

Generátor inverznej kinematiky

Genco Gabriel, Komenská Daniela

Abstrakt

Tento článok reprezentuje implementáciu a návrh generátora inverznej kinematiky (IK) pre robotické rameno KUKA LBR iiwa. Pozornosť sa venuje postupu generovania náhodných polôh v rámci pracovného priestoru robota, výpočtu inverznej kinematiky pre dané ciele a vizualizačnému zobrazeniu výsledkov pomocou simulačného prostredia PyBullet. Cieľom modelu projektu je overenie presnosti a spoľahlivosti výpočtov inverznej kinematiky a zároveň poskytnutie flexibilného nástroja pre testovanie algoritmov riadenia robota.

Kľúčové slová

Robotické rameno, Inverzná kinematika, generovania polôh, pracovný priestor, Denavit-Hartenbergova metóda, KUKA LBR iiwa.

I. ÚVOD DO PROBLEMATIKY

Inverzná kinematika (IK) predstaviuje jeden zo základných problémov v oblasti robotiky. Jej cieľom je nájsť takú konfiguráciu kĺbových uhlov robota, ktorá zabezpečí dosiahnutie požadovanej pozície a orientácie koncového efektora v trojrozmernom priestore. V porovnaní s priamou kinematikou ide o oveľa zložitejšiu úlohu – najmä v prípade robotov s vyšším počtom stupňov voľnosti (DoF), kde môžu existovať viaceré, nekonečne veľa, alebo naopak žiadne riešenia. To robí z inverznej kinematiky výpočtovo náročný a technicky významný problém. [1], [6]

Moderné robotické systémy preto vyžadujú efektívne a spoľahlivé riešenia IK, ktoré umožňujú prepojenie medzi požadovanou polohou nástroja a zodpovedajúcimi kĺbovými konfiguráciami. V tejto práci sa zameriavame na robotické rameno **KUKA LBR iiwa**, ktoré disponuje **7 stupňami voľnosti** a tým predstavuje **redundantný systém** – teda systém, ktorý môže dosiahnuť tú istú koncovú polohu viacerými spôsobmi. Táto vlastnosť umožňuje optimalizáciu pohybu z hľadiska rôznych kritérií, zároveň však zvyšuje zložitosť výpočtov.

Cieľom projektu je navrhnuť a implementovať **generátor inverznej kinematiky**, ktorý umožní:

- **generovanie náhodných cieľových polôh a orientácií** koncového efektora v rámci pracovného priestoru robota,
- **výpočet inverznej kinematiky** pomocou dostupných algoritmov (napr. interný IK riešič PyBulletu alebo numerické metódy),
- **overenie výsledkov pomocou vizualizácie v simulátore PyBullet.**

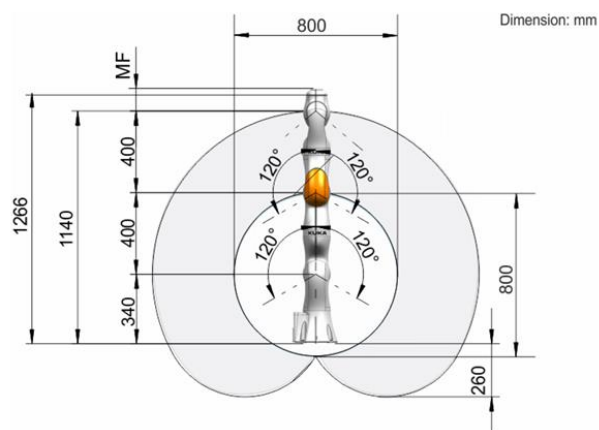
Výsledný generátor slúži ako testovacia platforma pre verifikáciu a analýzu výpočtov inverznej kinematiky v rôznych scenároch. Môže byť využitý nielen na výučbové účely, ale aj ako podporný nástroj pri vývoji a testovaní algoritmov riadenia robotických manipulátorov, plánovania pohybu alebo optimalizácie trajektórií.

II. MODEL ROBOTICKÉHO RAMENA

Robotické rameno **KUKA LBR iiwa (intelligent industrial work assistant)** je pokročilý kolaboratívny manipulátor navrhnutý pre bezpečnú interakciu s ľuďmi a flexibilné nasadenie v priemyselnom prostredí. Radí sa medzi **ľahký typ robot (Lightweight Robot)** so **7 stupňami voľnosti**, čo mu umožňuje vysokú manévrovateľnosť a schopnosť pracovať aj v stiesnených alebo dynamicky sa meniacich priestoroch. [3]

Maximum reach	800 mm
Rated payload	7 kg
Maximum payload	-
Pose repeatability (ISO 9283)	± 0.1 mm
Number of axes	7
Mounting position	Floor; Ceiling; Wall
Footprint	-
Weight	approx. 23.9 kg

Obr. 1. Technické parametre robota



Obr. 2. Zobrazenie pracovného priestoru

Robot je určený na úlohy, kde je dôležitá presnosť, citlivosť na kontakt a flexibilné riadenie. Vďaka snímačom síl a momentov v každom kĺbe vie reagovať na vonkajšie podnety a v prípade kontaktu bezpečne zastaviť. To je kľúčové pri spolupráci s človekom (tzv. human-robot collaboration, HRC).

III. TEORETICKÝ ROZBOR

KINEMATICKÉ RIEŠENIE MODELU ROBOTA

Kinematika opisuje pohyb robotického ramena bez ohľadu na sily, ktoré ho spôsobujú. V robotike rozlišujeme dva hlavné typy kinematiky:

- **Priama kinematika (FK)** určuje polohu a orientáciu koncového efektora na základe známych kĺbových uhlov. Výpočet sa robí pomocou transformačných matic medzi jednotlivými článkami robota.
- **Inverzná kinematika (IK)** hľadá opačný vzťah – teda hodnoty kĺbov, ktoré sú potrebné na dosiahnutie požadovanej polohy efektora. Tento problém môže mať viacero riešení, žiadne riešenie, alebo nekonečne veľa riešení – najmä pri robotoch s viacerými stupňami voľnosti.

Na opis geometrie robota sa často používa Denavit-Hartenbergova metóda alebo modelovanie pomocou homogénnych transformačných matic.

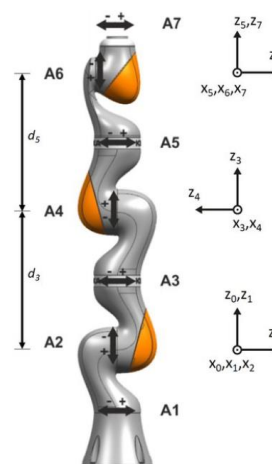
Pri riešení IK je dôležité zohľadniť pracovný priestor robota (dosiahnuteľné pozície) a singularity, ktoré môžu spôsobiť stratu riaditeľnosti. Manipulátory s viac ako 6 stupňami voľnosti (napr. 7 DoF) sú kinematicky redundantné, čo umožňuje vybrať z viacerých riešení to najvhodnejšie napríklad z hľadiska optimalizácie pohybu alebo vyhýbania sa prekážkam. [2]

A. Priama kinematika

Priama kinematika (Forward Kinematics - FK) popisuje vzťah medzi kĺbovými uhlami a pozíciou/orientáciou koncového efektora. Tento vzťah je určený na základe geometrickej konfigurácie robota a vyjadruje sa pomocou transformačných matic medzi jednotlivými kĺbmi, často s využitím Denavit-Hartenbergových (DH) parametrov alebo pomocou transformačných reťazcov definovaných v URDF modeli. Výsledkom výpočtu je homogénna transformačná matica, ktorá určuje polohu a orientáciu koncového efektora vzhľadom na základňu robota. [3]

Link i	θ_i	d_i	a_i	α_i
1	q_1	0	0	0
2	q_2	0	0	$-\pi/2$
3	q_3	0.42m	0	$\pi/2$
4	q_4	0	0	$\pi/2$
5	q_5	0.40m	0	$-\pi/2$
6	q_6	0	0	$-\pi/2$
7	q_7	0	0	$\pi/2$

Obr. 3. D-H parametre robota KUKA LRC iiwa



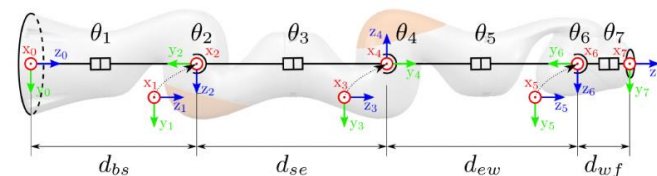
Obr. 4. Označenia a orientácie kĺbov robota

B. Inverzná kinematika

Inverzná kinematika (Inverse Kinematics – IK) rieši opačný problém – hľadá súbor kĺbových uhlov, ktoré vedú k požadovanej polohe a orientácii efektora. Pre 7-stupňový robot je riešenie zložitejšie než pri robotoch so 6 DoF, pretože existuje nekonečne veľa možných riešení. Na riešenie sa často používajú:

- **Numerické metódy** (napr. iteratívne riešiče),
- **Optimalizačné prístupy** (s penalizáciou prekážok alebo nežiadaných konfigurácií),
- alebo vstavané IK funkcie v simulačných knižniciach (napr. `calculateInverseKinematics()` v PyBullete).

Vzhľadom na to, že numerické riešenie je výpočtovo náročnejšie a pomalšie kvôli svojej iteratívnej povahe, uprednostňuje sa analytické riešenie. Toto riešenie vychádza z analytických výrazov alebo zo zjednodušenia problému na riešenie polynómu najviac štvrtého stupňa. Pri návrhu robotických ramien sa preto kladie dôraz na vhodné zarovnanie osí tak, aby hodnoty uhlov α_i boli čo najčastejšie 0 alebo $\pm 90^\circ$, čo zabezpečuje možnosť použitia uzavretého riešenia. Výpočet inverznej kinematiky vychádza zo znalosti DH parametrov. [5]



Obr. 5. Znázornenie zarovnania súradnicových sústav spolu s rozmermi

V tomto prípade je rameno namontované vertikálne.

i	α_i	d_i	a_i	θ_i
1	$-\frac{\pi}{2}$	d_{bs}	0	θ_1
2	$\frac{\pi}{2}$	0	0	θ_2
3	$\frac{\pi}{2}$	d_{se}	0	θ_3
4	$-\frac{\pi}{2}$	0	0	θ_4
5	$-\frac{\pi}{2}$	d_{ew}	0	θ_5
6	$\frac{\pi}{2}$	0	0	θ_6
7	0	d_{wf}	0	θ_7

Obr. 6. D-H parametre pre rameno na obr.5

Rozmery parametrov d_i : $d_{bs} = 340 \text{ mm}$, $d_{se} = 400 \text{ mm}$, $d_{ew} = 400 \text{ mm}$, $d_{ws} = 126 \text{ mm}$.

Na základe DH parametrov sa zostaví homogénna transformačná matica, ktorá transformuje súradnicovú sústavu $(i-1)$ do súradnicovej sústavy i .

Tvar transformačnej matice:

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Po vypočítaní transformačných matic pre každý článok robota možno ich postupným násobením získať výslednú transformačnú maticu, ktorá určuje celkovú pozíciu a orientáciu koncového efektora vzhľadom na základňu. Pri robote s n kĺbmi táto matica reprezentuje transformáciu od základne až po posledný článok – koncový efektor:

$${}^0T = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 {}^7T_8 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pri simuláciách je potrebné zabezpečiť, aby generované ciele boli dosiahnuteľné v rámci pracovného priestoru robota, inak IK nemusí nájsť žiadne riešenie. [3]

PRACOVNÝ PRIESTOR ROBOTA

Pri navrhovaní generátora inverznej kinematiky (IK) je dôležité definovať pracovný priestor robota, ktorý určuje, v akých oblastiach môže efektor vykonávať operácie. Tento priestor je obmedzený fyzickými vlastnosťami robota, ako aj rozsahom pohybu jeho kĺbov.

A. Definícia pracovného priestoru

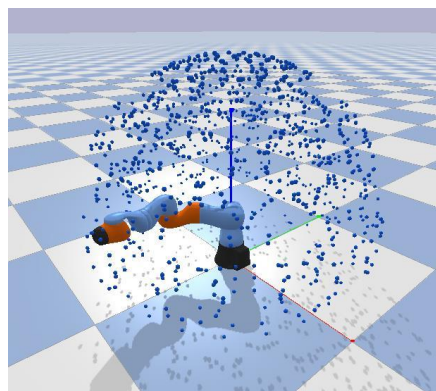
Pracovný priestor pre robot **KUKA LBR iiwa** je definovaný tromi hlavnými parametrami:

- **x-rozsah**: 0.5 m až 1.1 m, ktorý zahŕňa prednú časť pracovného priestoru.
- **y-rozsah**: -0.5 m až 0.5 m, čo predstavuje bočnú oblasť.
- **z-rozsah**: 0.4 m až 1.2 m, ktorý určuje výškovo dostupný rozsah pre efektor robota.

Tieto hranice sú nastavené na základe kinematických obmedzení robota a jeho schopnosti dosiahnuť tieto oblasti v rámci svojich fyzických limitov. [4]

B. Vizualizácia pracovného priestoru v PyBulle

V simulátore **PyBullet** bol pracovný priestor zobrazený pomocou funkcie `draw_workspace()`, ktorá vytvára mriežku bodov v 3D priestore. Tieto body reprezentujú všetky možné pozície efektora robota v rámci definovaných hraníc. Kroková hodnota bola nastavená na **0.1 m**, čím vznikla mriežka bodov, ktorá poskytuje detailný prehľad o dostupnej oblasti pre robotické rameno. [4]



Obr. 7. Pracovný priestor robota v prostredí PyBullet

Všetky bodky boli zobrazené **modrou farbou**, aby sa jasne označil pracovný priestor a umožnila vizualizácia dostupného priestoru, v ktorom môže robot manipulovať s objektmi.

C. Použitie pracovného priestoru pri generovaní inverznej kinematiky

Na základe definovaného pracovného priestoru boli generované náhodné cieľové pozície, ktoré sa nachádzali v rámci týchto hraníc. Pre každú z týchto pozícií bola následne vypočítaná inverzná kinematika, ktorá určila požadované kĺbové uhly robota na dosiahnutie daných pozícií.

Pohyb robota medzi jednotlivými cieľovými pozíciami bol simulovaný, pričom trajektória pohybu efektora bola zobrazená ako čiara v PyBullet. Táto simulácia ilustruje, ako robot využíva pracovný priestor na vykonanie rôznych úloh v rámci svojej operácie. [7]

IV. VÝSLEDOK PRÁCE

V tejto kapitole sú prezentované výstupy zo simulácie generátora inverznej kinematiky pre robotické rameno KUKA LBR iiwa, vytvoreného v prostredí PyBullet. Cieľom bolo navrhnuť, otestovať a vizualizovať funkčnosť algoritmu výpočtu inverznej kinematiky pri náhodne generovaných cieľových pozíciách efektora. [8]

A. Inicializácia simulácie

Na začiatku simulácie bola nastavená gravitácia a načítaný model robota z URDF súboru. Robot bol umiestnený do simulovaného prostredia na pevnú základňu. Následne boli všetky kĺby resetované do nulovej polohy a definovali sa limity pre pohyb kĺbov, ako aj predvolená (pokoiová) konfigurácia.

Okrem toho sa vo vizualizácii vykreslil pracovný priestor robota pomocou pravidelnej mriežky bodov, čo slúžilo na prehľadnejšie zobrazenie dosiahnuteľného priestoru.

B. Generovanie cieľových pozícií

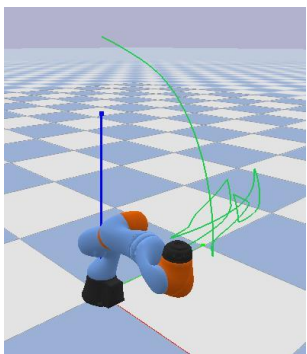
Pomocou funkcie `generate_random_pose()` sa generovalo 15 náhodných cieľových pozícií v pracovnom priestore robota. Tieto pozície boli zvolené tak, aby sa nachádzali v oblasti, ktorú robot dokáže efektívne dosiahnuť (t. j. v rozmedzí X: 0.7 – 1.0 m, Y: -0.3 – 0.3 m, Z: 0.6 – 0.9 m).

C. Výpočet inverznej kinematiky

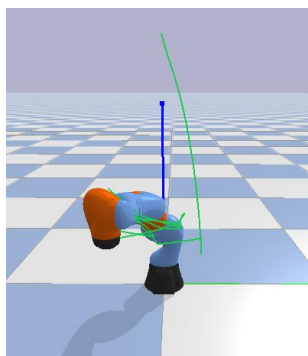
Pre každú cieľovú pozíciu bola pomocou vstavaných funkcií knižnice PyBullet vypočítaná konfigurácia kĺbov, ktorá umožňuje presne umiestniť efektor do požadovaného bodu. Výpočet rešpektoval definované limity, rozsahy a preferovanú pokojovú polohu kĺbov.

D. Riadenie pohybu a Vizualizácia trajektórie

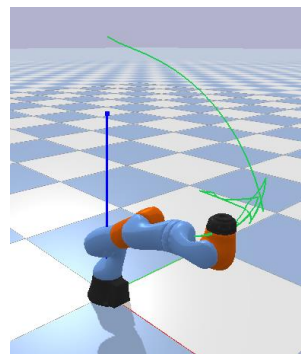
Po výpočte kinematiky nasledovalo riadenie pohybu ramena do vypočítanej konfigurácie. Riadenie bolo realizované v režime `POSITION_CONTROL`, pričom pre každý cieľ sa vykonalo 240 krokov simulácie. Počas pohybu bola priebežne zaznamenávaná pozícia efektora a z týchto bodov bola vykreslená trajektória vo forme spojitej čiary. Nasledujúce obrázky znázorňujú ukážky pohybu robota a trajektórie efektora počas simulácie:



Obr. 8. Vizualizácia dosiahnuteľných bodov efektora v simulovanom priestore.



Obr. 9. Robot sa presúva k jednej z náhodne vygenerovaných pozícií v rámci pracovného priestoru.



Obr. 10. Zobrazenie trajektórie efektora medzi cieľovými bodmi počas simulácie.

PSEUDOKÓD PROGRAMU

ZAČIATOK

```
FUNKCIA generate_random_pose()
// Vygeneruje náhodnú cieľovú pozíciu v dosahu robota
x ← náhodné číslo z intervalu <0.7, 1.0> // smerom dopredu
y ← náhodné číslo z intervalu <-0.3, 0.3> // do strán
z ← náhodné číslo z intervalu <0.6, 0.9> // vertikálne
VRÁŤ [x, y, z]
KONIEC FUNKCIE
```

```
PRÍPOJ sa k simulátoru PyBullet v režime GUI
NASTAV vyhľadávaciu cestu pre načítanie modelov
NASTAV gravitáciu na hodnotu (0, 0, -9.81)
```

```
NAČÍTAJ rovinu a URDF model robota KUKA LBR iiwa
ZÍSKAJ počet kĺbov robota
NASTAV index koncového efektora na 6
```

```
PRE každý kĺb od 0 po počet_kĺbov - 1
NASTAV počiatočnú hodnotu kĺbu na 0
KONIEC PRE
```

```
DEFINUJ obmedzenia kĺbov:
dolné_limit ← [-1.0, -1.0] + [-2.9] pre ostatné kĺby
horné_limit ← [1.0, 1.0] + [2.9] pre ostatné kĺby
rozsahy ← [2.0, 2.0] + [5.8] pre ostatné kĺby
východiskové_polohy ← [0, -0.5, 0, 1.0, 0, 0.5, 0]
```

```
posledná_pozícia ← NULL
```

```
PRE i od 1 do 15
cieľová_pozícia ← generate_random_pose()
cieľová_orientácia ← quaternion z Eulerových uhlov (π, 0, 0)
// efektor smeruje nadol
```

```
// Výpočet kĺbových uhlov pomocou inverznej kinematiky
uhly_kĺbov ← vypočítaj IK s použitím vstupných parametrov
```

```
PRE každý krok od 1 do 240
PRE každý kĺb j
NASTAV kĺb j na cieľový uhol uhly_kĺbov[j] pomocou
riadenia polohy
KONIEC PRE
```


VYKONAJ jeden krok simulácie

ZÍSKAJ aktuálnu pozíciu efektora

AK posledná_pozícia NIE JE NULL

VYKRESLI spojnicu medzi posledná_pozícia a aktuálnou pozíciou

KONIEC AK

posledná_pozícia ← aktuálna pozícia

POČKAJ 1/240 sekundy

KONIEC PRE

POČKAJ 0.5 sekundy

KONIEC PRE

ČAKAJ na vstup používateľa (napr. ENTER)

ODPOJ sa zo simulácie

KONIEC

V. ZÁVER

Celý program je dostupný v repozitári:

https://github.com/Daniela2607-max/Zadanie_VSMS

Tento projekt sa zameriava na vytvorenie generátora inverznej kinematiky pre robotické rameno KUKA LBR iiwa, ktorý umožňuje výpočet inverznej kinematiky pre zadané cieľové pozície. Implementované analytické a numerické metódy zabezpečujú presné riešenie inverznej kinematiky a validáciu pozícií v trojrozmernom priestore. Generátor je schopný generovať náhodné pozície, vykonávať simulácie a analyzovať dosiahnuteľnosť cieľov, čo poskytuje výkonné nástroje na testovanie a optimalizáciu robotických aplikácií. Tento systém je vhodný pre vzdelávacie účely v oblasti robotiky a umožňuje predbežné testovanie kinematických algoritmov pred ich nasadením na skutočné roboty, konkrétne KUKA LBR iiwa.

ZDROJE

- [1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control. London: Springer, 2010.
- [2] J. J. Craig, Introduction to Robotics: Mechanics and Control, 4th ed. Pearson, 2017.
- [3] KUKA Roboter GmbH, KUKA LBR iiwa 7 R800 Specification Sheet, 2020. [Online]. Available: <https://www.kuka.com>
- [4] E. Coumans and Y. Bai, "PyBullet, a Python module for physics simulation for games, robotics and machine learning," [Online]. Available: <https://pybullet.org>
- [5] R. Mukherjee, S. Ramalingam, and A. Sinha, "Inverse kinematics of redundant and non-redundant robotic manipulators using numerical methods," International Journal of Robotics and Automation, vol. 33, no. 1, pp. 45–53, 2018.
- [6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control, 2nd ed. Wiley, 2020.
- [7] S. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," Robotics and Autonomous Systems, vol. 47, no. 2–3, pp. 79–91, 2004.
- [8] PyBullet GitHub Repository, "bulletphysics/bullet3," [Online]. Available: <https://github.com/bulletphysics/bullet3>