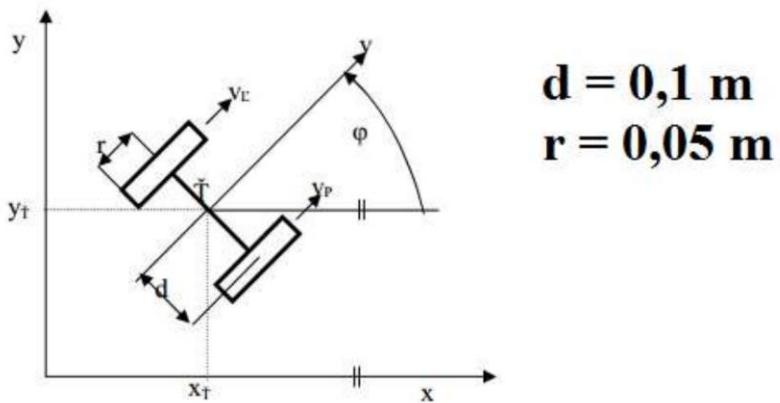


Mobilný kolesový robot

zadanie č. 3

Zadanie:

Zadanie: Navrhnite a realizujte vizualizáciu diferenciálneho podvozku. Na tomto type zadania by ste si mali precvičiť implementáciu odvodených kinematických rovníc diferenciálneho podvozku a zafixovať tak preberané učivo



Obr.1 Diferenciálny podvozok

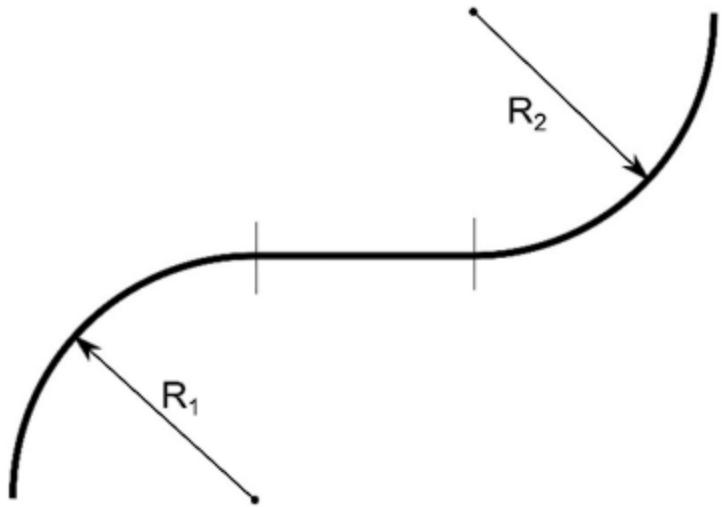
Parametre diferenciálneho podvozku:

$$L (\text{rozchod kolies}) = 200 \text{ [mm]}$$

$$r (\text{polomer kolesa}) = 50 \text{ [mm]}$$

V rámci riešenia zadania sa zamerajte na nasledovné úlohy:

1. Vykreslite trajektórie ľažiska a kolies (rôznymi farbami). Vstupným argumentom pre vykreslovanie budú vektoru: času, rýchlosťí ľavého, pravého kolesa. Majme napríklad takéto tri ľubovoľné vektoru: časový $t=[0 \ 5 \ 10 \ 15 \ 20]$, $\text{rychlosť_Laveho_kolesa}=[2 \ 0 \ 1 \ 2 \ 1]$, $\text{rychlosť_Praveho_kolesa}=[2 \ 1 \ 1 \ -2 \ 1]$. Vykreslite priebeh rýchlosťi pravého, ľavého kolesa a ľažiska v čase. Vykreslite trajektórie kolies a ľažiska.
2. Vykreslite trajektóriu štvorec prostredníctvom ľažiska robota. Dovoľte užívateľovi definovať dĺžku strany štvorca a na základe toho vygenerujte príslušné časy a rýchlosťi. Vykreslite priebeh rýchlosťi pravého, ľavého kolesa a ľažiska v čase. Vykreslite trajektórie kolies a ľažiska.
3. Vykreslite trajektóriu krivka podľa obr. 2 prostredníctvom ľažiska robota. Dovoľte užívateľovi definovať R_1, L_1, R_2 a na základe toho vygenerujte príslušné časy a rýchlosťi. Vykreslite priebeh rýchlosťi pravého, ľavého kolesa a ľažiska v čase. Vykreslite trajektórie kolies a ľažiska.
4. Vytvorte hru, kde pomocou šípok alebo W,A,S,D budete ovládať robota. Vykreslite priebeh rýchlosťi pravého, ľavého kolesa a ľažiska v čase. Vykreslite trajektórie kolies a ľažiska. Úlohy 1.-4. boli analyticky rozobraté na cvičení (pozrite PDF podklady k cvičeniam).



Obr. 2 Trajektória krivky z úlohy 3

Rozbor úlohy:

Rovnice použité na realizáciu úloh 1 až 4:

$$v_T = \frac{v_R + v_L}{2} \quad (1)$$

$$\omega_T = \frac{v_R - v_L}{L} \quad (2)$$

$$R = \frac{L}{2} \cdot \frac{v_R + v_L}{v_R - v_L} \quad (3)$$

$$v_x = v_T \cos(\phi) \quad (4)$$

$$v_y = v_T \sin(\phi) \quad (5)$$

$$\Delta x_T = v_x \cdot \Delta t \quad (6)$$

$$\Delta y_T = v_y \cdot \Delta t \quad (7)$$

$$x'_T = x_T + \Delta x_T \quad (8)$$

$$y'_T = y_T + \Delta y_T \quad (9)$$

$$\Delta\phi = \omega_T \Delta t \quad (10)$$

$$\phi' = \phi + \Delta\phi \quad (11)$$

$$v_R = v_T + \frac{\omega_T L}{2} \quad (12)$$

$$v_L = v_T - \frac{\omega_T L}{2} \quad (13)$$

Rozbor 1. úlohy:

Úlohou bolo vykresliť výsledné trajektórie ľažiska robota a trajektóriu jeho pravého a ľavého kolesa. Ako vstup sme mali zdefinované 3 vektor, jeden časový ďalšie dva boli rýchlosť kolies. Ďalší výstup mal byť priebeh rýchlosťí oboch kolies a ľažiska.

```
t_input = [0 5 10 15 20]; % Časový vektor [s]
RLKI = [2 0 1 2 1]; % Rýchlosť ľavého kolesa [m/s]
RPKI = [2 1 1 -2 1]; % Rýchlosť pravého kolesa [m/s]
```

Obr. 3 Ukážka kódu z úlohy 1

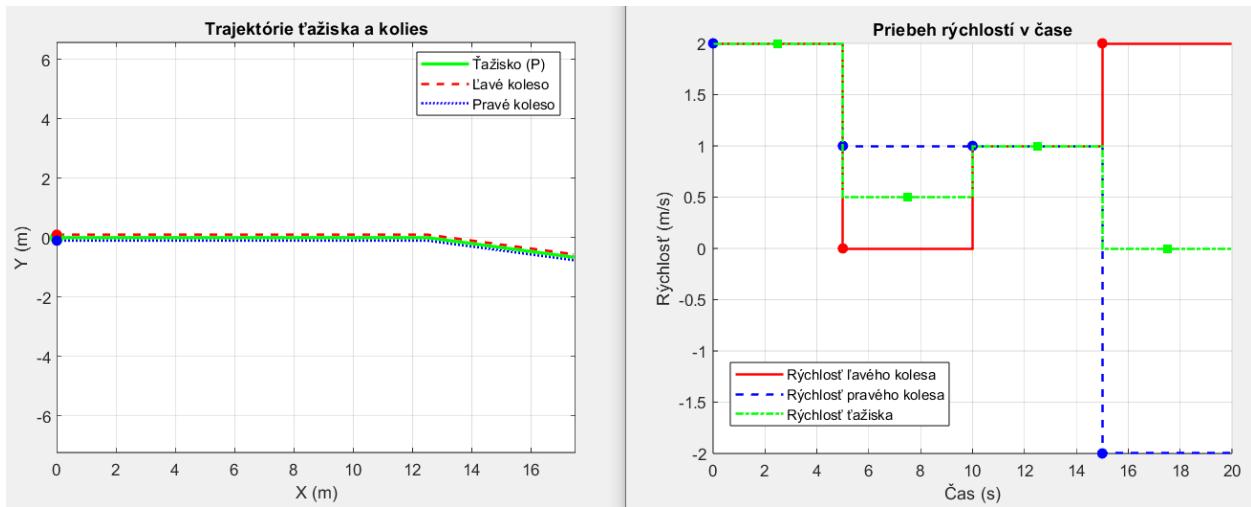
Po vytvorení vektorov som si vytvoril polia x, y a Φ do ktorých som si ukladal historiu polohy a uhlu robota. Ďalej som musel vypočítať rýchlosť ľažiska (vzorec (1) a uhlovú rýchlosť (vzorec (2) pre každý časový interval. Medzi časovými intervalmi som musel aktualizovať

pozíciu a orientáciu (alebo uhol). Následne som uložil polohu ťažiska a kolies, orientáciu a rýchlosťi.

```
x_L_hist(k+1) = x_T_current - (L/2) * sin(phi_current);
y_L_hist(k+1) = y_T_current + (L/2) * cos(phi_current);
x_R_hist(k+1) = x_T_current + (L/2) * sin(phi_current);
y_R_hist(k+1) = y_T_current - (L/2) * cos(phi_current);
```

Obr. 4 časť kodu z úlohy 1. Ukladanie polohy do poľa

Pomocou funkcie ‘figure()’ som vykreslil trajektóriu robota a jeho kolies tak isto aj priebehy rýchlosťí, ktoré vyzerajú nasledovne (viď. Obr. 5)



Obr. 5

Rozbor 2. úlohy:

Úlohou bolo vykresliť tvar štvorca, trajektóriou robota a jeho kolies. Užívateľ si má vybrať dĺžku strany dráhy štvorca. Na základe tohto máme vykresliť priebeh jednotlivých kolies a ťažiska v čase.

Ako prvé musím získať vstup od užívateľa, ak užívateľ nezdal správny vstup tak som ho nastavil na 0.5m .

```
S_strana = input('Zadajte dĺžku strany štvorca S [m]: ');
if isempty(S_strana) || ~isnumeric(S_strana) || S_strana <= 0
    S_strana = 0.5; % Predvolená hodnota, ak je vstup neplatný
    disp(['Neplatný vstup, použije sa predvolená dĺžka strany: ', num2str(S_strana), ' m']);
end
```

Obr. 6 časť scriptu kde sa načítava vstup od užívateľa

Ako prvé musí robot prejsť vzdialenosť S_strana, a po prejdení tejto vzdialenosťi sa otočiť o 90 ° čo som nastavil ako konštantu.

```
% --- Konštanty pre pohyb v Úlohe 2 ---
v_lin_task2 = 0.25; % Lineárna rýchlosť pre pohyb rovno [m/s]
omega_rot_task2 = pi/2; % Uhlová rýchlosť pre otáčanie [rad/s] (90 stupňov za 1s)
L = 0.2; % Rozchod kolies [m]
```

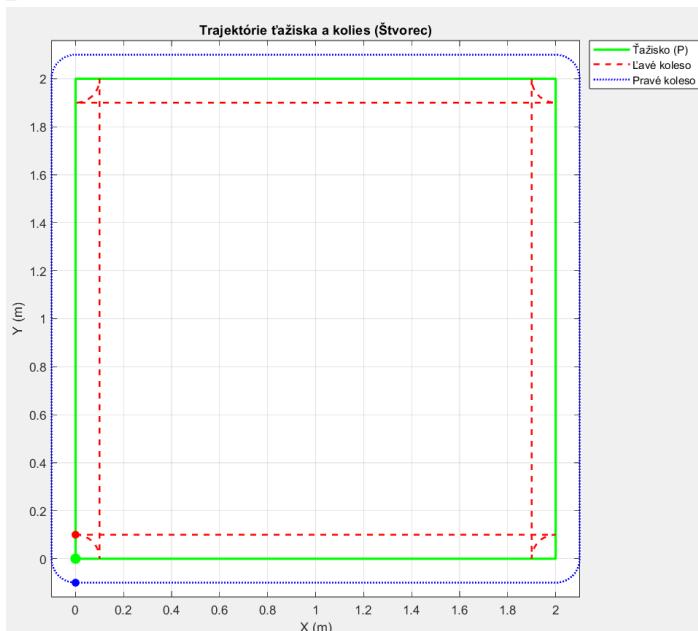
Obr. 7 časť scriptu

Z týchto krokov som vytvoril vektory časov (t_input_task2) a zodpovedajúcich rýchlosťí ľavého (RLKI_task2) a pravého (RPKI_task2) kolesa.

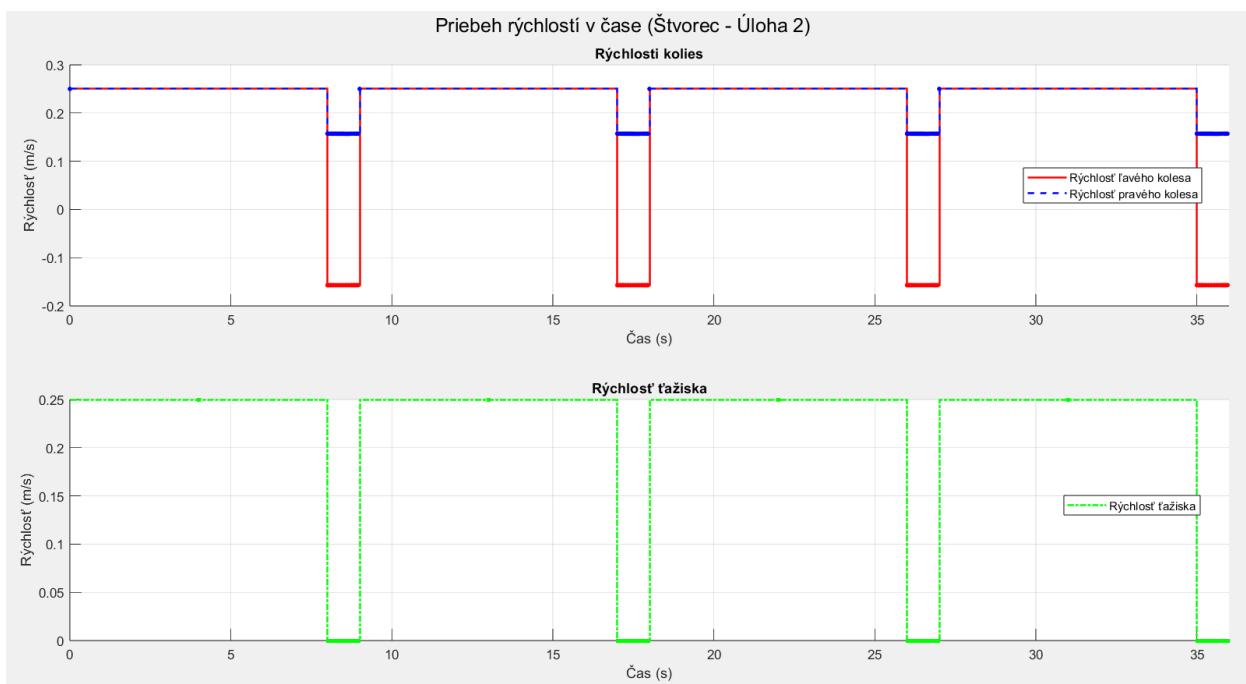
Simulácia mi vyšla krajšie ak som zvolil dĺžku strany menšiu ako 5m a to takto:

--- Úloha 2: Štvorec ---

Zadajte dĺžku strany štvorca S [m]: 2



Obr. 8 výsledok úlohy 2.



Obr. 9

Rozbor 3. úlohy:

Úlohou bolo vykresliť trajektóriu robota v tvare špecifickej krivky definovanej používateľom. Vždy to musia byť dva oblúky a rovný úsek medzi nimi, ako bolo znázornené na Obr. 2. Na základe týchto vstupov bolo potrebné vygenerovať príslušné časové priebehy a rýchlosť kolies. Výstupom mali byť grafy trajektórií ťažiska a kolies, ako aj priebehy rýchlosťí ľavého, pravého kolesa a ťažiska v čase.

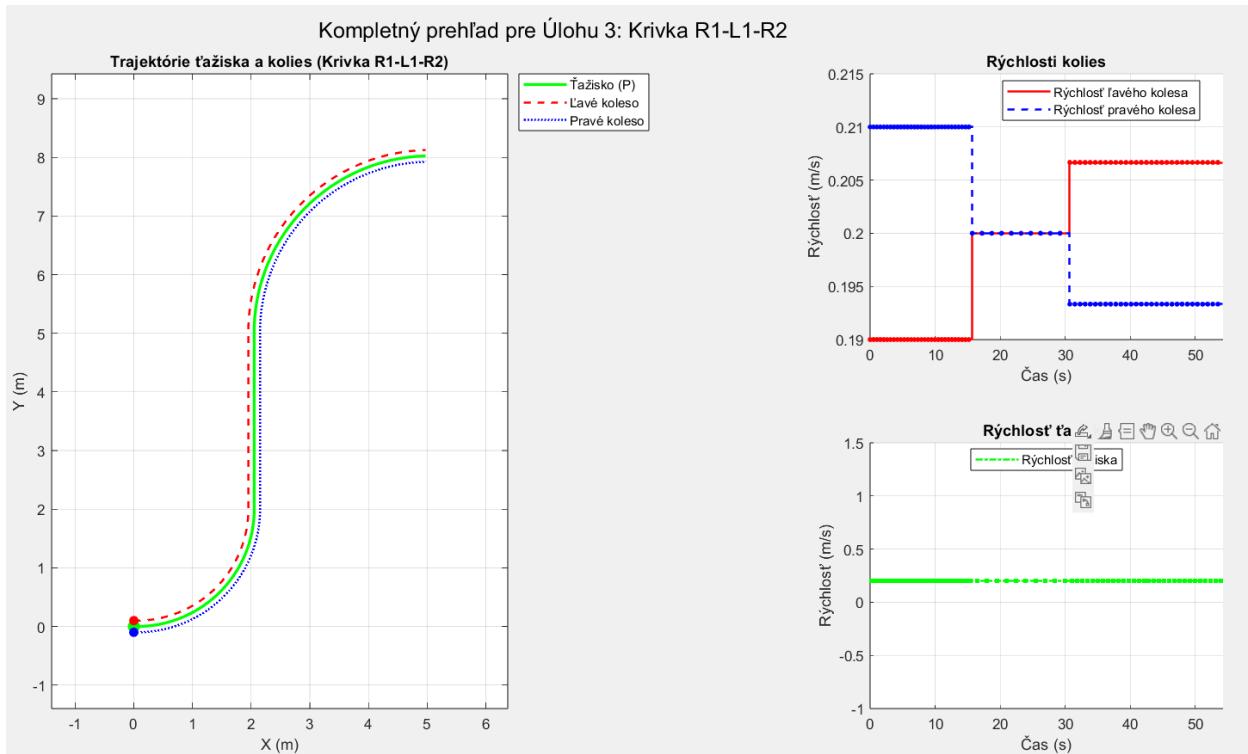
Ako prve som si od používateľa načítał hodnoty pre R1, L1 a R2. Pre pohyb po oblúkoch som najprv určil požadovanú uhlovú rýchlosť ťažiska na základe tangenciálnej rýchlosťi v_T a polomeru oblúka (R). Pre prvý oblúk bola ω_T kladná, pre druhý oblúk záporná. Následne som pomocou vzorcov inverznej kinematiky (vzorce (12) a (13)) vypočítal potrebné rýchlosťi v_L a v_R .

```
--- Úloha 3: Krivka R1-L1-R2 ---
zadajte polomer prvého oblúka R1 [m]: 2
zadajte dĺžku rovného úseku L1 [m]: 3
zadajte polomer druhého oblúka R2 [m]: 3
Úloha 3 dokončená.
```

Obr.10 Command window z úlohy 3.

Po vygenerovaní kompletných vektorov časov a rýchlosťí kolies som inicializoval polia pre ukladanie histórie polohy (x, y, Φ) ťažiska a polôh kolies, s počiatocnou pozíciou robota v bode (0,0) a uhlom 0. Samotná simulácia pohybu prebiehala v cykle, kde som pre každý časový interval dt vypočítal aktuálnu lineárnu rýchlosť ťažiska v_T (vzorec (1)) a uhlovú rýchlosť ω_1 (vzorec (2)). Následne som pomocou Eulerovej integrácie aktualizoval polohu (x, y) a orientáciu Φ ťažiska (vzorce (6)až(11)). Z novej polohy a orientácie ťažiska som vypočítal a uložil aj aktuálne pozície ľavého a pravého kolesa.

Nakoniec som pomocou funkcie figure a subplot vykreslil požadované grafy: trajektórie ťažiska a kolies a časové priebehy rýchlosťí ľavého kolesa, pravého kolesa a ťažiska.



Obr. 10

Rozbor 4. úlohy:

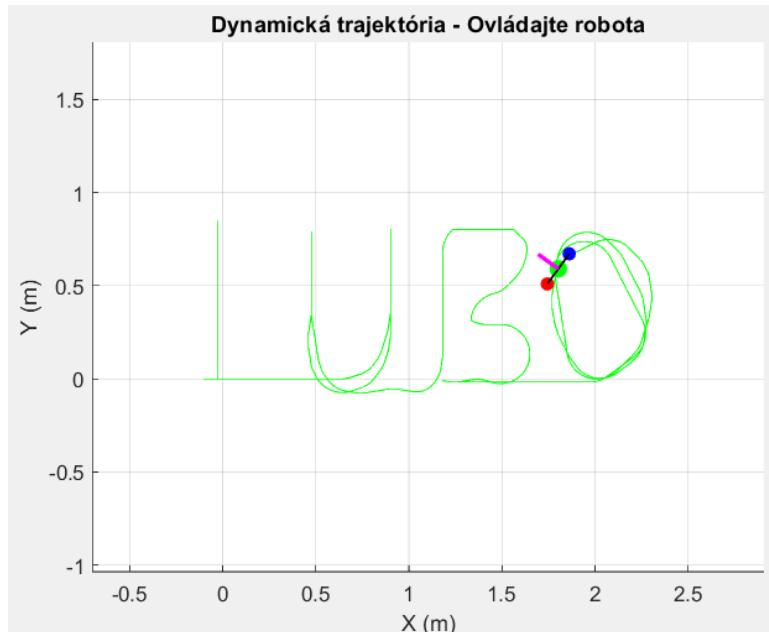
Cieľom tejto úlohy bolo vytvoriť interaktívnu hru, kde sa robot ovláda pomocou klávesnice (šípky alebo W,A,S,D). Po skončení hry sa mali vykresliť grafy jeho pohybu – trajektórie tāžiska a kolies, a tiež priebehy rýchlosťí kolies a tāžiska v čase.

Pre hru som vytvoril grafické okno. Toto okno muselo vedieť reagovať na stlačenie a uvoľnenie kláves. Na to slúžili špeciálne funkcie (KeyPressFcn, KeyReleaseFcn), ktoré si pamätali, ktoré ovládacie klávesy sú práve stlačené. Hra bežala v slučke, ktorá sa opakovala, kým používateľ nestlačil ESC alebo nezavrel okno. V každom opakování tejto hlavnej slučky sa dialo nasledovné:

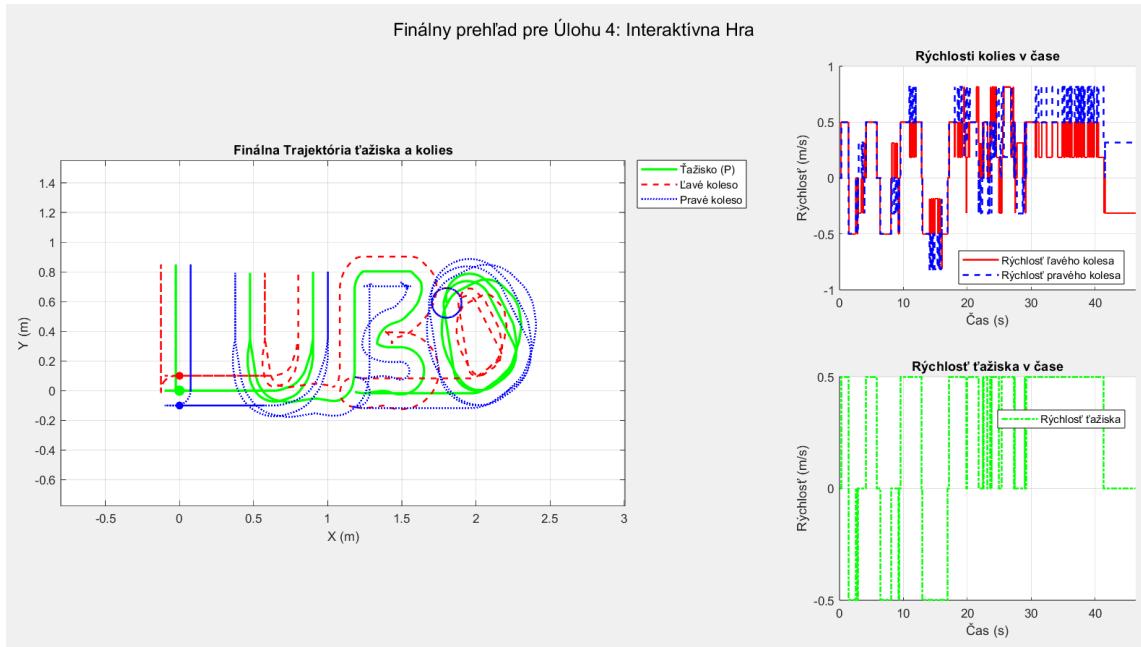
1. Zistenie ovládania: Program skontroloval, ktoré klávesy sú stlačené, a podľa toho určil, či má robot ísť dopredu, dozadu, otáčať sa doľava alebo doprava, alebo kombinovať tieto pohyby.
2. Výpočet rýchlosťí kolies: Na základe požadovaného pohybu sa vypočítali cieľové rýchlosťi pre ľavé (v_L) a pravé (v_R) koleso. Napríklad pre pohyb dopredu boli (v_L) a (v_R) rovnaké a kladné, pre otáčanie doľava bola (v_L) záporná a (v_R) kladná.

3. Prepočet pohybu robota: Z aktuálnych rýchlosí kolies (v_L) a (v_R) sa vypočítala celková rýchlosť ťažiska robota (v_T) a rýchlosť jeho otáčania (ω_T) pomocou známych vzorcov (ako vzorec (1) a (2)).
4. Aktualizácia polohy: Pomocou v_T , ω_T a malého časového kroku dt_{sim} sa vypočítala nová poloha (x, y) a nová orientácia (ϕ) ťažiska robota. V podstate sa predpokladalo, že počas krátkeho dt_{sim} sa robot pohybuje konštantnou rýchlosťou a otáčaním.
5. Výpočet polôh kolies: Z novej polohy a orientácie ťažiska sa určili aj nové pozície ľavého a pravého kolesa.
6. Ukladanie histórie: Všetky dôležité údaje – čas, polohy ťažiska a kolies, orientácia, rýchlosť kolies a ťažiska – sa ukladali do pripravených polí.
7. Dynamické zobrazenie: Obsah grafického okna sa prekreslil. Zobrazila sa celá doterajšia dráha ťažiska a aktuálna pozícia robota (napríklad ako malý krúžok s čiarou ukazujúcou smer).

Po ukončení hry (ked' slučka skončila) sa z uložených historických dát vykreslili finálne grafy. Jeden graf ukazoval kompletné trajektóriu ťažiska a kolies (s použitím axis equal pre správne proporzcie) a ďalšie grafy zobrazovali časové priebehy rýchlosí ľavého kolesa, pravého kolesa a ťažiska.



Obr. 12 výsledok 4. úlohy



Obr. 13 dráha a rýchlosť

Zhodnotenie:

V tomto zadaní sme sa venovali simulácií a vizualizácii pohybu mobilného kolesového robota s diferenciálnym podvozkom. Cieľom bolo prakticky si precvičiť implementáciu odvodených kinematických rovnic a pochopiť, ako rýchlosť jednotlivých kolies ovplyvňujú celkový pohyb a orientáciu robota.

Základom všetkých úloh bola implementácia priamej kinematiky, ktorá z rýchlosťí ľavého a pravého kolesa určuje lineárnu rýchlosť ťažiska a uhlovú rýchlosť robota. Následne sa pomocou jednoduchého numerického prístupu (podobného Eulerovej metóde) v každom časovom kroku aktualizovala poloha a orientácia ťažiska robota. Z týchto údajov sa potom odvodzovali aj pozície jednotlivých kolies pre ich vykreslenie.

Prvá úloha slúžila na overenie základnej funkčnosti simulácie priamo zadanými vektormi časov a rýchlosťí kolies. V druhej úlohe sme generovali pohyb robota tak, aby jeho ťažisko opísalo štvorec zadanej dĺžky strany, čo si vyžadovalo naplánovať sekvencie priamočiareho pohybu a otáčania na mieste. Tretia úloha bola podobná, no robot mal sledovať zložitejšiu krviku pozostávajúcu z dvoch kruhových oblúkov a rovného úseku, pričom polomery a dĺžku definoval používateľ. Tu bolo potrebné využiť aj princípy inverznej kinematiky na určenie rýchlosťí kolies pre pohyb po oblúku. Posledná, štvrtá úloha, bola najinteraktívnejšia – vytvorili sme jednoduchú hru, kde sa robot ovládal pomocou klávesnice. Program v reálnom čase spracúval stlačenia kláves, prepočítaval pohyb a dynamicky zobrazoval trajektóriu.

Pri všetkých úlohách bol dôležitý výstup vo forme grafov – jednak časové priebehy rýchlosťí ľavého, pravého kolesa a ťažiska, a jednak vykreslené trajektórie všetkých troch bodov. Pre správne zobrazenie

trajektórií bolo kľúčové použitie príkazu axis equal. Pri interaktívnej úlohe som sa stretol s výzvou správneho spracovania vstupu z klávesnice a plynulého dynamického prekreslovania grafu.

Toto zadanie mi umožnilo lepšie pochopiť základné princípy kinematiky diferenciálneho podvozku, previesť teoretické rovnice do funkčného kódu a vizualizovať výsledky simulácií. Bola to dobrá praktická skúsenosť s modelovaním a simuláciou pohybu mobilných robotov.

Návod na kompliaciu a spustenie vášho kódu

Človek si musí prve stiahnuť môj MatLab script a mat otvorenú zložku v MatLabe kde je tento script uložený. Potom stačí stlačiť tatíčko “Run all sections alebo stlačiť klávesu F5” potom sa úlohy spustia samé ak nie treba čítať inštrukcie ktoré sú v Command window

Zoznam použitých literatúry

Poznámky z cvičení

Čestne prehlásenie:

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