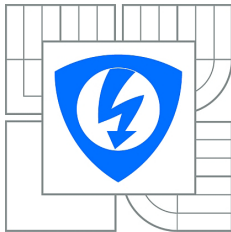


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

THE FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF CONTROL AND INSTRUMENTATION

# HUMANOIDNÍ ROBOT

HUMANOID ROBOT

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

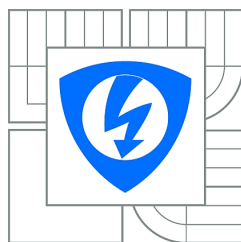
AUTOR PRÁCE  
AUTHOR

Bc. JAKUB VYORAL

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. LUKÁŠ KOPEČNÝ, PhD.

BRNO 2011



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav automatizace a měřicí techniky

# Diplomová práce

magisterský navazující studijní obor  
Kybernetika, automatizace a měření

**Student:** Bc. Jakub Vyoral  
**Ročník:** 2

**ID:** 72769  
**Akademický rok:** 2010/2011

## NÁZEV TÉMATU:

**Humanoidní robot**

## POKYNY PRO VYPRACOVÁNÍ:

Vytvořte matematický model podvozku a algoritmus chůze kráčejiho robotu. Výsledky práce vizualizujte.

## DOPORUČENÁ LITERATURA:

JONES, J., FLYNN, A., SEIGER, B. Mobile Robots Inspiration to Implementation. A K Peters, Massachusetts, 1999. 457 p. ISBN 1-56881-097-0

**Termín zadání:** 7.2.2011

**Termín odevzdání:** 23.5.2011

**Vedoucí práce:** Ing. Lukáš Kopečný, Ph.D.

**prof. Ing. Pavel Jura, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **Abstrakt**

Práce je zaměřena na generování hybridní dynamiky dvounohého robota pomocí symbolického toolboxu programu Matlab. Dále se zabývá linearizací nelineárního modelu a optimálním řízením stavovým regulátorem. Implementuje objektový návrh generátoru a simulátoru s využitím jádra Matlab Simulink. Ovládání s vizualizací výsledků je zprostředkováno uživatelským GUI.

## **Summary**

This work deals with automatic generation of hybrid dynamics of bipedal robot in Matlab symbolic toolbox. Next goal is to provide linearisation of nonlinear model and achieve optimal state space controller. Work implements object model generator and simulator based on core of Matlab Simulink and visualisation of the results in user friendly GUI.

## **Klíčová slova**

robot, biped, simulink

## **Keywords**

robot, biped, simulink

VYORAL, J. *Humanoidní robot*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 54 s. Vedoucí diplomové práce doc. Ing. Lukáš Kopečný, PhD.

Prohlašuji, že svoji diplomovou práci na téma **Humanoidní robot** jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

Bc. Jakub Vyoral

Děkuji vedoucímu diplomové práce Ing. Lukáši Kopečnému, PhD. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé práce.

Bc. Jakub Vyoral

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>Úvod</b>	<b>3</b>
<b>2 Matematický model</b>	<b>4</b>
2.1 Popis . . . . .	4
2.1.1 Modely založené na principu inverzního kyvadla . . . . .	4
2.1.2 Model otevřeného kinematického řetězce . . . . .	5
2.2 Značení . . . . .	7
2.3 Definice kloubového prostoru . . . . .	10
2.3.1 Relativní souřadnice . . . . .	10
2.3.2 Tvarové souřadnice . . . . .	11
2.4 Model zhoupnutí . . . . .	13
2.5 Impaktní model . . . . .	16
2.6 Řízení . . . . .	22
2.6.1 Linearizace modelu . . . . .	24
2.6.2 3 dof . . . . .	25
2.6.3 5 dof . . . . .	26
2.7 Ověření linearity dof=3 . . . . .	26
2.8 Návrh regulátoru . . . . .	28
<b>3 Simulace</b>	<b>32</b>
3.1 Program . . . . .	32
3.2 Generování modelu . . . . .	32
3.3 Simulátor . . . . .	35
3.4 Grafické rozhraní . . . . .	38
3.4.1 Chůze . . . . .	38
3.4.2 Generate . . . . .	40
3.5 Grafy 3 dof . . . . .	42
<b>4 Závěr</b>	<b>48</b>
<b>5 Výsledky</b>	<b>50</b>

# Seznam obrázků

2.1	Inverzní kyvadlo . . . . .	4
2.2	Hybridní model . . . . .	6
2.3	Kinematický řetězec . . . . .	11
2.4	Tvarové souřadnice . . . . .	12
2.5	Schéma ověření linearity . . . . .	27
2.6	Odezva na jednotkový skok . . . . .	27
2.7	Odezva na jednotkový skok s konstantou navýstupu . . . . .	28
2.8	Stavový regulátor . . . . .	29
3.1	Simulační schéma . . . . .	36
3.2	Blok <i>zero_dyn</i> . . . . .	37
3.3	Blok <i>robot</i> . . . . .	37
3.4	Blok <i>events</i> . . . . .	38
3.5	Vzhled GUI . . . . .	41
3.6	Chůze robota . . . . .	42
3.7	Průběh řídicích momentů $M$ . . . . .	42
3.8	Průběh kloubových souřadnic $q(t)$ . . . . .	43
3.9	Průběh rychlostí kloubových souřadnic $\dot{q}(t)$ . . . . .	43
3.10	Řez stavovým prostorem . . . . .	44
3.11	Poincarého mapa pro úhel $q_1$ . . . . .	44
3.12	Řez stavovým prostorem . . . . .	45
3.13	Poincarého mapa pro úhel $q_2$ . . . . .	45
3.14	Řez stavovým prostorem . . . . .	46
3.15	Poincarého mapa pro úhel $q_3$ . . . . .	46
3.16	Fázový diagram . . . . .	47

# 1. Úvod

Práce se zabývá sestavením matematického modelu nohou humanoidního robota se 3 a 5 stupni volnosti. Tento model lze generovat automaticky podle zvolených kritérií pomocí Lagrangeových rovnic 2.druhu a je obdržen jako sada m-file souborů vhodných pro simulaci. Při generování je kladen důraz na symbolický výpočet všech částí s obsluhou přes uživatelské GUI.

Simulátor vhodný pro hybridní model robota využívá jádra programu Matlab Simulink pro simulaci mezivýsledku každého kroku. *mdl* model pracuje s maticovými signály proměnné délky a je navázán na metody třídy simulátoru.

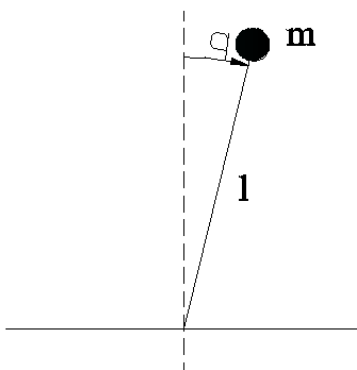
K dosažení mezního cyklu v soustavě je použit stavový regulátor. Obsluha simulace je řešena v GUI.



## 2. Matematický model

### 2.1. Popis

#### 2.1.1. Modely založené na principu inverzního kyvadla



Obrázek 2.1: Inverzní kyvadlo

Na (obr.2.1) je inverzní kyvadlo délky  $l$  s hmotností soustředěnou v hmotném bodě na konci. Je uchyceno v kloubu bez tření. Hmotnost odpovídá hmotnosti torza robota a kyvadlo jeho noze. Takto si lze představit zjednodušený model chůze po nohách. Pokud je spodní část kyvadla považována za chodidlo a vrchní část za kyčel, lze takto nahradit nohu kyvadlem s proměnnou délkou závislou na kloubových souřadnicích  $l(\mathbf{q})$ . Jedná se tak o nejjednodušší model chůze s jednou stojnou nohou. [9]

#### Kyvadlo s lineárním aktorem

Je ve své podstatě model inverzního kyvadla, který je rošířen o říditelnou délku kyvadla. Změnou délky se mění moment setrvačnosti a tak i pohyb kyvadla simulující chůzi.

#### Model s více klouby

Je ekvivalentem kyvadla s lineárním aktorem. Místo lineárního prvku zde figuruje jeden, nebo více kloubů, které mění moment setrvačnosti soustavy. Tento model už má více podobu jedné nohy bez chodidla.

#### Model s chodidly

U modelu se špičatým chodidlem jako u modelu s více klouby není možnost kontrolovat rychlost otáčení ve virtuálním kloubu se zemí. Tato myšlenka vede k zavedení chodidla. Může tak být změněno místo, kolem něhož robot rotuje a jeho dopředná rychlost přidáním řídicího momentu navíc.

S přidáváním stupňů volnosti a možnostmi chodidla je dynamická chůze řešena pomocí ZMP (zero moment point), nebo také nazývaný COP (center of pressure). Regulací polohy

tohoto bodu, který určuje střed rozložení tlaku vyvolaného stojnou nohou robota se řídí v současnosti nejpokročilejší humanoidní roboti.

Dopředná rychlost nemůže být regulována libovolně, jako i u složitějších modelů s dynamickou chůzí není možné řídit bez obezení z důvodu, že COP leží vně polygonu stojné nohy. Rychlost může být v omezené míře modifikována v průběhu kroku a mezi jednotlivými kroky s pomocí diskretních událostí.

1. změnou délky kroku
2. změnou událostí
3. přemisťováním pozice COP stojné nohy
4. nastavením rozložení sil ve fázi dvojité opory
5. změnou polohy těžiště náklonem
6. optimalizací akčních zásahů

Existuje řada dalších více, či méně složitých modelů s nohami bez váhy, planární s rozloženou vahou, etc.

### 2.1.2. Model otevřeného kinematického řetězce

Matematický model humanoidního robota je ze své podstaty hybridním systémem, spojuje v sobě prvky spojitých i diskretních modelů. Před samotným modelováním je nutné zvážit, jaké požadavky na robota klademe, jako jsou chůze, běh, chůze do schodů a popřípadě další složitější situace.

Uvažme jednoduchý planární model s rozloženou vahou. V případě chůze robota je vhodné modelovat nohy jako otevřený kinematický řetězec s pevně uchyceným koncem k zemi. Jednotlivé nohy jsou pak podřetězci, z nichž uchycená část se nazývá stojná noha a část s volným koncem švihová noha. Zvolením kloubového prostoru pak můžeme dosáhnout různých forem modelu, z nichž jsou některé jednodušší pro řízení.

Je třeba si zavést omezení, která budou směřovat k postupu řešení chůze robota :

1. robot kráčí po ideální rovině
2. střídá fáze s jednou a dvěma stojnými nohami (jedna nebo obě nohy se dotýkají podlahy)
3. ve fázi opory jednou nohou uvažujeme styk s podlahou jako ideální kloub
4. dále uvažujeme všechny klouby a vazby jako ideální, bez ztrátového tření
5. při nárazu nohy na podlahu nedojde k podklouznutí, je uvažován jako pevný bod

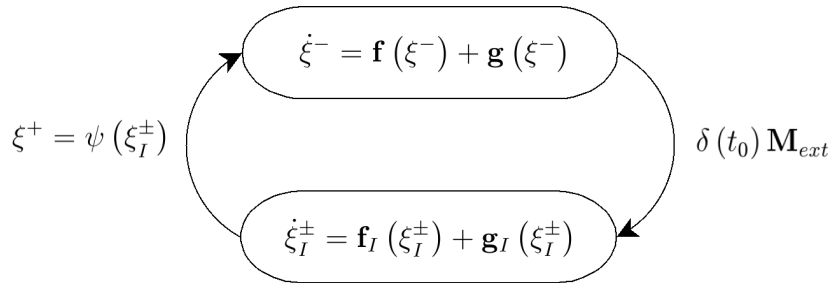
**Fáze s jednou stojnou nohou (fáze zhoupnutí)**

Fáze zhoupnutí probíhá ve chvíli, kdy má robot jednu nohu pevně na zemi a končí v okamžiku dotyku druhé nohy se zemí.

**Fáze se dvěma stojnými nohama (fáze nárazu)**

V okamžiku došlápnutí se provede přepoččet úhlů a rychlostí kloubového prostoru podle impaktního modelu. Fáze nárazu má impulzní charakter a redukuje se tak do jediného okamžiku [4]. Po impulzu se v modelu přesune pevný bod kinematického řetězce do místa došlápnutí. Akčními členy se systém řídí do dalšího kroku, kde opět dochází k fázi zhoupnutí.

Na (obr. 2.2) je znázorněn hybridní model, kde  $\xi$  je vektor stavového prostoru, index  $I$  značí model nárazu. K přepnutí mezi modely dochází při působení externího momentu  $\mathbf{M}_{ext}$ , který má impulzní charakter. Index „-“ představuje limitní stav před nárazem, „±“ limitní stav po nárazu nohy o zem. Protože nárazový model a model zhoupnutí mají stejnou strukturu, která se liší jen bodem uchycení a model nárazu se provede v limitně krátkém čase, úhlová část kloubového prostoru předimpaktních souřadnic  $\xi^-$  a poimpaktních souřadnic  $\xi_I^\pm$  představuje stejný prostor se stejnou stojnou nohou. Pro pokračování do nové fáze zhoupnutí je třeba přepočítat kloubový prostor při výměně nohou jako  $\psi(\xi_I^\pm)$  na nové poimpaktové souřadnice značené poimpaktovým „+“.



Obrázek 2.2: Hybridní model

## 2.2. Značení

Symbol	Popis
$x, y$	směr v kartézském souřadném systému
$i$	index z rozsahu $i \in \langle 1, n \rangle$
$n$	počet stupňů volnosti (dof)
$m$	počet akčních členů (n-1)
$ext$	označení externí veličiny
$c$	celkový (např. celková hmotnost, etc.)
$g$	značení polohy těžiště
$end$	konec kinematického řetězce
+	značení souřadnice před impaktem nohy o zem
-	značení souřadnice po impaktu nohy o zem

Tabulka 2.1: Symbolika indexů

Symbol	Jednotka	Popis
$x, y$	$[m]$	souřadnice v kartézském souřadném systému
$\xi$		stavový prostor (z důvodu použitého symbolu x)
$\mathbf{p}$	$[m]$	souřadnice bodu
$L$	$[N]$	Lagrangeova funkce
$\mathcal{L}$		součást operátoru Lieovy derivace (z důvodu použitého L)
$E_k, E_p, R$	$[J]$	kinetická, potenciální a ztrátová energie
$A$	$[J]$	práce v systému
$\mathbf{I}$	$[-]$	jednotková matice
$q_i$	$[rad]$	souřadnice kloubového prostoru
$\dot{q}_i$	$[rad.s^{-1}]$	rychlost v kloubovém prostoru
$\ddot{q}_i$	$[rad.s^{-2}]$	zrychlení v kloubovém prostoru
$M_i$	$[Nm]$	moment
$J_i$	$[kg.m^2]$	moment setrvačnosti
$m_i$	$[kg]$	hmotnost
$l_i$	$[m]$	délka části robota
$\mu$	$[-]$	součinitel tření

Tabulka 2.2: Symbolika veličin

Lagrangeovy rovnice vychází z optimalizace rozložení energie v systému. Systém s nejmenší energií odpovídá hledanému modelu, který nezávisí na fyzikálním principu.

Lagrangeova funkce je definována jako rozdíl potenciální a kinetické energie systému:

$$L = E_K - E_P \quad (2.1)$$

Zvolíme-li zobecněné výchylky a rychlosti jako:

$$\mathbf{q} = [q_1, \dots, q_n]^T \quad (2.2)$$

$$\dot{\mathbf{q}} = [\dot{q}_1, \dots, \dot{q}_n]^T \quad (2.3)$$

kde:

$$n = dof \quad (2.4)$$

Pak je tvar Lagrangeových rovnic II. Druhu :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} - \frac{\partial R}{\partial \mathbf{q}} = \frac{\partial A}{\partial \mathbf{q}} \quad (2.5)$$

kde:  $\mathbf{q}$  - vektor zobecněných výchylek  
 $\dot{\mathbf{q}}$  - vektor zobecněných rychlostí

Definujme další vektory a matice využitě při výpočtech. Souřadnice kloubů včetně koncových bodů a těžišť jednotlivých prvků robota v kartézském souřadném systému :

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad \mathbf{p}_{gi} = \begin{bmatrix} x_{gi} \\ y_{gi} \end{bmatrix} \quad (2.6)$$

Pozice nohy při došlápnutí (konec kinematického řetězce) a pozice těžiště

$$\mathbf{p}_{end} = \begin{bmatrix} x_{end} \\ y_{end} \end{bmatrix} \quad \mathbf{p}_{cog} = \begin{bmatrix} x_{cog} \\ y_{cog} \end{bmatrix} \quad (2.7)$$

Vektory x a y souřadnic těžišť jednotlivých prvků robota

$$\mathbf{x}_g = [x_{g1} \ x_{g2} \ \cdot \ x_{gn}]^T \quad \mathbf{y}_g = [y_{g1} \ y_{g2} \ \cdot \ y_{gn}]^T \quad (2.8)$$

Matice hmotností a momentů setrvačnosti

$$\mathbf{m} = \begin{bmatrix} m_1 & 0 & 0 & \cdot & 0 \\ 0 & m_2 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & m_n \end{bmatrix} \quad \mathbf{J} = \begin{bmatrix} J_1 & 0 & 0 & \cdot & 0 \\ 0 & J_2 & 0 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdot & J_n \end{bmatrix} \quad (2.9)$$

kde:  $m_i$  - hmotnost i-té části robota  
 $J_i$  - moment setrvačnosti i-té části robota  
 $m_n, J_n$  - hmotnost a moment setrvačnosti torza

Vstupní momenty:

$$\mathbf{M} = [M_1 \ M_2 \ \cdot \ M_m]^T \quad (2.10)$$

$$\mathbf{M}_{ext} = [M_{ext1} \ M_{ext2} \ \cdot \ M_{extm}]^T \quad (2.11)$$

Kinetickou energii určíme jako součet translační a rotační energie [7]:

$$E_K = \frac{1}{2} [\dot{\mathbf{x}}_g^T \mathbf{m} \dot{\mathbf{x}}_g + \dot{\mathbf{y}}_g^T \mathbf{m} \dot{\mathbf{y}}_g + \dot{\mathbf{q}}^T \mathbf{J} \dot{\mathbf{q}}] \quad (2.12)$$

$$E_P = \sum_{i=1}^n \mathbf{m}_{ii} g y_{gi} \quad (2.13)$$

$$A = \sum_{i=2}^n M_i q_i \quad (2.14)$$

kde:  $E_K$  - kinetická energie  
 $E_P$  - potenciální energie  
 $A$  - práce vykonaná v systému  
 $\mathbf{M}$  - vstupní momenty  
 $\mathbf{q}$  - vektor zobecněných výchylek  
 $\dot{\mathbf{q}}$  - vektor zobecněných rychlostí

## 2.3. Definice kloubového prostoru

### 2.3.1. Relativní souřadnice

DH-transformace je definována jako:

$$\mathbf{H} = Rot(\Theta)_z Trans(0, 0, d_i) Trans(a_i, 0, 0) Rot(\alpha)_x \quad (2.15)$$

Zjednodušíme si zápis této transformace jako matici  $\mathbf{H}$

$$\mathbf{H} = DH(a, \alpha, d, \Theta) \quad (2.16)$$

$i$ -tá transformace pak má tvar:

$$\mathbf{H}_i = \prod_{i=1}^n \mathbf{H}(a_i, 0, 0, q_i) \quad (2.17)$$

Matice transformace pro získání souřadnic těžišť částí robota, které jsou situovány v polovině délky dané části

$$\mathbf{H}_{\mathbf{g}_i} = DH\left(\frac{l_i}{2}, 0, 0, q_i\right) \mathbf{H}_{i-1} \quad (2.18)$$

Nultá transformace je brána jako jednotková matice o rozměru 4

$$\mathbf{H}_0 = \mathbf{E}_4 \quad (2.19)$$

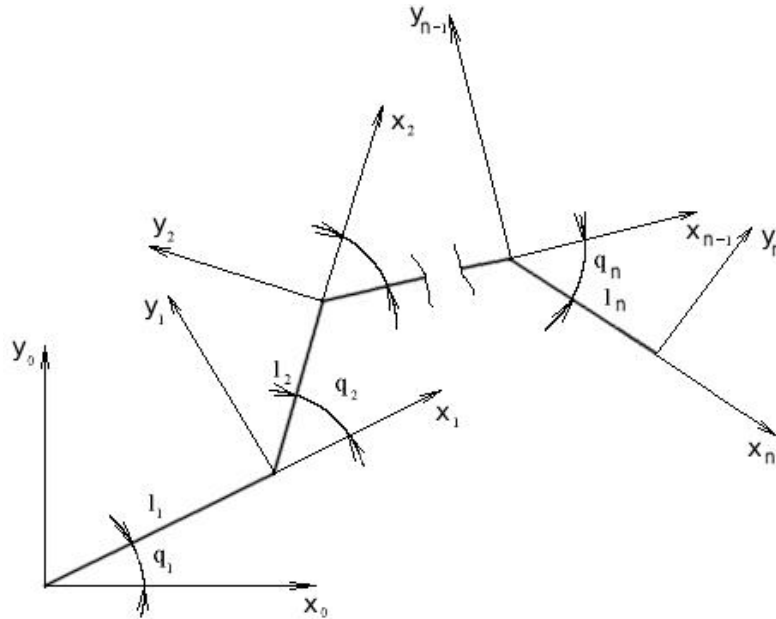
Počátek souřadnic je pevně nastaven na

$$\mathbf{p}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.20)$$

Rovnice přímé kinematiky pak mají tvar :

$$\begin{bmatrix} \mathbf{p}_i \\ 0 \\ 1 \end{bmatrix} = \mathbf{H}_i \begin{bmatrix} \mathbf{p}_0 \\ 0 \\ 1 \end{bmatrix} \quad (2.21)$$

### 2.3. DEFINICE KLOUBOVÉHO PROSTORU



Obrázek 2.3: Kinematický řetězec

Zvolením kloubových parametrů podle DH konvence získáme zároveň rovnice přímé kinematiky. Tabulka DH parametrů pak pro robota pohybujícího se pouze v 2D prostoru sagitální roviny vypadá následovně:

vazba	$a_i$	$\alpha_i$	$d_i$	$\theta_i$
1	$l_1$	0	0	$q_1$
2	$l_2$	0	0	$q_2$
.	.	.	.	.
n-1	$l_{n-1}$	0	0	$q_{n-1}$
n	$l_n$	0	0	$q_n$

Tabulka 2.3: DH parametry

(n-1)-ní transformace tabulky 2.3 je posledním členem kinematického řetězce. n-tý prvek je přidán navíc jako vazba mezi pánví robota a jeho torzem. Vytváří tak inverzní kyvadlo umístěné v kyčli robota.

#### 2.3.2. Tvarové souřadnice

Vychází z tvaru robota. Jedna souřadnice je zvolena vzhledem k inerciální soustavě. Ostatní souřadnice jsou pak definovány relativně v závislosti na souřadnici absolutní. Definujeme-li vztažnou souřadnici jako úhel natočení torza robota, pak zbylé relativní souřadnice určují jeho tvar a tím i polohu těžiště. Výhodou této volby je snadný přepočítání kloubového prostoru při nárazu nohy o zem.





$$\mathbf{p}_4 = \begin{bmatrix} l_1 \sin(q_3 + q_5) + l_1 \cos\left(-\frac{\pi}{2} - q_1 - q_2 - q_5\right) + l_1 \cos\left(\frac{3\pi}{2} - q_2 - q_5\right) + l_1 \sin(q_3 + q_4 + q_5) \\ l_1 \cos(q_3 + q_5) + l_1 \sin\left(-\frac{\pi}{2} - q_1 - q_2 - q_5\right) + l_1 \cos(q_3 + q_4 + q_5) + l_1 \sin\left(\frac{3\pi}{2} - q_2 - q_5\right) \end{bmatrix}$$

$$\mathbf{p}_5 = \begin{bmatrix} l_5 \sin(q_5) + l_1 \cos\left(-\frac{\pi}{2} - q_1 - q_2 - q_5\right) + l_1 \cos\left(\frac{3\pi}{2} - q_2 - q_5\right) \\ l_5 \cos(q_5) + l_1 \sin\left(-\frac{\pi}{2} - q_1 - q_2 - q_5\right) + l_1 \sin\left(\frac{3\pi}{2} - q_2 - q_5\right) \end{bmatrix}$$

Pro  $dof = 5$  lze přepočít mezi takto definovanými souřadnicemi a relativními souřadnicemi snadno odvodit:

$$\mathbf{q}_{rel} = \mathbf{R}\mathbf{q}_{tv} + \phi \quad (2.25)$$

$$\mathbf{R} = \begin{bmatrix} -1 & -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \phi = \begin{bmatrix} -\frac{\pi}{2} \\ 0 \\ \pi \\ 0 \\ -\pi \end{bmatrix} \quad (2.26)$$

## 2.4. Model zhoupnutí

Z fyzikálního hlediska je vhodné zapsat model ve tvaru:

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\mathbf{u} \quad (2.27)$$

- kde:  $\mathbf{D}$  - matice momentů setrvačnosti  
 $\mathbf{C}$  - Coriolisova a odstředivá matice  
 $\mathbf{G}$  - vektor gravitace  
 $\mathbf{B}$  - matice vlivu vstupních momentů v kloubech  
 $\mathbf{u}$  - vektor řídicích momentů  
 $\mathbf{q}$  - vektor relativních úhlů  
 $\dot{\mathbf{q}}$  - vektor relativních úhlových rychlostí  
 $\ddot{\mathbf{q}}$  - vektor relativních úhlových zrychlení

K výpočtu matic použijme následující rovnici, uvažujme systém beze ztrát:

$$\frac{d}{dt} \left( \frac{\partial E_k}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial E_k}{\partial \mathbf{q}} + \frac{\partial E_p}{\partial \mathbf{q}} = \frac{\partial A}{\partial \mathbf{q}} \quad (2.28)$$

Zavedme si mezivýpočet:

$$\mathbf{c} = \frac{d}{dt} \left( \frac{\partial E_k}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial E_k}{\partial \mathbf{q}} \quad (2.29)$$

Matice potom získáme následovně:

$$\mathbf{D}(\mathbf{q}) = \frac{\partial \mathbf{c}}{\partial \ddot{\mathbf{q}}} \quad (2.30)$$

$$\mathbf{G}(\mathbf{q}) = \left( \frac{\partial E_p}{\partial \mathbf{q}} \right)^T \quad (2.31)$$

$$\mathbf{B}(\mathbf{q}) = \frac{\partial^2 A}{\partial \mathbf{q} \partial \mathbf{M}} \begin{bmatrix} \mathbf{0}_{1 \times m} \\ \mathbf{I}_m \end{bmatrix} \quad (2.32)$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \frac{\partial}{\partial \dot{\mathbf{q}}} (\mathbf{c} - \mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}}) \quad (2.33)$$

$$\mathbf{u} = \mathbf{M} \quad (2.34)$$

$$(2.35)$$

Matice  $\mathbf{D}_{3 \times 3}(\mathbf{q})$ :

$$D_{1,1} = J_1 + \frac{l_1^2 m_1}{2} + \frac{l_1^2 m_3}{4} \quad (2.36)$$

$$D_{1,2} = -\frac{l_1^2 m_1 \cos(q_1 - q_2)}{4} \quad (2.37)$$

$$D_{1,3} = \frac{l_1 (2 l_1 m_1 + l_1 m_3 - l_3 m_3 \cos(q_1) - l_1 m_1 \cos(q_1 - q_2))}{4} \quad (2.38)$$

$$D_{2,2} = \frac{m_1 l_1^2}{4} + J_1 \quad (2.39)$$

$$D_{2,3} = -\frac{l_1^2 m_1 (\cos(q_1 - q_2) - 1)}{4} \quad (2.40)$$

$$D_{3,3} = J_3 + \frac{3l_1^2 m_1}{4} + \frac{l_1^2 m_3}{4} + \frac{l_3^2 m_3}{4} - \frac{l_1^2 m_1 \cos(q_1 - q_2)}{2} - \frac{l_1 l_3 m_3 \cos(q_1)}{2} \quad (2.41)$$

Matice  $\mathbf{C}_{3 \times 3}(\mathbf{q}, \dot{\mathbf{q}})$ :

$$C_{1,2} = \left(-\frac{1}{4}\right) l_1^2 \sin(q_1 - q_2) m_1 (\dot{q}_2 + \dot{q}_3) \quad (2.42)$$

$$C_{1,3} = \left(-\frac{1}{4}\right) l_1 (\dot{q}_3 l_3 m_3 \sin(q_1) + \dot{q}_2 l_1 m_1 \sin(q_1 - q_2) + \dot{q}_3 l_1 m_1 \sin(q_1 - q_2)) \quad (2.43)$$

$$C_{2,3} = \left(\frac{1}{4}\right) l_1^2 m_1 \sin(q_1 - q_2) (\dot{q}_1 + \dot{q}_3) \quad (2.44)$$

$$C_{3,3} = \left(\frac{1}{4}\right) l_1 (\dot{q}_1 l_3 m_3 \sin(q_1) + \dot{q}_1 l_1 m_1 \sin(q_1 - q_2) - \dot{q}_2 l_1 m_1 \sin(q_1 - q_2)) \quad (2.45)$$

Matice  $\mathbf{G}_{1 \times 3}(\mathbf{q})$ :

$$G_{1,1} = g l_1 m_1 \sin(q_1 + q_3) + \frac{g l_1 m_3 \sin(q_1 + q_3)}{2} \quad (2.46)$$

$$G_{2,1} = -\frac{g l_1 m_1 \sin(q_2 + q_3)}{2} \quad (2.47)$$

$$G_{3,1} = g m_1 \left( \frac{l_1 \sin(q_1 + q_3)}{2} - \frac{l_1 \sin(q_2 + q_3)}{2} \right) + g m_3 \left( \frac{l_1 \sin(q_1 + q_3)}{2} - \frac{l_3 \sin(q_3)}{2} \right) + \frac{g l_1 m_1 \sin(q_1 + q_3)}{2} \quad (2.48)$$

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.49)$$

Vyjádření (2.27) v explicitním tvaru vhodném pro simulaci:

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{D}^{-1}(\mathbf{q}) [\mathbf{B}(\mathbf{q}) \mathbf{u} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})] \end{bmatrix} \quad (2.50)$$

## 2.5. Impaktní model

Při rázu impulzního charakteru se v systému projeví působení vnějších momentů, které působí ve všech částech kinematického řetězce. Z principu virtuální práce je možné určit tyto vstupní momenty. Protože vyšetřujeme silové účinky na koncích řetězce, je potřeba uvolnit stojnou nohu, tedy dodat vztažený bod  $\mathbf{p}$ :

$$\mathbf{q}_I = \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix} \quad \dot{\mathbf{q}}_I = \begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} \quad (2.51)$$

Rovnice (2.51) platí za předpokladu, že  $\mathbf{p}$  je nezávislý na  $\mathbf{q}$ , tedy i na čase, proto při přepočtu je vhodné využít počátku souřadnic  $\mathbf{p}_0$ .

Derivace rozdílu energií podle zobecněné rychlosti před a po impaktu je rovna impulznímu externímu momentu [1],[2]:

$$\frac{\partial E_k}{\partial \dot{\mathbf{q}}_I^{\pm}} - \frac{\partial E_k}{\partial \dot{\mathbf{q}}_I^{-}} = \mathbf{P} \quad (2.52)$$

$$\mathbf{P} = \lim_{t \rightarrow t_0^+} \int_{t_0}^t \mathbf{M}_{ext} \delta(t) dt = \mathbf{M}_{ext} \quad (2.53)$$

kde:  $\delta(t)$  - dirakov impulz  
 $t_0$  - okamžik, kdy se projeví impulz  
 $\dot{\mathbf{q}}_I^{\pm}$  - kloubová rychlost po impaktu v předimpaktových souřadnicích

Symbol  $\pm$  je použit pro označení poimpaktového stavu. Protože při impaktu dochází ke změně modelu, mění se i kloubový prostor. Poimpaktové rychlosti  $\pm$  jsou tedy rychlosti na stejném modelu, pro získání kloubových proměnných jako počáteční podmínky nové fáze zhrounutí, je třeba provést přepočet podle (2.72).

Vyjděme z obecného vztahu pro výpočet kinetické energie [7],[6]:

$$E_k = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{D}(\mathbf{q}) \dot{\mathbf{q}} \quad (2.54)$$

$$\frac{\partial E_k}{\partial \dot{\mathbf{q}}} = \frac{1}{2} (\mathbf{D}(\mathbf{q}) \dot{\mathbf{q}})^T + \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{D}(\mathbf{q}) \quad (2.55)$$

Využijme vlastnosti, že matice  $\mathbf{D}(\mathbf{q})$  je symetrická

$$\mathbf{D}(\mathbf{q}) = \mathbf{D}(\mathbf{q})^T \quad (2.56)$$

Kombinací (2.55) a (2.56)

$$\frac{\partial E_k}{\partial \dot{\mathbf{q}}} = \mathbf{D}(\mathbf{q}) \dot{\mathbf{q}} \quad (2.57)$$

Pak (2.53) lze přepsat jako

$$\mathbf{D}_I(\mathbf{q}_I^\pm) \dot{\mathbf{q}}_I^\pm - \mathbf{D}_I(\mathbf{q}_I^-) \dot{\mathbf{q}}_I^- = \mathbf{M}_{\text{ext}} \quad (2.58)$$

Na konci kinematického řetěze vstupuje do systému impulzní síla, která má vliv na vnitřní momenty. Zavedeme-li si malé virtuální posunutí  $\delta$ , musí podle zákona zachování energie platit [7]

$$\mathbf{M}_{\text{ext}}^T \delta \mathbf{q}_I^- = \mathbf{F}_{\text{end}}^T \delta \mathbf{p}_{\text{end}} \quad (2.59)$$

Pokud jsou posunutí  $\delta$  nekonečně malá, závislost lze vyjádřit pomocí jakobiánu  $\mathbf{J}$

$$\delta \mathbf{q}_I^- = \mathbf{J} \delta \mathbf{p}_{\text{end}} \quad (2.60)$$

$$\mathbf{J}(\mathbf{q}_I^-) = \frac{\partial \mathbf{p}_{\text{end}}(\mathbf{q}_I^-)}{\partial \mathbf{q}_I^-} \quad (2.61)$$

Z rovnic (2.59), (2.60) a (2.61) dostaneme

$$\mathbf{M}_{\text{ext}} = \mathbf{J}(\mathbf{q}_I^-)^T \mathbf{F}_{\text{end}} \quad (2.62)$$

Dosadíme (2.62) do (2.58)

$$\mathbf{D}_I(\mathbf{q}_I^\pm) \dot{\mathbf{q}}_I^\pm - \mathbf{D}_I(\mathbf{q}_I^-) \dot{\mathbf{q}}_I^- = \mathbf{J}(\mathbf{q}_I^-)^T \mathbf{F}_{\text{end}} \quad (2.63)$$

Aby (2.63) platila, nesmí noha při kontaktu se zemí uklouznout [1]

$$\left| \frac{F_{xend}}{F_{yend}} \right| < \mu \quad (2.64)$$

kde:  $F_{xend}$  - síla ve směru osy x působící na konci kinematického řetězce při impaktu  
 $F_{yend}$  - síla ve směru osy y  
 $\mu$  - součinitel tření

Aby byla podmínka splněna, musí být rychlost na konci nohy po impaktu nulová. Tu lze získat z kloubových proměnných [4]:

$$\mathbf{J}(\mathbf{q}_I^-) \dot{\mathbf{q}}_I^\pm = \mathbf{0}_{2 \times 1} \quad (2.65)$$

Kombinací rovnic (2.65), (2.63) dostaneme soustavu:

$$\begin{bmatrix} \mathbf{D}_I(\mathbf{q}_I^-) & -\mathbf{J}(\mathbf{q}_I^-)^T \\ \mathbf{J}(\mathbf{q}_I^-) & \mathbf{0}_{2 \times 2} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_I^\pm \\ \mathbf{F}_{end} \end{bmatrix} = \begin{bmatrix} \mathbf{D}_I(\mathbf{q}_I^-) \dot{\mathbf{q}}_I^- \\ \mathbf{0}_{2 \times 1} \end{bmatrix} \quad (2.66)$$

$$\begin{bmatrix} \dot{\mathbf{q}}_I^\pm \\ \mathbf{F}_{end} \end{bmatrix} = \left( \begin{bmatrix} \mathbf{D}_I(\mathbf{q}_I^-) & -\mathbf{J}(\mathbf{q}_I^-)^T \\ \mathbf{J}(\mathbf{q}_I^-) & \mathbf{0}_{2 \times 2} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{D}_I(\mathbf{q}_I^-) \dot{\mathbf{q}}_I^- \\ \mathbf{0}_{2 \times 1} \end{bmatrix} \quad (2.67)$$

Hodnoty  $\mathbf{q}$  v limitním čase po impaktu:

$$\mathbf{q}^\pm = \mathbf{q}^- \quad (2.68)$$

Poimpaktní hodnoty pro novou fázi zhroupení dostaneme podle (obr.2.2):

$$\xi^+ = \psi(\xi_I^\pm) \quad (2.69)$$

$$\xi = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \quad (2.70)$$

Přepočtová funkce (2.69) se může rozložit na 2 rovnice pro kloubové úhly a jejich rychlosti. Platí mezi nimi následující lineární závislost.

$$\mathbf{q}^+ = \mathbf{Q}_{n \times n} \mathbf{q}^\pm + \mathbf{\Pi}_{1 \times n} \quad (2.71)$$

$$\dot{\mathbf{q}}^+ = \mathbf{Q}_{n \times n} \dot{\mathbf{q}}^\pm \quad (2.72)$$

kde:  $\mathbf{Q}$  - matice přepočtu úhlových souřadnic mezi modely před a po impaktu  
 $\mathbf{\Pi}$  - vektor konstantních uhlových natočení  
 $\dot{\mathbf{q}}^\pm$  - vypočítáno z (2.51)

Pro tvarové souřadnice podle (obr.2.4) platí:

$$\mathbf{Q}_{n \times n} = \begin{bmatrix} \begin{bmatrix} 0 & \cdot & 0 & 1 \\ 0 & \cdot & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & 0 & 0 \end{bmatrix}_{m \times m} & \mathbf{0}_{m \times 1} \\ \mathbf{0}_{1 \times m} & 1 \end{bmatrix} \quad (2.73)$$

$$\mathbf{\Pi} = \mathbf{0}_{1 \times n} \quad (2.74)$$

Pro detekci impaktu nohy o zem je vhodné použít podmínku:

$$y_{end} = 0 \quad \text{nebo} \quad y_{end} \leq 0 \quad (2.75)$$

kde podmínka nerovnosti je vhodnější pro simulaci, kde reálná čísla nejsou rovny identicky nule.



Matice  $\mathbf{D}_{i3 \times 3}(\mathbf{q})$ :

$$D_{i1,1} = J_1 + \frac{l_1^2 m_1}{2} + \frac{l_1^2 m_3}{4} \quad (2.76)$$

$$D_{i1,2} = -\frac{l_1^2 m_1 \cos(q_1 - q_2)}{4} \quad (2.77)$$

$$D_{i1,3} = \frac{l_1 (2 l_1 m_1 + l_1 m_3 - l_3 m_3 \cos(q_1) - l_1 m_1 \cos(q_1 - q_2))}{4} \quad (2.78)$$

$$D_{i1,4} = -\frac{l_1 \cos(q_1 + q_3) (2 m_1 + m_3)}{2} \quad (2.79)$$

$$D_{i1,5} = \frac{l_1 \sin(q_1 + q_3) (2 m_1 + m_3)}{2} \quad (2.80)$$

$$D_{i2,2} = \frac{m_1 l_1^2}{4} + J_1 \quad (2.81)$$

$$D_{i2,3} = -\frac{l_1^2 m_1 (\cos(q_1 - q_2) - 1)}{4} \quad (2.82)$$

$$D_{i2,4} = \frac{l_1 m_1 \cos(q_2 + q_3)}{2} \quad (2.83)$$

$$D_{i2,5} = -\frac{l_1 m_1 \sin(q_2 + q_3)}{2} \quad (2.84)$$

$$D_{i3,3} = J_3 + \frac{3 l_1^2 m_1}{4} + \frac{l_1^2 m_3}{4} + \frac{l_3^2 m_3}{4} - \frac{l_1^2 m_1 \cos(q_1 - q_2)}{2} - \frac{l_1 l_3 m_3 \cos(q_1)}{2} \quad (2.85)$$

$$D_{i3,4} = \frac{l_3 m_3 \cos(q_3)}{2} - l_1 m_1 \cos(q_1 + q_3) + \frac{l_1 m_1 \cos(q_2 + q_3)}{2} - \frac{l_1 m_3 \cos(q_1 + q_3)}{2} \quad (2.86)$$

$$D_{i3,5} = l_1 m_1 \sin(q_1 + q_3) - \frac{l_3 m_3 \sin(q_3)}{2} - \frac{l_1 m_1 \sin(q_2 + q_3)}{2} + \frac{l_1 m_3 \sin(q_1 + q_3)}{2} \quad (2.87)$$

$$D_{i4,4} = 2 m_1 + m_3 \quad (2.88)$$

$$D_{i5,5} = 2 m_1 + m_3 \quad (2.89)$$

Matice  $\mathbf{C}_{i3 \times 3}(\mathbf{q}, \dot{\mathbf{q}})$ :

$$C_{i1,2} = \left(-\frac{1}{4}\right) l_1^2 \sin(q_1 - q_2) m_1 (\dot{q}_2 + \dot{q}_3) \quad (2.90)$$

$$C_{i1,3} = \left(-\frac{1}{4}\right) l_1 (\dot{q}_3 l_3 m_3 \sin(q_1) + \dot{q}_2 l_1 m_1 \sin(q_1 - q_2) + \dot{q}_3 l_1 m_1 \sin(q_1 - q_2)) \quad (2.91)$$

$$C_{i2,3} = \left(\frac{1}{4}\right) l_1^2 m_1 \sin(q_1 - q_2) (\dot{q}_1 + \dot{q}_3) \quad (2.92)$$

$$C_{i3,3} = \left(\frac{1}{4}\right) l_1 (\dot{q}_1 l_3 m_3 \sin(q_1) + \dot{q}_1 l_1 m_1 \sin(q_1 - q_2) - \dot{q}_2 l_1 m_1 \sin(q_1 - q_2)) \quad (2.93)$$

Matice  $\mathbf{G}_i(\mathbf{q})$ :

$$G_i = \begin{bmatrix} g l_1 m_1 \sin(q_1 + q_3) + \frac{g l_1 m_3 \sin(q_1 + q_3)}{2} \\ -\frac{g l_1 m_1 \sin(q_2 + q_3)}{2} \\ g m_1 \left( \frac{l_1 \sin(q_1 + q_3)}{2} - \frac{l_1 \sin(q_2 + q_3)}{2} \right) + g m_3 \left( \frac{l_1 \sin(q_1 + q_3)}{2} - \frac{l_3 \sin(q_3)}{2} \right) + \frac{g l_1 m_1 \sin(q_1 + q_3)}{2} \\ 0 \\ 2 g m_1 + g m_3 \end{bmatrix} \quad (2.94)$$

$$B_i = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.95)$$

## 2.6. Řízení

Robot má  $n$  stupňů volnosti, ale pouze  $n-1$  akčních členů. Pro řízení těchto členů tedy musí mít  $n-1$  výstupů  $y$  řízených k nule. Tyto výstupy si můžeme definovat jako žádaný úhel torza robota, který zajistí moment potřebný pro chůzi robota. Další výstupy slouží k regulaci pohybu nohou. Pro  $dof=3$  :

Zaveďme  $\xi$  jako stavový prostor:

$$\xi_{2n \times 1} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \quad (2.96)$$

Model (2.50) pak lze přepsat ve formě:

$$\dot{\xi} = \mathbf{f}_{2n \times 1}(\xi) + \sum_{i=1}^m \mathbf{g}_i(\xi) u_i \quad (2.97)$$

$\mathbf{f}$  a  $\mathbf{g}_i$  jsou hladká vektorová pole.

Lieova derivace funkce  $h$  podél  $\mathbf{f}$  [3] je definována jako:

$$\mathcal{L}_f h(\xi) = \sum_{i=1}^n \frac{\partial h}{\partial \xi_i} f_i(\xi) \quad (2.98)$$

Mějme sloupcový vektor skalárních funkcí  $\mathbf{h}$ :

$$\mathbf{h} = [ h_1 \quad h_2 \quad \dots \quad h_m ]^T \quad (2.99)$$

Lieova derivace nabyde tvar:

$$\mathcal{L}_f \mathbf{h}(\xi)_{m \times 1} = \frac{\partial \mathbf{h}_{m \times 1}}{\partial \xi_{2n \times 1}} \mathbf{f}_{2n \times 1}(\xi) \quad (2.100)$$

$$\mathcal{L}_g \mathcal{L}_f \mathbf{h}(\xi)_{m \times m} = \frac{\partial (\mathcal{L}_f \mathbf{h})_{m \times 1}}{\partial \xi_{2n \times 1}} \mathbf{g}_{n \times m}(\xi) \quad (2.101)$$

kde:  $\mathbf{g} = [ \mathbf{g}_1 \quad \mathbf{g}_2 \quad \dots \quad \mathbf{g}_m ]$  podle (2.97),  $\mathbf{g}_i$  pro  $i = 1, 2, \dots, m$  jsou sloupce matice  $\mathbf{g}$ .

Derivace vyšších řádů lze vyjádřit rekurzivně, rozměry matic nadále platí podle (2.97), (2.100), (2.101) a (2.96):

$$\mathcal{L}_f^k \mathbf{h}(\xi) = \frac{\partial (\mathcal{L}_f^{k-1} \mathbf{h})}{\partial \xi} \mathbf{f}(\xi) \quad (2.102)$$

Zavedme si dále, že vektor  $\mathbf{y}$  je výstupem soustavy (2.97) a je funkcí stavů:

$$\mathbf{y} = \mathbf{h}(\xi) \quad (2.103)$$

Postupné derivace výstupu (2.103) získáme jako:

$$\dot{\mathbf{y}} = \mathcal{L}_f \mathbf{h}(\xi) + \mathcal{L}_g \mathbf{h}(\xi) \mathbf{u} \quad (2.104)$$

$$\ddot{\mathbf{y}} = \mathcal{L}_f^2 \mathbf{h}(\xi) + \mathcal{L}_g \mathcal{L}_f \mathbf{h}(\xi) \mathbf{u} \quad (2.105)$$

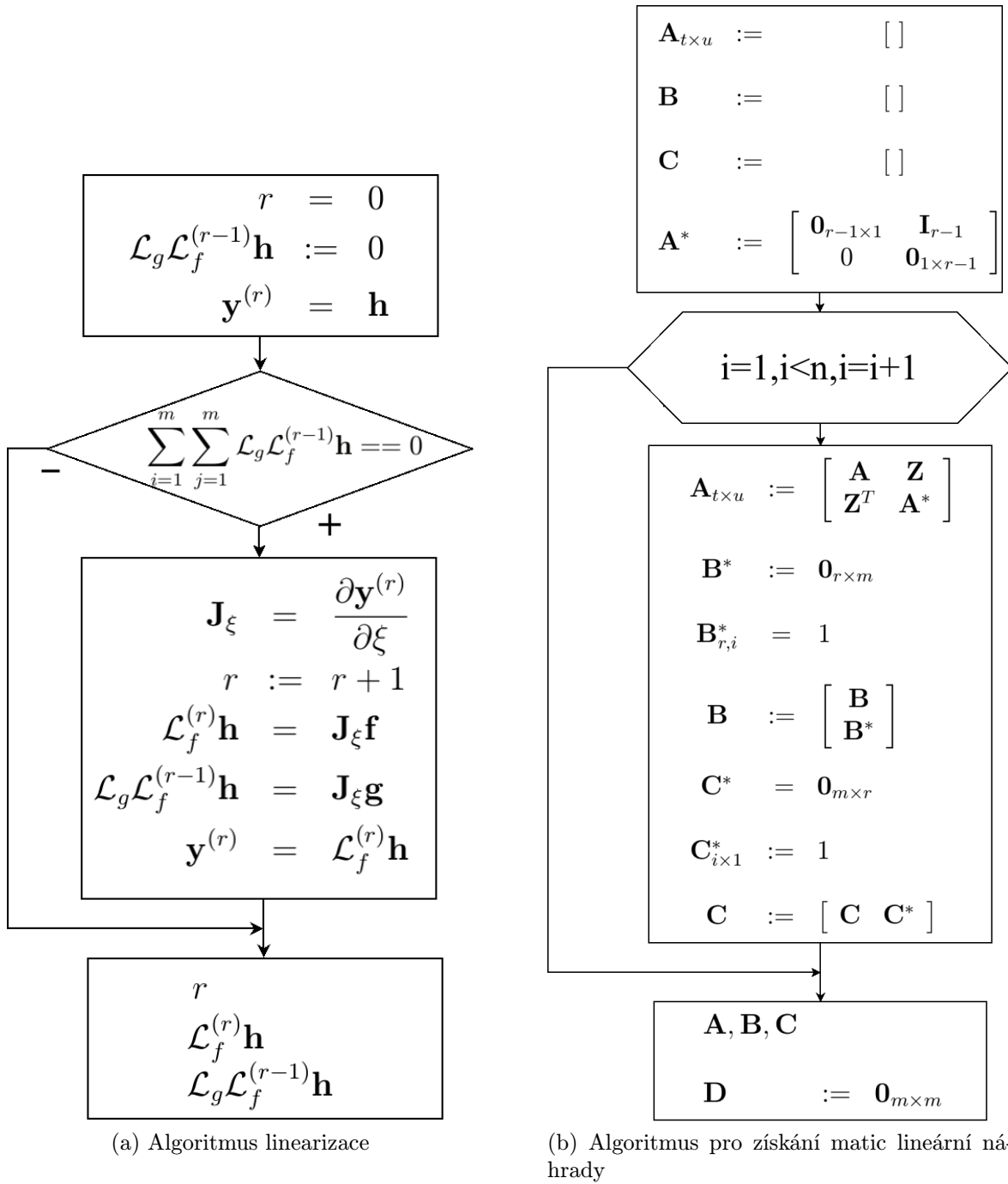
$$\mathbf{y}^{(r)} = \mathcal{L}_f^r \mathbf{h}(\xi) + \mathcal{L}_g \mathcal{L}_f^{(r-1)} \mathbf{h}(\xi) \mathbf{u} \quad (2.106)$$

Matice  $\mathcal{L}_g \mathcal{L}_f^{(r-1)} \mathbf{h}(\xi)_{m \times m}$  se nazývá Decoupling matrix a určuje křížové vazby mezi vstupy. Pokud není tato matice regulární, nelze provést vstupně výstupní linearizaci.

$$\mathbf{u} = \left( \mathcal{L}_g \mathcal{L}_f^{(r-1)} \mathbf{h}(\xi) \right)^{-1} (\mathbf{v} - \mathcal{L}_f^r \mathbf{h}(\xi)) \quad (2.107)$$

$$\mathbf{v} = \mathbf{y}^{(r)} \quad (2.108)$$

Využijeme-li těchto poznatků, můžeme definovat algoritmus, jehož výstupem jsou řád  $r$ , matice  $\mathcal{L}_g \mathcal{L}_f^{(r-1)} \mathbf{h}$  a sloupcový vektor  $\mathcal{L}_f^r \mathbf{h}(\xi)$ .



### 2.6.1. Linearizace modelu

Rozložením modelu (2.50) podle (2.97) dostaneme matice  $\mathbf{f}$ ,  $\mathbf{g}$ :

$$\mathbf{f}(\xi) = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{D}^{-1}(\mathbf{q}) [-\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})] \end{bmatrix} \quad (2.109)$$

$$\mathbf{g}(\xi) = \begin{bmatrix} \mathbf{0}_{n \times m} \\ \mathbf{B}(\mathbf{q}) \end{bmatrix} \quad (2.110)$$

Výsledný linearizovaný model má pak tvar:

$$\dot{\mathbf{z}}_{rm \times 1} = (\mathbf{A}_z)_{rm \times rm} \mathbf{z} + (\mathbf{B}_z)_{rm \times m} \mathbf{v} \quad (2.111)$$

$$\mathbf{y} = (\mathbf{C}_z)_{m \times rm} \mathbf{z} + (\mathbf{D}_z)_{m \times m} \mathbf{v} \quad (2.112)$$

### 2.6.2. 3 dof

Výstupní funkce je zvolena podle požadavků regulace robota. Požadavek  $(\mathbf{h}_2)_1$  zajišťuje řízení sklonu torza na žádanou hodnotu tak, aby svým náklonem dodávalo působením gravitace energii pro chůzi.  $(\mathbf{h}_2)_2$  klade zadní nohu před přední.

$$\mathbf{h}_2(\mathbf{q}) = \begin{bmatrix} q_3 - q_{3ref} \\ q_2 - q_{2ref} \end{bmatrix} \quad (2.113)$$

Po provedení symbolického výpočtu podle (2.5a) je výsledný řád derivace  $r = 2$ . Má tedy shodný řád jako řád nelineárního systému (2.50). Kdyby byl řád  $r$  menší, obsahoval by linearizovaný model skrytou dynamiku, u níž by se musela ověřovat stabilita.

Pro zjednodušení je výsledný  $\mathcal{L}_g \mathcal{L}_f^{(1)} \mathbf{h}$  zapsán s vytknutým dělitelem v následujícím tvaru a je k nalezení (5):

$$\mathcal{L}_g \mathcal{L}_f^{(1)} \mathbf{h}(\xi) = \frac{1}{den(\mathbf{q})} \mathcal{L}_g \mathcal{L}_f^{(1)} \mathbf{h}^* \quad (2.114)$$

$$\mathbf{z} = [z_1 \quad z_2 \quad z_3 \quad z_4]^T \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\mathbf{A}_z = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B}_z = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.115)$$

$$\mathbf{C}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \mathbf{D}_z = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

### 2.6.3. 5 dof

Řízení chůze 5 dof modelu je navrženo výstupní funkcí, která udržuje konstantní úhel torza  $q_5$  stejně jako v případě modelu s  $dof = 3$ . Dále je řízen úhel v kolenní stojné nohy k nule, aby byla co nejvíce natažená, když podmínka na úhel  $q_3$  táhne volnou nohu dopředu pokrčenou v kolenní o úhel  $q_{4ref}$ . Při došlápnutí nohy se role  $q_2$  a  $q_3$  vymění.

$$\mathbf{h}_5(\mathbf{q}) = \begin{bmatrix} q_1 \\ q_3 - q_{3ref} \\ q_4 - q_{4ref} \\ q_5 - q_{5ref} \end{bmatrix} \quad (2.116)$$

## 2.7. Ověření linearity dof=3

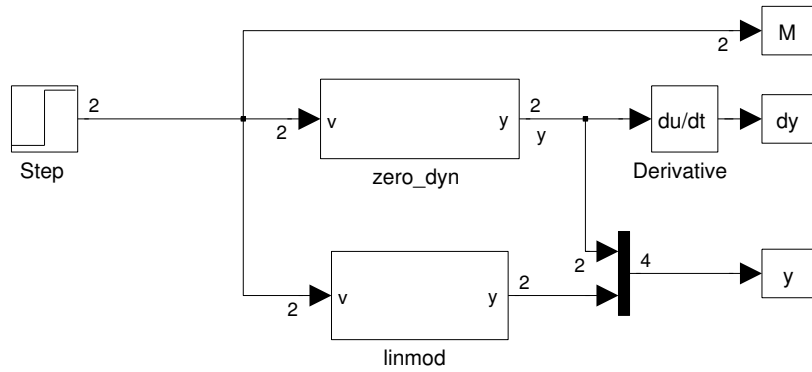
Jako ověření linearity modelu uvažme schéma (2.6). Jedná se o linearizovaný model blok *zero\_dyn* a jeho lineární náhradu *linmod*. Za předpokladu, že se jedná o identické systémy v nulových počátečních podmínkách tak, odezva na jednotkový skok musí být v obou případech stejná. Zvolme si počáteční podmínky takové, že lineární náhrada bude začínat v  $\mathbf{z} = \mathbf{0}_{4 \times 1}$ . Výstupy linearizovaného modelu a jejich derivace musí být stejné, tedy musí platit pro počáteční podmínky soustavy:

$$\begin{aligned} \mathbf{y} &= \mathbf{0}_{m \times 1} & \dot{\mathbf{y}} &= \mathbf{0}_{m \times 1} \\ \mathbf{h}_2 &= \mathbf{0}_{m \times 1} & \mathcal{L}_f \mathbf{h} &= \mathbf{0}_{m \times 1} \end{aligned} \quad (2.117)$$

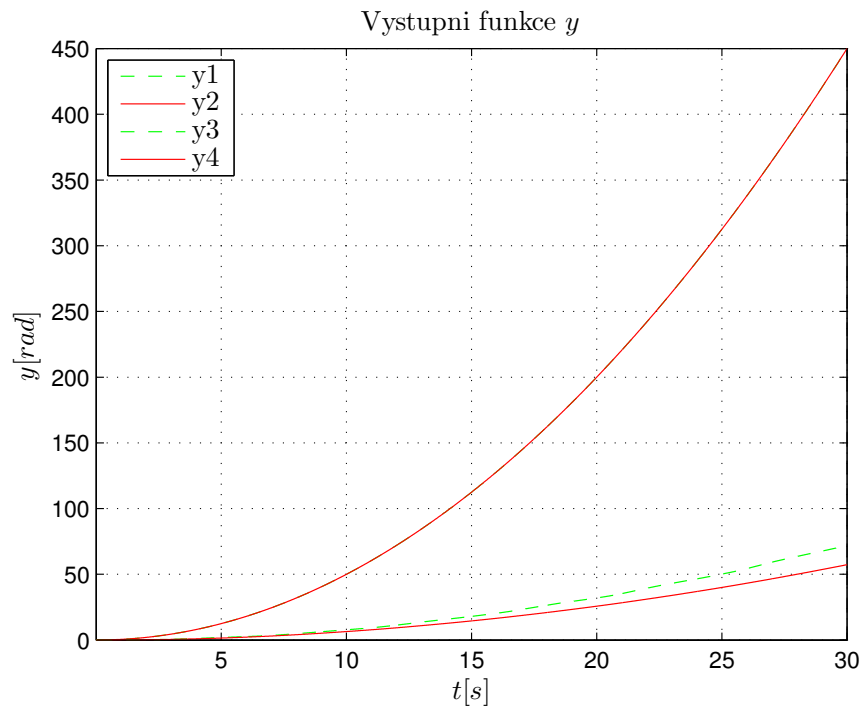
Snadno lze pak získat počáteční podmínky:

$$\begin{aligned} q_3 &= q_{3ref} & \dot{q}_3 &= 0 \\ q_2 &= q_{2ref} & \dot{q}_2 &= 0 \end{aligned} \quad (2.118)$$

## 2.7. OVĚŘENÍ LINEARITY DOF=3

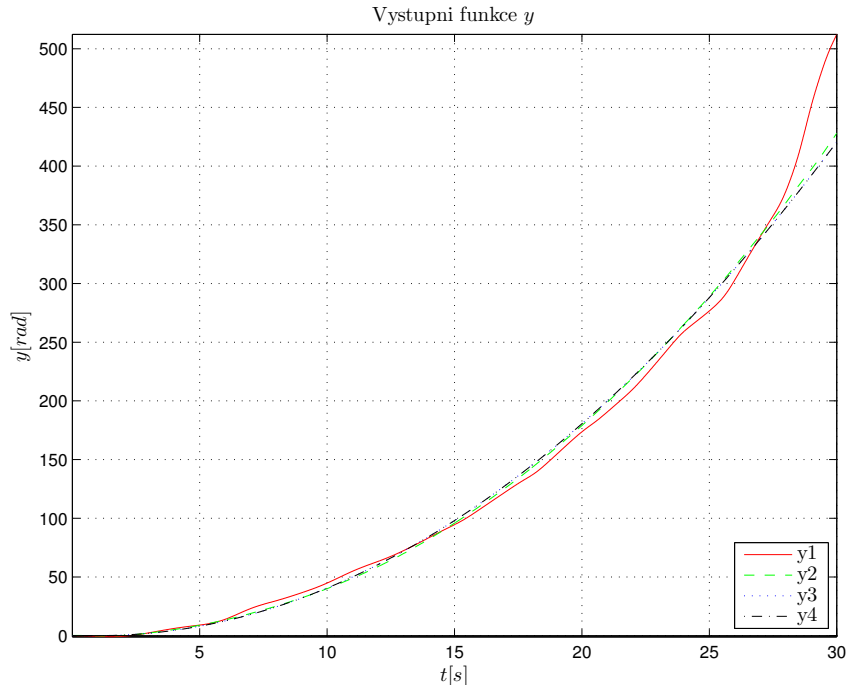


Obrázek 2.5: Schéma ověření linearity



Obrázek 2.6: Odezva na jednotkový skok





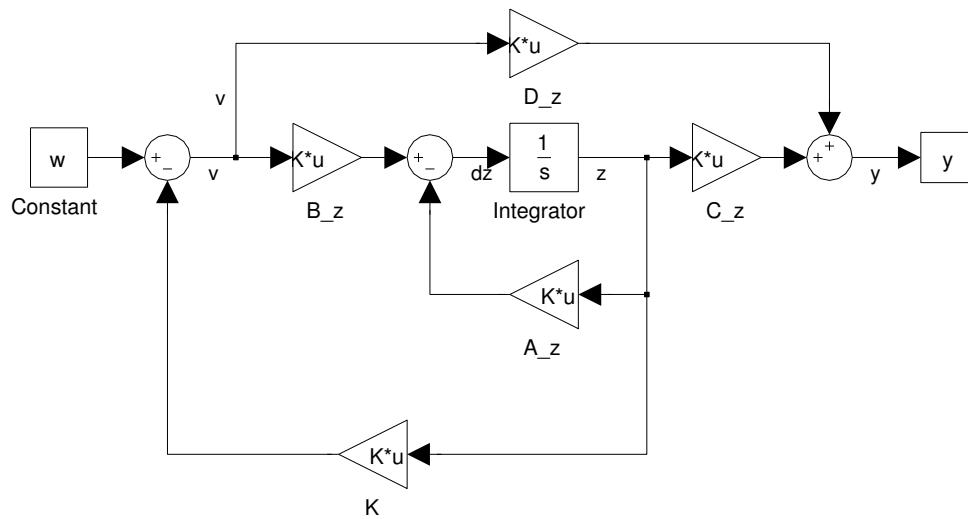
Obrázek 2.7: Odezva na jednotkový skok s konstantou navýstupu

Na grafu (obr.2.6) jsou  $y_3$ ,  $y_4$  identické a překrývající se výstupní průběhy lineární náhrady  $y_1$ ,  $y_2$  jsou pak výstupy linearizované soustavy. Je patrné, že výstupy soustavy jsou kvadratické, jak by se dalo očekávat z odezvy na jednotkový skok pro dvojitý integrátor. Zesílení je však  $< 1$ . Vezmeme-li výstup  $y_4$  jako referenční a podělíme výstupem  $y_1$  a  $y_2$  získáme konstanty zesílení, které jsou zahrnuty v simulaci (obr.3.2). Graf (obr.2.7) zobrazuje vynásobený výstup. Drobné odchylky od průběhu jsou způsobeny charakterem linearizace, která přidává virtuální energii do systému. V případě, že by byl vliv značnější, musela by se stanovit podmínka řešitelnosti.

## 2.8. Návrh regulátoru

Soustava se po linearizaci stane autonomním systémem. Chová se jako m na sobě nezávislých dvojitých integrátorů. Tím je umožněno řídit každý výstup samostatně za pomoci odpovídajícího vstupu. Použití m regulátorů, je jedna možnost. Použijeme-li 1 stavový regulátor, můžeme očekávat, že bude mít podobu nulové matice s  $1 \times 2$  prvkovými maticemi na diagonále.

Stavový regulátor využívá vazby přímo od stavu a působí na vstupu soustavy. Je tedy nutné znát stavy, nebo je umět zrekonstruovat pomocí rekonstruktoru stavu. V daném případě není systém pozorovatelný, obsahuje skrytou dynamiku. Mezi vnitřními stavy je ovšem známá závislost a lze je snadno dopočítat.



Obrázek 2.8: Stavový regulátor

Pro model ve tvaru (2.126):

$$\dot{\mathbf{z}} = \mathbf{A}_z \mathbf{z} + \mathbf{B}_z \mathbf{v} \quad (2.119)$$

$$\mathbf{y} = \mathbf{C}_z \mathbf{z} + \mathbf{D}_z \mathbf{v} \quad (2.120)$$

Rozšířením výstupu o vlastní derivace jsou získány nepozorovatelné stavy:

$$\mathbf{y}_z^* = [ y_1 \quad \dot{y}_1 \quad y_2 \quad \dot{y}_2 \quad \cdot \quad y_m \quad \dot{y}_m ] \quad (2.121)$$

$$\mathbf{y}_z^* = \mathbf{z} \quad (2.122)$$

Stavový regulátor má tvar:

$$\mathbf{v} = -\mathbf{K} \mathbf{z} \quad (2.123)$$

Dosazením (2.123) do (2.126):

$$\dot{\mathbf{z}} = (\mathbf{A}_z - \mathbf{B}_z \mathbf{K}) \mathbf{z} + \mathbf{B}_z \mathbf{K} \mathbf{w} \quad (2.124)$$

$$\mathbf{y} = (\mathbf{C}_z - \mathbf{D}_z \mathbf{K}) \mathbf{z} + \mathbf{D}_z \mathbf{K} \mathbf{w} \quad (2.125)$$

Protože z požadavku úlohy regulujeme výstup na žádanou hodnotu  $\mathbf{w} = \mathbf{0}_{m \times 1}$ , změní se soustava následovně:

$$\dot{\mathbf{z}} = (\mathbf{A}_z - \mathbf{B}_z \mathbf{K}) \mathbf{z} \quad (2.126)$$

$$\mathbf{y} = \mathbf{C}_z \mathbf{z} \quad (2.127)$$

K nastavení regulátoru je možné použít LQR algoritmus. Představuje kvadratické kritérium pro lineární systémy, které charakterizuje rovnice cenové funkce:

$$J_{cost} = \int_0^{\infty} (\mathbf{z}^T \mathbf{Q} \mathbf{z} + \mathbf{v}^T \mathbf{R} \mathbf{v} + \mathbf{z}^T \mathbf{N} \mathbf{v}) dt \quad (2.128)$$

Hledání minima cenové funkce vede na Riccatiho rovnici.

$$\mathbf{A}_z^T \mathbf{P} + \mathbf{P} \mathbf{A}_z - (\mathbf{P} \mathbf{B}_z + \mathbf{N}) \mathbf{R}^{-1} (\mathbf{B}_z^T \mathbf{P} + \mathbf{N}^T) + \mathbf{Q} = 0 \quad (2.129)$$

Vyřešením rovnice (2.129) získáme stavový regulátor.

$$\mathbf{K} = \mathbf{R}^{-1} (\mathbf{B}_z^T \mathbf{P} + \mathbf{N}^T) \quad (2.130)$$

Volba matic  $\mathbf{Q}$ ,  $\mathbf{R}$  a  $\mathbf{N}$  závisí na požadavcích, které klademe na systém. K řešení úlohy byly zvoleny  $\mathbf{Q}$  a  $\mathbf{R}$  jako jednotkové matice. Matice  $\mathbf{N} = \mathbf{0}_{m \times rm}$  je nulová, aby zůstal systém i nadále autonomní.

$$\mathbf{Q} = \begin{bmatrix} \frac{1}{z_{1max}^2} & 0 & 0 & 0 \\ 0 & \frac{1}{z_{2max}^2} & 0 & 0 \\ 0 & 0 & \frac{1}{z_{3max}^2} & 0 \\ 0 & 0 & 0 & \frac{1}{z_{4max}^2} \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} \frac{1}{v_{1max}^2} & 0 \\ 0 & \frac{1}{v_{2max}^2} \end{bmatrix} \quad (2.131)$$

S konstantami modelu  $dof = 3$

$$\begin{aligned}
 l_1 &= 1 & m & & J_1 &= 0.3 & kg.m^2 \\
 l_2 &= l_1 & m & & J_2 &= J_1 & kg.m^2 \\
 l_3 &= 1 & m & & J_3 &= J_1 & kg.m^2 \\
 m_1 &= 0.1 & kg & & q_{3ref} &= 0.16\pi & rad \\
 m_2 &= m_1 & kg & & q_{2ref} &= \frac{5\pi}{6} & rad \\
 m_3 &= 0.3 & kg & & & & 
 \end{aligned} \tag{2.132}$$

a natavením penalizačních matic (2.131)

$$\begin{aligned}
 z_{1max} &= 0.1 & rad \\
 z_{2max} &= 1.0 & rad.s^{-1} & v_{1max} &= 20 & Nm \\
 z_{3max} &= 0.1 & rad & v_{2max} &= 20 & Nm \\
 z_{4max} &= 1.0 & rad.s^{-1}
 \end{aligned} \tag{2.133}$$

Dostaneme stavový regulátor:

$$\mathbf{K} = \begin{bmatrix} 200 & 20\sqrt{2} & 0 & 0 \\ 0 & 0 & 40 & 4\sqrt{30} \end{bmatrix} \tag{2.134}$$

Aby byl systém stabilní, musí všechny póly ležet v levé polorovině „s“.

$$\mathbf{A}_z^* = \mathbf{A}_z - \mathbf{B}_z \mathbf{K} \tag{2.135}$$

Vyřešením charakteristického polynomu matice  $\mathbf{A}_z^*$  získáme póly soustavy, která je stabilní.

$$|s\mathbf{I}_m - \mathbf{A}_z^*| = 0 \tag{2.136}$$

$$\begin{aligned}
 s_1 &= -59.9666 \\
 s_2 &= -59.1359 \\
 s_3 &= -10.1461 \\
 s_4 &= -2.0011
 \end{aligned} \tag{2.137}$$

## 3. Simulace

### 3.1. Program

Program se skládá z více modulů. Jsou rozděleny podle funkce na část generování a část simulační. Ty představují 2 hlavní třídy *ModelGenerator2* a *TBiped*. Objektový model byl zvolen pro zapouzdření vlastností robota a metod nad těmito vlastnostmi do jednoho celku, ke kterému se snadno přistupuje. Tento typ programování podporuje program Matlab od verze 2008a.

TBiped.m	třída robota
ModelGenerator2.m	třída určená pro generování matematického modelu
biped.m	program GUI
shapedof3.m	defaultní 3 dof model bez ztrát
shapedof5.m	defaultní 5 dof model bez ztrát
constdof3.m	konstanty pro 3 dof model
constdof5.m	konstanty pro 5 dof model
inidof3.m	počáteční podmínky simulace pro 3 dof
inidof5.m	počáteční podmínky simulace pro 5 dof model
bipedmdl.mdl	schéma pro řešení solverem <i>sim</i>
constdof{3,5}.m	konstanty pro 3, 5 dof model
DH.m	Denavit-Hartenbergova transformace
subexpr.m	symbolické vytýkání z dynamických rovnic
biped.fig	deklarace grafiky GUI
+	soubory modelů

### 3.2. Generování modelu

Třída *ModelGenerator2.m* slouží k vygenerování linearizovaného modelu do m-filů rozdělených podle funkce. K výpočtu je použit symbolický toolbox, který ve formě výrazů počítá přímou kinematiku. Je zde implementován model v tvarových i relativních souřadnicích. Pro generování je lepší kloubový prostor s DH konvencí (obr.2.3), kdy se změnou počtu stupňů řetězce nastavuje počet stupňů volnosti. Výslednému modelu tak lze generovat libovolný počet kolen na nohou. Od relativních souřadnic bylo ovšem upuštěno z důvodu složitějšího řízení a přepočítávání poipaktových souřadnic. Model v relativních souřadnicích lze stále vygenerovat, ale bez řízení. Tato možnost však není podporována z GUI.

Z rovnic přímé kinematiky se získají derivováním podle času rychlosti kartézských souřadnicích jednotlivých kloubů a koncových bodů. Výpočtem potenciální (2.14) a ki-

netické energie (2.54) těchto bodů se sestaví Lagrangeovy rovnice (2.28). Tento postup je proveden pro model fáze zhoupnutí a pro impaktní model zvlášť.

Z důvodu, že symbolický toolbox nepodporuje derivaci funkce závislé na parametru jako je například  $\mathbf{q}$ , je třeba obejít tento problém, chceme-li derivovat, aniž by toolbox vyhodnotil výraz jako konstantu. První přístup je zadefinovat názvy proměnných s parametrem času například  $q_1(t)$ . Pak výsledek derivace dostaneme  $diff(q_1(t), t)$ , který se poté příkazem *subs* nahradí za  $dq_1(t)$ , popřípadě po dokončení všech derivování se časově proměnné nahradí  $q_1(t) \rightarrow q_1$ . Tento postup je bezpečný a podává korektní výsledky, dokud pracuje se stejnou verzí programu Matlab. Jádro symbolického toolboxu je tvořeno programem Maple. Jeho verze se mezi verzemi Matlabu liší. Forma zápisu vyšších derivací se liší zcela. Použitý zápis kódu v třídě generátoru modelu *ModelGenerator.m* byl podporován verzí Matlab R2009a a vyšší.

Druhá možnost, jak řešit danou úlohu, je použití jakobiánu, kde vektorovou funkci závislou na  $\mathbf{q}$  poderivujeme podle  $\mathbf{q}$  a výsledek vynásobme  $\dot{\mathbf{q}}$ .

$$\frac{d}{dt}\mathbf{f}(\mathbf{q}) = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial (\mathbf{q})} \dot{\mathbf{q}} = \dot{\mathbf{f}}(\mathbf{q}, \dot{\mathbf{q}}) \quad (3.1)$$

Zde  $\dot{\mathbf{q}}$  není konkrétní hodnota, ale obecné proměnná jako  $dq$ . Druhá derivace je ovšem závislá na první, proto je třeba zohlednit:

$$\frac{d}{dt}\dot{\mathbf{f}}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{\partial \dot{\mathbf{f}}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}} \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} \quad (3.2)$$

Vygenerovaný model může být mít dosazené konstanty při generování. Výhodou byla větší rychlost generování, byla-li inverze matice  $\mathbf{D}(\mathbf{q})$  počítána symbolicky. Tento postup je příliš dlouhý a neefektivní. Výsledek generování pro  $dof = 5$  vydá svým zápisem na desítky stran formátu A4 i po provedení víceúrovňového vytýkání a navíc se počítá v každém kroku, že podstatně výhodnější postup je počítat inverzi v každém kroku simulace, přičemž se provádí inverze nad maticí reálných čísel s rozměrem  $5 \times 5$ , což je jistě rychlejší, než přepočítaná inverze symbolicky.

Model se tedy generuje s nedosazenými konstantami, které lze při simulaci měnit v souboru i za běhu simulace a způsobit tak poruchu v soustavě. Název výstupního souboru s modelem je možné v GUI nastavit. Defaultní modely zachovávají konvenci:

$$shape\{dof\}\{postfix\}.m$$

kde postfix určuje, k čemu slouží daný modul modelu:

<i>imp</i>	-	přepočet poimpaktivního stavu (2.67)
<i>lin</i>	-	linearizace modelu (2.107)
<i>Zlin</i>	-	lineární náhrada (2.111)
<i>control</i>	-	regulátor (2.130)
<i>y</i>	-	výstupní funkce (2.103)

Zápis do souborů je zvolen z praktických důvodů. Je možné ukládat výsledky do souboru typu *.mat*, ale tento postup nese nevýhody, jako například potřeba programu Matlab na zobrazení obsahu souboru, popřípadě nelze zavolat jako funkci. Některé soubory jako funkce psané nejsou. Jsou vytvořeny jako skript a vkládají se na místo použití jménem souboru. Tyto soubory bylo možné do *mat-file* zapsat, ale vzhledem vybrané formě zápisu jsou všechny moduly modelu ukládány do souboru. Zápis matic je ve tvaru:

NazevMatice(i,j)=hodnota

kvůli výstupnímu formátu funkce `char()` pro argument symbolické matice. Výstup je nekompatibilní s programem Matlab a je tedy nutné převádět každý prvek matice zvlášť. Vzhledem k tomu, že matice obsahují velké množství nul, jsou vypisovány jen nenulové prvky dosazené do nulové matice příčného rozměru.

Třída *ModelGenerator2* umožňuje zápis každého modulu samostatně, což je výhoda při ladění programu. Metody této třídy jsou popsány v následující tabulce:

		<code>ModelGenerator2(n)</code>	konstruktor třídy
		<code>write_matrix(matrix)</code>	zápis matice do souboru
$[A, B, C, D]$	=	<code>lin(obj,r)</code>	výpočet lineární náhrady
		<code>write_latex_matrix(matrix)</code>	zápis do latexu
		<code>generate</code>	generování modelu
		<code>write_control</code>	zápis regulátoru
<code>out</code>	=	<code>write_out</code>	zápis výstupní funkce
<code>out</code>	=	<code>write_imp</code>	zápis impulzního modelu
<code>out</code>	=	<code>write_Zlin</code>	zápis lineární náhrady
<code>out</code>	=	<code>write_lin</code>	zápis linearizační funkce
<code>out</code>	=	<code>write_fcn</code>	zápis dynamiky
<code>out</code>	=	<code>write_auto</code>	automatické generování
		<code>set_dof(n)</code>	nastavení počtu stupňů volnosti
		<code>set_fcn_name(fcn_name)</code>	nastavení výstupního jména
		<code>set_const(bool)</code>	přepínač k dosazení konstant

### 3.3. Simulátor

Třída *TBiped.m* představuje objektový model simulátoru pro vygenerovaný model. Využívá svých metod k nastavení počátečních podmínek, počtu kroků k ujití etc. Hlavním úkolem této třídy je zajistit spouštění Simulinkového modelu pro každý krok robota s poskytnutím počátečních podmínek a impaktní dynamiky mezi kroky. Na konci simulace jsou v instanci třídy uloženy výsledky sestavené postupným přidáváním po každém kroku. Tyto výsledky si dále přebírá GUI ke zpracování. Data ze simulace jsou poskytnuty jak ve formě kloubových souřadnic a rychlostí, tak ve formě poloh koncových bodů a těžiště pro spuštění animace. S opětovným spuštěním simulace se data předchozí ztrácí.

Metody třídy *TBiped.m*

	<code>init(<math>\mathbf{q}_0, \dot{\mathbf{q}}_0, \mathbf{M}_0</math>)</code>	inicializace robota
<code>[<math>\mathbf{t}_{\text{real}}, \mathbf{q}_{\text{real}}</math>]</code>	<code>interp_result(<math>\mathbf{t}, \mathbf{q}</math>)</code>	interpolace výsledků na reálný čas při $T_s \neq \text{konst}$
	<code>TBiped(dof)</code>	konstruktor třídy <i>TBiped</i>
<code>v</code>	<code>controler(<math>\mathbf{q}</math>)</code>	regulátor R (obr.3.1)
<code>u</code>	<code>lin(<math>\mathbf{q}</math>)</code>	vstupně-výstupní linearizace
<code>y</code>	<code>output(<math>\mathbf{q}</math>)</code>	funkce výstupu z robota
<code>[   value   isterminal   direction ]</code>	<code>events(<math>\mathbf{t}, \mathbf{q}</math>)</code>	
<code>out</code>	<code>fun(<math>\mathbf{q}</math>)</code>	volitelná funkce obsahující dynamiku robota
<code><math>\mathbf{p}</math></code>	<code>q2p(<math>\mathbf{q}</math>)</code>	přímá kinematika robota
<code>out</code>	<code>cog(<math>\mathbf{q}</math>)</code>	výpočet těžiště robota
<code>out</code>	<code>cogs(<math>\mathbf{q}</math>)</code>	vrátí těžiště všech částí robota
	<code>draw(<math>\mathbf{q}</math>)</code>	vykreslení robota podle kloubových souřadnic
	<code>walk</code>	spustí simulaci chůze robota
<code>[<math>\mathbf{q}_p, \dot{\mathbf{q}}_p</math>]</code>	<code>switch_mod(<math>\mathbf{q}</math>)</code>	přepočít souřadnic v impaktní fázi

Pomocné metody:

<code>set_Ts(<math>T_s</math>)</code>	perioda vzorkování
<code>set_tsim(<math>t_{\text{sim}}</math>)</code>	čas simulace
<code>set_tstep(<math>t_{\text{step}}</math>)</code>	max čas simulace na 1 krok
<code>set_steps2go(<math>steps2go</math>)</code>	počet kroků k ujití
<code>set_q0(<math>\mathbf{q}_0</math>)</code>	vektor inicializačních úhlů
<code>set_dq0(<math>\mathbf{dq}_0</math>)</code>	vektor inicializačních rychlostí
<code>set_dof(<math>dof, \mathbf{q}_0, \mathbf{dq}_0, \mathbf{M}_0</math>)</code>	nastavení dof s inicializací
<code>set_torzoref(<math>torzoref</math>)</code>	žádaná hodnota úhlu torza
<code>set_endqref(<math>endqref</math>)</code>	žádaná hodnota úhlu volné nohy vzhledem k torzu
<code>set_kneeref(<math>kneeref</math>)</code>	žádaná hodnota úhlu v kolenu
<code>set_fcn(<math>fcn</math>)</code>	nastavení funkce k simulaci
<code>set_SW_interp(<math>bool</math>)</code>	zapínání interpolace - 1, vypnuto - 0

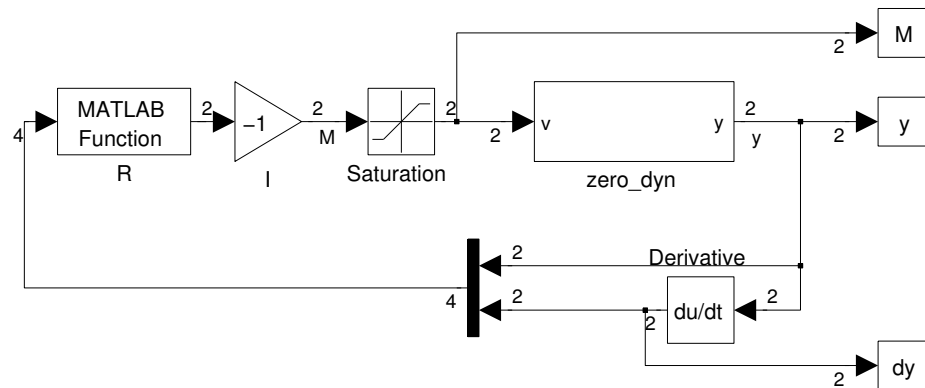


Jádro programu Matlab Simulink je voláno s lokálním workspace. V metodách *lin*, *fun*, *controler*, *output* jsou volány moduly modelu ze simulinku. Zde je zastoupena jako Matlab Function. Funkce pak obsahuje např.:

$$\text{eval}([\text{char}(\text{obj.fn}) \text{'Zlin'}])$$

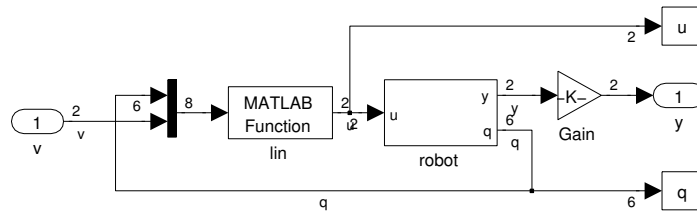
Tímto způsobem je možné libovolně měnit volané funkce z jendoho *mdl* souboru. K maticovým signálům proměnné velikosti je zajištěna spolupráce s dynamickou změnou volaných funkcí v blocích regulátoru R, linearizace, dynamiky robota a výstupní funkce. Impaktní dynamika se v Simulinku nevyhodnocuje.

Simulační schéma (obr.3.1) představuje regulační smyčku s regulátorem R. Je zde použita saturace na sražení momentových impulzů, což je patrné z výsledků simulace (obr.3.7). Tento skok se objevuje i na ostatních průbězích.

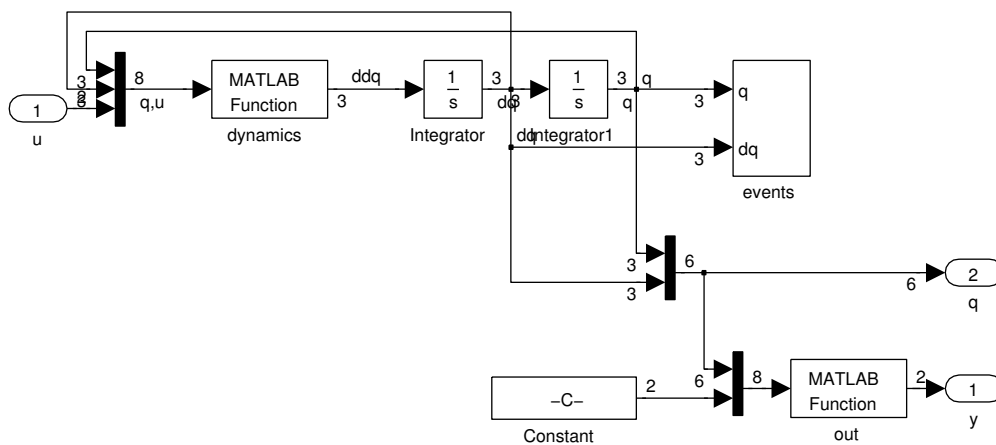


Obrázek 3.1: Simulační schéma

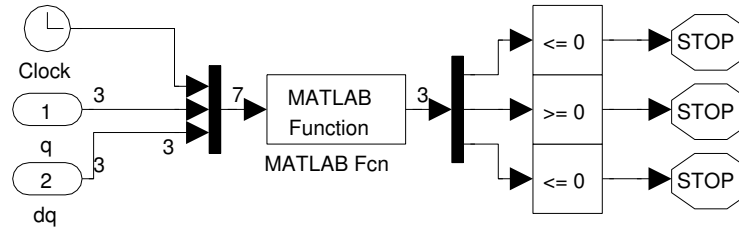
Blok *zero\_dyn* obsahuje linearizovaný model robota a podle žádané výstupní funkce  $y(q)$  představuje nulovou dynamiku systému. Vstupní momenty  $M$  řídí výstupy  $y$  k nule. Z důvodu potřeby pozorovatelného systému pro LQR regulaci jsou výstupní hodnoty derivovány jako  $\dot{y}$  a tak jsou získány nepozorovatelné stavy systému (2.111). Čísla u signálů jsou rozměry signálových vektorů. Tyto rozměry odpovídají pro simulaci s  $dof = 3$ . Schéma je vytvořeno s dynamicky měnitelným rozměrem stavového prostoru, proto je možné použít stejný model pro simulaci s  $dof = 5$ .

Obrázek 3.2: Blok *zero\_dyn*

Z výstupního vektoru bloku *biped*, který obsahuje dynamiku robota, vychází druhé derivace kloubových proměnných. Integrací se získávají jednotlivé vektory stavů, jejichž časové průběhy se vrací do workspace po ukončení simulace.

Obrázek 3.3: Blok *robot*

Blok *events* vyhodnocuje poruchy jako je pád robota na zem, nebo impakt nohy. Využívá stejnou metodu třídy *TBiped.m*, jako při použití solveru bez programu Simulink. Stop bloky zastavují simulaci při detekci příslušné události. Pro zjištění impaktu nohy je použita podmínka (3.4.1). Vyhodnocování doteku se zemí vyžaduje vytvořit pásmo necitlivosti v okolí počátku, aby se při míjení nohou nevyhodnotila chybná událost. Naopak tato necitlivost může způsobovat pozdní detekci, což se projeví odskoky nebo více smyčkami na fázových trajektoriích.

Obrázek 3.4: Blok *events*

## 3.4. Grafické rozhraní

GUI se spouští příkazem *biped*.

Nastavení solveru zajišťuje rozhraní programu Simulink přímo v modelu *bipedmdl.mdl*. Může se použít algoritmus s pevným i proměnným krokem simulace s definováním absolutních a relativních tolerancí výpočtu. Pokud je použita metoda s nekonstantním krokem, výsledky dílčích simulací jsou převzorkovány s periodou  $T_s$ , kterou lze zvolit v GUI. Je nutné však použít volbu *interpolate*. Průběhy se interpolují polynomem pomocí knihovní funkce *interp1*. Převzorkování probíhá současně nad celými maticemi výsledků kloubových souřadnic, momentů etc. V základním nastavení je zvolena metoda konstantním krokem Runge Kutta 4.řádu.

### 3.4.1. Chůze

V režimu *Walk* je jednou provedena simulace podle zadaných nastavení a spustí se animace na základě simulačních výsledků. Po spuštění se na informačním panelu ve spodní části zobrazí text *běží*. Text se nezobrazí ihned z důvodu, že simulace bere výkon procesoru a GUI je obslouženo v mezičase. Pro běh simulace nemusí být mdl soubor otevřen, přesto je vhodné při větším počtu kroků pozorovat probíhající výpočet tím, že při každém spuštění solveru se na stavové liště modelu objeví progress bar kompilace modelu.

Doba výpočtu se může lišit v závislosti na počtu kroků, které má robot ujit *Steps* a periodě vzorkování  $T_s$ . Při volbě malé periody je výsledná doba simulace delší a výsledky přesnější. Pokud je vybrán solver s proměnnou délkou kroku, je třeba uvážit, že pro hodnoty  $T_s > 0,1s$  je po převzorkování interpolovaných průběhů na konstantní délku výrazně snižena přesnost. Pro  $T_s < 0,01s$  je rozměr výstupních dat výrazně zvyšován. Je třeba na počkat, než se simulace dokončí. Je-li vybrána volba *animace*, pak se ihned po dokončení výpočtu zobrazí animace chůze robota v horním grafu. Na závěr je do grafu vložena trajektorie těžiště.

Se zvoleným malým  $T_s$  výsledná animace zjemní krok a znatelně zpomalí. Aby se předešlo čekání, může se průběh zastavit tlačítkem *Stop* a změnit hodnotu *Drop*. Ta

slouží pro redukci zobrazených kroků a je defaultně nastavena na 10, vykreslí se tedy jen každý 10. prvek výstupních vektorů. Toto nastavení platí pouze pro animaci, ne pro grafy. K opětovnému přehrání slouží tlačítko *Play*. Naopak pokud je animace velmi rychlá, je vhodné nastavit menší  $T_s$ , nebo menší *Drop*.

S vybraným parametrem *show figs* se zobrazí grafy veličin kloubových proměnných  $\mathbf{q}$ , rychlostí kloubových proměnných  $\dot{\mathbf{q}}$ , řídicích momentů  $\mathbf{M}$ , vstupních momentů robota  $\mathbf{u}$  a výstupní funkce  $\mathbf{y}$ . Tyto grafy lze vyvolat také po kliknutí na příslušné tlačítko. Graf  $\mathbf{u}$  se zobrazuje zároveň s grafy  $\mathbf{M}$ .

Tlačítkem *q orb* jsou vykresleny fázové trajektorie úhlů  $\mathbf{q}$  a jejich rychlostí  $\dot{\mathbf{q}}$ . Současně s nimi se objeví Poincarého mapy, které ukazují asymptotickou konvergenci k meznímu cyklu.

Stisknutím *Gait* je vygenerován řez animací chůze v externím okně, kromě toho je zobrazen ve spodním grafickém poli.

*Reset view* resetuje okno animace a inicializuje robota do výchozí polohy podle polí  $q_0$  a  $dq_0$ .

Pole  $q_0$  je zvláštní svým nastavením. Požaduje vstup ve tvaru vektoru se závorkami tento tvar může být pro  $dof = 3$ :

$$\left[ q_1 * pi \quad q_2 * pi \quad q_3 * pi \right] \text{ nebo } \left[ q_1 \quad q_2 \quad q_3 \right]$$

Je interpretovaný, tudíž nezáleží na tom, jestli zde řetězec *pi* figuruje, nebo ne. V obou případech je vstup chápán jako násobek  $\pi$  [rad]. Důležitý rozdíl je, že pro  $dof = 3$  je vstupní vektor vyhodnocován jako tvarové souřadnice (obr.2.4), zatímco pro  $dof = 5$  jako relativní souřadnice (obr.2.3). Tato zvláštnost je způsobena nepříjemnou vlastností tvarových souřadnic, chceme-li nastavit požadovaný tvar robota, kdy úhel mezi stojnou nohou a zemí je závislý na  $q_1$ ,  $q_2$  a  $q_5$ . V relativních souřadnicích je však daný úkol snadný a po přepočtu podle (2.25) se nastaví jako počáteční podmínka pro simulaci.

Vektor rychlostí  $dq_0$  je přímo v [rad.s<sup>-1</sup>], jak pro  $dof = 3$ , tak pro  $dof = 5$ . Důležitá je pouze forma zadání v řádkovém vektoru se závorkami a rychlosti jsou jen pro tvarové souřadnice.

Input box *tstep* nastavuje maximální dobu simulace pro 1 krok. Je výhodný při ověřování linearity linearizovaného modelu. Pokud je ovšem doba  $T_{sim}$  kratší než *tstep*, simulace končí. Tato podmínka je jednou z událostí *events* v metodě třídy *TBiped* a vyhodnocená v bloku *events* (obr.3.4).

Pokud je vygenerovaných více modelů, je možné změnit typ modelu ve vstupu *model*. Vyhodnocuje se jako ukazatel na funkci. Podle jeho názvu se pak hledají moduly modelu s přidáním postfixy podle (3.2).

### Zobrazení animace

Pro zobrazení kloubových souřadnic do kartézských je použita přímá kinematika.

Červený bod v zobrazení značí těžiště robota, které je dáno následujícím vztahem.

$$\mathbf{p}_{\text{cog}} = \frac{\sum_{i=1}^n m_i \mathbf{p}_{\text{gi}}}{\sum_{i=1}^n m_i} \quad (3.3)$$

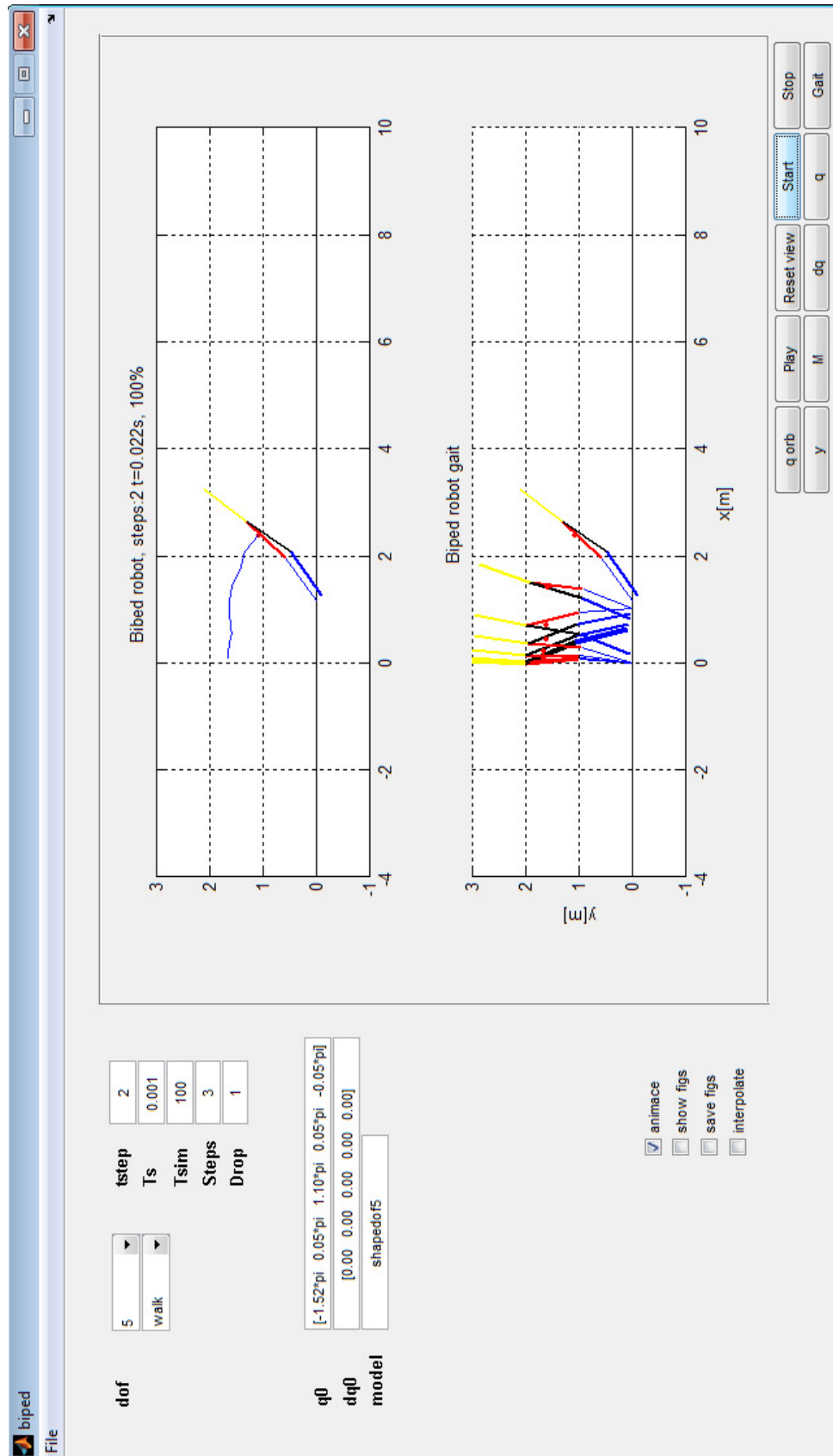
kde:  $\mathbf{p}_{\text{gi}}$  - těžiště i-té části robota  
 $m_i$  - hmotnost i-té části robota

### 3.4.2. Generate

Generování modelu pomocí GUI. Vybráním počtu *dof* se nastaví druh výstupního modelu. Volba *Subexpresion* měla smysl při generování třídou *ModelGenerator* nyní je ponechána pro případ, že se vyskytne složitý výraz s potřebou symbolického zjednodušení. V poli *Output m-file* se zadává jméno hlavního souboru výsledného modelu. K tomuto jménu se pak přidávají postfixy podle příslušného modulu modelu (3.2).

Vybrané funkce GUI:

	qorb_Callback	vykreslí orbity mezních cyklů kloub. proměnných
	gait	zobrazí chůzi robota
qshape	= rel2shape(qrel)	přepočít souřadnic
qrel	= shape2rel(qshape)	přepočít souřadnic
	in_dof_Callback	změna počtu dof, reinitializace $\mathbf{q}_0$ , $\dot{\mathbf{q}}_0$
	draw_dq	vykreslení úhlových rychlostí
	draw_q	kloubové souřadnice
	draw_y	vykreslení výstupů a fázových map
	draw_M	vykreslení řídicích momentů
	draw_u	vykreslení vstupních momentů
	draw_phase'diag	výpočet fázového diagramu
	draw	vykreslení animace
	CoG	zobrazení trajektorie těžiště
	start_Callback	zavaděč simulace)
	reminder	zvukové upozornění
	init_draw	překreslení robota
	biped_OpeningFcn	inicializace GUI

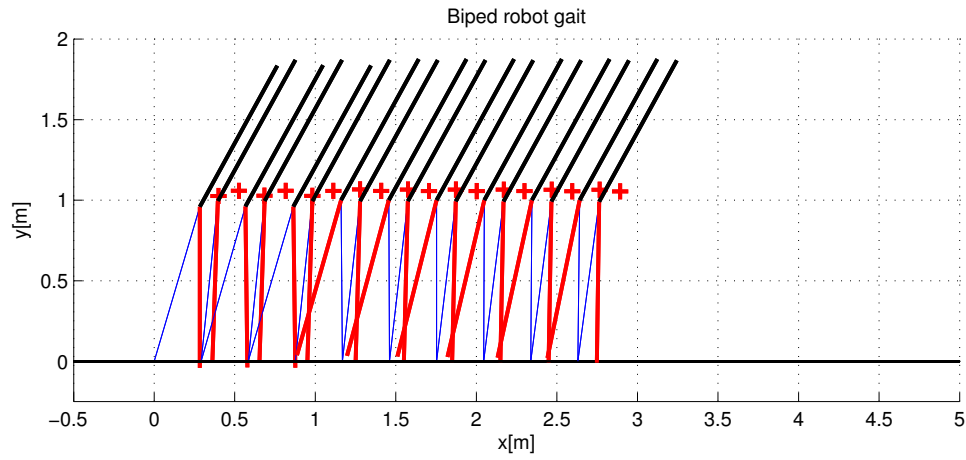


Obrázek 3.5: Vzhled GUI

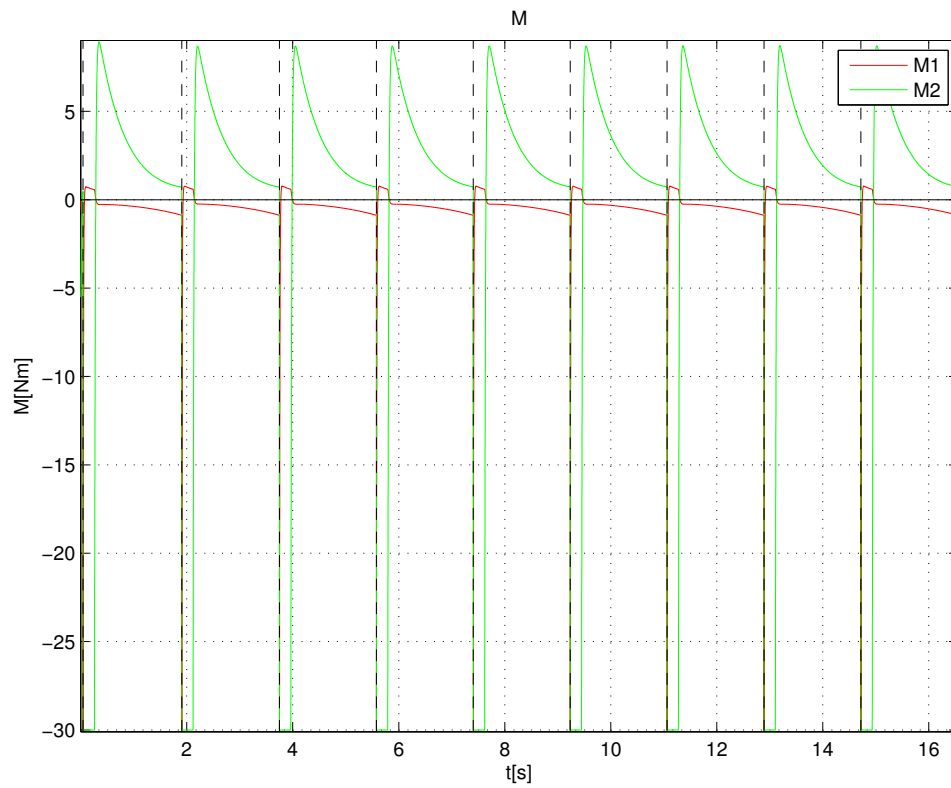
### 3.5. Grafy 3 dof

Simulace 10 kroků provedena metodou Runge Kutta 4.řádu,  $T_s = 1ms$ ,

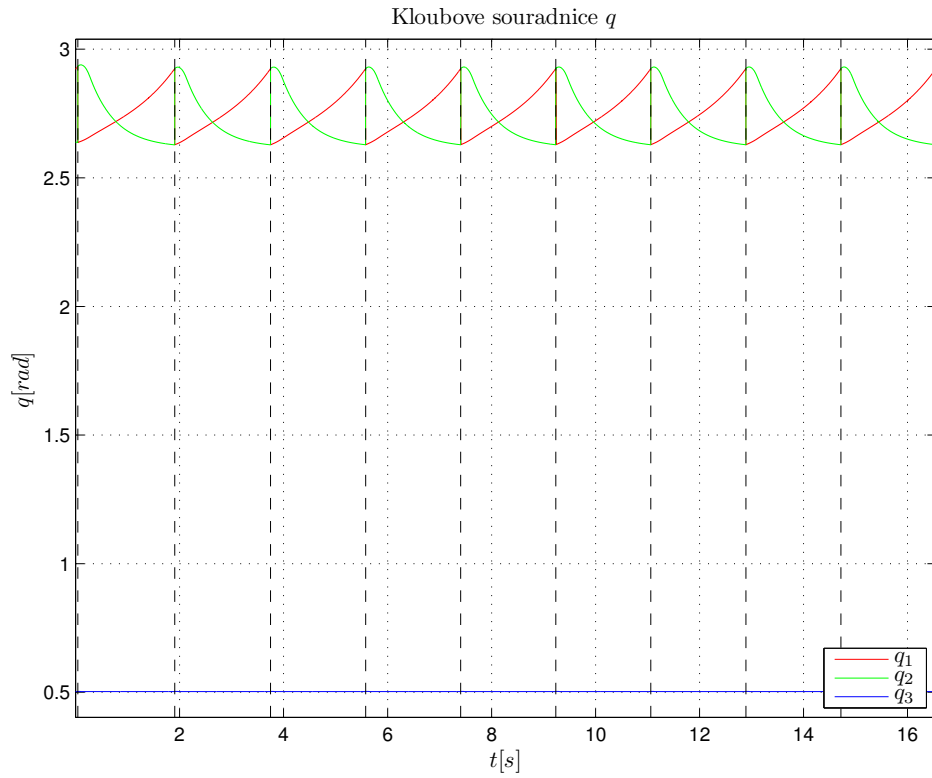
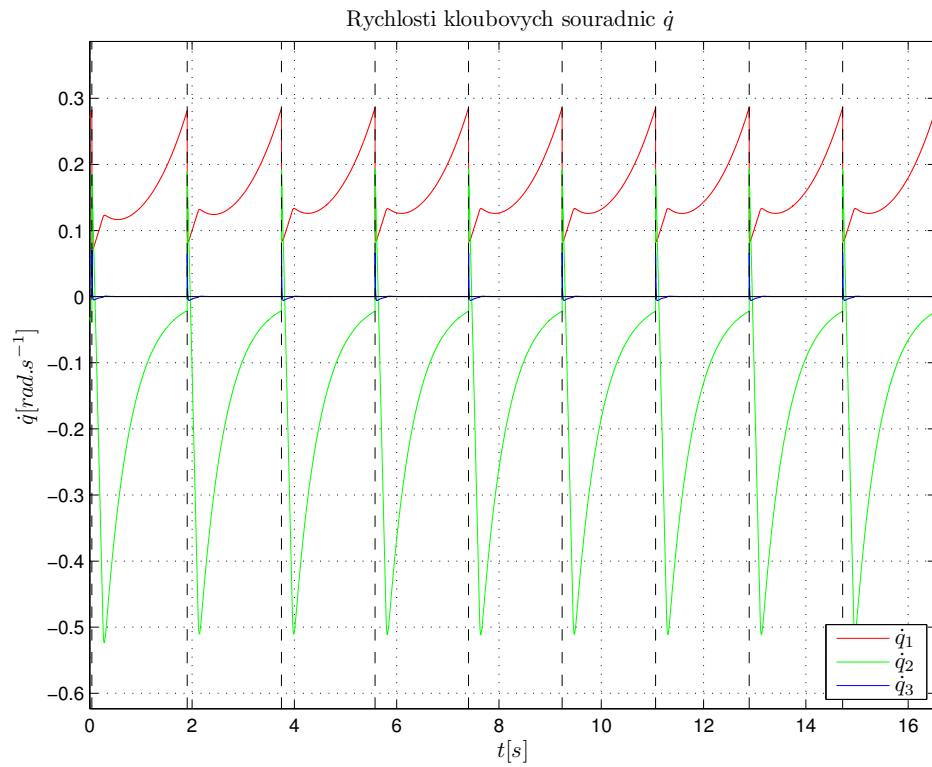
$$q_0 = [ 0,93\pi \quad 0,84\pi \quad 0,16\pi ] \quad \dot{q}_0 = [ 0,27 \quad -0,03 \quad 0,00 ]$$



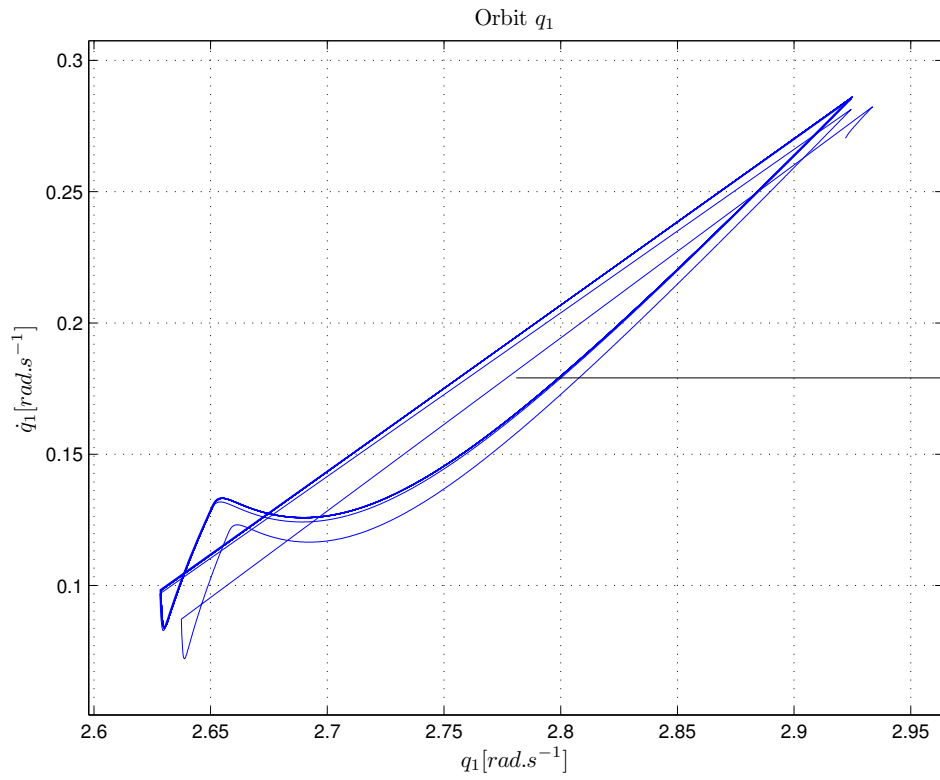
Obrázek 3.6: Chůze robota



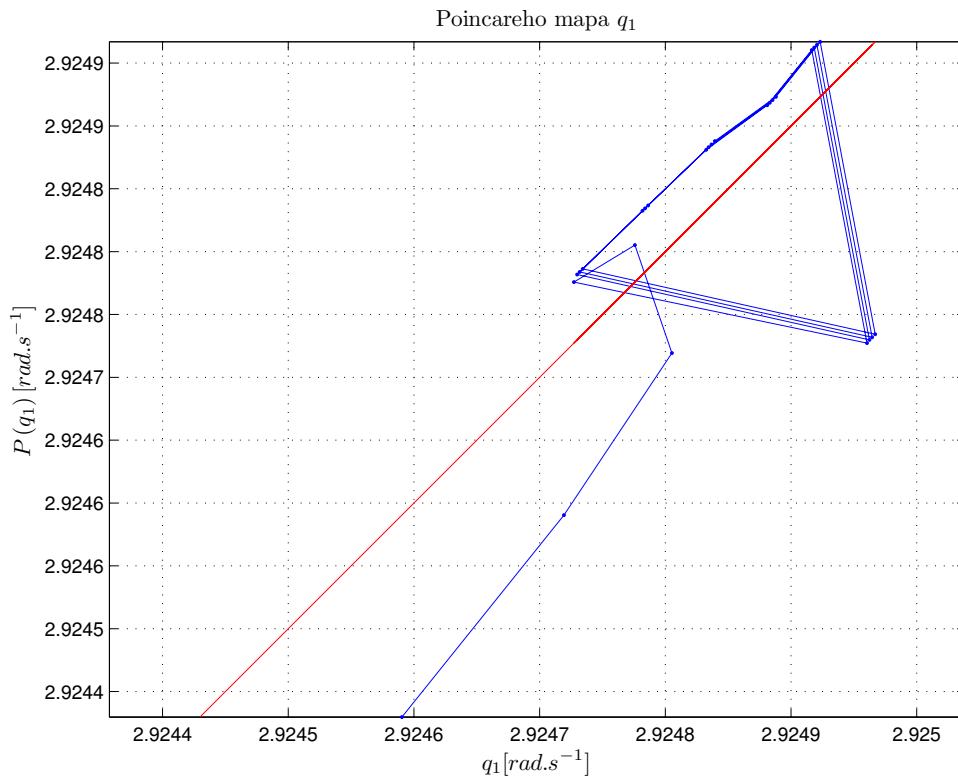
Obrázek 3.7: Průběh řídicích momentů  $M$

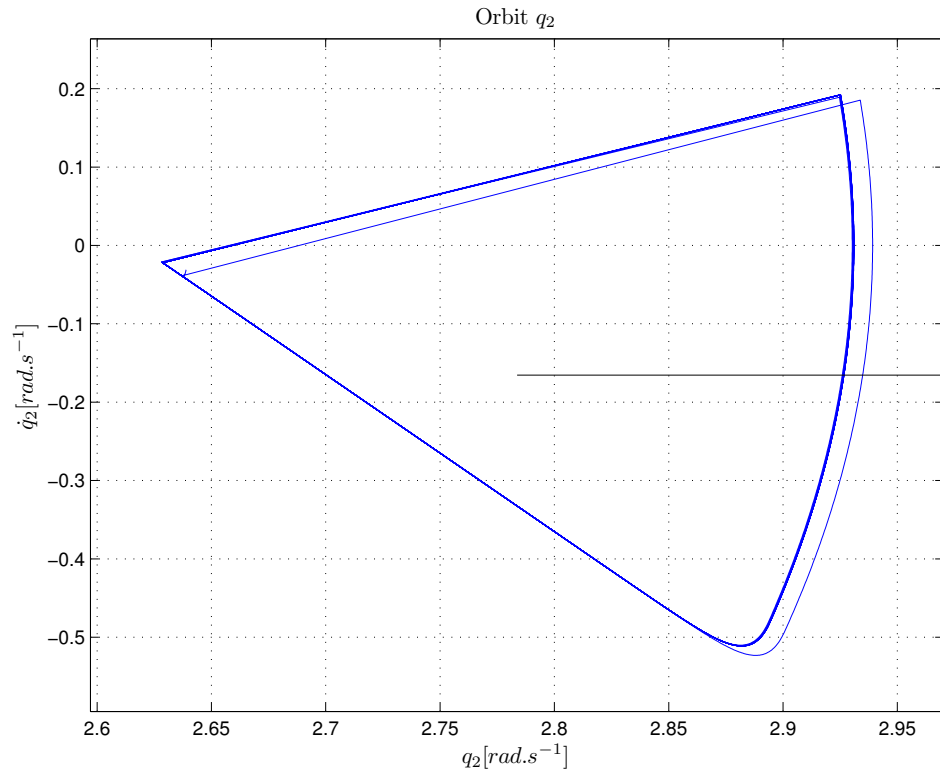
Obrázek 3.8: Průběh kloubových souřadnic  $q(t)$ Obrázek 3.9: Průběh rychlostí kloubových souřadnic  $\dot{q}(t)$



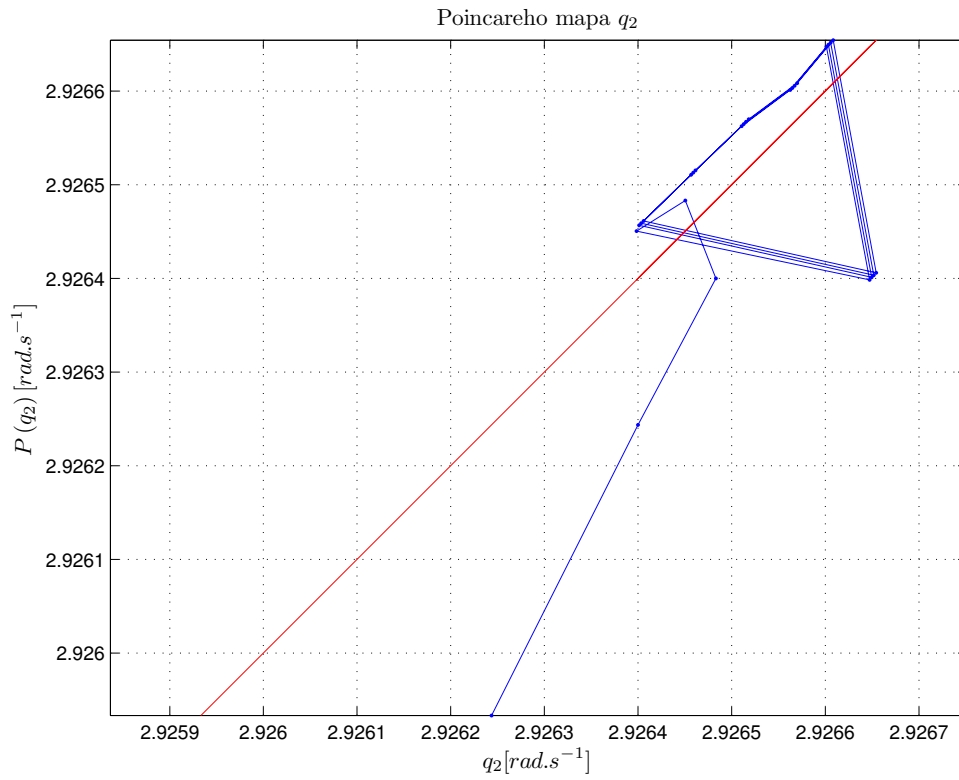


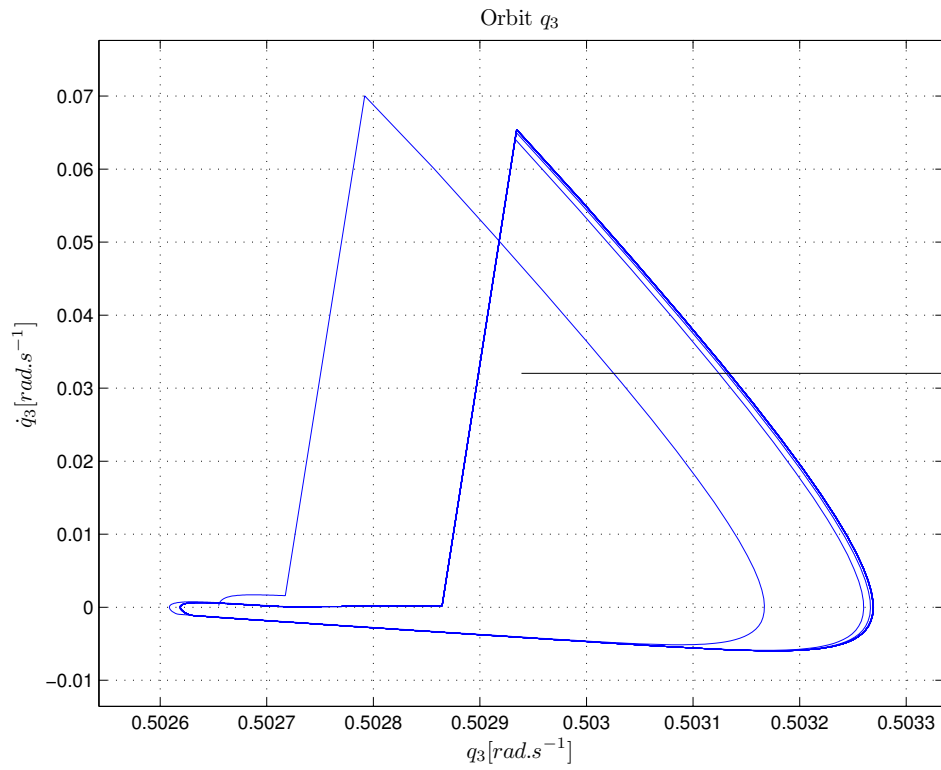
Obrázek 3.10: Řez stavovým prostorem

Obrázek 3.11: Poincarého mapa pro úhel  $q_1$

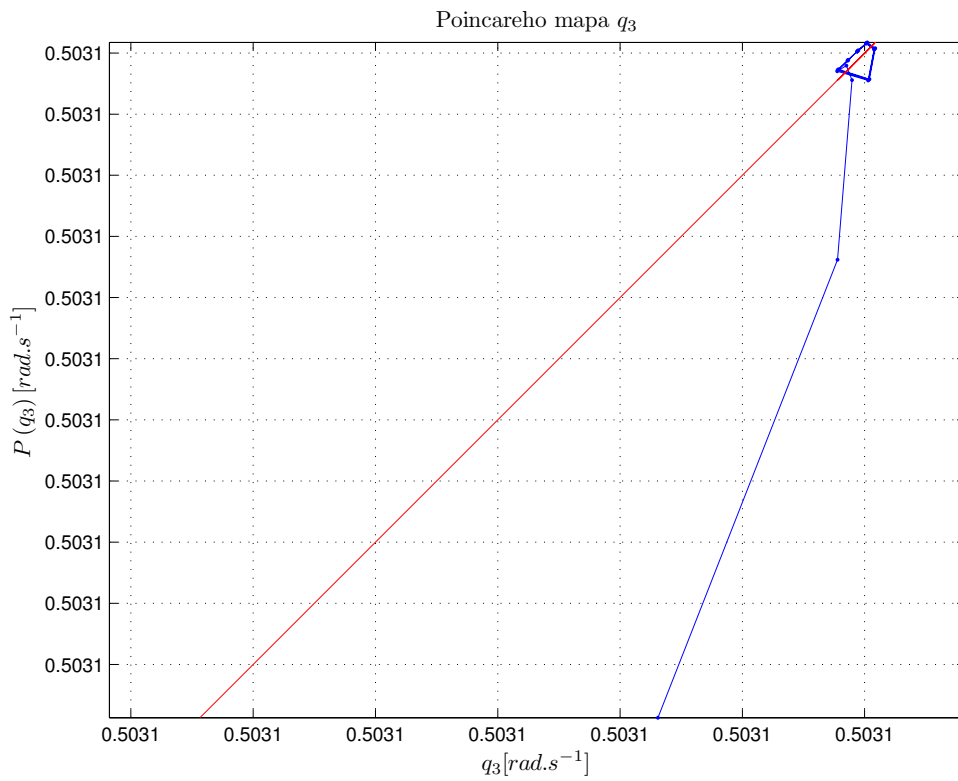


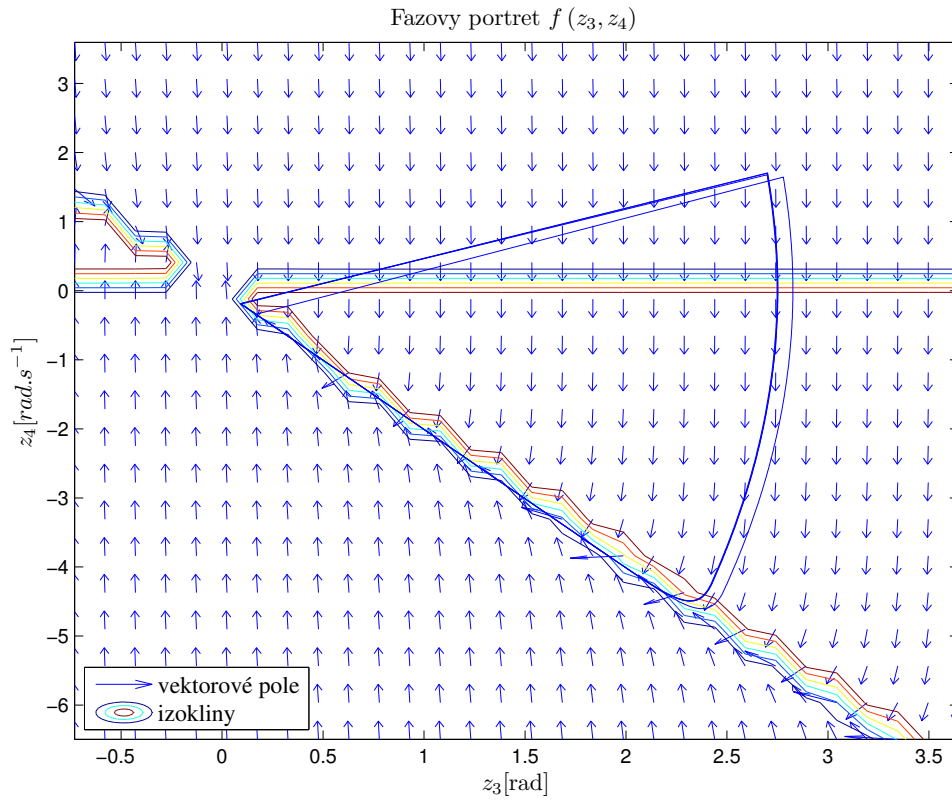
Obrázek 3.12: Řez stavovým prostorem

Obrázek 3.13: Poincarého mapa pro úhel  $q_2$



Obrázek 3.14: Řez stavovým prostorem

Obrázek 3.15: Poincarého mapa pro úhel  $q_3$



Obrázek 3.16: Fázový diagram

## 4. Závěr

Program simulace chůze svounohého robota je složen ze 3 částí, které jsou na sobě závislé a vyvíjí se proto současně.

1. Generační (3.2)
2. Simulační (3.3)
3. GUI (3.4)

Třída *ModelGenerator2* představuje generační část programu, má na starosti symbolické výpočty. Generuje z přímé kinematiky otevřeného kinematického řetězce energie  $E_p$ ,  $E_p$  a  $A$ , ze kterých jsou metodou Lagrangeových rovnic vypočítány pohybové rovnice systému. Výsledný model je převeden do tvaru typického pro mechanické systémy (2.27). Protože dvounohý kráčející robot je ze své povahy hybridním systémem (obr.2.2) skládá se ze spojitého modelu a diskretních událostí. V případě jednoduchého robota bez kolen má pouze 2 střídající se fáze. Zhupová fáze se opakuje při každém zhoupnutí nohy a druhá, nárazová fáze, proběhne v čase limitně jsoucím k nule v okamžiku rázu nohy o zem. Každou z těchto fází popisuje vlastní model. Model zhupové fáze je čistě modelem tvořeným na základě přímé kinematiky otevřeného kinematického řetězce s pevně uchyceným koncem. Model pro nárazovou fázi se liší 2mi stupni volnosti navíc v ose x,y, které jsou zahrnuty do stavového prostoru, tím je robot uvolněn v prostoru pro symbolický výpočet. Při vyhodnocování modelu v simulaci již tento bod umístěný do počátku souřadnic  $p_0$ . Není tak závislý na  $\mathbf{q}$  jako v případě, kdyby byl zvolen v těžišti robota, a proto i rychlost v tomto bodě  $\dot{p}_0$  je nulová a není třeba s ní dále pracovat. Matice obou modelů jsou použity při generování funkcí zápisem do souboru. Generování spustitelných funkcí bylo zvoleno z praktických důvodů.

Model se třemi stupni volnosti má pouze 2 stupně volnosti aktuované. Ve zvolených tvarových souřadnicích (2.4) jsou řízeny úhly torza  $q_3$  a nohy  $q_2$ . Úhel  $q_1$  zůstává bez řízení. Sklon torza je jako jediná souřadnice kloubového prostoru definována absolutně vzhledem k okolnímu prostředí. Ostatní úhly jsou zdefinovány relativně k torzu. Přímá kinematika je vyjádřena vzhledem k počátku (2.20) v závislosti na  $\mathbf{q}$ .

Tento model je linearizován vstupně-výstupní linearizací (2.6.1) pro dosažení nulové dynamiky systému, známé také pod názvem virtuální vazby. Řízení robota pak závisí na tvaru výstupní funkce (2.103). Udržování torza pod konstantním sklonem zajišťuje překonání fáze s nedostatečným řízením s pomocí gravitace se dodává moment hybnosti pro pohyb robota.

S použitím stavového regulátoru (2.123) je dosaženo mezního cyklu v systému. Vzhledem k tomu, že úhel torza  $q_3$  je regulátorem udržován velmi přesně v malém rozsahu hodnot, není jeho fázový portrét názorný tak, jako pro úhel  $q_2$  reprezentovaný výstupem  $y_2$  a tedy souřadnicemi  $z_3, z_4$  (3.16) lineární náhrady (2.115).

Poincarého mapy (3.11), (3.13), (3.15) nejsou mapy s analyticky zjištěnou závislostí  $(q_i)_{k+1} = P((q_i)_k)$ . Představují nalezené průsečíky mezního cyklu s polopřímku, které jsou patrné na příslušných cyklech podle úhlové souřadnice (3.10), (3.12) a (3.14). Poincarého mapa pro lineární systém by měla znázorňovat 2 přímky. Blíží-li se vývoj  $P^k((q_i)_0)$  k průsečíku, existuje mezní cyklus a systém je stabilní.

V tomto případě se přímky přibližují a v oblasti kolem  $P(q_{iref}) = q_{iref}$  se vytvoří oblast, odkud se už trajektorie nedostane. I zde je patrné, že podle Poincaré-Bendixonova teoremu je systém stabilní.

Na průběžích momentů **M** (3.7) je poznat vliv saturace (obr.3.1), která simuluje omezený akční zásah.

V simulaci s  $dof = 5$  modelem se nepodařilo dosáhnout mezního cyklu. Nejvýše několik kroků s následným pádem. Model je linearizován obdobným způsobem, jako systém s  $dof = 3$ . Při ověření linearity vykazuje nestabilitu matice  $\mathcal{L}_g \mathcal{L}_f \mathbf{h}(\xi)$ .

I zjednodušené matice zhupového a impaktního modelu jsou příliš rozsáhlé vypsání. Jsou k nalezení ve vygenerovaných souborech modelu *shapedof5* a v jeho příslušných modulech.

## 5. Výsledky

Matice  $\mathbf{D}(\mathbf{q})$  a  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  jsou symetrické, vypsány jsou jen prvky na diagonále a nad diagonálou, chybějící indexy značí nulové prvky v maticích.

$\mathcal{L}_g \mathcal{L}_f^{(1)} \mathbf{h}^*$  je symetrická matice, prvky na diagonále a pod diagonálou jsou následující:

$$L_g L_f^1 h_{1,1}^* = 4l_1^2 m_1 (4J_1 \cos(q_1 - q_2) - 2l_1^2 m_1 - l_1^2 m_3 - 4J_1 + l_1^2 m_1 \cos(q_1 - q_2))^2 + l_1 l_3 m_3 \cos(q_1 - q_2) \cos(q_1) \quad (5.1)$$

$$L_g L_f^1 h_{1,2}^* = 2 (32J_1^2 + 3l_1^4 m_1^2 + 2l_1^4 m_1 m_3 - l_1^4 m_1^2 \cos(2q_1 - 2q_2) + 24J_1 l_1^2 m_1 + 8J_1 l_1^2 m_3^2) \quad (5.2)$$

$$L_g L_f^1 h_{2,2}^* = 4l_1^2 m_1 (4J_1 \cos(q_1 - q_2) - 2l_1^2 m_1 - l_1^2 m_3 - 4J_1 + l_1^2 m_1 \cos(q_1 - q_2))^2 + l_1 l_3 m_3 \cos(q_1 - q_2) \cos(q_1) \quad (5.3)$$

$$\begin{aligned} den(\mathbf{q}) = & -32J_1^2 l_1^2 m_1 \cos(q_1 - q_2) + 48J_1^2 l_1^2 m_1 + 16J_1^2 l_1^2 m_3 - 32J_1^2 l_1 l_3 m_3 \cos(q_1) \\ & + 16J_1^2 l_3^2 m_3 + 64J_3 J_1^2 - 8J_1 l_1^4 m_1^2 \cos(q_1 - q_2)^2 + 16J_1 l_1^4 m_1^2 \\ & + 8J_1 l_1^4 m_1 m_3 - 8J_1 l_1^3 l_3 m_1 m_3 \cos(q_1 - q_2) \cos(q_1) - 8J_1 l_1^3 l_3 m_1 m_3 \cos(q_1) \\ & + 12J_1 l_1^2 l_3^2 m_1 m_3 - 4J_1 l_1^2 l_3^2 m_3^2 \cos(q_1)^2 + 4J_1 l_1^2 l_3^2 m_3^2 \\ & + 48J_3 J_1 l_1^2 m_1 + 16J_3 J_1 l_1^2 m_3 - l_1^4 l_3^2 m_1^2 m_3 \cos(q_1 - q_2)^2 \\ & + 2l_1^4 l_3^2 m_1^2 m_3 - l_1^4 l_3^2 m_1 m_3^2 \cos(q_1)^2 + l_1^4 l_3^2 m_1 m_3^2 \\ & - 4J_3 l_1^4 m_1^2 \cos(q_1 - q_2)^2 + 8J_3 l_1^4 m_1^2 + 4J_3 l_1^4 m_1 m_3 \end{aligned} \quad (5.4)$$

$$\begin{aligned}
L_f^2 h_{1,1}^* = & 32 J_1^2 \dot{q}_1^2 l_1^2 m_1 \sin(q_1 - q_2) - 32 J_1^2 \dot{q}_2^2 l_1^2 m_1 \sin(q_1 - q_2) \\
& + 16 J_1 \dot{q}_1^2 l_1^4 m_1^2 \sin(q_1 - q_2) - 8 J_1 \dot{q}_2^2 l_1^4 m_1^2 \sin(q_1 - q_2) \\
& + 8 J_1 \dot{q}_3^2 l_1^4 m_1^2 \sin(q_1 - q_2) - 64 J_1^2 g l_3 m_3 \sin(q_3) + 24 J_1 g l_1^3 m_1^2 \sin(q_1 + q_3) \\
& - 16 J_1 g l_1^3 m_1^2 \sin(q_2 + q_3) + 4 J_1 \dot{q}_1^2 l_1^4 m_1^2 \sin(2q_1 - 2q_2) \\
& - 4 J_1 \dot{q}_2^2 l_1^4 m_1^2 \sin(2q_1 - 2q_2) + 8 J_1 g l_1^3 m_1^2 \sin(q_1 - 2q_2 - q_3) \\
& + 16 J_1 g l_1^3 m_1^2 \sin(2q_1 - q_2 + q_3) + 128 J_1^2 g l_1 m_1 \sin(q_1 + q_3) \\
& - 64 J_1^2 g l_1 m_1 \sin(q_2 + q_3) + 64 J_1^2 g l_1 m_3 \sin(q_1 + q_3) \\
& - \dot{q}_2^2 l_1^5 l_3 m_1^2 m_3 \sin(2q_1 - q_2) - \dot{q}_3^2 l_1^5 l_3 m_1^2 m_3 \sin(2q_1 - q_2) \\
& - 4 J_1 \dot{q}_3^2 l_1^2 l_3^2 m_3^2 \sin(2q_1) - 2 g l_1^4 l_3 m_1 m_3^2 \sin(q_3) - 3 g l_1^4 l_3 m_1^2 m_3 \sin(q_3) \\
& + g l_1^4 l_3 m_1^2 m_3 \sin(2q_1 - 2q_2 + q_3) + 64 J_1^2 \dot{q}_1 \dot{q}_3 l_1^2 m_1 \sin(q_1 - q_2) \\
& - 64 J_1^2 \dot{q}_2 \dot{q}_3 l_1^2 m_1 \sin(q_1 - q_2) + 32 J_1 \dot{q}_1 \dot{q}_3 l_1^4 m_1^2 \sin(q_1 - q_2) \\
& - 16 J_1 \dot{q}_2 \dot{q}_3 l_1^4 m_1^2 \sin(q_1 - q_2) + 8 J_1 \dot{q}_1^2 l_1^4 m_1 m_3 \sin(q_1 - q_2) \\
& + 8 J_1 \dot{q}_3^2 l_1^4 m_1 m_3 \sin(q_1 - q_2) + 8 J_1 g l_1^2 l_3 m_3^2 \sin(2q_1 + q_3) \\
& - \dot{q}_3^2 l_1^4 l_3^2 m_1 m_3^2 \sin(2q_1) + 2 g l_1^4 l_3 m_1 m_3^2 \sin(2q_1 + q_3) \\
& + 3 g l_1^4 l_3 m_1^2 m_3 \sin(2q_1 + q_3) - g l_1^4 l_3 m_1^2 m_3 \sin(2q_2 + q_3) \\
& + 8 J_1 \dot{q}_1^2 l_1^3 l_3 m_3^2 \sin(q_1) + 8 J_1 \dot{q}_3^2 l_1^3 l_3 m_3^2 \sin(q_1) \\
& + 16 J_1 g l_1^3 m_1 m_3 \sin(q_1 + q_3) - 8 J_1 g l_1^3 m_1 m_3 \sin(q_2 + q_3) \\
& + 2 \dot{q}_1^2 l_1^5 l_3 m_1 m_3^2 \sin(q_1) + 3 \dot{q}_1^2 l_1^5 l_3 m_1^2 m_3 \sin(q_1) + 2 \dot{q}_3^2 l_1^5 l_3 m_1 m_3^2 \sin(q_1) \\
& + 3 \dot{q}_3^2 l_1^5 l_3 m_1^2 m_3 \sin(q_1) + \dot{q}_2^2 l_1^5 l_3 m_1^2 m_3 \sin(q_2) + \dot{q}_3^2 l_1^5 l_3 m_1^2 m_3 \sin(q_2) \\
& + 8 J_1 \dot{q}_1 \dot{q}_3 l_1^4 m_1^2 \sin(2q_1 - 2q_2) - 8 J_1 \dot{q}_2 \dot{q}_3 l_1^4 m_1^2 \sin(2q_1 - 2q_2) \\
& + 8 J_1 g l_1^3 m_1 m_3 \sin(2q_1 - q_2 + q_3) + \dot{q}_1^2 l_1^5 l_3 m_1^2 m_3 \sin(q_1 - 2q_2) \\
& + \dot{q}_3^2 l_1^5 l_3 m_1^2 m_3 \sin(q_1 - 2q_2) + 32 J_1^2 \dot{q}_1^2 l_1 l_3 m_3 \sin(q_1) - 8 J_1 g l_1^2 l_3 m_3^2 \sin(q_3) \\
& - 32 J_1 g l_1^2 l_3 m_1 m_3 \sin(q_3) - 2 \dot{q}_2 \dot{q}_3 l_1^5 l_3 m_1^2 m_3 \sin(2q_1 - q_2) \\
& + 16 J_1 \dot{q}_1 \dot{q}_3 l_1^4 m_1 m_3 \sin(q_1 - q_2) + 16 J_1 g l_1^2 l_3 m_1 m_3 \sin(2q_1 + q_3) \\
& + 16 J_1 \dot{q}_1 \dot{q}_3 l_1^3 l_3 m_3^2 \sin(q_1) + 24 J_1 \dot{q}_1^2 l_1^3 l_3 m_1 m_3 \sin(q_1) \\
& + 16 J_1 \dot{q}_3^2 l_1^3 l_3 m_1 m_3 \sin(q_1) + 4 J_1 \dot{q}_2^2 l_1^3 l_3 m_1 m_3 \sin(q_2) \\
& + 4 \dot{q}_1 \dot{q}_3 l_1^5 l_3 m_1 m_3^2 \sin(q_1) + 6 \dot{q}_1 \dot{q}_3 l_1^5 l_3 m_1^2 m_3 \sin(q_1) \\
& + 2 \dot{q}_2 \dot{q}_3 l_1^5 l_3 m_1^2 m_3 \sin(q_2) + 2 \dot{q}_1 \dot{q}_3 l_1^5 l_3 m_1^2 m_3 \sin(q_1 - 2q_2) \\
& - 4 J_1 \dot{q}_2^2 l_1^3 l_3 m_1 m_3 \sin(2q_1 - q_2) - 8 J_1 \dot{q}_3^2 l_1^3 l_3 m_1 m_3 \sin(2q_1 - q_2) \\
& + 64 J_1^2 \dot{q}_1 \dot{q}_3 l_1 l_3 m_3 \sin(q_1) + 48 J_1 \dot{q}_1 \dot{q}_3 l_1^3 l_3 m_1 m_3 \sin(q_1) \\
& + 8 J_1 \dot{q}_2 \dot{q}_3 l_1^3 l_3 m_1 m_3 \sin(q_2) - 8 J_1 \dot{q}_2 \dot{q}_3 l_1^3 l_3 m_1 m_3 \sin(2q_1 - q_2)
\end{aligned} \tag{5.5}$$



$$\begin{aligned}
L_f^2 h_{2,1}^* = & l_1 m_1 (g l_1^2 l_3^2 m_3^2 \sin(2q_1 - q_2 + q_3) - 2 \dot{q}_1^2 l_1^4 l_3 m_3^2 \sin(q_1) - 2 \dot{q}_3^2 l_1^4 l_3 m_3^2 \sin(q_1)) \\
& - 64 J_1 J_3 g \sin(q_2 + q_3) - 4 J_1 \dot{q}_1^2 l_1^3 m_1 \sin(2q_1 - 2q_2) - 4 J_1 \dot{q}_2^2 l_1^3 m_1 \sin(2q_1 - 2q_2) \\
& \quad - 8 J_1 \dot{q}_3^2 l_1^3 m_1 \sin(2q_1 - 2q_2) - 4 J_3 \dot{q}_2^2 l_1^3 m_1 \sin(2q_1 - 2q_2) \\
& \quad - 4 J_3 \dot{q}_3^2 l_1^3 m_1 \sin(2q_1 - 2q_2) - 8 J_1 g l_1^2 m_1 \sin(q_1 - 2q_2 - q_3) \\
& \quad + 16 J_1 g l_1^2 m_1 \sin(2q_1 - q_2 + q_3) + 8 J_1 g l_1^2 m_3 \sin(2q_1 - q_2 + q_3) \\
& \quad + 16 J_3 g l_1^2 m_1 \sin(2q_1 - q_2 + q_3) + 8 J_3 g l_1^2 m_3 \sin(2q_1 - q_2 + q_3) \\
& \quad + 32 J_1 J_3 \dot{q}_1^2 l_1 \sin(q_1 - q_2) + 32 J_1 J_3 \dot{q}_3^2 l_1 \sin(q_1 - q_2) + \dot{q}_1^2 l_1^3 l_3^2 m_3^2 \sin(q_1 + q_2) \\
& \quad + \dot{q}_3^2 l_1^3 l_3^2 m_3^2 \sin(q_1 + q_2) + g l_1^2 l_3^2 m_3^2 \sin(q_2 - q_3) - \dot{q}_3^2 l_1^2 l_3^3 m_3^2 \sin(q_2) \\
& \quad + \dot{q}_1^2 l_1^3 l_3^2 m_3^2 \sin(q_1 - q_2) + \dot{q}_3^2 l_1^3 l_3^2 m_3^2 \sin(q_1 - q_2) + 2 g l_1^3 l_3 m_3^2 \sin(q_3) \\
& + g l_1^2 l_3^2 m_3^2 \sin(2q_1 + q_2 + q_3) + 16 J_1 \dot{q}_1^2 l_1^3 m_1 \sin(q_1 - q_2) + 8 J_1 \dot{q}_2^2 l_1^3 m_1 \sin(q_1 - q_2) \\
& + 24 J_1 \dot{q}_3^2 l_1^3 m_1 \sin(q_1 - q_2) + 8 J_1 \dot{q}_1^2 l_1^3 m_3 \sin(q_1 - q_2) + 16 J_3 \dot{q}_1^2 l_1^3 m_1 \sin(q_1 - q_2) \\
& + 8 J_1 \dot{q}_2^2 l_1^3 m_3 \sin(q_1 - q_2) + 16 J_3 \dot{q}_3^2 l_1^3 m_1 \sin(q_1 - q_2) + 8 J_3 \dot{q}_1^2 l_1^3 m_3 \sin(q_1 - q_2) \\
& \quad + 8 J_3 \dot{q}_2^2 l_1^3 m_3 \sin(q_1 - q_2) + \dot{q}_3^2 l_1^3 l_3^2 m_3^2 \sin(2q_1) - 2 g l_1^3 l_3 m_3^2 \sin(2q_1 + q_3) \\
& \quad - g l_1^2 l_3^2 m_3^2 \sin(q_2 + q_3) - \dot{q}_3^2 l_1^2 l_3^3 m_3^2 \sin(2q_1 - q_2) - 24 J_1 g l_1^2 m_1 \sin(q_1 + q_3) \\
& \quad - 16 J_1 g l_1^2 m_1 \sin(q_2 + q_3) - 16 J_1 g l_1^2 m_3 \sin(q_1 + q_3) - 8 J_1 g l_1^2 m_3 \sin(q_2 + q_3) \\
& \quad - 16 J_3 g l_1^2 m_1 \sin(q_2 + q_3) - 16 J_1 g l_3^2 m_3 \sin(q_2 + q_3) - 8 J_3 g l_1^2 m_3 \sin(q_2 + q_3) \\
& \quad - \dot{q}_2^2 l_1^3 l_3^2 m_1 m_3 \sin(2q_1 - 2q_2) - \dot{q}_3^2 l_1^3 l_3^2 m_1 m_3 \sin(2q_1 - 2q_2) \\
& \quad - 4 \dot{q}_1 \dot{q}_3 l_1^4 l_3 m_3^2 \sin(q_1) - 3 \dot{q}_1^2 l_1^4 l_3 m_1 m_3 \sin(q_1) - 3 \dot{q}_3^2 l_1^4 l_3 m_1 m_3 \sin(q_1) \\
& \quad - \dot{q}_2^2 l_1^4 l_3 m_1 m_3 \sin(q_2) - \dot{q}_3^2 l_1^4 l_3 m_1 m_3 \sin(q_2) + 4 g l_1^2 l_3^2 m_1 m_3 \sin(2q_1 - q_2 + q_3) \\
& \quad - 8 J_1 \dot{q}_1 \dot{q}_3 l_1^3 m_1 \sin(2q_1 - 2q_2) - 8 J_1 \dot{q}_2 \dot{q}_3 l_1^3 m_1 \sin(2q_1 - 2q_2) \\
& \quad - 8 J_3 \dot{q}_2 \dot{q}_3 l_1^3 m_1 \sin(2q_1 - 2q_2) + 8 J_1 \dot{q}_1^2 l_1 l_3^2 m_3 \sin(q_1 - q_2) \\
& + 8 J_1 \dot{q}_3^2 l_1 l_3^2 m_3 \sin(q_1 - q_2) + 16 J_1 g l_1 l_3 m_3 \sin(q_3) - 8 J_1 g l_1 l_3 m_3 \sin(q_1 - q_2 - q_3) \\
& + 64 J_1 J_3 \dot{q}_1 \dot{q}_3 l_1 \sin(q_1 - q_2) + 2 \dot{q}_1 \dot{q}_3 l_1^3 l_3^2 m_3^2 \sin(q_1 + q_2) - \dot{q}_1^2 l_1^4 l_3 m_1 m_3 \sin(q_1 - 2q_2) \\
& \quad - \dot{q}_3^2 l_1^4 l_3 m_1 m_3 \sin(q_1 - 2q_2) + 16 J_1 g l_1 l_3 m_3 \sin(q_1 + q_2 + q_3) \\
& \quad - 4 J_1 \dot{q}_1^2 l_1^2 l_3 m_3 \sin(2q_1 - q_2) - 8 J_1 \dot{q}_3^2 l_1^2 l_3 m_3 \sin(2q_1 - q_2) \\
& \quad - 4 J_3 \dot{q}_3^2 l_1^2 l_3 m_3 \sin(2q_1 - q_2) + 2 \dot{q}_1 \dot{q}_3 l_1^3 l_3^2 m_3^2 \sin(q_1 - q_2) \\
& \quad + \dot{q}_2^2 l_1^4 l_3 m_1 m_3 \sin(2q_1 - q_2) + \dot{q}_3^2 l_1^4 l_3 m_1 m_3 \sin(2q_1 - q_2) \\
& + 4 \dot{q}_1^2 l_1^3 l_3^2 m_1 m_3 \sin(q_1 - q_2) + 4 \dot{q}_3^2 l_1^3 l_3^2 m_1 m_3 \sin(q_1 - q_2) + 3 g l_1^3 l_3 m_1 m_3 \sin(q_3) \\
& \quad - g l_1^3 l_3 m_1 m_3 \sin(2q_1 - 2q_2 + q_3) + 32 J_1 \dot{q}_1 \dot{q}_3 l_1^3 m_1 \sin(q_1 - q_2) \\
& \quad + 16 J_1 \dot{q}_2 \dot{q}_3 l_1^3 m_1 \sin(q_1 - q_2) + 16 J_1 \dot{q}_1 \dot{q}_3 l_1^3 m_3 \sin(q_1 - q_2) \\
& \quad + 32 J_3 \dot{q}_1 \dot{q}_3 l_1^3 m_1 \sin(q_1 - q_2) + 16 J_3 \dot{q}_1 \dot{q}_3 l_1^3 m_3 \sin(q_1 - q_2) \\
& \quad - 8 J_1 g l_1 l_3 m_3 \sin(q_1 - q_2 + q_3) - 3 g l_1^3 l_3 m_1 m_3 \sin(2q_1 + q_3) \\
& + g l_1^3 l_3 m_1 m_3 \sin(2q_2 + q_3) - 4 g l_1^2 l_3^2 m_1 m_3 \sin(q_2 + q_3) - 8 J_1 \dot{q}_1^2 l_1^2 l_3 m_3 \sin(q_1) \\
& \quad + 12 J_1 \dot{q}_1^2 l_1^2 l_3 m_3 \sin(q_2) + 8 J_1 \dot{q}_3^2 l_1^2 l_3 m_3 \sin(q_2) - 4 J_3 \dot{q}_3^2 l_1^2 l_3 m_3 \sin(q_2) \\
& \quad - 2 \dot{q}_2 \dot{q}_3 l_1^3 l_3^2 m_1 m_3 \sin(2q_1 - 2q_2) - 6 \dot{q}_1 \dot{q}_3 l_1^4 l_3 m_1 m_3 \sin(q_1) \\
& \quad - 2 \dot{q}_2 \dot{q}_3 l_1^4 l_3 m_1 m_3 \sin(q_2) + 16 J_1 \dot{q}_1 \dot{q}_3 l_1 l_3^2 m_3 \sin(q_1 - q_2) \\
& \quad - 2 \dot{q}_1 \dot{q}_3 l_1^4 l_3 m_1 m_3 \sin(q_1 - 2q_2) - 8 J_1 \dot{q}_1 \dot{q}_3 l_1^2 l_3 m_3 \sin(2q_1 - q_2) \\
& + 2 \dot{q}_2 \dot{q}_3 l_1^4 l_3 m_1 m_3 \sin(2q_1 - q_2) + 8 \dot{q}_1 \dot{q}_3 l_1^3 l_3^2 m_1 m_3 \sin(q_1 - q_2) - 16 J_1 \dot{q}_1 \dot{q}_3 l_1^2 l_3 m_3 \sin(q_1) \\
& \quad + 24 J_1 \dot{q}_1 \dot{q}_3 l_1^2 l_3 m_3 \sin(q_2))
\end{aligned}$$

$$\begin{aligned}
den(\mathbf{q}) = & (-2) \left( -32 J_1^2 l_1^2 m_1 \cos(q_1 - q_2) + 48 J_1^2 l_1^2 m_1 + 16 J_1^2 l_1^2 m_3 \right. \\
& - 32 J_1^2 l_1 l_3 m_3 \cos(q_1) + 16 J_1^2 l_3^2 m_3 + 64 J_3 J_1^2 - 8 J_1 l_1^4 m_1^2 \cos(q_1 - q_2)^2 \\
& + 16 J_1 l_1^4 m_1^2 + 8 J_1 l_1^4 m_1 m_3 - 8 J_1 l_1^3 l_3 m_1 m_3 \cos(q_1 - q_2) \cos(q_1) \\
& - 8 J_1 l_1^3 l_3 m_1 m_3 \cos(q_1) + 12 J_1 l_1^2 l_3^2 m_1 m_3 - 4 J_1 l_1^2 l_3^2 m_3^2 \cos(q_1)^2 \\
& + 4 J_1 l_1^2 l_3^2 m_3^2 + 48 J_3 J_1 l_1^2 m_1 + 16 J_3 J_1 l_1^2 m_3 - l_1^4 l_3^2 m_1^2 m_3 \cos(q_1 - q_2)^2 \\
& + 2 l_1^4 l_3^2 m_1^2 m_3 - l_1^4 l_3^2 m_1 m_3^2 \cos(q_1)^2 + l_1^4 l_3^2 m_1 m_3^2 \\
& \left. - 4 J_3 l_1^4 m_1^2 \cos(q_1 - q_2)^2 + 8 J_3 l_1^4 m_1^2 + 4 J_3 l_1^4 m_1 m_3 \right) \\
& (5.7)
\end{aligned}$$

# Literatura

- [1] ELEONOR D. Stoenescu *Dynamics and Synthesis of Kinematic Chains with Impact and Clearance*
- [2] IBORT A., LÉN M., LACOMBA E., MARTÍN D., PINTAGA P. *Mechanical systems subjected to impulsive constraints*
- [3] HRNČÁREK M. *Moderní struktury řízení servosystémů se střídavými pohony*
- [4] GRIZZLE W., a kol. *Feedback Control of Dynamic Bipedal Robot Locomotion*
- [5] ZHAO M., ZHANG J., DONG H., LIU Y., LI L., SU X. *Humanoid Robot Gait Generation Based on Limit Cycle Stability*
- [6] MATHAI A. M. *Jacobians of Matrix Transformations and Functions of Matrix Argument*
- [7] HAND L., FINCH J. *Analytical Mechanics*
- [8] Prof. Ing. ŠOLC F.; Ing. ŽALUD L., Ph.D. *Robotika, Skriptum VUT Brno, 2006*
- [9] LAM P. *Walking Algorithm for Small Humanoid*