

# POUŽITÍ MATLABU PŘI OVĚŘOVÁNÍ ŘÍDÍCÍHO SYSTÉMU KRÁČEJÍCÍHO ROBOTA

*Ing. HOŠEK Jaromír, Ph.D.*

Vojenská akademie v Brně, Katedra technické kybernetiky a vojenské robotiky

## 1 Úvod

Od začátku 90 let začíná mnoho laboratoří a vědeckých pracovišť na vysokých školách pracovat na řešení problémů spojených s konstrukcí mobilního robotického systému (MRS).

Obecně se řeší inteligentní kráčejíci podvozek jako základní platforma pro nadstavby požadované jednotlivými zadavateli. Pro řízení MRS v terénu je velmi důležitá střední část navrženého hierarchického systému řízení MRS – operativní úroveň řízení. Cílem této úrovně je na základě předem zadaných informací o předmětné oblasti a informací získaných ze senzorů při průchodu terénem, generovat řídicí veličiny pro lokomoční úroveň tak, aby se MRS pohyboval terénem v souladu se stanovenými cíli.

## 2 Plánování trasy

Pro zabezpečení tohoto úkolu je nutné souběžně řešit několik úloh, včetně aktualizace vnitřního modelu mobilního robota informacemi získanými ze senzorů, plánování trasy a řešení průchodu nad tímto aktualizovaným vnitřním modelem, bezkolizní obejití objevené překážky, sledování trasy, atd.

Pro řízení byl navržen hierarchický řídicí systém využívající metod prohledávání stavového prostoru. Mapa terénu byla převedena do cenové mapy. Cenová mapa je čtvercová matice kde hodnoty prvků-uzlů představují cenu přechodu přes daný uzel. Základ ceny tvoří převýšení mezi sousedními uzly doplněné o další charakteristiky terénu jako půdní a rostlinný typ, sídla, vodní toky, apod. Pro určení optimální trasy zvolené pracovní oblasti bylo použito Bellmanovo dynamické programování. Pro ověření byla použita mapa ve formě matice o rozměru 41x41 a s využitím Bellmanova dynamického programování byla vybrána optimální trasa. Na trasu byla umístěna překážka a provedeno ověření trasy a její možná korekce. Na obr.1a,b je znázorněna původně vypočtená trasa a korigovaná trasa po vložení překážky.

Algoritmus plánování trasy založený na Bellmanově dynamickém programování je následující:

- 1) Jako cílový uzel je ustaven uzel  $b_N$ .

- 2) Je vytýčen seznam všech potenciálních bodů  $b_{N-1}$ , tj. bodů sousedících s  $b_N$ .
- 3) Jsou vyloučeny nepřístupné body, tj. nepřístupné nebo je neuskutečnitelný přechod.
- 4) Každému přípustnému bodu z  $\{b_{N-1}\}$  je přiřazena hodnota kritéria  $J(N-1)=C_{N,N-1}$  a je zaznamenán směr přechodů z  $\{b_{N-1}\}$  do  $b_N$ .
- 5) Je vytvořen seznam všech potenciálních bodů  $\{b_{N-2}\}$ , tj. bodů, které sousedí s některým přípustným bodem, které byly zahrnuty do seznamu  $\{b_{N-1}\}$ .
- 6) Opět je provedena selekce těchto bodů z hlediska přístupnosti a uskutečnitelnosti přechodu.
- 7) Pro každý z přípustných bodů  $\{b_{N-2}\}$  je nalezen přechod do některého z bodů ze seznamu  $\{b_{N-1}\}$  pro který platí

$$J(N-2)=\min\{J^i(N-1)+C_{N-1,N-2}^i\}, \quad (1)$$

kde index  $i$  označuje jednotlivé body v seznamu  $\{b_{N-1}\}$  a je zaznamenán směr přechodu z  $\{b_{N-2}\}$  do odpovídajícího bodu  $b_{N-1}$ . Dále jsou opakovány body 5),6) a 7) pro další seznamy bodů  $\{b_{N-3}\},\{b_{N-4}\}...$  s minimalizací ve tvaru

$$J(k-1)=\min\{J^i(k)+C_{k,k-2}^i\}. \quad (2)$$

- 8) Konec řešení nastává až všechny body mapy jsou zahrnuty do některého seznamu  $\{b_k\}$ .

Výsledkem řešení je trasa z každého přístupného bodu mapy do cílového bodu  $b_N$ , včetně bodu, ve kterém je robot umístěn (startovní bod).

#### Poznámka 1

Pokud striktně požadujeme, aby seznamy  $\{b_k\}$  a  $\{b_{k-1}\}$  byly disjunktní, dosáhneme pouze suboptimálních tras, tj. nemůžeme vyloučit, že některá z tras, kdyby vedla např. přes dva body jednoho a téhož seznamu, by nezískala lepší hodnotu kritéria  $J$ . Protože však dominantním požadavkem je získání průchodné trasy a míra hodnocení  $C_i$  je vždy zatížena jistou chybou způsobenou konečnou přesností vyjádření vlivu jednotlivých parametrů terénu, je získaná suboptimální trasa dostačující.

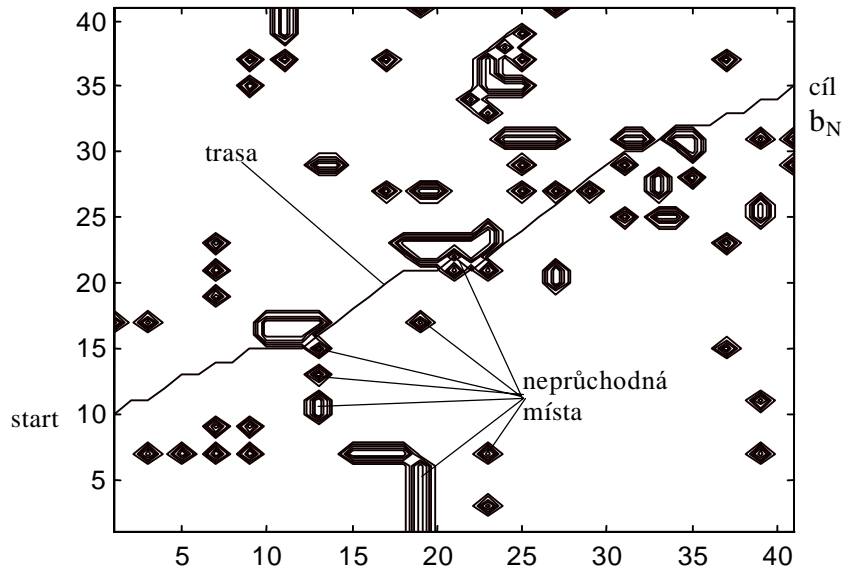
#### Poznámka 2

V případě nejednoznačnosti určení optimálního přechodu (kdy např. přechod z  $b_{k-1}$  do  $b_k$  má shodnou minimální hodnotu kritéria  $J(k-1)$  pro dva body ze seznamu  $\{b_k\}$ ) je nutné zajistit vhodnou strategii výběru např.

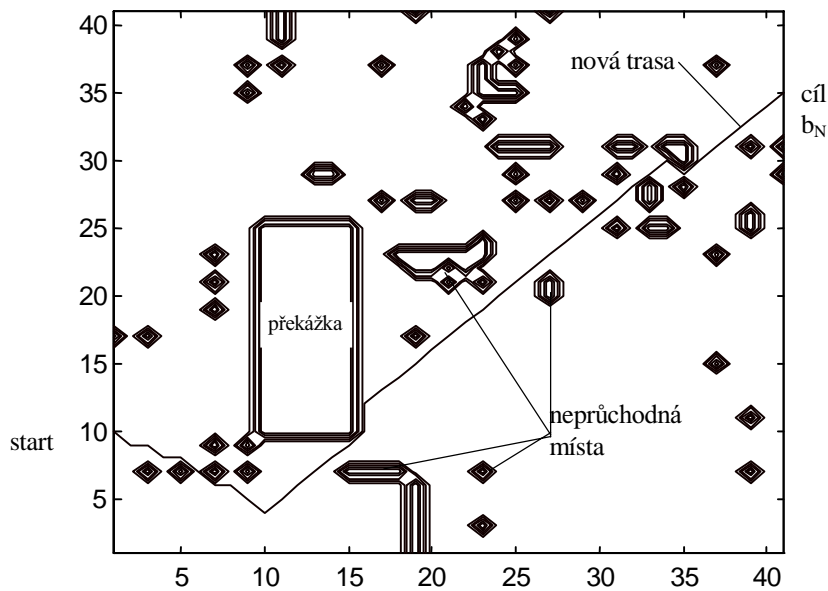
- náhodný výběr,

- podle pořadí vyhodnocení (první, poslední),
- na základě další doplňkové informace.

V programech je vesměs použito preference takového přechodu  $b_{k-1} \rightarrow b_k$  jehož směrový vektor se nejvíc blíží směru z bodu  $b_k$  do bodu  $b_N$ .



**Obr. 1a** Plán původní trasy



**Obr. 1b** Přeplánování trasy po vložení překážky

Uzly na místě trasy, kde byla překážka měly cenu přechodu vyšší než povolená mez. Výslednou mapu s překážkou jsem potom získal jako logický součet matice mapy a matice překážky.

$$C^{(a)} = C \vee P \quad (3)$$

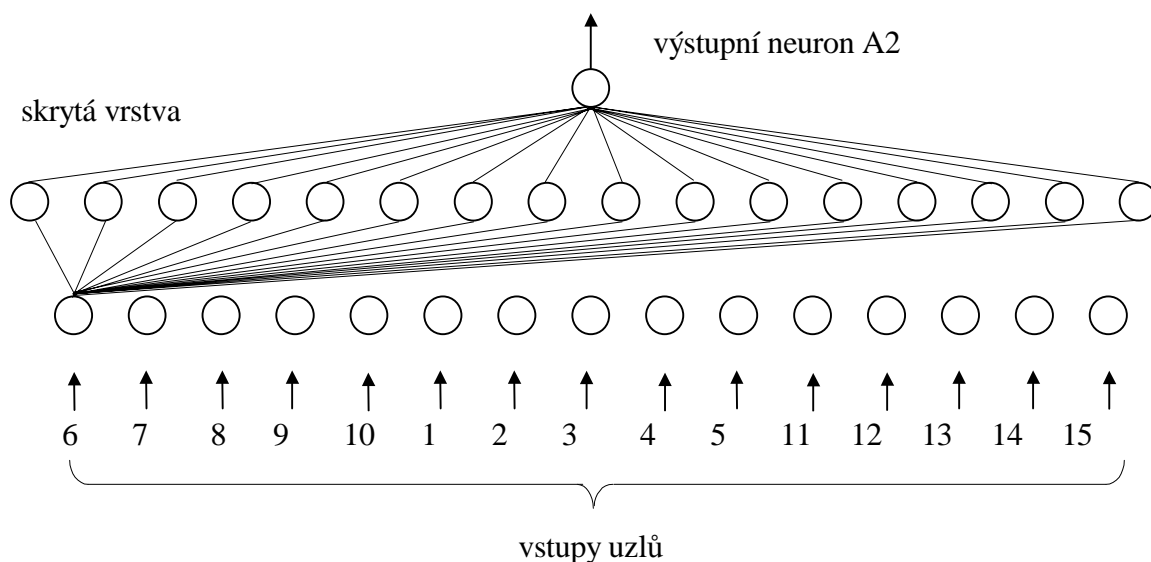
kde  $C^{(a)} = || C_{i,j}^{(a)} ||$  výsledná mapa po vložení překážky  
 $C = || C_{i,j} ||$  původní cenová mapa  
 $P = || P_{i,j} ||$  matice překážky

Výpočet posloupnosti uzlů optimální trasy i celé mapy tras trval asi 4 minuty.

### 3 Průchod terénem

Problémem při řízení průchodu terénem jsou malé, nahodilé překážky (větší kameny, výmoly, apod.). Protože se nacházejí v bezprostřední blízkosti mobilního robota byly odhaleny „na poslední chvíli“ a o jejich překonání je nutno rozhodnout okamžitě, aby nedošlo ke kolizi.

Rozhodovací člen je tvořen třívrstvou umělou neuronovou sítí, obr.2.



**Obr. 2** Konfigurace neuronové sítě

Vstupní vrstva zabezpečuje pouze rozvedení každého ze vstupních signálů na všechny výkonné prvky druhé (skryté) vrstvy. Na obr.2 jsou pro přehlednost vyznačena spojení pouze jednoho neuronu vstupní vrstvy se všemi neurony druhé vrstvy. Jedná se o uniformní propojení, kde je vstupní signál každého neuronu nižší vrstvy přiveden na vstup každého neuronu následující vrstvy. Neurony ve vstupní vrstvě nemají jinou funkci než rozvedení vstupních signálů. Lze je tedy považovat za lineární prvky s jednotkovým přenosem a s rozvětveným výstupem. Počet těchto neuronů odpovídá počtu vstupů. Druhá vrstva obsahuje funkční výkonné prvky přenosová funkce těchto

neuronů je nelineární. Poslední vrstva je tvořena jedním výstupním neuronem. Jeho činnost je obdobná jako u neuronů druhé vrstvy, ale přenosová funkce je lineární.

Pro naučení neuronové sítě bylo třeba získat mohutnou a vhodnou množinu vzorů. Tyto vzory byly získány z části mapy, kterou vede plánovaná trasa pro průchod terénem. Pro ověření tohoto návrhu bylo použito „Neural Networks Toolbox“ v programu MATLAB. Vzory byly získány výběrem zvoleného bloku uzlů (část prostoru před robotem v dosahu senzorů s nejkratším dosahem) z mapy-matice doplněné o nepřekročitelné překážky nad níž byla plánována trasa. Překážky byly dány hodnotou některých uzlů jež byla vyšší než maximální hodnota pro průchod. Protože mezi těmito vzory nemusely být všechny možné neprůchodné vzory byla množina těchto vzorů doplněna o všechny možné neprůchodné vzory.

Pro učení sítě byla použita metoda učení se zpětným šířením korekce (back propagation). Signál od trénovacích vzorů postupuje sítí v přímém směru šíření k výstupu. Výstup sítě A2 je porovnán s požadovanou hodnotou T.

$$E=T-A2 \quad (4)$$

Rozdíl E mezi požadovanou a skutečnou hodnotou je porovnán se zadanou přesností a pokud je větší jsou citlivosti vah na výstupní signál neuronu počítány zpětně od výstupní vrstvy ke vstupní vrstvě. Pro nastavení vah bylo použito Delta pravidlo

$$\Delta w_{ij}(k+1)=\eta (s_i(k)-a_i(k))y_j(k) \quad (5)$$

kde:  $\Delta w_{ij}(k+1)$  - přírůstek hodnoty váhy od j-tého neuronu k i-tému neuronu

$a_i(k)$  - hodnota aktivační funkce na i-tém neuronu v čase k

$y_j(k)$  - výstup j-tého neuronu v čase k

$s_i(k)$  - vzor na který je i-tý neuron trénován

$\eta$  - konstanta ovlivňující rychlost učení neuronu

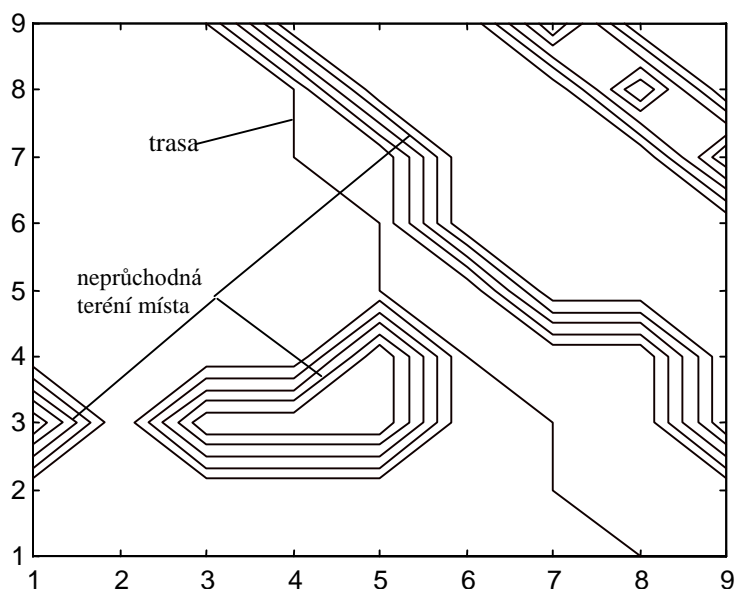
Nová hodnota citlivosti vah potom bude

$$w_{ij}(k+1)=w_{ij}(k)+\Delta w_{ij}(k+1) \quad (6)$$

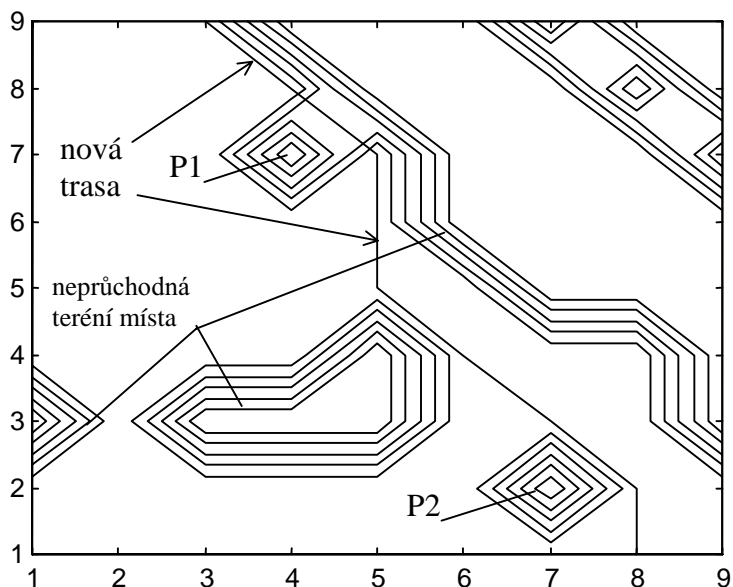
Pro zvolený případ matice terénu o rozměru 41x41 polí bylo získáno 332 vzorů. Učební množina byla tvořena každým lichým vzorem a testovací množina každým sudým, tj. každá množina, po 166 vzorech. K učebním vzorům byla přidána množina všech neprůchodných vzorů odvozených ze zvoleného bloku uzlů, těchto vzorů bylo 164.

Po 10 000 epochách učení touto množinou 330 vzorů bylo z testovací množiny 166 vzorů nesprávně určeno 5 vzorů, tj. asi 3%. Při dalším učení se toto procento dále snižovalo. Takto naučenou síť lze použít jako rozhodovací podsystém pro výběr průchozí trasy před mobilním robotem.

Na obr. 3a a obr. 3b je znázorněna trasa vybraná neuronovou sítí v terénu bez překážek a s překážkami.



**Obr. 3a** Výběr trasy v terénu bez překážek



**Obr. 3b** Výběr trasy s překážkami P1 a P2

## 4 Závěr

Navržený systém řízení mobilního robota zahrnuje jak plánování optimální trasy, tj. úspory času nebo energie dle zvoleného kritéria optimality, tak i vyhýbání se náhodným překážkám.

Výhodou navrženého způsobu řešení rozhodovacího podsystemu-neuronové sítě je jeho univerzálnost. Je možné měnit rozměry testovaného bloku uzlů použitého pro výrobu vzorů v závislosti na dosahu senzorů a tím i typ použitého terénu (rovinný, hornatý,...).

## Literatura

[1] Neural Network Toolbox, The MATHWORKS Inc., 1998

[2] Hošek J.: Inteligentní řízení autonomního robota v terénu, Disertační práce, Vojenská akademie Brno, Brno 1999

Jaromír HOŠEK

e-mail: [jara@mrkev.vabo.cz](mailto:jara@mrkev.vabo.cz)

Vojenská akademie Brno

tel.: 05/41182555

PS-13, K308-VA Kounicova 65

612 00 Brno