

Slovenská Technická Univerzita – Fakulta elektrotechniky a informatiky
Ilkovičova 3, 812 19 Bratislava

Vizualizácia priamej kinematickej úlohy

Robotika

Vizualizácia priamej kinematickej úlohy

Zadanie: Hasičské auto s výsuvnou plošinou.

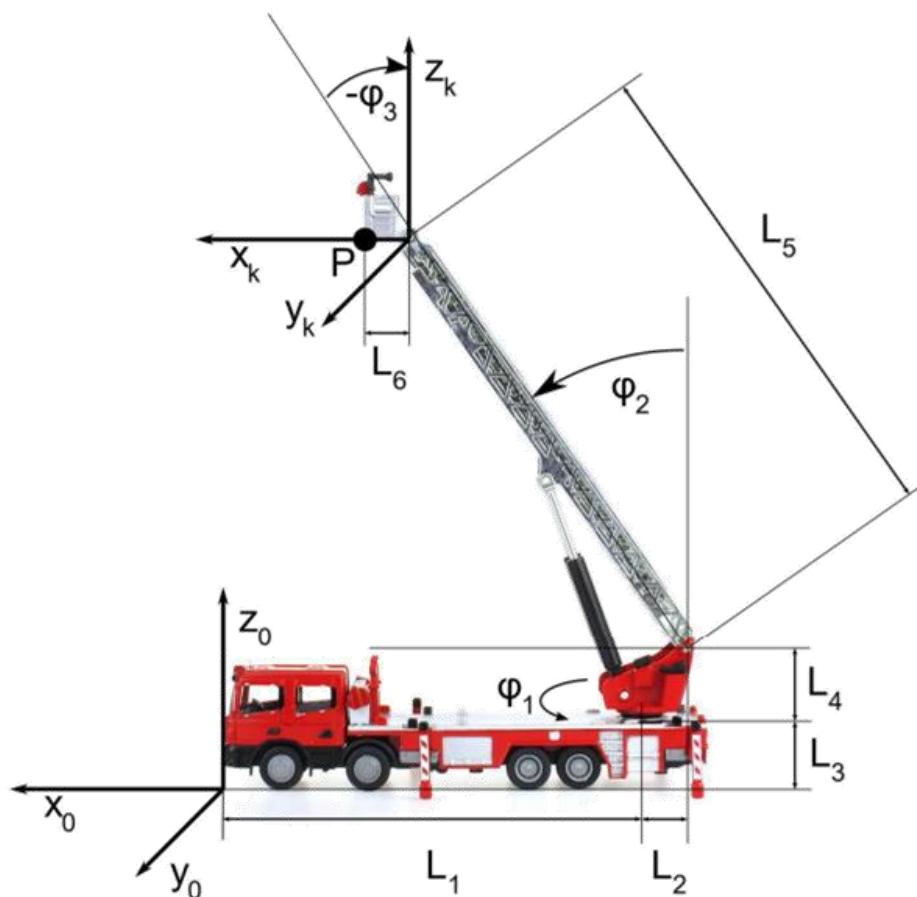
Zariadenie tohto typu slúži na evakuáciu osôb z výškových budov. Je preto nesmierne dôležité pomôcť požiarnikom pri realizácii výpočtov pri polohovaní takéhoto zariadenia. Aj vy práve teraz môžete prispieť vašimi robotickými vedomosťami k záchrane ľudských životov. 😊

Majme svetový súradnicový systém $x_0y_0z_0$ podľa obrázka (Obr. 1). V koncovom súradnicovom systéme $x_ky_kz_k$ sa nachádza bod P, ktorý predstavuje aj koncový bod plošiny resp. kontaktný bod plošiny s budovou.

Štyri hydraulické pohony zabezpečujú: rotáciu základne (φ_1), nakláňanie rebríka (φ_2), vysúvanie rebríka (L_5) a nakláňanie plošiny (φ_3).

Konfigurácia takejto automobilovej plošiny pri prevoze nasledovná:

$L_5 = L_{5\min}$, $\varphi_1 = 0$, $\varphi_2 = -90^\circ$ a $\varphi_3 = -180^\circ$.



Obr. 1 Požiarna výšková automobilová plošina

Rozbor úlohy

Úloha sa zaoberá **priamou kinematickou úlohou** hasičského auta s výsuvnou plošinou. Cieľom je:

1. Odvodiť **transformačnú maticu** medzi svetovým a koncovým súradnicovým systémom.
2. Vypočítať **polohu koncového bodu P** (miesta, kde plošina dotýka budovu).
3. Vizualizovať mechanizmus a **pracovný priestor** v dvoch rovinách (X_0Z_0 a X_0Y_0).

Najdôležitejšie je si uvedomiť, že výsuvná plošina hasičského auta má 4 pohyblivé časti:

- **Rotácia základne** (uhol φ_1)
- **Nakláňanie rebríka** (uhol φ_2)
- **Vysúvanie rebríka** (dĺžka L_5)
- **Nakláňanie plošiny** (uhol φ_3)

Každý pohyb sa dá popísať rotačnou alebo translačnou maticou. Vďaka vynásobeniu všetkých matíc v poradí od základne až po koniec vieme zistiť výslednú maticu T_{0k} . Treba ale podotknúť, že nie len pohyblivé časti spadajú do výslednej matice T_{0k} . Keďže náš súradnicový systém x_0, y_0, z_0 začína pri prednom nárazníku hasičského auta, treba do výslednej matice zaradiť dĺžku začiatku vozidla po stred základne, ktorá rotuje. Taktiež je potrebné si uvedomiť, že základňa, ktorá rotuje nie je priamo na zemi a tento pohyb treba taktiež brať do T_{0k} spolu s tým, že bod kde je zabezpečené nakláňanie rebríka je vo výške L_4 s posunom L_2 od základne, ktorá rotuje. Pri správnom určení naša výsledná matica bude T_{0k} vyzeráť:

$$T_{0k} = T_{08} = T_x(-L_1) * T_z(L_3) * R_z(\varphi_1) * T_x(-L_2) * T_z(L_4) * R_y(\varphi_2) * T_z(L_5) * R_y(\varphi_3)$$

A náš výsledný vektor pre bod \vec{P}_8 bude:

$$\vec{P}_8 = \begin{pmatrix} L_6 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Pomocou výslednej matice a aj výsledného vektora, vieme zapísať vzťah pre \vec{P}_0 :

$$\vec{P}_0 = T_{08} * \vec{P}_8$$

$$\vec{P}_0 = T_x(-L_1) * T_z(L_3) * R_z(\varphi_1) * T_x(-L_2) * T_z(L_4) * R_y(\varphi_2) * T_z(L_5) * R_y(\varphi_3) * \begin{pmatrix} L_6 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Pre uľahčenie výpočtov si vieme vyššie uvedený vzťah rozdeliť pomocou substitúcie na:

$$A = T_x(-L_1) * T_z(L_3) * R_z(\varphi_1)$$

$$B = T_x(-L_2) * T_z(L_4) * R_y(\varphi_2)$$

$$C = T_z(L_5) * R_y(\varphi_3)$$

$$A = \left[\begin{pmatrix} 1 & 0 & 0 & -L1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L3 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos(\varphi_1) & -\sin(\varphi_1) & 0 & 0 \\ \sin(\varphi_1) & \cos(\varphi_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right]$$

$$A = \begin{pmatrix} \cos(\varphi_1) & -\sin(\varphi_1) & 0 & -L1 \\ \sin(\varphi_1) & \cos(\varphi_1) & 0 & 0 \\ 0 & 0 & 1 & L3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$B = \left[\begin{pmatrix} 1 & 0 & 0 & -L2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L4 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos(\varphi_2) & 0 & \sin(\varphi_2) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi_2) & 0 & \cos(\varphi_2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right]$$

$$B = \begin{pmatrix} \cos(\varphi_2) & 0 & \sin(\varphi_2) & -L2 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi_2) & 0 & \cos(\varphi_2) & L4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$C = \left[\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L5 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos(\varphi_3) & 0 & \sin(\varphi_3) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi_3) & 0 & \cos(\varphi_3) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right]$$

$$C = \begin{pmatrix} \cos(\varphi_3) & 0 & \sin(\varphi_3) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi_3) & 0 & \cos(\varphi_3) & L5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Tieto 3 substitúcie môžeme následne vyjadriť ako jednu veľkú maticu T_{08} :

$$T_{08} = A * B * C$$

$$T_{08} = \begin{pmatrix} \cos(\varphi_1) & -\sin(\varphi_1) & 0 & -L1 \\ \sin(\varphi_1) & \cos(\varphi_1) & 0 & 0 \\ 0 & 0 & 1 & L3 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos(\varphi_2) & 0 & \sin(\varphi_2) & -L2 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi_2) & 0 & \cos(\varphi_2) & L4 \\ 0 & 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} \cos(\varphi_3) & 0 & \sin(\varphi_3) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\varphi_3) & 0 & \cos(\varphi_3) & L5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{08} = \begin{pmatrix} \cos(\varphi_1) \cos(\varphi_2 + \varphi_3) & -\sin(\varphi_1) & \cos(\varphi_1) \sin(\varphi_2 + \varphi_3) & -L1 + \cos(\varphi_1) * [L5 \sin(\varphi_2) - L2] \\ \sin(\varphi_1) \cos(\varphi_2 + \varphi_3) & \cos(\varphi_1) & \sin(\varphi_1) \sin(\varphi_2 + \varphi_3) & -\sin(\varphi_1) * [-L5 \sin(\varphi_2) + L2] \\ -\sin(\varphi_3 + \varphi_2) & 0 & \cos(\varphi_2 + \varphi_3) & L3 + L4 + L5 \cos(\varphi_2) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Takže po vyjadrení vzťah bude vyzerat':

$$\vec{P_0} = T_{08} * \begin{pmatrix} L6 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\vec{P_0} = \begin{pmatrix} L6 * \cos(\varphi_1) \cos(\varphi_2 + \varphi_3) - L_1 + \cos(\varphi_1) * [L_5 \sin(\varphi_2) - L2] \\ L6 * \sin(\varphi_1) \cos(\varphi_2 + \varphi_3) - \sin(\varphi_1) * [-L_5 \sin(\varphi_2) + L2] \\ -L6 * \sin(\varphi_3 + \varphi_2) + L_3 + L_4 + L_5 \cos(\varphi_2) \\ 1 \end{pmatrix}$$

Vďaka aj výslednej matici si vieme určiť vektory $\vec{a}, \vec{b}, \vec{c}$, ktoré použijeme pri zobrazení výsuvnej plošiny v prostredí MATLAB:

$$\vec{a} = R_z(\varphi_1) * \vec{P_1} * \vec{P_2}$$

$$\vec{b} = R_z(\varphi_1) * T_x(-L2) * T_z(L4) * R_y(\varphi_2) * \vec{P_3}$$

$$\vec{c} = R_z(\varphi_1) * T_x(-L2) * T_z(L4) * R_y(\varphi_2) * T_z(L5) * R_y(\varphi_3) * \vec{P_4}$$

Pri čom platí, že $\vec{P_1}, \vec{P_2}, \vec{P_3}, \vec{P_4}$ budú:

$$\vec{P_1} = \begin{pmatrix} -L2 \\ 0 \\ 0 \\ 1 \end{pmatrix}; \vec{P_2} = \begin{pmatrix} 0 \\ 0 \\ L4 \\ 1 \end{pmatrix}; \vec{P_3} = \begin{pmatrix} L5 \\ 0 \\ 0 \\ 1 \end{pmatrix}; \vec{P_4} = \vec{P_9} = \begin{pmatrix} 0 \\ 0 \\ L6 \\ 1 \end{pmatrix}$$

Najskôr som si všetky matice vyjadril v prostredí MATLAB a použil ich pri výpočte pomocných vektorov pre zobrazenie častí výsuvnej plošiny.

```
TX_L1 = [  
    1 0 0 -L1;  
    0 1 0 0;  
    0 0 1 0;  
    0 0 0 1;  
];  
  
TZ_L3 = [  
    1 0 0 0;  
    0 1 0 0;  
    0 0 1 L3;  
    0 0 0 1;  
];  
  
TX_L2 = [  
    1 0 0 -L2;  
    0 1 0 0;  
    0 0 1 0;  
    0 0 0 1;  
];  
  
TZ_L4 = [  
    1 0 0 0;  
    0 1 0 0;  
    0 0 1 L4;  
    0 0 0 1;  
];  
  
TZ_L5 = [  
    1 0 0 0;  
    0 1 0 0;  
    0 0 1 L5;  
    0 0 0 1;  
];  
  
Rz = [cosd(fi1) -sind(fi1) 0 0; sind(fi1) cosd(fi1) 0 0; 0 0 1 0; 0 0 0 1];  
Ry2 = [cosd(fi2) 0 sind(fi2) 0; 0 1 0 0; -sind(fi2) 0 cosd(fi2) 0; 0 0 0 1];  
Ry3 = [cosd(fi3) 0 sind(fi3) 0; 0 1 0 0; -sind(fi3) 0 cosd(fi3) 0; 0 0 0 1];
```

Najprv si z danej rovnice postupne dosadíme povolené hodnoty premenných φ_1 , φ_2 a φ_3 . Tieto hodnoty nám vytvoria body v priestore, ktoré môžeme následne vykresliť v súradnicových systémoch XY a YZ. Výsledný priestor bude reprezentovať oblasť, v ktorej sa naša hasičská plošina dokáže pohybovať. Nižšie uvedený kód slúži na vykreslenie pracovný priestor XY, ktorý sa ukladá postupne do súboru *XYpoints.txt*. Rovnakou logikou je aj vyjadrenie XZ pracovného bodu:

```

for i = 0: 10: 360
    for j = 0 : 30 : 90
        k=0;
        for l = 0: 1 : 50
            Rz = [sind(i) -cosd(i) 0 0; cosd(i) sind(i) 0 0; 0 0 1 0; 0 0 0 1];
            Ry2 = [cosd(j) 0 sind(j) 0; 0 1 0 0; -sind(j) 0 cosd(j) 0; 0 0 0 1];
            Ry3 = [cosd(k) 0 sind(k) 0; 0 1 0 0; -sind(k) 0 cosd(k) 0; 0 0 0 1];

            % pomocne vypocty
            Tz_L5 = [1 0 0 0;
                    0 1 0 0;
                    0 0 1 1;
                    0 0 0 1];
            A = Tx_L1*Tz_L3*Rz*Tx_L2*Tz_L4;
            B = Tx_L1*Ry2*Tz_L5;
            C = Ry3*p4_vekt;

            P_XY = A*B*C;

            % uloženie do premenných
            x_points = [x_points; P_XY(1)];
            y_points = [y_points; P_XY(2)];
            z_points = [z_points; P_XY(3)];

            % okamžitý zápis do súboru
            fprintf(fileID, '%f %f %f\n', P_XY(1), P_XY(2), P_XY(3));
        end
    end
end
end

```

S vykreslenými pracovnými priestormi ich vieme premietnuť do 3D priestoru:

```

fill3(y2_points(f), zeros(size(y2_points(f))), z2_points(f), 'm', 'FaceAlpha', 0.2);
fill3(x_points(e), y_points(e), zeros(size(y2_points(e))), 'b', 'FaceAlpha', 0.2);

```

Taktiež môžeme obohatiť vykreslenie plošiny s označením pomocných súradnicových systémov. V kóde je označenie A,B,C kde napr. A predstavuje súradnicový systém, ktorý zabezpečuje rotáciu plošiny okolo osi Z. Nižšie uvedená časť kódu približuje zobrazenie označenia osí pomocného súradnicového systému:

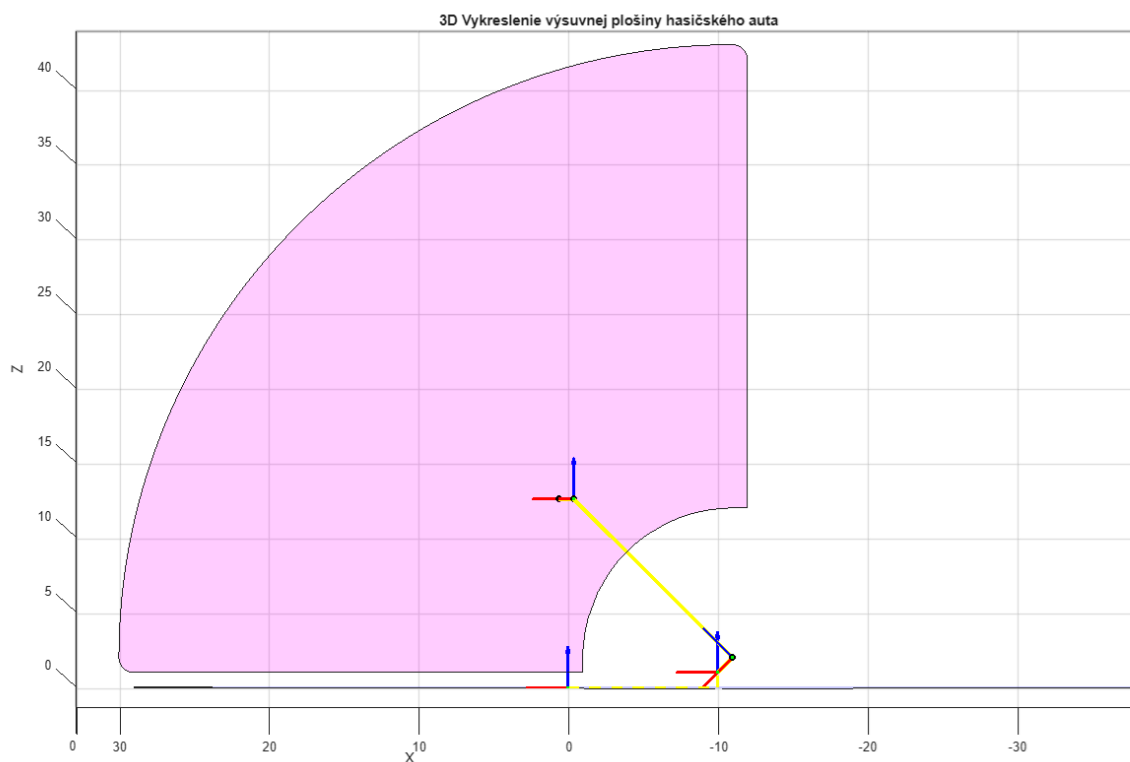
```
% Pre A
scatter3(a_vekt(1), a_vekt(2), a_vekt(3), markersize, markerColor, 'filled');
origin = [zaciatok(1,4), zaciatok(2,4), zaciatok(3,4)];
% ošetrenie rotácie

vecX = [scale; 0; 0; 0];
vecY = [0; scale; 0; 0];
vecZ = [0; 0; scale; 0];

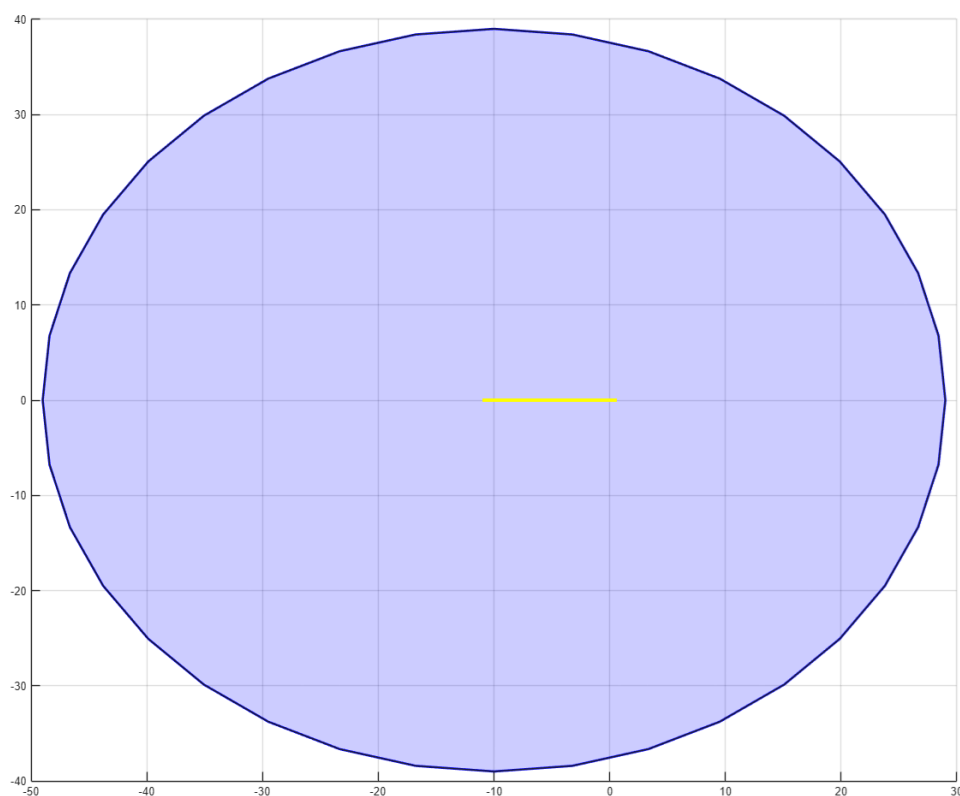
rotX = Rz * vecX;
rotY = Rz * vecY;
rotZ = Rz * vecZ;

quiver3(origin(1), origin(2), origin(3), rotX(1), rotX(2), rotX(3), 'r', 'LineWidth', 2,
'MaxHeadSize', 0.5);
quiver3(origin(1), origin(2), origin(3), rotY(1), rotY(2), rotY(3), 'g', 'LineWidth', 2,
'MaxHeadSize', 0.5);
quiver3(origin(1), origin(2), origin(3), rotZ(1), rotZ(2), rotZ(3), 'b', 'LineWidth', 2,
'MaxHeadSize', 0.5);
```

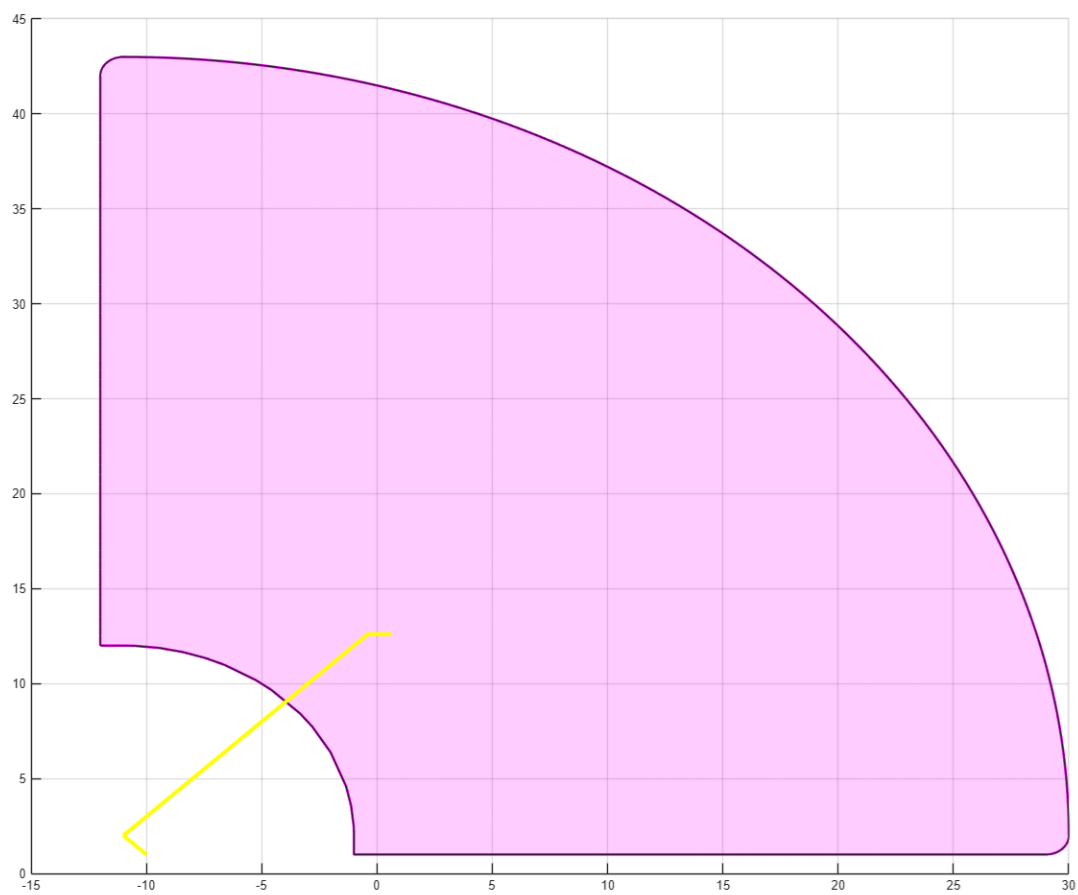
3D zobrazenie by malo vyzeráť ako je uvedené na obrázku 1.1 a zobrazenie pracovných priestorov je uvedené 1.2 a 1.3 :



1.1 3D Vykreslenie plošiny

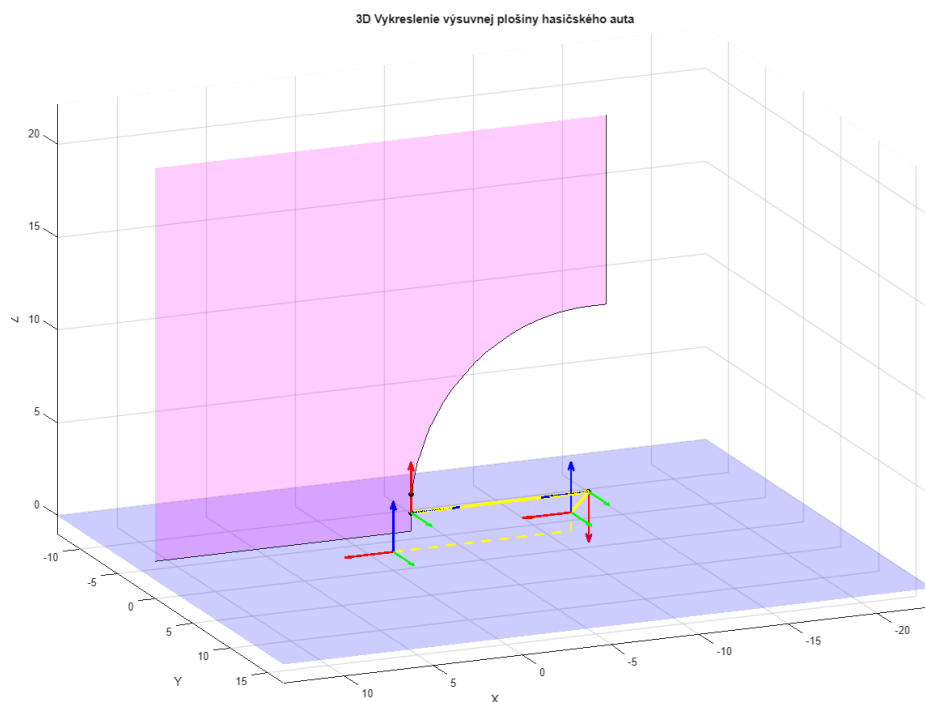


1.2 XY pracovný priestor

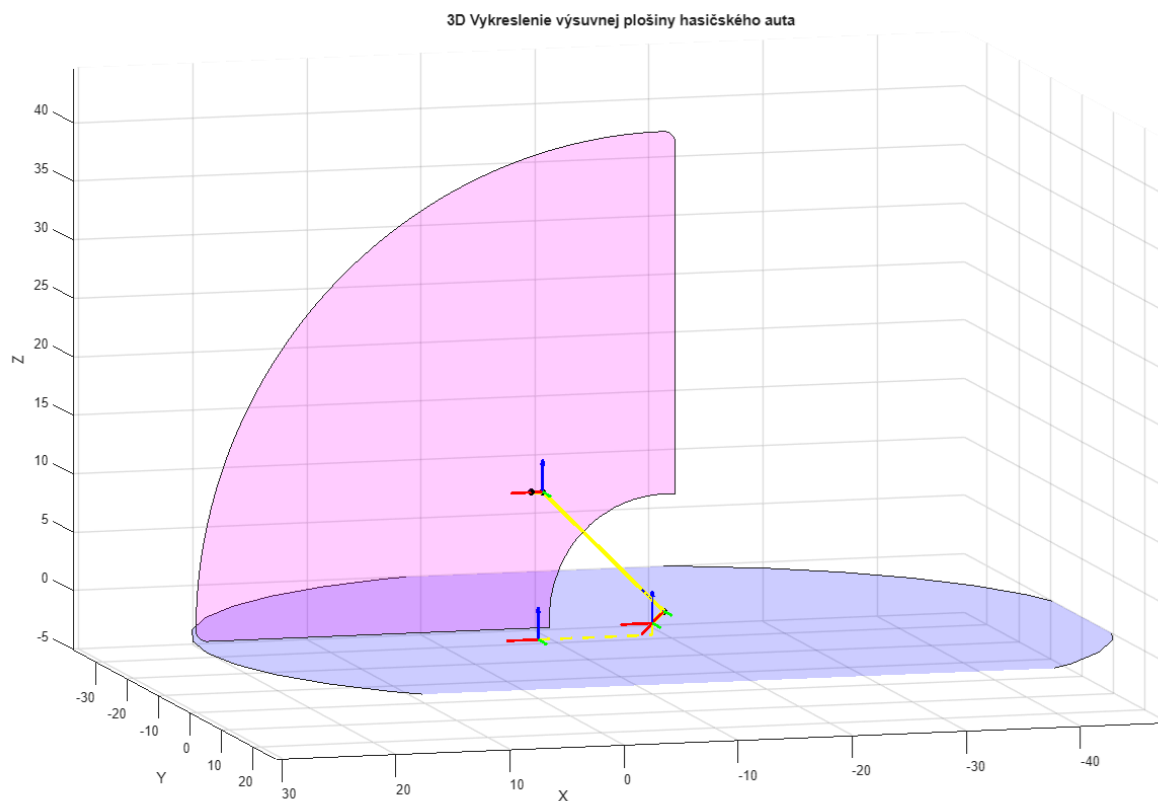


1.3 XZ pracovný priestor

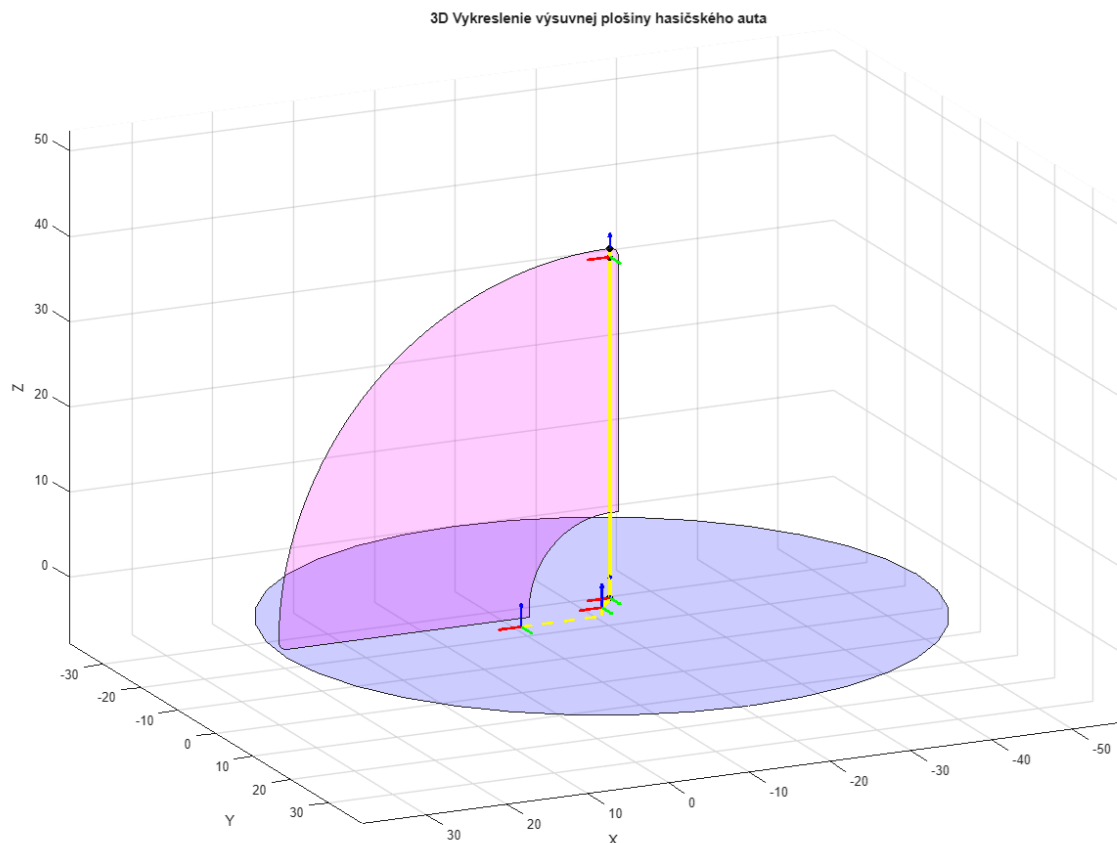
Zadanie som z klasického MATLAB scriptu taktiež upravil ako MATLAB APP, kde je zmena dĺžky rebríka alebo aj všetky uhly automaticky aktualizovaná pre lepšiu vizualizáciu. Na obrázkoch 1.4, 1.5 a 1.6 sú zobrazené rôzne polohy výsuvnej plošiny hasičského auta:



1.4 Konfigurácia automobilovej plošiny pri prevoze



1.5 Konfigurácia automobilovej plošiny podľa vzorového obrázka zadania



1.6 Konfigurácia automobilovej plošiny pri úplnom vystretí

SPUSTENIE

V prílohe sa nachádzajú dva MATLAB súbory. Manipulator3D.m slúži na čisté zobrazenie plošiny bez aktualizácie parametrov v reálnom čase. Táto funkcia je ale zabezpečená v PlosinaAPP.m. Oba scripty boli robené na MATLAB verzii 2025a, ale taktiež idú spustiť v prostredí MATLAB Online.

ZÁVER

Na základe vypracovaného zadania sme úspešne aplikovali princípy využitia translačných a rotačných matic na simuláciu pohybu výsuvnej plošiny. Simulácia nám umožnila vizualizovať pracovný priestor plošiny a lepšie pochopiť jej funkčné obmedzenia a možnosti. Riešenie úlohy v prostredí MATLAB nám navyše poskytlo cennú príležitosť zopakovať si a prehĺbiť naše zručnosti vo vykresľovaní 3D grafov, čo môže byť užitočné aj pri ďalších technických výpočtoch a návrhoch. Záverom, zadanie ponúklo praktický pohľad do analýzy pohybu a efektívneho využitia technických zariadení, ako je výsuvná plošina.

ČESTNÉ PREHLÁSENIE

Zadanie som vypracoval sám. Čestne prehlasujem, že som ho neskopíroval a nikomu inému neposkytol. Nech mi je Isaac Asimov svedkom.

ZDROJE

Všetky zdroje, ktoré boli použité na vypracovanie zadania sú tu:

- <https://www.youtube.com/watch?v=TGK17fUA5Nw>
- <https://stackoverflow.com/questions/45270235/3d-visualization-of-some-data-in-matlab>
- <https://www.mathworks.com/help/matlab/ref/plot3.html>
- <https://www.mathworks.com/help/matlab/ref/fill3.html>
- <https://www.mathworks.com/help/matlab/ref/fopen.html>
- <https://www.mathworks.com/help/matlab/ref/readmatrix.html>
- <https://www.mathworks.com/help/matlab/ref/boundary.html>
- <https://www.mathworks.com/help/matlab/ref/quiver3.html>
- https://www.mathworks.com/help/matlab/creating_guis/apps-overview.html

Ostatné informácie sú z poznámok cvičení predmetu ROBOTIKA.

POUŽITÝ KÓD

V tejto časti je uvedený kód, ktorý som použil pre vypracovanie zadania č. 1

```
clc;
clear;

L1 = 10;
L2 = 1;
L3 = 1;
L4 = 1;
L5 = 10;
L6 = 1;

fi1= 0;
fi2= 90;
fi3= -180;

TX_L1 = [
    1 0 0 -L1;
    0 1 0 0;
    0 0 1 0;
    0 0 0 1;
];

TZ_L3 = [
    1 0 0 0;
    0 1 0 0;
    0 0 1 L3;
    0 0 0 1;
];

TX_L2 = [
    1 0 0 -L2;
```

```

        0 1 0 0;
        0 0 1 0;
        0 0 0 1;
    ];

    Tz_L4 = [
        1 0 0 0;
        0 1 0 0;
        0 0 1 L4;
        0 0 0 1;
    ];

    Tz_L5 = [
        1 0 0 0;
        0 1 0 0;
        0 0 1 L5;
        0 0 0 1;
    ];

    Rz = [cosd(fi1) -sind(fi1) 0 0; sind(fi1) cosd(fi1) 0 0; 0 0 1 0; 0 0 0 1];
    Ry2 = [cosd(fi2) 0 sind(fi2) 0; 0 1 0 0; -sind(fi2) 0 cosd(fi2) 0; 0 0 0 1];
    Ry3 = [cosd(fi3) 0 sind(fi3) 0; 0 1 0 0; -sind(fi3) 0 cosd(fi3) 0; 0 0 0 1];

    p1_vekt = [-L2 0 0 1]';
    p2_vekt = [0 0 L4 1]';

    p1_plus_p2 = [-L2 0 L4 1]';
    p3_vekt = [0 0 L5 1]';
    p4_vekt = [L6 0 0 1]';

    A = Tx_L1* Tz_L3*Rz*Tx_L2*Tz_L4;
    B = Ry2* Tz_L5;
    C = Ry3*p4_vekt;

    zaciatok = Tx_L1*Tz_L3;
    a_vekt = Tx_L1* Tz_L3* Rz * p1_plus_p2;
    b_vekt = A* Ry2 * p3_vekt;
    c_vekt = A *B * C;

    filename = 'data/XYpoints.txt';
    filename2 = 'data/XZpoints.txt';

    if isfile(filename) && isfile(filename2)
        % Súbor existuje → načítanie dát
        data = readmatrix(filename);
        x_points = data(:, 1);
        y_points = data(:, 2);
        z_points = data(:, 3);

        data2 = readmatrix(filename2);
        x2_points = data2(:, 1);
        y2_points = data2(:, 2);
    end

```

```

z2_points = data2(:, 3);
disp('Dáta načítané zo súboru.');
```

else

```

fileID = fopen(filename, 'w');
fileID2 = fopen(filename2, 'a+');
```

```

x_points = [];
y_points = [];
z_points = [];

x2_points = [];
y2_points = [];
z2_points = [];
for i = 0: 10: 360
    for j = 0 : 30 : 90
        k=0;
        for l = 0: 1 : 50
            Rz = [sind(i) -cosd(i) 0 0; cosd(i) sind(i) 0 0; 0 0 1 0; 0 0 0 1];
            Ry2 = [cosd(j) 0 sind(j) 0; 0 1 0 0; -sind(j) 0 cosd(j) 0; 0 0 0 1];
            Ry3 = [cosd(k) 0 sind(k) 0; 0 1 0 0; -sind(k) 0 cosd(k) 0; 0 0 0 1];

            % pomocne vypocty
            Tz_L5 = [1 0 0 0;
                    0 1 0 0;
                    0 0 1 1;
                    0 0 0 1];
            A = Tx_L1*Tz_L3*Rz*Tx_L2*Tz_L4;
            B = Tx_L1*Ry2*Tz_L5;
            C = Ry3*p4_vekt;

            P_XY = A*B*C;

            % ulozenie do premenných
            x_points = [x_points; P_XY(1)];
            y_points = [y_points; P_XY(2)];
            z_points = [z_points; P_XY(3)];

            % okamžitý zápis do súboru
            fprintf(fileID, '%f %f %f\n', P_XY(1), P_XY(2), P_XY(3));
        end
    end
end

end
i=0;
for j = 0 : 1 : 90
    if (j == 0 || j == 90)
        for k = -180:1:0
            for l = 10: 0.5 : 40
                Rz = [sind(i) -cosd(i) 0 0; cosd(i) sind(i) 0 0; 0 0 1 0; 0 0 0 1];
                Ry2 = [cosd(j) 0 sind(j) 0; 0 1 0 0; -sind(j) 0 cosd(j) 0; 0 0 0 1];
                Ry3 = [cosd(k) 0 sind(k) 0; 0 1 0 0; -sind(k) 0 cosd(k) 0; 0 0 0 1];

                % pomocne vypocty
                Tz_L5 = [1 0 0 0;
                        0 1 0 0;
                        0 0 1 1;
                        0 0 0 1];
```

```

        A = Tx_L1 * Tz_L3 * Rz * Tx_L2 * Tz_L4;
        B = Tx_L1 * Ry2 * Tz_L5;
        C = Ry3 * p4_vekt;

        P_XZ = A * B * C;

        % Uloženie bodov
        x2_points = [x2_points; P_XZ(1)];
        y2_points = [y2_points; P_XZ(2)];
        z2_points = [z2_points; P_XZ(3)];

        % Okamžitý zápis do súboru
        fprintf(fileID2, '%f %f %f\n', P_XZ(1), P_XZ(2), P_XZ(3));
    end
end
else
    k = -90;
    for l = 9: 0.5 : 40
        Rz = [sind(i) -cosd(i) 0 0; cosd(i) sind(i) 0 0; 0 0 1 0; 0 0 0 1];
        Ry2 = [cosd(j) 0 sind(j) 0; 0 1 0 0; -sind(j) 0 cosd(j) 0; 0 0 0 1];
        Ry3 = [cosd(k) 0 sind(k) 0; 0 1 0 0; -sind(k) 0 cosd(k) 0; 0 0 0 1];

        % pomocne vypocty
        Tz_L5 = [1 0 0 0; 0 1 0 0; 0 0 1 1; 0 0 0 1];
        A = Tx_L1 * Tz_L3 * Rz * Tx_L2 * Tz_L4;
        B = Tx_L1 * Ry2 * Tz_L5;
        C = Ry3 * p4_vekt;

        P_XZ = A * B * C;

        % Uloženie bodov
        x2_points = [x2_points; P_XZ(1)];
        y2_points = [y2_points; P_XZ(2)];
        z2_points = [z2_points; P_XZ(3)];

        % Okamžitý zápis do súboru
        fprintf(fileID2, '%f %f %f\n', P_XZ(1), P_XZ(2), P_XZ(3));
    end
end
end

% Zavretie súborov
fclose(fileID);
fclose(fileID2);
disp('Dáta vypočítané a uložené.');
```

```

end

%XY
figure(2);clf;
grid on;
hold on;
e = boundary(x_points, y_points, 1);
plot(x_points(e), y_points(e), 'b-', 'Linewidth', 2);
fill(x_points(e), y_points(e), 'b', 'FaceAlpha', 0.2);

plot([zaciatok(1,4), a_vekt(1)], [zaciatok(2,4), a_vekt(2)], 'y', 'Linewidth', 3);
plot([a_vekt(1), b_vekt(1)], [a_vekt(2), b_vekt(2)], 'y', 'Linewidth', 3);
plot([b_vekt(1), c_vekt(1)], [b_vekt(2), c_vekt(2)], 'y', 'Linewidth', 3);

```

```

%XZ
figure(3);clf;
grid on;
hold on;
f = boundary(y2_points, z2_points, 0.1);

plot(y2_points(f), z2_points(f), 'm-', 'Linewidth', 2);
fill(y2_points(f), z2_points(f), 'm', 'FaceAlpha', 0.2);

plot([zaciatok(1,4), a_vekt(1)], [zaciatok(3,4), a_vekt(3)], 'y', 'Linewidth', 3);
plot([a_vekt(1), b_vekt(1)], [a_vekt(3), b_vekt(3)], 'y', 'Linewidth', 3);
plot([b_vekt(1), c_vekt(1)], [b_vekt(3), c_vekt(3)], 'y', 'Linewidth', 3);

figure(1); clf;
hold on;
markerSize = 25;
markerColor = 'k';

origin = [0, 0, 0];
scale = 3; % dĺžka každej osi

% Zobrazenie označenia súradnicového systému
% Os X (červená)
quiver3(origin(1), origin(2), origin(3), scale, 0, 0, 'r', 'Linewidth', 2, 'MaxHeadSize',
0.5);
% Os Y (zelená)
quiver3(origin(1), origin(2), origin(3), 0, scale, 0, 'g', 'Linewidth', 2, 'MaxHeadSize',
0.5);
% Os Z (modrá)
quiver3(origin(1), origin(2), origin(3), 0, 0, scale, 'b', 'Linewidth', 2, 'MaxHeadSize',
0.5);

grid on;
xlabel('X');
ylabel('Y');
zlabel('Z');
title('3D vykreslenie výsuvnej plošiny hasičského auta');
axis equal;

plot3([0, Tx_L1(1,4)], [0, Tx_L1(2,4)], [0, Tx_L1(3,4)], 'y--', 'Linewidth', 2);
plot3([Tx_L1(1,4),zaciatok(1,4)], [Tx_L1(2,4),zaciatok(2,4)], [Tx_L1(3,4),zaciatok(3,4)], 'y--',
'Linewidth', 2);

plot3([zaciatok(1,4), a_vekt(1)], [zaciatok(2,4), a_vekt(2)], [zaciatok(3,4), a_vekt(3)], 'y',
'Linewidth', 3);
plot3([a_vekt(1), b_vekt(1)], [a_vekt(2), b_vekt(2)], [a_vekt(3), b_vekt(3)], 'y',
'Linewidth', 3);
plot3([b_vekt(1), c_vekt(1)], [b_vekt(2), c_vekt(2)], [b_vekt(3), c_vekt(3)], 'y',
'Linewidth', 3);

% PRE A
scatter3(a_vekt(1), a_vekt(2), a_vekt(3), markerSize, markerColor, 'filled');
origin = [zaciatok(1,4), zaciatok(2,4), zaciatok(3,4)];
% ošetrovanie rotácie

```



```

vecX = [scale; 0; 0; 0];
vecY = [0; scale; 0; 0];
vecZ = [0; 0; scale; 0];

rotX = Rz * vecX;
rotY = Rz * vecY;
rotZ = Rz * vecZ;

quiver3(origin(1), origin(2), origin(3), rotX(1), rotX(2), rotX(3), 'r', 'Linewidth', 2,
'MaxHeadSize', 0.5);
quiver3(origin(1), origin(2), origin(3), rotY(1), rotY(2), rotY(3), 'g', 'Linewidth', 2,
'MaxHeadSize', 0.5);
quiver3(origin(1), origin(2), origin(3), rotZ(1), rotZ(2), rotZ(3), 'b', 'Linewidth', 2,
'MaxHeadSize', 0.5);

% PRE B
scatter3(b_vekt(1), b_vekt(2), b_vekt(3), markersize, markerColor, 'filled');
origin = [a_vekt(1), a_vekt(2), a_vekt(3)];
% ošetrenie rotácie
vecX = [scale; 0; 0; 0];
vecY = [0; scale; 0; 0];
vecZ = [0; 0; scale; 0];

rotX = Ry2 * vecX;
rotY = Ry2 * vecY;
rotZ = Ry2 * vecZ;

quiver3(origin(1), origin(2), origin(3), rotX(1), rotX(2), rotX(3), 'r', 'Linewidth', 2,
'MaxHeadSize', 0.5);
quiver3(origin(1), origin(2), origin(3), rotY(1), rotY(2), rotY(3), 'g', 'Linewidth', 2,
'MaxHeadSize', 0.5);
quiver3(origin(1), origin(2), origin(3), rotZ(1), rotZ(2), rotZ(3), 'b', 'Linewidth', 2,
'MaxHeadSize', 0.5);

% PRE C
scatter3(c_vekt(1), c_vekt(2), c_vekt(3), markersize, markerColor, 'filled');
origin = [b_vekt(1), b_vekt(2), b_vekt(3)];
% ošetrenie rotácie
vecX = [scale; 0; 0; 0];
vecY = [0; scale; 0; 0];
vecZ = [0; 0; scale; 0];
fi3 = fi3+90;
Ry3 = [cosd(fi3) 0 sind(fi3) 0; 0 1 0 0; -sind(fi3) 0 cosd(fi3) 0; 0 0 0 1];
rotX = Ry3 * vecX;
rotY = Ry3 * vecY;
rotZ = Ry3 * vecZ;

hx=quiver3(origin(1), origin(2), origin(3), rotX(1), rotX(2), rotX(3), 'r', 'Linewidth', 2,
'MaxHeadSize', 0.5);
hy=quiver3(origin(1), origin(2), origin(3), rotY(1), rotY(2), rotY(3), 'g', 'Linewidth', 2,
'MaxHeadSize', 0.5);
quiver3(origin(1), origin(2), origin(3), rotZ(1), rotZ(2), rotZ(3), 'b', 'Linewidth', 2,
'MaxHeadSize', 0.5);

fill3(y2_points(f), zeros(size(y2_points(f))), z2_points(f), 'm', 'FaceAlpha', 0.2);
fill3(x_points(e), y_points(e), zeros(size(y2_points(e))), 'b', 'FaceAlpha', 0.2);

```

```
view(3);  
hold off;
```