

Dynamický model dvojramenného manipulátora

Robotika, zadanie č. 2

Autor: **Matúš Nebus**
Dátum vypracovania: **25.4.2025**
Ročník štúdia: **2. Bc.**
Študijný program: **Robotika a
kybernetika**

Zadanie:

Vytvorte dynamický model dvojramenného manipulátora a navrhните riadenie polohovania robota. Simuláciu si vytvoríte pomocou diferenciálnych rovníc odvodených na cvičení. Na tomto type zadania by ste si mali precvičiť implementáciu Lagrangeových rovníc a zafixovať tak preberané učivo.

Parametre manipulátora:

$$l_1 = 0.25 \text{ [m]}, l_2 = 0.25 \text{ [m]};$$

$$m_1 = 3 \text{ [kg]}, m_2 = 3 \text{ [kg]};$$

$$B_1 = 2 \text{ [kg.m}^{-1}\text{]}, B_2 = 2 \text{ [kg.m}^{-1}\text{]};$$

$$g = 9.81 \text{ [m.s}^{-2}\text{]}$$

V rámci riešenia zadania sa zamerajte na nasledovné úlohy:

1. Vytvorte simulačný model dvojramenného robota (detailne popíšte jednotlivé časti schémy, ako vznikli jednotlivé bloky, čo predstavujú).
2. Overte funkčnosť modelu (vykresliť priebehy: uhlov, rýchlostí)
3. Nájdite na internete konkrétny typ motora+prevodovky, ktorý by bol vhodný pre manipulátor. Vypíšte z katalógu jeho dôležité parametre.
4. Navrhните polohové riadenie pre obe osi manipulátora. Popíšte riadiacu štruktúru. Uveďte, ako ste hľadali vhodné parametre riadiacej štruktúry.
5. Overte navrhnuté riešenie pre ľubovoľné polohy (uhly) ramena - vyskúšajte viac zmien (malé, veľké zmeny).
6. Experimentujte s obmedzením akčného zásahu regulátora (pridajte blok saturácie, ktorý zohľadní reálne obmedzenie momentov motor, prípadne aj rýchlostí).
7. Vykreslite grafy (uhly, momenty). Grafy by mali byť dobre čitateľné a vhodné na tlač.

Vypracovanie:

Pri vypracovaní zadania som vychádzal zo siedmeho, ôsmeho a deviateho cvičenia, kde sme pomocou Lagrangeových rovníc druhého druhu pre ideálny dvojčlánok manipulátora odvodili dve diferenciálne rovnice:

$$j_{11}\ddot{q}_1 + j_{12}\ddot{q}_2 = \tau_1 - B_1\dot{q}_1 - c_1 - g_1 \quad (1)$$

$$j_{21}\ddot{q}_1 + j_{22}\ddot{q}_2 = \tau_2 - B_2\dot{q}_2 - c_2 - g_2 \quad (2)$$

Kde

$$j_{11} = m_1\ell_1^2 + m_2\ell_1^2 + 2m_2\ell_1\ell_2 \cos(q_2) + m_2\ell_2^2 \quad (3)$$

$$j_{12} = j_{21} = m_2\ell_1\ell_2 \cos(q_2) + m_2\ell_2^2 \quad (4)$$

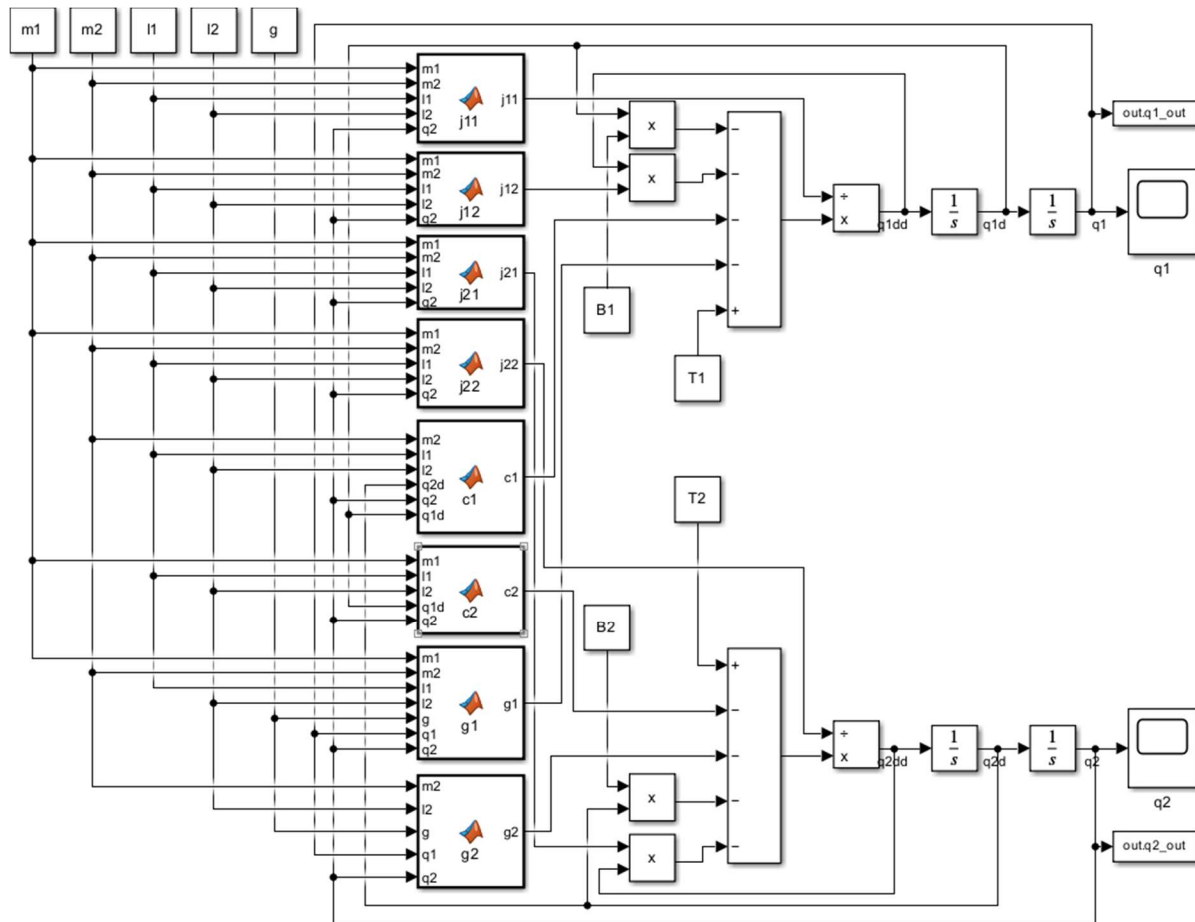
$$j_{22} = m_2\ell_2^2 \quad (5)$$

Rovnice som upravil – osamostatnil v prvej rovnici \ddot{q}_1 , v druhej \ddot{q}_2 .

$$\ddot{q}_1 = \frac{\tau_1 - B_1\dot{q}_1 - c_1 - g_1 - j_{12}\ddot{q}_2}{j_{11}} \quad (6)$$

$$\ddot{q}_2 = \frac{\tau_2 - B_2\dot{q}_2 - c_2 - g_2 - j_{22}\ddot{q}_2}{j_{21}} \quad (7)$$

Z takto upravených rovníc som vytvoril simulačný model dvojramenného robota – blokovú schému v prostredí MATLAB Simulink (obr. 1).



Obrázok 1 - simulačný model dvojramenného manipulátora

V ľavej hornej časti schémy sú definované konštanty – hmotnosti článkov m_1, m_2 , dĺžky ramien l_1, l_2 a gravitačné zrýchlenie g . Vstupujú do blokov „MATLAB Function“, ktorých výstupmi sú $j_{11}, j_{12}, j_{21}, j_{22}, c_1, c_2, g_1, g_2$.

- $j_{11}, j_{12}, j_{21}, j_{22}$ - popisujú vplyv zotrvačnosti na pohyb článkov manipulátora,
- c_1, c_2 – Coriolisove a odstredivé členy (vznikajú pri zmene smeru pohybu článkov),
- g_1, g_2 – gravitačné členy (vznikajú v dôsledku pôsobenia gravitácie).

Na obrázku 2 je ukážka kódu v jednom z blokov „MATLAB Function“ – j_{11} .

```

1  function j11 = j11(m1,m2,l1,l2,q2)
2  j11 = m1*(l1^2) + m2*(l1^2) + 2*m2*l1*l2*cos(q2) + m2*(l2^2);
3  end

```

Obrázok 2

Výstupmi z týchto blokov, podľa rovníc (5) a (6), sú násobené \ddot{q}_1 a \ddot{q}_2 a delené výstupy z dvoch veľkých „Add“ blokov. Po dva integrátory v každej z dvoch vetiev vytvárajú z uhlového zrýchlenia \ddot{q} uhlovú rýchlosť \dot{q} a uhol q .

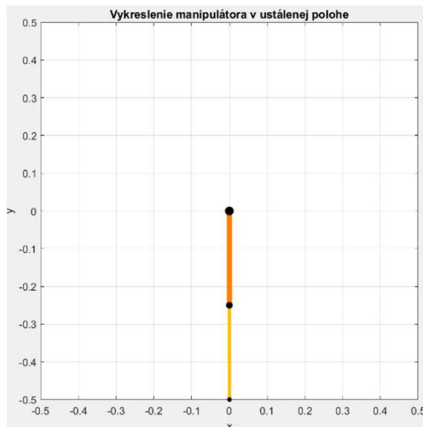
Overenie funkčnosti systému:

Spravil som kód, ktorý vykresľuje manipulátor, na základe hodnôt ustálených uhlov q_1 a q_2 po simulácii (obr. 3). Premenné q_1 a q_2 beriem z bloku „To Workspace“ v schéme na obr. 1.

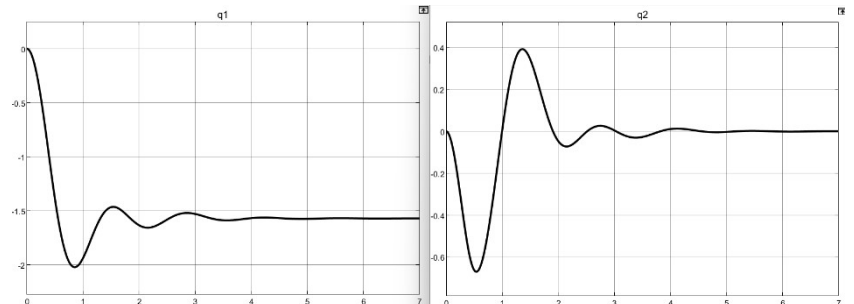
```
51 function vykresliManipulator(fig, q1, q2, l1, l2)
52     figure(fig);
53     clf(fig);
54     x0 = 0; y0 = 0;
55     x1 = l1 * cos(q1); y1 = l1 * sin(q1);
56     x2 = x1 + l2 * cos(q1 + q2); y2 = y1 + l2 * sin(q1 + q2);
57     plot([x0, x1], [y0, y1], 'Color', [1, 0.5, 0], 'LineWidth', 5); hold on;
58     plot([x1, x2], [y1, y2], 'Color', [1, 0.75, 0], 'LineWidth', 3);
59     plot(x0, y0, 'ko', 'MarkerSize', 8, 'MarkerFaceColor', 'k');
60     plot(x1, y1, 'ko', 'MarkerSize', 6, 'MarkerFaceColor', 'k');
61     plot(x2, y2, 'ko', 'MarkerSize', 4, 'MarkerFaceColor', 'k');
62     axis equal;
63     grid on;
64     xlim([-l1-l2, l1+l2]);
65     ylim([-l1-l2, l1+l2]);
66     title('Vykreslenie manipulatora v ustálenej polohe');
67     xlabel('x'); ylabel('y');
68     hold off;
69 end
```

Obrázok 3

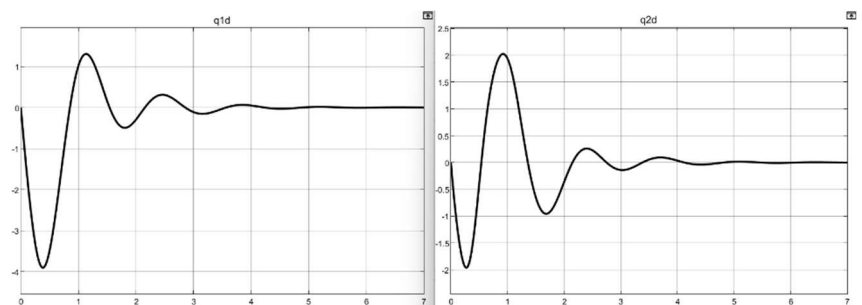
Vďaka tomuto kódu som si mohol jednoducho overiť funkčnosť systému – pri nastavení konštant T_1 a T_2 (ako keby momenty motorov) na 0, sa manipulátor v ustálenom stave vykreslil spadnutý dole, čiže $q_1 = -\frac{\pi}{2}$ a $q_2 = 0$ (Obr. 4). Na obrázku 5 sú priebehy uhlov, na obrázku 6 priebehy rýchlostí - vidieť, že sa všetko ustálilo.



Obrázok 5



Obrázok 4

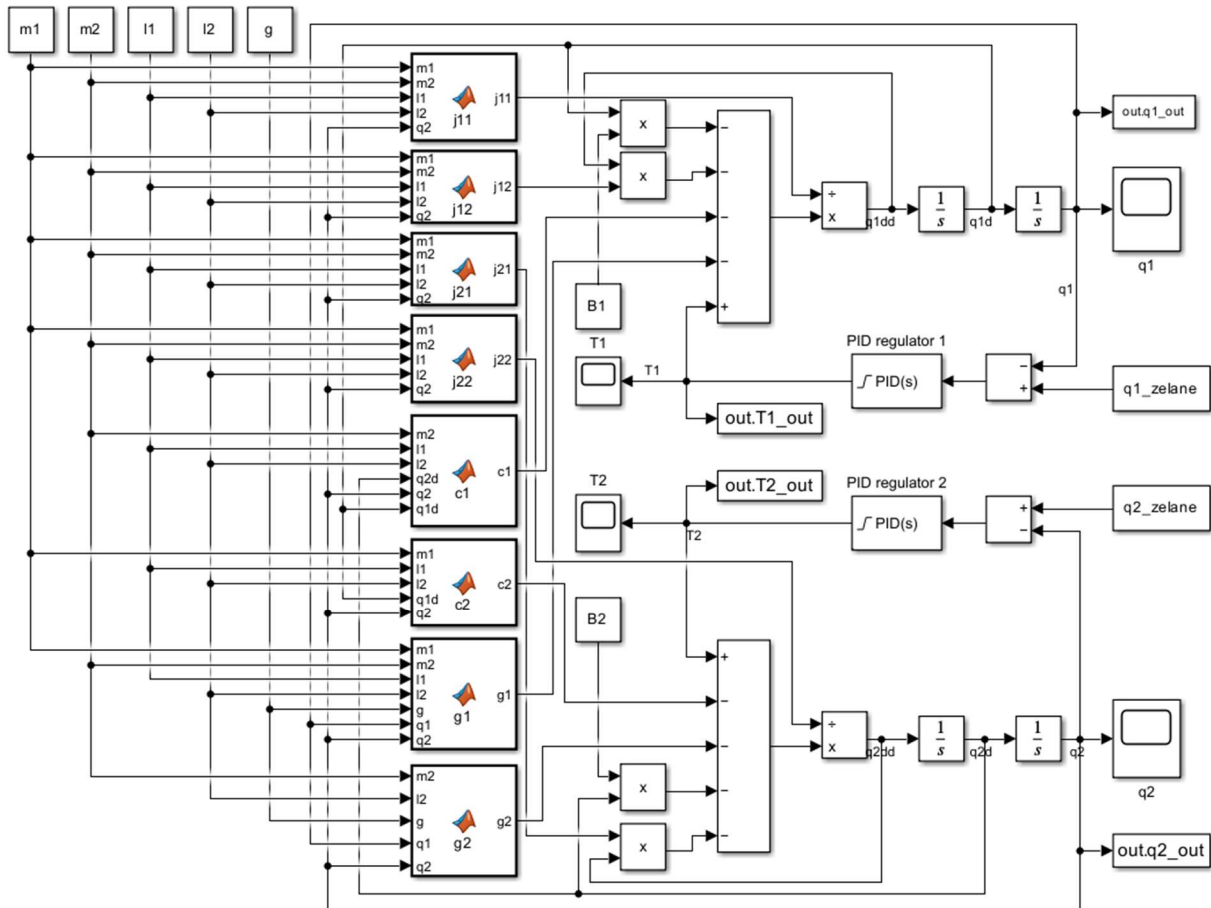


Obrázok 6

Návrh polohového riadenia:

Pre polohové riadenie som sa rozhodol použiť PID regulátor – blok „PID Controller“ v Simulinku. Výhodou sú jednoduchosť a intuitívne ladenie. Namiesto konštánt T_1 a T_2 v schéme na obr. 1 som teda zapojil výstupy z PID regulátorov (obr. 7). Ich parametre, ako aj saturácia, viem nastavovať v aplikácii, o ktorej píšem nižšie.

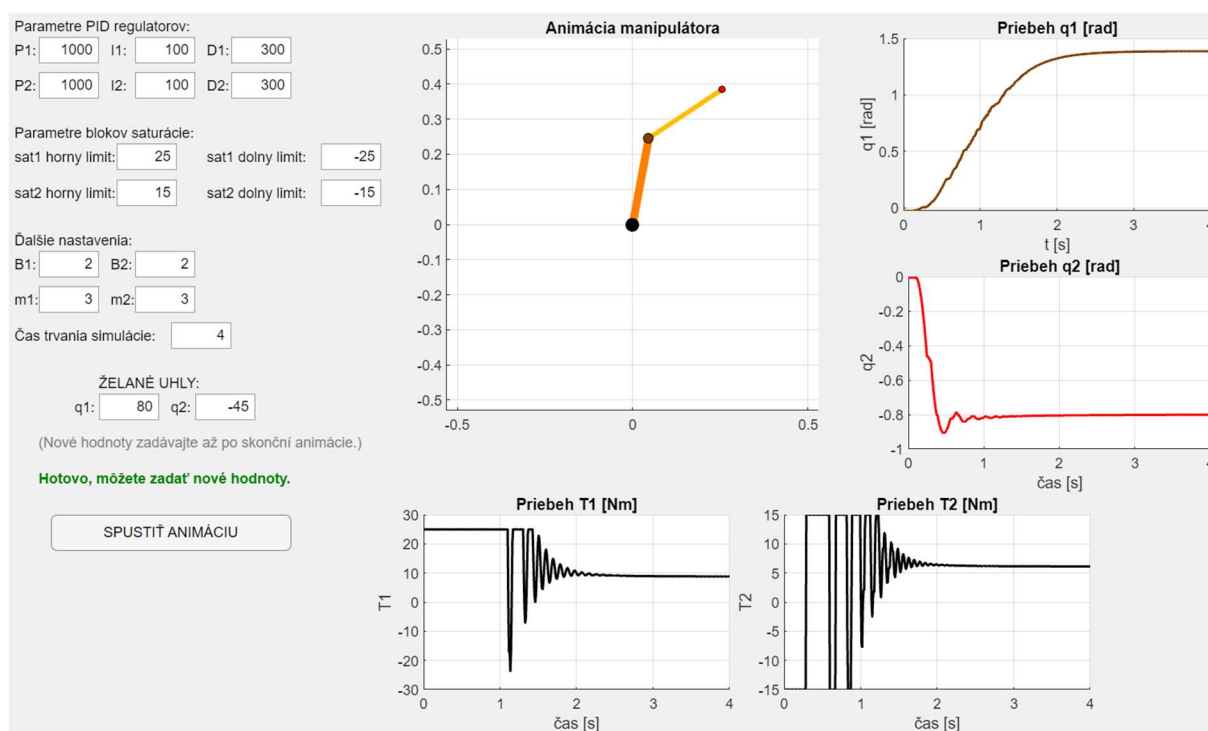
Taktiež som pridal dva bloky „To Workspace“ pre vykreslenie priebehov τ_1 a τ_2 v aplikácii.



Obrázok 7 - schéma manipulátora aj s PID regulátormi

Aplikácia na nastavovanie parametrov a animovanie:

Spravil som aplikáciu, v ktorej vie používateľ nastavovať parametre oboch regulátorov, saturácie, meniť B_1 a B_2 , m_1 a m_2 , a zobrazovať animáciu manipulátora a priebehy q_1 , q_2 , τ_1 a τ_2 . Snímka obrazovky z tejto aplikácie je na obrázku 8.



Obrázok 8

Vďaka tejto aplikácii som si mohol jednoducho skúšať rôzne nastavenia PID regulátora, s viditeľným výsledkom. Vhodné parametre regulátora som získal manuálnym skúšaním, postupným zvyšovaním alebo znižovaním jednotlivých zložiek. Začal som s proporcionálnym členom, ktorý ovplyvnil rýchlosť odozvy, postupne som ho zvyšoval. Následne som pridal integračný člen, ktorý odstraňuje trvalú odchýlku. Nakoniec som použil derivačný člen na tlmenie oscilácií. Po každej zmene som sledoval grafy, potom znovu upravoval. Nakoniec som sa dostal k hodnotám

$$P = 1000$$

$$I = 100$$

$$D = 300$$

pre oba regulátory.

Obmedzoval som aj saturáciu, ktorá sa tiež nastavuje v „PID Controller“ v Simulinku (a aj v mojej aplikácii). Postupne som zistil, že vhodné obmedzenie je horný, resp. dolný limit ± 25 Nm pre prvý regulátor PID₁ (riadi vetvu s uhlom q_1), a horný, resp. dolný limit ± 15 Nm pre druhý regulátor PID₂ (riadi vetvu s uhlom q_2). Zníženie absolútnej hodnoty obmedzenia má za následok, že manipulátor sa do žiadanej polohy nikdy nemusí dostať. Pri zvýšení sú zase potrebné zbytočne silné motory.

Výber motora a prevodovky:

Keďže som teda poznal približné hodnoty momentov, ktoré motor s prevodovkou musí mať, mohol som na internete nájsť konkrétny typ.

Pre prvý kĺb, ktorého motor by mal mať moment okolo 25-30 Nm, som našiel kombináciu BLDC motora a planetárnej prevodovky od výrobcu Maxon (obr. 9).

GEAR



Planetary Gearhead GP 52 C Ø52 mm, 4 - 30 Nm,
Ceramic Version
Part No.: 223089

Diameter: 52 mm
Reduction Ratio: 43 : 1
Torque: 30 Nm

€471.46

[Details](#)

[> Change](#)

[> Remove](#)

MOTOR



EC-i 52 Ø52 mm, brushless, 200 W, with Hall
sensors
Part No.: 606793

Diameter: 52 mm
Type performance: 200 W
Nominal voltage: 24 V
No load speed: 3340 rpm
Nominal torque: 646 mNm

€509.25

[Details](#)

[> Change](#)

Obrázok 9

Maximálny moment motora je $646 \text{ mNm} = 0,646 \text{ Nm}$. Prevodový pomer prevodovky je 43:1 – prevod do pomala. Moment motora s prevodovkou je teda $0,646 * 43 = 27,778 \text{ Nm}$, čo je pre tento účel ideálne.

Pri druhom kĺbe stačí motor s menším momentom - okolo $15\text{-}20 \text{ N*m}$. Planetárnu prevodovku som nechal rovnakú, k nej som našiel iný motor (obr. 10).

GEAR



Planetary Gearhead GP 52 C Ø52 mm, 4 - 30 Nm,
Ceramic Version
Part No.: 223089

Diameter: 52 mm
Reduction Ratio: 43 : 1
Torque: 30 Nm

€471.46

[Details](#)

[> Change](#)

[> Remove](#)

MOTOR



EC-i 52 Ø52 mm, brushless, 180 W, with Hall
sensors
Part No.: 574741

Diameter: 52 mm
Type performance: 180 W
Nominal voltage: 24 V
No load speed: 4720 rpm
Nominal torque: 438 mNm

€393.15

[Details](#)

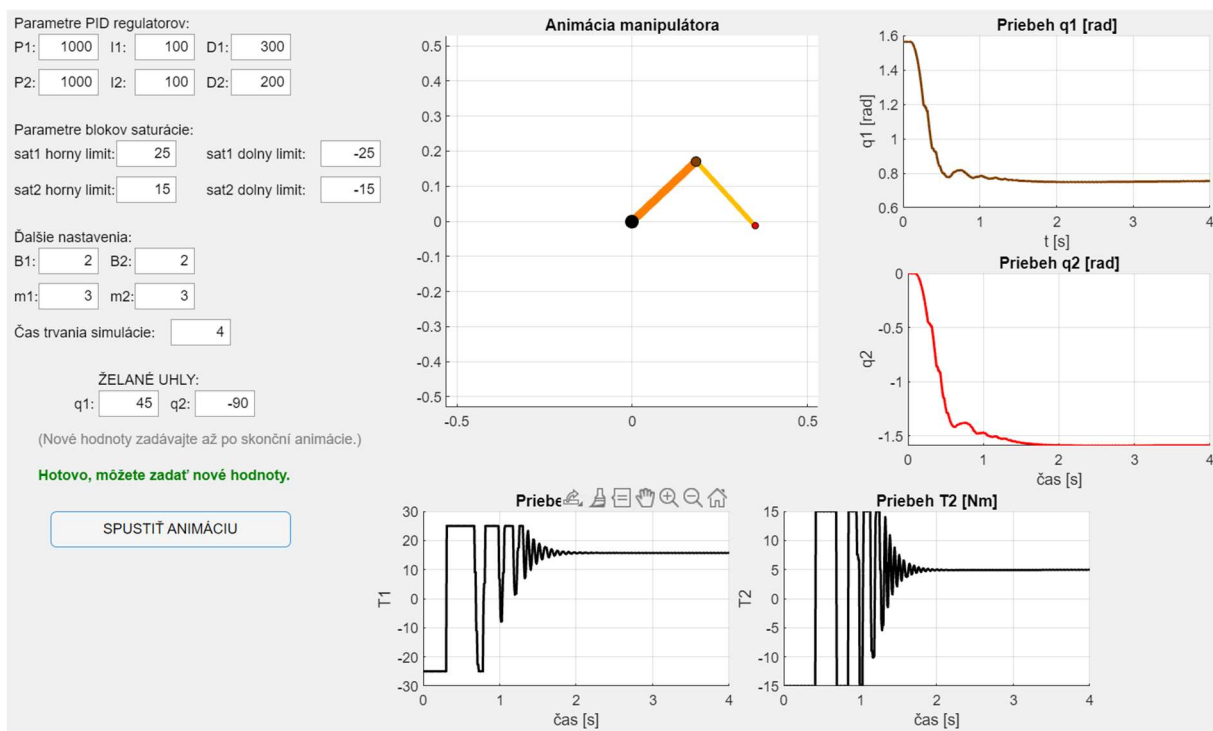
[> Change](#)

Obrázok 10

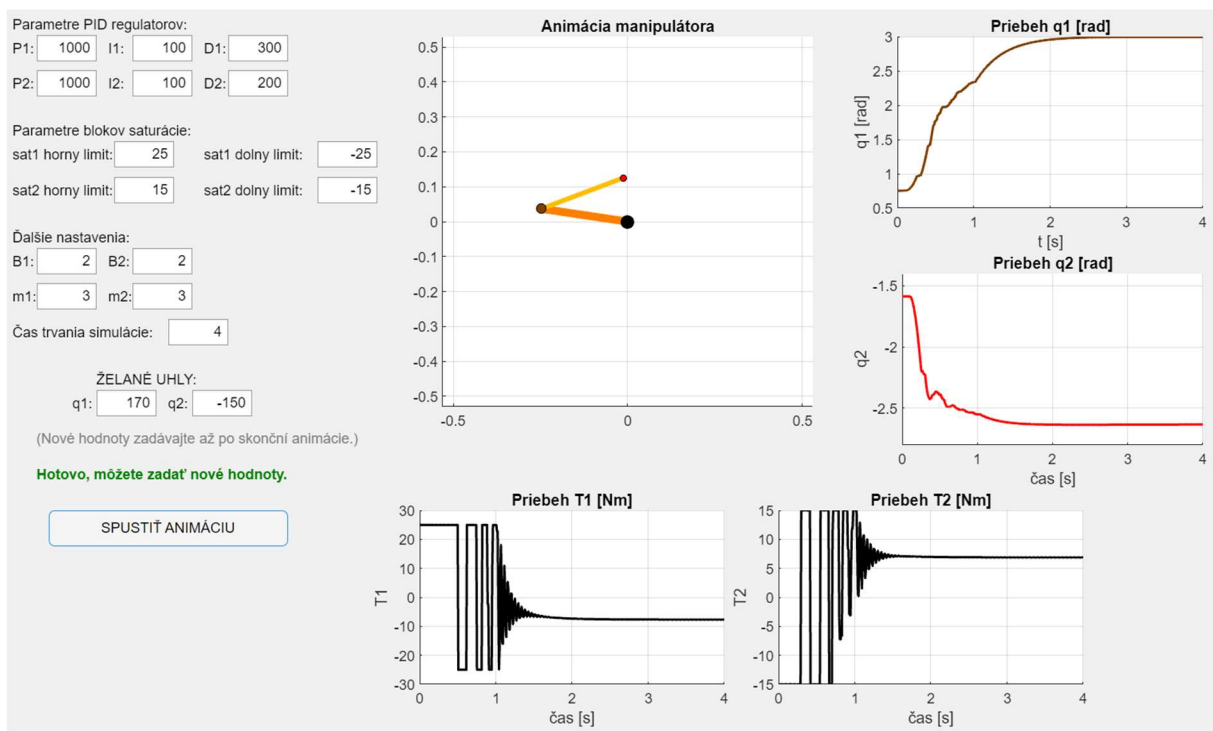
Maximálny moment motora je $438 \text{ mNm} = 0,438 \text{ Nm}$. Prevodový pomer prevodovky je 43:1 – prevod do pomala. Moment motora s prevodovkou je teda $0,438 * 43 = 18,834 \text{ Nm}$, čo je pre druhý kĺb ideálne.

Overenie navrhnutého polohového riadenia:

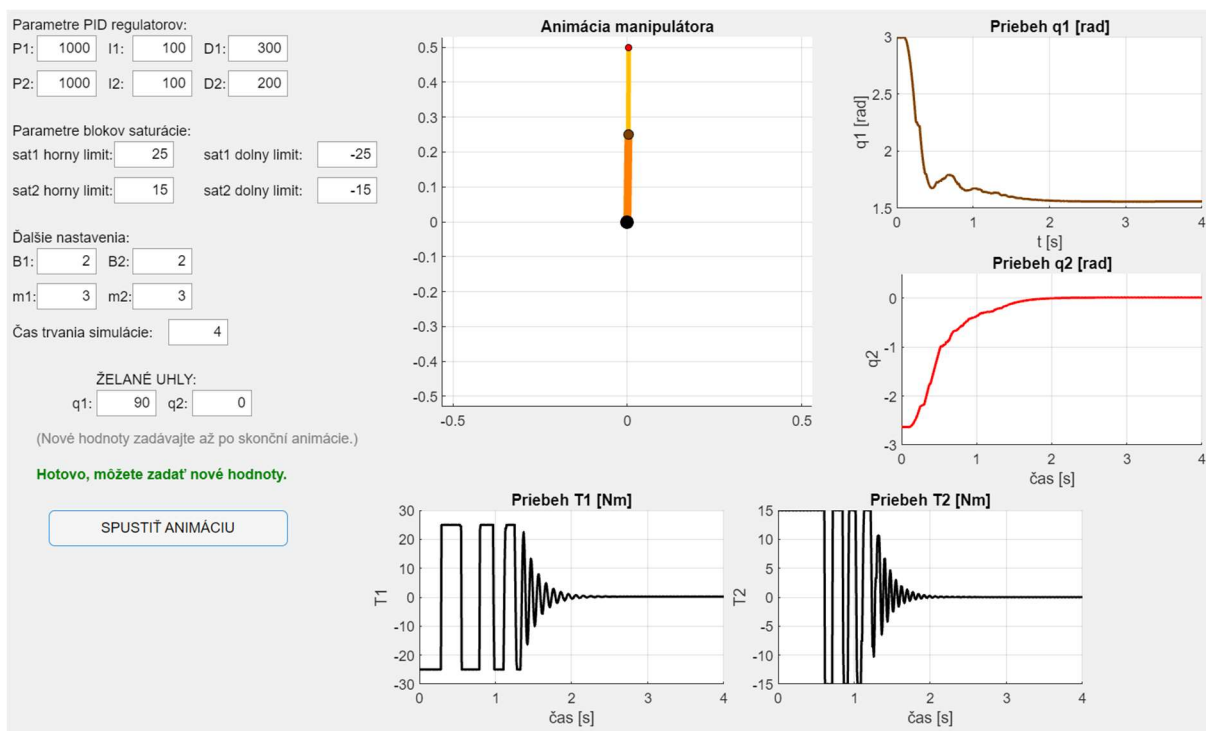
V aplikácii som skúšal rôzne polohy manipulátora, väčšie aj menšie zmeny, ako vidieť na obrázkoch nižšie.



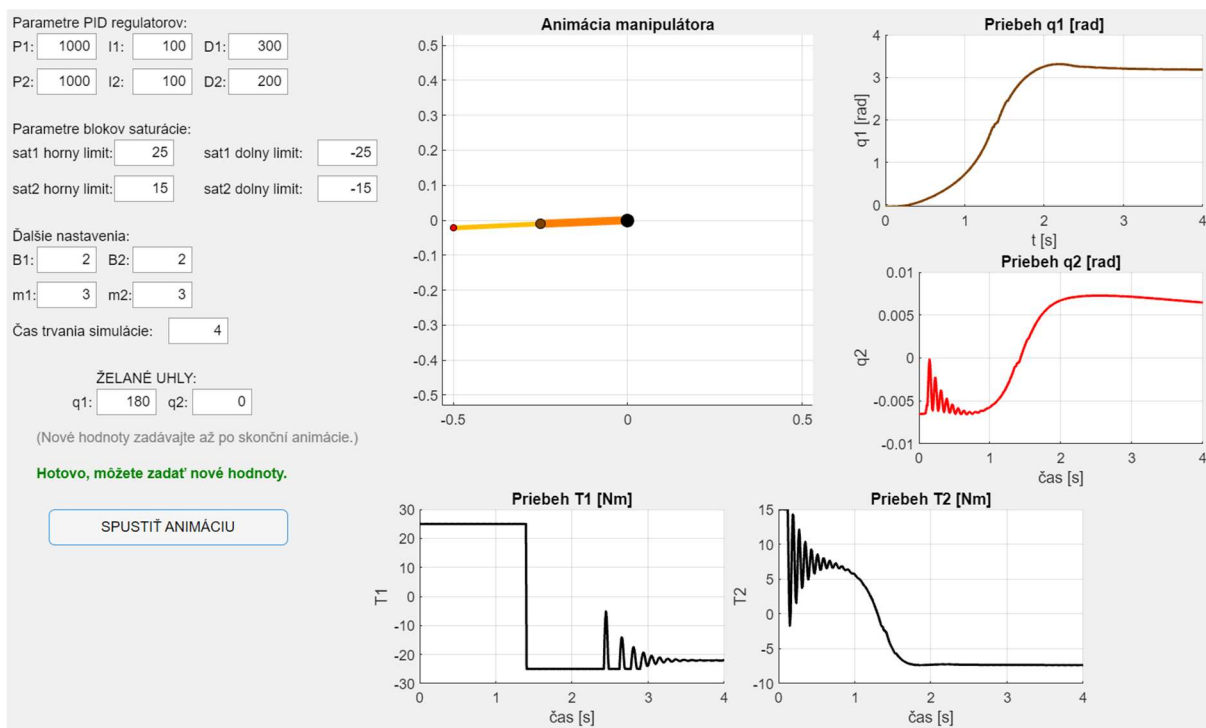
Obrázok 11 - z polohy $q_1=90^\circ$, $q_2=0^\circ$ do polohy $q_1=45^\circ$, $q_2=-90^\circ$



Obrázok 12 - z polohy $q_1=45^\circ$, $q_2=-90^\circ$ do polohy $q_1=170^\circ$, $q_2=-150^\circ$



Obrázok 13- z polohy $q_1=170^\circ$, $q_2=-150^\circ$ do polohy $q_1=90^\circ$, $q_2=0^\circ$



Obrázok 14 - z polohy $q_1=0^\circ$, $q_2=0^\circ$ do polohy $q_1=180^\circ$, $q_2=0^\circ$

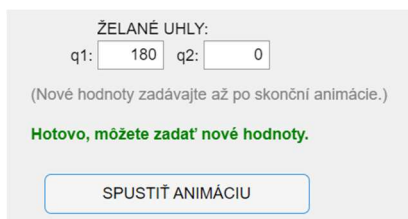
Záver:

Úlohou bolo vytvoriť dynamický model dvojramenného manipulátora a navrhnuť riadenie polohovania robota.

Z diferenciálnych rovníc, ktoré sme si odvodili na cvičení, som zostavil v prostredí MATLAB Simulink schému manipulátora a overil jej funkčnosť. Navrhol som polohové riadenie – PID regulátor. Jeho parametre som ladil ručne, postupným zvyšovaním alebo znižovaním jednotlivých parametrov, pričom som sledoval priebehy q_1 , q_2 a τ_1 , τ_2 a animáciu. Môj navrhnutý PID regulátor určite nie je dokonalý, ale svoj účel plní – manipulátor sa nakoniec dostane blízko k žiadanej polohe, aj keď s malými trvalými odchýlkami, niekedy veľkými prekmitmi a osciláciami. Experimentoval som aj s ohrozením akčného zásahu, až kým som sa nedostal k približne optimálnym hodnotám.

Simulačnú schému prikladám v súbore „zad2_sim.slx“.

Vytvoril som aplikáciu, v ktorej vie používateľ nastavovať parametre oboch regulátorov, saturácie, meniť B_1 a B_2 , m_1 a m_2 , a zobrazovať animáciu manipulátora a priebehy q_1 , q_2 , τ_1 a τ_2 . Táto aplikácia je v súbore „zad2_aplikacia.m“. Spustiť ju možno ako bežný MATLAB skript, pričom treba mať v rovnakom priečinku uložený a zároveň otvorený v Simulinku súbor „zad2_sim.slx“, s ktorým aplikácia pracuje. Pred zadaním ďalších hodnôt treba pre správnu funkciu aplikácie počkať, kým sa animácia skončí. Niekedy sa voľným okom môže zdať, že animácia už skončila, ale treba čakať, kým sa objaví zelený nápis ako na obrázku 15.



Obrázok 15

Animácia môže niekedy trvať dlhší čas (pár desiatok sekúnd, možno až minúty). Ak by naopak bola prírychla, treba odkomentovať riadok 208 v „zad2_aplikacia.m“ (obr. 16).

```
206         set(h(5), 'XData', x2, 'YData', y2);
207         drawnow;
208         % pause(0.001);
209     end
210 end
```

Obrázok 16

Prikladám aj súbor „zad2.m“. Tento skript vykresľuje manipulátor v ustálenej polohe, a robí aj animáciu, ale keďže som neskôr vytvoril aplikáciu, nie je veľmi podstatný (iba ak na rýchlejšie vykreslenie ustálenej polohy).

Toto zadanie som vypracoval vďaka poznámkam z cvičení a ChatGPT. Motory a prevodovku som našiel na <https://www.maxongroup.com/maxon/view/catalog>.

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