

Planiranje poti za neholonomni avtonomni sistem v tesnem zemljevidu

Matevž Bošnjak, Igor Škrjanc
Fakulteta za elektrotehniko, Univerza v Ljubljani
E-pošta: matevz.bosnak@fe.uni-lj.si

Path planning for a nonholonomic autonomous system in a tight map

This article presents an approach to path planning for a nonholonomic autonomous system based on an arbitrary system footprint for obstacle detection. The system footprint and a map of obstacles are joined using a convolution method in order to divide the limited configuration space into the obstructed and the obstacle-free regions, where A* algorithm is then used to find a path between the start and the goal system states. The algorithm is used to navigate a nonholonomic autonomous system in an environment with very limited space around the autonomous system.

1 Uvod

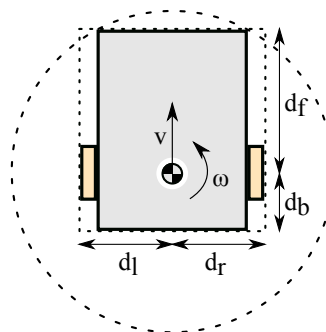
Uspešen avtonomni sistem mora biti sposoben zaznavanja okolja okoli sebe ter premikanja po prostoru na podlagi zaznanih informacij o okolju. Planiranje poti je pri tem eden ključnih korakov, saj se na podlagi poznavanja zemljevida okolja (položaj ovir ali drugih elementov, pomembnih pri načrtovanju premikov avtonomnega sistema) in položaja robota v okolju poskuša najti najkrajšo oz. optimalno pot med dvema točkama, ki sta najpogosteje trenutni položaj sistema ter zeleni položaj. V članku je predstavljen postopek za iskanje najkrajše možne poti pri strogih omejitvah glede fizične velikosti prostora, v katerem se avtonomni sistem premika.

2 Avtonomen sistem

Obravnavan avtonomni sistem lahko predstavimo z neholonomnim mobilnim robotom s diferencialnim pogonom, kjer je pogonska os premaknjena iz središča (slika 1). Sistem opišemo z naslednjim kinematičnim modelom [1]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (1)$$

kjer sta (x, y) koordinati izhodišča sistema ter ϕ rotacija sistema okoli istega izhodišča. Obravnavan avtonomni sistem ima naslednje dimenzije (slika 1): $d_l = d_r = d_b = 0,45$ m in $d_f = 0,90$ m.

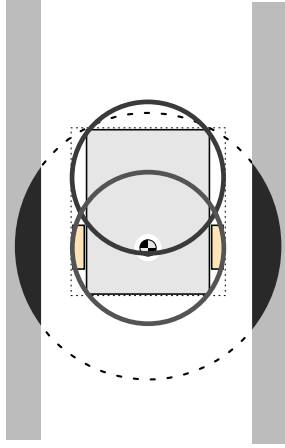


Slika 1: Obravnavan neholonomni sistem z diferencialnim pogonom

Vplivni prostor sistema lahko opišemo na več načinov. Pri postopkih planiranja poti se za namene preverjanja trkov najpogosteje uporablja opis z najmanjšo očrtano krožnico s središčem v koordinatnem izhodišču sistema (črtan krog na sliki 1), saj je za grobo preverjanje trka z oviro potrebno preveriti le, če se katera izmed ovir nahaja znotraj te krožnice. V naslednjih korakih se nato uporabi drobljenje oz. po korakih natančnejši opis geometrije robota (polni krožnici na sliki 2), ki povečuje računsko kompleksnost preverjanja trka. V tem članku predstavljamo pristop, pri katerem vnaprej poznamo poljuben rasterski opis vplivnega sistema robota ter zemljevid okolja z ovirami.

3 Zemljevid okolja

Zemljevid okolja predstavlja kataloški zapis znanih podatkov o okolju, v katerem se avtonomni sistem giblje. Najpogosteje vsebuje podatke o konfiguraciji prostora ter položaje značilnik v tem prostoru. V našem primeru je zemljevid (Slika 3) zgrajen na podlagi meritev laserskega senzorja tipa LIDAR (angl. *Light Detection and Ranging*) in je predstavljen v obliki kvadratne mreže $\mathcal{O}(x, y)$, kjer ima vsako polje tri možna stanja – neznano, prosto, zasedeno. Z 'neznano' so označena polja, za katera na podlagi meritev stanj ne poznamo, s 'prosto' polja, ki označujejo mesta v prostoru brez ovir ter z 'zasedeno' polja, ki označujejo neprehodna mesta v prostoru.



Slika 2: Problem načrtovanja vožnje v ozkem prehodu



Slika 3: Zemljevid laboratorija z vrisanim kvadratom, ki predstavlja avtonomni sistem

V nadaljevanju označimo polja 'neznano' in 'prosto' z vrednostjo 0 ter polja 'zasedeno' z vrednostjo 1.

4 Priprava konfiguracijskega prostora sistema

Konfiguracijski prostor \mathcal{C} sistema je prostor vseh možnih stanj sistema. V predstavljenem primeru stanje sistema vsebuje položaj (koordinati x in y v prostoru) ter orientacijo ϕ obravnavanega avtonomnega sistema. Konfiguracijski prostor \mathcal{C} je zato tridimenzionalen, stanje pa opisano kot (x, y, ϕ) . Na podlagi ovir v prostoru, prostor \mathcal{C} razdelimo v prostor \mathcal{C}_{obs} , ki vsebuje vsa prepovedana stanja sistema, ter v prostor \mathcal{C}_{free} , ki vsebuje vsa dovoljena stanja sistema. Prepovedana stanja sistema so stanja, pri katerih pride do trka sistema z ovirami v okolju.

Za potrebe določanja prostora \mathcal{C}_{obs} na podlagi opisa vplivnega prostora avtonomnega sistema definirajmo prostor \mathcal{C}_{obs}^t , ki vsebuje vsa prepovedana stanja sistema zaradi točkaste ovire v $t(0, 0)$, ter funkcijo v ravnini točkastih ovir $f_{\mathcal{O}}(x, y)$, za katero velja

$$f_{\mathcal{O}}(x, y) = \begin{cases} 1; & \text{na } (x, y) \text{ je ovira} \\ 0; & (x, y) \text{ je prosto} \end{cases} \quad (2)$$

Naj bo $f_{\mathcal{C}_{obs}^t}(x, y, \phi)$ funkcija z vrednostjo večjo od 0, če stanje (x, y, ϕ) predstavlja prepovedano stanje obravnavanega sistema zaradi točkaste ovire v $t(0, 0)$, ter z vrednostjo 0, če stanje (x, y, ϕ) predstavlja dovoljeno stanje.

Prostor \mathcal{C}_{obs} definirajmo kot podmnožico prostora \mathcal{C} , kjer velja $f_{\mathcal{C}}(x, y, \phi) > 0$. Vrednost $f_{\mathcal{C}}(x, y, \phi)$ dobimo s konvolucijo

$$\begin{aligned} f_{\mathcal{C}}(x, y, \phi) &= f_{\mathcal{C}_{obs}^t}(x, y, \phi) * f_{\mathcal{O}}(x, y) = \\ &= \int_u \int_v f_{\mathcal{C}_{obs}^t}(x - u, y - v, \phi) f_{\mathcal{O}}(x, y) du dv \end{aligned} \quad (3)$$

Pri nadaljevanju se omejimo na interval $(x_{min} < x < x_{max}, y_{min} < y < y_{max})$, kjer so x_{min} , x_{max} , y_{min} in y_{max} robne točke prostora, opisanega z zemljevidom.

Za namene postopka iskanja poti, opisanega v nadaljevanju, potrebujemo predstavitev dobljenega konfiguracijskega podprostora \mathcal{C}' s končno mnogo stanji. Podprostor \mathcal{C}' diskretiziramo v $m \times n \times d$ stanj, ter za vsako stanje določimo pripadnost prostoru \mathcal{C}_{obs} ali prostoru \mathcal{C}_{free} . Diskretiziramo tudi prepovedani prostor točke $\mathcal{C}_{obs}^t(x, y, \phi)$ na intervalu $(x_{min}^r < x < x_{max}^r, y_{min}^r < y < y_{max}^r)$, kjer so x_{min}^r , x_{max}^r , y_{min}^r in y_{max}^r robne točke največjega okvirja robota pri vseh možnih orientacijah ϕ . Zagotovimo tudi, da imajo vsi prostori izven podanih intervalov vrednost 0.

Konvolucijo zapisano z integrali v zveznem prostoru zapišimo v diskretni obliki z vsotami

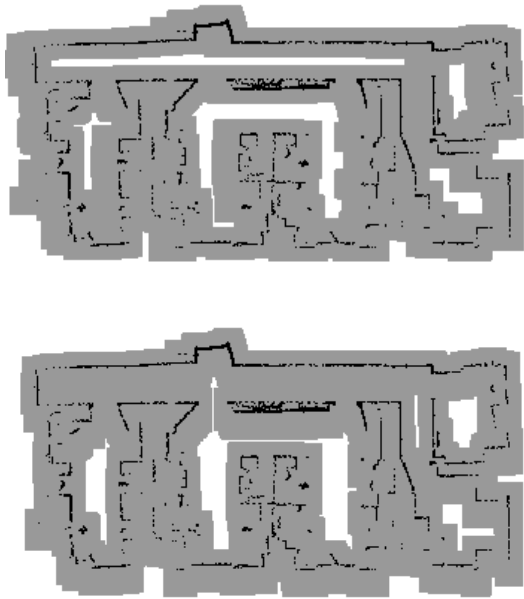
$$\begin{aligned} f_{\mathcal{C}'}(x', y', \phi') &= \\ \sum_{u'=x_{min}^r}^{x_{max}^r} \sum_{v'=y_{min}^r}^{y_{max}^r} f_{\mathcal{C}_{obs}^t}(x' - u', y' - v', \phi') f'_{\mathcal{O}}(x', y'). \end{aligned} \quad (4)$$

Funkcija $f_{\mathcal{C}'}(x', y', \phi')$ podaja rezultat preverjanja trka sistema z ovirami v prostoru za vsako diskretno stanje sistema (x', y', ϕ') . Za namene iskanja poti smo velikosti prostora m in n izenačili z velikostjo bitne slike, ki predstavlja osnovni zemljevid okolja (slika 3) ter za d izbrali vrednost 16, oziroma 16 možnih orientacij sistema. Izbira števila stanj predstavlja kompromis med natančno predstavitvijo osnovnega (zveznega) prostora, kjer želimo čim več stanj, ter porabo spomina in hitrostjo izvajanja algoritma, kjer število stanj direktno vpliva na porabo spomina in hitost izvajanja algoritma.

5 Iskanje poti

Iskanje poti je postopek, s katerim najdemo pot med točkama oz. stanjema sistema $A(x_A, y_A, \phi_A)$ in $B(x_B, y_B, \phi_B)$, ki ustreza pogoju, da mobilni sistem pri premiku po tej poti ne trči v nobeno od znanih ovir v zemljevidu. Iskanje poti tako predstavlja osnovni korak pri načrtovanju poti za namene vodenja avtonomnega sistema v prostoru.

Za iskanje poti smo uporabili algoritem informiranega iskanja v grafu A^* [2], ki najde optimalno pot

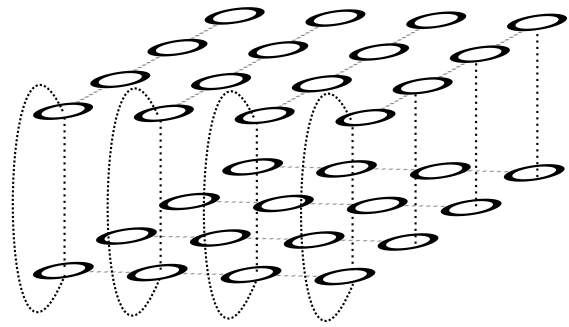


Slika 4: Prikaz rezultatov $f_{C'}(x', y', \phi')$ za $\phi' = 1$ (zgoraj) in $\phi' = 5$ (spodaj). Bel prostor predstavlja stanja, ki pripadajo prostoru C'_{free}

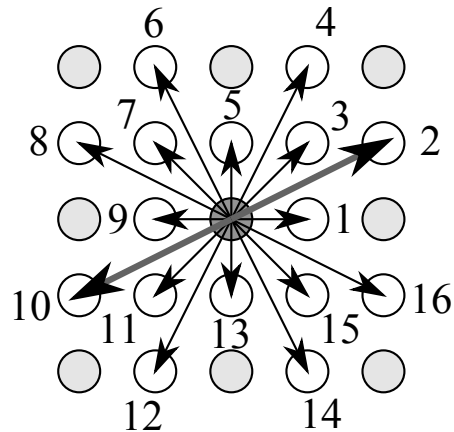
v grafu v najmanj korakih. Algoritem A* išče najugodnejšo pot iz vozlišča A do vozlišča B tako, da sproti beleži strošek poti $f(C)$ od začetnega vozlišča A do trenutnega vozlišča C ter optimistično oceno stroška $h(C)$ od trenutnega vozlišča C do končnega vozlišča B . Ocena stroška predstavlja dopustno hevrstiko, za katero velja $h(A) \leq f(B)$, oziroma da je ocenjena pot vedno krajša ali enaka najkrajši možni poti. Vključitev ocene poti do cilja B daje algoritmu lastnost informiranosti, saj so nadaljne koraki izvajanja algoritma odvisni od hevrstične ocene poti do cilja.

Za namene izvajanja algoritma sta potrebna dva seznama vozlišč – odprti seznam, ki vsebuje vozlišča v obdelavi, ter zaprti seznam, ki vsebuje vozlišča z že določenimi najmanjšimi razdaljami od začetnega vozlišča $f(C)$ ter ocenami skupne razdalje med A in B , $g(C) = f(C) + h(C)$. Postopek začnemo tako, da v odprti seznam dodamo začetno vozlišče A . Algoritem se izvaja iterativno po naslednjih korakih:

- za trenutno vozlišče C iz odprtega seznama izberi vozlišče z najmanjšo oceno skupne razdalje $g(C)$, ga prestavi v zaprti seznam ter preveri, če je vozlišče končno (če je, končaj postopek)
- preveri vse možne poti iz trenutnega vozlišča C do povezanih vozlišč D , pri tem preskoči poti, ki vodijo v vozlišča, ki pripadajo prostoru C'_{obs} . Za vozlišče D na podlagi stroška poti izračunaj novo oceno skupne razdalje $g(D)$ ter preveri, če ta razdalja predstavlja najkrajšo pot od A do B preko vozlišča D – če je pogoj izpolnjen,



Slika 5: Del povezanega grafa možnih poti

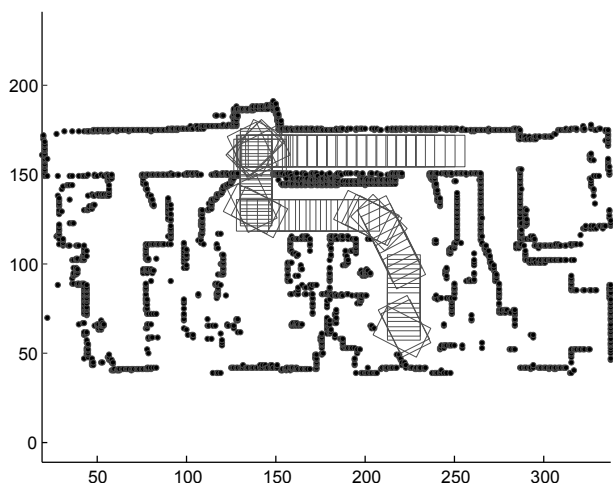


Slika 6: Možne poti na enem nivoju vozlišč z označenimi orientacijami ϕ'

shrani to razdaljo ter dodaj vozlišče D na seznam odprtih vozlišč.

Graf smo zgradili na podlagi rezultatov diskretizacije konfiguracijskega prostora – določili smo $m \times n \times d$ vozlišč, organiziranih kot d nivojev mrež z $m \times n$ vozlišči ter poimenovanih z ustreznimi koordinatami (x', y', ϕ') . Vozlišča v nivoju ϕ' so povezana z istoležnimi vozlišči v nivojih $\phi' - 1$ ter $\phi' + 1$ (prostor je v tretji dimenziji krožno zaprt, zato velja, da je predhodnik prvega nivoja zadnji nivo ter obratno, naslednik zadnjega nivoja je prvi nivo, kot prikazano na sliki 5). Na posameznih nivojih pa so dovoljene le smeri, ki ustrezajo orientaciji sistema v stanju ϕ' . Npr. na sloju 1, sta dovoljena premika le 1 in 9 (premika naprej in nazaj), na sloju 2 pa premika 2 in 10 (prikazano na sliki 6), itd. Iz vsakega vozlišča so tako na voljo 4 poti – spremembi orientacije v levo ali desno smer brez premika ter premika naprej in nazaj brez spremembe orientacije. Vsakemu od vozlišč smo tudi določili pripadnost prostoroma C'_{obs} in C'_{free} .

Vsaki izmed možnih poti iz vozlišča $G = (G_x, G_y, G_\phi)$ (do vozlišč $H = (H_x, H_y, H_\phi)$) smo nato določili stro-



Slika 7: Najdena pot v zemljevidu

šek, ki je enak vsoti

$$d(G, H) = \sqrt{(G_x - H_x)^2 + (G_y - H_y)^2} + \min\{|H_\phi - G_\phi|, 2\pi - |H_\phi - G_\phi|\} + d_{dir}(G, H), \quad (5)$$

kjer $d_{dir}(G, H)$ predstavlja strošek vožnje nazaj ali obračanja na mestu.

Hevristični prispevek oz. oceno stroška poti od vmesnega vozlišča C do cilja B smo določili na podlagi evklidske razdalje med stanjema sistema, ki pripadata vozliščema C in B .

6 Rezultati

Algoritem smo realizirali v okolju Matlab ter uporabili vnaprej zgrajen zemljevid Laboratorija za avtonomne mobilne sisteme na Fakulteti za elektrotehniko v Ljubljani (slika 3). S predstavljenim postopkom lahko uspešno generiramo tudi zapletene poti (slika 7), ki so posledica omejenega prostora za premikanje avtonomnega sistema v laboratoriju. Za izračun poti, prikazane na sliki 7, je neoptimiran algoritem v okolju Matlab na podlagi predpripravljenega diskretiziranega konfiguracijskega prostora z dimenzijami $355 \times 224 \times 16$ skupno preiskal 30.910 vozlišč. Za pripravo konfiguracijskega prostora C' oz. za vnaprejšnji izračun vrednosti funkcije $f_{C'}(x', y', \phi')$ je implementiran algoritem potreboval 119 sekund, za iskanje poti v tako pripravljenem prostoru pa 3,2 sekund. Z optimizacijo algoritma in implementacijo v programskem jeziku C pričakujemo znatno pohitritev obeh postopkov.

7 Zaključek

V članku je predstavljen koncept za osnoven postopek pri iskanju poti z dodatnimi omejitvami pri izvajanju manevrov zaradi ovir v prostoru. Za uspešno implementacijo na realnem avtonomnem sistemu

je potrebno upoštevati še dinamične parametre sistema pri premikanju – potrebna sta glajenje načrtane poti ter izračun profilov hitrosti vzdolž poti. Implementiran algoritem v okolju Matlab je potrebno pri prenosu v sistem vodenja na avtonomnem mobilnem sistemu tudi optimizirati z namenom zmanjšanja porabe procesorskega časa in podatkovnega spomina.

Literatura

- [1] G. Klančar, L. Teslić, I. Škrjanc, Mobile-robot pose estimation and environment mapping using an extended kalman filter, *International Journal of Systems Science*, 2013.
- [2] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006. Available at <http://planning.cs.uiuc.edu/>.