

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-104376-104853

PLÁNOVANIE TRAJEKTÓRIE PRE ROBOT S
KINEMATICKOU ŠTRUKTÚROU SCARA
DIPLOMOVÁ PRÁCA

2024

Bc. Michal Bíro

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Evidenčné číslo: FEI-104376-104853

PLÁNOVANIE TRAJEKTÓRIE PRE ROBOT S
KINEMATICKOU ŠTRUKTÚROU SCARA
DIPLOMOVÁ PRÁCA

Študijný program:	Robotika a kybernetika
Názov študijného odboru:	kybernetika
Školiace pracovisko:	Ústav robotiky a kybernetiky
Vedúci záverečnej práce:	Ing. Michal Dobiš, PhD..

Bratislava 2024

Bc. Michal Bíro

SÚHRN

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Študijný program:	Robotika a kybernetika
Autor:	Bc. Michal Bíro
Diplomová práca:	Plánovanie trajektórie pre robot s kinematickou štruktúrou SCARA
Vedúci záverečnej práce:	Ing. Michal Dobiš, PhD..
Miesto a rok predloženia práce:	Bratislava 2024

Kľúčové slová: SCARA, plánovanie trajektórie, RRT

ABSTRACT

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY

Study Programme:	Robotics and cybernetics
Author:	Bc. Michal Bíro
Master's thesis:	Trajectory planning for a SCARA kinematic structure robot
Supervisor:	Ing. Michal Dobiš, PhD..
Place and year of submission:	Bratislava 2024

Abstract

Keywords: SCARA, trajectory planning, RRT

Pod'akovanie

Chcel by som sa pod'akovať môjmu vedúcemu práce ...

Obsah

Úvod	1
1 Rozbor problému	2
1.1 SCARA - kinematická štruktúra	2
1.1.1 2 stupne voľnosti	2
1.1.2 3 stupne voľnosti	2
1.2 Algoritmy plánovania trajektórie	2
1.2.1 RRT	2
1.2.2 RRT*	3
1.2.3 RRT - connect (prepojený)	3
1.2.4 Potenciálové pole	3
Záver	5
Zoznam použitej literatúry	6
Prílohy	6
A Štruktúra elektronického nosiča	7
B Algoritmus	8
C Výpis sublimu	9

Zoznam obrázkov a tabuliek

Obrázok 1.1	Vizualizácia princípu RRT algoritmu	2
Obrázok 1.2	Vplyv počtu iterácií na výslednú trajektóriu [<empty citation>] . .	3
Obrázok 1.3	Prepojenie 2 stromov [<empty citation>]	4
Obrázok 1.4	Potenciálové pole [<empty citation>]	4

Zoznam algoritmov

B.1	Vypočítaj $y = x^n$	8
-----	-------------------------------	---

Zoznam výpisov

C.1 Ukážka sublime-project	9
--------------------------------------	---

Úvod

1 Rozbor problému

1.1 SCARA - kinematická štruktúra

1.1.1 2 stupne voľnosti

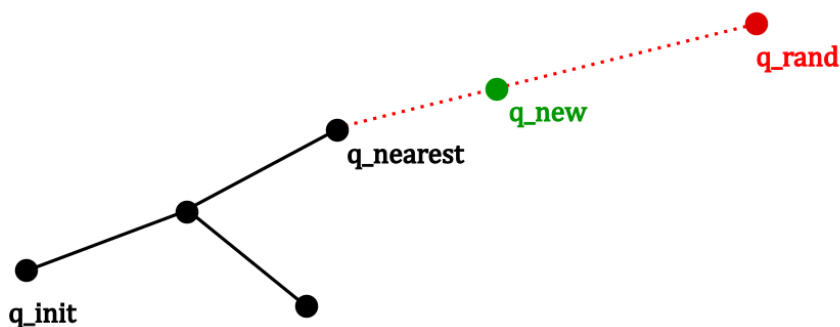
1.1.2 3 stupne voľnosti

1.2 Algoritmy plánovania trajektórie

1.2.1 RRT

Algoritmus RRT – z anglického Rapidly-exploring Random Tree, by sa dal do slovenčiny preložiť ako rýchlo rastúci náhodný strom. Algoritmus je v oblasti robotiky veľmi rozšírený a obľúbený vďaka svojej schopnosti rýchlo prehľadávať vysoko dimenzionálny konfiguračný priestor, v ktorom zohľadňuje prekážky v priestore ako aj dynamiku telesa.

Ide o algoritmus založený na prehľadávaní konfiguračného priestoru C , kde sa v iteráciách vytvárajú náhodné uzly - q , ktorých spájaním vznikajú nové potenciálne cesty. Každý uzol q reprezentuje pozíciu a orientáciu telesa v 2D alebo 3D priestore [1]. Pri plánovaní trajektórie je generovaný kontinuálny rad uzlov (obr. 1.1), ktorý začína v počiatočnom stave q_{init} , a postupne sa rozrastá až dosiahne koncový stav q_{goal} . Generovanie stromu prebieha v iteráciách kedy sa vždy vygeneruje náhodná konfigurácia q_{rand} a nájde sa k nej najbližší uzol patriaci stromu. Od daného uzla je následne vo vzdialenosti d od uzla $q_{nearest}$ vytvorený nový uzol q_{new} . Proces rozrastania stromu sa končí v momente ak sa q_{new} nachádza v okolí cieľovej konfigurácie q_{goal} , ktoré je definované vzdialenosťou d . Ak neuvažujeme voľný konfiguračný priestor C_{free} treba v procese generovania taktiež overovať kolíziu s danými prekážkami v priestore - C_{obs} . Novo generovaný uzol stromu sa nesmie nachádzať v priestore prekážky - C_{obs} , a ani cesta medzi dvoma susednými uzlami nesmie kolidovať s prekážkou. Pokiaľ uzol spĺňa tieto podmienky, je zaradený do štruktúry stromu, v opačnom prípade je vyradený a proces pokračuje ďalej.



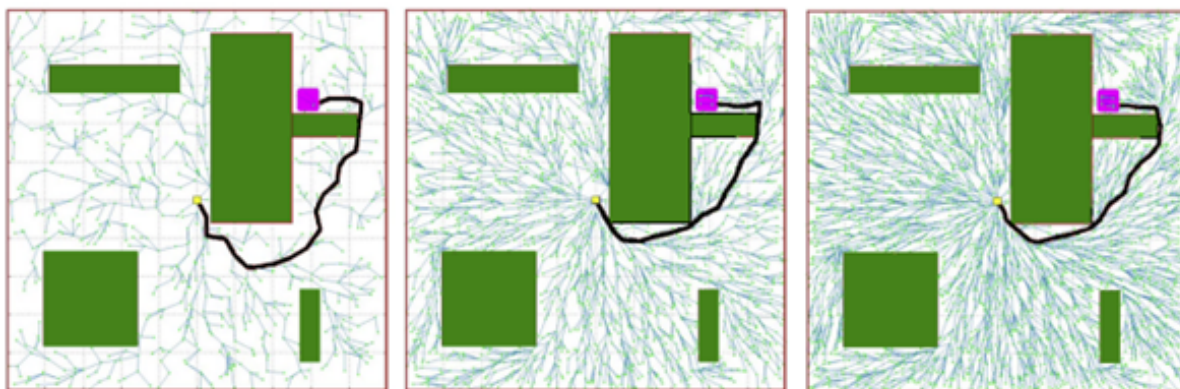
Obr. 1.1: Vizualizácia princípu RRT algoritmu

Ide o neoptimálne riešenie, keďže body v konfiguračnom priestore sú generované náhodne

a výsledný tvar trajektórie vzniká ako najkratšia cesta medzi nami vygenerovanými bodmi. Na zlepšenie trajektórie existuje viacero možností, ktoré ponúkajú suboptimálne riešenia problému, jedným z nich je napríklad RRT*.

1.2.2 RRT*

Jedná sa o modifikáciu RRT algoritmu, ktorej cieľom je v porovnaní s pôvodným RRT nájsť kratšiu trajektóriu. Hlavnou úpravou oproti jednoduchému RRT algoritmu je výpočet vzdialenosti z počiatočného do aktuálneho uzlu a následné overenie, či neexistuje v okolí uzol, ktorého vzdialenosť by bola menšia ako vzdialenosť momentálneho prepojenia. Ak táto situácia nastane, algoritmus vyberie prepojenie uzlov, ktoré vytvorí kratšiu trasu. Algoritmy sa tiež líšia ukončovacou podmienkou, kde pre RRT* je určený jasný počet iterácií. Od počtu iterácií závisí výsledný tvar trajektórie, kde so zvyšujúcim sa počtom je algoritmus schopný nájsť kratšie a plynulejšie trasy (obr. 1.2).



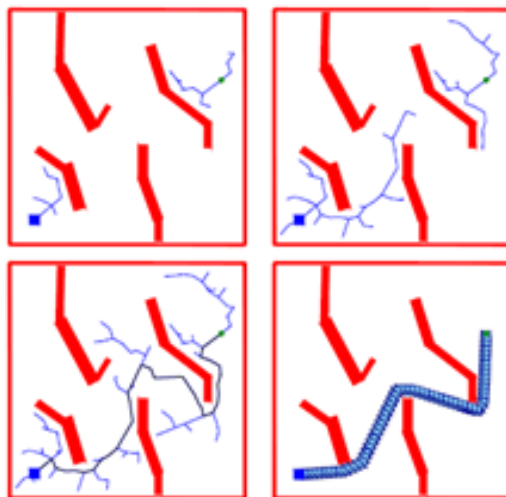
Obr. 1.2: Vplyv počtu iterácií na výslednú trajektóriu [empty citation]

1.2.3 RRT - connect (prepojený)

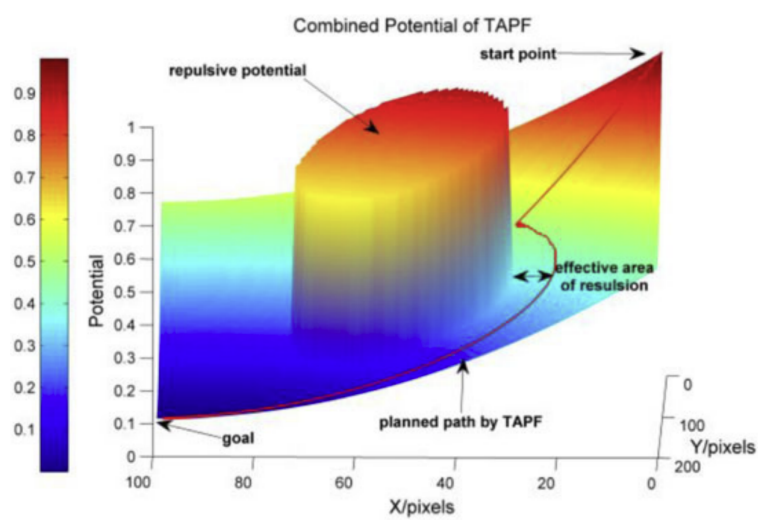
Ďalšou z modifikácií je algoritmus RRT Connect (spojenie). Algoritmus v procese rozširovania stromovej štruktúry vytvára 2 stromy z počiatočnej a koncovej konfigurácie, ktoré sa šíria v konfiguračnom priestore až dokým nedôjde k ich spojeniu a tak vytvoreniu cesty (obr. 1.3).

1.2.4 Potenciálové pole

Plánovanie trajektórie na základe potenciálového poľa využíva koncept odpudivých a príťažlivých polí. Príťažlivé pole generuje veľmi nízke hodnoty so stredom v cieľovom bode, ktoré sa so zväčšujúcou sa vzdialenosťou od cieľa zvyšujú. Odpudivé pole zase naopak generuje veľmi vysoké hodnoty v okolí prekážok v priestore. Kombináciou daných dvoch polí dostávame tzv. potenciálové pole (obr. 1.4) so silným sklonom ku cieľu tak aby generovaná trajektória mala tendenciu vyhýbať sa prekážkam.



Obr. 1.3: Prepojenie 2 stromov [**<empty citation>**]



Obr. 1.4: Potenciálové pole [**<empty citation>**]

Záver

Conclusion is going to be where?

Here.

Prílohy

A	Štruktúra elektronického nosiča	7
B	Algoritmus	8
C	Výpis subline	9

A Štruktúra elektronického nosiča

/CHANGELOG.md

- file describing changes made to FEIstyle

/example.tex

- main example *.tex* file for diploma thesis

/example_paper.tex

- example *.tex* file for seminar paper

/Makefile

- simply Makefile – build system

/fei.sublime-project

- is project file with build in Build System for Sublime Text 3

/img

- folder with images

/includes

- files with content

/bibliography.bib

- bibliography file

/attachmentA.tex

- this very file

B Algoritmus

Algoritmus B.1 Vypočítaj $y = x^n$

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

$y \leftarrow 1$

if $n < 0$ **then**

$X \leftarrow 1/x$

$N \leftarrow -n$

else

$X \leftarrow x$

$N \leftarrow n$

end if

while $N \neq 0$ **do**

if N is even **then**

$X \leftarrow X \times X$

$N \leftarrow N/2$

else { N is odd }

$y \leftarrow y \times X$

$N \leftarrow N - 1$

end if

end while

C Výpis sublime

```
../.. ./ fei .sublime-project
```

Výpis C.1: Ukážka sublime-project