

MOBILE ROBOT MOTION CONTROL BY MEANS OF FUZZY RULES

P. Krček*, J. Dvořák*

Summary: *This paper deals with an application of a fuzzy expert system to autonomous mobile robot control. In the first part, we describe the process of developing a fuzzy expert system, consisting of introducing input and output variables, defining linguistic values and the corresponding fuzzy sets, designing fuzzy rules, choosing an inference engine and defuzzification method. The next part provides an example of applying this fuzzy expert system to solving a partial task of local robot navigation, which involves reaching a given goal without collision with unknown obstacles. For simplicity, we assume only convex obstacles the distances of which are sufficient for the robot to pass. Finally, we outline the possibilities of a further development of the implemented system.*

1. Úvod

Řízení pohybu autonomního mobilního robotu zajišťuje jeho řídicí systém. Velmi často se řídicí systém rozkládá do dvou úrovní. Vyšší úroveň představuje *deduktivní řídicí systém*, nižší pak *reaktivní řídicí systém*.

Deduktivní řídicí systém plánuje optimální dráhu robotu z počátečního bodu do koncového formou posloupnosti kroků na základě dané mapy prostředí. Činnost tohoto systému se obvykle nazývá *globální navigace*. Existuje řada sofistikovaných metod plánování dráhy robotu. Jejich přehled je možno najít např. v práci (Doyle, 1995). Řízení pohybu robotu pouze pomocí deduktivního řídicího systému je možné pouze v případě, že prostředí je plně známé a statické. V případě pouze částečně známého nebo dynamického prostředí je nutno naplánovanou dráhu předat ve formě posloupnosti podcílů reaktivnímu řídicímu systému, jehož úkolem je dosahovat zadané podcíle bez kolize s neznámými překážkami.

Činnost reaktivního řídicího systému je známa pod pojmem *lokální navigace*. Tento systém řídí pohyb robotu na základě zadaných dílčích cílů a periodicky dodávaných dat ze senzorů. Periodicky dodávaná data o okolním prostředí zaručují jejich aktuálnost a robot tedy může reagovat na překážky, které nebyly známé v době výpočtu dráhy deduktivním systémem řízení. Reaktivní řídicí systém je schopen řídit pohyb robotu i v případě, že prostředí je zcela neznámé a deduktivní systém tudíž není schopen vygenerovat žádný plán cesty. V tomto článku se věnujeme právě problematice lokální navigace.

* Ing. Petr Krček, RNDr. Jiří Dvořák, CSc.: Ústav automatizace a informatiky, Fakulta strojního inženýrství, Vysoké učení technické v Brně; Technická 2; 616 69 Brno; e-mail: petr.krcek@seznam.cz

Lokální navigace robotu v neznámém nebo pouze částečně známém a dynamickém prostředí je složitý a náročný úkol. Klasické metody řízení se v těchto podmínkách nemusí vždy dobře vyrovnávat s takovými požadavky kladenými na řídicí systém jako je řízení v reálném čase, robustnost a schopnost zpracovávat neurčitost. Z tohoto důvodu se v poslední době při řízení složitých systémů stále více prosazují *inteligentní metody řízení* (viz např. Tzafestas, 1997; Zilouchian & Jamshidi, 2001), k nimž patří zejména *fuzzy logika*, *neuronové sítě* a *evoluční metody*. Kromě výše jmenovaných základních inteligentních metod řízení se používají i hybridní metody kombinující více přístupů. Např. činnost fuzzy řídicího systému je možno zdokonalit pomocí neuronových sítí, genetických algoritmů či genetického programování (Akbarzadeh-T et al., 2000). Jedním z motivů a důvodů pro používání inteligentních metod řízení je skutečnost, že lidé jsou schopni úspěšně realizovat řídicí strategie ve složitých dynamických podmínkách (např. při řízení auta), aniž by k tomu potřebovali nějaké exaktní matematické modely a metody jejich řešení.

Reaktivní řídicí systém robotu lze s úspěchem založit na *fuzzy expertním systému*. Expertní systém poskytuje řešení daného problému pomocí tzv. inferenčního mechanismu na základě zadaných vstupních údajů a znalostí uložených ve znalostní bázi. Fuzzy systémy jsou charakteristické používáním fuzzy logiky a slovních hodnot reprezentovaných pomocí fuzzy množin. Hlavní přínos fuzzy expertních systémů je v netradičním zpracování neurčitosti, které překonává nedostatky klasických expertních systémů založených na dvouhodnotové logice a pravděpodobnostním přístupu. Fuzzy systémy umožňují zpracovávat jak nepřesné vstupní údaje, tak také používat vágně definované znalosti ve tvaru pravidel typu IF-THEN.

Lokální navigaci robotu pomocí fuzzy expertního systému používají např. Aguirre & González (2000) a Maaref & Barret (2000). V práci (Maaref & Barret, 2000) je lokální navigace v neznámém prostředí založena na kombinaci elementárních chování (udržování se ve středu volného prostoru, hledání cíle, sledování stěny). V případě částečně známého prostředí je použita metoda koordinující globální a lokální navigační strategii. Aguirre & González (2000) navrhují třívrstvý řídicí systém. Nejvyšší je plánovací vrstva, která představuje deduktivní řídicí systém a produkuje abstraktní plán cesty jako jazykový popis cesty z počátečního bodu do koncového (např. následuj zeď, projdi dveřmi, zahni vlevo, jdi chodbou, ...). Další dvě vrstvy vytvářejí reaktivní řídicí systém. Nejnižší vrstva se skládá z několika dílčích chování zajišťujících různé řídicí strategie. Střední vrstvou je vrstva výkonná, jež rozhoduje o tom, která z těchto strategií bude v řízení uplatněna. Rozhodování výkonné vrstvy je založeno na fuzzy metaprávidlech (např. *IF je překážka THEN objed' překážku*; *IF není překážka AND blízko je stěna THEN jed' podél stěny*) a aktuálních informací o okolním prostředí. Metaprávidla si výkonná vrstva vytváří podle plánu z plánovací vrstvy.

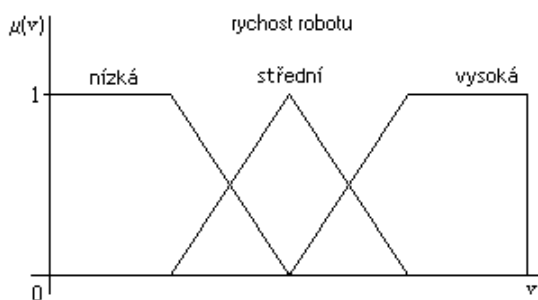
2. Charakteristika fuzzy expertního systému

Fuzzy expertní systém se skládá z báze znalostí, báze dat, bloku vyhodnocování a také z bloků fuzzifikace vstupních údajů a defuzzifikace výstupních fuzzy množin. Ve znalostní bázi jsou reprezentovány znalosti expertního systému formou slovních pravidel typu IF-THEN. Bázi dat tvoří data, potřebná k definici funkcí příslušnosti slovních hodnot slovních proměnných a ke stanovení měr příslušnosti vstupních dat k těmto jazykovým hodnotám. Blok vyhodnocování tvoří jádro expertního systému a provádí odvození fuzzy závěru s využitím některého z fuzzy inferenčních mechanismů. Fuzzifikace vstupů zohledňuje nepřesnosti vstupních senzorů a defuzzifikace poskytuje potřebné ostré (*crisp*) výstupní

hodnoty. Proces tvorby fuzzy expertního systému sestává ze zavedení vstupních a výstupních proměnných, definice slovních hodnot a odpovídajících fuzzy množin, návrhu fuzzy pravidel, výběru inferenčního mechanismu a defuzzifikační metody.

Při aplikaci fuzzy expertního systému v reaktivním řídicím systému robotu se na vstupní proměnné přivádějí informace o okolním prostředí robotu (vzdálenost od cíle nebo od překážky, úhel natočení robotu vzhledem k překážce nebo k cíli) a hodnoty výstupních proměnných (rychlost nebo zrychlení, úhel natočení kol) se přivádějí na akční členy robotu.

Přiblížit běžné lidské vyjadřování k matematickému modelu nám pomáhá slovní proměnná. Slovní proměnná může totiž nabývat slovních hodnot a to umožňuje vytvořit matematický model velmi blízký slovnímu popisu a tím pracovat s vágními informacemi. Každá slovní hodnota je reprezentována fuzzy množinou definovanou na univerzu slovní proměnné. Nejčastěji se pro reprezentaci slovních hodnot definovaných na intervalu reálných čísel používají trojúhelníková nebo lichoběžníková fuzzy čísla. Např. slovní proměnná *rychlost robotu* může nabývat tří slovních hodnot *nízká*, *střední* a *vysoká*, kterým odpovídají fuzzy čísla definovaná na intervalu reálných čísel (viz. obr. 1). Obvykle se požaduje, aby součet funkcí příslušnosti byl pro každý bod univerza roven 1.



Obr. 1. Slovní proměnná rychlost robotu

Znalostní báze fuzzy expertního systému obsahuje pravidla tvaru IF *antecedent* (předpoklad, podmínka) THEN *konsekvent* (závěr, důsledek). Předpoklad i závěr jsou obvykle vyjádřeny slovními proměnnými a jejich hodnotami:

$$\text{IF } x \text{ is } A \text{ THEN } y \text{ is } B \quad (1)$$

kde x je vstupní proměnná, y je výstupní proměnná, A je hodnota vstupní proměnné a B je hodnota výstupní proměnné. Pravidlo se chápe jako fuzzy implikace. Antecedent se většinou skládá z více dílčích předpokladů, které bývají spojeny spojkami AND nebo OR a mohou být také negovány pomocí NOT. Příkladem pravidla s několika dílčími podmínkami je pravidlo

$$\text{IF } \textit{rychlost_robotu} \text{ is } \textit{vysoká} \text{ AND } \textit{cil} \text{ is } \textit{blizko} \text{ THEN } \textit{rychlost_robotu} \text{ is } \textit{nizka} \quad (2)$$

Existuje řada způsobů, jak vyhodnocovat fuzzy logické operace. Nejčastěji se AND vyhodnocuje jako minimum a OR jako maximum. Implikace $\alpha \Rightarrow \beta$, kde $\alpha, \beta \in \langle 0; 1 \rangle$, bývá obvykle ve fuzzy řízení vyhodnocována pomocí *Mamdaniho implikační funkce* $I(\alpha, \beta) = \min(\alpha, \beta)$ nebo pomocí funkce $I(\alpha, \beta) = \alpha \cdot \beta$. Označování těchto funkcí jako „implikačních“ je ovšem chybné, neboť pro hodnoty $\alpha, \beta \in \{0; 1\}$ se nechovají jako klasická implikace. Tento nedostatek nemá např. *Lukasiewiczova implikační funkce* $I(\alpha, \beta) = \min(1, 1 - \alpha + \beta)$.

Pravidla ve znalostní bázi jsou také navzájem spojena spojkami (obvykle OR nebo AND) a dohromady vytvářejí tzv. slovní model.

Jako inferenční mechanismus se pro aplikace fuzzy expertních systémů v oblasti řízení obvykle používá Mamdaniho inferenční mechanismus. Tento mechanismus, který je použit i v námi vytvořeném systému, je založen na jednoúrovňové dopředné inferenci GMP (*general modus ponens*), implikaci interpretuje jako minimum a pro spojení pravidel používá spojku OR. Uvažujme případ, kdy máme n vstupních proměnných, jednu výstupní proměnnou a znalostní báze je tvořena m pravidly tvaru

$$\text{IF } x_1 \text{ is } A_1^i \text{ AND } \dots x_n \text{ is } A_n^i \text{ THEN } y \text{ is } B^i \quad (3)$$

kde $i = 1, \dots, m$. Potom po položení bodového dotazu (dosazení konkrétních ostrých hodnot x_j^0 za vstupní proměnné x_j) dostaneme odpověď ve tvaru fuzzy množiny definované na univerzu výstupní proměnné, jejíž funkce příslušnosti je dána vztahem

$$\mu_{R^*}(y) = \max_{1 \leq i \leq m} \min(\mu_{A_1^i}(x_1^0), \dots, \mu_{A_n^i}(x_n^0), \mu_{B^i}(y)) \quad (4)$$

Při řízení robotu je ovšem třeba zohlednit nepřesnost senzorů a tedy místo ostré hodnoty x_j^0 použít fuzzy hodnotu X_j^0 vytvořenou fuzzifikačním členem např. jako lichoběžníkové fuzzy číslo $(x_j^0 - \varepsilon_j, x_j^0, x_j^0, x_j^0 + \varepsilon_j)$, kde ε_j udává chybu j -tého senzoru. V případě fuzzy dotazu je odpověď fuzzy systému dána vztahem

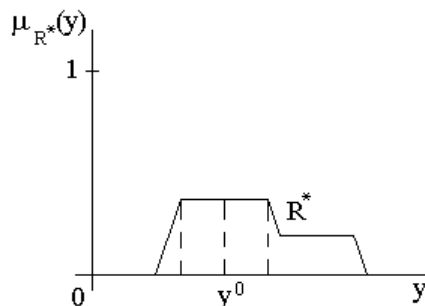
$$\mu_{R^*}(y) = \max_{1 \leq i \leq m} \min(\alpha_1^i, \alpha_2^i, \dots, \alpha_n^i, \mu_{B^i}(y)) \quad (5)$$

kde

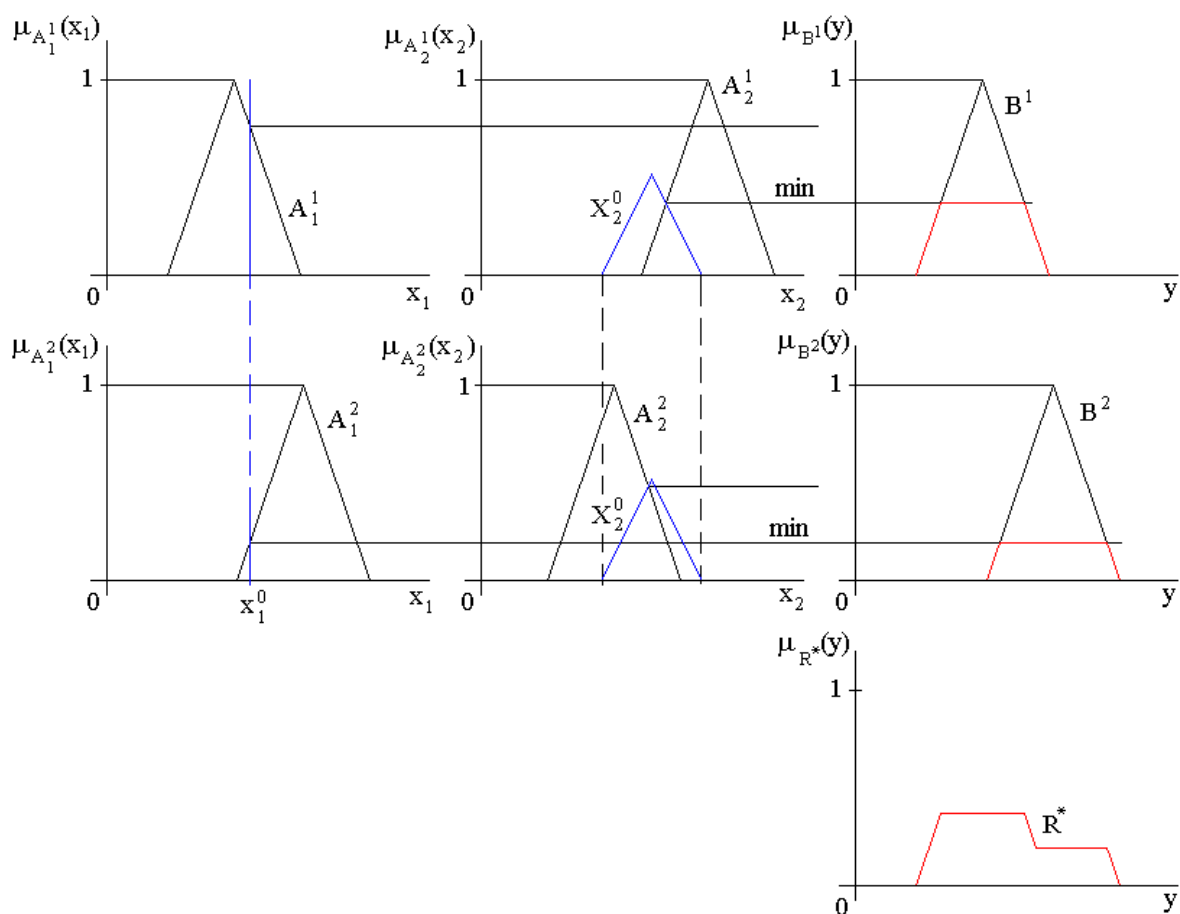
$$\alpha_j^i = \max_{x \in U_j} \min(\mu_{X_j^0}(x), \mu_{A_j^i}(x)) \quad (6)$$

Příklad činnosti inferenčního mechanismu pro dvě pravidla, jednu ostrou a jednu neostrou vstupní hodnotu ilustruje obrázek 3.

Konkrétní ostrá hodnota y^0 výstupní proměnné y se vypočte z průběhu funkce příslušnosti odpovědi $\mu_{R^*}(y)$ defuzzifikací. Nejčastěji používanými metodami defuzzifikace je metoda těžiště plochy pod funkcí příslušnosti a metoda střední hodnoty bodů globálního maxima (viz obr. 2), kterou používáme i v našem systému.



Obr. 2. Defuzzifikace metodou střední hodnoty bodů globálního maxima



Obr. 3. Mamdaniho inferenční mechanismus

3. Příklad pro aplikaci fuzzy expertního systému

Uvažujme tříkolový robot se zadní nápravou hnanou a předním kolem řízeným (viz. obr. 4). Na vstupní proměnné fuzzy expertního systému přivedeme informace o prostředí robotu a požadujeme, aby ze vstupních hodnot byly odvozeny dva závěry: nová rychlost robotu a natočení jeho řídicího kola. Hodnoty těchto dvou výstupních proměnných se budou přivádět na akční členy robotu. Vyvození závěrů se bude periodicky opakovat, takže náš fuzzy expertní systém bude řídit natáčení předního kola a ovlivňovat také aktuální rychlost robotu.

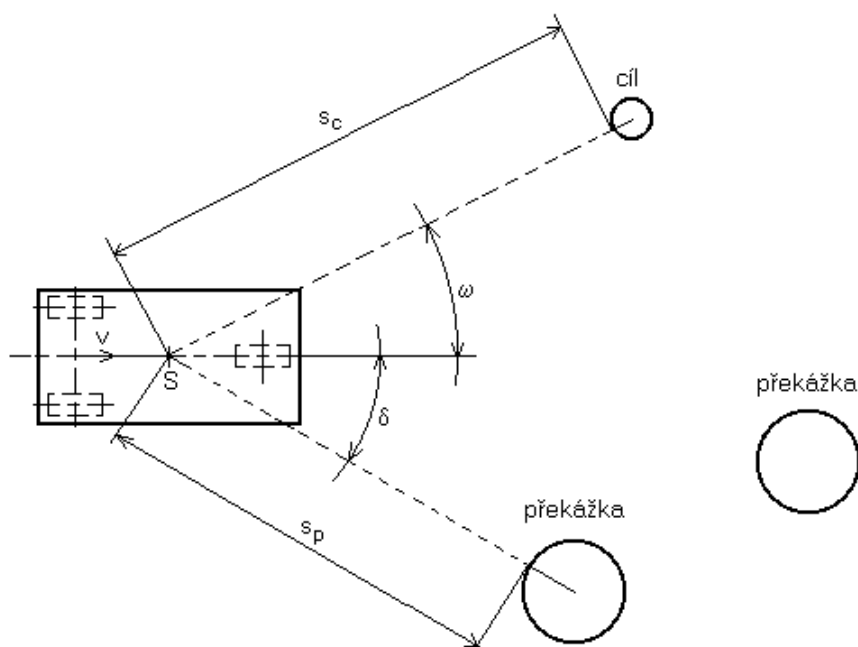
Po našem expertním systému požadujeme, aby zajistil dosažení daného cíle a přitom zamezil kolizím s neznámými překážkami. Pro zjednodušení problému budeme dále předpokládat, že se v prostředí robotu mohou vyskytovat pouze překážky, které mají konvexní tvar, přičemž vzdálenosti mezi nimi jsou pro průjezd robotu dostatečné. Tato zjednodušení byla volena proto, aby stačilo zavést pouze dvě vstupní proměnné charakterizující polohu robotu vůči nejbližší překážce. Další dvě proměnné pak charakterizují polohu robotu vzhledem k cíli. Veškeré vstupní a výstupní proměnné fuzzy expertního systému jsou uvedeny v tabulkách 1 a 2.

Tab. 1. Vstupní proměnné

Vstupní proměnná	Popis vstupní proměnné
x_1	vzdálenost s_c robotu od cíle
x_2	úhel ω natočení robotu vzhledem k cíli
x_3	vzdálenost s_p robotu od nejbližší překážky
x_4	úhel δ natočení robotu vzhledem k nejbližší překážce

Tab. 2. Výstupní proměnné

Výstupní proměnná	Popis výstupní proměnné
y_1	rychlost v robotu
y_2	úhel α natočení předního kola robotu



Obr. 4. Uvažovaný robot

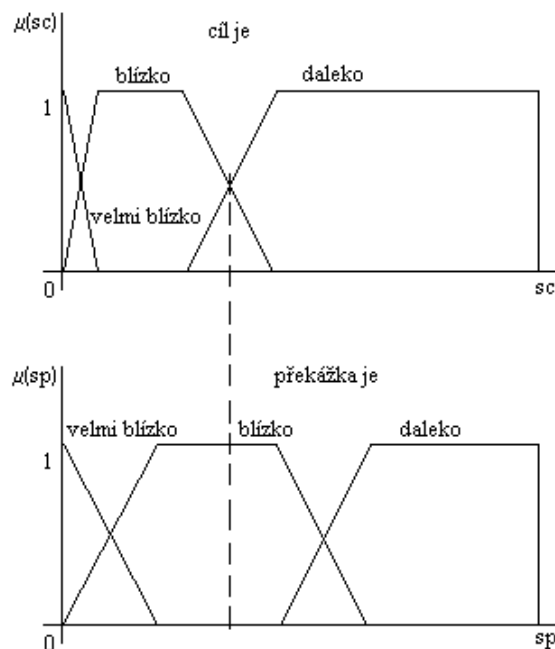
4. Slovní proměnné a jim odpovídající slovní hodnoty

Abychom mohli definovat konkrétní pravidla, je nutné nejprve zavést slovní hodnoty pro každou slovní proměnnou (viz. tab. 3). Všechny slovní proměnné definujeme na univerzu reálných čísel a pro reprezentaci slovních hodnot používáme vhodná lichoběžníková nebo trojúhelníková fuzzy čísla (viz. obr. 5).

Pro dále uvedenou řídicí strategii musí platit $CVB \leq PVB$, $CBL \leq PBL$, $CDA \leq PDA$ a podle toho je nutné volit příslušné fuzzy množiny (viz. obr. 5). Dále ze zavedení našich slovních hodnot pro proměnnou *úhel mezi osou robotu a spojnicí robotu s nejbližší překážkou* (tab. 3) není zcela jasná situace, kdy je překážka přímo před robotem. Proto je zapotřebí volit fuzzy množinu pro PVP resp. PVL tak, aby zahrnovala také nulový úhel. Tím zajistíme, že robot začne reagovat i na přímou překážku a bude ji objíždět zleva resp. zprava.

Tab. 3. Slovní proměnné a jejich slovní hodnoty

Slovní proměnná	Slovní hodnota	Identifikátor
x_1 vzdálenost od cíle	<i>velmi blízko</i>	<i>CVB</i>
	<i>blízko</i>	<i>CBL</i>
	<i>daleko</i>	<i>CDA</i>
x_2 úhel mezi osou robotu a spojnicí robotu s cílem	<i>vlevo</i>	<i>CVL</i>
	<i>vpravo</i>	<i>CVP</i>
	<i>rovně</i>	<i>CRO</i>
x_3 vzdálenost od nejbližší překážky	<i>velmi blízko</i>	<i>PVB</i>
	<i>blízko</i>	<i>PBL</i>
	<i>daleko</i>	<i>PDA</i>
x_4 úhel mezi osou robotu a spojnicí robotu s nejbližší překážkou	<i>vlevo před robotem</i>	<i>PVL</i>
	<i>vpravo před robotem</i>	<i>PVP</i>
	<i>vlevo za robotem</i>	<i>PZL</i>
	<i>vpravo za robotem</i>	<i>PZP</i>
y_1 rychlost robotu	<i>nulová</i>	<i>RNU</i>
	<i>nizká</i>	<i>RNI</i>
	<i>vysoká</i>	<i>RVY</i>
y_2 natočení předního kola	<i>vlevo</i>	<i>KVL</i>
	<i>rovně</i>	<i>KRO</i>
	<i>vpravo</i>	<i>KVP</i>



Obr. 5. Slovní hodnoty pro vzdálenosti od cíle a od překážky

5. Fuzzy pravidla

Pravidla reprezentují znalosti, říkající jak pokračovat z různých poloh robotu dále. Je v nich tedy zakódována řídicí strategie, kterou můžeme charakterizovat následovně. Po zahájení navigace bude robot navigován přímou čarou na cíl. Jestliže se na této přímé čáře vyskytne překážka, potom ji musí robot objet. Pokud bude překážka na pravé straně od osy robotu, robot objede překážku zleva. Pokud bude na levé, objede ji zprava. Pokud bude překážka rovně před robotem, je volbou fuzzy množin pro slovní hodnoty pevně určen jeden směr objíždění, dejme tomu vpravo. V okamžiku, kdy je robot blízko u překážky, začne ji objíždět. Dále bude natáčením předního kola udržovat dostatečnou vzdálenost od překážky až do okamžiku, kdy mu tato překážka přestane bránit v pohybu po přímé čáře z aktuální polohy k cíli.

Tato strategie funguje bez problémů pro konvexní tvary překážek a pro předpoklad dostatečné vzdálenosti mezi nimi. Problém by nastal, kdyby překážka konvexní nebyla. Řídicí systém by předpokládal, že se jedná o dvě různé překážky a navigoval by takto robot do slepé uličky. Podobný problém nastane, pokud by byly překážky blízko sebe tak, že mezi nimi robot neprojede. Tím, že strategie uvažuje jen jednu nejbližší překážku, zůstává druhá překážka opomenuta a může takto dojít snadno ke kolizi. Tyto nevýhody jsou důsledkem našeho omezení na informace o pouze nejbližší překážce.

Pravidla pro naši strategii řízení robotu jsou uvedena v tab. 4. Pro lepší přehlednost zapisujeme pouze identifikátory. Celý zápis např. prvního pravidla vypadá takto:

$$\text{IF } x_1 \text{ is } CDA \text{ AND } x_3 \text{ is } PBL \text{ AND } x_4 \text{ is } PVP \text{ THEN } y_1 \text{ is } RNI \text{ AND } y_2 \text{ is } KVL \quad (7)$$

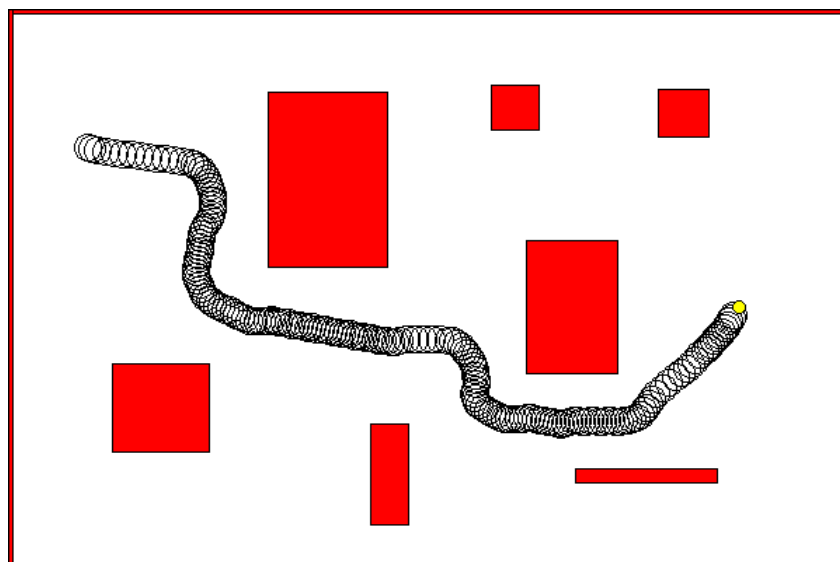
Tab. 4. Navržená pravidla

x_1	x_2	x_3	x_4	y_1	y_2
<i>CDA</i>		<i>PBL</i>	<i>PVP</i>	<i>RNI</i>	<i>KVL</i>
<i>CDA</i>		<i>PBL</i>	<i>PVL</i>	<i>RNI</i>	<i>KVP</i>
<i>CDA</i>	<i>CRO</i>	<i>PBL</i>	<i>PZP</i>	<i>RVY</i>	<i>KRO</i>
<i>CDA</i>	<i>CVP</i>	<i>PBL</i>	<i>PZP</i>	<i>RNI</i>	<i>KVP</i>
<i>CDA</i>	<i>CVL</i>	<i>PBL</i>	<i>PZP</i>	<i>RNI</i>	<i>KVL</i>
<i>CDA</i>	<i>CRO</i>	<i>PBL</i>	<i>PZL</i>	<i>RVY</i>	<i>KRO</i>
<i>CDA</i>	<i>CVP</i>	<i>PBL</i>	<i>PZL</i>	<i>RNI</i>	<i>KVP</i>
<i>CDA</i>	<i>CVL</i>	<i>PBL</i>	<i>PZL</i>	<i>RNI</i>	<i>KVL</i>
<i>CBL</i>	<i>CVP</i>	<i>PBL</i>		<i>RNI</i>	<i>KVP</i>
<i>CBL</i>	<i>CVL</i>	<i>PBL</i>		<i>RNI</i>	<i>KVL</i>
<i>CBL</i>	<i>CRO</i>	<i>PBL</i>		<i>RNI</i>	<i>KRO</i>
<i>CDA</i>	<i>CRO</i>	<i>PDA</i>		<i>RVY</i>	<i>KRO</i>
<i>CBL</i>	<i>CRO</i>	<i>PDA</i>		<i>RNI</i>	<i>KRO</i>
<i>CDA</i>	<i>CVP</i>	<i>PDA</i>		<i>RNI</i>	<i>KVP</i>
<i>CBL</i>	<i>CVP</i>	<i>PDA</i>		<i>RNI</i>	<i>KVP</i>
<i>CDA</i>	<i>CVL</i>	<i>PDA</i>		<i>RNI</i>	<i>KVL</i>
<i>CBL</i>	<i>CVL</i>	<i>PDA</i>		<i>RNI</i>	<i>KVL</i>
<i>CVB</i>				<i>RNU</i>	
		<i>PVB</i>		<i>RNU</i>	

Protože uvažujeme dvě výstupní proměnné, uvažujeme také dvě znalostní báze: jednu pro rychlost robota a druhou pro natočení předního kola. Podmínkovou část pravidel můžeme ale u obou znalostních bází volit stejnou, proto uvádíme společný zápis pravidel tak, že důsledková část obsahuje oba konsekventy spojené spojkou *AND*. Tento zápis pravidla (7) není z hlediska výpočtu pravdivostního ohodnocení odpovědi zcela korektní, protože se vlastně jedná o dvě samostatná pravidla.

6. Ověření funkce navrženého systému

Pro ověření funkce navrženého expertního systému byl vytvořen simulační program. Tento program v sobě zahrnuje jak zmíněný fuzzy expertní systém, tak i celé prostředí s modelem uvažovaného robota a překážkami. Podrobný popis vytvořeného programu a provedených ověřovacích experimentů je uveden v práci (Krček, 2003). Na základě výsledků z tohoto simulačního programu lze konstatovat, že v prostředí robota, pro které je fuzzy expertní systém navržen, pracuje tento systém bez závad. Na obr. 6 je ukázka výstupu ze simulačního programu zobrazující průběh navigace robota.



Obr. 6. Průběh navigace robota mezi konvexními překážkami

7. Závěr a možnosti dalšího vývoje

Uvedený příspěvek je úvodem do řešení problému navigace robota pomocí fuzzy expertního systému. Po popisu principů tvorby fuzzy expertního systému je další část příspěvku věnována konkrétní aplikaci vytvořeného systému pro řešení dílčího úkolu lokální navigace robota spočívajícího v dosažení daného cíle bez kolizí s neznámými překážkami. Funkčnost navrženého systému byla úspěšně ověřena simulačním programem. Další vývoj implementovaného systému se bude ubírat následujícími třemi směry.

První směr je se týká rozdělení problému lokální navigace na dílčí problémy, jimž odpovídají dílčí znalostní báze. Každá znalostní báze reprezentuje jinou strategii. Výsledné chování je kombinací dílčích chování na základě metapravidel.

Druhý směr vývoje se zaměřuje na problém zvládnutí konkávních překážek. Jeden ze způsobů navrhuji Maaref & Barret (2000). Nejprve řeší problém detekce konkávní překážky použitím tří senzorů (vlevo, vepředu a vpravo). V případě že všechny senzory udávají blízké vzdálenosti k překážce (jedná se nespíš o konkávní překážku), zahajuje se navigace se strategií „sleduj stěnu překážky“. Pokud tato strategie není do určité vzdálenosti úspěšná, robot se vrací zpět do výchozího bodu a zkouší strategii „sleduj stěnu překážky“ v opačném směru, nebo do větší vzdálenosti.

Třetí směr se týká použití evolučních metod pro optimalizaci znalostní báze. Optimalizace se provádí programově off-line a po nalezení nejvhodnější báze znalostí se tato předá do reaktivního řídicího systému robotu, který pracuje v reálném čase, tedy on-line. K optimalizaci se používají *genetické algoritmy* (Pratihari et al., 1999) či *genetické programování* (Tunstel & Jamshidi, 1996).

8. Poděkování

Tento článek byl zpracován v rámci vědecko-výzkumného záměru MSM 262100024 „Výzkum a vývoj mechatronických soustav“ a záměru MSM 261100009 „Netradiční metody studia komplexních a neurčitých systémů“.

9. Literatura

- Aguirre, E. & González, A. (2000) Fuzzy behaviors for mobile robot navigation: design, coordination and fusion. *International Journal of Approximate Reasoning*, 25, 3, pp. 255-289.
- Akbarzadeh-T, M. R., Kumbla, K., Tunstel, E. & Jamshidi, M. (2000) Soft computing for autonomous robotic systems. *Computers & Electrical Engineering*, 26, 1, pp. 5-32.
- Doyle, A. B. (1995) *Algorithms and computational techniques for robot path planning*. PhD thesis, University of Wales, School of Electronics Engineering and Computer Systems, Bangor.
- Krček, P. (2003) *Využití expertního systému pro plánování dráhy robota*. Diplomová práce, FSI VUT, Brno.
- Maaref, H. & Barret, C. (2000) Sensor-based fuzzy navigation of an autonomous mobile robot in an indoor environment. *Control Engineering Practice*, 8, 7, pp. 757-768.
- Pratihari, D. K., Deb, K. & Ghosh, A. (1999) A genetic-fuzzy approach for mobile robot navigation among moving obstacles. *International Journal of Approximate Reasoning*, 20, 2, pp. 145-172.
- Tunstel, E. & Jamshidi, M. (1996) On genetic programming of fuzzy rule-based systems for intelligent control. *International Journal of Intelligent Automation & Soft Computing*, 2, 3, pp. 273-284.
- Tzafestas, S. G., ed. (1997) *Methods and applications of intelligent control*. Kluwer Academic Publishers, Dordrecht.
- Zilouchian, A. & Jamshidi, M., eds. (2001) *Intelligent control systems using soft computing technologies*. CRC Press, Boca Raton.