

**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

<b>Úvod</b>	<b>1</b>
<b>1 Rozbor problému</b>	<b>2</b>
1.1 SCARA . . . . .	2
1.1.1 SCARA - 2 stupne voľnosti . . . . .	3
1.1.2 SCARA - 3 stupne voľnosti . . . . .	3
1.2 Algoritmy plánovania trajektórie . . . . .	3
1.2.1 RRT . . . . .	4
1.2.2 RRT* . . . . .	5
1.2.3 RRT - connect (prepojený) . . . . .	6
1.2.4 Potenciálové pole . . . . .	6
<b>Záver</b>	<b>8</b>

# Zoznam obrázkov a tabuliek

Obrázok 1.1	SCARA robot - smery pohybov[] . . . . .	2
Obrázok 1.2	Model robota - Rviz . . . . .	4
Obrázok 1.3	Vizualizácia princípu RRT algoritmu . . . . .	5
Obrázok 1.4	Vplyv počtu iterácií na výslednú trajektóriu [] . . . . .	5
Obrázok 1.5	Prepojenie 2 stromov [] . . . . .	6
Obrázok 1.6	Potenciálové pole [] . . . . .	7





# **Zoznam algoritmov**

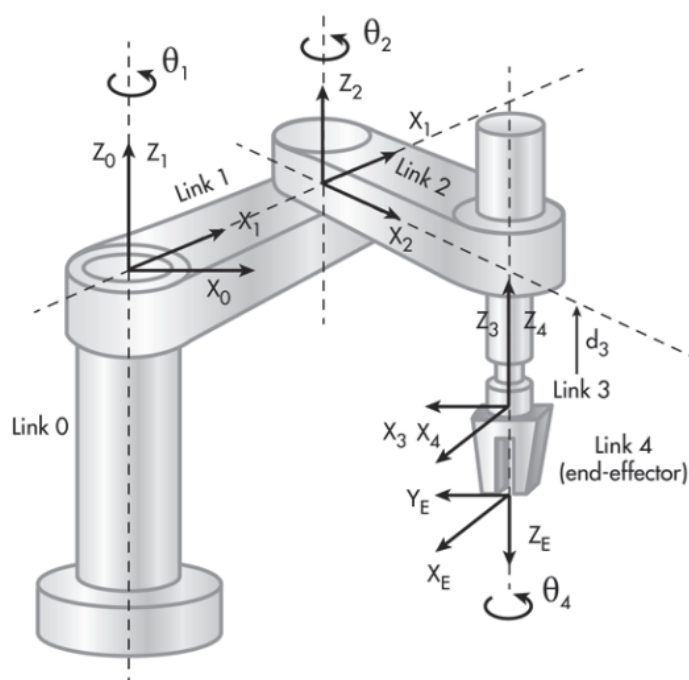
# **Zoznam výpisov**

# Úvod

# 1 Rozbor problému

## 1.1 SCARA

Scara robot je typ priemyselného robotického systému, ktorý bol vyvinutý na vykonávanie presných a opakujúcich sa úloh. Navrhol ho v roku 1979 vedec Hiroshi Makino z Yamanashi Univerzity[1]. Názov SCARA je akronym (Selective Compliance Assembly Robot Arm), ktorý značí, že vertikálna poddajnosť (compliance) je väčšia ako poddajnosť v horizontálnom smere. Najčastejšou úlohou je presun objektu z jedného miesta na druhé. Scara roboty majú zväčša 4 stupne voľnosti, a svojou konštrukciou pripomínajú ľudskú ruku(obr. 1.2). Ide o 3 rotačné



Obr. 1.1: SCARA robot - smery pohybov[1]

a 1 posúvny kĺb, ktoré zabezpečujú pohyb robota vo všetkých osiach  $x$ ,  $y$  a  $z$ . Umiestneným otáčavých kĺbov v jednej rovine je pohyb robota v porovnaní s robotickými ramenami so 6 stupňami voľnosti, značne znevýhodnený. Taktiež nosnosť týchto robotov je značne obmedzená, z dôvodu konštrukčných parametrov ako je nosnosť ložiska. Kinematická štruktúra SCARA robotov však prináša aj výhody, pre ktoré sú tieto roboty veľmi obľúbené a často používané.

Výhody:

- presnosť
- rýchlosť

- nízka hmotnosť
- jednoduché ovládanie
- menšie rozmery
- nižšia cena

Tieto vlastnosti sú dôvodom prečo SCARA roboty nachádzajú široké uplatnenie a to hlavne v montáži, balení, manipulácii s chemikáliami a oblastiach ako potravinárska výroba, laboratória, farmaceutický priemysel, automobilový priemysel atď.

V tejto práci sa nebudeme venovať len klasickej štruktúre SCARA robota so 4 stupňami voľnosti ale budeme skúmať a overovať riešiteľnosť daného problému aj pomocou iných kinematických štruktúr SCARA. Keďže posuvný pohyb z smere osi  $z$  zabezpečuje dopravník, robot SCARA nepotrebuje posuvný kĺb, čím sa nám eliminuje 1 stupeň voľnosti. Z dôvodu zníženia ceny robota budeme tiež testovať, či je pre náš problém potrebný robot s 3 stupňami voľnosti (3 rotačné kĺby) alebo vieme štruktúru ešte viac zjednodušiť.

### **1.1.1 SCARA - 2 stupne voľnosti**

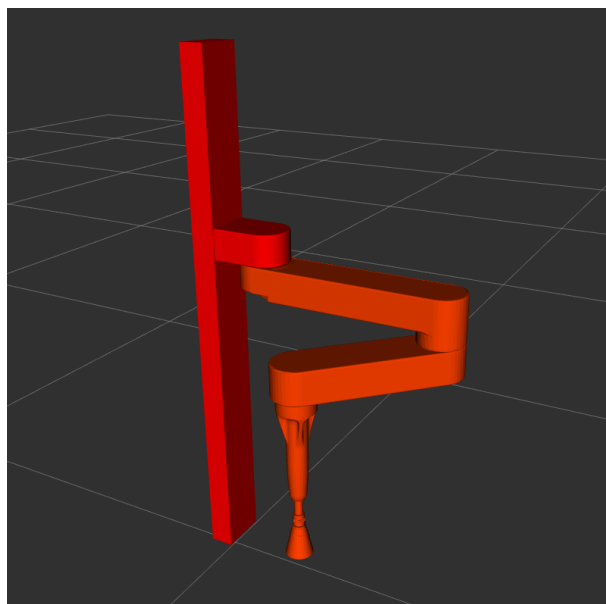
Prvou testovanou štruktúrou bude SCARA s 2 rotačnými kĺbmi - 2 stupne voľnosti. V tejto kinematickej štruktúre sa budú otáčať 2 ramená robota a nástroj na uchytenie objektu, ktorý sa nachádza na konci kinematického reťazca bude fixný. Na obrázku ?? môžeme vidieť model nášho ramena. Pri tejto konfigurácii je dôležité overiť na viacerých objektoch, rôznych tvarov a rozmerov, aké sú limitácie a schopnosti robota preniesť objekt zo štartovacej konfigurácie do cieľovej. Problém môže nastať hlavne pri väčších podlhovastých objektoch, kedy robot nebude vedieť v rámci svojich 2 rotácií ramien otočiť objekt do takej pozície aby dosiahol cieľovú konfiguráciu alebo nebude vedieť obísť stĺp na ktorom je rameno upevnené.

### **1.1.2 SCARA - 3 stupne voľnosti**

Druhou štruktúrou, ktorú budeme testovať sa bude líšiť od prvej pridaním ďalšieho rotačného kĺbu, ktorý nám umožní otáčať nástrojom na uchytenie objektu. Pri tejto konfigurácii by mal byť robot schopný vždy nájsť trajektóriu na presun objektu zo štartu do cieľa. Tu budeme sledovať hlavne parametre a čas, za ktorý túto trajektóriu prejde a jej efektivitu. Tieto dáta budú potrebné na zhodnotenie, či je z hľadiska financií lepšie zvoliť väčšie náklady na efektívnejšieho robota v porovnaní s lacnejším robotom s väčšími obmedzeniami.

## **1.2 Algoritmy plánovania trajektórie**

Úloha plánovania trajektórie rohráva pri našom probléme veľkú rolu. Je potrebné zvoliť správny algoritmus aby sme vedeli dobre porovnať naše 2 kinematické štruktúry robota.



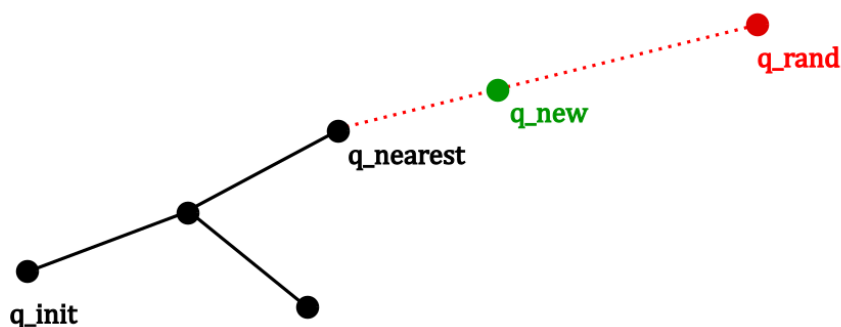
Obr. 1.2: Model robota - Rviz

### 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 prehl'adávať vysoko dimenzionálny konfiguračný priestor, v ktorom zohľadňuje prekážky v priestore ako aj dynamiku telesa.

Ide o algoritmus založený na prehl'adávaní 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 []. Pri plánovaní trajektórie je generovaný kontinuálny rad uzlov (obr. 1.3), 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.

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

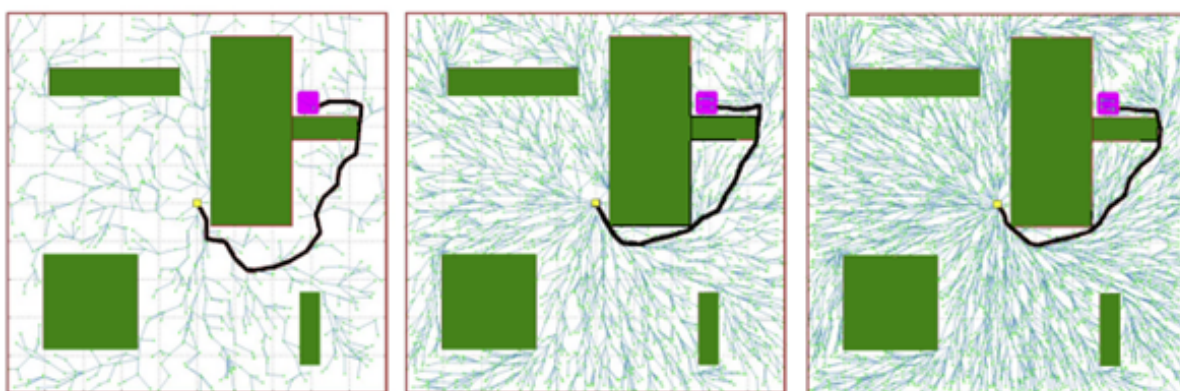


Obr. 1.3: Vizualizácia princípu RRT algoritmu

zlepšenie trajektórie existuje viacero možností, ktoré ponúkajú suboptimálne riešenia problému, jedným z nich je napríklad RRT\*.

### 1.2.2 RRT\*

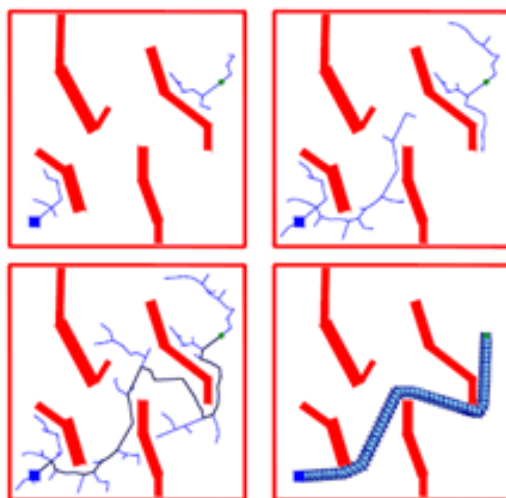
Jedná sa o modifikáciu RRT algoritmu, ktorej cieľom je v porovnaní s pôvodným RRT nájsť kratšiu trajektóriu. Hlavnou úpravou oproti jednoduchému RRT algoritmu je výpočet vzdialenosti z počiatočného do aktuálneho uzlu a následné overenie, či neexistuje v okolí uzol, ktorého vzdialenosť by bola menšia 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. 1.4).



Obr. 1.4: Vplyv počtu iterácií na výslednú trajektóriu []

### 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. 1.5). Výhodou tohoto algoritmu je fakt, že nájdenie trajektórie býva často ľahšie dosiahnuteľný jedným z 2 možných smerov pohybu medzi dvoma bodmi, čo závisí od prekážok v priestore -  $C_{obs}$ . Použitím RRT - connect zvýšime pravdepodobnosť nájdenia trajektórie od počiatočného bodu ku koncovému.

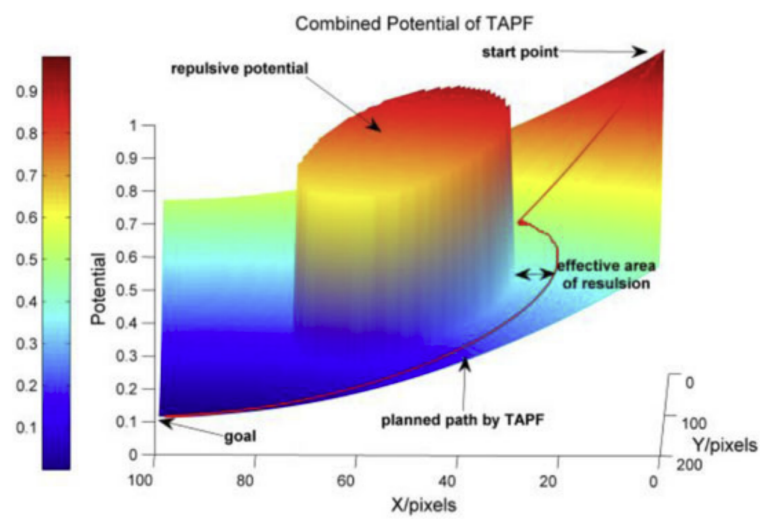


Obr. 1.5: Prepojenie 2 stromov []

### 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. 1.6) so silným sklonom ku cieľu tak aby generovaná trajektória mala tendenciu vyhýbať sa prekážkam.





Obr. 1.6: Potenciálové pole []

# **Záver**

Conclusion is going to be where?

Here.