Dynamický model dvojramenného manipulátora

zadanie č. 2

Imrich Dzvoňár
RK 2. Bc

28.4. 2025

Zadanie:

Vytvorte dynamický model dvojramenného manipulátora a navrhnite riadenie polohovania robota. V 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:
*l1=0.25l*1 =0.25 [m], *l2=0.25l*2 =0.25 [m]
*m1=3m*1 =3 [kg], *m2=3m*2 =3 [kg]
*B1=2B*1 =2 [kg.m^−1]; *B2=2B*2 =2 [kg.m^−1]
g=9.81g=9.81 [m.s^ −2]

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. Navrhnite 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 rýchlosti).
7. Vykreslite grafy (uhly, momenty). Grafy by mali byť dobre čitateľné a vhodné na tlač.

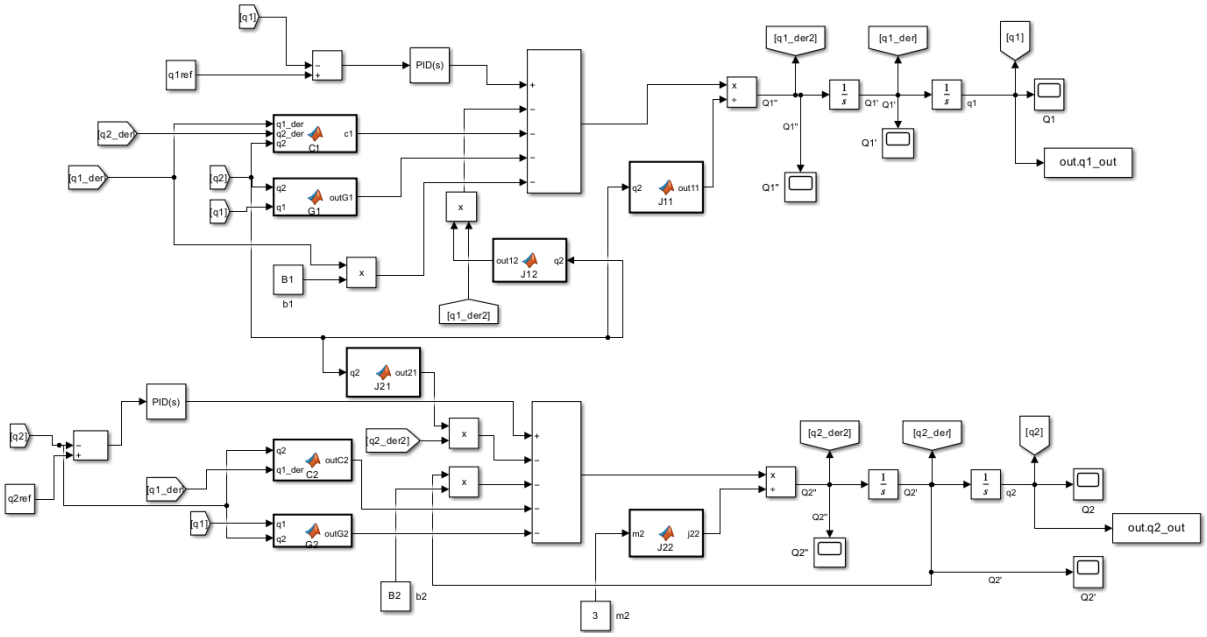
Simulačný model dvojramenného robota:

Základ vyjadrenia dynamického modelu dvojramenného manipulátora sme dostali z Lagrangeových rovníc druhého druhu v tvare (1). Ktoré sme si na cvičení odvodili do tvaru ktorý je v rovniciach (2). Prvotnú schému modelu som zostavil tak že som si vyjadril q''_1 a q''_2 z rovníc (2), po vyjadrení mi vyšli rovnice (3). Ktoré sú navzájom prepojené alebo vzájomne závisle vďaka prvkom J12 a J21.

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{q}_i}\right) - \left(\frac{\partial (E_k - E_p)}{\partial q_i}\right) = \tilde{\tau}_i - B_i\dot{q}_i \qquad (1)$$

$$\begin{aligned} j_{11}(q)\ddot{q}_1 + j_{12}(q)\ddot{q}_2 &= \tau_1 - B_1\dot{q}_1 - c_1(q,\dot{q}) - g_1(q) \\ j_{21}(q)\ddot{q}_1 + j_{22}(q)\ddot{q}_2 &= \tau_2 - B_2\dot{q}_2 - c_2(q,\dot{q}) - g_2(q) \end{aligned} \qquad (2)$$

$$\begin{aligned} \ddot{q}_1 &= \frac{\tau_1 - B_1' \cdot \dot{q}_1 - C_1 - g_1 - j_{12}\ddot{q}_2}{j_{11}(q)} \\ \ddot{q}_2 &= \frac{\tau_2 - B_2' \cdot \dot{q}_2 - C_2(q)q_2 - j_{21}(q)\ddot{q}_1}{j_{22}(q)} \end{aligned} \qquad (3)$$



Popis schémy:

V schéme aj v rovniciach (3) sú časti ako J11, C1 alebo g1, kde J11(q) a J22(q) sú hlavné členy zotrvačnosti pre každý kĺb zvlášť. J12(q) a J21(q) sú členy vzájomnej zotrvačnosti. Ukazujú, ako pohyb jedného kĺbu ovplyvňuje zotrvačnosť druhého kĺbu.

Celá schéma sa dá rozdeliť na dve hlavné časti, jedna pre každý kĺb, horná pre prvý kĺb a dolná pre druhý kĺb. Vstupy do modelu sú požadované uhly pre každý kĺb (q1ref, q2ref), a taktiež spätná väzba aktuálnych uhlov (q1, q2). Ďalej pri kontrole funkčnosti modelu som vytvoril dva bloky Scope.

V podstate model simuluje dynamické správanie robota podľa rovnice dynamiky - vzťah (4).

$$M(q)q + C(q,q)q + G(q) = \tau \qquad (4)$$

kde **M** je matica zotrvačnosti, **C** obsahuje Coriolisové a odstredivé sily, **G** je vektor gravitačných síl a **τ** je vektor kĺbových momentov.

Momenty sa potom spolu s členmi C(q,q‘)q‘ (bloky C1, C2 vynásobené príslušnými rýchlosťami) a G(q)(bloky G1, G2) sčítajú na tých sčítacích blokoch.

Výsledok sa potom delí maticou zotrvačnosti M(q). Bloky označené J11,J12,J21,J22 spolu reprezentujú túto maticu. V schéme je vidieť násobenie (blok 'x' a maticové bloky J) a sčítanie, čo realizuje to maticové násobenie a inverziu. Výstupom z tejto časti sú kĺbové zrýchlenia (q’’1, q’’2).

Zrýchlenia sa potom integrujú, čím získame kĺbové rýchlosti (q’1, q’2). Ďalšou integráciou získame kĺbové polohy, uhly q1, q2. Tieto uhly sú aj výstupom modelu a zároveň spätnou väzbou pre PID regulátor a výpočet dynamických členov.

Bloky ako [q1], [q1_der], [q1_der2] a podobné sú len výstupné porty, ktoré ukazujú hodnoty týchto veličín v schéme, slúžia na vizualizáciu a pripojenie k vzdialeným blokom v schéme. Bloky B1, B2 reprezentujú viskózne trenie sú to v podstate iba konštanty.

Ako som vytvoril bloky J,C,G:

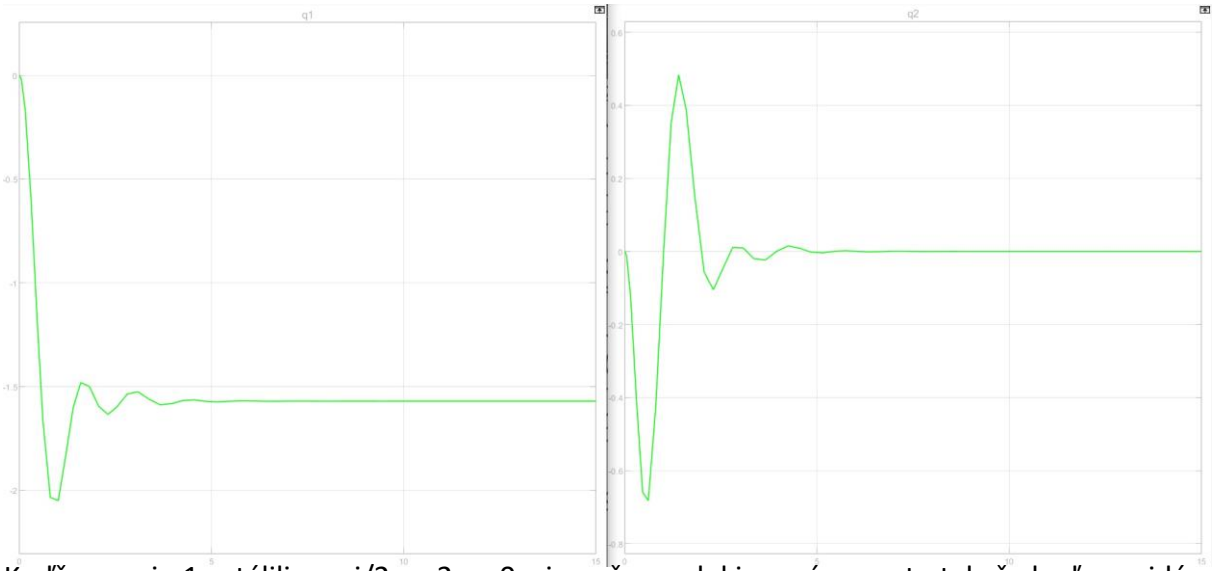
Na vytvorenie napr. bloku J12 som použil blok MatLab Function. Blok MatLab Function má pre nás takú výhodu, že si vieme vytvoriť x vstupov a mať jeden výstup (môže mať aj viac pre naše zadanie to nemá význam) čo sa deje so signálom medzi vstupom a výstupom je na nás. Meniť vstupný signál je jednoduché pretože ho vieme upravovať štandardným Matlab kódom.

```
function out12 = J12(q2)
m2 = 3; %kg

l1 = 0.25;
l2 = 0.25; %m

j12 = m2*l1*l2*cos(q2) + m2*l2^2;

out12 = j12;
```

Vo vnútri tohto bloku som si zadefinoval aj premenné, ktorých hodnoty sa nemenia. Má to jednu výhodu, a to, že v modeli je menej čiar (ak predpokladám, že druhá možnosť je pripájať všetky tieto premenné cez blok constant). Nevýhoda: musím spustiť môj script, aby sa mi do workspaceu uložili moje premenné.

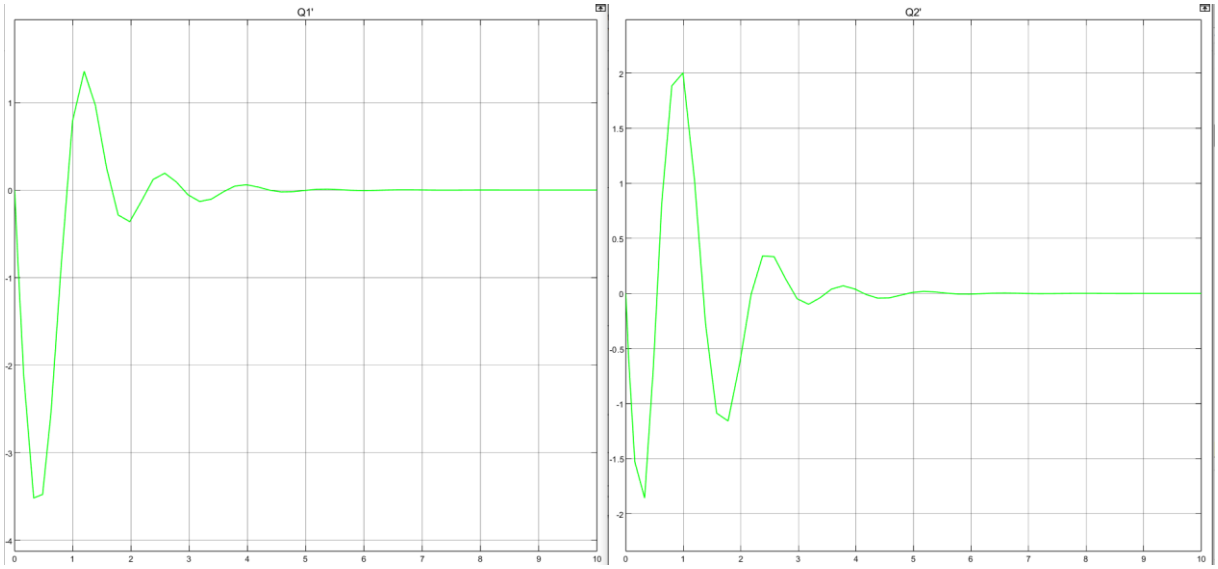
Overenie funkčnosti modelu:

Priebeh uhlov:



Keďže sa mi q1 ustálili na pi/2 a q2 na 0, viem, že model je správny, a to tak, že keď nepridám na vstup žiaden moment, môj výstup sa ustáli vo zvislej pozícii – „spadne a ostane visieť“.

Priebeh rýchlostí:



Na tomto grafe je vidieť akou rýchlosťou rameno padalo.

Výber motora a prevodovky:

Motor Harmonic Drive FHA-C-17C-100-UT som zvolil pre prvý kĺb, lebo má pre moje požiadavky dostatočný krútiacy moment aj po odčítaní hmotnosti druhého motora a ramena, ktoré musí tiež uniesť. Druhý motor Harmonic Drive FHA-C-14C-100-UT mi prišiel vhodný, pretože má podstatne menšiu hmotnosť oproti prvému motoru.

Parametre motora a integrovanej prevodovky Harmonic Drive FHA-C-17C-100-UT:

Typ: Integrovaný AC Servopohon

Prevodovka: Harmonic Drive CSD/FHA typ

Menovité napätie: 200V (AC)

Redukčný pomer: 100:1

Maximálny kontinuálny výstupný krútiaci moment: 31 Nm

Maximálna rýchlosť (na výstupe): 60 ot./min

Vôľa : Nulová

Max. krátkodobý výstupný moment: 73 Nm

Hmotnosť: 1.4 kg

Parametre motora a integrovanej prevodovky Harmonic Drive FHA-C-14C-100-UT:

Typ: Integrovaný AC Servopohon

Prevodovka: Harmonic Drive CSD/FHA typ

Menovité napätie: 200V (AC)

Redukčný pomer: 100:1

Maximálny kontinuálny výstupný krútiaci moment: 18 Nm

Maximálna rýchlosť (na výstupe): 80 ot./min

Vôľa: Nulová

Max. krátkodobý výstupný moment: 35 Nm

Hmotnosť: 0.9 kg

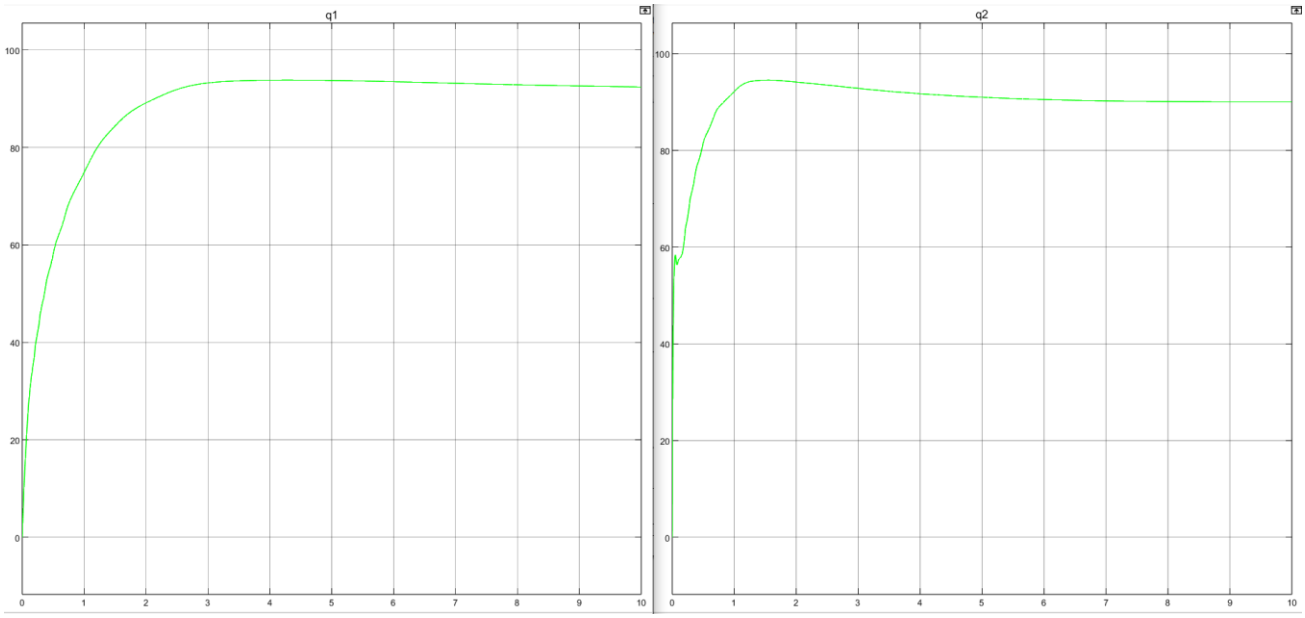
Polohové riadenie pre osi manipulátora

Pre riadenie polohy kĺbov som použil PID regulátory, jeden pre každú os. Ich úlohou je vypočítať moment motora τ, ktorý spôsobí, že sa kĺb pohne na požadované miesto, požadovanou rýchlosťou a s požadovanou presnosťou.
PID regulátor dostáva na vstup rozdiel medzi tým, kde má kĺb byť a kde skutočne je. Na základe tohto rozdielu a toho, ako rýchlo sa mení, vypočíta moment motora.

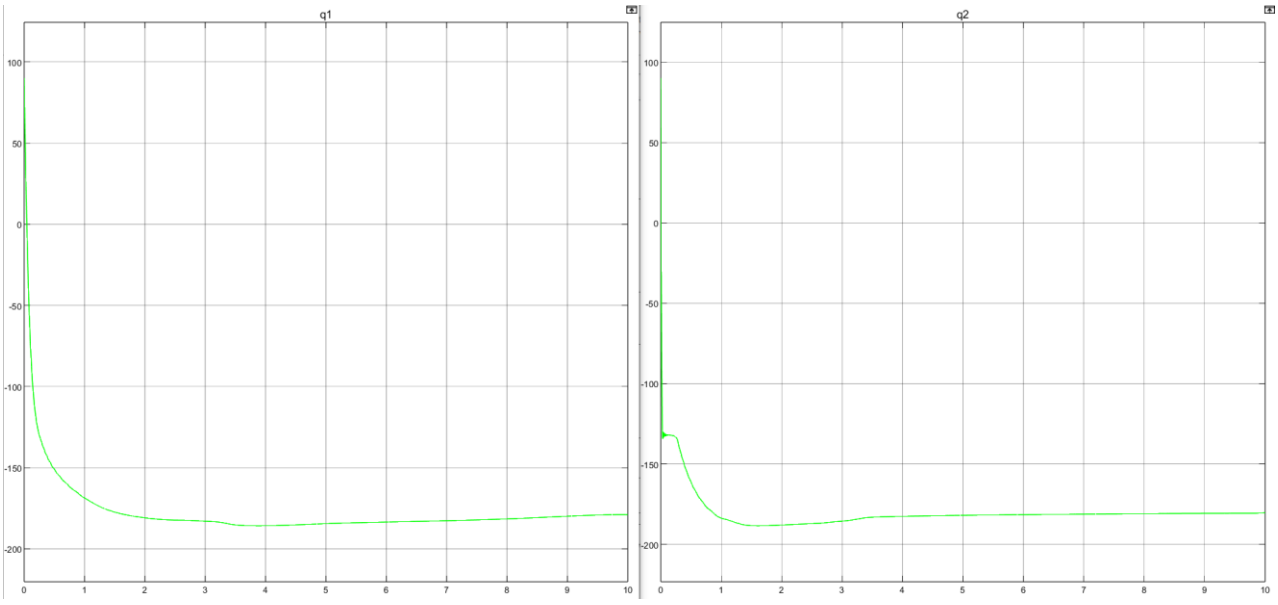
Po tom ako som pridal blok PID controller do môjho modelu som sa snažil nastaviť P proporcionálnu– I – integračnú D – derivačnú zložku regulátora. Najskôr som skúšal postupne zvyšovať P zložku kým sa mi systém nerozkmital. Ďalej som pridal D zložku aby som kmity zminimalizoval, aby som eliminoval trvalú chybu a systém sa usadil presne na požadovanej hodnote. Táto metóda bola časovo náročná a moje výsledky neboli dostatočné voči mojim očakávaniam. Skúsil som použiť funkciu bloku PID controller ktorá sa vola “Tune” ktorá mi môj regulátor pomôže naladiť. S touto funkciou som mal podstatne lepšiu reguláciu, avšak keď som pridal blok saturácie narazil som na problém. Funkcia “Tune” môj model zlinearizovala a potom našla správne hodnoty pre môj systém, to sa ale s blokom Saturácie nedá keďže je nelineárny. Je nelineárny preto že obmedzuje náš moment motora. Po tomto zistení som PID controller nastavoval zase manuálne ako som opisoval pred pridaním bloku Saturácie.

Overenie riešenia

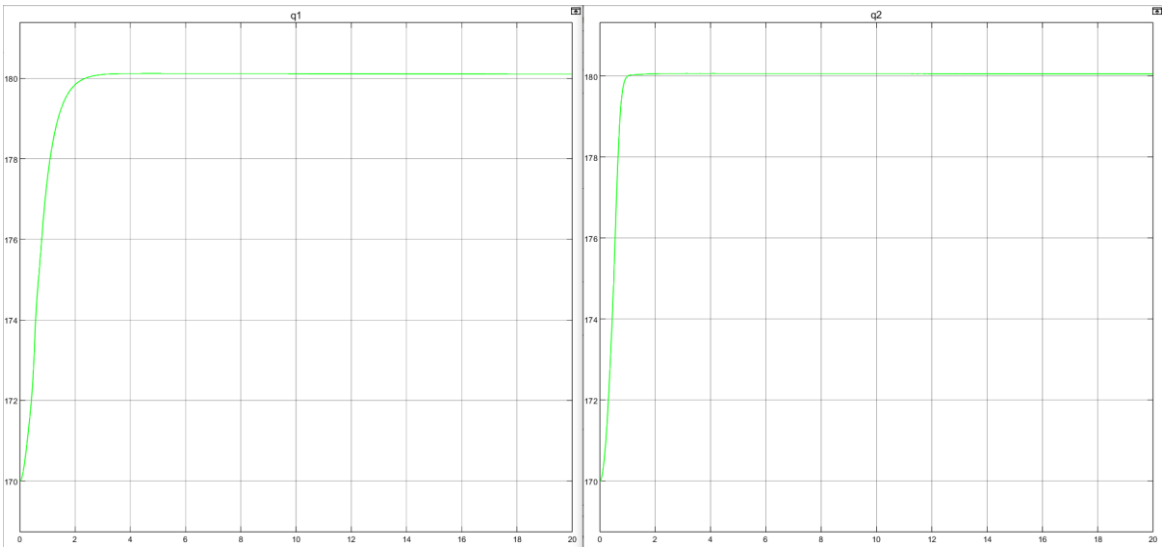
Z q1 = 0 na q1 = 90, Z q2 = 0 na q2 = 90:



Z q1 = 90 na q1 = -180, Z q2 = 90 na q2 = -180:

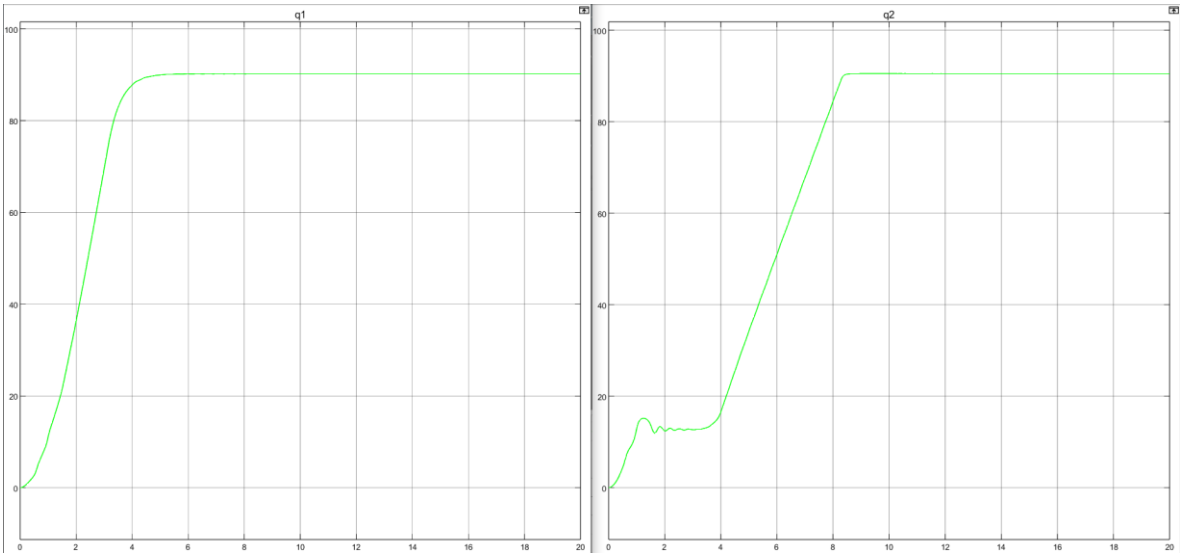


Z q1 = 170 na q1 = 180, Z q2 = 170 na q2 = 180:

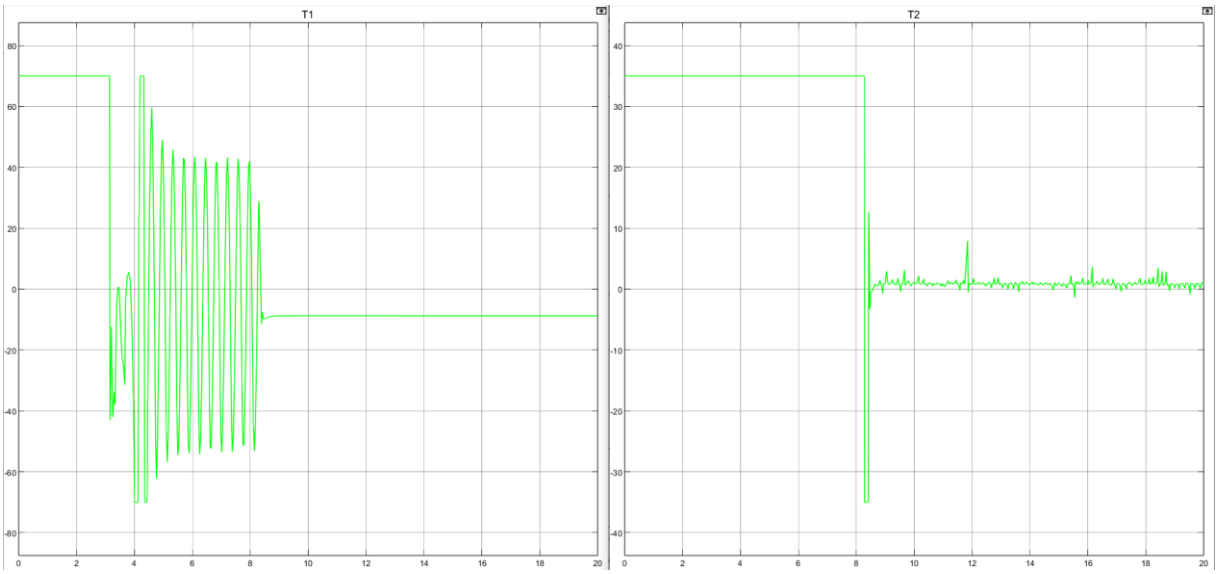


Grafy uhlov a momentov

Graf uhlov q1 a q2:



Graf momemtov τ1 a τ2:



Návod na kompiláciu a spustenie môjho kódu

Môj skript ZADANIE2_ROBOTIKA.m musí byť spustený predtým, ako sa spustí simulácia v Simulinku, a to preto, aby bloky Constant a MATLAB Function mali odkiaľ načítavať premenné ako m1, m2 atď.

ZADANIE2_ROBOTIKA.m

```
clear all; clc; close all;
m1 = 3;
m2 = 3; %kg

l1 = 0.25;
l2 = 0.25; %m

B1 = 2;
B2 = 2; %kg*m^-1

g = 9.81; %m*s^-2

q1ref = 10;
q2ref = 10;
```

Zoznam použitých literatúry

Poznámky z cvičení

https://www.youtube.com/watch?v=1A75jIfu2Pc

https://en.wikipedia.org/wiki/Proportional%E2%80%93integral%E2%80%93derivative_controller

Zhodnotenie:

V tomto zadaní sme riešili dynamický model dvojramenného robota a riadenie polohy jeho ramien. Išlo o to pochopiť, ako sa robot hýbe pod vplyvom síl a momentov, a ako ho prinútiť ísť tam, kam chceme.

Základom bol dynamický model odvodený pomocou Lagrangeových rovníc, ktorý som implementoval v Simulinku. Na vytvorenie blokov pre členy dynamiky (ako sú matica zotrvačnosti J, Coriolisove a gravitačné sily C a G) som využil napríklad aj blok MATLAB Function. Funkčnosť samotného dynamického modelu bez riadenia som si overil pozorovaním, či sa robot správne ustáli v stabilnej polohe, kde ramená voľne visia vplyvom gravitácie. Pre polohové riadenie som použil PID regulátory pre každú os. Nastavoval som ich najprv manuálne, čo si vyžadovalo správne odhadnúť hodnôt. Potom som skúsil použiť funkciu „Tune“ v Simulinku, ktorá mi rýchlejšie poskytla lepšie počiatočné výsledky. Avšak narazil som na problém, keď som pridal blok saturácie, ktorý obmedzuje maximálny moment motora. Funkcia „Tune“ linearizuje model a nevie pracovať s touto nelinearitou. Preto som sa vrátil k manuálnemu ladeniu, tentoraz už s týmto reálnym obmedzením v modeli.

Napriek výzvam s ladením PID pri obmedzenom momente sa mi nakoniec podarilo nastaviť regulátory tak, aby robot sledoval požadované polohy a usadil sa na nich. V grafe momentu druhého kĺbu bolo ale vidieť v ustálenom stave šum, čo súvisí s derivačnou zložkou a filtráciou. Tento šum sa mi podarilo jemne znížiť tým, že som zvýšil D zložku, no jav ďalej pretrváva. Pri tomto zadaní som si dobre precvičil implementáciu dynamických rovníc, pochopil som vplyv reálnych obmedzení, ako je saturácia momentu na riadenie, a naučil som sa, že ladenie regulátorov je často iteratívny proces, najmä pri nelineárnych systémoch. Bola to dobrá skúsenosť so simuláciou a riadením robotických systémov.

Čestne prehlásenie:

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