

## Navigácia mobilného robota s vizuálnym systémom ku značke

Bc. Peter Pásztó, doc. Ing. Peter Hubinský, PhD. \*  
Automatizácia a regulácia, ÚRPI  
peterpaszto@gmail.com

### Abstrakt

Cieľom tejto práce bolo navigovať mobilného robota vybaveného kamerovým systémom ku kruhovej značke modrej farby len na základe vizuálnych informácií získaných z kamery. To znamená, že robot nevyužíva odometriu a nemá žiadne iné snímače.

Správna navigácia ku značke na základe získaného obrazu z kamery prebieha vo viacerých krokoch. Prvým krokom je vyhľadanie značky v prostredí. Na tento účel je použitý farebný filter, ktorý z nasnímaného obrazu vyberie pixely určitej farby. Ďalší krok z vyfiltrovaného obrazu získa hrany a použitím Houghovej transformácie sa nájde kružnica v obraze. Z tejto informácie sa určí vzdialenosť od značky a z natočenia kamery sa zistí smer ku značke. Na základe týchto informácií sa robot natočí ku značke a prejde polovicu vypočítanej vzdialenosti smerom k nej. Tieto kroky sa opakujú, až kým robot nepríde ku značke.

Vytvorený program pre ovládanie robota má možnosť zmeniť hlavné parametre potrebné pre jednotlivé kroky a tým sa môže adaptovať pre rôzne prostredia. Myslí sa tým napr. zmena osvetlenia alebo iný typ podlahy, na ktorej sa robot pohybuje.

Vykonalé experimenty sú dokumentované vo forme video súboru.

### 1. Úvod

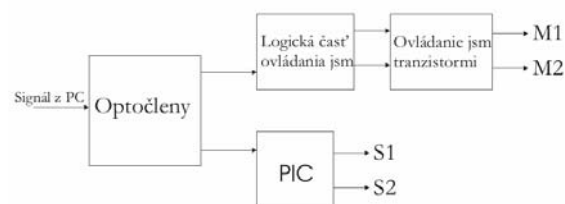
Existuje veľa metód navigácie mobilného robota. Niektoré roboty využívajú fuzzy logiku, iné majú ultrazvukové a infračervené snímače vzdialenosti, sonary a tak isto môžu byť vybavené kamerovým systémom. Všeobecne sa dá konštatovať, že moderné roboty sú vybavené veľkým množstvom snímačov, ktoré im dávajú veľa informácií. Všetky tieto snímače majú svoje využitie pre rôzne typy úloh, ktoré mobilný robot musí splniť, majú svoje výhody a nevýhody. Tak isto existuje veľa teórií, ako optimálne využiť všetky informácie získané zo snímačov.

V tejto práci mobilný robot využíva jeden snímač – kameru a prostredníctvom vizuálnej informácie sa snaží správne dostať ku značke. Z toho je zrejmé, že na navigáciu sú použité metódy počítačového videnia –

filtrácia obrazu, získanie hrán a rozpoznanie kružníc pomocou Houghovej transformácie.

### 1.1. Opis mobilného robota

Mobilný robot je postavený z upraveného hračkárskeho pásového vozidla, na ktoré je pripravená otočná veža s kamerou. Na komunikáciu robota s počítačom je použitý LPT komunikačný kábel. Pripustená kamera komunikuje s počítačom cez USB port. Napájanie robota je z externého zdroja. Pohyb robota zabezpečujú dva jednosmerné motory. Pohyb kamery zabezpečujú dva servomotory, ktoré sú ovládané mikroprocesorom PIC16F84A.



Obr. 1. Bloková schéma zapojenia mobilného robota

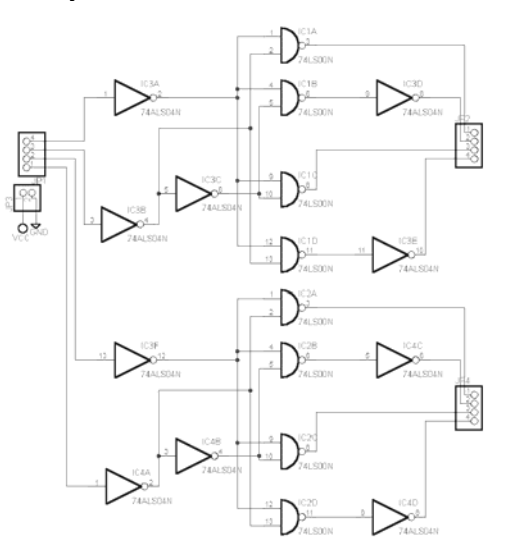
Ovládanie robota z počítača je formou posielania príslušných kombinácií bitov na LPT port. Robot sa pohybuje v rôznych smeroch taký čas, na aký je na LPT porte aktívna daná kombinácia bitov.

Optočleny sú použité na ochranu počítača pred skratom na strane robota. Z tohto miesta sa 8 bitový signál delí na dve časti po 4 bitoch. Prvé štyri bitu ovládajú jednosmerné motory - otvárajú alebo zatvárajú príslušné tranzistory, ktoré zabezpečujú tok prúdu motormi. Pred obvodom s tranzistorami je ešte obvod pozostávajúci z logických členov (NAND a NOT), ktorý upravuje signál z počítača tak, aby sa stále otvorila príslušná kombinácia tranzistorov.

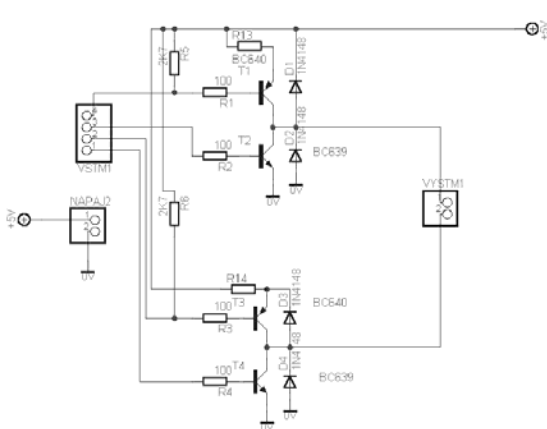
Ovládanie servomotorov zabezpečuje mikroprocesor PIC16F84A, ktorý periodicky generuje dva impulzy pre dva servomotory. Natočenie servomotorov závisí od striedy 20 ms impulzu, ktorá sa pohybuje medzi hodnotami 0,5 ms (uhol otočenia 0°) do 2,5 ms (uhol otočenia 180°). Štyri bity z počítačového signálu sú použité pre mikroprocesor, ktorý na základe týchto

\* Vedúci práce

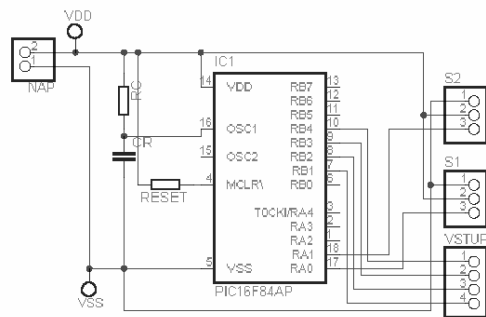
údajov zvýši alebo zníži striedu pre daný servomotor. Na nasledujúcich obrázkoch sú schémy zapojenia jednotlivých častí robota.



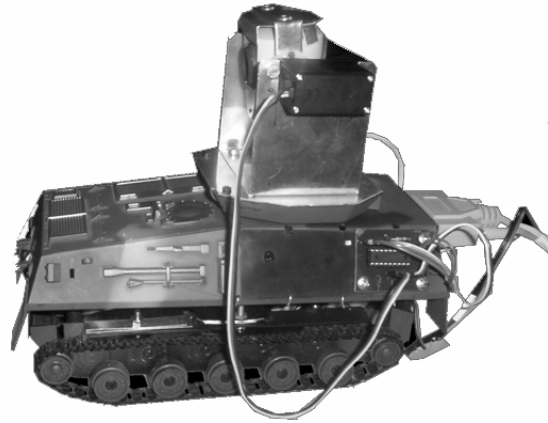
**Obr. 2.** Schéma zapojenia logickej časti ovládania jednosmerných motorov.



**Obr. 3.** Schéma zapojenia ovládania jednosmerných motorov tranzistormi.



**Obr. 4.** Schéma zapojenia ovládania servomotorov.



**Obr. 5.** Mobilný robot s vizuálnym systémom.

### 1.2. Detekcia hrán v šedotónovom obraze

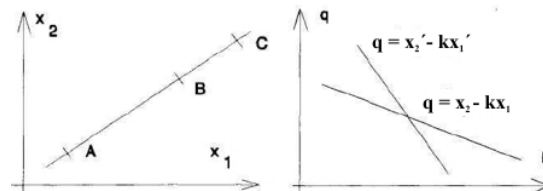
Detekcia hrán je založená na poznatku, že druhá derivácia obrazovej funkcie má v mieste hrany nulovú hodnotu. Druhou deriváciou obrazovej funkcie

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

sa získajú konvolyčné masky, ktorými sa vynásobia všetky body šedotónového obrazu a pri vhodnom prahovaní získame hrany.

### 1.3. Princíp Houghovej transformácie

Houghovu transformáciu je vhodné použiť, ak hľadáme objekty, ktorých tvar vieme opísať rovnicou. Princíp metódy sa jednoducho dá vysvetliť na detekcii priamok. Uvažujme priamku prechádzajúcu bodmi  $A=(x_1, x_2)$  a  $B=(x_1', x_2')$ . Všetky priamky prechádzajúce bodom A musia vyhovovať rovnici priamky  $x_2=kx_1+q$ . Pretože  $x_1, x_2$  sú pre bod A konštanty, rovnicu môžeme chápať ako rovnicu v priestore parametrov  $k, q$ . Tam sú všetky priamky prechádzajúce v obrazovom priestore bodom A reprezentované priamkou  $q=x_2-kx_1$ . Aj všetky priamky prechádzajúce bodom B môžeme reprezentovať v priestore parametrov priamkou  $q=x_2'-kx_1'$ . Jediným spoločným bodom oboch priamok v priestore parametrov je bod  $(k, q)$  odpovedajúci v obrazovom priestore priamke spájajúcej body A a B.



**Obr. 6.** Princíp Houghovej transformácie.

Ak chceme Houghovou transformáciou vyhľadať kružnice, tak namiesto rovnice priamky použijeme rovnicu kružnice  $(x - a)^2 + (y - b)^2 = r^2$ . V tomto prípade sú tri parametre  $(a, b, r)$ . Zovšeobecnený algoritmus je nasledovný:

1. Diskretizujte  $n$  – rozmerný priestor parametrov v rozsahu ich možných hodnôt.
2. Vytvorte počítadlá, pre každú kombináciu parametrov jedno a vynulujte ich. ( $n$  – rozmerná matica počítadiel).
3. Pre každý obrazový element  $(x, y)$ , kde sa nachádza významná hrana, inkrementujte všetky počítadlá  $p$ , pre ktoré platí  $f(x, y, p) = 0$
4. Významné lokálne maximá počítadiel zodpovedajú parametrom hľadaných kriviek.

## 2. Programová realizácia

Ako bolo spomenuté v úvodnej časti, ovládanie robota prebieha na základe generovania kombinácie bitov na LPT port počítača a doba chodu robota je určená časom, ktorým je kombinácia bitov na výstupe počítača aktívna. Inak povedané, ak sa má robot pohnúť na sekundu dopredu, na LPT porte počítača musí byť poslaná daná kombinácia bitov na jednu sekundu. Takýto spôsob ovládania vnáša do navigácie robota značné nepresnosti, ktoré sa vytvorený program snaží korigovať.

Po spustení programu sa kamera robota vytočí úplne doľava a vertikálne sa kamera natočí do stredu. Následne sa objaví úvodná obrazovka, ktorá sa skladá z niekoľkých častí. Ľavá horná časť zobrazuje obraz z kamery robota, v pravej hornej časti sa zobrazujú vizuálne informácie získané počas spracovania obrazu. V dolnej časti obrazovky sa nachádzajú tlačidlá START a STOP, ktoré slúžia na spustenie a zastavenie automatickej navigácie robota. Pre kontrolu činnosti programu, výpis dôležitých údajov a premenných slúži textové okienko v strede dolnej časti obrazovky. V menu OPTIONS sa nachádzajú možnosti na zmenu parametrov navigácie a na nastavenie pôvodných parametrov.

Automatická navigácia robota sa začne po stlačení tlačidla start. Spôsob hľadania značky je taký, že sa po spustení navigácie filtruje obraz z kamery. Filter je nastavený na modrú farbu (presnejšie bude popísaný v ďalších odsekoch). Obraz po filtrovaní sa zobrazí v pravej hornej časti obrazovky. Ak počet pixelov po filtrovaní je väčší ako nastavená hranica, nasledujú ďalšie kroky spracovania obrazu. Inak sa kamera otočí o krok doprava a znovu prebehne filtrovanie obrazu. Krok otočenia kamery je daný meniteľnou časovou konštantou. Ak sa kamera otočí o svoj maximálny rozsah  $180^\circ$ , tak sa ukončí automatická navigácia robota.

Ďalšie kroky spracovania obrazu nasledujú po úspešnej filtrácii, tzn. po získaní nadprahového počtu pixelov. Nasledujúcim krokom je získanie hrán z výsledného obrazu. V tomto prípade nebolo vhodné použiť konvolučné masky, keďže po filtrácii v obraze nezostanú všetky potrebné body značky a to by viedlo k detekcii falošných hrán. Postup získania hrán bude opísaný tiež v nasledujúcich odsekoch. Výsledok po získaní hrán sa znova zobrazí v pravej hornej časti obrazovky.

Po tomto kroku môže nasledovať detekcia kružnice Houghovou transformáciou, ktorá už bola opísaná.

Získaním hrán sa značne zrýchli tento krok, pretože vstupný obraz do tejto funkcie obsahuje redukovaný počet pixelov. Čím je väčší počet pixelov po získaní hrán, tým dlhšie trvá tento krok.

Po detekcii kružnice má program k dispozícii údaje o polohe  $(x, y)$  kružnice v obraze, jej polomer  $r$  a počet krokov, ktoré sa kamera natočila doprava, kým sa kružnica našla. Z týchto údajov program vypočíta čas, na ktorý musí vyslať na LPT port signál na spustenie jednosmerných motorov, aby sa robot natočil smerom ku značke. Tak isto z polomeru kružnice  $r$  vypočíta vzdialenosť od značky a určí čas, na ktorý sa musí robot pohybovať dopredu smerom ku značke. Nakoniec na LPT port vyšle signál na chod smerom dopredu, ale len na polovicu vypočítaného času. Robot takto spraví prvý krok smerom ku značke, ale kvôli rôznym nepresnostiam môže stratiť značku z dohľadu. Tieto nepresnosti sú napr. nerovnomerné otáčanie sa jednosmerných motorov robota, kĺzanie sa na podlahe, atď. Po prejení polovičnej vzdialenosti ku značke robot skontroluje, či značku nestratil z dohľadu (prebehnú znova funkcie na filtrovanie, určenie hrán a detekciu kružnice). Ak ju nestratil, znova k nej prejde polovicu vypočítanej vzdialenosti a ak ju stratil, kamera sa natočí doľava a znova ju začne hľadať. Prejdením polovičnej vzdialenosti v každom kroku sa znižuje chyba, ktorá robota môže navigovať v zlom smere.

Robot sa zastaví pri značke, keď vypočítaná vzdialenosť klesne pod 50 cm.

### 2.1. Postup filtrovania obrazu

Farebné zložky RGB každého pixlu obrazu sa porovnávajú so zadanými hodnotami, ktoré užívateľ môže meniť. Ak zložky farebného bodu zodpovedajú zadanému rozsahu, tak sa pixel v obraze ponechá tak, že jeho nová farba bude čierna. Súčasne sa zvýši počítadlo ponechaných pixelov. Ak zložky RGB daného bodu nezodpovedajú rozsahu, tak sa stanú bielymi.

Rozsah zložiek RGB sa určil pokusom, keďže sa menia v závislosti osvetlenia, druhu kamery a šumu. Správne nastavenie filtra je predpokladom k úspešnej detekcii značky. Výhodou filtra je, že ponechá len body z určitého farebného rozsahu a všetky ostatné objekty v miestnosti vynechá, čiže nevstupujú do ďalšieho spracovania a nespomaľujú navigáciu robota.

### 2.2. Postup získania hrán

Keďže filtrovaním v obraze zostali už len body určitej farby, tak je pomerne jednoduché oddeliť „vnútorné“ body kružnice od hrany kružnice.

Postup je taký, že pre každý (čierny) bod obrazu sa spočíta počet jeho susedov v každom smere (hore, doľava, doprava) v danom okolí. Okolie si môže nastaviť užívateľ. Ak je počet susedov menší ako maximálny počet susedov (možnosť meniť), tak sa daný bod považuje za hranu, inak sa nastaví ako biely.

Takýmto spôsobom do ďalšieho spracovania postupuje už len obraz, ktorý obsahuje tie body kružnice, ktoré predstavujú hranu.

### 2.3. Natočenie robota smerom k nájdenej kružnici, výpočet vzdialenosti a času potrebného na prejdenie vzdialenosti

Pri hľadaní značky otáčaním kamery program počíta počet krokov, o koľko sa kamera otočila. Po nájdení značky sa natočenie celého robota smerom k nej delí na dve časti podľa toho, či kamera prešla cez stredovú os robota alebo nie. Ak ňou kamera neprešla, robot sa točí doľava o čas **(počet krokov do stredu – počet spravených krokov)\*časová konštanta robota**. Ak kamera prešla stredovou osou, tak sa čas počíta ako **(počet spravených krokov – počet krokov do stredu)\*časová konštanta robota**. Tak isto sa otočí kamera na stred s tým rozdielom, že má inú časovú konštantu. Dané časové konštanty sa dajú meniť v nastaveniach, pôvodné boli získané meraním. Výpočet vzdialenosti od kružnice na základe polomeru sa realizovalo tiež meraním. Bola vytvorená tabuľka, kde sa merala skutočná vzdialenosť robota od značky a polomer rozpoznanej kružnice. Z týchto údajov sa vyhotovil graf závislosti vzdialenosti od polomeru rozpoznanej kružnice a vytvorila sa mocninová náhrada tejto závislosti.

**Tab. 1.** Závislosť polomeru rozpoznanej kružnice od vzdialenosti

Vzdialenosť (mm)	Polomer (pixely)
2000	36
1900	38
1800	40
1700	43
1600	46
1500	49
1400	53
1300	58
1200	62
1100	68
1000	78
900	81
800	95
700	108
600	130
500	154
400	210
300	262

Mocninová náhrada grafu vytvoreného z tejto tabuľky je:

$$l = 58958 \cdot r^{-0.9434} \quad (2)$$

Čas potrebný na chod robota smerom dopredu sa vypočíta zo známeho vzťahu:

$$t = \frac{s}{v} \quad (3)$$

Dráha s sa počítala v predošlom kroku. Ostáva vedieť iba rýchlosť robota. Tá tiež bola nameraná na rôznych vzdialenostiach a vypočítaná priemerná rýchlosť.

Po vypočítaní času potrebného k dosiahnutiu značky sa robot môže pohnúť smerom k nej. Ako bolo uvedené v predošlých odsekoch, robot vždy prechádza iba polovičnú vzdialenosť a potom kontroluje, či nestratil značku z dohľadu.

### 3. Experimenty a výsledky

Navigácia robota bola otestovaná v reálnom prostredí s mnohými objektmi a s pestrými farbami prostredia. Vhodné je však zvoliť takú farbu značky, aká farba sa v prostredí nenachádza. Preto som zvolil bledomodrú farbu. Výsledky navigácie robota sú dokumentované vo forme digitálnych video nahrávok. Robot bol otestovaný aj na väčšie vzdialenosti ako boli spravené merania v **Tab. 1.** „Závislosť polomeru rozpoznanej kružnice od vzdialenosti“.

Nastavenie filtrovania farieb závisí od osvetlenia v prostredí. Preto bol filter nastavený podľa reálne nasnímaného obrazu s kamerou a zistenými zložkami RGB. Ak by robot pracoval v inom prostredí, je vhodné pred spustením navigácie nasnímať značku a nastaviť farebný filter podľa nových hodnôt zložiek RGB.

Tak isto pri zmene prostredia je vhodné namerať časové konštanty robota spomínané v tomto článku a zmeniť ich v menu OPTIONS.

### 4. Zhodnotenie

Cieľom bolo čo najpresnejšie navigovať mobilného robota ku zadanej značke. Pre tento účel boli využité len vizuálne informácie získané z robota a navigácia bola sťažovaná aj časovým spôsobom ovládania robota.

Výsledky experimentov dokazujú, že aj bez použitia iných snímačov sa dá robot úspešne navigovať k zadanému cieľu. Metódou prejdenia polovičnej vzdialenosti v každom kroku sa redukuje chyba, ktorú robot spraví nesprávnym uhlom natočenia ku značke. Takýmto spôsobom je možné vykorigovať nepresnosti, ktoré vznikajú klzaním pásov po povrchu miestnosti a vzhľadom na vek robota aj zasekávaním sa jednosmerných motorčekov. Pri experimentoch sa často stávalo, že robot stratil značku z dohľadu, ale po kontrole, ktorá bola zavedená, ju znovu našiel a úspešne prišiel do jej okolia.

Pri dlhších vzdialenostiach ako 2 m sa robot tak isto dostal do okolia značky. Odhad vzdialenosti je aj pri vyšších vzdialenostiach relatívne presný, pretože sa polomer rozpoznanej kružnice pri veľkých vzdialenostiach mení oveľa menej ako pri malých. Treba však dbať na to, aby vzdialenosť bola dostačujúca na rozpoznanie malej kružnice, teda aby kružnica bola v kamere ešte viditeľná.

## 5. Odkazy na literatúru

- [1] Šonka, M., Hlaváč, V., “Počítačové vidění”, Praha: Grada, 1992
- [2] Hlaváč, V., Sedláček, M., “Zpracování signálu a obrazu”, Praha: Vydavatelství ČVUT, 2001
- [3] Šonka, M., Hlaváč, V., Boyle, R., “Image Processing, Analysis and Machine Vision”, Brooks/Cole Publishing Company, 1999
- [4] Klotton, G., Pásztó, P., “Polohovacie zariadenie pre kameru“, Stredná priemyselná škola elektrotechnická S. A. Jedlika, 2004