

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

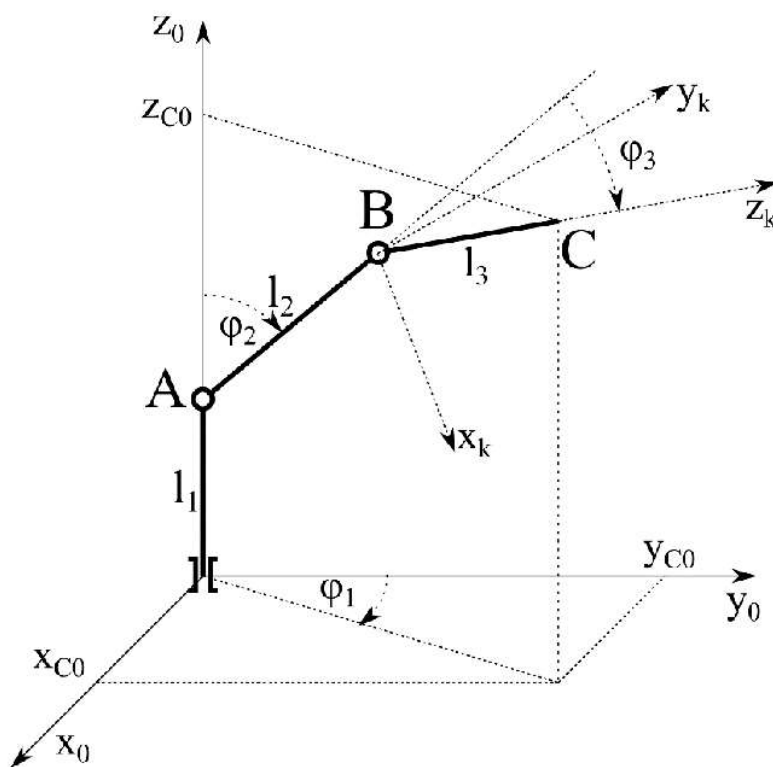
VIZUALIZÁCIA PRIAMEJ KINEMATICKEJ ÚLOHY

Dokumentácia k zadaniu č. 1 z predmetu Robotika

Študijný program:	Robotika a kybernetika
Študijný odbor:	kybernetika
Pracovisko:	Ústav robotiky a kybernetiky
Mesiac a rok odovzdania:	04/2021
Vypracoval:	Filip Gerhát

1. ZADANIE

Navrhните a realizujte vizualizáciu robotického ramena uvedeného na Obr. 1. Na tomto type zadania by ste si mali precvičiť implementáciu homogénnych transformácií a zafixovať tak preberané učivo Priamej kinematickej úlohy trojramenného manipulátora typu RRR. Všetko potrebné bolo odvodené na cvičeniach.



Obr. 1: Robotické rameno typu RRR

Parametre manipulátora:

$$l_1 = 203 \text{ mm}, l_2 = 178 \text{ mm}, l_3 = 178 \text{ mm}, \varphi_1 = \langle -90^\circ, 90^\circ \rangle, \varphi_2 = \langle -55^\circ, 125^\circ \rangle, \varphi_3 = \langle 0^\circ, 150^\circ \rangle$$

V rámci riešenia zadania sa zamerajte na nasledovné úlohy:

1. Vykreslenie manipulátora v 3D priestore v zvolenej konfigurácii.
2. Vykreslenie jednotlivých pomocných súradných systémov 0-k (x-červenou farbou, y-zelenou, z-modrou).
3. Vykreslenie pracovného priestoru pre ľubovoľné hodnoty kĺbových premenných v básovej rovine manipulátora (t. j. rovina tvorená bodmi A,B,C), v rovine XY a môžete vykresliť aj 3D bublinu, avšak obmedzte počet vykresľovaných bodov).

2. VYPRACOVANIE

2.1. Priama kinematická úloha

Robotické rameno na Obr. 1 sa skladá z troch kĺbov a troch rovných článkov. Prvý kĺb tvorí otáčavú základňu robota a nachádza sa v strede „svetovej“ súradnicovej sústavy (súradnicový systém 0). Druhý kĺb sa nachádza v bode A a tretí kĺb v bode B. Koncovým bodom ramena je bod C.

Pre vizualizáciu robotického ramena v trojrozmernom priestore a riešenie ďalších úloh tohto zadania je potrebné vedieť určiť polohy všetkých kĺbov a koncového bodu ramena v súradnicovom systéme 0. Ich poloha sa mení v závislosti od natočenia jednotlivých kĺbov a dĺžok jednotlivých článkov ramena. Ide o tzv. priamu kinematickú úlohu. Na prepočet polohy bodu z ľubovoľného súradnicového systému do svetového systému sa používajú homogénne transformačné matice. Označujú sa ako $T_{i,j}$, kde j je pôvodný súradnicový systém a i je cieľový súradnicový systém.

Začínajúc v strede súradnicovej sústavy 0 je vytvorený pomocný súradnicový systém pre každé posunutie po článku ramena alebo rotáciu v kĺbe ramena. Pre každý z nich je potom odvodená homogénna transformačná matica z daného systému do predchádzajúceho systému. Pomocou týchto matic je možné v ľubovoľnej konfigurácii robota transformovať bod vyjadrený polohovým vektorom z ktoréhokoľvek pomocného súradnicového systému do svetového.

2.1.1. Transformačné matice robotického ramena

Poznámka: v transformačných maticiach sa pre skrátenie zápisu označuje funkcia sínus písmenom „s“ a funkcia kosínus písmenom „c“.

- Rotácia v prvom kĺbe o uhol φ_1

$$T_{0,1} = R_z(90^\circ - \varphi_1) = \begin{pmatrix} s\varphi_1 & -c\varphi_1 & 0 & 0 \\ c\varphi_1 & s\varphi_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Translácia po prvom článku o dĺžku l_1 do bodu A

$$T_{1,2} = T_z(l_1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Rotácia v druhom kĺbe o uhol φ_2

$$T_{2,3} = R_y(\varphi_2) = \begin{pmatrix} c\varphi_2 & 0 & s\varphi_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s\varphi_2 & 0 & c\varphi_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Translácia po druhom článku o dĺžku l_2 do bodu B

$$T_{3,4} = T_z(l_2) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Rotácia v treťom kĺbe o uhol φ_3

$$T_{4,5} = R_y(\varphi_3) = \begin{pmatrix} c\varphi_3 & 0 & s\varphi_3 & 0 \\ 0 & 1 & 0 & 0 \\ -s\varphi_3 & 0 & c\varphi_3 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Translácia po treťom článku o dĺžku l_3 do koncového bodu C

$$T_{5,6} = T_z(l_3) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Súradnicový systém 6 je súradnicovým systémom koncového bodu robota.

Transformácia polohového vektora $\overline{p_k}$ zo súradnicového systému k do súradnicového systému 0 je daná vzťahom

$$\overline{p_0} = T_{0,k} * \overline{p_k} = \prod_{i=1}^k (T_{(i-1),i}) * \overline{p_k}$$

2.2. Vykreslenie manipulátora v 3D priestore

Manipulátor je možné vykresliť ako postupnosť úsečiek (graf), ktoré spájajú prvý kĺb (bod 0), druhý kĺb (bod A), tretí kĺb (bod B) a koncový bod C. Kvôli tomu je však potrebné zistiť súradnice týchto 4 bodov v súradnicovom systéme 0 v danej konfigurácii robota.

Prvý kĺb sa nezávisle od konfigurácie vždy nachádza v strede svetového súradnicového systému, jeho polohový vektor je

$$\overline{p_{00}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Pre zvyšné body A, B, C platí, že:

- bod A sa nachádza v strede súradnicového systému 2

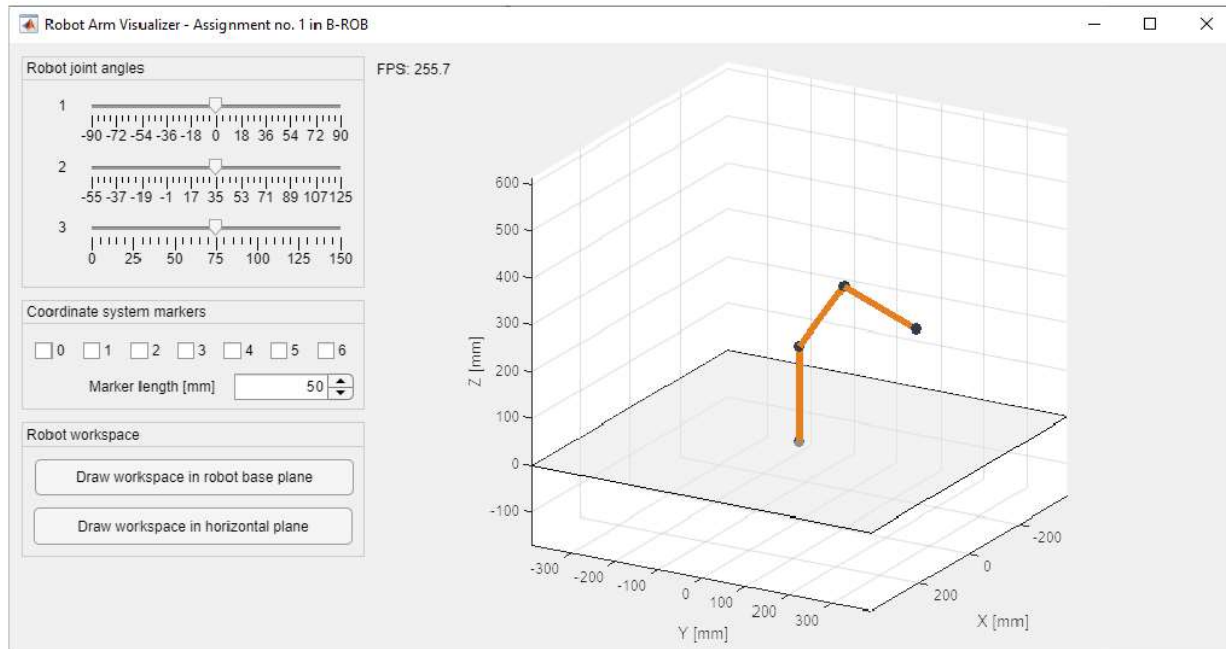
$$\overline{p_{A0}} = T_{0,2} * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- bod B sa nachádza v strede súradnicového systému 4

$$\overline{p_{B0}} = T_{0,4} * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- bod C sa nachádza v strede súradnicového systému 6

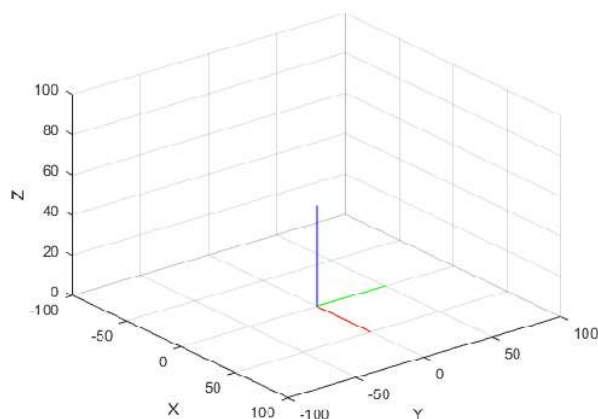
$$\overline{p_{C0}} = T_{0,6} * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$



Obr. 2: Ukážka vykreslenia robota v 3D priestore

2.3. Vykreslenie značky súradnicového systému

Jednou z úloh tohto zadania je označiť vo vizualizácii robotického ramena jednotlivé pomocné súradnicové systémy. Ukážka značky súradnicového systému je na Obr. 3.



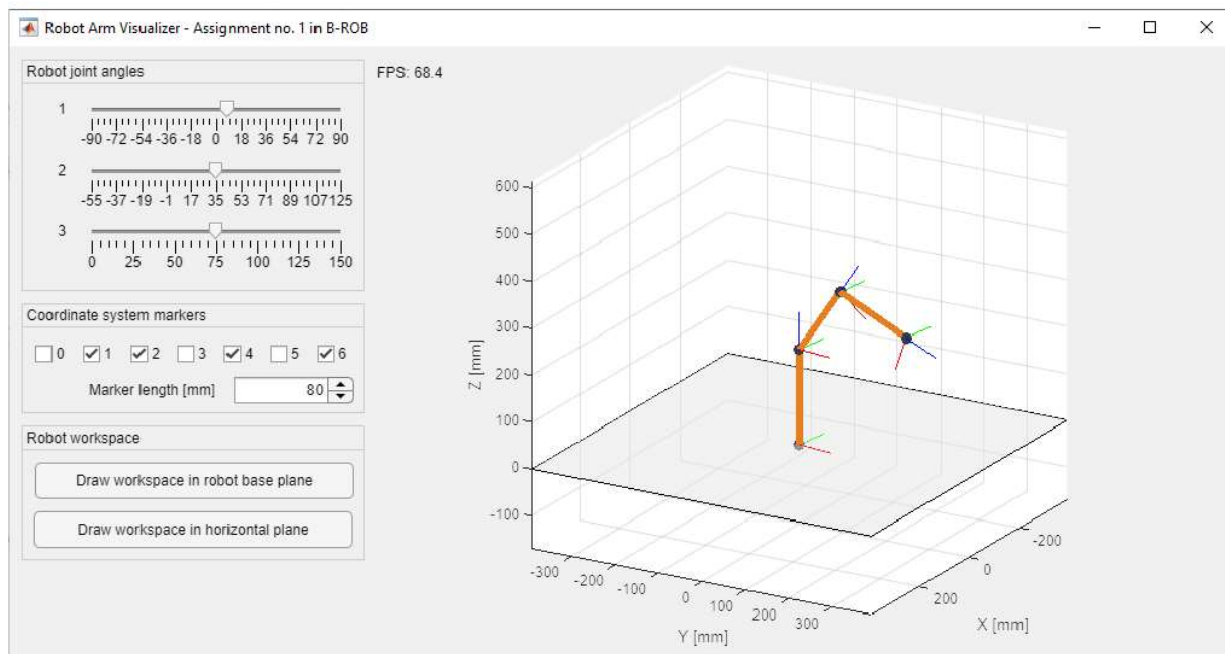
Obr. 3: Značka svetového súradnicového systému

Značka sa skladá z troch čiar, pričom všetky vychádzajú zo stredu súradnicového systému: červená čiara v smere X, zelená čiara v smere Y a modrá čiara v smere Z. Na vykreslenie tejto značky stačí poznať polohu stredu súradnicového systému a koncové body jednotlivých čiar, celkovo teda 4 body v priestore. Polohu týchto bodov v danom súradnicovom systéme môžeme vyjadriť polohovými vektormi a tie zoskupiť do matice

$$C = \begin{pmatrix} 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 0 & d \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

kde d je dĺžka čiar. Výhodou takejto reprezentácie je to, že body v matici C môžeme transformovať z ľubovoľného súradnicového systému k do svetového a tým vykresliť túto značku v strede súradnicového systému k v správnej orientácii. Maticu $C_{0,k}$ popisujúcu polohu značky súradnicového systému k v súradnicovom systéme 0 dostaneme vzťahom

$$C_{0,k} = T_{0,k} * C$$



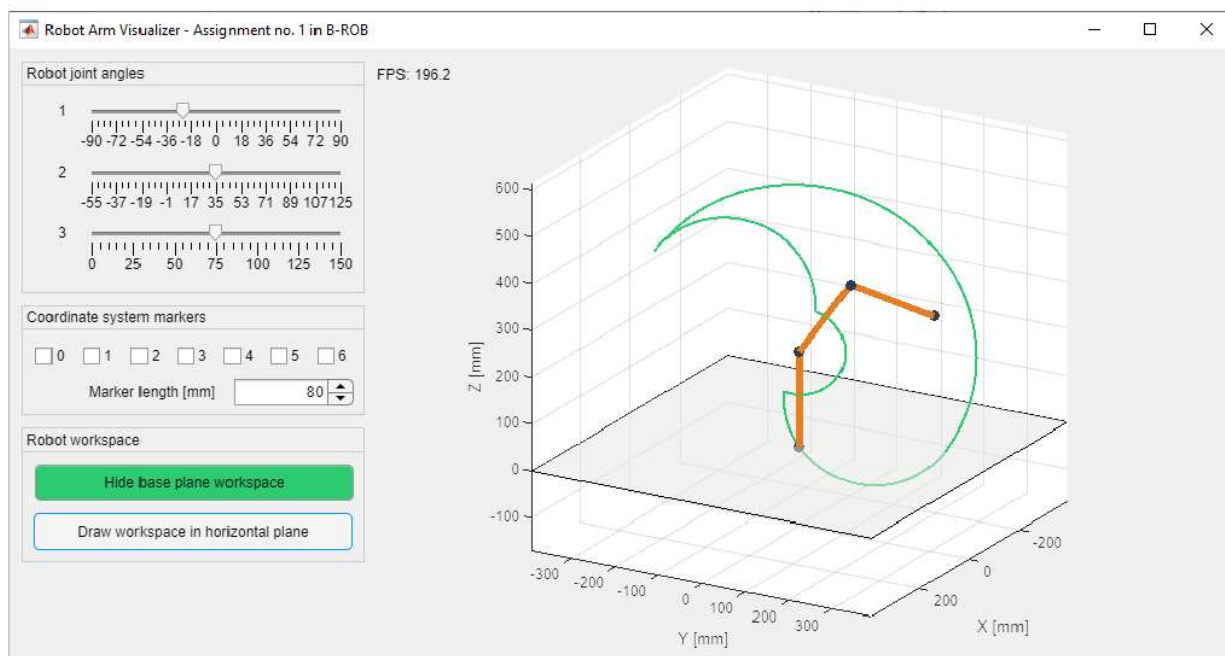
Obr. 4: Ukážka vykreslenia robota s označenými pomocnými súradnicovými systémami vo všetkých kĺboch a v koncovom bode

2.4. Zistenie a vykreslenie pracovného priestoru robota

2.4.1. Pracovný priestor v básovej rovine robota

Bázová rovina robota je tvorená kĺbmi robota, resp. bodmi A, B, C. Pracovný priestor robota v tejto rovine je obmedzený dĺžkou článkov robota a rozsahom kĺbov 2 a 3. Kĺb 1 mení iba rotáciu básovej roviny okolo osi Z.

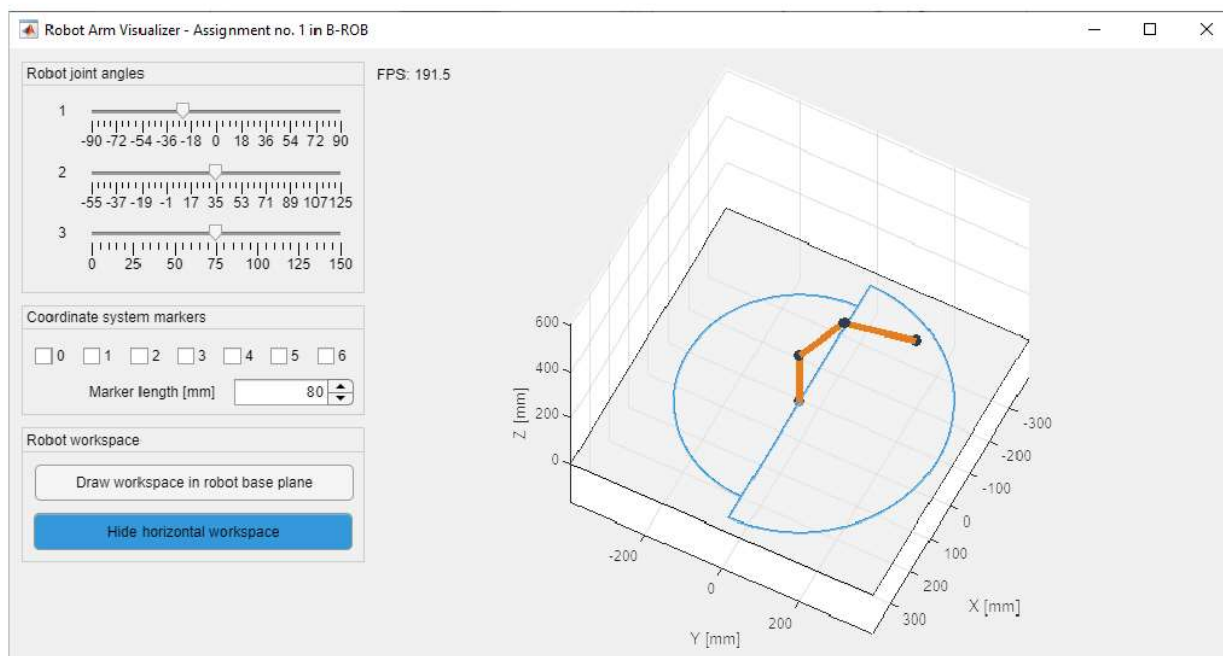
Na opísanie pracovného priestoru robota v básovej rovine bol použitý jednoduchý algoritmus, ktorý striedavo mení natočenie kĺbov 2 a 3 od minimálnej po maximálnu hodnotu a späť, čím koncový bod robota opíše tento pracovný priestor.



Obr. 5: Vykreslenie pracovného priestoru robota v bázeovej rovine

2.4.2. Pracovný priestor v rovine XY

Pracovný priestor v rovine XY popisuje maximálny dosah robota bez ohľadu na výšku Z, v akej je daný bod dosiahnuteľný. Na opísanie tohto priestoru bol použitý zložitejší algoritmus, ktorý najskôr nájde kombinácie natočení kĺbov 2 a 3 pre najdlhší dosah robota vpred a vzad. Následne je v oboch týchto konfiguráciách krokovaná poloha kĺbu 1 od minimálnej po maximálnu hodnotu, čím koncový bod robota opíše kružnicový výsek ohraničujúci pracovný priestor robota. Na záver sú súradnice Z všetkých nameraných bodov prepísané na 0, čím dôjde k projekcii alebo „stlačeniu“ všetkých bodov do roviny XY.



Obr. 6: Vykreslenie pracovného priestoru robota v rovine XY

2.5. Implementácia riešenia a návrh vizualizačnej aplikácie

Na návrh aplikácie bolo využité prostredie MATLAB [1], pričom na tvorbu konečného užívateľského rozhrania bol použitý nástroj App Designer [2].

Pre lepšiu organizáciu kódu boli najskôr navrhnuté samostatne funkčné triedy a metódy, do ktorých bola rozdelená a postupne implementovaná požadovaná funkcionálna. Pri návrhu boli využité niektoré princípy objektovo orientovaného programovania [3].

2.5.1. Trieda (knižnica) `StdMatrix`

Ako prvá bola navrhnutá jednoduchá trieda (class, [4]) `StdMatrix`, ktorá je knižnicou šiestich štandardných translačných a rotačných matic, z ktorých niektoré boli využité v ďalších programoch. Táto trieda je statická [5], čo znamená, že z triedy `StdMatrix` sa nevytvára objekt, ale jej metódy je možné volať priamo volaním `StdMatrix.<názov metódy>(parametre)`.

2.5.2. Trieda `RobotRRR`

Podstatnou časťou riešenia je trieda `RobotRRR`, ktorej úlohou je uchovávať vlastnosti manipulátora - dĺžky článkov, natočenia kĺbov a limity natočenia jednotlivých kĺbov. Objektu typu `RobotRRR` je možné kedykoľvek zmeniť uhly natočenia kĺbov, ostatné vlastnosti sú od vytvorenia objektu nemenné. Ďalej je táto trieda schopná vypočítať v aktuálnej konfigurácii transformačné matice manipulátora popísané v kapitole 2.1.1 a polohový vektor koncového bodu vo svetovom súradnicovom systéme. S ich využitím poskytuje trieda funkcie na zistenie polohy koncového bodu, zistenie polohy všetkých kĺbov a na transformáciu ľubovoľných bodov z ktoréhokoľvek pomocného súradnicového systému robota do svetového systému.

Trieda `RobotRRR` je optimalizovaná tak, aby nepočítala transformačné matice, pokiaľ to nie je nutné. K výpočtu matic dôjde až vtedy, keď je potrebné ich použiť v nejakom výpočte a zároveň iba

vtedy, ak ešte neboli pre aktuálnu konfiguráciu vypočítané. Pri výpočte transformačných matic jednotlivých pomocných súradnicových systémov sa postupuje od systému 0, pričom sa vždy ďalšia matica vynásobí predchádzajúcou. Tým je docielené, že trieda uchováva iba matice $T_{0,1}$, $T_{0,2}$ až $T_{0,6}$ pre transformáciu priamo do svetového súradnicového systému, nakoľko transformácie medzi dvoma pomocnými systémami nie sú v tomto zadaní potrebné.

2.5.3. Trieda (knížnica) `CoordMarker`

Táto statická trieda poskytuje dve funkcie slúžiace na prácu so značkami súradnicových systémov. `CoordMarker.Create` vytvára maticu C popísanú v kapitole 2.3 a funkcia `CoordMarker.Draw` vykreslí značku definovanú vstupnou maticou do zadaného grafu.

2.5.4. Ďalšie pomocné funkcie

Medzi ďalšie samostatné funkcie patrí:

- `DrawRobot` – funkcia na vykreslenie objektu robota do konkrétneho grafu.
- `GetRobotBasePlaneWorkspace` – funkcia na určenie pracovného priestoru robota v básovej rovine. Táto funkcia implementuje algoritmus popísaný v kapitole 2.4.1.
- `GetRobotHorizontalWorkspace` – funkcia na určenie pracovného priestoru robota v rovine XY. Táto funkcia implementuje algoritmus popísaný v kapitole 2.4.2.

2.5.5. Vizualizačná aplikácia

Vizualizačná aplikácia využíva vyššie popísané funkcie a triedy a poskytuje užívateľovi ovládacie prvky na riadenie zobrazenia robota v 3D priestore, zobrazenie jeho pomocných súradnicových systémov a pracovných priestorov. Všetky spomenuté vizualizácie sa zobrazujú do spoločného 3D grafu [6] v pravej časti okna aplikácie.

Aplikácia si vnútorne uchováva jeden objekt typu `RobotRRR`. Parametre robota sú načítané pri spustení aplikácie zo súboru `robot_config.json` a počiatočné uhly kĺbov sú nastavené do polovice ich rozsahu. Podľa načítaných parametrov sú upravené aj krajné hodnoty posuvníkov pre jednotlivé kĺby robota. Aktuálne natočenie kĺbov v objekte robota sa mení vždy keď dochádza k interakcii s týmito posuvníkmi.

Po interakcii s ktorýmkoľvek ovládacím prvkom aplikácie dôjde k spusteniu hlavnej vykresľovacej funkcie `app.DrawAll`. Jej činnosť pozostáva z nasledujúcich krokov:

- Vyčistenie zobrazenia.
- Vykreslenie „podlahy“.
- Vykreslenie robota v aktuálnej konfigurácii.
- Vykreslenie značiek súradnicových systémov, ktoré pozostáva z nasledujúcich krokov pre každý k -ty súradnicový systém ($k = 0, 1, 2, \dots, 6$):
 - Ak nie je aktívny k -ty checkbox v sekcii „Coordinate system markers“, vykresľovanie sa vynechá.
 - Vytvorí sa nová matica C s požadovanou dĺžkou čiar funkciou `CoordMarker.Create`.
 - Ak $k \neq 0$, body v matici C sa transformujú zo súradnicového systému k do súradnicového systému 0 pomocou funkcie `TransformFrom` v objekte robota (objekt typu `RobotRRR`).

- Výsledná matica sa vykreslí funkciou `CoordMarker.Draw`.
- Ak je aktívne vykresľovanie pracovného priestoru v rovine XY, tento pracovný priestor sa vykreslí.
- Ak je aktívne vykresľovanie pracovného priestoru v bázeovej rovine robota, tento pracovný priestor sa vykreslí.
- Zmeria sa čas vykonávania celej vykresľovacej funkcie a jeho prevrátená hodnota sa zapíše do ukazovateľa snímkov za sekundu (FPS).

3. ZHODNOTENIE

Cieľom zadania bolo precvičenie riešenia priamej kinematickej úlohy pre robotické rameno typu RRR a návrh vizualizačnej aplikácie. Pri kinematickom rozbere bolo definovaných pre toto rameno 6 pomocných súradnicových systémov a s nimi súvisiacich 6 homogénnych transformačných matic. Pre označenie a vizualizáciu týchto súradnicových systémov bola definovaná maticová reprezentácia značky súradnicového systému, ktorú je možné po transformácii vykresliť v ktoromkoľvek súradnicovom systéme robota. Na záver teoretickej časti boli navrhnuté algoritmy na vykreslenie pracovného priestoru robota v bázej rovine a v rovine XY.

Teoretické znalosti boli následne implementované v prostredí MATLAB vo forme samostatných knižníc, tried a funkcií, ktoré boli potom využité pri návrhu vizualizačnej aplikácie pomocou nástroja App Designer. Navrhnutá aplikácia umožňuje užívateľovi zobrazíť robota v ľubovoľnej konfigurácii, t. j. s ľubovoľným natočením jednotlivých kĺbov. Ovládanie kĺbov robota posuvníkmi umožňuje nielen zobrazenie robota v statickej konfigurácii, ale aj zobrazenie plynulého pohybu robota pri rotácii v jednom kĺbe. Taktiež je možné zobrazíť alebo skryť značky svetového aj všetkých pomocných súradnicových systémov a pracovný priestor v bázej rovine robota a v rovine XY. Navrhnutá aplikácia spĺňa všetky dizajnové požiadavky uvedené v zadaní.

Na záver bola vypracovaná Užívateľská dokumentácia k aplikácii ako aj Programátorská dokumentácia k vytvoreným knižniciam, triedam a funkciám.

4. UŽÍVATEĽSKÁ DOKUMENTÁCIA

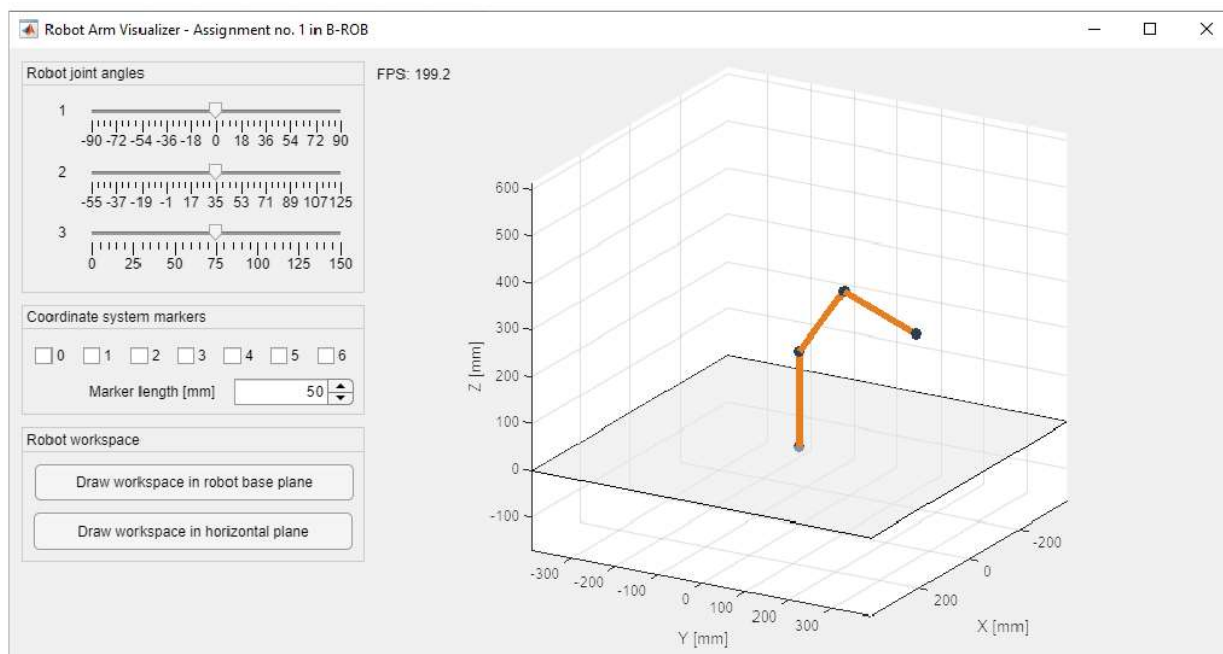
4.1. Potrebný softvér tretích strán

Na spustenie programu zo zdrojového kódu je potrebné mať nainštalované prostredie MATLAB vo verzii R2020b alebo novšej. Kompatibilita so staršími verziami nie je garantovaná.


4.2. Spustenie programu

Program spustíte vo Windows Prieskumníku dvojitým kliknutím na súbor RobotArmVisualizer.mlapp. Alternatívne môžete program spustiť z prostredia MATLAB príkazom RobotArmVisualizer, ak sa nachádzate v priečinku s programom.

4.3. Prehľad užívateľského rozhrania programu



Obr. 7: Užívateľské rozhranie programu RobotArmVisualizer

Na pravej strane okna programu sa nachádza 3D zobrazenie robotického ramena. Ak presuniete myš do tejto časti okna a kurzor myši sa zmení na , môžete držaním ľavého tlačidla a ťahaním myši otáčať 3D zobrazenie.

Na ľavej strane sa nachádzajú ovládacie prvky zobrazenia rozdelené do troch sekcií. Ich funkcionlita je popísaná nižšie.

4.4. Zmena natočenia kĺbov robota

Natočenie jednotlivých kĺbov robota je ovládané posuvníkmi v sekcií „Robot joint angles“. Posuvníky sú očíslované podľa príslušných kĺbov robota.

4.5. Zobrazenie pomocných súradnicových sústav

V sekcii „Coordinate system markers“ môžete ľubovoľne zapínať a vypínať zobrazenie súradnicových systémov robota vrátane svetového súradnicového systému (0) a súradnicového systému koncového bodu robota (6). Súradnicové systémy sú zobrazené tromi čiarami vychádzajúcimi zo stredu danej súradnicovej sústavy – červená čiara označuje smer X, zelená čiara smer Y a modrá čiara smer Z. Veľkosť súradnicových značiek môžete zmeniť v poli Marker length, ktorého hodnota udáva dĺžku jednotlivých čiar v milimetroch.

4.6. Zobrazenie pracovného priestoru robota

V sekcii „Robot workspace“ sa nachádzajú tlačidlá na zobrazenie/skrytie pracovných priestorov robota.

4.6.1. Pracovný priestor v básovej rovine

Tlačidlom „Draw workspace in robot base plane“ zobrazíte pracovný priestor robota v básovej rovine, t. j. rovina tvorená kĺbmi robota. Pracovný priestor sa zobrazí zelenou farbou a jeho poloha závisí od aktuálnej konfigurácie robota, konkrétne od natočenia prvej osi.

Po vykreslení zostane tlačidlo svietiť zelenou farbou až kým neskryjete zobrazenie pracovného priestoru v básovej rovine, čo môžete urobiť opätovným stlačením tlačidla. Pracovný priestor sa automaticky skryje pokiaľ zmeníte natočenie prvej osi robota (viď. sekcia 4.4).

4.6.2. Pracovný priestor v rovine XY

Tlačidlom „Draw workspace in horizontal plane“ zobrazíte pracovný priestor robota v rovine XY. Pracovný priestor sa zobrazí modrou farbou, rovnako aj tlačidlo bude svietiť modrou farbou. Pracovný priestor v rovine XY skryjete opätovným stlačením tlačidla.

4.7. Ukazovateľ plynulosti vykresľovania

V ľavom hornom rohu 3D zobrazenia sa nachádza ukazovateľ rýchlosti vykresľovania alebo „FPS counter“. Číslo v ukazovateli vychádza z doby trvania posledného 3D vykreslenia a je prepočítané na počet snímkov (vykreslení) za sekundu.

5. PROGRAMÁTORSKÁ DOKUMENTÁCIA

5.1. StdMatrix.m

Statická knižnica štandardných homogénnych transformačných matic.

Funkcie

`T = StdMatrix.Tx(d)`

Vráti maticu T translácie v smere X o vzdialenosť d .

`T = StdMatrix.Ty(d)`

Vráti maticu T translácie v smere Y o vzdialenosť d .

`T = StdMatrix.Tz(d)`

Vráti maticu T translácie v smere Z o vzdialenosť d .

`T = StdMatrix.Rx(fi)`

Vráti maticu T rotácie okolo osi X o uhol fi . Parameter fi musí byť zadáný v stupňoch.

`T = StdMatrix.Ry(fi)`

Vráti maticu T rotácie okolo osi Y o uhol fi . Parameter fi musí byť zadáný v stupňoch.

`T = StdMatrix.Rz(fi)`

Vráti maticu T rotácie okolo osi Z o uhol fi . Parameter fi musí byť zadáný v stupňoch.

5.2. RobotRRR.m

Trieda pre manipulátor typu RRR. Uchováva parametre a konfiguráciu robota a poskytuje funkcie na výpočet a využitie jeho transformačných matíc.

Parametre (iba na čítanie)

`lengths`

Vektor 1x3 obsahujúci dĺžky článkov robota v milimetroch.

`angles`

Vektor 1x3 obsahujúci aktuálne natočenie kĺbov robota v stupňoch.

`angleLimitsLow`

Vektor 1x3 obsahujúci spodné hranice natočenia kĺbov robota v stupňoch.

`angleLimitsHigh`

Vektor 1x3 obsahujúci horné hranice natočenia kĺbov robota v stupňoch.

Konštruktor

`obj = RobotRRR(lengths, angleLimitsLow, angleLimitsHigh, initAngles)`

Vytvorí nový objekt `obj` typu `RobotRRR` s uvedenými parametrami alebo vyhlási chybu, ak sú zadané parametre neplatné.

Vstupné parametre:

- `lengths` – dĺžky článkov robota v milimetroch, vektor 1x3.
- `angleLimitsLow` – spodné hranice natočenia kĺbov robota v stupňoch, vektor 1x3.
- `angleLimitsHigh` – horné hranice natočenia kĺbov robota v stupňoch, vektor 1x3.
- `initAngles` – počiatočné natočenie kĺbov robota v stupňoch, vektor 1x3.

Funkcie

`obj = WithAngles(obj, angles)`

Vráti objekt robota `obj` so zmenenými uhlami natočenia kĺbov.

Vstupné parametre:

- `obj` – objekt robota, na ktorom sa funkcia vykoná; tento parameter dosadzuje MATLAB implicitne, programátor ho nepíše.
- `angles` – nové uhly natočenia kĺbov robota v stupňoch, vektor 1x3.

`obj = CalcTcpVector(obj)`

Pomocná funkcia, vráti objekt robota `obj` s vypočítaným polohovým vektorom koncového bodu robota vo svetových súradniciach. Pokiaľ už pre aktuálnu konfiguráciu robota bol tento vektor vypočítaný, kalkulácia sa vynechá.

Vstupné parametre:

- `obj` – objekt robota, na ktorom sa funkcia vykoná; tento parameter dosadzuje MATLAB implicitne, programátor ho nepíše.

```
obj = CalcTMatrices(obj)
```

Pomocná funkcia, vráti objekt robota `obj` s vypočítanými transformačnými maticami $T_{0,1}$, $T_{0,2}$ až $T_{0,6}$. Pokiaľ už pre aktuálnu konfiguráciu robota boli tieto matice vypočítané, kalkulácia sa vynechá.

Vstupné parametre:

- `obj` – objekt robota, na ktorom sa funkcia vykoná; tento parameter dosadzuje MATLAB implicitne, programátor ho nepíše.

```
worldCoords = TransformFrom(obj, coordSystem, coordinates)
```

Transformuje body definované maticou stĺpcových vektorov `coordinates` zo súradnicového systému robota `coordSystem` do svetového súradnicového systému. Návratová hodnota `worldCoords` je matica stĺpcových vektorov $4 \times n$, kde n je počet stĺpcov parametra `coordinates`.

Vstupné parametre:

- `obj` – objekt robota, na ktorom sa funkcia vykoná; tento parameter dosadzuje MATLAB implicitne, programátor ho nepíše.
- `coordSystem` – číslo súradnicového systému robota, z ktorého sa majú body transformovať. Musí byť celé číslo z rozsahu 1 až 6.
- `coordinates` – matica $4 \times n$, kde každý stĺpec je stĺpcovým vektorom polohy jedného bodu v súradnicovom systéme `coordSystem`. Funkcia umožňuje jedným volaním transformovať od jedného po ľubovoľný počet bodov (n).

```
worldCoords = GetJointWorldCoords(obj)
```

Vráti súradnice všetkých kĺbov robota a koncového bodu vo svetovom súradnicovom systéme. Matica `worldCoords` má rozmer 4×4 a je zložená zo 4 stĺpcových vektorov, kde prvý stĺpec udáva polohu prvého kĺbu, druhý stĺpec polohu druhého kĺbu, tretí stĺpec polohu tretieho kĺbu a posledný stĺpec polohu koncového bodu robota.

Vstupné parametre:

- `obj` – objekt robota, na ktorom sa funkcia vykoná; tento parameter dosadzuje MATLAB implicitne, programátor ho nepíše.

```
worldCoords = GetTcpWorldCoords(obj)
```

Vráti súradnice koncového bodu robota vo svetovom súradnicovom systéme. Vektor `worldCoords` je stĺpcový polohový vektor s rozmerom 4x1.

Vstupné parametre:

- `obj` – objekt robota, na ktorom sa funkcia vykoná; tento parameter dosadzuje MATLAB implicitne, programátor ho nepíše.

5.3. CoordMarker.m

Statická knižnica pre prácu so značkami súradnicových systémov.

Funkcie

```
coords = CoordMarker.Create(size)
```

Vráti maticu `coords` s rozmerom 4x4, ktorá obsahuje stĺpcové polohové vektory pre stred súradnicovej sústavy a konce jednotlivých čiar v smeroch x , y a z .

Vstupné parametre:

- `size` – udáva veľkosť značky, t. j. dĺžku jednotlivých čiar.

```
CoordMarker.Draw(marker, axes)
```

Vykreslí značku súradnicového systému danú maticou `marker` do grafu `axes`.

Vstupné parametre:

- `marker` – matica 4x4 obsahujúca stĺpcové polohové vektory pre stred súradnicovej sústavy a konce jednotlivých čiar v smeroch x , y a z .
- `axes` – objekt grafu, do ktorého sa má značka vykresliť.

5.4. DrawRobot.m

```
DrawRobot(axes, robot)
```

Funkcia na vykreslenie robota do konkrétneho grafu. Články robota kreslí oranžovou farbou, kĺby robota a koncový bod kreslí sivým krúžkom.

Vstupné parametre:

- `axes` – objekt grafu, do ktorého sa má značka vykresliť.
- `robot` – objekt typu `RobotRRR` popisujúci robota, ktorý sa má vykresliť.

5.5. GetRobotBasePlaneWorkspace.m

```
points = GetRobotBasePlaneWorkspace(robot, angleStep)
```

Funkcia, ktorá vypočíta maticu $4 \times n$ bodov `points` na okraji pracovného priestoru robota v bázej rovine. Jednotlivé body v matici `points` sú definované stĺpcovými polohovými vektormi; každý stĺpec matice zodpovedá jednému bodu.

Vstupné parametre:

- `robot` – objekt typu `RobotRRR` popisujúci robota, ktorého pracovný priestor sa má určiť.
- `angleStep` – krok v stupňoch, s akým sa majú meniť natočenia kĺbov robota pri určovaní pracovného priestoru.

5.6. GetRobotHorizontalWorkspace.m

```
points = GetRobotHorizontalWorkspace(robot, angleStep)
```

Funkcia, ktorá vypočíta maticu $4 \times n$ bodov `points` na okraji pracovného priestoru robota v rovine XY. Jednotlivé body v matici `points` sú definované stĺpcovými polohovými vektormi; každý stĺpec matice zodpovedá jednému bodu.

Vstupné parametre:

- `robot` – objekt typu `RobotRRR` popisujúci robota, ktorého pracovný priestor sa má určiť.
- `angleStep` – krok v stupňoch, s akým sa majú meniť natočenia kĺbov robota pri určovaní pracovného priestoru.

6. ZDROJE A ODKAZY

- [1] Hlavná stránka produktu MATLAB. online:
<https://www.mathworks.com/products/matlab.html>
- [2] Hlavná stránka nástroja MATLAB App Designer. online:
<https://www.mathworks.com/products/matlab/app-designer.html>
- [3] Objektovo orientované programovanie v prostredí MATLAB. online:
<https://www.mathworks.com/products/matlab/object-oriented-programming.html>
- [4] Dokumentácia ku tvorbe tried (class) v prostredí MATLAB. online:
<https://www.mathworks.com/help/matlab/object-oriented-programming.html>
- [5] Statické metódy v MATLAB triedach. online:
https://www.mathworks.com/help/matlab/matlab_oop/static-methods.html
- [6] Dokumentácia k vykresľovaniu grafiky v prostredí MATLAB App Designer. online:
https://www.mathworks.com/help/matlab/creating_guis/graphics-support-in-app-designer.html