



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

## ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

## MODELOVÁNÍ A ŘÍZENÍ POHYBU HUMANOIDNÍHO ROBOTU

MODELOVÁNÍ A ŘÍZENÍ POHYBU HUMANOIDNÍHO ROBOTU

### BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

### AUTOR PRÁCE

AUTHOR

Alexandra Sokolová

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Robert Grepl, Ph.D.

BRNO 2023

# Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Studentka:	<b>Alexandra Sokolová</b>
Studijní program:	Mechatronika
Studijní obor:	bez specializace
Vedoucí práce:	<b>doc. Ing. Robert Grepl, Ph.D.</b>
Akademický rok:	2022/23

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

## Modelování a řízení pohybu humanoidního robotu

### Stručná charakteristika problematiky úkolu:

V prostředí MATLAB/SimScape Multibody je možné relativně snadno modelovat i kontakt dvou těles a zrealizovat tedy jednostrannou vazbu. Využitím evolučních algoritmů je zase možné vytvořit a optimalizovat algoritmus pohybu robotu.

Cílem práce je vytvořit/adaptovat dynamický model humanoidního robotu a dále s pomocí vybraných metod implementovat a optimalizovat jeho chůzi. Inspirací jsou např. roboty firmy Boston Dynamics.

Student získá zkušenosti jak s modelováním kinematiky a dynamiky chodicích robotů, tak s plánováním trajektorie a evolučními metodami.

### Cíle bakalářské práce:

- nastudovat problematiku modelování v SimScape + modelování chodicích robotů
- vytvořit model robotu
- vytvořit algoritmus chůze (patrně s využitím evolučních metod)

### Seznam doporučené literatury:

\* <https://www.mathworks.com/products/simscape.html>

\* <https://www.mathworks.com/videos/modeling-and-simulation-of-walking-robots-1576560207573.html>

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2022/23

V Brně, dne

L. S.

---

prof. Ing. Jindřich Petruška, CSc.  
ředitel ústavu

---

doc. Ing. Jiří Hlinka, Ph.D.  
děkan fakulty

## ABSTRAKT

S nárastom popularity mobilných robotov v našom každodennom živote a ich čoraz častejším využitím v mnohých oblastiach výskumu je nevyhnutné sa hlbšie zaoberať schopnosťou robotov pohybovať sa v rôznom prostredí. Schopnosť pohybu robotov na kolesách a pásoch je obmedzená a nedovoľuje dostatočnú voľnosť pohybu v ťažko dostupnom teréne. Špeciálnu kategóriu robotov schopných pohybu v takomto teréne predstavujú roboty pohybujúce sa pomocou končatín. Aktivity vo vývoji tohto typu robotov za posledné roky výrazne vzrástli.

V tejto bakalárskej práci sme sa venovali návrhu jednonohého skákajúceho robota s chodidlom, ktorého sme následne modelovali v SimScape Multibody. Navrhnutý robot zvláda skupinu základných pohybov. Výsledkom práce je krátka choreografia, ktorá má predstaviť schopnosť robota pohybovať sa v priestore ako je napríklad drep alebo skok na a cez prekážky.

### Kľúčové slová

robot, skákajúci robot, jednonohý robot, simulácia pohybu, riadenie, regulácia, PID regulátor

## ABSTRACT

With the rise in popularity of mobile robots in our day-to-day life and their advantages in exploration, we need to explore a broader scope of methods of movement that would allow these machines to comfortably live by our sides or access the terrain which is inaccessible to humans. We have long since realised that wheels do not allow for the necessary range of movement within environment that is harder to navigate. As a result, the development of robots with legs has seen a rapid rise in prominence in the recent years. Within this text we explored the possibilities of design of one such robot, a monopedal hopper with a foot, and then created a model in SimScape Multibody with the final result being a short choreography displaying the robot's abilities to move in space, such as: crouching and jumping on and over obstacles.

### Keywords

robot, jumping robot, monopedal robot, simulation of movement, regulation, PID controller

---

SOKOLOVÁ, Alexandra. *Modelování a řízení pohybu humanoidního robotu* [online]. Brno, 2023 [cit. 2023-05-12]. Dostupné z: <https://www.vut.cz/studenti/zav-prace/detail/153404>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Robert Grepl.

---

## ČESTNÉ PREHLÁSENIE

Prehlasujem, že som bakalársku prácu na tému *Modelování a řízení pohybu humanoidního robotu* vypracovala samostatne s využitím uvedenej literatúry a podkladov, na základe konzultácií a pod vedením vedúceho práce.

.....  
dátum

.....  
Alexandra Sokolová

---

## **POĎAKOVANIE**

Ďakujem týmto doc. Ing. Robertovi Greplovi, Ph.D., za podporu a trpezlivosť pri vypracovaní tejto bakalárskej práce.

---

## OBSAH

Úvod.....	10
1 Súčasný stav .....	11
2 Proces návrhu .....	13
2.1 Prehľad základných činností .....	13
2.2 Prístup k mechanickej konštrukcii .....	13
2.2.1 Dvojramenný model.....	14
2.2.2 Zjednodušenie konceptu .....	16
2.3 Dynamika pohybu (teoretický rozbor) .....	18
2.3.1 Statika pre stoj na mieste .....	18
2.3.2 Drep.....	19
2.3.3 Skok .....	20
2.3.4 Skok na mieste .....	20
2.3.5 Skok do diaľky.....	21
3 Simulácia robota v prostredí Matlab / Simulink.....	25
3.1 Definovanie parametrov robota v prostredí Matlab .....	25
3.2 Simulácia robota v prostredí Simulink.....	27
3.2.1 Opis blokov Simulinku .....	28
3.2.2 Zostavenie robota .....	28
3.2.3 Tvrdé obmedzenia.....	30
3.2.4 Riadenie pohybu .....	31
3.2.5 Podmienky a priebeh simulácie .....	34
3.2.6 Stoj na mieste – ovládanie robota pomocou momentu .....	35
3.2.7 Stoj na mieste – ovládanie robota pomocou uhlu natočenia.....	37
3.2.8 Drep.....	38
3.2.9 Skok na mieste .....	40
3.2.10 Skok do diaľky.....	41
3.2.11 Exhíbia .....	42
3.2.12 PID regulátor.....	43
4 Záver.....	44
5 Zoznam použitých zdrojov .....	46
6 Zoznam použitých skratiek.....	49
7 Prílohy .....	50

---

7.1	Zoznam elektronických príloh .....	50
7.2	Stoj .....	51
7.3	Drep.....	54
7.4	Skok na mieste .....	57
7.5	Skok do diaľky .....	60
7.6	Exhibícia .....	63
7.7	Porovnanie nelineárneho a lineárneho zrýchlenia pohybu.....	66

---

## ÚVOD

Hoci termín „robot“ vznikol len pred asi sto rokmi, koncept „umelého života“ bol súčasťou ľudskej existencie už od nepamäti. Originálne vymyslený Jozefom Čapkom a následne spopularizovaný jeho bratom Karolom Čapkom vďaka jeho známej divadelnej hre R.U.R. – Rossum’s Universal Robots – termín robot, z českého „robotovat“ – je len najnovším termínom opisujúcim živú bábkú. Na pôvod slova sa odvolávajú aj v Humanoid arm robot simulation on SIMULINK MATLAB [1].

Prvý robot, ktorý spadá pod modernú definíciu robota, bol vyvinutý až v 30. rokoch minulého storočia. Jednonohé roboty – monopedy – majú svoje počiatky o asi 50 rokov neskôr, a dodnes je to málo preskúmaná oblasť robotiky v porovnaní s dvojnohými (bipedami) alebo viacnohými modelmi.

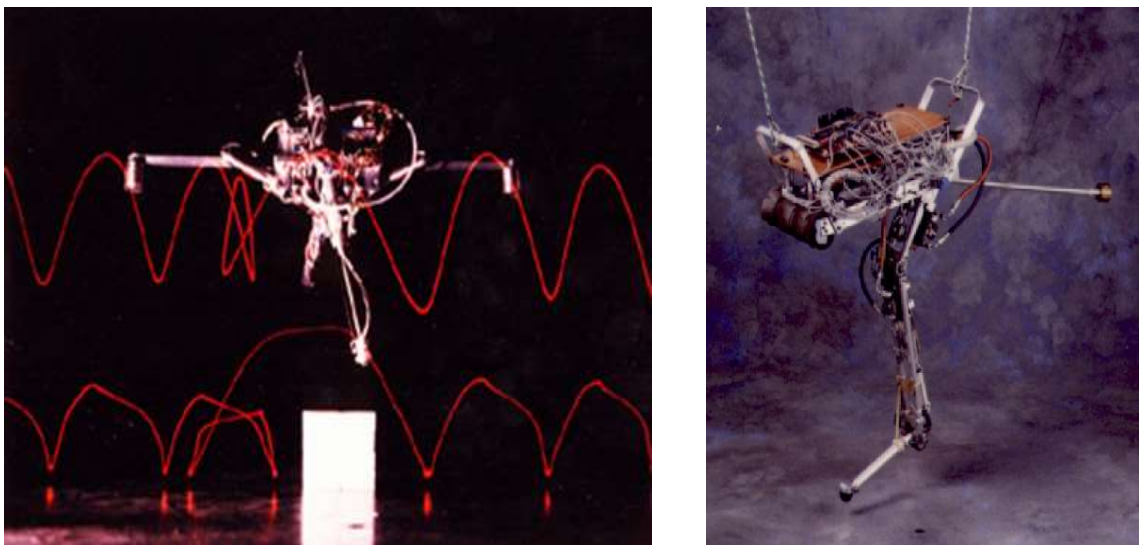
V dnešnej dobe sú rôzne typy robotov využívané vo všetkých oblastiach ľudskeho života. Prítomnosť špecifických typov robotov je nevyhnutná hlavne v prostredí, ktoré je nevhodné až nebezpečné pre ľudí (napríklad pri vesmírnom výskume). Na druhej strane sú ale čoraz bežnejšie aj roboty, ktoré robia spoločnosť ľuďom. Rôzne typy prostredia, v ktorom sa roboty musia pohybovať, sú často ťažko dostupné pre roboty, ktoré sa pohybujú na kolesách alebo pásoch. Prispôsobenie pohybu robota terénu je nutnou súčasťou jeho návrhu. Do hry teda vstupuje schopnosť robota prekážky prekonať, napr. prekročiť alebo preskočiť.

V tejto práci sa venujeme návrhu a simulácii jednoduchého jednonohého robota, a jeho schopnosti základného pohybu v priestore.

## 1 SÚČASNÝ STAV

K najstarším projektom v oblasti monopedov patria roboty vyvíjané na MIT v 80-tych rokoch [2]. Jedná sa o pneumatického robota – *Planar One-Leg Hopper* – ktorý sa dokáže pohybovať v priestore. V prípade nárazu si dokázal aj napriek jednoduchej konštrukcii udržať rovnováhu. V 90-tych rokoch univerzita predstavila hydraulického robota *Uniroo*, inšpirovaného klokanmi [3]. Jedná sa taktiež o prvého robota, ktorý nebol symetrický a ťažisko mal posunuté mimo tela. Rovnováhu udržiaval pomocou „chvosta“.

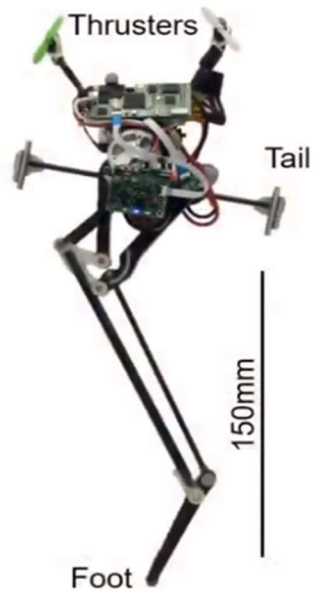
Výrazný pokrok vo vývoji môžeme sledovať od roku 2016, keď univerzita Berkeley predstavuje robota *Salto-1P* [4], ktorý adaptoval software zo staršieho robota *3D One-Leg Hopper* [5 6]. Opis robota nájdeme v článku *Hopping in legged systems — Modeling and simulation for the two-dimensional one-legged case* [7]. Na jeho vývoji sa pracuje dodnes.



Obr. 1 Planar One-Leg Hopper (1980-1982) [2] vs. Uniroo (1991-1993) [3]

V dnešnej dobe sa monopedom venuje niekoľko projektov, aj keď ich počet je stále mnohonásobne menší než u iných typov robotov. Mnohé sa venujú hlavne matematickej analýze pohybu [8, 9, 10, 11] ale vznikajú aj reálne modely [5, 8, 12, 13]. Tieto návrhy vychádzajú z princípu inverzného kyvadla. Väčšina jednonohých robotov je taktiež stavaná bez chodidla, teda ich kontakt s podložkou je minimalizovaný na bodový [2, 3, 5, 6]. Zavedením chodidla sa zaoberali autori publikácií [10, 17, 18].

Trendom vo vývoji monopedov je hlavne inšpirácia prírodou. Zatiaľ čo veľká časť vývoja pohybu robota sa inšpirovala zvieratami, primárne cicavcami žijúcimi v korunách stromov [14] viaceré projekty vychádzajú z výskumu hmyzu kvôli ich schopnosti vyskočiť mnohonásobne vyššie a ďalej v porovnaní s veľkosťou ich tela [8, 13].



Obr. 2 Salto 1P

Stručný prehľad niektorých projektov, ktoré nás inšpirovali, sme zhrnuli v Tab. 1.

Tab. 1 Stručný prehľad aktuálnych projektov

Projekt	Simulácia	HW realizácia	Analýza
<b>Ribak: Insect-inspired jumping robots [8]</b>			x
<b>EPFL jumpglider [8]</b>		x	x
<b>Salto-1P (Drift-free) [5]</b>	x	x	x
<b>Vertical Jumping for Legged Robot Based on Quadratic Programming [9]</b>	x		x
<b>Simulation of Upward Jump Control for One-Legged Robot Based on QP Optimization [10]</b>	x		x
<b>Study on One-legged Robot Jumping [12]</b>		x	x
<b>Control of one-legged robot hopping in place [20]</b>		x	x
<b>Neural network balance control of hopping robots in flight phase under unknown dynamics [19]</b>	x		x
<b>A Single-legged Robot Inspired by the Jumping Mechanism of Click Beetles and Its Hopping Dynamics Analysis [13]</b>		x	x
<b>Modeling and Control of a Single-legged Robot [11]</b>	x		x

## 2 PROCES NÁVRHU

Predtým, ako sme zvolili finálnu podobu robota, ktorému sa venujeme v tejto práci, sústredili sme pozornosť na fyzikálne a matematické princípy. Na základe ich analýzy sme dospeli k finálnej koncepcii, s ktorou sme následne pracovali. Najprv sme si museli definovať, čo od robota očakávame, a aký prístup pri jeho návrhu bude najvhodnejší. Vychádzali sme zo všeobecných požiadaviek zadania, ktoré sme postupne prispôbili našej vízii.

### 2.1 Prehľad základných činností

Skôr než sme začali uvažovať nad fyzickým výzorom robota, museli sme sa rozhodnúť, aké funkcie od neho očakávame. Hlavným cieľom robota bola mobilita so zameraním na rôzne typy pohybu v teréne. Nakoľko robot bude mať len jednu nohu, pohyb bude pozostávať zo skokov, ktorými sa bude pohybovať dopredu a dozadu aby prekonal požadovanú dráhu. Zároveň by mal zvládnuť vertikálny pohyb, napr. vyskočenie na objekt, skákanie po schodoch a podobne. Toto mu dovoľuje prekonávať prekážky v teréne.

Okrem pohybu v priestore uvažujeme aj pohyb na mieste. Základnou požiadavkou je schopnosť robota zotrvať v stabilizovanej polohe na mieste, vedieť sa skrčiť alebo vystrieť a držať túto polohu bez preváženia. V práci sme sa rozhodli riešiť nasledujúce činnosti:

- stoj na mieste,
- drep,
- skok,
  - na mieste,
  - do diaľky.

Každá činnosť vyžaduje samostatnú analýzu, ktorej sa venujeme v nasledujúcich kapitolách.

### 2.2 Prístup k mechanickej konštrukcii

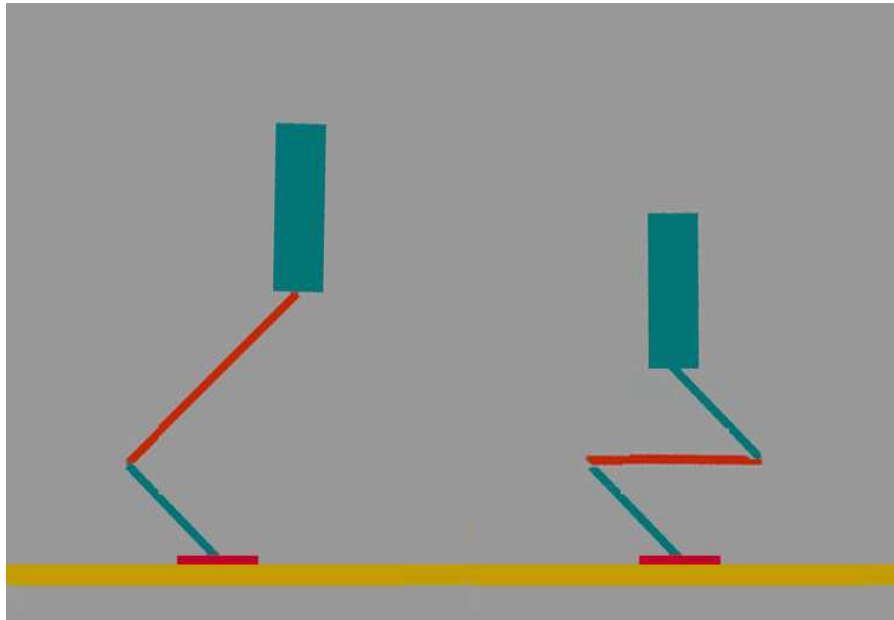
Pre vybrané pohyby sme následne určili požiadavky fyzického návrhu. Model má pozostávať z jednej nohy a tela, ktoré reprezentuje zvyšok robota (napr. riadiace mechanizmy, zložitejšiu konštrukciu robota). Ďalej sme museli zväziť, akým spôsobom bude mať model kontakt z podložkou. Do úvahy prišiel bodový kontakt, ako v prípade robotov [2, 3, 5, 6], alebo celé chodidlo [10, 17, 18]. Chodidlo poskytuje väčšiu stabilitu, nakoľko presunutie ťažiska nespôsobí okamžité preváženie. Pri práci však budeme jeho hmotnosť považovať za zanedbateľnú, podobne ako [10].

Aby sme dosiahli požadované pohyby, väčšina hmotnosti je sústredená v tele, ktoré je s chodidlom spojené  $n$ -počtom ramien. Ich hmotnosť priamo ovplyvňuje gymnastiku – prekonanie gravitácie, aby sa robot zdvihol z podlahy, nakláňanie tela ako spôsob riadenia pohybu presunutím ťažiska.

Počet ramien robota nie je striktno preddefinovaný. Ak uvažujeme konštrukciu s telom a chodidlom, dá sa uvažovať viacerých prístupoch návrhu. Dve pre nás najzaujímavejšie riešenia pozostávajú z 2 a 3 ramien (Obr. 3).

Tieto modely s ohľadom na konštrukciu tela robota môže klasifikovať ako

- dvojramenný – najbežnejší typ monopeda
- trojramenný - teoreticky ide o zaujímavý koncept, ale vzhľadom na konštrukciu je komplikovaný a nezodpovedá definícii humanoida.



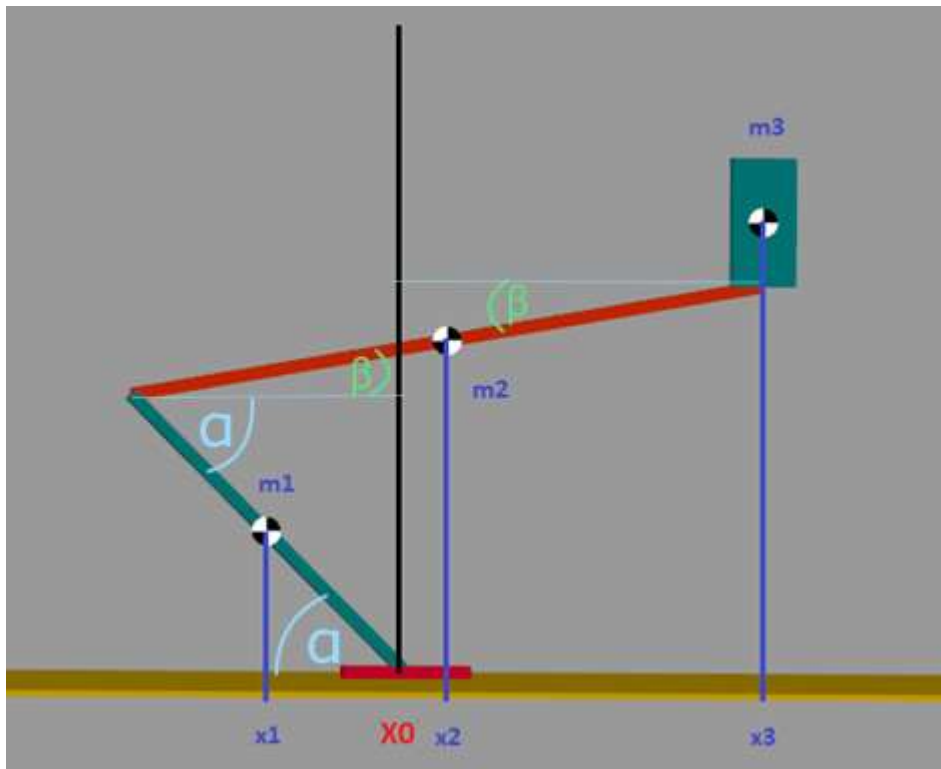
Obr. 3 Typy modelov monopeda

Z uvedených modelov sme našu pozornosť sústredili na dvojramenný model.

### 2.2.1 Dvojramenný model

Dvojramenný model pozostáva zvyčajne z dvoch rôzne dlhých ramien a je najuniverzálnejším modelom jednoohého robota. Výberom dĺžky ramien a rôznych hmotností jednotlivých komponentov vieme zostaviť v podstate akéhokoľvek robota. Tomuto konceptu sa budeme viacej venovať v ďalších kapitolách. Tento prístup využíva viacero robotov [3, 5]. Váhu ramien v tomto prípade kompenzuje posun podstatne ťažšieho tela robota oproti chodidlu, ktoré predstavuje referenčný bod systému. Ak sa ťažisko celého systému nachádza nad chodidlom, robot sa nepreváži.

V tomto prípade sa musia ťažiská dolného ramena ( $m_1$ ), horného ramena ( $m_2$ ) a tela ( $m_3$ ) vyrovnať a ťažisko bude teda ležať na osi, ktorá prechádza cez chodidlo robota. Vo východzej polohe pritom uhol, ktorý dolné rameno zvierá so zemou je uhol  $\alpha$ , uhol medzi telom a horným ramenom je  $\beta$ , a uhol medzi ramenami je  $\alpha + \beta$ .



Obr. 4 Univerzálny robot - poloha ťažísk

Pre zachovanie stability z pohľadu uloženia ťažiska univerzálného robota platí nasledujúci vzťah:

$$m_1 \cdot x_1 + m_2 \cdot x_2 + m_3 \cdot x_3 = 0 \quad (2-1)$$

pričom bod  $X_0$  na Obr.4 je považovaný za začiatok súradného systému

$m_1$	hmotnosť ramena 1 [kg]
$m_2$	hmotnosť ramena 2 [kg]
$m_3$	hmotnosť tela robota [kg]
$\alpha$	uhol ramena 1 so zemou [°]
$\beta$	uhol ramena 2 s telom [°]
$x_1$	poloha ťažiska $m_1$ vzhľadom k zemi [m]
$x_2$	poloha ťažiska $m_2$ vzhľadom k zemi [m]
$x_3$	poloha ťažiska $m_3$ vzhľadom k zemi [m]

Z tejto analýzy teda vyplýva:

- v prípade, že uvažujeme aj hmotnosť ramien, musíme telo čiastočne posunúť smerom doprava od vertikálnej osi, aby sme celkové ťažisko robota presunuli na os,
- ak os považujeme za nulu a hodnoty od nej vľavo za záporne a vpravo kladné, dostávame podmienku (2-1).

Musíme taktiež zvážiť, že gravitácia túto konštrukciu stiahne dole, takže kĺby musíme v simulácii fixovať aby sa nepohli.

V tomto prípade by nám zmena hmotnosti ktorejkoľvek častí robota spôsobí posun celkového ťažiska, ktorá by sa musela kompenzovať (napr. zmenou uhlu ramien), resp. by spôsobila kolaps robota.

### 2.2.2 Zjednodušenie konceptu

Pri vyvažovaní robota musíme uvažovať rozloženie váhy medzi ramená a telo. Toto rozloženie má veľký vplyv na dĺžku ramien a následne aj rozsah pohybu.

Prístup firmy Boston Dynamics k návrhu robotov slúžil pre nás ako hlavný zdroj inšpirácie. Väčšina hmotnosti robotov firmy je v „tele“ robota, pričom nohy zostávajú tenké a ich hmotnosť je v porovnaní so zvyškom tela malá až zanedbateľná.



Obr. 5 Roboty spoločnosti Boston Dynamics Spot [15] a Atlas [16]

Podobný prístup sme zvolili aj my. Boston Dynamics zatiaľ jednonohé roboty nepredstavila, ale väčšina ich modelov sleduje uvedený princíp, ktorý sme prevzali.

Z praktického hľadiska je tento prístup rozumnejší – ak uvažujeme využitie reálneho robota v teréne, napríklad na prenos materiálu, táto hmotnosť bude súčasťou hlavnej masy tela a nie ramien. Prakticky to teda znamená, že telo má premenlivú hmotnosť, ktorá je v každom prípade výrazne väčšia ako hmotnosť zvyšku robota.

V prípade, že by sme pracovali s dvojramenným robotom, aký je znázornený na Obr. 4, robot by mal byť sám o sebe vyvážený a hmotnosť ramien je kompenzovaná konštantnou hmotnosťou tela. Ak by prišlo k zaťaženiu tela pridaním závažia na pozíciu tela robota, došlo by k posunutiu ťažiska a následne by robot bez ďalšej korekcie pozícií ramien skolaboval.

Zjednodušený model na princípe Boston Dynamics predpokladá, že hmotnosť ramien je zanedbateľná – vo všeobecnosti do 5% celkovej hmotnosti robota bez vonkajších vplyvov – a robot je v staticky stabilnej polohe. Teoreticky môže byť nekonečne zaťažovaný, nakoľko sa ťažisko nachádza nad miestom kontaktu s plochou a preto nedôjde k preváženiu robota. Hmotnosť závažia je potom limitovaná pevnosťou materiálu ramien a kĺbov.

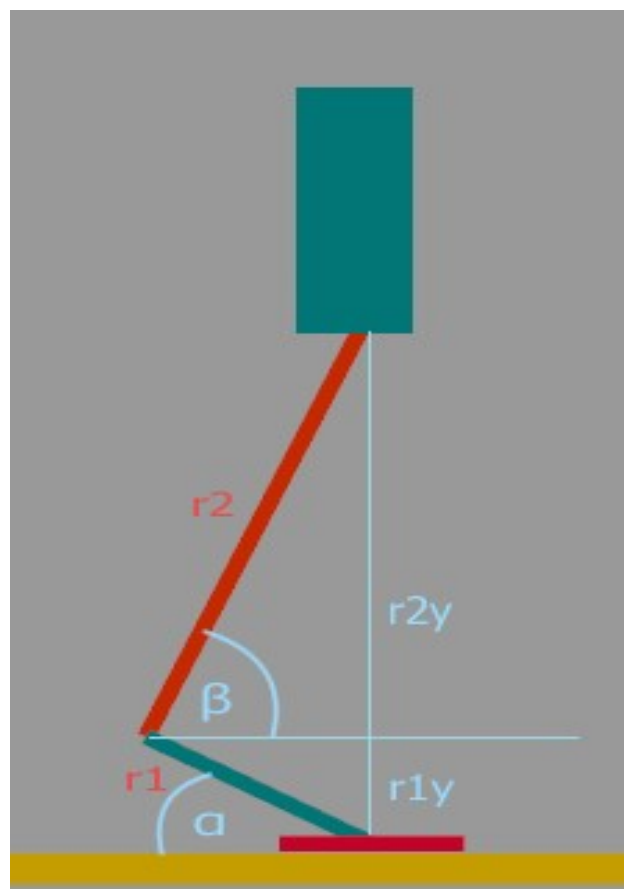
Pri návrhu riešenia sme teda uvažovali model, kde sa telo robota nachádza nad chodidlom a tvorí teoreticky 95% hmotnosti celého robota – ramená sme pôvodne uvažovali nulové, na čom boli stavané pôvodné rovnice, ale v *SimScape* bolo potrebné zaviesť určitú hmotnosť, inak by model nebol funkčný.

Je vhodné poznamenať, že ťažisko nebude presne na osi, ktorá prechádza spojom medzi chodidlom a dolným ramenom, ale bude minimálne vychýlené kvôli hmotnosti ramien. Táto výchylka je však zanedbateľne malá a nebudeme ju brať do úvahy v počiatočnom návrhu riešenia.

### 2.2.2.1 Matematické definovanie modelu

Pri tomto riešení predpokladáme, že hmotnosť ramien robota je teoreticky rovná nule. V realite uvažujeme o hmotnosti ramien na úrovni do približne 5% hmotnosti robota.

Hmotnosť je sústredená v tele a ramenách robota.



Obr. 6 Definovanie parametrov robota

**X** - ová pozícia je spoločná pre obe ramená ( $r_1$ ,  $r_2$ ).

$$r_{1x} = r_1 \cdot \cos \alpha \quad (2-2)$$

$r_1$  dĺžka dolného ramena [m]

$r_{1x}$  X-ová súradnica ramena  $r_1$  [m]

$\alpha$  uhol, ktorý rameno  $r_1$  zvierá so zemou [°]

Y – ová pozícia ramena  $r_1$  je daná vzťahom

$$r_{1y} = r_1 \cdot \sin \alpha \quad (2-3)$$

$r_{1y}$  Y-ová súradnica ramena  $r_1$  [m]

Pre rameno  $r_2$  platia nasledujúce vzťahy:

$$\cos(\beta) = \frac{r_{1x}}{r_2} \quad (2-4)$$

$r_2$  dĺžka horného ramena [m]

$\beta$  uhol, ktorý zvierá rameno  $r_2$  so zemou [°]

A toho následne určíme uhol  $\beta$

$$\beta = \arccos\left(\frac{r_{1x}}{r_2}\right) = \arccos\left(\frac{r_1 \cdot \cos(\alpha)}{r_2}\right) \quad (2-5)$$

Y – ovú zložku posunu ramena  $r_2$  vypočíta s použitím vzťahu:

$$r_{2y} = r_2 \cdot \sin(\beta) \quad (2-6)$$

$r_{2y}$  Y-ová súradnica ramena  $r_2$  [m]

Vzhľadom na pomer ramien  $r_1 - r_2$  môžeme zostrojiť 3 varianty robota:  $r_1 > r_2$ ,  $r_1 = r_2$  a  $r_1 < r_2$ . Prípád  $r_1 = r_2$  je z pohľadu pohybov najzaujímavejší, nakoľko povoľuje najširší rozsah pohybu.

Pre vyvažovanie platí rovnica 2-1, kde sú však  $m_1$  aj  $m_2$  sú rovné 0. Z toho vyplýva, že  $m_3$  môže byť ľubovoľné, lebo  $x_3 = 0$ .

## 2.3 Dynamika pohybu (teoretický rozbor)

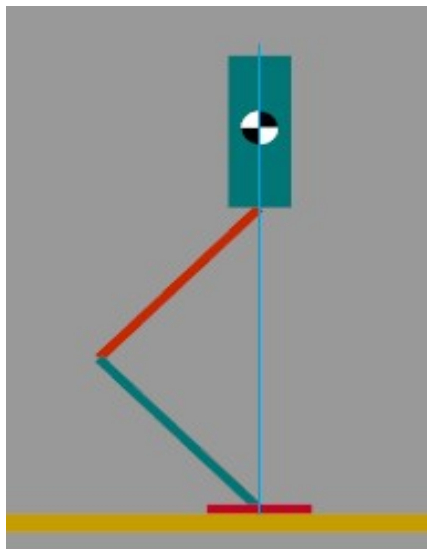
Táto kapitola poskytuje teoretický základ pre analýzu a návrh jednonohého robota. V každom štádiu pohybu treba zohľadniť statické, kinematické a dynamické vlastnosti modelu. Poznatky z tejto kapitoly sú neskôr využité pri simulácii v *SimScape Multibody* a detailne sa im budeme venovať v kapitole 3.

### 2.3.1 Statika pre stoj na mieste

Predtým, ako sa môžeme zaoberať pohybom robota, musíme dosiahnuť jeho stabilitu. Robot sa musí udržať v stoji, pričom musí byť v statickej rovnováhe, Vo všeobecnosti to znamená, že telo robota musí byť podporené v ťažisku alebo pod ním tak, že os kolmá k podložke prechádza ťažiskom. Polohy ťažísk ramien a tela sa musia navzájom kompenzovať. V prípade nášho modelu, kde uvažujeme ramená s malou hmotnosťou plne stačí, aby ťažisko tela bolo umiestnené nad chodidlom. V skutočnosti je ťažisko posunuté mimo os, ale toto vychýlenie je dostatočne malé, aby sme ho nemuseli zohľadniť pri našich výpočtoch. Prítomnosť chodidla taktiež pomáha pri udržaní stability.

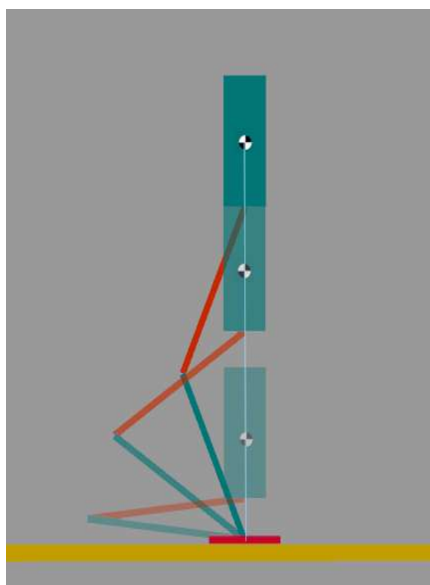
### 2.3.2 Drep

Drep ako pohyb definujeme ako presun ťažiska z východzej polohy smerom dolu, pričom sa robot nesmie prevážiť. Fyzikálne to znamená, že sa ťažisko musí pohybovať po priamke v smere osi Y.



Obr. 7 Robot v základnej pozícii

Po skrčení do drepu sa robot musí vedieť vrátiť do východzej polohy, pričom sa chodidlo nesmie odtrhnúť od zeme, z čoho vyplýva, že zrýchlenie vo vertikálnom smere pri vztyku musí byť menšie ako gravitačné zrýchlenie, teda  $a < g$ . Túto podmienku detailnejšie preberieme v kapitole 2.3.4 podmienka skoku.



Obr. 8 Priebeh drepu

Drep teda predstavuje dynamickú sekvenciu teoretických možností zostavenia robota. X-ová poloha ťažiska je zachovaná, mení sa iba konfigurácia uhlov ramien.

### 2.3.3 Skok

Skok je dynamický pohyb, pri ktorom dochádza k „odtrhnutiu“ sa chodidla od podložky. V našej práci hovoríme primárne o dvoch typoch skokov, a to skoku na mieste, ktorý tvorí nastavbu na drep, a následne skok do priestoru, kde robot prekonáva určitú vzdialenosť v smere osí X aj Y.

#### 2.3.3.1 Podmienka skoku

Aby sa robot dokázal odtrhnúť od zeme, musí byť ťažisko schopné dosiahnuť zrýchlenie, ktoré prevyšuje gravitačné zrýchlenie. Čím je menšia zmena výšky presunu ťažiska pri skoku, tým musí byť zrýchlenie väčšie.

Vyhádzame z rovnice pre dráhu, závislé na zrýchlení

$$s = \frac{1}{2} \cdot a \cdot t^2 \quad (2-7)$$

$s$  výška zdvihu [m]

$a$  zrýchlenie [ $\text{ms}^{-2}$ ]

$t$  čas [s]

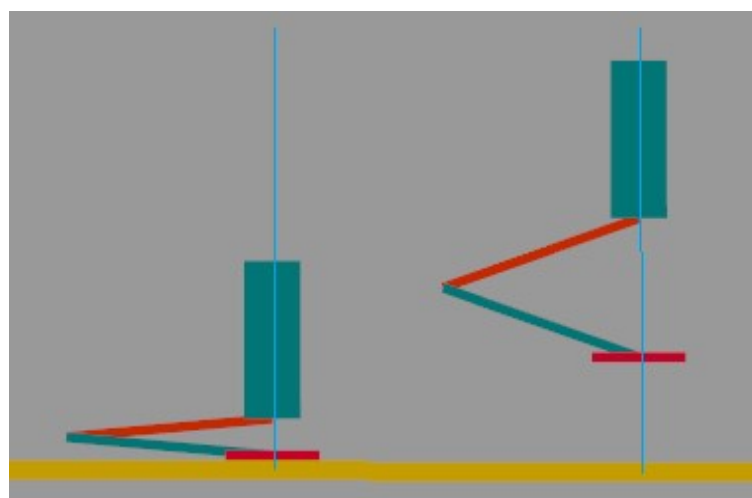
Vyjadrením času z tejto rovnice dostávame

$$t = \sqrt{\left(\frac{2 \cdot s}{a}\right)} \quad (2-8)$$

Týmto sme definovali limitný čas, kedy prichádza k odtrhnutiu sa robota od podložky. Reálne musí byť táto hodnota nižšia ako vypočítaná, aby prišlo k skoku.

### 2.3.4 Skok na mieste

Rozumieme pod ním skok, pri ktorom má robot dopadnúť na to isté miesto, z ktorého sa odrážal, s minimálnou odchýlkou. Pozostáva z troch fáz, príprava, let a dopad.



Obr. 9 Drep vs. skok na mieste

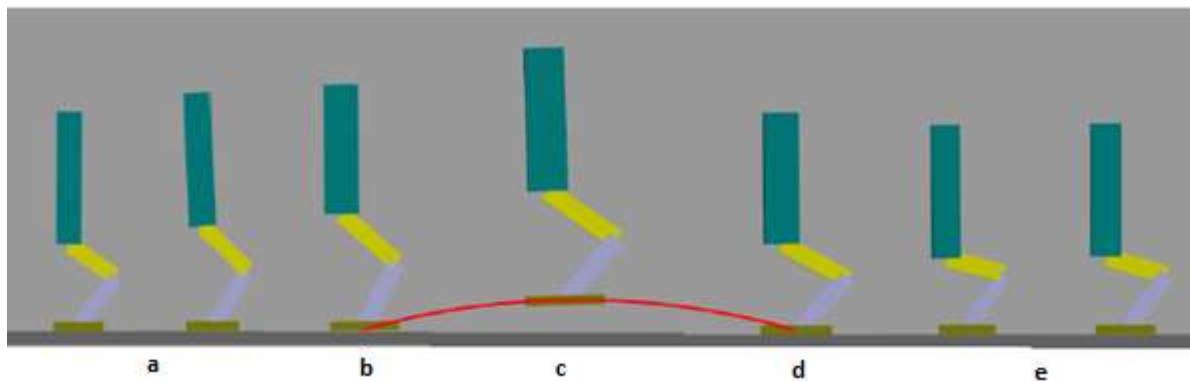
Pri všetkých týchto fázach bude ťažisko na tej istej osi, nakoľko nedochádza k posunu v rámci osi X.

Obrázok 10 zachytáva skok robota na mieste:

- Prípravná fáza (Obr. 10a) – robot sa skrčí na mieste do východzej požadovanej polohy
- Fáza odrazu (Obr. 10b) – v dôsledku zrýchlenia sa ťažisko výrazne posúva v Y-ovej osi
- Fáza letu (Obr. 10c) – po presiahnutí minimálneho potrebného zrýchlenia sa robot odráza od zeme. Podľa udeleného zrýchlenia dosiahne výšku definovanú vzťahom 2-7. Po dosiahnutí maximálnej výšky robot klesá v dôsledku gravitačného zrýchlenia
- Fáza dopadu (Obr. 10d) – robot pristáva na podložke, dochádza k opätovnému styku chodidla s podložkou.
- Fáza stabilizácie (Obr. 10e) – stabilizácia do základného postoja.

Detailnú analýzu fáz skoku rozoberajú [17, 18].

Grafy súvisiace zo skokom sú súčasťou kapitoly 3.2.4



Obr. 10 Fázy skoku na mieste [10]

### 2.3.5 Skok do diaľky

Pri skoku do diaľky robot prekonáva aj určitú vzdialenosť v horizontálnom smere. Pri tomto pohybe môže ísť o prekonanie dlhšej vzdialenosti alebo prekonaniu vertikálneho posunu, napríklad skok na schod alebo cez prekážku. Pre každý prípad dochádza k vychýleniu tela a tým aj ťažiska mimo rovnovážnu polohu, čím zvyšok robota „potiahne“ do zvoleného smeru. Čím je väčšie vychýlenie ťažiska od osi, o to ďalej sa robot potiahne.

Pre skok platia nasledujúce vzťahy

$$x = v \cdot \cos(\alpha) \cdot t \quad (2-9)$$

$$y = v \cdot \sin(\alpha) \cdot t \quad (2-10)$$

Pričom zároveň platí pre výšku skoku

$$y = \frac{1}{2} \cdot (a_r - g) \cdot t^2 \quad (2-11)$$

$\alpha$	uhol, ktorý dolné rameno zvierá zo zemou [°]
$v$	celková rýchlosť [ms <sup>-1</sup> ]
$x$	vzdialenosť [m]
$y$	výška výskoku [m]
$t$	čas trvania skoku [s]
$g$	gravitačné zrýchlenie [ms <sup>-2</sup> ]
$a_r$	vertikálne zrýchlenie voči podložke pri výskoku [ms <sup>-2</sup> ]

Pohyb robota je riadený dvomi ramenami. Smer skoku je závislý od výsledného vektora rýchlostí.

Pre pohyb v Y-ovej osi platí

$$v_y = r_1 \cdot \sin(\alpha) + r_2 \cdot \sin(\beta) \quad (2-12)$$

V prípade X-ovej osi platí

$$v_x = r_1 \cdot \cos(\alpha) - r_2 \cdot \cos(\beta) \quad (2-13)$$

Pre polohu ťažiska vychádzame z rotačných matíc odvodených pre rotáciu v rovine. Predošlé dve rovnice môžeme prepísať do maticového tvaru pre 2D:

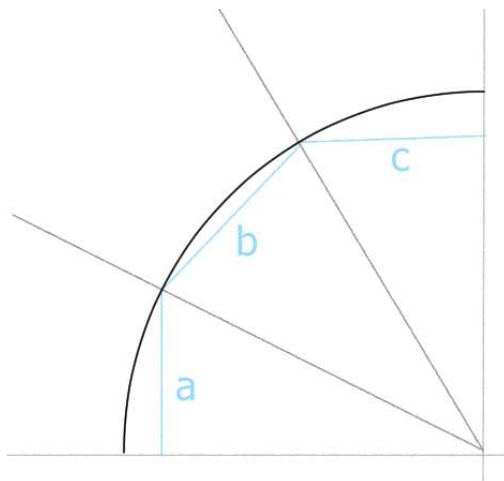
$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\cos\beta \\ \sin\alpha & \sin\beta \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \quad (2-14)$$

$v_x$	X-ová zložka rýchlosti pohybu [ms <sup>-1</sup> ]
$v_y$	Y-ová zložka rýchlosti pohybu [ms <sup>-1</sup> ]
$r_1$	dĺžka dolného ramena [m]
$r_2$	dĺžka horného ramena [m]
$\alpha$	uhol, ktorý dolné rameno zvierá zo zemou [°]
$\beta$	uhol, ktorý horné rameno zvierá zo zemou [°]

Uvedená matica je platná pre všetky typy pohybov, uvádzame ju však až pre skok do diaľky, nakoľko doteraz sme pracovali len s pohybom v jednej ose (os Y).

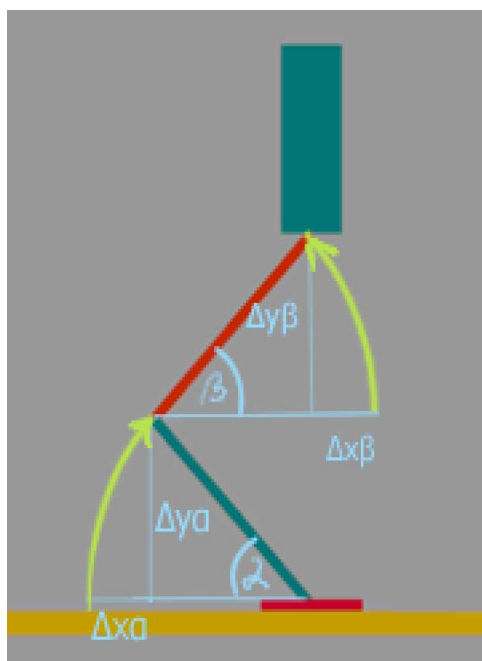
### 2.3.5.1 Závislost' pohybu od goniometrie

Pri skoku zohráva úlohu natočenie ramien a ich súradnice v osiach X a Y, ktoré sú goniometrickými funkciami polohy. V rozmedzí prvých  $30^\circ$  dochádza k najväčšej zmene v osi Y –  $\cos(30^\circ)$  je rovný 0,5 – Obr. 11, časť a. Z toho vyplýva, že rameno prejde 50% dráhy v smere osi Y za prvých  $30^\circ$ . V porovnaní s tým, na osi X je posun o necelých 0,3 dĺžky ramena. Jedná sa o viac ako 40% rozdiel.



Obr. 11 Závislosť zmeny posunu v osy X a Y od uhla natočenia

Opačná situácia nastáva pre uhly  $60^\circ - 90^\circ$ . Y-ová súradnica musí prejsť posledných 29% dráhy, zatiaľ čo X-ová zložka prejde 50% celkovej dráhy, čo je zrejmé z Obr. 11, časť c. Pre uhly  $30^\circ - 60^\circ$  je táto zmena nie až tak výrazná (Obr. 11 časť b). Tieto zmeny sú ďalej významné pri skoku do diaľky/výšky.



Obr. 12 Pomerná zmena polohy v smere osi X a Y

Na Obr. 12 máme znázornenú zmenu polohy ramien v závislosti od uhlu natočenia  $\alpha$  a  $\beta$ . Výsledný vektor pohybu tela robota je potom daný ako súčet vektorov znázornených na Obr. 13.

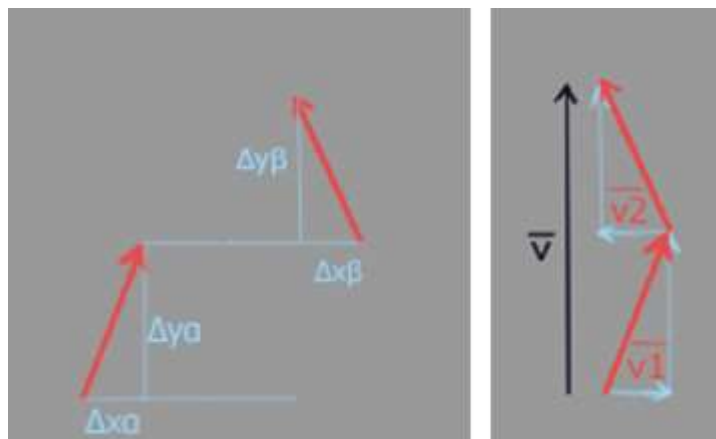
$$\vec{v} = \vec{v}_1 + \vec{v}_2 \quad (2-15)$$

$\vec{v}_1$  vektor rýchlosti pohybu dolného ramena [ $\text{ms}^{-1}$ ]

$\vec{v}_2$  vektor rýchlosti pohybu horného ramena [ $\text{ms}^{-1}$ ]

$\vec{v}$  výsledný vektor rýchlosti pohybu [ $\text{ms}^{-1}$ ]

Pokiaľ sa X-ové zložky tohto vektoru rovnajú, prichádza iba k posunu robota v smere osi Y. Celkový posun v ose Y je potom daný súčtom Y-ových zložiek pohybu jednotlivých ramien, znázornené na Obr. 13. V prípade, že sa X-ová veľkosť vektora  $v_1$  nerovná X-ovej veľkosti  $v_2$ , dochádza k skoku doprava, resp. doľava, v závislosti od rozdielu X-ových zložiek týchto vektorov.



Obr. 13 Vektory smeru pohybu

Jednotlivé zložky vektora rýchlosti zohrávajú zásadnú úlohu v procese simulácie skokov do diaľky, ktorej sa venujeme v kapitole 3.2.10.

### 3 SIMULÁCIA ROBOT A V PROSTREDÍ MATLAB / SIMULINK

Simulácia navrhnutého riešenia prebiehala v prostredí Simulink, verzia R2022b a bola realizovaná v dvoch úrovniach:

- Matlab – definovanie parametrov robota a podmienok simulácie pre Simulink,
- Simulink – zostavenie robota a simulácia činnosti robota v 3D prostredí, pričom sa pohybujeme iba v rámci 2 osí.

V simulácii sme sa zamerali na nasledujúce oblasti:

- zostavenie robota na základe požadovaných parametrov
- stoj na mieste
- drep
- skok
  - na mieste
  - do diaľky.

Celá simulácia je sústredená do 7 skriptov:

- *robot\_definicia.m* – definovanie parametrov na zostavenie robota,
- *robot\_stoj.m* – základný stoj robota v definovanej pozícii,
- *robot\_drep.m* – prezentácia drepu,
- *robot\_skok\_miesto.m* – skok na mieste,
- *robot\_skok\_strana.m* – skok do strán,
- *robot\_exhibicia.m* – prezentácia zostavy cvikov,
- *robot\_prezentacia.slx* – simulácia zostavenia robota na základe vstupných požiadaviek, prezentujúceho zostavu preddefinovaných cvikov.

#### 3.1 Definovanie parametrov robota v prostredí Matlab

V práci sme sa venovali modelu robota na princípe Boston Dynamics. Robot je definovaný množinou nasledujúcich premenných. Pomenovanie jednotlivých častí robota v programe (skripte *robot\_definicia.m*) uvádzame v nasledujúcej tabuľke Tab.2.

Výška robota je daná súčtom dĺžka dolného a horného ramena a je preddefinovaná na 22 cm. Pri voľbe výšky robota sme sa inšpirovali robotom Salto-1P [5]. Výšku tela do výšky robota nezapočítavame. Jedným z dôvodov tohto rozhodnutia je skutočnosť, že z pohľadu aktivít robota nás zaujíma iba rozsah pohybu ramien – min/max pozícia.

Po spustení skriptu *robot\_definicia.m* nás program vyzve k voľbe požadovaného typu zostavenia robota: preddefinovaná (Default) alebo užívateľsky definovaná (Custom) verzia robota. V prípade voľby užívateľskej verzie je požadované zadať parametre definujúce východzie postavenie robota. (viď. Obr. 6)

Na základe voľby dĺžky dolného ramena program umožní definovať uhol medzi ramenom a podložkou v rozsahu, kedy môže robot zaujať stabilnú polohu. Dĺžka horného ramena ako aj príslušné uhly pre horné rameno a telo sú dopočítané aplikáciou.

Podľa zvoleného spôsobu zostavenia robota k nemu v rámci simulácii pristupujeme jedným z uvedených spôsobov:

- užívateľom definované parametre robota, slúži ako demonštrácia možnosti konštrukcie robota (vid'. Obr. 14)
- preddefinovaným nastavením, ktoré je v práci *využívané v procese simulácie aktivít robota*

$$\text{ArmDown\_Y} = 0.11$$

$$\text{ArmUp\_Y} = 0.11$$

$$\text{alfa} = 45^\circ$$

Tab. 2 Parametre robota

Premenná	Veličina	Hodnota	Opis
Foot_X	m	0,050	Rozmer chodidla v smere X
Foot_Y	m	0.005	Rozmer chodidla v smere Y
Foot_Z	m	0,,030	Rozmer chodidla v smere Z
Body_X	m	0,030	Rozmer tela robota v smere X
Body_Y	m	0,080	Rozmer tela robota v smere Y
Body_Z	m	0,050	Rozmer tela robota v smere Z
ArmDown_X	m	0,005	Rozmer dolného ramena v smere X
ArmDown_Y	m	premenná	Rozmer dolného ramena v smere Y
ArmDown_Z	m	0,020	Rozmer dolného ramena v smere Z
ArmUp_X	m	0,005	Rozmer horného ramena v smere X
ArmUp_Y	m	premenná	Rozmer horného ramena v smere Y
ArmUp_Z	m	0,020	Rozmer horného ramena v smere Z
Total_Height	m	0.22	Celková výška robota

```
Select [D]efault or [C]ustom robot definition: c

Hight of body: 0.22

Custome configuration

Lower arm length [m]: .07

Accepted angle: 5.00 - 90 degrees
Lower arm angle: 40

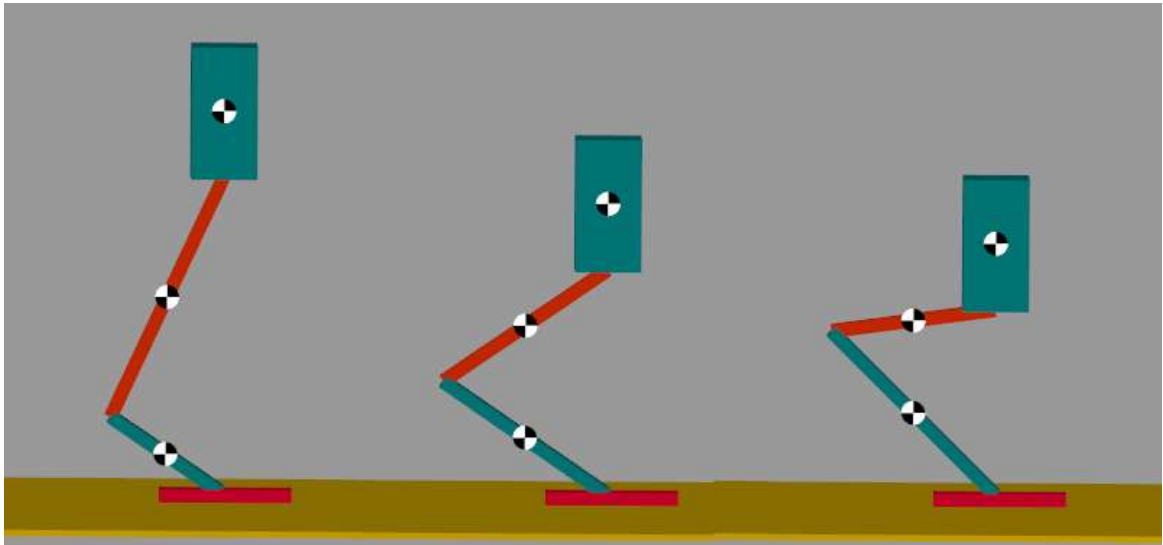
Required parameters
* Down arm: 0.07 m
* Down angle: 40.00 °

Calculated parameters
* Up arm: 0.15 m
* Up angle: 69.05 °

Robot is completed
```

Obr. 14 Proces zostavenia užívateľsky definovaného

Po spustení skriptu *robot\_prezentacia.slx* simulátor zostaví robota. Na Obr.15 je zobrazený robot v troch režimoch v závislosti od dĺžky dolného ramena.



Obr. 15 Tri možnosti zostavenia robota

### 3.2 Simulácia robota v prostredí Simulink

Cieľom práce bolo vytvoriť model robota schopného realizovať množinu plánovaných aktivít, uvedených v predošlej kapitole. Pri návrhu sme postupovali od jednoduchšej štruktúry k zložitejšej – ucelenej.

Prvou úlohou bolo zostaviť stabilného robota na základe vstupných parametrov. Následne sme pristúpili k drepu, skoku na mieste, skoku do diaľky, na a cez prekážku.

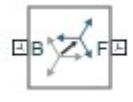




Simulácia prebiehala na modeli robota vychádzajúceho z princípu zjednodušeného modelu definovaného v kap. 2.2.2.

### 3.2.1 Opis blokov Simulinku

Pri budovaní robota sme využili nasledujúce moduly/bloky z knižnice SimScape.

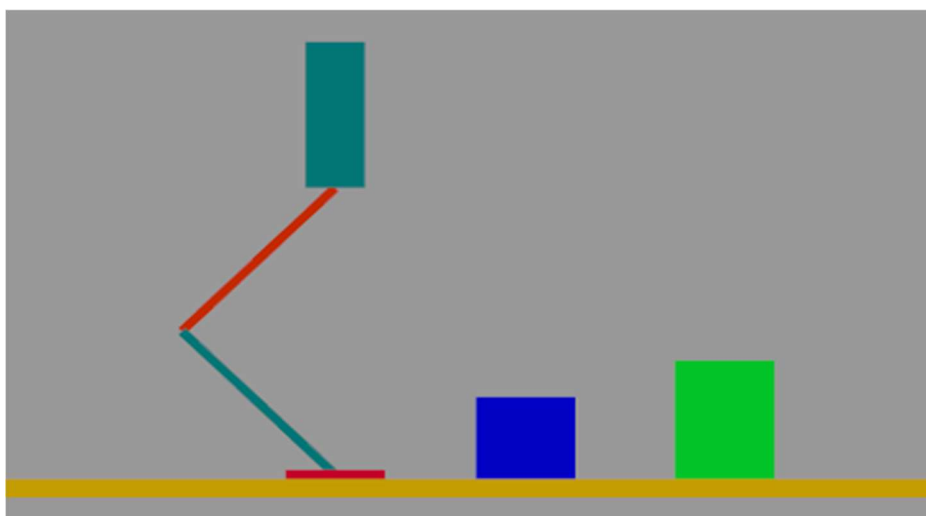
Detailný opis všetkých blokov je súčasťou manuálu Matlab resp. je dostupný na web stránke [21].

Tab. 3 Bloky Simulink použité pri simulácii

Blok	Názov	Opis
	Rigid Transform [23]	Natočenie, posun
	Brick Solid [24]	Telo komponentu
	Spatial Contact Force [25]	Kontakt medzi telesami
	Prismatic Joint [26]	Kĺb s jedným stupňom voľnosti (pohyb)
	Revolute Joint [27]	Kĺb s jedným stupňom voľnosti (rotácia)

### 3.2.2 Zostavenie robota

Robot je v priebehu simulácie umiestnený v preddefinovanom prostredí, v ktorom prezentuje jednotlivé cviky. Spolu s robotom sme zároveň vytvorili okolité prostredie a definovali väzby medzi robotom a jeho okolím.



Obr. 16 Robot a jeho okolie

Na obrázku Obr. 16 vidíme robota v jednoduchom simulovanom prostredí. Za prostredie, v ktorom sa robot pohybuje pokladáme podložku a dve kocky. Definícia prostredia je súčasťou modrého bloku Obr.17. Bloky *Solid* definujú parametre jednotlivých komponentov. (rozмеры, farbu, hmotnosť...) Umiestnenie komponentu v priestore je realizované pomocou blokov *Rigid Transform*. Tento blok umožňuje definovať polohu a natočenie každého simulovaného objektu vzhľadom na referenčný element. Referenčným bodom je ťažisko podložky, po ktorej sa robot pohybuje. Pre každý objekt sme pomocou bloku *Spatial Contact Force* definovali spôsob jeho interakcie s iným objektom. Nastavenie parametrov je rozdelené do sekcii

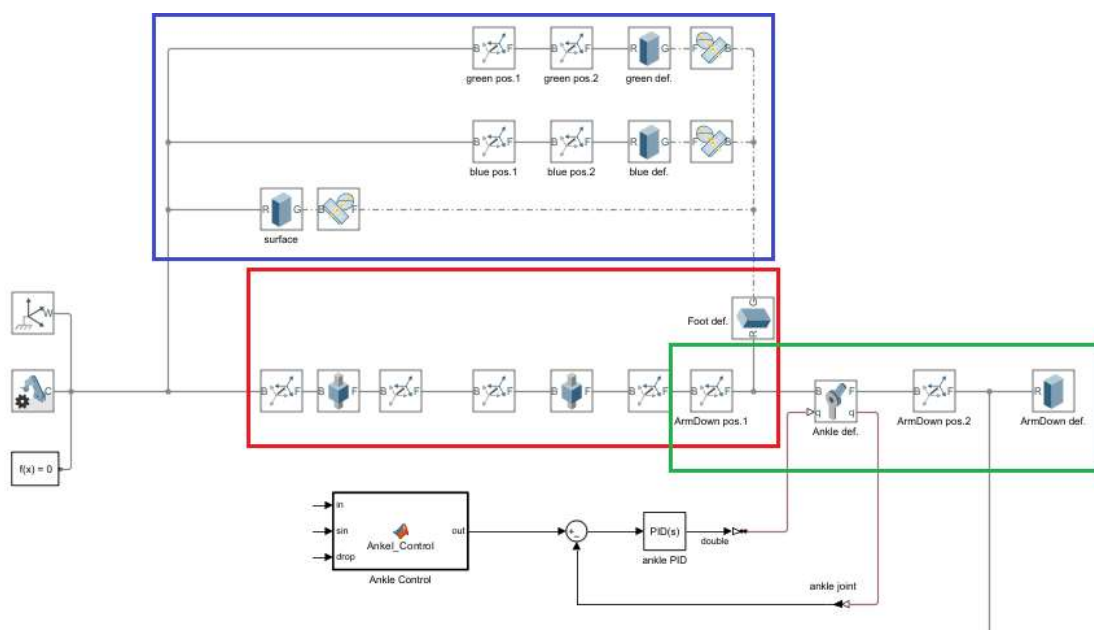
- Normal Force
- Frictional Force
- Sensing
- Zero-Crossings

Nastavením požadovaných parametrov tak môžeme simulovať interakciu medzi telesami za rôznych podmienok. My sme pracovali s prednastavenými hodnotami SimScape.

Červený rámček na Obr.17. obsahuje schému pre definíciu chodidla. Východziu pozíciu chodidla sme opäť definovali pomocou blokov *Rigid Transform*. Zároveň sme s využitím *Prismatic Joint* zabezpečili pohyb robota v smere osy X a Y. Parametre pre chodidlo si Simulink preberá z vygenerovaných dát.

V zelenom rámčeku je proces vytvorenia dolného ramena robota. Pozíciu ramena vzhľadom na chodidlo sme nastavili pomocou *Rigid Transform* blokov. Blok *Revolute Joint* predstavuje otočný kĺb s jedným stupňom voľnosti. Ovládaním polohy kĺbu realizujeme pohyb robota. Konfiguračné parametre kĺbu sú opäť prevzaté z vygenerovaných dát.

Obdobným spôsobom sme pridali horné rameno a telo robota.

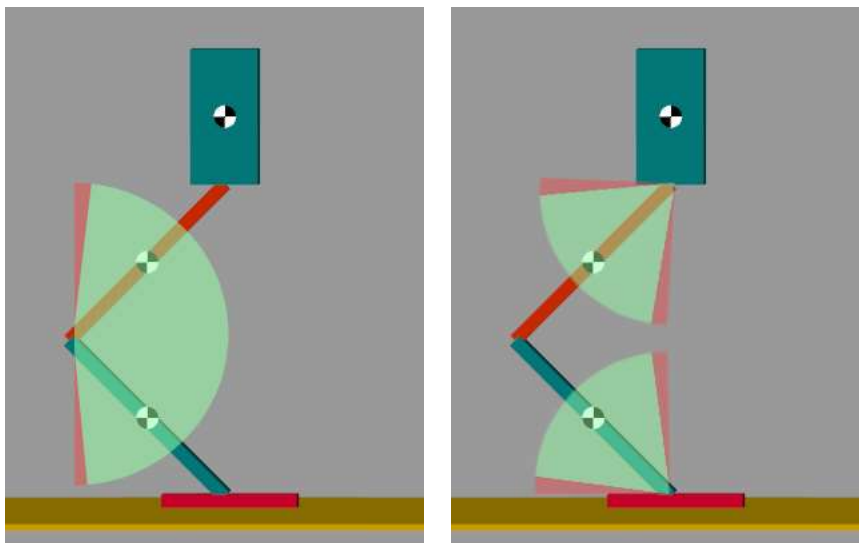


Obr. 17 Proces zostavenie robota a jeho okolia

### 3.2.3 Tvrdé obmedzenia

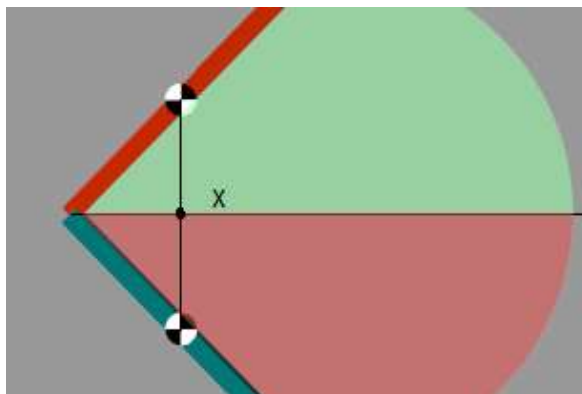
Pri návrhu robota sme museli zaviesť určité limity a obmedzenia pohybu. Tieto obmedzenia zahŕňajú extrémny, ktoré by reálny/fyzický robot nebol schopný dosiahnuť, respektíve by boli príliš náročné a pre jeho funkciu úplne zbytočné. Majú taktiež matematický význam.

Našou snahou je vyhýbanie sa okrajovým hodnotám  $0^\circ - 90^\circ - 180^\circ$ , ktoré sú pre sínus, resp. kosínus rovné nule – čiže polohy, ktoré sú rovnobežné alebo kolmé na plochu. Kĺby členku a bedra sa pohybujú v rozmedzí  $5^\circ - 89^\circ$ , zatiaľ čo koleno nesmie dosiahnuť celých  $180^\circ$  a teda sa úplne vystrieť, ako je znázornené na Obr. 18.



Obr. 18 Zóny pohybu ramien robota

Ak by sme dosiahli jeden z týchto extrémov, model sa dostáva do pozície, kde sa rameno môže pohybovať dvoma smermi, kde stále splňa podmienku rovnováhy. Rameno sa v tomto prípade môže zdvihnúť hore alebo prejsť nižšie pod myslenú nulu (v prípade vertikálnej polohy sa rameno preklápa zľava doprava, resp. naopak), nakoľko X-ová súradnica je v oboch prípadoch zhodná. V prípade prechodu pod túto nulu sa však rameno môže pohybovať nižšie, ako je schopné, a robot by sa zrútil.



Obr. 19 Veľkosť momentu sily vzhľadom na os X

Na Obr. 19 vodorovná os tvorí hraničnú zónu. Bod  $x$  naznačuje relatívnu hodnotu ramena na X-ovej osi: môžeme jasne vidieť, že táto vzdialenosť je zhodná hore aj dole vzhľadom k X-ovej ose, ale rameno by bolo v jednom prípade korektne uložené (zelená zóna) a v druhom by ležalo príliš nízko (červená zóna).

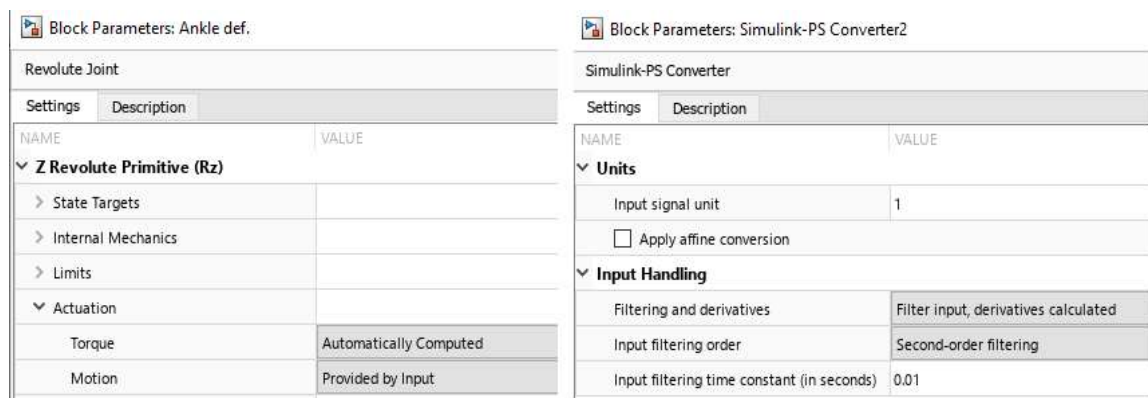
### 3.2.4 Riadenie pohybu

Simulink umožňuje riadenie pohybu / polohy kĺbov dvomi metódami:

- Torque – moment,
- Motion – uhol natočenia.

V práci sme použili metódu ovládania pohybu zmenou uhla natočenia. Tento spôsob riadenia využívajú napr. autori článku [9].

V prípade riadenia pohybu pomocou nastavenia uhla ramena si simulátor načíta požadovanú hodnotu z preddefinovanej riadiacej tabuľky. Tejto hodnote uhla následne Simulink dopočíta zodpovedajúcu hodnotu momentu. (Obr. 20 vľavo)



Obr. 20 Nastavenie parametrov pre vstupný signál

Aby simulácia prebehla korektne, musíme simulátoru poskytnúť informáciu, akú vzorkovaciu frekvenciu sme pri výpočte riadiacej tabuľky použili. V našom prípade sme sa ohľadom na použitý počítač pracovali so vzorkovaním 1000 Hz (0,001 sekundy).

Riadiaca tabuľka pohybu jednotlivých kĺbov sa generuje skriptom *robot\_exhibicia.m*. Časový priebeh (scenár) požadovanej akcie je definovaný v tabuľke *action\_scenar*. Scenár z nášho pohľadu prezentuje časový sled aktivít vykonávaných robotom. Výstupom je tabuľka *action\_control*, ktorá obsahuje hodnoty premenných v čase, s krokom 0,001 sekundy.

V nasledujúcich riadkoch opíšeme proces tvorby scenára jednotlivých akcií ako aj následného generovania riadiacej matice. Všetky aktivity robota sú riadené pomocou tabuľky *action\_scenar*. Táto tabuľka má 4 stĺpce:

- to* – definuje nasledujúcu finálnu polohu robota, do ktorej sa má robot dostať,
- dur* – doba potrebná na prechod do novej požadovanej polohy,
- left* – veľkosť vektora pri skoku doľava,
- right* – veľkosť vektora pri skoku doprava.

V Tab. 4 prezentujeme spôsob zápisu požadovaného scenára.

Prvý riadok obsahuje počiatočnú, východziu pozíciu robota. Táto pozícia zodpovedá požiadavke základného postoja robota po zostavení. V nasledujúcom riadku definujeme novú požadovanú polohu a čas potrebný na prechod do novej pozície. Ak zachováme hodnotu premennej *to* nezmenenú, robot zotrvá v predošlej polohe po definovanú dobu. Doba trvania akcie je udávaná v počte vzorkovacích krokov.

Signál je vzorkovaný frekvenciou 1000 Hz. Na základe toho potom platí, pre dobu trvania akcie vzťah

$$t = 0,1 \cdot dur \quad (3-1)$$

*t* doba trvania akcie [s]

*dur* počet vzorkovaní / taktov [-]

Zmenou hodnoty premennej *to* tak vieme kontrolovať polohu robota. Pokiaľ je rýchlosť zmeny pomalšia ako gravitačné zrýchlenie, robot robí drepy, ako bolo definované v kapitole 2.3.3.1. Po prekročení gravitačného zrýchlenia robot vyskočí. Smer skoku je riadený pomocou premenných *left* a *right*. Zmenou hodnoty jednej z týchto premenných meníme vektor pohybu príslušného ramena. Pre skok doprava je potrebné spomaliť pohyb horného ramena. V prípade skoku doľava zasa dolného ramena. Teoreticky sme túto problematiku rozobrali v kapitole 2.3.5.

Tab. 4 Matica riadenia scenára aktivít robota

```

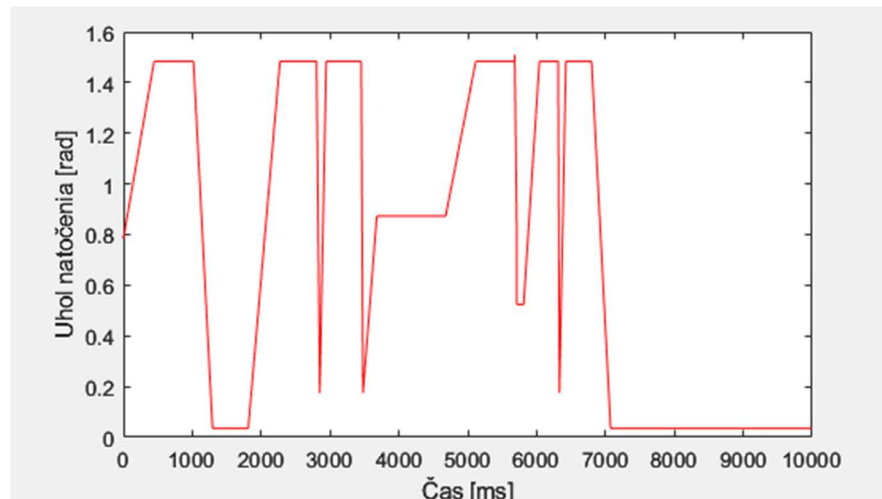
action_scenar = [
% to      dur.      left      right
start_pos 100      1         1;      % vychodzia poloha
0         500      1         1;      % dolu
0         500      1         1;      % pauza
90        300      1         1;      % vztyk
90        500      1         1;      % pauza
0         500      1         1;      % dolu
0         500      1         1;      % pauza
120       20       1         0.7;    % skok doprava
0         100      1         1;      % dolu
0         250      1         1;      % pauza
120       20       0.7       1;      % skok dolava
0         100      1         1;      % dolu
90        300      1         1;      % vztyk
90        5000     1         1;      % pauza
...
]

```

Skutočné riadenie pohybov ramien v procese simulácie je realizované pomocou hodnôt matice *action control*. Matica má 4 stĺpce. Prvý stĺpec obsahuje časovú informáciu. Zvyšné stĺpce riadia pohyb kľbov.

Riadiacu maticu generuje skript *robot\_exhibicia.m* na základe scenára hodnôt v tabuľke *action\_scenar*. Príklad scenára je zobrazený na Obr. 21. X-ová os predstavuje čas v krokoch/ms. Pri vzorkovaní 0,001s tak 10 000 krokov zodpovedá času 10s. Na Y-ovej osi je vynesená hodnota uhlu  $\alpha$  v rad.

Generovanie simulácie (matice *action\_control*) beží ako cyklus v cykle. Hlavný cyklus je cyklus spojený so scenárom. Vnútorňý cyklus zabezpečuje výpočet hodnôt riadiacich premenných v jednotlivých krokoch simulácie.



Obr. 21 Grafická prezentácia scenára

Algoritmus generovania riadiacej matice (Obr. 22) najskorej v rámci cyklu scenára načíta parametre kroku vykonávanej akcie. Na základe týchto parametrov modifikuje hodnoty premenných potrebné pre výpočet riadenia pohybu robota. Zároveň nastaví premenné pre cyklus generovania matice. Na základe vstupných hodnôt a požadovanej akcie nasleduje výpočet hodnôt riadiacich signálov pre jednotlivé kĺby v danom časovom okamžiku. Vypočítané hodnoty sú zapísané do tabuľky *action\_control*.

Nasleduje overenie hodnôt premenných riadiacich vnútorňý cyklus výpočtu. Ak neboli splnené podmienky pre ukončenie vnútorného cyklu, algoritmus modifikuje hodnotu časovej značky a nastaví hodnotu pre cyklus. Následne pokračuje vo výpočte riadiacich hodnôt.

V prípade, že je vnútorňý cyklus ukončený, riadenie preberá cyklus riadenia scenára. Príde k načítaniu dát potrebných k vykonaniu ďalšieho kroku scenára. Opäť sa modifikujú premenné pre proces výpočtu riadiacich hodnôt. Počítadlo cyklov scenára sa nastaví na nasledujúcu hodnotu. Následne sa spracovania dát posúva do vnútorného cyklu.



Obr. 22 Vývojový diagram generovania riadiacej matice

### 3.2.5 Podmienky a priebeh simulácie

Doba trvania simulácie zvolená je 5 až 10 sekúnd v závislosti od scenára tak, aby bolo možné do nej umiestniť celú zostavu.

Aby sme vytvorili robota, spustili sme skript *robot\_definicia.m* v Default móde. Skrip nám zostaví robota, ktorého celková výška, dĺžka ramiemien bez tela, je 22 cm. Dolné rameno zvierá s podložkou  $45^\circ$  uhol. Toto je základný, východzí postoje robota.

Následne podľa simulovanej situácie sme si zvolili požadovaný skrip, ktorý vygeneruje riadiacu maticu pre pohyb robota. K dispozícii sú nasledujúce skripty:

- *robot\_stoj.m* – základný stoj robota v definovanej pozícii,
- *robot\_drep.m* – prezentácia drepu,
- *robot\_skok\_miesto.m* – skok na mieste,
- *robot\_skok\_strana.m* – skok do strán,
- *robot\_exhibicia.m* – prezentácia zostavy cvikov,

Po spustení zvoleného skriptu je vytvorená matica riadiaca pohyb robota. Až následne je možné spustiť simuláciu v prostredí Simulinku spustením skriptu *robot\_prezentacia.slx*.

Simuláciu sme realizovali na robotovi s nasledujúcimi parametrami. Ako vidieť v tabuľke tab. 5, priblížili sme sa k našim teoretickým predpokladom.

Tab. 5 Hmotnostné parametre robota

	horné rameno	horné rameno	chodidlo	telo
<b>Hmotnosť [kg]</b>	0,0055	0,0055	0,0038	0,1800
<b>Podiel na hmotnosti v %</b>	3%	3%	2%	92%

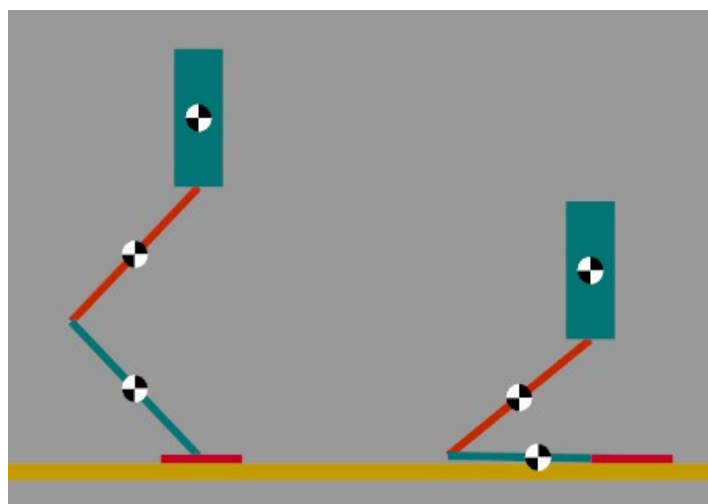
Počas simulácie sme v Simulinku ponechali základné nastavenia kĺbov ako aj parametrov definujúcich kontakt medzi chodidlom a podložkou. Voľba a definovanie týchto parametrov predstavuje zložitý proces, ktorý je časovo náročný a vyžaduje isté skúsenosti v tejto oblasti.

### 3.2.6 Stoj na mieste – ovládanie robota pomocou momentu

V prípade stoji na mieste sme sa rozhodli otestovať obe možnosti jeho ovládania. Ako prvú možnosť sme otestovali riadenie robota pomocou zmeny momentu v kĺboch.

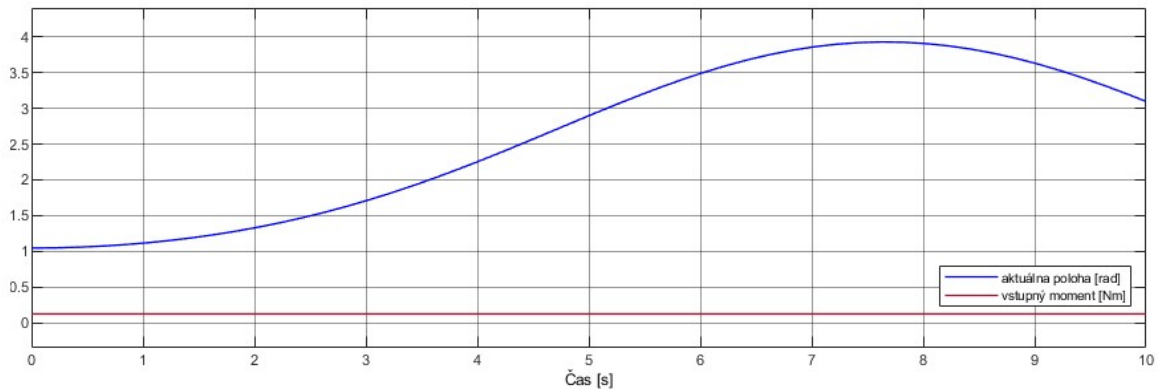
Prvým krokom simulácie bolo zostavenie robota na základe požadovaných parametrov a dosiahnuť, aby zostal stáť v požadovanej, vzpriamenej podobe. Proces zostavenia robota sme opísali v predošlej kapitole. Udržanie robota v stabilizovanej polohe je realizované pomocou riadiacich signálov privádzaných do jednotlivých kĺbov. Aby robot zaujal fixný postoj, signál na vstupe musí mať konštantnú hodnotu.

Pri zachovaní pomeru súčtu hmotností ramien robota k hmotnosti jeho tela na úrovni rádovo niekoľko percent, sa robot neudrží vo vzpriamenej polohe. Dôvodom je moment, ktorý vytvárajú ramená a ten následne spôsobí kolaps robota (Obr. 23).



Obr. 23 Kolaps robota

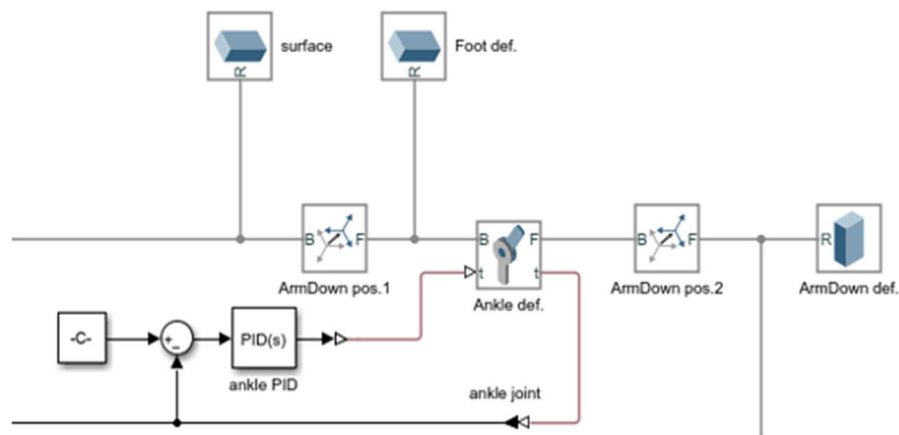
Priebeh kolapsu robota znázorňuje nasledujúci graf na Obr. 24. Tu si treba uvedomiť, že pre konštantnom momente na vstupe simulátor dopočítava uhol natočenia.



Obr. 24 Priebeh kolapsu - grafy

### 3.2.6.1 Implementácia PID regulátora

Aby sme zabránili neželanému kolapsu robota, bolo potrebné implementovať spätnú väzbu s využitím PID regulátora pri riadení polohy kĺbu. Zapojenie PID regulátora do obvodu je na Obr.25.



Obr. 25 Implementácia PID regulátora

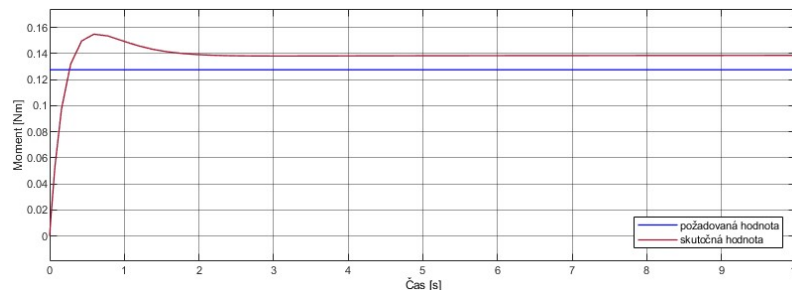
V počiatočnom štádiu nastavovania hodnôt regulátora sme využili možnosti ladenia hodnôt regulátora aplikáciou. Následne sme sa manuálnym testovaním hodnôt postupne dopracovali k udržaniu stabilnej polohy robota. Manuálnemu prístupu ladenia PID regulátora sa nevyhli ani autori článku [22].

Tab. 6 Experimentálne získané hodnoty PID regulátora

Zložka	Hodnota
P	20
I	30
D	0
N	0

Použitím PID regulátora pre všetky časti robota príde k rýchlemu ustáleniu sa v požadovanej polohe, ako vidíme na Obr. 26.

Na tomto mieste si treba uvedomiť, že získané parametre PID regulátora sú úzko späté s konštrukciou robota. Menia sa v závislosti na rozmeroch a hmotnosti jeho jednotlivých častí, ako aj na rýchlosti pohybu. K problematike nastavenia hodnôt regulátora sa vrátíme v nasledujúcich kapitolách.



Obr. 26 Stabilizácia polohy robota po implementácii PID regulátora

### 3.2.7 Stoj na mieste – ovládanie robota pomocou uhlu natočenia

Následne sme sa zamerali na ovládanie polohy robota zmenou uhlu natočenia ramien. Poloha postoja robota je aj v tomto prípade riadená tabuľkou *action\_control*.

Priebeh sledovaných signálov v jednotlivých kĺboch je súčasťou príloh. Prílohy obsahujú grafy pre všetky ďalej opísané scenáre. V každom scenári sledujeme pre všetky kĺby priebeh z dosiahnutia požadovanej polohy (horný graf na obrázku) ako aj zmenu rýchlosti a momentov (dolný graf na obrázku). Vo všetkých grafoch predstavuje modrá farba požadovanú hodnotu sledovanej veličiny, oranžová farba jej skutočnú hodnotu. Toto pravidlo platí pre všetky scenáre.

Simuláciu stabilného stoja na mieste sme nechali spustenú 5 s, aby sme si boli istí, že nepríde ku kolapsu robota, ako v prípade riadenia pomocou zmeny momentu. Ako vidieť na grafoch Obr. 31 – Obr. 33, robot sa dokáže udržať v stabilizovanej polohe počas celej tejto doby.

Priebeh sledovaných veličín počas simulácie je zachytený na grafoch v prílohe. (Obr. 31 – Obr. 33). V počiatočnom štádiu simulácie vidíme pri všetkých sledovaných veličinách prekmity. Tento jav sa objavuje pri všetkých simuláciách a preto predpokladáme, že môže súvisieť so zostavením robota simulátorom. K ustáleniu v požadovanej polohe príde v čase 1 s. Zároveň sa stabilizujú hodnoty momentov a uhlová rýchlosť klesne na nulu. Priebeh sledovaných veličín má rovnaký charakter vo všetkých kĺboch.

Rýchlosť ustálenia prechodových javov je závislá od parametrov PID regulátora. Touto problematikou sa budeme zaoberať v kapitole 3.2.12.

### 3.2.8 Drep

Drep predstavuje plynulý prechod z jednej stabilnej polohy do druhej. Definíciu drepu sme sa venovali v kapitole 2.3.2 a 2.3.4.

Riadenie priebehu drepu ako aj skoku je realizované pomocou matice hodnôt *action\_control*. Tento princíp riadenia využívajú aj autori publikácií [9].

Požadovanú riadiacu maticu sme generovali v samostatnom skripte. Drep je simulovaný v 2 režimoch:

- pomalý plynulý drep,
- rýchly drep – ak vertikálne zrýchlenie počas vztyku prekoná veľkosť gravitačného zrýchlenia, predchádza do skoku.

Na to, aby robot vyskočil, je potrebné splniť nasledujúcu podmienku, zrýchlenie robota vo vertikálnom smere musí byť väčšie ako je gravitačné zrýchlenie.

Vychádzame z rovnice pre dráhu. Ako je uvedené v kapitole 3.2.1.

Pri maximálnom možnom zdvihu robota 20 cm dostaneme na základe rovnice 3-2 čas potrebný na vykonanie zdvihu. Dosadením týchto hodnôt do vzorca 2-8 dostávame:

$$t = \sqrt{\left(\frac{2 \cdot 0,20}{9,81}\right)} \quad (3-2)$$

Na základe vložených parametrov dostaneme čas 0,202 s. V tomto čase príde k strate kontaktu nohy robota s podložkou. Aby robot reálne vyskočil do vzduchu, musí byť tento čas kratší.

Vychádzajúc z rovnice

$$s = \frac{1}{2} \cdot a \cdot t^2 \quad (3-3)$$

pre výšku skoku platí

$$s = \frac{1}{2} \cdot (a_r - g) \cdot t^2 \quad (3-4)$$

$a_r$  zrýchlenie robota vo vertikálnom smere [ $\text{ms}^{-2}$ ]

$g$  gravitačné zrýchlenie [ $\text{ms}^{-2}$ ]

Táto dráha odpovedá zmene polohy ťažiska robota počas prvotnej fázy skoku.

$$\Delta y = y_{max} - y_{min} \quad (3-5)$$

$\Delta y$  zmena polohy ťažiska robota v Y-ovej osi [m]

$y_{min}$  dolná, východzia poloha ťažiska robota v Y-ovej osi [m]

$y_{max}$  horná, koncová poloha ťažiska robota v Y-ovej osi [m]

Z rovníc si dokážeme spočítať dráhu pohybu tak ako aj veľkosť potrebného zrýchlenia.

Následne sme schopní z času potrebného na dosiahnutie skoku určiť maximálny počet krokov iterácií výpočtu pre riadiaci signál jednotlivých kĺbov robota.

$$n = \frac{t}{t_{sim}} \quad (3-6)$$

$n$  počet krokov iterácie [-]

$t$  čas požadovaný na vykonanie akcie [s]

$t_{sim}$  čas vzorkovania signálu – 0,001 sekundy

Najjednoduchší spôsob generovania riadiacich signálov pre pohyb kĺbov je postavený na rovnomernom rozdelení zmeny veľkosti uhla ramena na časti zodpovedajúce počtu krokov iterácie.

$$\alpha_{step} = \frac{\Delta\alpha}{n} \quad (3-7)$$

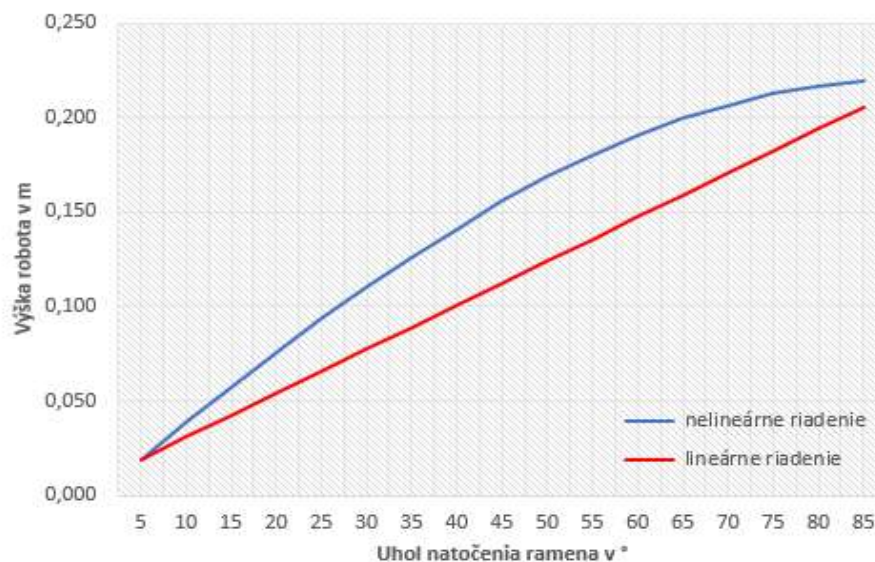
$\alpha_{step}$  krok iterácie [°]

$\Delta\alpha$  zmena uhla ramena [°]

$n$  počet krokov iterácie [-]

Z vypočítaných hodnôt sme v cykle algoritmu postupne menili uhol natočenia ramena z východzej polohy až do požadovanej koncovej hodnoty.

Na grafe na Obr. 27 si môžeme všimnúť nelineárny priebeh zmeny polohy ťažiska v Y-ovom smere. Táto nelinearita súvisí s uhlom natočenia. Venovali sme sa jej v kapitole 2.3.5.1. V tomto prípade sa zrýchlenie v záverečnej fáze pohybu blíži k nule.



Obr. 27 Porovnanie aktuálnej výšky robota v závislosti na uhle natočenia

Ak by sme požadovali konštantné zrýchlenie počas celého pohybu, v cykle pre výpočet natočenia ramena by sa uhol natočenia počítal na základe konštantnej zmeny výšky robota. Porovnanie aktuálnej polohy ťažiska robota v závislosti na uhle natočenia pre oba spôsoby riadenia je znázornený na Obr. 27.

Simuláciu bežala v režime lineárneho nárastu uhlu natočenia ramien. V tomto prípade sme simulovali sériu 4 drepov. Vo všetkých prípadoch robot vykonáva drep v tom istom rozsahu uhlov  $10 - 60^\circ$ . Mení sa však rýchlosť drepu - robot postupne zrýchľuje. Priebiehy sledovaných veličín sú súčasťou príloh na Obr. 34 – Obr. 36.

Aj v tomto prípade vidíme, že systém sleduje požadovanú polohu s oneskorením, spôsobeným PID regulátorom.

Uhlová rýchlosť pohybu je závislá od počiatočnej a koncovej polohy postavenia robota ako aj času potrebného na prekonanie požadovanej dráhy. Pretože je pohyb riadený lineárnou zmenou uhla ramena voči podložke, je uhlová rýchlosť konštantná. V prípade zmeny rýchlosti, zrýchlení alebo spomalení, sú na grafoch viditeľné krátke špičky. V čase, keď sa rameno nepohybuje, klesá uhlová rýchlosť na nulu.

Zatiaľ čo priebeh rýchlosti je vo všetkých kĺboch veľmi podobný, iná situácia je v prípade momentov. Na grafoch môžeme sledovať zmenu momentu v závislosti od aktuálnej polohy ramena. Podobne ako v prípade rýchlosti sa i tu objavujú špičky. Najhoršie situácia s „rušivým signálom“ je v prípade kolenného kĺbu. Naopak, najlepšie je na tom bedrový kĺb. Vzhľadom na navrhnutú koncepciu robota sa telo nachádza nad nulovým bodom zostavy, a preto sa moment, ktorý generuje, blíži k nule. Ale aj v tomto prípade sa objavujú špičky. Ich úroveň je rádovo nižšia.

### 3.2.9 Skok na mieste

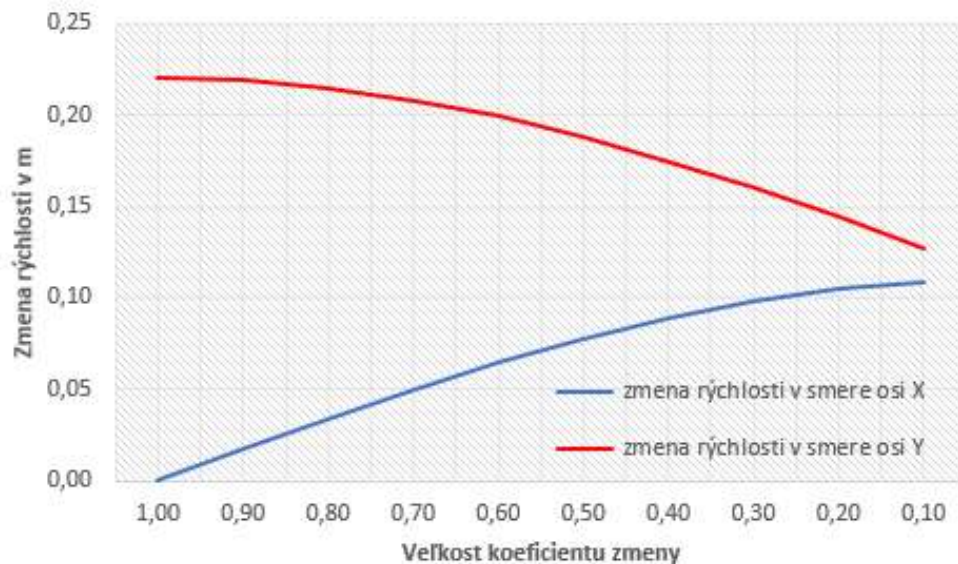
V ďalšom kroku sme simulovali skok na mieste. Robot postupne vykoná sériu skokov na mieste. Výška skoku sa postupne zvyšuje. Priebiehy simulácie sú zachytené na grafoch Obr. 37 – Obr. 39.

Požadovanú polohu systém sleduje bez problémov. Prvú zmenu vidíme v prípade rýchlosti. Podmienke skoku sme sa venovali v predošlej kapitole. Aby robot vyskočil, musí dosiahnuť požadované zrýchlenie. To dosiahne zmenou polohy ramena za požadovaný čas. Tejto zmene zodpovedá uhlová rýchlosť. Preto na grafe vidíme 4 krátke, úseky s uhlovou rýchlosťou rádovo desiatky rad/s.

V prípade momentov sa situácia komplikuje. Impulzy nadobúdajú výrazne väčšie hodnoty. V kolennom kĺbe môžeme sledovať nárast intenzity, spojený s výškou skoku. Ako vyplýva z grafu Obr. 38, čím je skok vyšší, tým sú rušivé signály intenzívnejšie a je ich viac. S týmto rušivým javom sa stretávajú aj autori článku [20]. Tento problém je možné riešiť nastavením tlmenia v jednotlivých kĺboch ako aj parametrov kontaktu chodidla robota s povrchom. Dôležitú úlohu potlačení vzniknutých rušivých momentov samozrejme zohráva aj kvalitný PID regulátor.

### 3.2.10 Skok do diaľky

Fyzikálnej podstate skoku sa venujeme v kapitole 2.3.3 - 2.3.5. Parametre skoku dokážeme riadiť veľkosťou X a Y zložky vektora zrýchlenia. Problematike sa venujeme v kapitole 2.3.5.1. Vychádzajúc z týchto rovníc v uvedených kapitolách vieme zmenou rýchlosti ramien ovládať smer skoku.



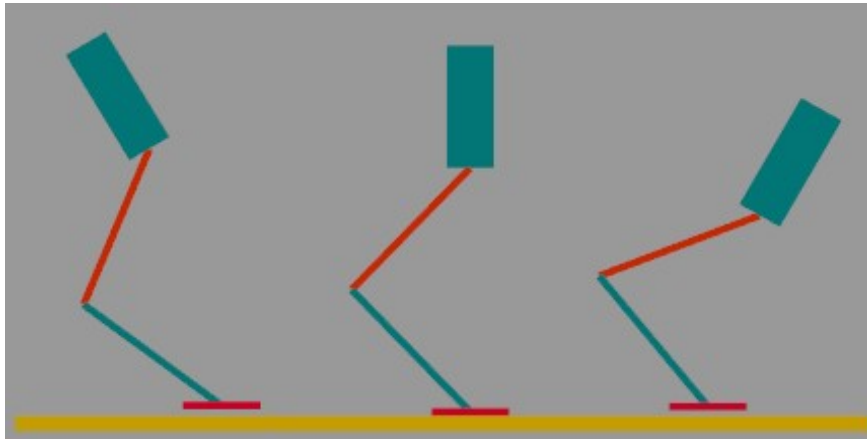
Obr. 28 Závislosť X-ovej a Y-ovej zložky vektora rýchlosti na pomere uhlov

Matematicky vieme presne určiť výšku a dĺžku skoku. Pre naše účely sme si však zvolili jednoduchšie riešenia. Ako vyplýva z kapitoly 2.3.5, smer a výška skoku závisia od výsledného vektora rýchlosti. Preto sme sa rozhodli riadiť skok pomocou zmeny pomeru rýchlosti pohybu jednotlivých ramien, ako bolo vysvetlené v kapitole 2.3.5.1. Opis realizácie riadenia v tomto pohybu je súčasťou kapitoly 3.2.4. Pomocou premenných *left* a *right* tak definujeme smer pohybu robota. Na grafe Obr. 28 je znázornený pomer X -ovej a Y-ovej zložky vektora v závislosti od pomeru zmeny rýchlosti pohybu ramien. Ak sa obe ramená pohybujú rovnakou rýchlosťou, robot robí drep, resp. dokáže vyskočiť iba vo vertikálnom smere. Jedná sa o skok na mieste. Ak ponecháme jednu z hodnôt na úrovni 1 a druhú budeme meniť, príde v závislosti od veľkosti zmeny druhej hodnoty k skoku v smere osy X.

V dôsledku zmeny vektora rýchlosti následne príde k vychýleniu ťažiska a tým pádom nakloneniu robota do smeru skoku. V letovej fáze je preto potrebné toto vychýlenie kompenzovať, aby po dosadnutí robota na zem neprišlo k jeho prevráteniu.

Náklon robota v počiatkovej fáze letu pre skok jedným alebo druhým smerom je znázornený na Obr. 29

Priebeh sledovaných veličín nájdeme v prílohe na Obr. 40 – Obr. 42.



Obr. 29 Zmena postavenia robota v závislosti od požadovaného smeru skoku

Simulovaný scenár pozostáva z 3 skokov. Robot skočí doľava, následne vyskočí na mieste a ďalším skokom sa vráti na pôvodné miesto.

V tomto scenári je zaujímavé sledovať priebehy momentov. K výraznej zmene momentu prichádza už v členkovom kĺbe.

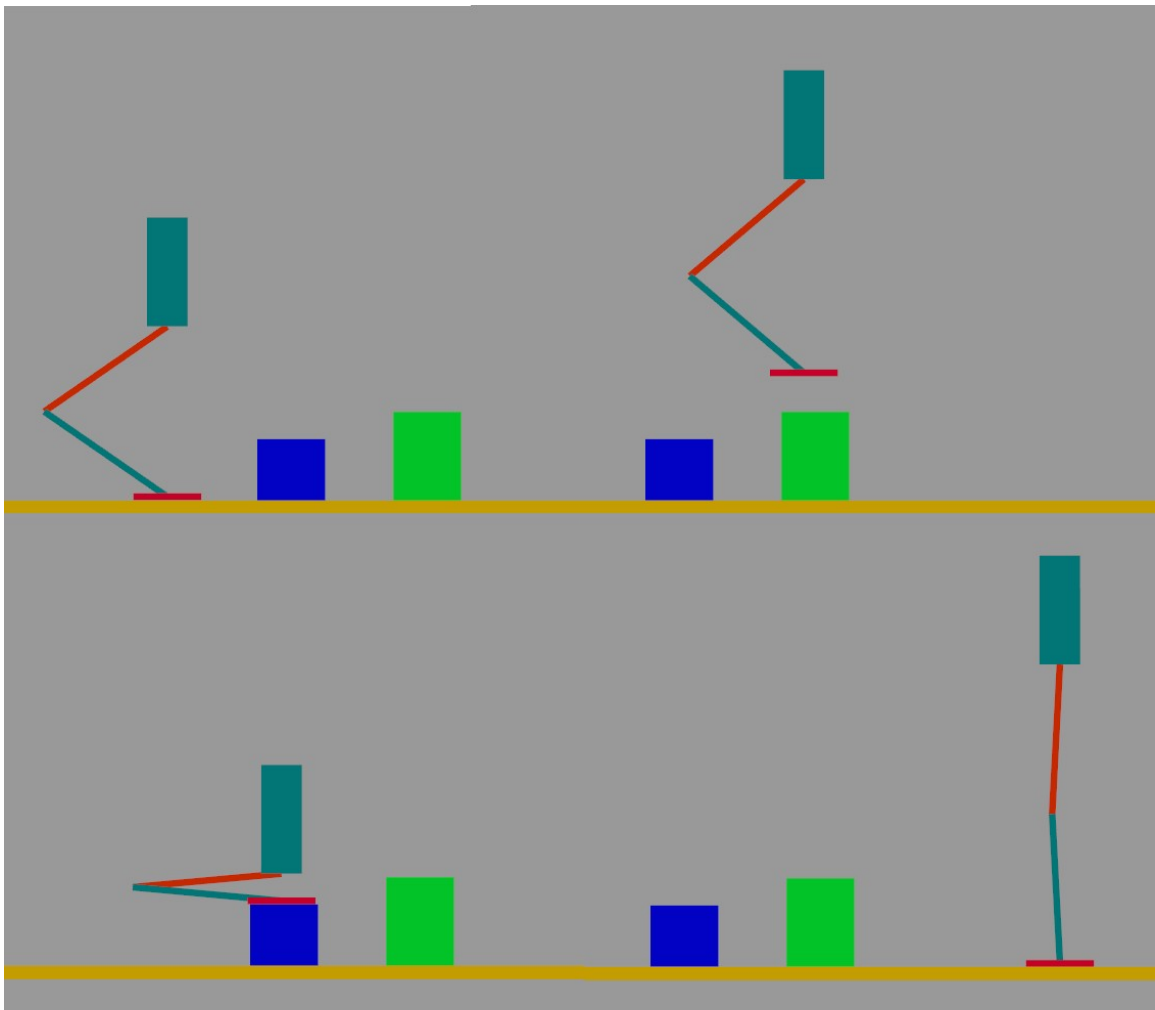
V prípade skoku na mieste pozorujeme minimálne rušenie popísané už v predošlej kapitole. Ak robot skáče do strán objaví sa tu intenzívne rušenie. I napriek tom, že sa robot vráti na pôvodne miesto, priebeh momentov pre skok doľava a doprava je odlišný. Tento rozdiel vychádza z nesymetrie robota. Intenzívne rušenie pozorujeme pri všetkých troch skokoch aj v kolennom kĺbe. Priebeh momentov k bedrovom kĺbe kopíruje priebeh momentov v členkovom kĺbe.

### 3.2.11 Exhibícia

Na záver sme vytvoril krátku choreografiu, ktorá zahŕňa množinu všetkých opísaných aktivít robota. Robot tu prezentuje pohyb v prostredí. Predvádza sériu drepov, skok doprava, doľava spojený so skokmi na a cez prekážku.

Po zostavení robot prejde do drepu. Následne sa vzpriami a vracia sa naspäť do drepu. Nasleduje poskok na mieste. Po krátkej pauze robot skočí na zelený hranol, kde zotrúva vo východzej polohe. Nasleduje prechod do drepu, nasledovaný skokom robota na modrú kocku. Ďalším skokom sa robot vracia na zelený hranol. Prichádza posledný skok, po ktorom robot skončí vo vzpriamenej polohe vpravo od hranola.

Priebehy sledovaných signálov sú k dispozícii na obrázkoch Obr. 43 – Obr. 45 prílohy.



Obr. 30 Prezentácia pohybu robota počas exhibície

### 3.2.12 PID regulátor

V prípade pohybu a skokov sme upozornili na prechodové javy v dobe zmeny polohy ramena. Úroveň rušivých signálov momentu dosahujú vysoké hodnoty a na ich potlačenie je potrebný kvalitný PID regulátor.

Pri návrhu kvalitného regulátora je potrebné brať do úvahy množstvo premenných. Minimálne je potrebné sa sústrediť na:

- Konštrukciu robota
  - rozmery
  - hmotnosť
- Aktivity robota
  - typ akcie
  - rýchlosť

Návrh kvalitného PID regulátora svojou náplňou presahuje rozsah tejto práce a ani nebol jej náplňou.

---

## 4 ZÁVER

Obsah bakalárskej práce postupne naplňuje jednotlivé ciele vyplývajúce zo zadania. Hlavným cieľom bolo vytvoriť simuláciu robota a vytvoriť preňho algoritmus priestorového pohybu, ktorým sú v našom prevedení skoky.

Pri rešerši sme sa primárne stretli s robotmi, ktoré sa pohybovali jedným smerom na rovnom povrchu. Autori sa venujú analýze kinematiky a dynamiky pohybu, pričom sa sústreďujú na jeden konkrétny typ pohybu a problematiky, ktorá ho obklopuje. Z týchto poznatkov sme vychádzali pri našom návrhu.

V práci sme sa zamerali na návrh jednonohého skákajúceho robota schopného uniesť veľký náklad v porovnaní s váhou jeho konštrukcie. Predstavili sme 3 alternatívy prístupu k jeho konštrukcii. V procese návrhu a simulácie sme sa sústredili na zjednodušený model robota s dvomi ramenami rovnakej dĺžky.

Nami navrhnutý robot dokáže realizovať aktivity, ktoré sme si položili za cieľ na začiatku práce. Simuláciou sme overili:

- stoj na mieste,
- drep,
- skok na mieste,
- skok do diaľky,
- skok cez prekážky.

Kombináciou jednotlivých aktivít sme zostavili záverečnú exhibíciu, ktorá spája všetky aktivity do ucelenej zostavy.

Simuláciu sme realizovali v prostredí Simulink, verzia R2022b. Navrhnutý robot je schopný v 3D priestore zaujať stabilný postoj, dokáže robiť drepy ako aj skoky na mieste. Zároveň sa dokáže pohybovať dvomi smermi a zdolávať jednoduché prekážky. Stabilita robota počas pohybu je korigovaná pomocou PID regulátora.

Okrem zostavenia mobilného robota bolo cieľom práce sledovanie momentov a rýchlostí v kĺboch robota. Konceptia návrhu spočíva na dosiahnutí nulového momentu. Detailné výsledky sú vykreslené v grafoch, ktoré sú súčasťou prílohy 7.1 - 7.6 a boli popísané v kapitolách 3.2.6 – 3.2.11.

Tieto výsledky sme dosiahli pri vzorkovaní 0,001 sekundy. Predpokladáme, že vyššia frekvencia vzorkovania by mala pozitívny dopad na správanie sa PID regulátora a na základe toho by sme dosiahli konzistentnejšie priebehy sledovaných veličín.

Tak isto by malo mať pozitívny dopad na priebeh simulácie zavedenie reálnych parametrov pre jednotlivé komponenty. Tým myslíme napr. tlmenie v kĺboch, vzájomnú interakciu telies, reálne parametre materiálov atď.

Zároveň sme identifikovali množstvo oblastí, na ktoré sa treba zamerať v ďalšom štádiu skúmania, vývoja, simulácií tohto typu robotov, a to napríklad:

- vplyv vzorkovania signálu na priebeh simulácie,
- nesymetrické správanie pri skoku doprava verzus doľava,
- pohyb v nerovnom teréne a odhad/plánovanie trajektórie,
- vplyv typu zrýchlenia...

V porovnaní s našimi řešenými jsme si zvolili širší záběr řešení problematiky, aby jsme pokryli širší spektrum možností pohybu robota. Z toho důvodu jsme nemohli proniknout do takové hloubky problematiky, jako autoři článků. Víaceré aspekty návrhu robota jsou jen náznaky toho, na čem se třeba soustředí při dlouhodobém vývoji.

## 5 ZOZNAM POUŽITÝCH ZDROJOV

1. E. Hashem, (PDF) *Humanoid Arm Robot Simulation on SIMULINK MATLAB - Researchgate*, [Online]. [cit. 2023-05-21]. Dostupné z: [https://www.researchgate.net/publication/350089291\\_Humanoid\\_arm\\_robot\\_simulation\\_on\\_SIMULINK\\_MATLAB](https://www.researchgate.net/publication/350089291_Humanoid_arm_robot_simulation_on_SIMULINK_MATLAB)
2. “*Planar one-leg hopper (1980-1982)*,” 2D One-Leg Hopper., [Online]. [cit. 2023-05-21]. Dostupné z: [http://www.ai.mit.edu/projects/leglab/robots/2D\\_hopper/2D\\_hopper.html](http://www.ai.mit.edu/projects/leglab/robots/2D_hopper/2D_hopper.html)
3. “*Uniroo (1991-1993)*,” MIT Leg Lab’s Uniroo Robot, [Online]. [cit. 2023-05-21]. Dostupné z: <http://www.ai.mit.edu/projects/leglab/robots/uniroo/uniroo.html>
4. A. Malewar, “*Scientists invent leaping robot for search and rescue during disasters*,” Tech Explorist, 11-Dec-2016. [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.techexplorist.com/scientists-invented-leaping-robot-used-search-rescue-during-disasters/4364/>
5. E. Ackerman, “*Salto-1P is the most amazing jumping robot we’ve ever seen*,” IEEE Spectrum, 29-Jun-2017. [Online]. [cit. 2023-05-21]. Dostupné z: <https://spectrum.ieee.org/salto1p-is-the-most-amazing-jumping-robot-weve-ever-seen>
6. “*3D one-leg Hopper (1983-1984)*,” 3D One-Leg Hopper, [Online]. [cit. 2023-05-21]. Dostupné z: [http://www.ai.mit.edu/projects/leglab/robots/3D\\_hopper/3D\\_hopper.html](http://www.ai.mit.edu/projects/leglab/robots/3D_hopper/3D_hopper.html)
7. M. H. Raibert, “*Hopping in legged systems — modeling and simulation for the two-dimensional one-legged case*,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-14, no. 3, pp. 451–463, 1984. doi:10.1109/tsmc.1984.6313238. [Online]. [cit. 2023-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/6313238>
8. G. Ribak, “*Insect-inspired jumping robots: Challenges and solutions to jump stability*,” Current Opinion in Insect Science, vol. 42, pp. 32–38, 2020. doi:10.1016/j.cois.2020.09.001. [Online]. [cit. 2023-05-21]. Dostupné z: <https://doi.org/10.1016/j.cois.2020.09.001>
9. D. Tian, J. Gao, X. Shi, Y. Lu, and C. Liu, “*Vertical jumping for legged robot based on quadratic programming*,” Sensors, vol. 21, no. 11, p. 3679, 2021. doi:10.3390/s21113679. [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.mdpi.com/1424-8220/21/11/3679>
10. D. Tian, J. Gao, C. Liu, and X. Shi, “*Simulation of upward jump control for one-legged robot based on QP Optimization*,” Sensors, vol. 21, no. 5, p. 1893, 2021. [Online]. [cit. 2023-05-21]. Dostupné z: <https://doi.org/10.3390/s21051893>
11. X. Yu, C. Fu, and K. Chen, “*Modeling and control of a single-legged robot*,” Procedia Engineering, vol. 24, pp. 788–792, 2011. doi:10.1016/j.proeng.2011.11.2738 [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S1877705811055901?via%3Dihub>
12. M. Miwa, H. Sakane, K. Nagase, Y. Koshimoto, and S. Tuchitani, “*Study on one-legged robot jumping*,” Service Robotics and Mechatronics, pp. 119–124, 2010. doi:10.1007/978-1-84882-694-6\_21 [Online]. [cit. 2023-05-21]. Dostupné z: [https://link.springer.com/chapter/10.1007/978-1-84882-694-6\\_21](https://link.springer.com/chapter/10.1007/978-1-84882-694-6_21)
13. G. Chen, J. Tu, X. Ti, and H. Hu, “*A single-legged robot inspired by the jumping mechanism of click beetles and its hopping dynamics analysis*,” Journal of

- Bionic Engineering, vol. 17, no. 6, pp. 1109–1125, 2020. doi:10.1007/s42235-020-0099-z, 21 [Online]. [cit. 2023-05-21]. Dostupné z: <https://link.springer.com/article/10.1007/s42235-020-0099-z>
14. D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing, “*Robotic vertical jumping agility via series-elastic power modulation*,” *Science Robotics*, vol. 1, no. 1, 2016. doi:10.1126/scirobotics.aag2048, [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.science.org/doi/10.1126/scirobotics.aag2048>
  15. “*Spot® - The Agile Mobile Robot*,” Boston Dynamics, [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.bostondynamics.com/products/spot>
  16. “*Atlas™*,” Boston Dynamics, [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.bostondynamics.com/atlas>
  17. Y. Aoustin and A. M. Formalskii, “*Modeling, control and simulation of upward jump of a biped*,” *Multibody System Dynamics*, vol. 29, no. 4, pp. 425–445, 2012. doi:10.1007/s11044-012-9319-6, [Online]. [cit. 2023-05-21]. Dostupné z: [https://www.researchgate.net/publication/281156486\\_Modeling\\_Control\\_and\\_Simulation\\_of\\_Upward\\_Jump\\_of\\_a\\_Biped](https://www.researchgate.net/publication/281156486_Modeling_Control_and_Simulation_of_Upward_Jump_of_a_Biped)
  18. Y. AOUSTIN and A. FORMALSKII, “*Upward jump of a biped*,” *International Journal of Humanoid Robotics*, vol. 10, no. 04, p. 1350032, 2013. doi:10.1142/s0219843613500321, [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.worldscientific.com/doi/abs/10.1142/S0219843613500321>
  19. H. Chaoui, S. Miah, M. R. Kafi, and B. Hamane, “*Neural network balance control of hopping robots in flight phase under unknown dynamics*,” 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT), 2015, [Online]. [cit. 2023-05-21]. Dostupné z: <https://ieeexplore.ieee.org/document/7232996/>
  20. M. Chen, Q. Li, S. Wang, K. Zhang, H. Chen, and Y. Zhang, “*Single-leg structural design and foot trajectory planning for a novel bioinspired quadruped robot*,” *Complexity*, vol. 2021, pp. 1–17, 2021. doi:10.1155/2021/6627043, [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.sciencegate.app/document/10.1155/2021/6627043>
  21. “*Documentation*,” Documentation - MATLAB & Simulink, [https://www.mathworks.com/help/index.html?s\\_tid=CRUX\\_lftnav](https://www.mathworks.com/help/index.html?s_tid=CRUX_lftnav)
  22. C. H. Llanos, D. Munoz, and S. A. Pertuz Mendez, “*Simulation and implementation of impedance control in Robotic hand*,” Proceedings of the 24th ABCM International Congress of Mechanical Engineering, 2017. doi:10.26678/abcm.cobem2017.cob17-2327, [Online]. [cit. 2023-05-21]. Dostupné z: [https://www.researchgate.net/publication/324363084\\_Simulation\\_and\\_Implementation\\_of\\_Impedance\\_Control\\_in\\_Robotic\\_Hand](https://www.researchgate.net/publication/324363084_Simulation_and_Implementation_of_Impedance_Control_in_Robotic_Hand)
  23. “*World frame*,” Fixed spatial relationship between frames - MATLAB. [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.mathworks.com/help/sm/ref/rigidtransform.html>
  24. “*Variable brick solid*,” Solid brick element with geometry, inertia, and color - MATLAB. [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.mathworks.com/help/sm/ref/bricksolid.html>
  25. “*Spatial Contact Force*” Model contact between two geometries - MATLAB. [Online]. [cit. 2023-05-21]. Dostupné z: <https://www.mathworks.com/help/sm/ref/spatialcontactforce.html>

26. “*Prismatic joint*,” Joint that allows relative motion along single axis - MATLAB. [Online]. [cit. 2023-05-21]. Dostupné z:  
<https://www.mathworks.com/help/sm/ref/prismaticjoint.html>
27. “*Revolute joint*,” Joint with one revolute primitive - MATLAB. [Online]. [cit. 2023-05-21]. Dostupné z:  
<https://www.mathworks.com/help/sm/ref/revolutejoint.html>

---

## 6 ZOZNAM POUŽITÝCH SKRATIEK

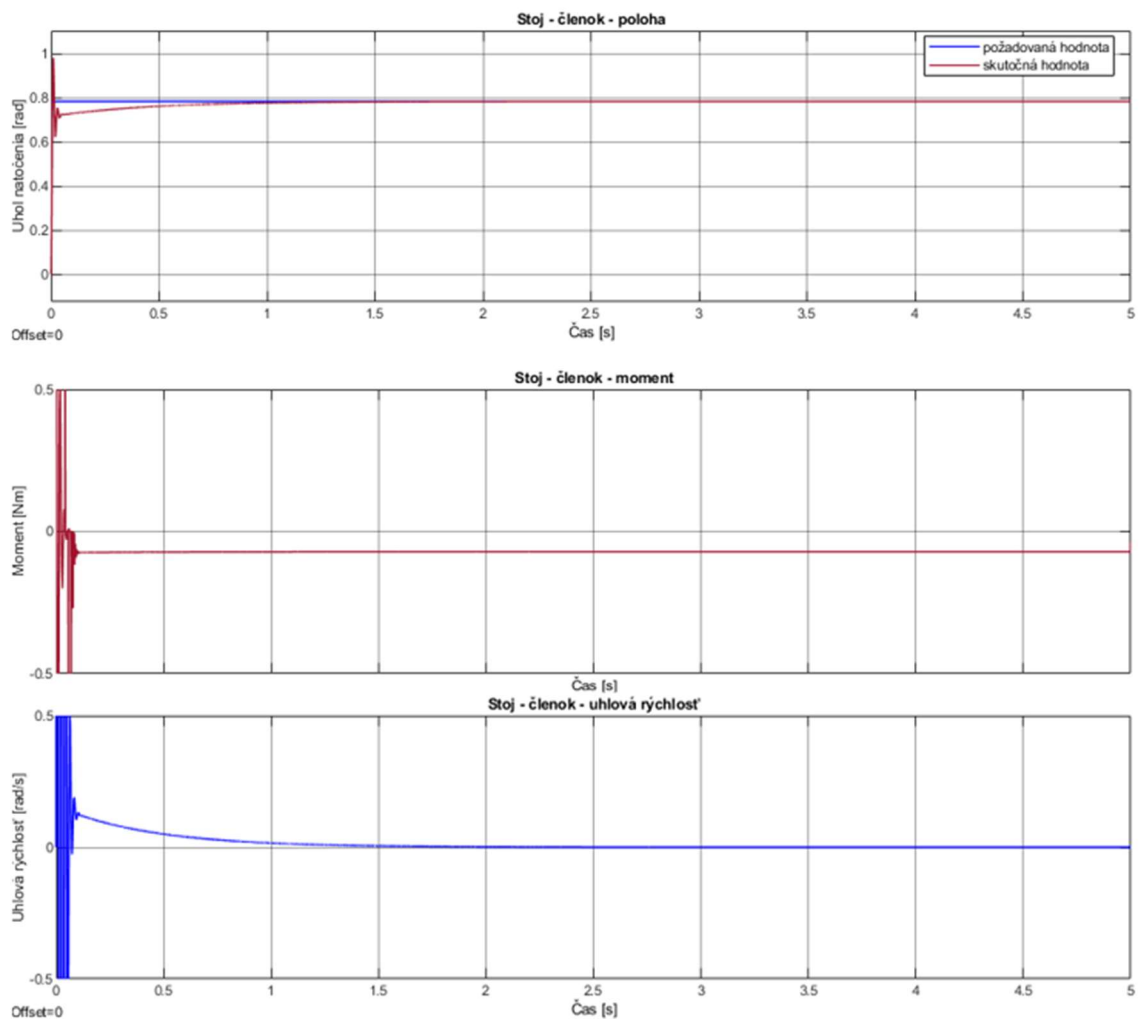
cm	centimeter, odvodená jednotka dĺžky
Hz	Hertz, jednotka frekvencie
kg	kilogram, jednotka hmotnosti SI sústavy
m	meter, jednotka dĺžky SI sústavy
ms	milisekunda, jednotka času
$ms^{-1}$	meter za sekundu, rýchlosť
$ms^{-2}$	meter za sekundu, zrýchlenie
N	Newton, jednotka sily
Nm	Newton-meter, jednotka momentu
rad	radián, uhlová jednotka – rovinný uhol
s	sekunda, jednotka času SI sústavy
°	stupeň, uhlová jednotka – rovinný uhol

## 7 PRÍLOHY

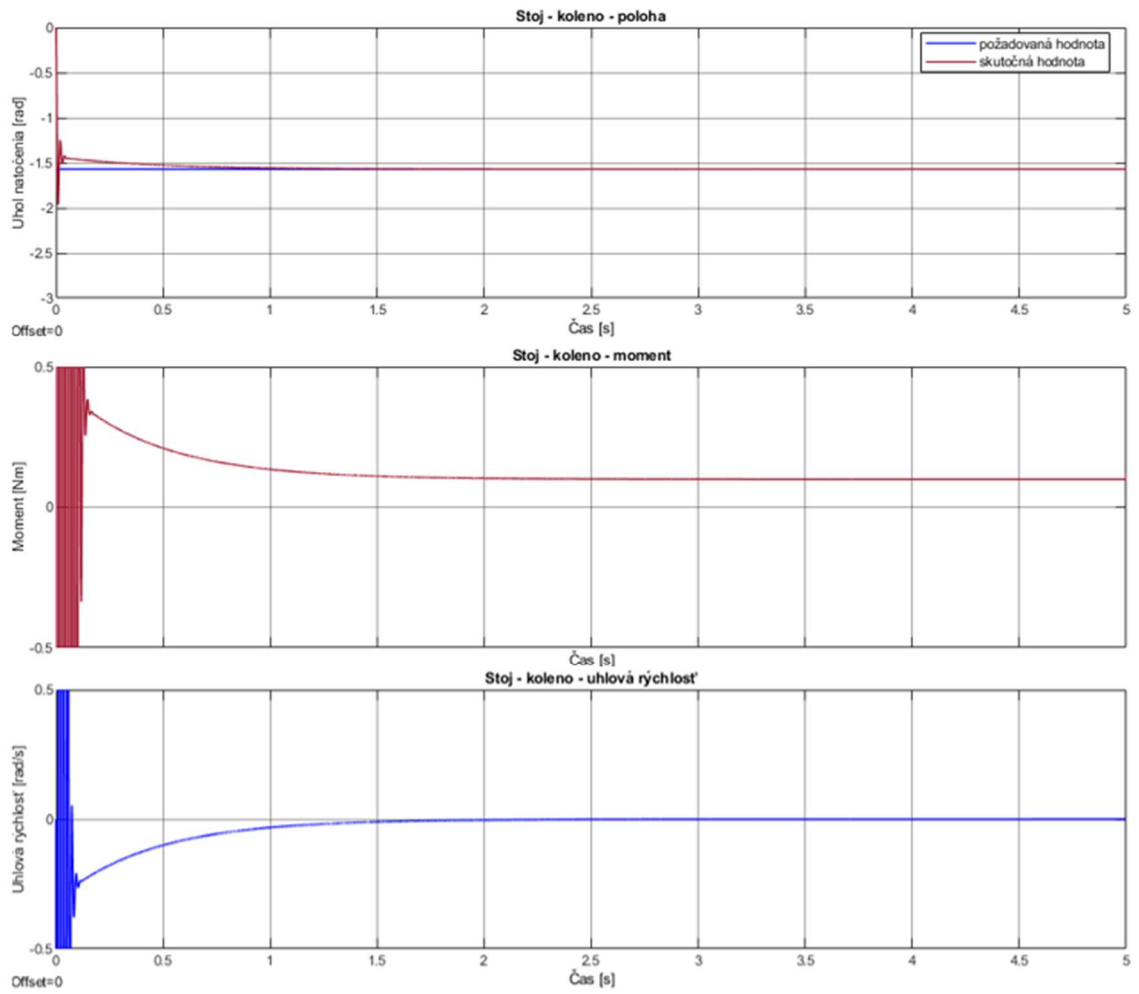
### 7.1 Zoznam elektronických príloh

- *robot\_definicia.m* – definovanie parametrov na zostavenie robota
- *robot\_stoj.m* – základný stoj robota v definovanej pozícii,
- *robot\_stoj.avi* – video - základný stoj robota v definovanej pozícii,
- *robot\_drep.m* – prezentácia drepu,
- *robot\_drep.avi* – video - prezentácia drepu,
- *robot\_skok\_miesto.m* – skok na mieste,
- *robot\_skok\_miesto.avi* – video - skok na mieste,
- *robot\_skok\_strana.m* – skok do strán,
- *robot\_skok\_strana.avi* – video - skok do strán,
- *robot\_exhibicia.m* – prezentácia zostavy cvikov,
- *robot\_exhibicia.avi* – video - prezentácia zostavy cvikov,
- *robot\_prezentacia.slx* – simulácia zostavenia robota na základe vstupných požiadaviek, prezentujúceho zostavu preddefinovaných cvikov.

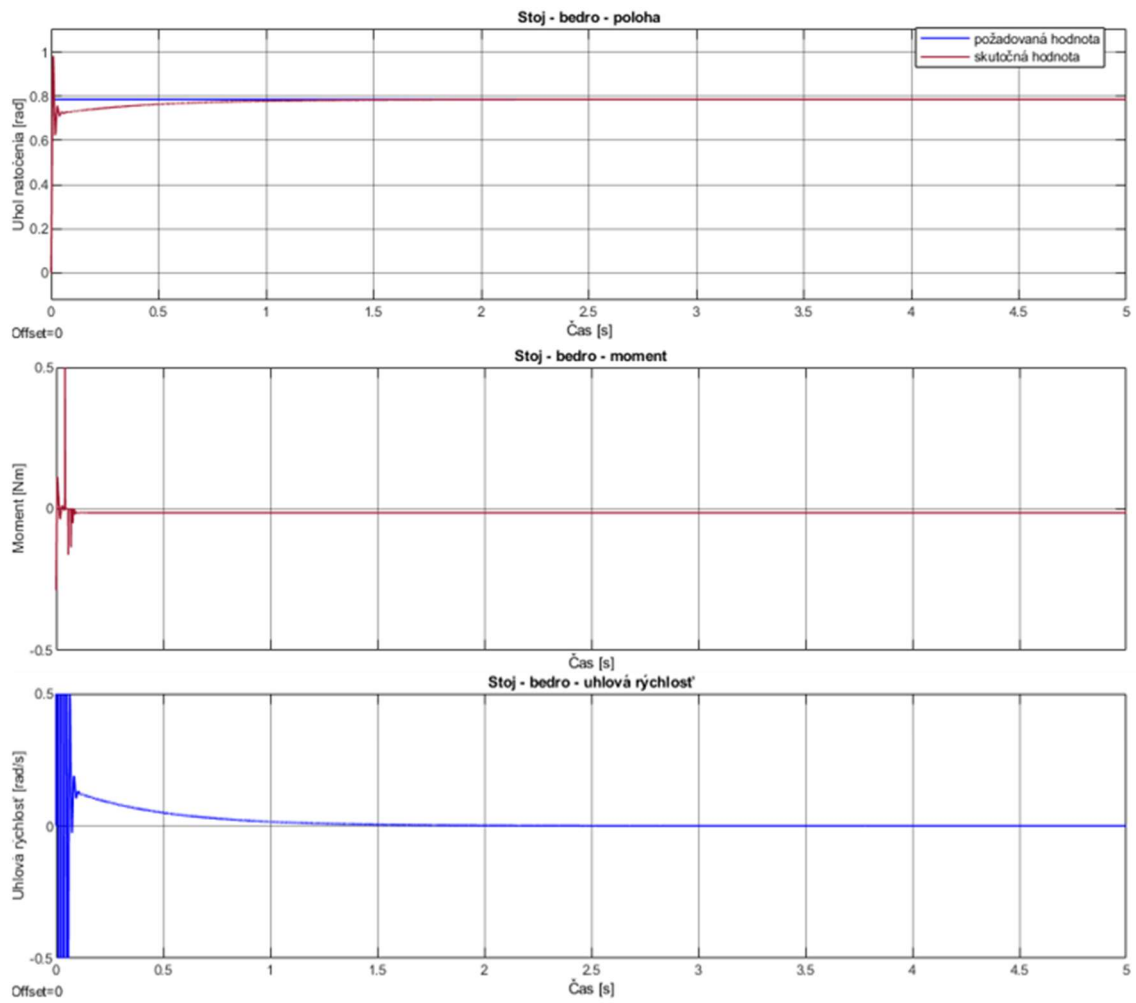
## 7.2 Stoj



Obr. 31 Stoj - členok

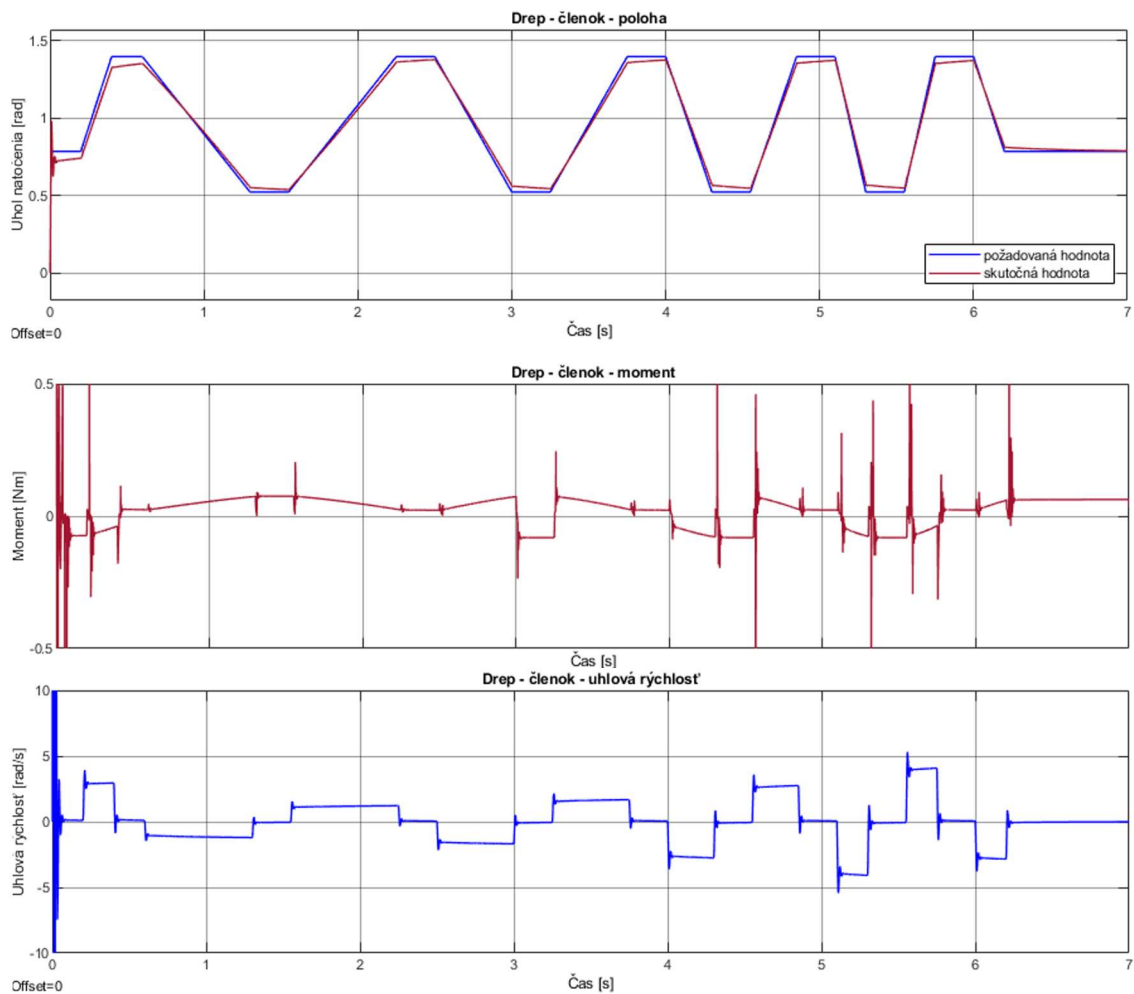


Obr. 32 Stoj - koleno

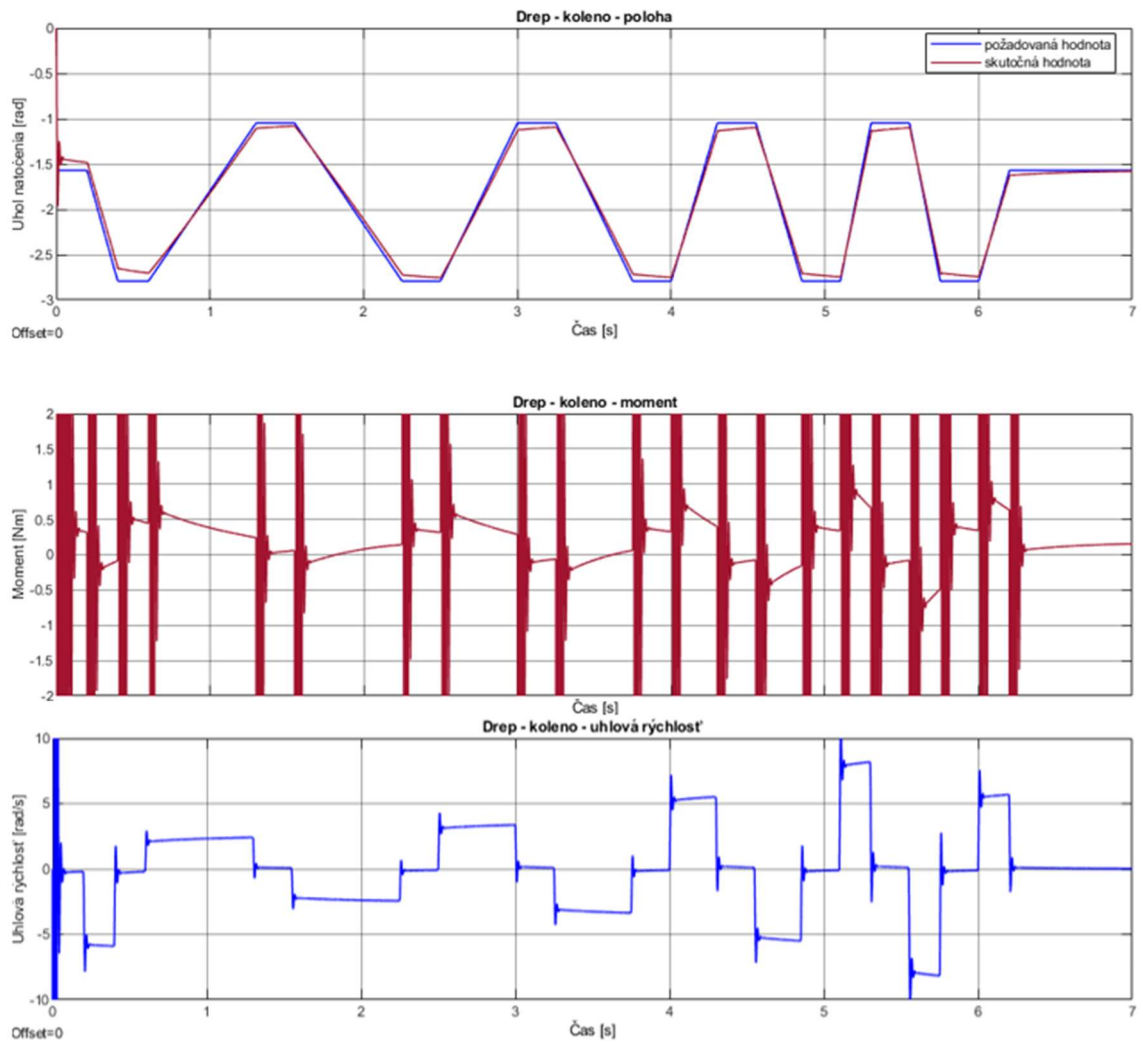


Obr. 33 Stoj - bedro

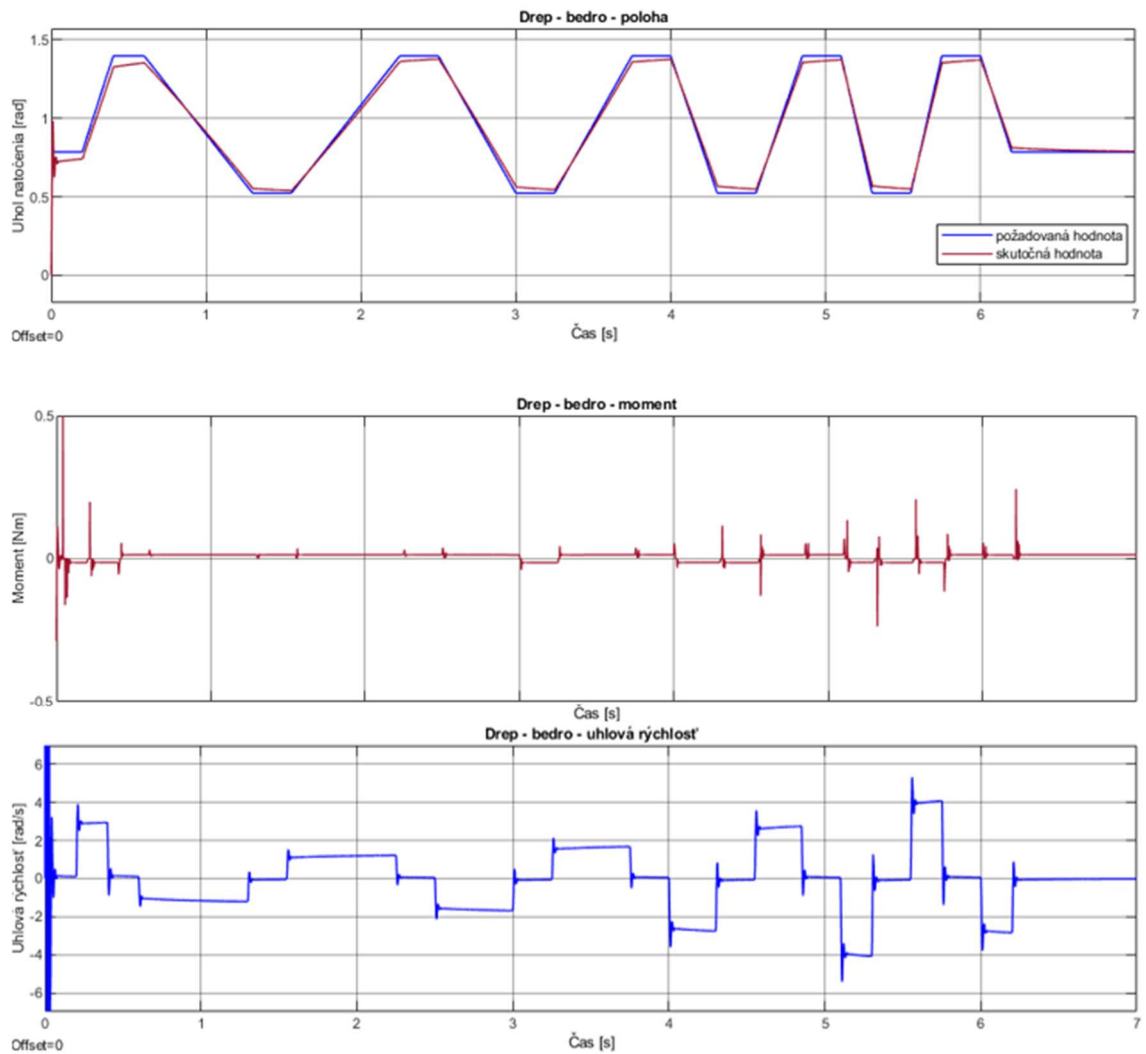
### 7.3 Drep



Obr. 34 Drep - členok

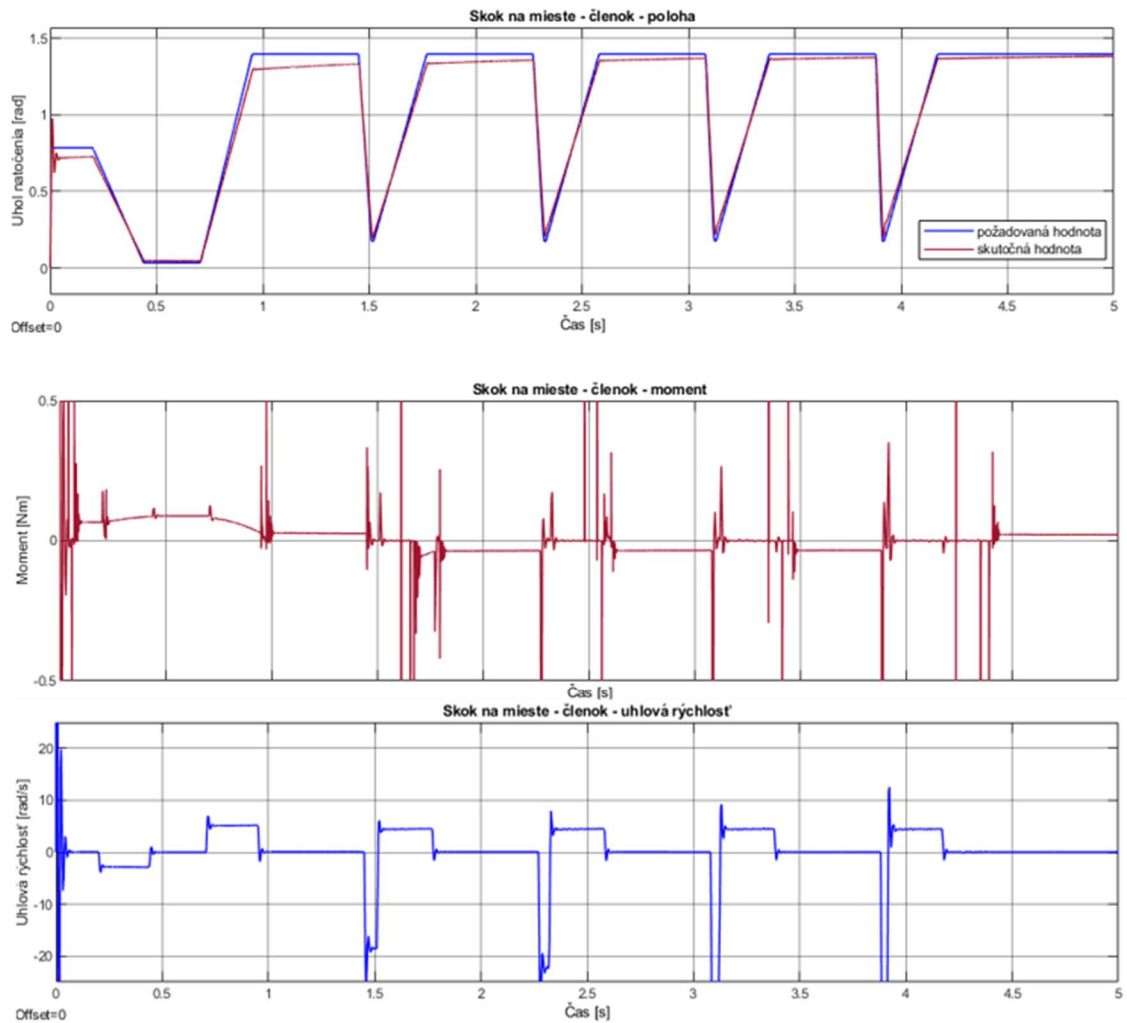


Obr. 35 Drep - koleno

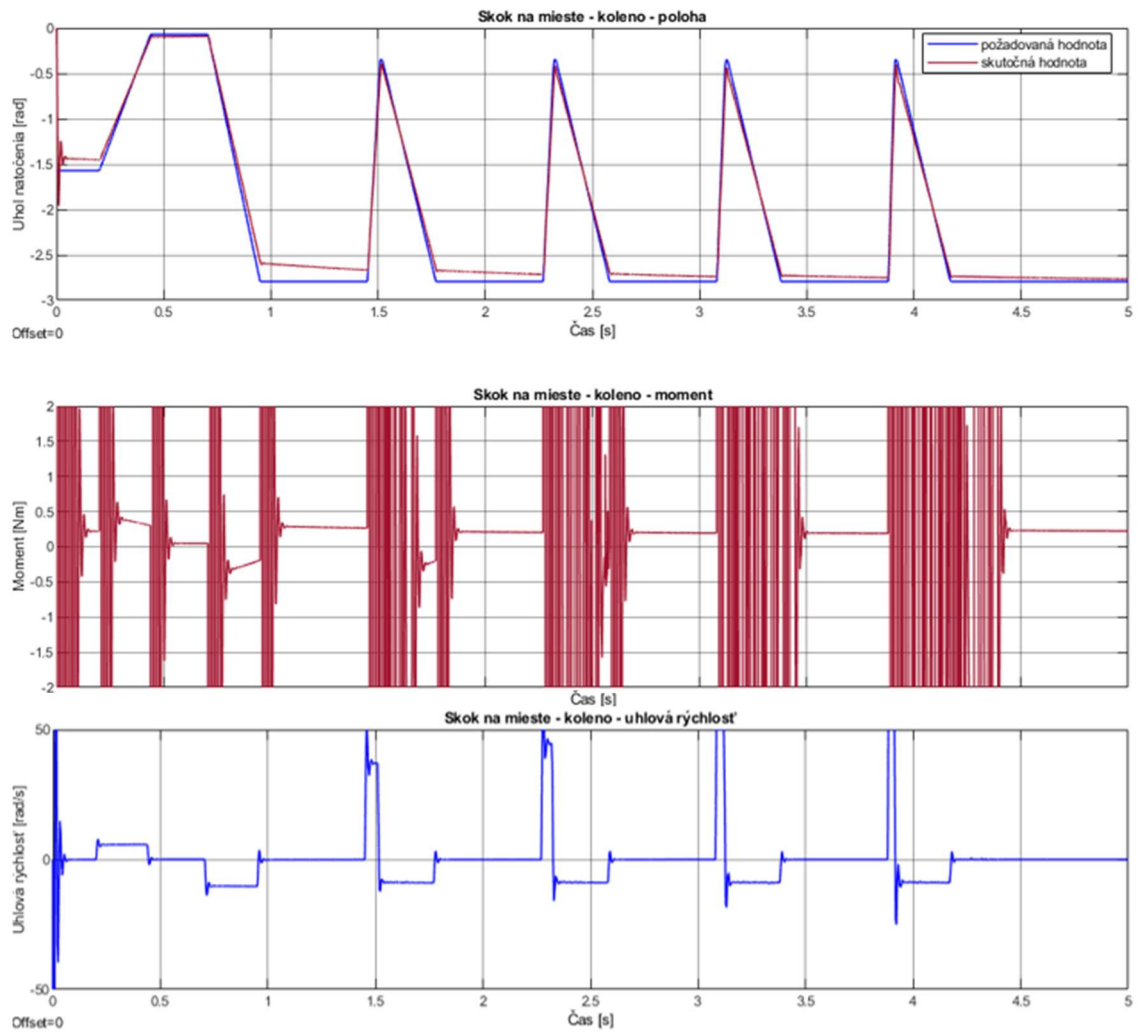


Obr. 36 Drep - bedro

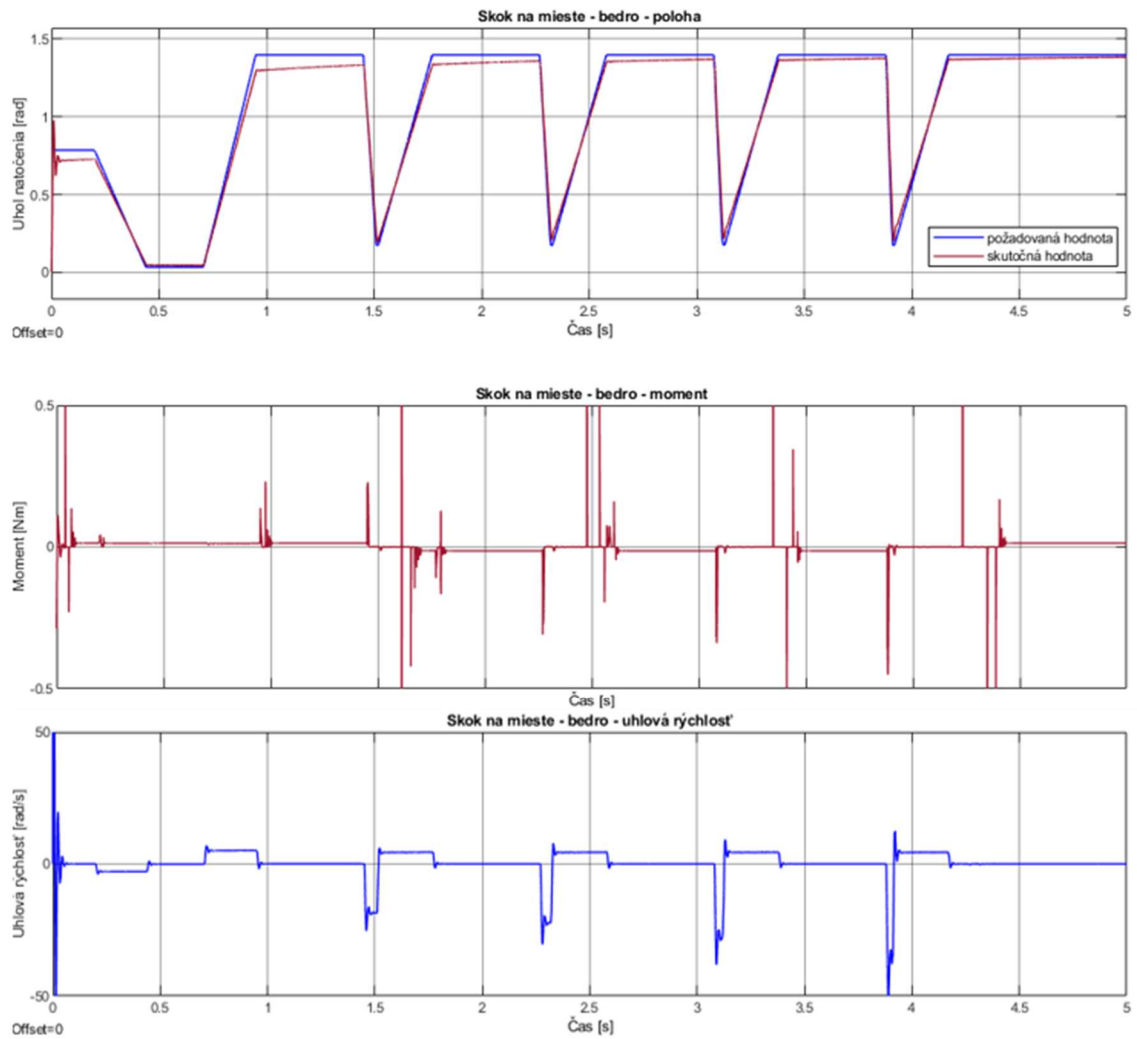
## 7.4 Skok na mieste



Obr. 37 Skok na mieste - členok

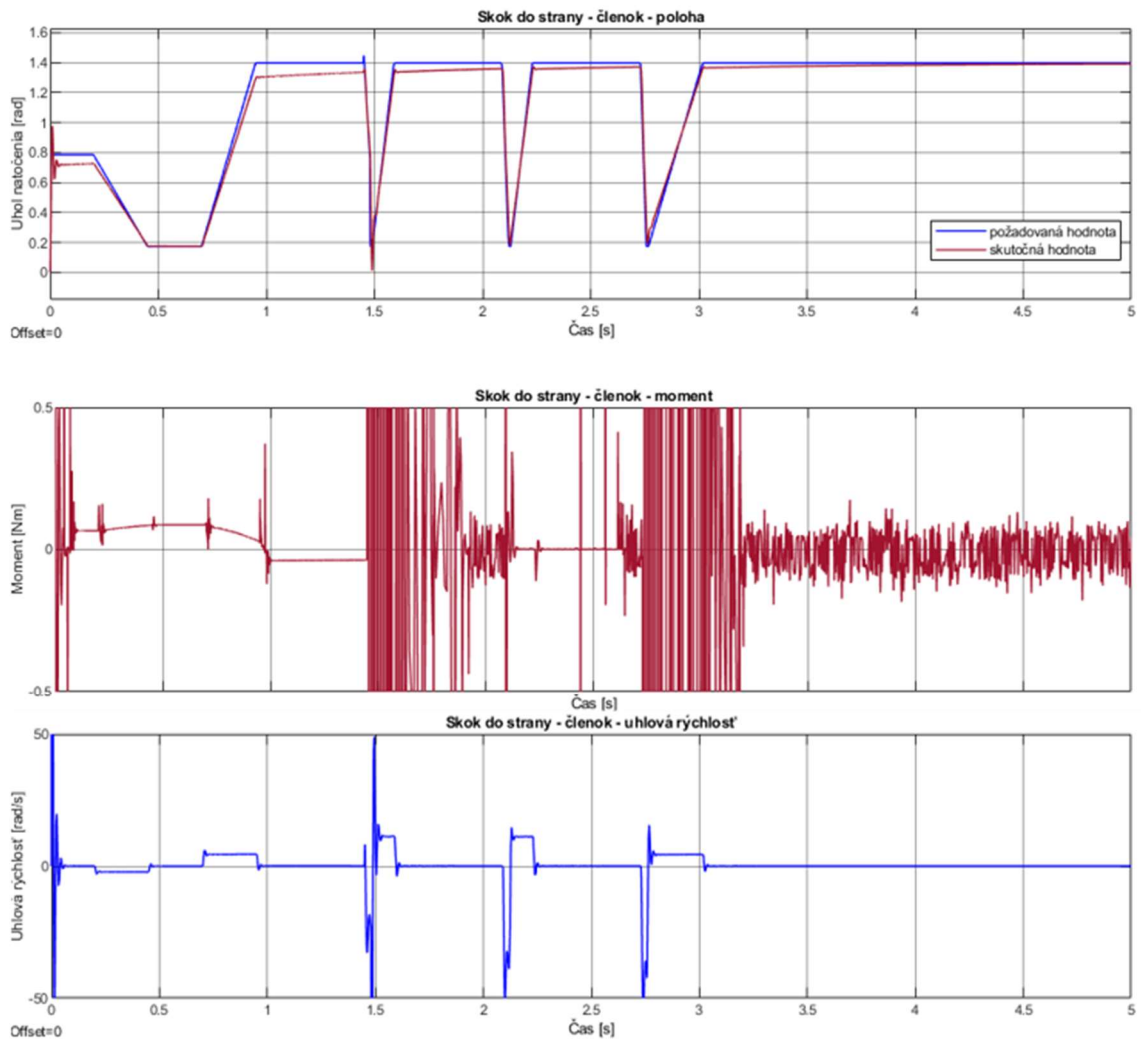


Obr. 38 Skok na mieste - koleno

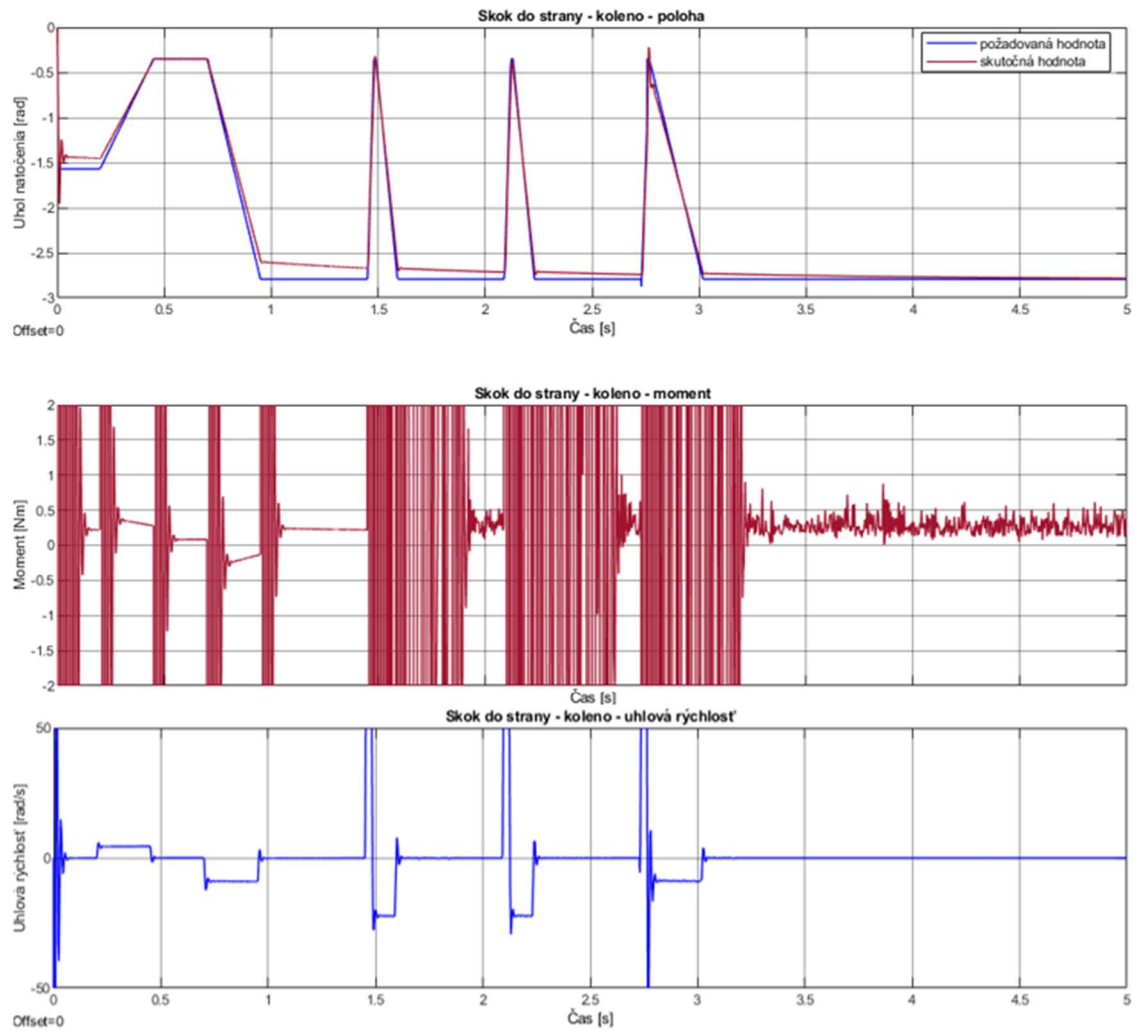


Obr. 39 Skok na mieste - bedro

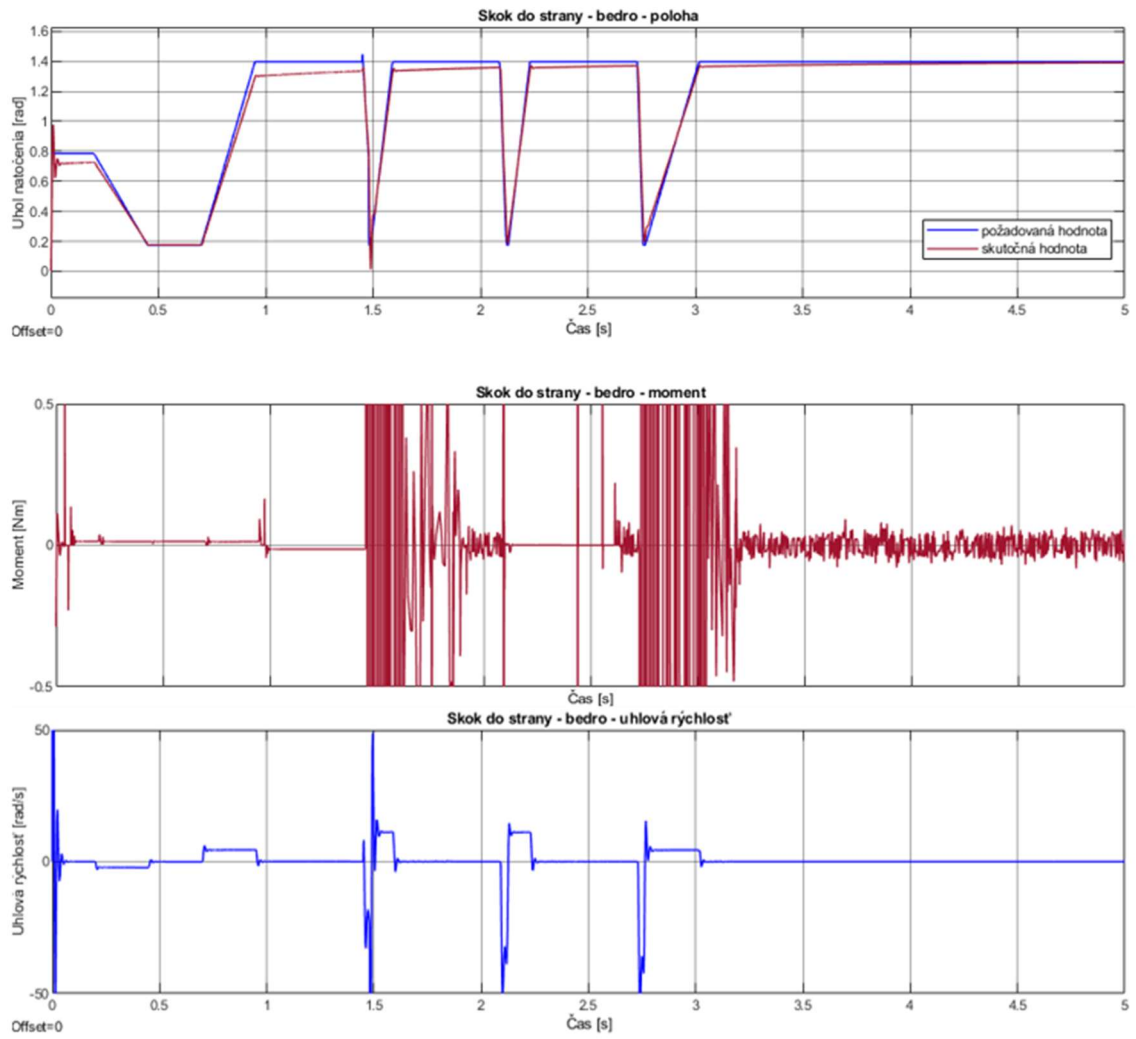
## 7.5 Skok do diaľky



Obr. 40 Skok do diaľky - členok

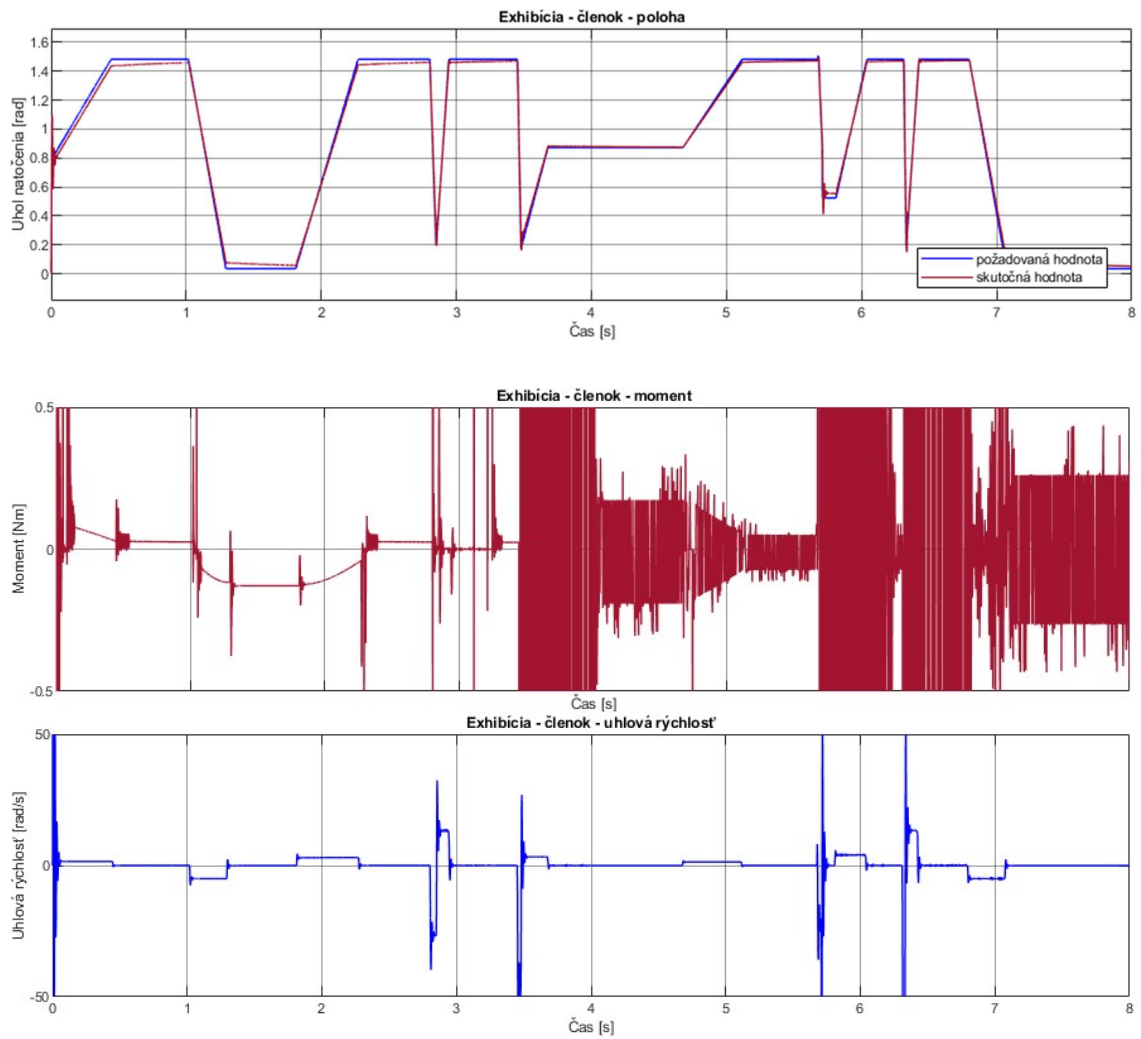


Obr. 41 Skok do dálky - koleno

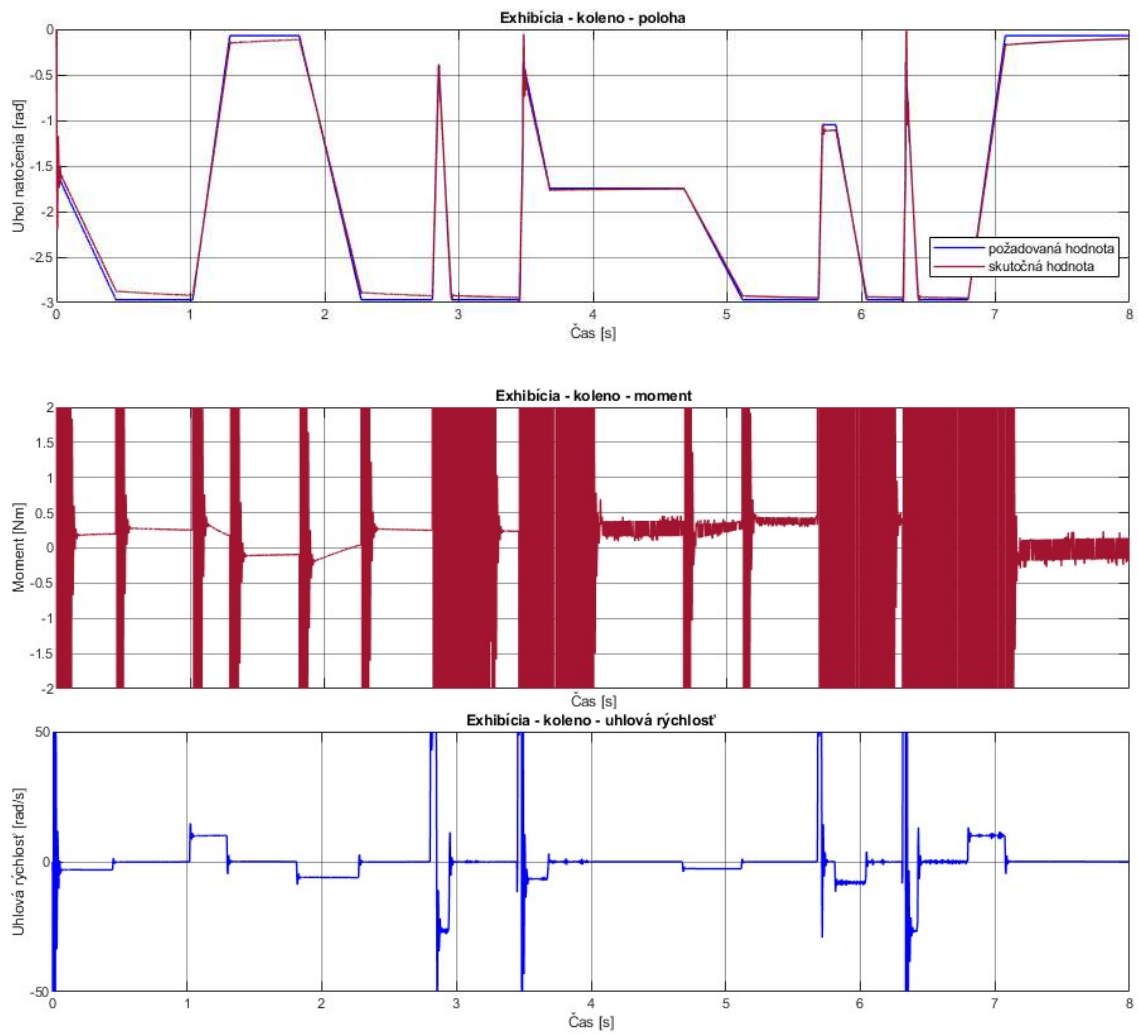


Obr. 42 Skok do diaľky - bedro

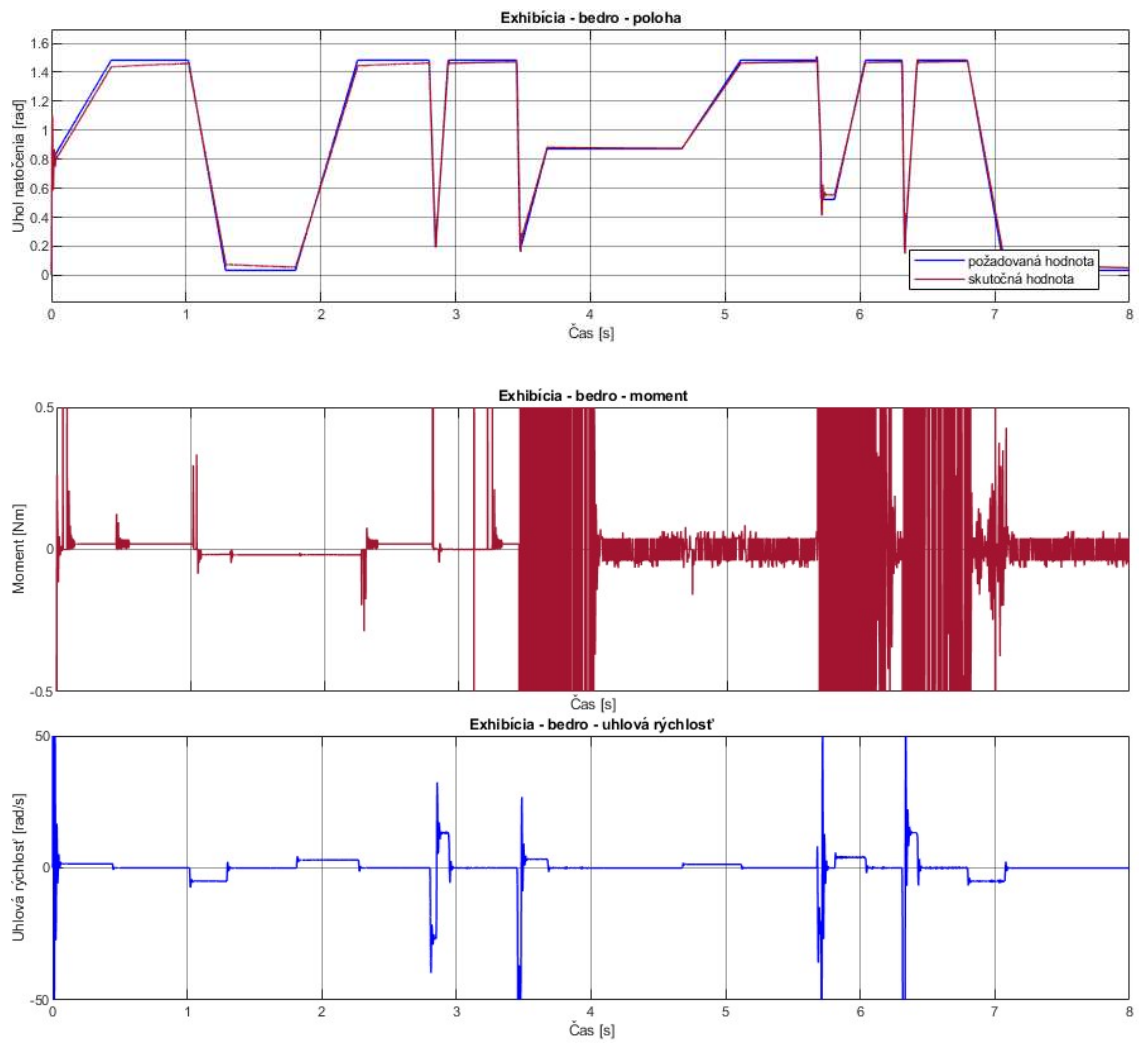
## 7.6 Exhíbia



Obr. 43 Exhíbia - členok

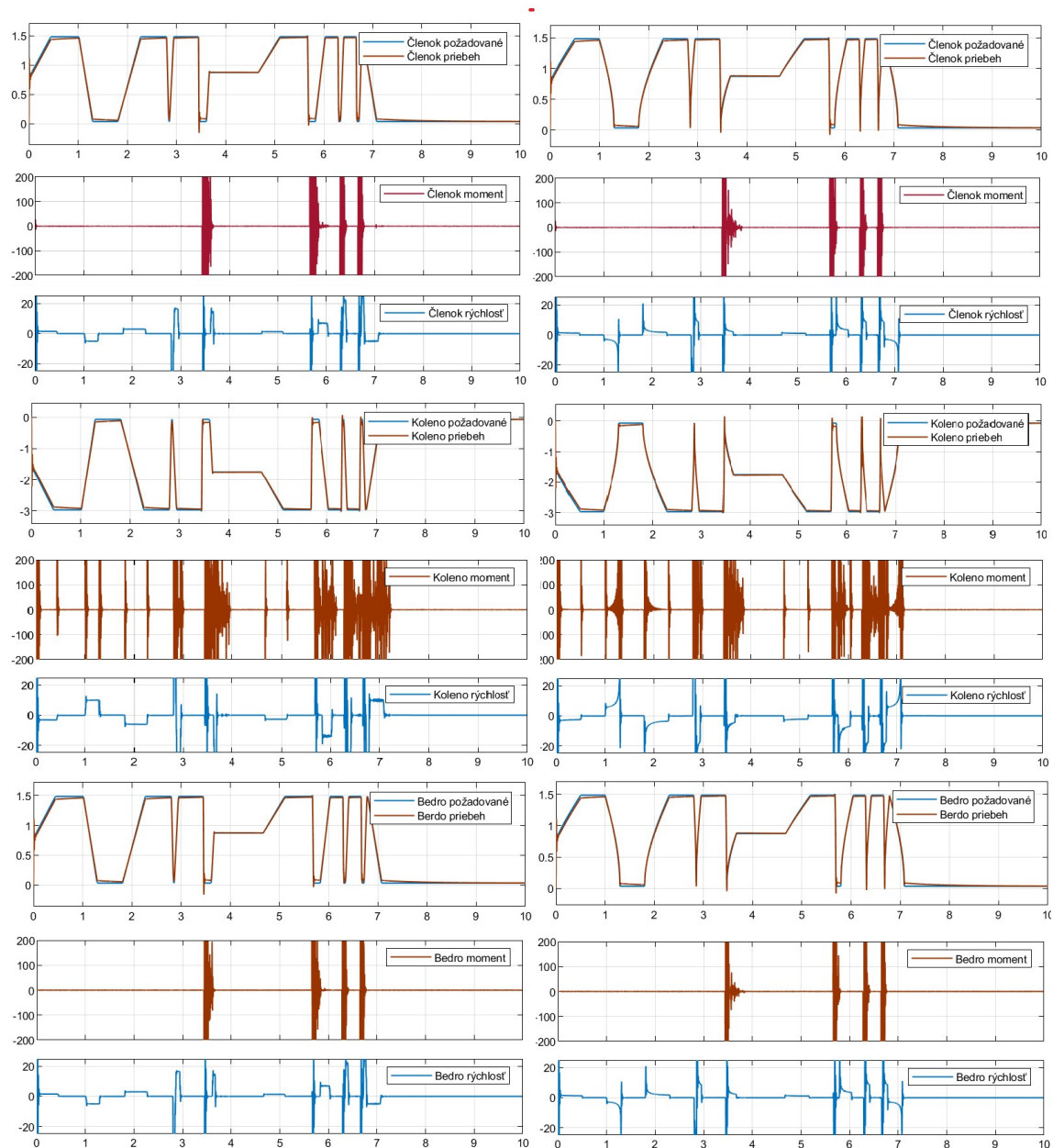


Obr. 44 Exhíbia - koleno



Obr. 45 Exhibícia - bedro

## 7.7 Porovnanie nelineárneho a lineárneho zrýchlenia pohybu



Obr. 46 Nelineárne vs. lineárne zrýchlenie pohybu

Nakoniec sme sa rozhodli otestovať správanie sa robota pri riadení kĺbov pri lineárnom zrýchlení vertikálneho pohybu. Na Obr. 46 vľavo je zachytený priebeh exhibície pri nelineárnom zrýchlení. Na obrázku vpravo sú zobrazené priebehy, zodpovedajúce konštantnému zrýchleniu pohybu vo vertikálnom smere. Momentové priebehy sú si do značnej miery podobné. Výrazné rozdiely môžeme sledovať v prípade rýchlosti pohybu. Tento proces sme popísali v kapitole 2.3.5.1.

Na základe zachytených priebehov sledovaných veličín celkovo pôsobí riadenie s nelineárnym priebehom stabilnejšie. Overenie, ktorý spôsob riadenia je v tej ktorej situácii vhodnejší môže byť námetom na samostatnú prácu.