestoru (napr. hodnôt veličín na výstupoch snímačov) do výstupného priestoru ich lingvistického vyjadrenia. Expresívnosť fuzzy logiky je značne podobná „každodennej bežnej logike vyjadrovania“ (common sense), čím sa stáva prirodzeným nástrojom formálneho spracovania bežnej ľudskej komunikácie. Napr. fuzzy opis správania (dynamiky) objektov je ľahko generovateľný expertom z danej vednej či technickej oblasti a ďalej spracovateľný vo fuzzy procesore.

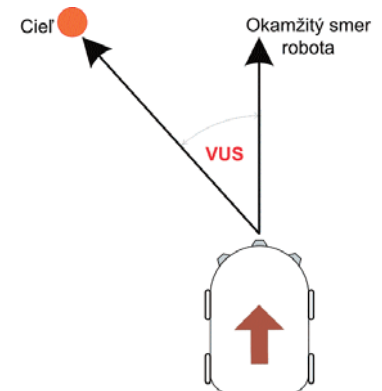
Ako bolo spomenuté, transformácia „exaktnej“ hodnoty na lingvistickú sa realizuje prostredníctvom funkcie príslušnosti, (z angl. Membership Function – MF). Táto transformácia sa nazýva fuzzyfikácia. V procese navigácie sa fuzzyfikované hodnoty vstupných veličín spracujú pomocou navrhnutej množiny fuzzy pravidiel. Následne sa defuzzyfikujú, čím opäť vzniká príslušná numerická hodnota výstupnej veličiny navigátora. Podrobnosti tohto procesu nebudeme opisovať, nakoľko sú všeobecne známe. Poznamenajme len, že v experimentoch boli použité pravidlá Mamdaniho typu a defuzzyfikácia sa realizovala metódou výpočtu ťažiska výstupných MF.

Senzorický systém robota

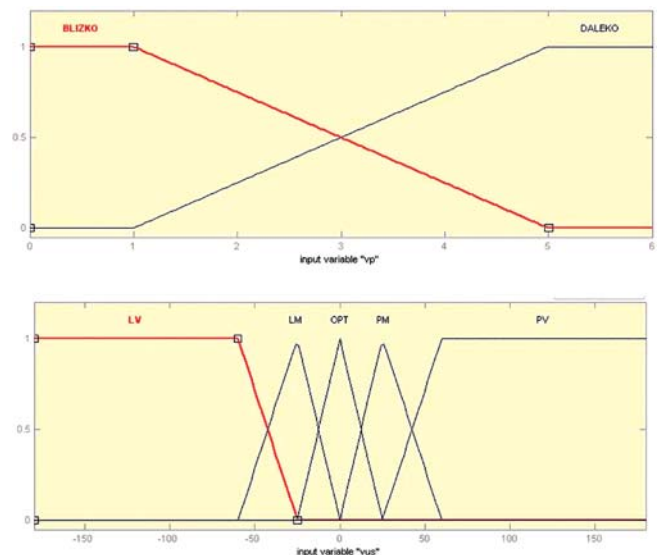
Ide o reaktívny spôsob navigácie, to znamená, že robot disponuje informáciami len o jeho začiatočnej polohe a o polohe cieľa, ale mapu prostredia nemá k dispozícii. Svoj pohyb v neznámom prostredí riadi len na základe informácie o prekážkach v jeho blízkom okolí. Robot je vybavený senzorovým systémom pozostávajúcím z 3 ultrazvukových snímačov prítomnosti prekážky (obr. 1). Sensory sú rozmiestnené na snímanie v troch smeroch – pred, vľavo a vpravo. Schéma rozmiestnenia snímačov na konštrukcii robota je zrejme z obrázka. Informácie zo



Obr.1 Senzorický systém robota



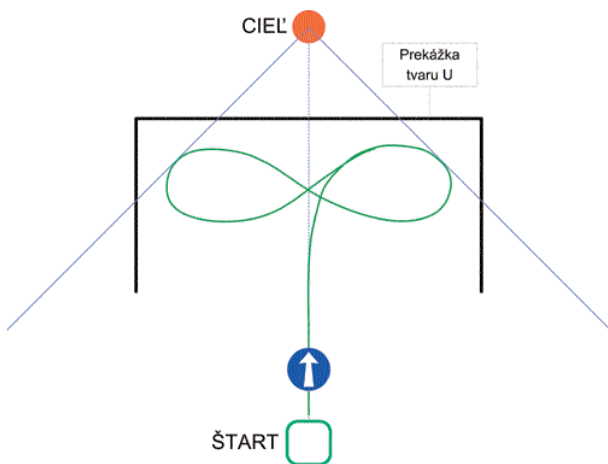
Obr.2 Definícia virtuálneho uhla smerovania



Obr.3 Nadefinovanie kritérií pre vstupné premenné vo FIS editore

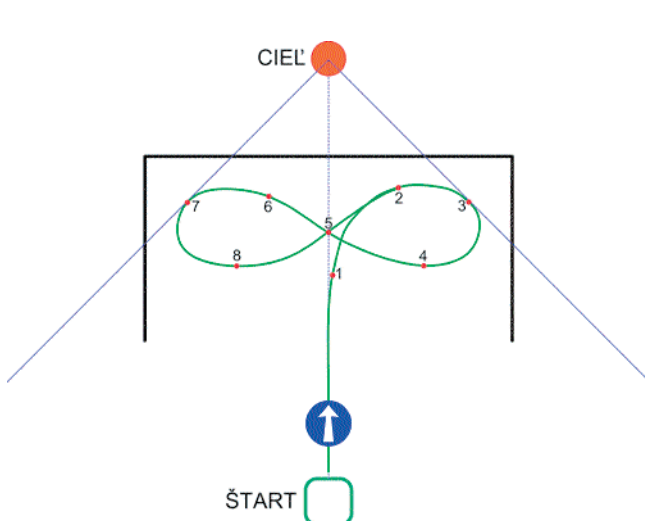
senzorov o vzdialenosti prekážok v jednotlivých smeroch vs, vl, vp a virtuálny uhol smerovania vus slúžia ako vstupy do navigátora. Virtuálny uhol smerovania – vus je definovaný ako uhol medzi okamžitým smerom pohybu robota a spojnicou robota s cieľom (obr. 2). Výstupnými premennými z navigátora je riadiaci uhol ru (uhol, o ktorý sa má robot pootočiť) a zr (zrýchlenie, resp. spomalenie pohybu robota). Vstupné hodnoty sú klasifikované lingvistickými veličinami BLÍZKO a ĎALEKO. Premenná vus môže nadobúdať 5 hodnôt: LV, LM, OPT, PM, PV (viac vľavo, menej vľavo, optimálne, menej vpravo, viac vpravo). Implementácia vstupných premenných zo senzorov do fuzzy reaktívneho navigátora sa realizuje prostredníctvom FIS editora v programovom prostredí Matlab (obr. 3).

Cestu robota za cieľom môže skomplikovať jeho uviaznutie v lokálnej prekážke. Takéto prekážky majú zväčša tvar U alebo iných pomerne uzavretých ohraničení. Príčina uviaznutia je v tom, že robot registruje len prekážky v jeho relatívne blízkom okolí a potrebuje realizovať istú sériu meraní, aby takúto dutú prekážku identifikoval. Snažiac sa dostať k cieľu začne oscilovať v prekážke, čo nakoniec identifikuje ako znak uviaznutia (obr. 4). Problém sa dá riešiť niekoľkými spôsobmi, napr. definovaním a prepínaním virtuálnych cieľov. Princíp funguje tak, že ak ro-



Obr.4 Demonstrácia uviaznutia robota v lokálnej prekážke

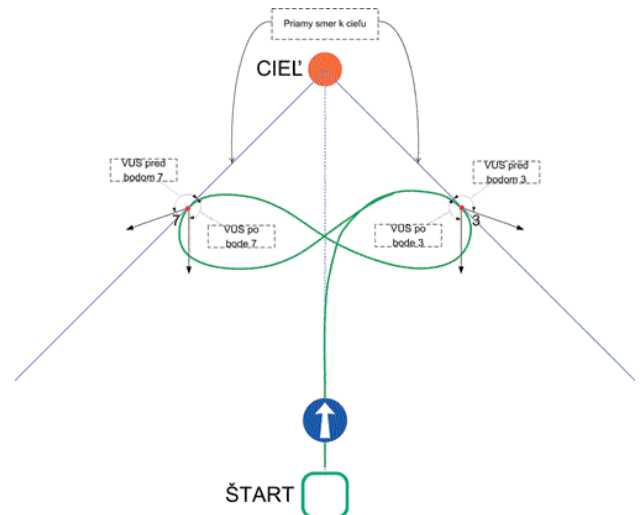
bot „zistí“ (opísané ďalej), že uviazol v lokálnej prekážke, vygeneruje si virtuálny cieľ, ktorému v danej chvíli priradí najvyššiu prioritu a iniciuje riadenie vedúce k tomuto cieľu. Keď sa úspešne dostane z nej von, cieľ prepne na pôvodný – skutočný cieľ a napreduje k nemu. Iným riešením toho problému môže byť stratégia sledovania steny. Aj v tomto prípade však musí robot najprv identifikovať uviaznutie v lokálnej prekážke a potom zmeniť svoje správanie a začne sa pohybovať pozdĺž steny až dovtedy, kým sa nedostane von z prekážky. Následne sa správanie ro-



Obr.5 Oscilovanie robota v lokálnej prekážke

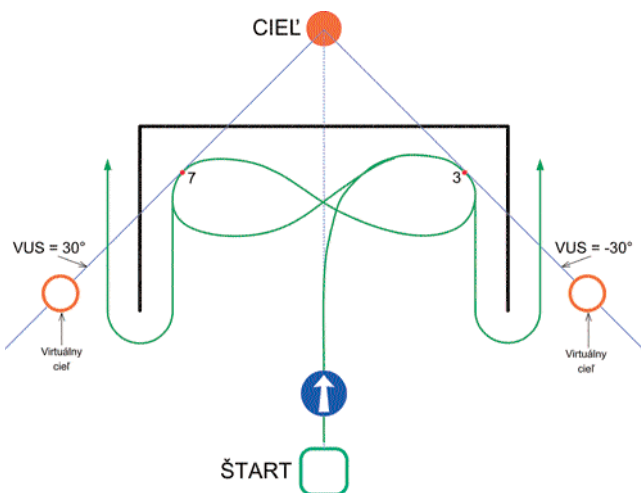
bota zase zmení na smerovanie za cieľom. Riešenie navrhnuté v tomto článku kombinuje stratégiu prepínania virtuálnych cieľov a stratégiu sledovania steny.

Keďže navigátor je postavený len na báze fuzzy logiky, nemá schopnosť učiť sa, t. j. generovať nové pravidlá alebo vhodne prispôbovať tie pôvodné poznatkom akumulovaným do daného okamihu. Na to by bolo potrebné do navigátora implementovať učiaci sa mechanizmus, napr. vo forme neurónovej siete. Keďže súčasný navigačný systém robota je akoby bez pamäte, rieši len súčasný problém, teda na navigáciu v danom momente nevyužíva žiadne predchádzajúce skúsenosti. Napriek výlučne reaktívnej podstate navigácie možno vhodným výberom vstupných signálov a bázy pravidiel dosiahnuť prijateľné výsledky. Ďalej je opísaný princíp rozpoznania a navigácie v prekážke tvaru U, čo je častý problém čisto reaktívnych prístupov. Ako bolo naznačené, robot po vstupe do takejto prekážky opisuje dráhu v tvare číslice osem, ktorú možno rozdeliť na niekoľko segmentov znázornených na obr. 5. V dôsledku atraktívnej sily cieľa a súčasne pôsobiacej repulzívnej sily prekážky začne navigačný systém navádzať robot po dráhe 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 5 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 5 – ..., čiže začne v prekážke oscilovať a nevie sa samostatne dostať von z prekážky. Problém je v tom, že keď je prekážka ešte mimo dosahu snímačov, robot ešte nevie identifikovať, že sa nachádza v značne uzavretom priestore a snaží sa napredovať k prekážke. Len čo ju deteguje, snaží sa ju obísť, preto sa zatáča vľavo alebo vpravo, až sa dostane do bodu 3, resp. 7. V tejto situácii hrajú body 3 a 7 rozhodujúcu úlohu a sú významné v nasledujúcom zmysle: V bode 3 leží vektor okamžitého pohybu robota na jednej priamke s priamym vektorom k cieľu, ale majú navzájom opačnú orientáciu. Tento prípad nastáva aj v bode 7. V týchto kritických bodoch 3 a 7 sa robot otáča a smeruje naspäť do vnútorného priestoru prekážky. V bodoch 3 a 7 môže robot zistiť, že sa nachádza v prekážke. Ak totiž viackrát prejde cez kritické body, postupne pomocou stavu počítadla prechodov deteguje, že v prekážke uviazol. Preto keď je robot v kritickom bode, bolo by potrebné zmeniť jeho smerovanie k cieľu. Problémom sa môže stať identifikácia, kedy sa práve robot v takomto kritickom bode nachádza. Na určenie kritických bodov možno využiť nasledujúci princíp: Keď sa robot blíži ku kritickému bodu 3, cieľ je po jeho ľavej strane, to znamená, že virtuálny uhol smerovania vus je blízky hodnote -180° . Po dosiahnutí kritického bodu 3 vus nadobudne kladnú hodnotu blízku $+180^\circ$ - cieľ je už na jeho pravej strane, pretože bezprostredne po prejení kritického bodu 3 sa robot začne otáčať smerom doprava. Týmto spôsobom môže navigačný systém robota určiť kritické body. V druhom kritickom bode – v bode číslo 7 je situácia podobná. Keď sa robot k nemu po vyznačenej trajektórii blíži, tak cieľ má na pravej strane, t. j. vus má hodnotu z úzkeho intervalu okolo $+180^\circ$. Po dosiahnutí kritického bodu 7 sa robot zas začne otáčať doprava a vus nadobudne zápornú hodnotu blízku -180° – cieľ má tentoraz na ľavej strane. Toto potom znamená, že robot sa nachádza v kri-



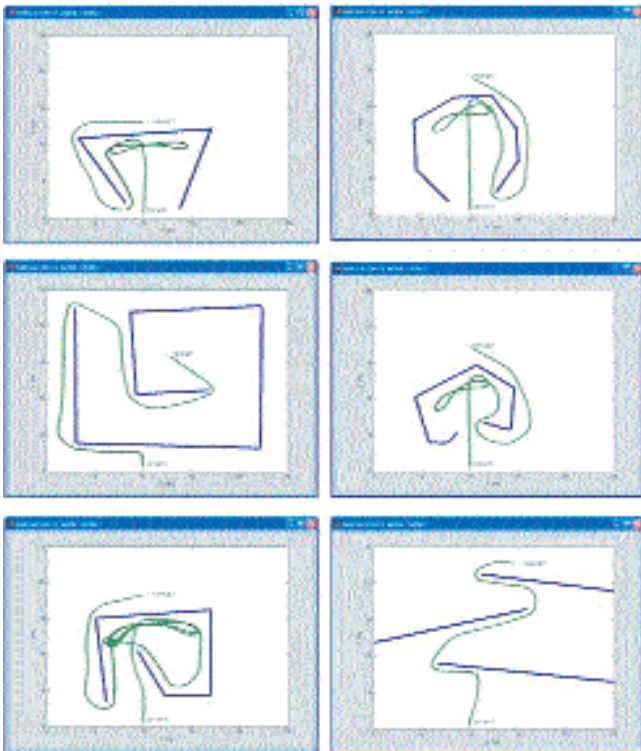
Obr.6 Znázornenie virtuálnych uhlov smerovania (vus) pred prejením kritických bodov a po ňom s cieľom ich definovania

tickom bode 7. Tento problém je demonštrovaný na obr. 6. Po detekcii skutočnosti, že robot blúdi v lokálnej prekážke a prechádza práve kritickým bodom, virtuálny uhol smerovania sa modifikuje k novému virtuálnemu cieľu na hodnotu $vus = -30^\circ$ v bode 3 a na hodnotu 30° v bode 7 (obr. 7). Daný virtuálny cieľ má v tejto chvíli vyššiu prioritu ako skutočný cieľ. Od tohto momentu začne robot sledovať stenu prekážky. Teda robot sa teraz naviguje pomocou stratégie sledovania steny. Hodnota $vus = -30^\circ$, resp. $vus = 30^\circ$ je určená na základe rovnováhy príťažlivých síl od cieľa a odpudivých síl od prekážky. Efekt je taký, že robot kopíruje stenu. Toto správanie navigačného systému trvá dovtedy, kým ľavý senzor deteguje prekážku na úrovni blízko pre prípad s bodom 3, alebo pravý senzor registruje prekážku na úrovni blízko pre prípad bodu 7. Potom sa virtuálny uhol smerovania prepne naspäť na pôvodný skutočný cieľ. Tento prepínací mechanizmus je hierarchicky nadradený a navigátor robota sa prepína iba vtedy, keď začne robot blúdiť v neznámom prostredí, resp. prekážke.



Obr.7 Generovanie virtuálnych cieľov

Verifikácia prístupu k danej problematike riadiaceho algoritmu bola vykonaná v prostredí Matlab prostredníctvom simulácií. Po spustení programu možno nadefinovať počet aj tvar prekážok a polohu cieľa. Virtuálny robot sa počas simulácie snaží vyhýbať prekážkam a čo najrýchlejšie sa dostať k cieľu. Robot môže meniť svoju rýchlosť vzhľadom



Obr.8 Vybrané výsledky simulácie

dom na to, ako sú okolo neho rozmiestnené prekážky. Bolo zvolené maximálne posunutie robota v jednom kroku 0,3 m a minimálne posunutie 0,05 m. Na obr. 8 sú uvedené výsledky niektorých simulácií.

Záver

Opísali sme reaktívny navigačný algoritmus na navigáciu mobilného robota v neznámom prostredí s prekážkami. Prednosťou použitého prístupu je jeho relatívna jednoduchosť, vysoká efektívnosť a odolnosť výsledného algoritmu. Robot sa vo väčšine prípadov dokáže elegantným spôsobom naviesť k cieľu. Simulácie ukázali, že prípady, keď to nedokáže, predstavujú len extrémne zložité prostredia. Vždy je to však len otázka vhodnej definície množiny navigačných pravidiel. Väčšinou stačí množinu pravidiel vhodne rozšíriť a problém je odstránený. Navigátor je výlučne reaktívny a založený na báze fuzzy logiky. Znamená to, že na rozhodovanie využíva len aktuálne vstupy zo senzorov snímajúcich najbližšie okolie robota. Efektívnosť navigačného systému možno zvýšiť implementáciou neurónovej siete, ktorá zabezpečí schopnosť učiť sa z predchádzajúcich situácií a pohyb robota optimalizovať s ohľadom na zvolené kritérium. Návrh bol zatiaľ verifikovaný len v simulačnom prostredí. Implementácia navigačného algoritmu do reálneho robota prebieha v súčasnosti.

Problematika je riešená s podporou grantu VEGA 1/3120/06.

Literatúra

- [1] Qi Chen – Ozguner, O.: Real-time navigation for autonomous vehicles: a fuzzy obstacle avoidance and goal approach algorithm. American Control Conference, 2005. Proceedings of the 2005 Volume, Issue, 8 – 10 June 2005, p. 2 153 – 2 158, vol. 3.
- [2] Marley Maria B. R. Vellaso et al.: Mobile Robot Control Using Fuzzy Logic http://www.ica.ele.puc-rio.br/publicacoes/download/cnf_0113.pdf
- [3] Cuesta, F. – Ollero, A.: Intelligent Mobile Robot Navigation. Springer-Verlag Berlin Heidelberg 2005. ISBN 3-540-23956-1.
- [4] Probert Smith, P.: Active Sensors for Local Planning in Mobile Robotics. World Scientific Publishing, Co. 2001. ISBN 981-02-4681-1.
- [5] Xu, W. L. – Tso, S. K. – Fung, Y. H.: Sensor-based reactive navigation of a mobile robot through local target switching. Advanced Robotics, 1997. ICAR apos; 97. Proceedings., 8th International Conference on Volume, Issue, 7 – 9 July 1997, p. 361 – 366.
- [6] Borenstein, J. – Everett, H. R. – Feng, L.: Where I am? Sensors and Methods for Mobile Robot Positioning. The Univ. of Michiqan, 1996.

Ing. Mark Hemedi
doc. Ing. Anton Vitko, PhD.
prof. Ing. Ladislav Jurišica, PhD.

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ústav riadenia a priemyselnej informatiky
Ilkovičova 3, 812 19 Bratislava