

# Vnorené systémy mechatronických sústav: Analýza dosahu robota (2025)

Bc. Sivák Lukáš, Bc. Tomáš Danilo<sup>1</sup>

<sup>1</sup>Katedra mechatroniky, Technická univerzita v Košiciach, Strojnícka fakulta, Košice, Slovensko

## Abstrakt

Tento článok sa detailne zaoberá simulačnou analýzou dosahu robotického manipulátora Emika Franka Panda. Cieľom je generovať a vizualizovať pracovný priestor pomocou kombinácie analytického výpočtu a náhodného vzorkovania. Pre zadávanie cieľových bodov implementujeme algoritmus inverznej kinematiky. Simulácia je realizovaná v PyBullet, pričom pre vizualizáciu je použitá aj knižnica Matplotlib. Práca zároveň opisuje štruktúru robota, rozsahy kĺbov, a postupy výpočtov doprednej aj inverznej kinematiky, s cieľom vytvoriť plnohodnotný nástroj pre testovanie robotických aplikácií.

## Kľúčové slová

Robotický manipulator, Emika Franka Panda, Dopredná kinematika, Inverzná kinematika, Pracovný priestor, Kinematická analýza.

## I. ÚVOD DO PROBLEMATIKY

Robotické ramená s viacerými stupňami voľnosti zohrávajú kľúčovú úlohu v priemyselnej automatizácii. Ich správne nastavenie a programovanie si vyžaduje presné pochopenie kinematických vlastností. Táto práca sa zameriava na simulačné overenie pracovného priestoru robota Emika Franka Panda a jeho schopnosti dosiahnuť požadované body v trojrozmernom priestore. Cieľom je vytvoriť analýzu dosahu robota. Zadanie má obsahovať nasledovné kritériá:

A. Užívateľ zadá BOD a program vypočíta, či je v pracovnom priestore alebo nie je.

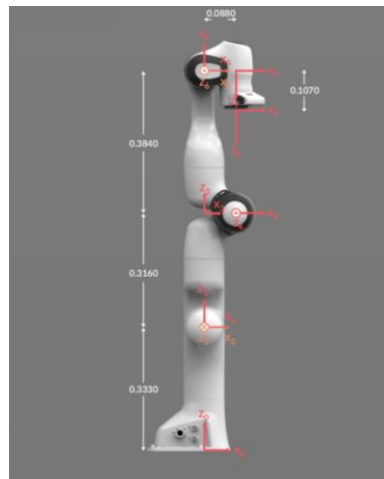
B. ROBOT a BOD v pybullet – ak je v pracovnom priestore zobrazenie pracovného priestoru a vyznačiť bod v ňom, ak nie je, v matplotlib vyznačiť pracovný priestor, BOD a najbližší bod z pracovného priestoru.

## II. MODEL ROBOTICKÉHO MANIPULÁTORA

Emika Franka Panda je 7-osový manipulačný robot s rotačnými (revolučnými) kĺbmi. Načítaním jeho URDF (Unified Robot Description Format) súboru do simulačného prostredia PyBullet bolo možné vytvoriť realistický 3D model robota vrátane definície základne, kĺbov a koncového efektora. Kĺby sú definované nasledovnými rozsahmi:

Name	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Unit
$q_{max}$	2.8973	1.7628	2.8973	-0.0698	2.8973	3.7525	2.8973	rad
$q_{min}$	-2.8973	-1.7628	-2.8973	-3.0718	-2.8973	-0.0175	-2.8973	rad

Obr. 1. Rozsahy kĺbov robota Emika Franka Panda [1]



Obr. 2. Dĺžky ramien robota Emika Franka Panda

Dĺžky jednotlivých ramien a parametre spojení boli upravené na základe dostupnej technickej dokumentácie robota, aby bola zabezpečená zhoda so skutočným fyzickým robotom [1].

## III. TEORETICKÝ ZÁKLAD

### 1. RIEŠENIE KINEMATICKÝCH ÚLOH

Kinematika robotického manipulátora je štúdium pohybov robota bez zohľadnenia síl, ktoré tieto pohyby spôsobujú. Cieľom kinematických úloh je popísať vzťah medzi konfiguráciou robota (uhly kĺbov) a pozíciou koncového efektora v priestore.

#### A. Dopredná kinematika (Forward kinematics)

Dopredná kinematika je proces, ktorý umožňuje určiť pozíciu koncového efektora na základe známych hodnôt pre všetky kĺby robota. Tento proces je pomerne jednoduchý, pretože ide o postupné aplikovanie transformačných matic medzi jednotlivými segmentami robota. Predstavme si manipulátor, ktorý je tvorený sériou segmentov, pričom každý segment je spojený kĺbom. Cieľom doprednej kinematiky je vypočítať pozíciu koncového efektora (alebo iného segmentu) na základe týchto kĺbových uhlov a dĺžok segmentov.

- Transformácia medzi segmentmi robota: Každý segment robota môže byť popísaný transformáciou, ktorá spája tento segment so segmentom pred ním. Tento vzťah môžeme vyjadriť pomocou transformačných matic [2].

- Denavit-Hartenbergove parametre (DH parametre): Na efektívne popísanie vzťahu medzi segmentmi sa používajú DH parametre. Tieto parametre umožňujú definovať transformácie medzi po sebe nasledujúcimi segmentami robota:

$\theta_i$ -uhol otáčania okolo osi z,  
 $d_i$ -vzdialnosť pozdĺž osi z,  
 $a_i$ -dĺžka segmentu pozdĺž osi x  
 $\alpha_i$ -uhol medzi osami z a  $z_{i-1}$

Pomocou DH parametrov sa zostaví transformačná matica pre každý segment, ktorá popisuje rotáciu a transláciu medzi dvoma po sebe idúcimi segmentmi. Transformácia medzi dvoma segmentmi  $i-1$  a  $i$  je daná maticou [3]:

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

Po získaní matic pre každý segment robota je možné pomocou postupného násobenia matic získať celkovú transformačnú maticu, ktorá určuje pozíciu a orientáciu koncového efektora. Pre robot s  $n$  kĺbmi je celková transformačná matica od základne po koncový efektor [2]:

$$T = T_1^0 \cdot T_2^1 \cdot \dots \cdot T_n^{n-1}$$

### B. Inverzná kinematika (Inverse kinematics)

Inverzná kinematika je proces, ktorý sa používa na určenie hodnôt uhlov kĺbov, ktoré umožňujú dosiahnuť požadovanú pozíciu a orientáciu koncového efektora. Tento problém je oveľa zložitejší ako dopredná kinematika, pretože zadaná pozícia koncového efektora nevedie jednoznačne k jediným hodnotám uhlov kĺbov.

Na rozdiel od doprednej kinematiky, kde sa pozícia koncového efektora vypočítava z hodnôt uhlov kĺbov, pri inverznej kinematike sa snažíme nájsť tieto uhly, keď už máme pozíciu koncového efektora. Pre jednoduchšie manipulátory je možné tento problém riešiť analyticky, avšak pre komplexnejšie roboty s viacerými kĺbmi (napr. 6-DOF roboty) sa často používajú numerické metódy [2].

- Analytický prístup (uzavretá forma): Vhodný pre manipulátory so špecifickou geometriou. Umožňuje presné a rýchle výpočty, avšak nie je univerzálne aplikovateľný.

- Numerický prístup (iteračné metódy): Založený na minimalizácii chyby medzi požadovanou a aktuálnou pozíciou a orientáciou. Využíva sa napr. Jacobian a jeho pseudoinverzia, metódy gradientného zostupu či CCD (Cyclic Coordinate Descent).

V našej práci sa na vizualizáciu pracovného priestoru využíva generovanie množiny konfigurácií robota, pričom sa doprednou kinematikou vypočítavajú jednotlivé pozície efektora. Takto získané body sú následne vizualizované v 3D priestore ako

zhhluk bodov alebo obálka (škrupina), ktorá reprezentuje dosiahnuteľný priestor robota.

Riešenie kinematických úloh sa môže stať veľmi zložitým pri robotoch s viacerými kĺbmi alebo pri robotoch, ktoré majú nelineárne správanie. Výzvy pri riešení kinematických úloh zahŕňajú:

- Počet možných riešení: Inverzná kinematika môže mať viac ako jedno riešenie alebo žiadne riešenie (napr. ak je pozícia mimo dosiahnuteľného priestoru).
- Singularity: Kinematické singularity sa vyskytujú, keď robot stráca jeden alebo viac stupňov voľnosti a nemôže sa pohnúť v určitých smeroch.
- Numerická stabilita: Pri použití numerických metód sa môže vyskytnúť problém s konvergenciou alebo stabilitou metódy, najmä pri robotoch s komplexnými geometriami.

## 2. POČÍTAČOVÁ ANALÝZA A VIZUALIZÁCIA

### A. Generovanie pracovného priestoru

Generovanie pracovného priestoru robota je kľúčovým aspektom v procese modelovania a analýzy, pretože umožňuje získať informácie o tom, akú oblasť môže robot dosiahnuť, a ako efektívne vykonávať operácie v rôznych polohách. Pre tento účel boli použité dve hlavné metódy: analytická metóda a náhodné vzorkovanie.

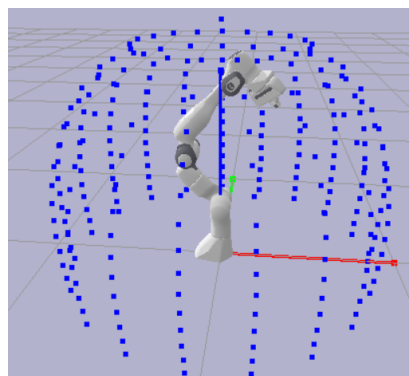
Pozn.: Priestor pod základňou robota neberieme do úvahy.

#### • Analytická metóda

Pri tejto metóde bol pracovný priestor modelovaný ako elipsoid, ktorého rozmer závisí od výšky  $z$  v rozsahu od 0 do 1.19 metra. Tento elipsoid je obmedzený vzdialenosťou, ktorú robot môže dosiahnuť s ohľadom na fyzické parametre manipulátora, ako sú dĺžky ramien. Vypočítaný polomer  $r$  v závislosti od výšky  $z$  je definovaný rovnicou:

$$r = R_{MAX} \cdot \sqrt{1 - \left(\frac{z}{z_{max}}\right)^3} \quad (1)$$

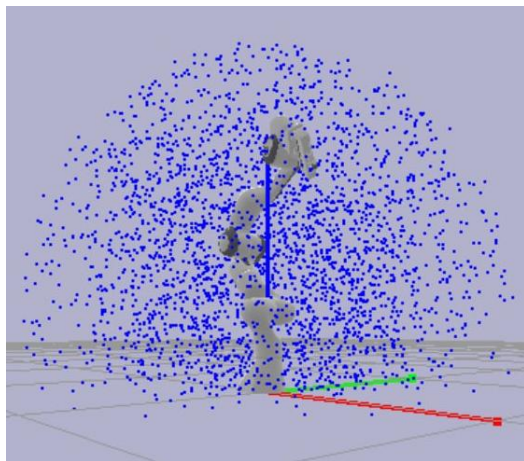
Kde  $R_{MAX}$  predstavuje maximálny dosah manipulátora v horizontálnom smere (0.855 m) a  $z_{max}$  je maximálna povolená výška (1.19 m). Výsledkom je hladká hranica pre maximálne dosiahnuteľné body efektora v rámci 3D priestoru.



Obr. 3. Pracovný priestor analytickou metódou

- **Náhodné vzorkovanie (sampling)**

Náhodné vzorkovanie je technika, ktorá sa používa na generovanie rôznych konfigurácií robotického manipulátora v rámci jeho povolených rozsahov pohybu. Tento proces je veľmi užitočný pri výpočtoch pracovného priestoru robota, keď chceme získať reprezentatívne body v priestore, ktoré sú dosiahnuteľné robotom, bez toho, aby sme museli ručne vypočítvať každý možný bod. Tento proces sa vykonáva opakovane, pričom pre každú generovanú konfiguráciu sa získa pozícia koncového efektora (end-effector) robota. Výstupom tejto analýzy je mračno bodov reprezentujúce všetky možné dosiahnuteľné pozície koncového efektora pri daných obmedzeniach [4].



**Obr. 4.** Pracovný priestor náhodným vzorkovaním

Postup vzorkovania môžeme zadeliť do nasledujúcich bodov:

1. Definovanie rozsahu pre každý kĺb:

Každý kĺb robota má určitý rozsah pohybu, ktorý je definovaný minimálnymi a maximálnymi hodnotami. Tieto hodnoty sú dané fyzikálnymi obmedzeniami robota a sú základom pre generovanie náhodných hodnôt.

2. Generovanie náhodných hodnôt pre kĺby:

Pre každý kĺb sa v rámci jeho rozsahu generuje náhodná hodnota. Na tento účel sa využíva náhodná funkcia, ktorá vytvára hodnoty z rovnomerného rozdelenia medzi minimálnym a maximálnym rozsahom kĺbu. Tieto hodnoty predstavujú aktuálnu konfiguráciu robotických kĺbov v danom okamihu.

3. Výpočet pozície koncového efektora:

Po vygenerovaní náhodnej konfigurácie kĺbov sa vykoná výpočet pozície koncového efektora pomocou doprednej kinematiky. Tento krok zahŕňa výpočet transformácie, ktorá spája jednotlivé segmenty robota a určuje polohu koncového efektora v trojdimenzionálnom priestore. Na tento účel použijeme v PyBullet funkciu `getLinkState`, ktorá vracia aktuálnu pozíciu a orientáciu konkrétného segmentu alebo koncového efektora robota.

4. Overenie dosiahnuteľnosti bodu:

Po získaní pozície koncového efektora je dôležité overiť, či tento bod leží v pracovnom priestore robota. Tento krok môže byť implementovaný ako porovnanie pozície s limitmi pracovného priestoru.

5. Uloženie validných konfigurácií:

Ak je pozícia koncového efektora v rámci pracovného priestoru robota, konfigurácia sa považuje za platnú a ukladá sa do zoznamu platných bodov. Tento zoznam môže byť použitý neskôr na analýzu pohybu robota alebo na optimalizáciu trajektórií.

6. Opakovanie procesu:

Tento proces sa opakuje mnohokrát, čím sa generuje množstvo rôznych konfigurácií a získa sa tak reprezentatívna vzorka z pracovného priestoru robota. Počet opakovaní závisí od požiadaviek na presnosť modelovania pracovného priestoru a rýchlosť výpočtu.

## **B. Zadanie a validácia cieľového bodu**

Po vygenerovaní pracovného priestoru je dôležité zabezpečiť, aby robot dokázal efektívne dosiahnuť predom určené cieľové body v rámci tohoto priestoru. Tento proces sa rozdeľuje do niekoľkých kľúčových krokov:

1. Validácia cieľového bodu

Predtým, než robot začne vykonávať pohyb, je potrebné overiť, či je cieľový bod v rámci jeho dosiahnuteľného pracovného priestoru. Validácia zahŕňa kontrolu:

Vzdialenosti od základne: Cieľový bod by mal byť v rozmedzí dosiahnuteľnosti robota a zároveň by do neho nemal narážať.

Rozsah kĺbov: Zabezpečuje sa, aby požadovaná pozícia nevyžadovala polohy, ktoré sú mimo fyzických limitov kĺbov robota.

Dosiahnuteľnosť: Kontroluje sa, či je bod v skutočne dosiahnuteľnej oblasti vzhľadom na obmedzenia robota.

2. Výpočet inverznej kinematiky

Ak je cieľový bod validovaný, robot vykoná výpočet inverznej kinematiky na získanie potrebných uhlov kĺbov, ktoré sú potrebné na dosiahnutie cieľa. Tento výpočet je realizovaný pomocou PyBullet funkcie:

```
joint_angles = p.calculateInverseKinematics(robot_id, 11, target_position)
```

číslo cieľového linku (tu je to 11 pre koncový efektor) a cieľovú pozíciu. Vrátí uhly kĺbov, ktoré umožňujú dosiahnuť požiadavku.

3. Vizualizácia najbližšieho bodu

Po výpočte inverznej kinematiky je ďalším krokom vizualizácia cieľového bodu v rámci pracovného priestoru. Ak je požiadavka, aby robot dosiahol bod, ktorý je mimo jeho dosahu, určí sa najbližší bod v pracovnom priestore, ktorý je dosiahnuteľný. Na túto úlohu sa používa výpočet vzdialenosti medzi cieľovým bodom a všetkými bodmi v pracovnom priestore. Na tento účel bola použitá funkcia z knižnice SciPy:

```
dist = distance.cdist([target_position], workspace_points)
nearest_point_idx = np.argmin(dist)
```

Táto funkcia vypočíta euklidovskú vzdialenosť medzi cieľovým bodom a všetkými bodmi v pracovnom priestore a vyberie bod, ktorý je najbližšie k cieľu.

#### IV. VÝSLEDOK PRÁCE

##### GENEROVANIE A VIZUALIZÁCIA BODOV PRACOVNÉHO PRIESTORU

Vizualizácia bola realizovaná v prostredí Python a knižnice PyBullet pre 3D zobrazenie s možnosťou animácie (Obr.4). Pracovný priestor bol získaný pomocou iterácie všetkých stupňov voľnosti v definovaných rozsahoch a výpočtom doprednej kinematiky. Po vizualizácii užívateľ zadá parameter bodu koncového efektora vzhľadom na svetovú suradnicovú sústavu uloženú v podstave robota. Údaje zadávame v milimetroch postupne pre os x,y,z.

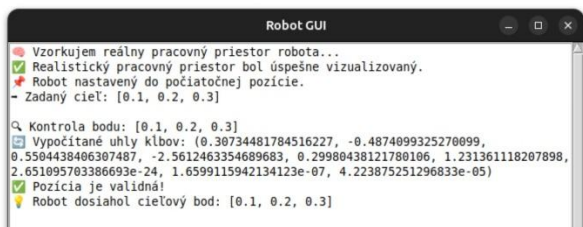
Vytvorený program generoval všetky možné kombinácie uhlov jednotlivých kĺbov v definovanom rozsahu (napr.  $[-90^{\circ}, +90^{\circ}]$  so zvoleným uhlovým krokom, napr.  $5^{\circ}$ ). Pre každú kombináciu sa pomocou homogénnych transformačných matic vypočítala pozícia koncového bodu v kartézskom priestore. Prostredie GUI očakáva vstup. Stlačením tlačidla Zadať cieľový bod sa otvorí dialogové okno kde zadáme parameter.



Obr. 5. Výpis z GUI, Vstup vo vyskakovacom okne

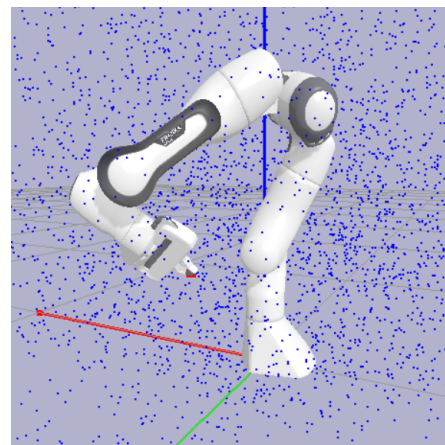
##### SIMULÁCIA POHYBU ROBOTA A VIZUALIZÁCIA ZADANÉHO BODU – ANALÝZA DOSIAHNUTEĽNOSTI

Po zadaní vstupnej hodnoty 100 200 300, sa kontroluje či sme v dosahu a následne prebieha výpočet uhlov jednotlivých kĺbov.



Obr. 6. Výpis z GUI, Kontrola zadaného bodu

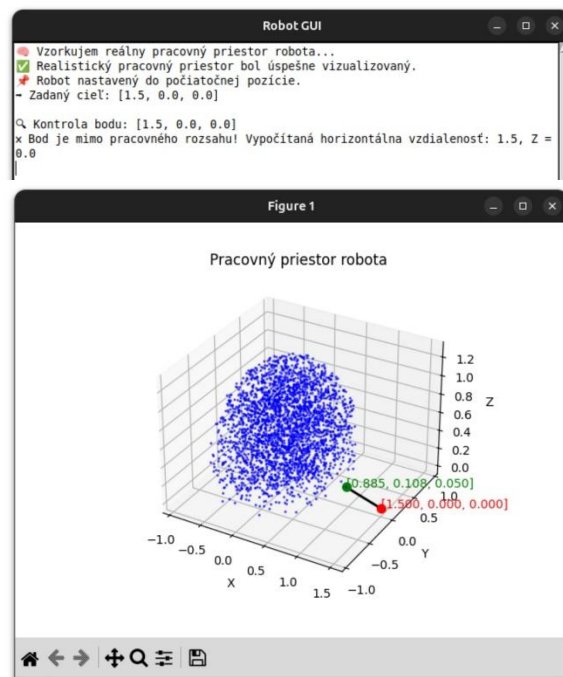
Validovaný bod sa zobrazí v pracovnom prostredí pybullet červenou farbou.



Obr. 7. ROBOT a BOD v pybullet

##### SIMULÁCIA POHYBU ROBOTA A VIZUALIZÁCIA ZADANÉHO BODU MIMO PRACOVNÝ ROZSAH

Pomocou knižnice Matplotlib zobrazíme hodnoty mimo rozsahu a zistíme najbližší bod, ktorý je v pracovnom priestore. Želaný bod a validovaný bod je prepojený spojnicou pričom v hranatých zátvorkách sa nachádza hodnota bodu x,y,z v metroch.



Obr. 8. Výpis z GUI mimo rozsah a vizualizácia v matplotlib s najbližším vhodným bodom

V rámci zadávania bodov môže nastať situácia že zadáme bod zdanlivo v pracovnom priestore robota avšak limity kĺbov nám neumožňujú dostať sa do tohto bodu. Prikladom je bod -800, 0, 658. Nakoľko robot sa nedokáže otáčať v kĺboch o  $360^{\circ}$  v matplotlib zistíme najbližší bod, ktorý bude robot schopný vykonať.





Obr. 9. Výpis z GUI, mimo rozsah kĺbu

Pomocou tejto metódy je možné nielen vizuálne overiť rozsah pohybu robota, ale tiež identifikovať problematické oblasti (napr. singularity, oblasti mimo dosahu), čo je dôležité pri návrhu algoritmov pre inverznú kinematiku.

### PSEUDOKÓD PROGRAMU

#### Input:

- Rozsahy uhlov pre kĺby:  $\theta_1 \in [\theta_{1\_min}, \theta_{1\_max}]$ ,  $\theta_2 \in [\theta_{2\_min}, \theta_{2\_max}]$ ,  $\theta_3 \in [\theta_{3\_min}, \theta_{3\_max}]$
- Dĺžky článkov:  $L_1, L_2, L_3$
- Krok vzorkovania  $\Delta\theta$
- Zadaný cieľový bod:  $P_{target} = [X, Y, Z]$

#### Output:

- Vizualizácia dosiahnuteľných bodov
- Vyhodnotenie dosiahnuteľnosti bodu  $P_{target}$

#### 1. Inicializácia

Vytvor prázdny zoznam `reachable_points`  
Nastav krok vzorkovania  $\Delta\theta$

#### 2. Numerické vzorkovanie priestorového dosahu robota

**For**  $\theta_1$  from  $\theta_{1\_min}$  to  $\theta_{1\_max}$  step  $\Delta\theta$

**For**  $\theta_2$  from  $\theta_{2\_min}$  to  $\theta_{2\_max}$  step  $\Delta\theta$

**For**  $\theta_3$  from  $\theta_{3\_min}$  to  $\theta_{3\_max}$  step  $\Delta\theta$

- Vypočítaj doprednú kinematiku pomocou SE(3)

- Získaj pozíciu koncového bodu  $P = [x, y, z]$

- Pridaj  $P$  do zoznamu `reachable_points`

**End For**

**End For**

**End For**

#### 3. Vizualizácia pracovného priestoru- Zobraz všetky body zoznamu `reachable_points` v 3D priestore

#### 4. Zadanie cieľového bodu používateľom

- Načítaj  $P_{target} = [X, Y, Z]$

#### 5. Vyhodnotenie dosiahnuteľnosti

**If**  $P_{target}$  sa nachádza v `reachable_points`

- Zobraz bod  $P_{target}$  v priestore červenou farbou

**Else**

- Nájdi najbližší bod  $P_{nearest}$  v `reachable_points` podľa Euklidovskej vzdialenosti

- Zobraz  $P_{target}$  ako červený bod,  $P_{nearest}$  ako zelený bod.

- Vypíš: „Bod je mimo pracovného priestoru.“

**End If**

## 6. Opakovanie alebo ukončenie

- Umožni opakovanie pre nové zadanie alebo ukonči program krížikom.

## V. ZÁVER

Celý program je k dispozícii v repozitári: <https://github.com/LuSivo/Zadanie2025>

Výsledkom práce je robustná a interaktívna simulačná platforma pre model Emika Franka Panda, ktorá umožňuje vizualizovať pracovný priestor, riešiť úlohy doprednej a inverznej kinematiky a validovať zadané cieľové pozície. Použitím analytických a numerických metód je možné presne charakterizovať schopnosti robota v 3D priestore. Navrhnutý systém môže byť použitý na didaktické účely v oblasti robotiky, ako aj na praktické testovanie robotických aplikácií pred ich fyzickým nasadením.

## POĎAKOVANIE

V prvom rade ďakujem môjmu cvičiacemu za poskytnutie hodnotných programov, ktoré boli neoceniteľnou pomôckou pri riešení zadaných úloh. Jeho vedenie a podpora počas celého procesu, ako aj schopnosť navrhnúť takto zaujímavé a zároveň komplikované zadanie, boli kľúčové pre úspešné zvládnutie tejto výzvy.

Ďalej by som chcel poďakovať za pomocné programy a nástroje, ktoré mi výrazne uľahčili samotný proces programovania a riešenia problémov.

Všetkým, ktorí prispeli svojou pomocou a podporou, patrí moje veľké poďakovanie.

## ZDROJE

#### Examples:

- [1] FRANKA. (n.d.). Robot and interface specifications. [Online]. Available: [https://frankaemika.github.io/docs/control\\_parameters.html](https://frankaemika.github.io/docs/control_parameters.html)
- [2] SICILIANO, Bruno; SCIAVICCO, Lorenzo; VILLANI, Luigi; ORIOLO, Giuseppe. Robotics: Modelling, Planning and Control. London: Springer, 2009. ISBN 978-1846286414.
- [3] DENAVIT, Jacques; HARTENBERG, Richard S. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. ASME Journal of Applied Mechanics, 1955, 77, 215-221.
- [4] CAO, Yi; ZANG, Haihe; WU, Lan; LU, Tao. An Engineering-oriented Method for the Three Dimensional Workspace Generation of Robot Manipulator. Journal of Information & Computational Science, 2011. ISSN 1548-7741.
- [5] YIM, Aldrin; CHUNG, Claire; YU, Allen. Matplotlib for Python Developers: Effective Techniques for Data Visualization with Python. Packt Publishing Ltd, 2018. ISBN 978-1788625176.