



## Diplomová práce

# Identifikace setrvačných parametrů tiskového robotu

*Studijní program:*

N0714A270010 Mechatronika

*Autor práce:*

**Bc. Vojtěch Marinič**

*Vedoucí práce:*

Ing. Daniel Kajzr

Ústav mechatroniky a technické informatiky

Liberec 2024



## Zadání diplomové práce

# Identifikace setrvačných parametrů tiskového robotu

<i>Jméno a příjmení:</i>	<b>Bc. Vojtěch Marinič</b>
<i>Osobní číslo:</i>	M22000037
<i>Studijní program:</i>	N0714A270010 Mechatronika
<i>Zadávací katedra:</i>	Ústav mechatroniky a technické informatiky
<i>Akademický rok:</i>	2023/2024

### Zásady pro vypracování:

1. Proveďte rešerši stávajících metod identifikace setrvačných parametrů u robotických manipulátorů.
2. Sestavte diferenciální rovnice popisující dynamiku planárního robotu a proveďte jejich přepis do lineární podoby vhodné pro identifikaci setrvačných parametrů.
3. Navrhněte identifikační experiment pro možnost identifikace setrvačných parametrů na experimentálním modelu tiskového robotu.
4. Zvolte několik metod identifikace setrvačných parametrů a proveďte identifikaci na experimentálním modelu tiskového robotu.
5. Proveďte zhodnocení dosažených výsledků a porovnání jednotlivých metod.
6. Zvolte některou z metod řízení robotických manipulátorů založených na modelu a proveďte její implementaci na experimentální model tiskového robotu.

*Rozsah grafických prací:* dle potřeby dokumentace  
*Rozsah pracovní zprávy:* 40 až 50 stran  
*Forma zpracování práce:* tištěná/elektronická  
*Jazyk práce:* čeština

### **Seznam odborné literatury:**

- [1] KHALIL, W. a DOMBRE, E. Modeling identification & control of robots. New York, NY: Taylor & Francis, 2002. ISBN 1560329831.
- [2] DE WIT, Carlos Canudas; SICILIANO, Bruno a BASTIN, Georges (ed.). Theory of Robot Control. Online. Communications and Control Engineering. London: Springer London, 1996. ISBN 978-1-4471-1503-8.
- [3] ZÁDA, Václav. Robotika: matematické aspekty analýzy a řízení. Liberec: Technická univerzita v Liberci, 2012. ISBN 978-80-7372-882-3.
- [4] GREPL, Robert. Kinematika a dynamika mechatronických systémů. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3530-8.

*Vedoucí práce:* Ing. Daniel Kajzr  
Ústav mechatroniky a technické informatiky

*Datum zadání práce:* 12. října 2023  
*Předpokládaný termín odevzdání:* 14. května 2024

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

doc. Dr. Ing. Jaroslav Hlava  
garant studijního programu

V Liberci dne 12. října 2023

## Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.



# Identifikace setrvačných parametrů tiskového robotu

## Abstrakt

Automatizace a zvyšování efektivity prostřednictvím robotických systémů se stává klíčovým prvkem napříč různými průmyslovými odvětvími, lékařstvím a stavebnictvím.

První část této práce se zaměřuje na tvorbu komplexního modelu planárního robotu, který je využitelný ve 3D tisku. Cílem je představit metodologii tvorby přesných virtuálních modelů robotů, zahrnující analýzu kinematiky a dynamiky prostřednictvím analytických výpočtů. Následně je tento model ověřen pomocí Robotic System Toolboxu implementovaného v prostředí MATLAB Simulink.

Prostřední část práce se zabývá identifikací dynamických parametrů robotických systémů. Jsou poskytnuty základní informace pro uskutečnění identifikačního experimentu a jsou představeny tři identifikační metody, které jsou podrobně popsány. Identifikace je provedena s využitím benchmarku BIRDy [24], který poskytuje kvalitní nástroje pro zjednodušení identifikační procedury.

Výsledkem práce jsou dva navržené řídicí algoritmy založené na estimovaných koeficientech z identifikační procedury. Tyto metody jsou porovnány a aplikovány na reálný systém, přičemž jsou podpořeny experimenty.

**Klíčová slova:** Robotický manipulátor, Planární robot, Kinematika, Dynamika, Identifikace, Řízení, BIRDy, Momentové řízení, Dopředné řízení

# Identification of inertial parameters of the printing robot

## Abstract

Automation and enhancing efficiency through robotic systems is becoming a pivotal element across various industrial sectors, health-care, and construction.

The first part of this thesis focuses on creating a comprehensive model of a planar robot applicable in 3D printing. The objective is to introduce a methodology for generating precise virtual robot models, involving kinematic and dynamic analysis through analytical calculations. Subsequently, this model is validated using the Robotic System Toolbox implemented in MATLAB Simulink environment.

The middle section of the thesis deals with the identification of dynamic parameters of robotic systems. Fundamental information for conducting the identification experiment is provided, and four identification methods are introduced in detail. Identification is performed using the BIRDy benchmark [24], which provides quality tools to simplify the identification procedure.

The result of the work is two proposed control algorithms based on estimated coefficients from the identification procedure. These methods are compared and applied to a real system, supported by experiments.

**Keywords:** Robotic manipulator, Planar robot, Kinematics, Dynamics, Identification, Control, BIRDy, Computed torque control, Feedforward control

## Poděkování

Tímto bych chtěl poděkovat Ing. Danielu Kajzrovi za odborné vedení, spolupráci a rady při zhotovování této práce. Dále bych rád poděkovat své snoubence a rodině, kteří mě při psaní této práce podporovali.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>12</b>
<b>2</b>	<b>3D tisk s využitím robotických manipulátorů ve stavebnictví</b>	<b>14</b>
2.1	Historie a momentální vývoj . . . . .	14
2.1.1	Budoucnost perspektivy . . . . .	14
2.2	Výhody a výzvy . . . . .	15
2.3	Tiskový materiál . . . . .	15
<b>3</b>	<b>Kinematika a dynamika v robotických systémech</b>	<b>16</b>
3.1	Kinematika . . . . .	16
3.1.1	Denavit-Hartenbergova konvence . . . . .	16
3.2	Dynamika . . . . .	18
3.2.1	Matice hmotností . . . . .	19
3.2.2	Matice Coriolisových a odstředivých sil . . . . .	20
3.2.3	Vektor gravitačních sil . . . . .	21
<b>4</b>	<b>Matematický model planárního robotu</b>	<b>23</b>
4.1	Kinematika planárního robotu . . . . .	24
4.2	Dynamika planárního robotu . . . . .	24
4.3	Robotics system toolbox . . . . .	28
4.4	Simulační porovnání . . . . .	30
<b>5</b>	<b>Identifikace</b>	<b>33</b>
5.1	Tvorba dynamického modelu . . . . .	33
5.2	Vektor koeficientů . . . . .	33
5.3	Identifikační experiment a trajektorie . . . . .	34
5.3.1	Návrh identifikační trajektorie . . . . .	34
5.4	Identifikační metody . . . . .	35
5.4.1	Metoda nejmenších čtverců . . . . .	35
5.4.2	Metoda vážených nejmenších čtverců . . . . .	36
5.4.3	ML . . . . .	36
5.5	Vyhodnocení identifikace . . . . .	37
5.6	Benchmark for Identification of Robot Dynamics . . . . .	38
5.6.1	Robot description file . . . . .	39
5.6.2	Struktura reálného robotu . . . . .	39

5.6.3	Data trajektorie . . . . .	39
5.6.4	Hlavní program . . . . .	39
5.7	Porovnání a vyhodnocení výsledků identifikace . . . . .	40
<b>6</b>	<b>Řízení</b>	<b>43</b>
6.1	Momentové řízení . . . . .	43
6.2	Dopředné řízení . . . . .	44
6.3	Experimentální porovnání metod řízení . . . . .	45
6.3.1	Simulační porovnání bez regulátorů . . . . .	46
6.3.2	Polohové řízení na reálném systému . . . . .	48
6.3.3	Dopředné řízení na reálném systému . . . . .	49
6.3.4	Momentové řízení na reálném systému . . . . .	50
6.3.5	Zhodnocení . . . . .	52
<b>7</b>	<b>Závěr</b>	<b>53</b>

## Seznam obrázků

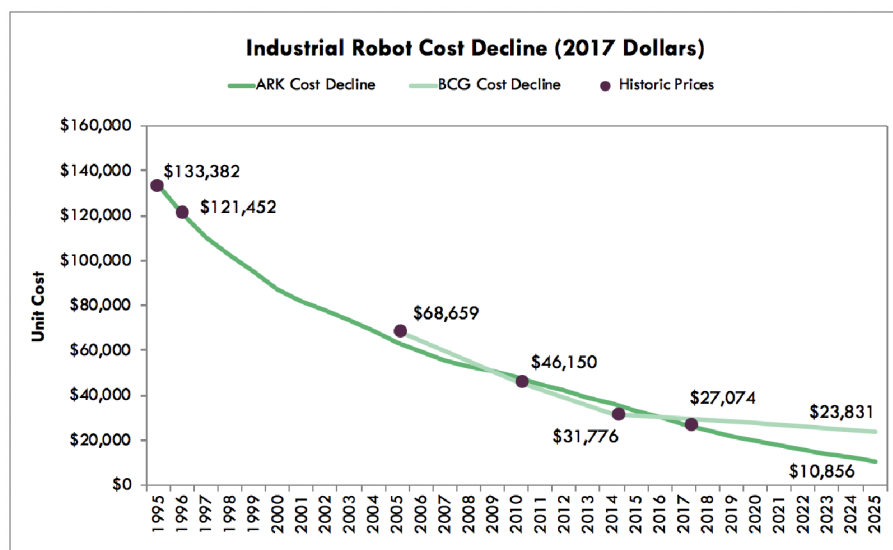
1.1	Cena za robota v průběhu času [3]	12
1.2	Každoroční počet nových instalací robotických systémů [6]	13
3.1	Parametry DH konvence [12]	17
4.1	Planární laboratorní robot	23
4.2	Schéma robotického systému [14]	23
4.3	Struktura robotického systému v toolboxu	29
4.4	Nastavení hodnot pro jednotlivá ramena	29
4.5	Home pozice planárního robotu	30
4.6	Vstupní trajektorie pro ověření výsledků	31
4.7	Vypočtené momenty z analytického výpočtu a toolbox modelu	32
4.8	Rozdíl výsledných momentů	32
5.1	Rozdíl momentů spočítaných pomocí BIRDy a analytického řešení	40
5.2	Trajektorie vložená do BIRDy	41
5.3	Výsledky momenty spočítány s výsledků identifikace v porovnání s naměřeným momentem (identifikační trajektorie)	42
5.4	Výsledky momenty spočítány s výsledků identifikace v porovnání s naměřeným momentem (validační trajektorie)	42
6.1	Schéma momentového řízení	44
6.2	Schéma dopředného řízení	44
6.3	Trajektorie vygenerovaná blokem Polynomial trajectory	46
6.4	Maximální rychlosti u trajektorií	46
6.5	Odchytky od požadované trajektorie při pomalém pohybu	47
6.6	Odchytky od požadované trajektorie při rychlém pohybu	47
6.7	Chyby na jednotlivých osách (bez regulátorů)	47
6.8	Průběh regulačních odchylek z polohového řízení	48
6.9	Porovnání naměřeného momentu a vypočítaného momentu po lineárním přepisu	49
6.10	Průběh regulačních odchylek	50
6.11	Průběhy regulačních odchylek s chybou na koeficientech, získány pomocí simulace	50
6.12	Průběh regulačních odchylek	51

6.13 Průběh regulačních odchylek s chybou na koeficientech, získány pomocí simulace . . . . .	52
---	----

# 1 Úvod

Od svých počátků v polovině 20. století prošla robotika významným vývojem. První robotický manipulátor Unimate (představen Georgem Devolem v roce 1961), byl používán pro manipulaci s odlitky a svařování těchto dílů na karoserii automobilů. Tento typ robotů představoval první krok k automatizaci výrobních procesů a byl schopný vykonávat jednoduché úkoly s opakujícími se pohyby. Ačkoliv se jednalo o revoluční vynález, v té době většina společností neviděla jejich potenciál, a proto se vyskytovaly jen zřídka.

Postupem času společně s modernizací přišlo i snižování výrobních cen za robotické systémy. To co na začátku stálo přes sto tisíc amerických dolarů, je dnes za několikanásobně nižší cenu. Ostatně to lze hezky vidět na obrázku 1.1.



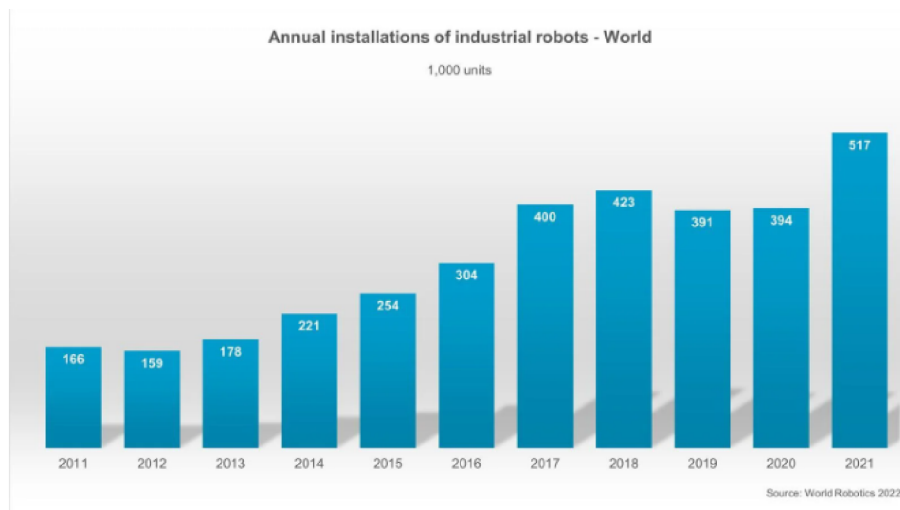
Source: ARK Investment Management LLC, 2019 | [ark-invest.com](https://ark-invest.com); Data from: United Nations Economic Commission for Europe; International Federation of Robotics; Sirkin, Hal, et al. "How Robots Will Redefine Competitiveness." BCG, 23 Sept. 2015, <https://arkinv.st/2VLo0Jt>.

Obr. 1.1: Cena za robota v průběhu času [3]

S tímto trendem se začaly manipulátory objevovat daleko častěji. Usnadňovaly každodenní repetitivní lidskou práci a umožňovaly postupně snižovat náklady na výrobu a tudíž generovat mnohem vyšší zisky. Toho se chytla většina společností a v dnešní době je používání robotických systémů neodmyslitelnou součástí průmyslových odvětví po celém světě. Tento trend není omezen pouze na průmyslové sektory, ale nachází uplatnění i v oblastech jako lékařství, stavebnictví a služby.



Toto potvrzuje i následující graf, kde je zobrazen počet instalací nových robotů v jednotlivých uplynulých letech.



Obr. 1.2: Každoroční počet nových instalací robotických systémů [6]

Tento průzkum je sice už z roku 2022, ale krásně demonstruje, jak rychle se tato oblast techniky vyvíjí. Ve článku z roku 2023 [5] je na základě výzkumu udělána předpověď, že do roku 2026 bude toto číslo nadále stoupat, a to každoročně o 7%.

## Motivace

Vzhledem k rostoucí poptávce po robotických systémech se často setkáváme s časovým zpožděním při dodání objednaných systémů, které může dosahovat řádů měsíců. Nicméně, specifikace a využití robotů jsou stanoveny ještě před jejich objednáním. Je tedy čím dál tím větší zájem o to, aby byl daný systém zmapován do nějakého simulačního prostředí a mohli se na něm dělat experimenty. V podstatě se jedná o velmi přesný model systému. Ten nám nabízí nespočet možností pro navržení konkrétního ovládacího nebo řídicího algoritmu bez toho, aniž bychom museli systém fyzicky mít před sebou. Následně je po dodání během chvíle nahrán algoritmus do fyzického zařízení. Všechny problémy, které by se mohli vyskytnout a zbytečně by pak prodlužovali nasazení do výroby, jsou eliminovány předem.

S tímto inovativním přístupem však může být spojena řada technických výzev. Některé systémy se obtížně přenášejí do digitálního prostředí a vyžadují specifický výzkum a experimenty. V rámci této práce se zaměřujeme na získání komplexního porozumění procesu tvorby přesných digitálních modelů pro robotické manipulátory. Jakmile je model vytvořen, pozornost je věnována identifikaci dynamických parametrů. V tomto procesu přecházíme z nelineární dynamické rovnice na rovnici lineární vzhledem k určitým parametrům. Tento krok je zásadní, neboť nám poskytne spolehlivé výsledky, které následně aplikujeme v metodách řízení.

Tímto čtenář získá podrobný přehled o tvorbě modelů a následném návrhu řízení založeném na těchto modelech a výsledcích identifikace.

## 2 3D tisk s využitím robotických manipulátorů ve stavebnictví

Využívání robotických manipulátorů ve stavebnictví nabízí inovativní řešení a efektivní přístup k rychlé a cenově dostupné výstavbě budov, domů a dalších staveb. Jako v každém jiném odvětví nabízí toto řešení mnoho výzev, ale také výhod, které si v této kapitole podrobněji rozebereme.

### 2.1 Historie a momentální vývoj

První experimenty s použitím robotických manipulátorů pro 3D tisk ve stavebnictví se datují do počátku 21. století. V této fázi byl ale výzkum zaměřen spíše na nalezení vhodného materiálu a na procesy potřebné k samotnému tisku.

Větší pokrok přišel až mezi lety 2010 a 2020, kdy se na trhu začali objevovat první společnosti, zaměřené na vývoj a výrobu robotických tiskáren pro stavebnictví. To vedlo k průlomu v komercializaci technologie 3D tisku domů.

V dnešní době došlo k rapidnímu rozšíření používání robotických manipulátorů pro 3D tisk. Technologie se stává více dostupnou a mnohem více společností začíná investovat do vývoje a výroby takovýchto systémů. Díky novým možnostem experimentů a pokroku v materiálovém výzkumu bylo dosaženo značného zlepšení v rychlosti a přesnosti tisku, ale také v rozmanitosti tisknutých struktur. Stále se ale objevují nové inovace, které potenciál těchto robotů stále zvyšují. Jedním z nejnovějších přínosů je využívání tiskových robotů ve výškách nebo nehostinných prostředích. [1]

#### 2.1.1 Budoucnost perspektivy

Budoucnost používání robotických manipulátorů pro 3D tisk je plná nadějí a možností. Očekává se, že pokračující inovace a vývoj technologie povede k ještě efektivnějším a více škálovatelným procesům tisku. S tím, jak se technologie stává více běžnou a dostupnou, může se 3D tisk stát standardní metodou výstavby budov, což má potenciál změnit způsob, jakým budou budovy navrhovány, stavěny a udržovány v budoucnosti.

## 2.2 Výhody a výzvy

Oproti klasickým metodám umožňují robotické manipulátory automatizovaný proces, což vede ke zvýšení efektivity a produktivity výstavby. Díky přesným řídicím systémům jsou schopny aplikovat stavební materiál s vysokou přesností a konzistencí, což minimalizuje chyby a zajišťuje kvalitní výsledek práce. Jelikož se zde nepoužívají klasické cihly, mají možnost tisknout mnohem více tvarů a velikostí budov, což vede k novým možnostem pro design a návrh nových budov. Tato flexibilita umožňuje tvorbu unikátních staveb. Není snad ani potřeba zmiňovat, že využívání robotických manipulátorů ve velkém snižuje náklady spojené s pracovní silou a materiálem.

Samozřejmě s tímto vším přichází i spousta nových výzev, které je potřeba brát v úvahu. Jedny z největších jsou technická omezení samotného robotického systému a náročná údržba a servis. Je zapotřebí brát v potaz maximální dosahy ramen, nosnost, ale také rychlost pohybu. To nás přivádí na další z omezení, a to potřebu kvalitně vyškolenou obsluhu. Ta musí disponovat technickými znalostmi a dovednostmi v oblasti robotiky, aby byla schopna nejen programovat požadované trajektorii pro tisk, ale také řešila problémy spojené s celým procesem. Zároveň musí znát všechny bezpečnostní nástrahy, aby se nikomu nic nestalo. Protože bezpečnost lidí je vždy na prvním místě. [2]

## 2.3 Tiskový materiál

Nejčastěji používanými materiály pro tisk ve stavebnictví je beton a cementové směsi. Tyto materiály poskytují vynikající pevnost, trvanlivost a schopnost přizpůsobit se různým tvarům a velikostem. Zároveň poskytují uspokojivý čas tuhnutí, který je u 3D tisku velmi důležitým faktorem.

Při tvorbě těchto tiskových směsí je možnost použití i ostatních betonových produktů a tedy velmi snížit množství odpadu. Do směsi lze použít stavební odpad, zbytky cihel, malty, omítek a dalších podobných produktů. Nevyužitý materiál lze velmi snadno recyklovat, což vede ke snížení dopadů na životní prostředí. Je zde i velký potenciál v používání ekologických materiálů, jako například recyklovaný beton.

Alternativou pro betonové směsi může být polymerní beton. Ten vzniká na bázi odpadních recyklovaných plastů. Ty doplňuje nejčastěji písek nebo drcené odpadní sklo. Výsledný produkt je voděodolný. [8] [2]

## 3 Kinematika a dynamika v robotických systémech

Kinematika a dynamika jsou v robotice zásadními prvky, které nám pomohou porozumět vlastnostem robota a jeho chování, ale jsou zároveň klíčovými prvky u sestavování modelu robotického systému, který můžeme dále využívat pro potřeby identifikace, testování a řízení. Tato kapitola nabízí přehled v těchto dvou aspektech a poskytuje teoretický základ pro vytvoření matematického modelu.

### 3.1 Kinematika

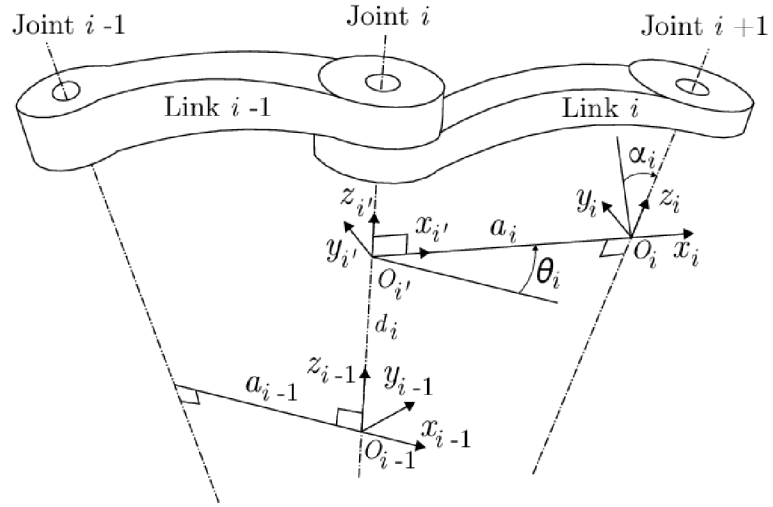
Kinematika robotů je disciplínou, která se zabývá studiem pohybu a geometrií robotických systémů. Hlavním úkolem kinematiky je analýza pohybu robota bez ohledu na síly a momenty, které tento pohyb způsobují. Zahrnuje studium polohy, rychlosti a zrychlení jednotlivých částí robotického systému v prostoru. V kontextu kinematiky robotických systémů jsou důležité pojmy přímá a inverzní kinematika. My si pro účely této práce představíme DH konvenci, protože se využívá pro tvorbu všech modelů v této práci.

#### 3.1.1 Denavit-Hartenbergova konvence

Denavit-Hartenbergova (dále jen DH) konvence je speciálním případem přímé kinematiky, která slouží pro popis vzájemného uspořádání všech článků robota. DH konvence nám poskytuje způsob popisu transformačních vztahů sousedících článků systému. Skládá se ze čtyř parametrů, které popisují vzájemnou polohu a natočení:

1. **Alpha** ( $\alpha$ ): úhel rotace kolem osy X
2. **A** (**a**): posun vzdálenosti na ose X
3. **Theta** ( $\theta$ ): úhel rotace kolem osy Z
4. **D** (**d**): posun vzdálenosti na ose Z

Pro nalezení všech parametrů mezi klouby můžeme využít následující obrázek:



Obr. 3.1: Parametry DH konvence [12]

Vždy je potřeba vyjádřit tyto parametry mezi všemi články robotického systému. Nutno podotknout, že do celkového počtu článků robotického systému se řadí i základna a koncový bod celého systému. Z této analýzy nám vznikne tzv. DH tabulka. Z této tabulky dosadíme každý řádek do základního tvaru transformační matice:

$$\mathbf{T} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cos(\alpha_i) & \sin(\theta_i) \sin(\alpha_i) & a_i \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cos(\alpha_i) & -\cos(\theta_i) \sin(\alpha_i) & a_i \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Tyto transformační matice následně ve správném pořadí vynásobíme mezi sebou, z čehož dostaneme celkovou transformační matici systému.

DH konvence je univerzální přístup k řešení kinematiky robotu a poskytuje víceméně jednoduchý a systematický popis, který usnadňuje analýzu a modelování robotických systémů.

## 3.2 Dynamika

Dynamika robotů je klíčovým aspektem v oblasti robotiky, který se zabývá studiem pohybu a chování robotických systémů. Jedná se o klíčový aspekt pro řízení a optimalizaci robotických systémů. Na rozdíl od kinematiky zahrnuje vliv sil a momentů, jež působí na jejich konstrukci během manipulačních operací. Tento proces zahrnuje identifikaci jednotlivých článků, jejich geometrické vlastnosti (jako jsou délky článků a polohy kloubů) a fyzikální vlastnosti (hmotnost, moment setrvačnosti). V této kapitole budeme zkoumat základní principy dynamiky robotických manipulátorů a představíme si metody pro modelování jejich chování.

K odvození pohybových rovnic je možné použít dva základní přístupy:

### Newton-Eulerova metoda

Tato metoda využívá Newtonovy zákony pohybu a Eulerovy rovnice pohybu. Funguje na principu rozdělení systému na jednotlivé části, u kterých se určí pohybové rovnice. Výsledná rovnice je potom součet těchto jednotlivých částí. Tato metoda je vhodná zejména pro roboty s méně složitou strukturou.

### Euler-Lagrangeova metoda

Jedním z efektivních přístupů k odvození diferenciálních rovnic, které popisují dynamiku robotu, je využití Lagrangeových rovnic druhého druhu. [27] Tento postup zahrnuje výpočet celkové kinetické a potenciální energie systému a následné odvození dynamických rovnic pomocí Lagrangeových rovnic pro všechny stupně volnosti systému. Jedna z hlavních výhod této metody spočívá v tom, že umožňuje řešení dynamických rovnic v symbolické uzavřené formě, což usnadňuje porozumění chování systému a umožňuje analytické postupy.

Přepsání dynamických rovnic do maticového tvaru pak umožňuje vytvoření univerzálního nástroje pro tvorbu dynamického modelu, což je užitečné pro simulační účely, návrh regulátorů a další inženýrské aplikace. Maticový tvar dynamického modelu poskytuje přehlednou a kompaktní reprezentaci systému, což usnadňuje jeho analýzu a implementaci různých řídicích strategií.

Přepis do maticového tvaru vypadá následovně:

$$\mathbf{M}(q) \cdot \ddot{q} + \mathbf{C}(q, \dot{q}) \cdot \dot{q} + \mathbf{G}(q) = u \quad (3.2)$$

kde:

- $q$  je vektor zobecněných proměnných
- $\dot{q}$  je vektor rychlostí zobecněných proměnných
- $\ddot{q}$  je vektor zrychlení zobecněných proměnných
- $M(q)$  je matice hmotností

- $C(q, \dot{q})$  je matice Coriolisových a odstředivých sil
- $q(q)$  je vektor gravitačních sil
- $u$  je vektor momentu

Za zobecněné proměnné v robotice obvykle nejčastěji volíme kloubové proměnné, což jsou polohy, rychlosti a zrychlení jednotlivých os robota. Tyto veličiny jsou vzájemně propojeny pomocí derivací jedna od druhé. V důsledku toho jsou dynamické rovnice popisující chování robotického systému obvykle nelineární. Euler-Lagrangeova metoda umožňuje práci s těmito nelineárními dynamickými rovnicemi a poskytuje mechanismy pro jejich řešení a analýzu. I když maticový zápis této metody může být náročnější, je univerzálnější pro různé kinematické struktury.

Představme si nyní jednotlivé části dynamické rovnice.

### 3.2.1 Matice hmotností

Matice hmotností (Homogenní matice setrvačnosti, Inertia matrix) s označením  $\mathbf{M}$  (v některých, zejména anglických literaturách se můžeme setkat s označením  $\mathbf{H}$ ) je pro modelování dynamiky klíčovým prvkem. Udává nám, jakým způsobem je v závislosti na obecných souřadnicích (vektor  $\mathbf{q}$ ) jednotlivých ramen distribuována hmotnost a rozložení momentů setrvačnosti. Matice hmotností je symetrická a je pozitivně definitní. To znamená, že prvky této matice jsou pouze reálná vlastní kladná čísla. Z definice symetričnosti je zřejmé, že se vždy bude jednat o matici čtvercovou, která bude mít počet řádků a sloupců podle toho, kolik stupňů volnosti daný robotický systém obsahuje. Každá jednotlivá složka se následně počítá dle vzorce:

$$H_{j,l} = \sum_{k=\max(j,l)}^n \text{tr}(U_j^k \cdot J_k \cdot (U_l^k)^T) \quad \text{pro } j, l = 1, \dots, n \quad (3.3)$$

kde:

- $U$  jsou parciální derivace transformačních matic podle úhlu
- $J$  jsou matice setrvačnosti

Nejprve tedy bude zapotřebí sestavit matice momentů setrvačností a spočítat parciální derivace transformačních matic  $\mathbf{T}$ .

#### Matice momentů setrvačnosti

Hlavními prvky této matice, ležícími na diagonále, jsou prvky, které popisují distribuci hmotnosti v jednotlivých člácích robota a jak se rozložení hmotnosti projeví na rotaci těchto článků kolem hlavních os. Označujeme je  $J_{xx}$ ,  $J_{yy}$  a  $J_{zz}$ .

Dalšími prvky, která matice obsahuje, jsou tzv. deviační momenty. Ty nám popisují rotaci kolem "vedlejších" os. V podstatě se jedná o momenty mezi osami.



Stane-li se, že tento prvek není nulový, znamená to, že hmotnost článku systému není rovnoměrně rozložena vzhledem k daným osám.

Posledními prvky jsou hmotnostní charakteristiky. Ty obsahují informace o hmotnostech a polohách hmotných bodů v každém článku. Tyto informace jsou zahrnuty v posledním sloupci a posledním řádku matice  $\mathbf{J}$ .

Matice momentů setrvačnosti pak pro každý robotický článek vypadá následujícím způsobem:

$$J_i = \begin{bmatrix} J_{xxi} & J_{xyi} & J_{xzi} & m_i \cdot x_{ti} \\ J_{xyi} & J_{yyi} & J_{yzi} & m_i \cdot y_{ti} \\ J_{xzi} & J_{yzi} & J_{zzi} & m_i \cdot z_{ti} \\ m_i \cdot x_{ti} & m_i \cdot y_{ti} & m_i \cdot z_{ti} & m_i \end{bmatrix} \quad (3.4)$$

### Parciální derivace transformačních matic

Tyto derivace nám popisují, jak se mění prvky přechodové matice  $\mathbf{T}$  v závislosti na změně jednotlivých úhlů  $\mathbf{q}$ . Důležité je určit, jakou matici  $\mathbf{T}$  derivujeme jakým  $\mathbf{q}$ . Rozhodujeme se na základě toho, který úhel je přímo spojen s pohybem, který chceme analyzovat.

### 3.2.2 Matice Coriolisových a odstředivých sil

Coriolisův efekt je fyzikální jev, který popisuje odchylku pohybu tělesa v rotačním referenčním systému. V dynamice pohybu robotů může tento efekt způsobit nežádoucí odchylky a vibrace, které je pro správnou analýzu potřeba brát v úvahu. [13]

Matice Coriolisových sil, označována ( $\mathbf{C}$ ), obsahuje důležité informace o vztazích mezi vektorovými rychlostmi článků systému, úhlovými rychlostmi ( $\dot{q}$ ) a vznikajícími Coriolisovými a středovými silami. Tyto síly ovlivňují chování a stabilitu robota v průběhu pohybu. Tato matice nám umožňuje přesnější simulace a řízení pohybů robota.

Matice je dána vztahem:

$$C(q, \dot{q}) = \frac{1}{2} \dot{M}(q) + S(q, \dot{q}) \quad (3.5)$$

kde:

- $\dot{M}$  je derivace matice hmotností
- $S$  je antisymetrická matice

Matici hmotností jsme si již rozebrali dříve, nyní jen spočteme její derivaci. Věnujme se tedy matici  $\mathbf{S}$ . Ta bude vždy čtvercová a jak již bylo zmíněno, tato matice je antisymetrická, což znamená, že její transponovaná hodnota se rovná opačné hodnotě původní matice. Vztahem vyjádřeno:  $S^T = -S$ .

Tato matice popisuje vliv rotace jednotlivých článků a jejich rychlostí na vznikající Coriolisovy síly. Často se může stát, že její prvky jsou nulové. To je způsobeno



tím, že není přítomen Coriolisův efekt a nebo je tak malý, že může být zanedbán. Tato situace může nastat v situacích, kdy robot provádí pouze lineární pohyby a nebo když se robot pohybuje velmi malou rychlostí. [13] [11]

Jednotlivé prvky matice  $\mathbf{S}$  se tedy počítají podle následujícího vzorce:

$$S_{i,j} = \sum_{k=1}^n \left( \dot{q}_k \cdot \frac{\partial M_{ik}}{\partial q_j} - \dot{q}_k \cdot \frac{\partial M_{jk}}{\partial q_i} \right) \quad \text{pro } i, j = 1, \dots, n \quad (3.6)$$

### 3.2.3 Vektor gravitačních sil

Gravitační síly jsou dalšími silami, které přispívají k celkovým silám a momentům, působícím na robotický systém. Jednotlivé složky tohoto vektoru obsahují informaci o velikosti síly, působící na jednotlivé články robota v závislosti na konfiguraci  $\mathbf{q}$ . Pokud má tedy robotický systém  $n$  článků, bude gravitační vektor obsahovat  $n$  prvků. Pro přesnou analýzu a následné řízení je nutné tento vektor správně zahrnout do matematického modelu.

Pro výpočet složek vektoru je nutné nejdříve spočítat potenciální energii systému. Ta se rovná součtu potenciálních energií všech jednotlivých částí. Vyjádřeno vzorcem:

$$P = - \sum_{i=1}^n (m_i \cdot G^T \cdot T_0^i \cdot R_{ic}) \quad (3.7)$$

kde:

- $m$  je hmotnost článku
- $G$  je gravitační matice
- $T$  je transformační matice
- $R$  je matice polohy těžiště

Gravitační matice nám udává, v jakém směru, nebo spíše v jaké ose na náš systém působí gravitační síla. V jaké ose bude tato síla působit záleží na zvolení souřadného systému. Podle osy působení je pak na určitém místě matice gravitační konstanta a na místech ostatních jsou nuly.

Ve valné většině případů bude gravitační síla působit záporně v ose  $z$ , proto bude mít následující tvar:

$$G = [0 \quad 0 \quad -g \quad 0] \quad (3.8)$$

Matice polohy těžiště má svůj daný tvar, do kterého se zaznamená jeho poloha ve všech osách. Pro každou osu pak vypadá následovně:

$$R_{ic} = \begin{bmatrix} x_{ti} \\ y_{ti} \\ z_{ti} \\ 1 \end{bmatrix} \quad \text{pro } i = 1, \dots, n \quad (3.9)$$

Nyní už je možnost vypočítat gravitační vektor. Ten dostaneme dosazením potenciální energie do vzorce:

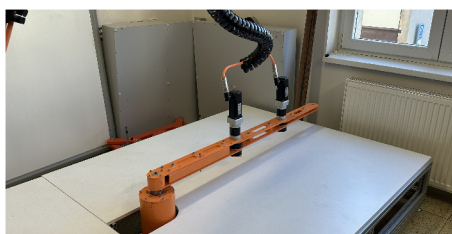
$$g(q) = \left( \frac{\partial P}{\partial q} \right)^T \quad (3.10)$$

Výsledný vektor bude mít tolik prvků, kolik je v daném robotickém systému článků.

## 4 Matematický model planárního robotu

V této kapitole využijeme poznatky z předchozí kapitoly k tomu, abychom si analyticky sestavili matematický model planárního robotu se třemi stupni volnosti. Následně využijeme možnosti Robotics System Toolboxu, který je dostupný v softwaru Matlab. Jakmile budeme mít obě varianty, přejdeme do simulačního prostředí Matlab Simulink, kde porovnáme shodu výsledku z obou těchto variant.

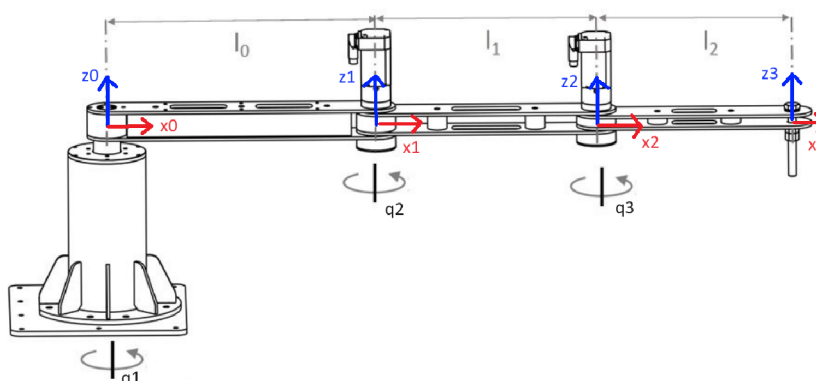
Naším cílem je tedy vytvořit matematický model pro planárního robota:



Obr. 4.1: Planární laboratorní robot

Planární roboti jsou charakterizováni svou schopností pohybu v rovině, což znamená, že mají omezenou nebo zcela chybějící schopnost rotace kolem 2 os. Tento laboratorní robot vznikl jako zmenšená a zjednodušená verze unikátního tiskového robotu "Printing Mantis", který vznikl, v rámci projektu 3D STAR, na Technické univerzitě v Liberci. [2]

Volbu souřadných systémů zvolených dle DH konvence dokumentuje následující obrázek:



Obr. 4.2: Schéma robotického systému [14]

Odvození matematického modelu bude nejprve provedeno v symbolické podobě a následně budou dosazeny numerické hodnoty jednotlivých parametrů. Ty následně využijeme až pro simulační účely.

## 4.1 Kinematika planárního robotu

Jako první věc, kterou je třeba udělat, je DH tabulka. Ze schématu je zřejmé, že v našem systému bude docházet pouze k rotaci v ose  $z$  a posunech v ose  $x$ . Všechny ostatní parametry budou nulové. DH tabulka bude tedy vypadat následujícím způsobem:

	$\theta$	$d$	$a$	$\alpha$
0 – 1	$q_1$	0	$l_0$	0
1 – 2	$q_2$	0	$l_1$	0
2 – 3	$q_3$	0	$l_2$	0

Table 4.1: Denavit-Hartenbergova matice

Nyní vytvoříme transformační matice pro každý řádek DH tabulky, tedy pro přechod mezi všemi klouby systému:

$$T_{01} = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & L_0 \cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & L_0 \sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$T_{12} = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & L_1 \cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & L_1 \sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$T_{23} = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & L_2 \cos(q_3) \\ \sin(q_3) & \cos(q_3) & 0 & L_2 \sin(q_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

Výslednou transformační matici (značenou  $\mathbf{T}_{03}$ ) tedy dostaneme vynásobením výše uvedených matic. Ta je z důvodu své velikosti k nahlédnutí pouze v kódu programu, tedy příloze 1. Nyní pomocí této matice můžeme kdykoliv dopočítat polohu koncového bodu robotu, pokud by to pro něco bylo potřeba.

## 4.2 Dynamika planárního robotu

Nyní se přesuneme na řešení dynamiky. Nejprve si sestavíme matice setrvačností. Ty pro nás v tuto chvíli, jelikož stále bereme všechny proměnné symbolicky, budou

mít základní tvar, uvedený v kapitole 3.2.1. Nyní provedeme parciální derivace potřebné pro výpočet složek hmotnostní matice:

$$U_{11} = \frac{\partial T_1}{\partial q_1} \quad U_{12} = \frac{\partial T_{02}}{\partial q_1} \quad U_{13} = \frac{\partial T_{03}}{\partial q_1} \quad (4.4)$$

$$U_{22} = \frac{\partial T_{02}}{\partial q_2} \quad U_{23} = \frac{\partial T_{03}}{\partial q_2} \quad U_{33} = \frac{\partial T_{03}}{\partial q_3} \quad (4.5)$$

K parciálním derivacím transformačních matic podle jednotlivých zobecněných proměnných lze využít diferenciální operátor rotace (derivační matice) jak je definováno v [27]. Jak již bylo zmíněno dříve, není potřeba počítat derivace všech převodových matic podle každé konfigurace, protože víme, že matice  $\mathbf{H}$  je symetrická.

Výpočet prvků matice  $\mathbf{H}$  bude tedy dle vzorce z kapitoly 3.2.1 vypadat:

$$H_{11} = \text{trace}(U_{11}J_1(U_{11})^T) + \text{trace}(U_{12}J_2(U_{12})^T) + \text{trace}(U_{13}J_3(U_{13})^T) \quad (4.6)$$

$$H_{12} = \text{trace}(U_{12}J_2(U_{22})^T) + \text{trace}(U_{13}J_3(U_{23})^T) \quad (4.7)$$

$$H_{21} = H_{12} \quad (4.8)$$

$$H_{13} = \text{trace}(U_{13}J_3(U_{33})^T) \quad (4.9)$$

$$H_{31} = H_{13} \quad (4.10)$$

$$H_{22} = \text{trace}(U_{22}J_2(U_{22})^T) + \text{trace}(U_{23}J_3(U_{23})^T) \quad (4.11)$$

$$H_{23} = \text{trace}(U_{23}J_3(U_{33})^T) \quad (4.12)$$

$$H_{32} = H_{23} \quad (4.13)$$

$$H_{33} = \text{trace}(U_{33}J_3(U_{33})^T) \quad (4.14)$$

Výsledky z těchto operací jsou správné, ale obsahují spousty zanořených funkcí, které zahrnují mnoho proměnných a trigonometrických funkcí. Tudíž se naskýtá možnost je ještě nějak upravit, aby byly výrazy jednodušší a lépe čitelné. K tomu použijeme funkci **simplify()**, která je implementována v matlabu. Tato funkce provádí různé algebraické operace, jako je například redukce podobných členů. [17] Zjednodušení prvků a vytvoření matice  $\mathbf{H}$  pak vypadá následovně:

$$H_{11} = \text{simplify}(H_{11}) \quad (4.15)$$

$$H_{12} = \text{simplify}(H_{12}) \quad (4.16)$$

$$H_{21} = H_{12} \quad (4.17)$$

$$H_{13} = \text{simplify}(H_{13}) \quad (4.18)$$

$$H_{31} = H_{13} \quad (4.19)$$

$$H_{22} = \text{simplify}(H_{22}) \quad (4.20)$$

$$H_{23} = \text{simplify}(H_{23}) \quad (4.21)$$

$$H_{32} = H_{23} \quad (4.22)$$

$$H_{33} = \text{simplify}(H_{33}) \quad (4.23)$$

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (4.24)$$

Nyní budeme s touto maticí dále pracovat a zhotovíme si antisymetrickou matici **S**. Všimněme si, že na diagonále jsou nuly. Nuly na těchto místech budou vždy, protože na diagonále jsou derivace vzhledem k sobě samým. Z definice antisymetričnosti vyplývá, že nám dle vzorce stačí dopočítat pouze 3 hodnoty.

$$s_{11} = 0$$

$$s_{22} = 0$$

$$s_{33} = 0$$

$$s_{12} = \frac{1}{2} \left( q_{1,d} \cdot \frac{\partial H_{11}}{\partial q_2} - q_{1,d} \cdot \frac{\partial H_{21}}{\partial q_1} + q_{2,d} \cdot \frac{\partial H_{12}}{\partial q_2} - q_{2,d} \cdot \frac{\partial H_{22}}{\partial q_1} + q_{3,d} \cdot \frac{\partial H_{13}}{\partial q_2} - q_{3,d} \cdot \frac{\partial H_{23}}{\partial q_1} \right)$$

$$s_{21} = -s_{12}$$

$$s_{13} = \frac{1}{2} \left( q_{1,d} \cdot \frac{\partial H_{11}}{\partial q_3} - q_{1,d} \cdot \frac{\partial H_{31}}{\partial q_1} + q_{2,d} \cdot \frac{\partial H_{12}}{\partial q_3} - q_{2,d} \cdot \frac{\partial H_{32}}{\partial q_1} + q_{3,d} \cdot \frac{\partial H_{13}}{\partial q_3} - q_{3,d} \cdot \frac{\partial H_{33}}{\partial q_1} \right)$$

$$s_{31} = -s_{13}$$

$$s_{23} = \frac{1}{2} \left( q_{1,d} \cdot \frac{\partial H_{21}}{\partial q_3} - q_{1,d} \cdot \frac{\partial H_{31}}{\partial q_2} + q_{2,d} \cdot \frac{\partial H_{22}}{\partial q_3} - q_{2,d} \cdot \frac{\partial H_{32}}{\partial q_2} + q_{3,d} \cdot \frac{\partial H_{23}}{\partial q_3} - q_{3,d} \cdot \frac{\partial H_{33}}{\partial q_2} \right)$$

$$s_{32} = -s_{23}$$

Když máme všechny prvky, sestavíme z nich výslednou matici. Na výsledek ještě aplikujeme funkci **simplify()**.

$$S = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \quad (4.25)$$

$$S = \text{simplify}(S) \quad (4.26)$$

Ještě před zhotovením matice **C** je zapotřebí malá programová úprava. Jelikož stále pracujeme se symbolickými proměnnými, musíme udělat substituci symbolických proměnných za proměnné časové v hmotnostní matici. Tyto změny nám umožní časovou derivaci této matice, kterou budeme potřebovat pro výpočet. Pokud bychom od začátku pracovali s čísly, tato úprava by nebyla třeba.

K tomuto kroku použijeme funkci **subs()**. Následně takto upravenou matici **H** zderivujeme.

$$H_t = \text{subs}(H, q_1, q_2, q_3, f_1(t), f_2(t), f_3(t)) \quad (4.27)$$

$$H_{t_d} = \text{diff}(H_t, t) \quad (4.28)$$

Nyní už vše dosadíme do vzorce a sestavíme matici Coriolisových sil:

$$C = (1/2 \cdot * H_{pm}) + S \quad (4.29)$$

Posledním článkem, který je nutné vypočítat a je potřebný pro sestavení rovnice dynamiky, je gravitační vektor. Nejprve sestavíme matice polohy těžišť, ze kterých vypočítáme celkovou potenciální energii našeho systému. Z té už pak pomocí derivací dostaneme jednotlivé složky gravitačního vektoru.

$$Gt = [0 \quad 0 \quad -g \quad 0] \quad (4.30)$$

$$R1c = \begin{bmatrix} xt1 \\ yt1 \\ zt1 \\ 1 \end{bmatrix} \quad (4.31)$$

$$R2c = \begin{bmatrix} xt2 \\ yt2 \\ zt2 \\ 1 \end{bmatrix} \quad (4.32)$$

$$R3c = \begin{bmatrix} xt3 \\ yt3 \\ zt3 \\ 1 \end{bmatrix} \quad (4.33)$$

$$P1 = -m1 \cdot Gt \cdot T_1 \cdot R1c \quad (4.34)$$

$$P2 = -m2 \cdot Gt \cdot T_{02} \cdot R2c \quad (4.35)$$

$$P3 = -m3 \cdot Gt \cdot T_{03} \cdot R3c \quad (4.36)$$

$$P = P1 + P2 + P3 \quad (4.37)$$

$$fg1 = \text{diff}(P, q1) \quad (4.38)$$

$$fg2 = \text{diff}(P, q2) \quad (4.39)$$

$$fg3 = \text{diff}(P, q3) \quad (4.40)$$

$$G = \begin{bmatrix} fg1 \\ fg2 \\ fg3 \end{bmatrix}^T \quad (4.41)$$

V našem případě vyjdou všechny složky gravitačního vektoru nulové, jelikož náš robotický systém je planární a jeho pohyb vůči vertikální ose je pouze rotační. To znamená, že jeho pohyb nebude nijak ovlivněn gravitační silou.

V tento moment již máme sestaveny všechny matice potřebné pro výpočet momentu dle rovnice 3.2:

$$u = H_{pm} * qdd + C * qd + G * q \quad (4.42)$$

Nyní jsme analyticky (v symbolickém tvaru) sestavili dynamickou rovnici našeho systému. Výsledné matice jsou vzhledem k jejich velikosti v symbolickém tvaru k nahlédnutí v příloze 1.

### 4.3 Robotics system toolbox

Než přejdeme do simulačního prostředí a ověříme, zda je naše analytické řešení správné, musíme nejprve sestavit model v robotics system toolboxu. Tento toolbox je implementován a k dispozici v Matlabu. Nabízí nám možnost nástrojů a algoritmů pro designování, simulace a testování. Máme v něm možnost modelování kinematiky a dynamiky velké škály robotických systémů. [18] My ho pro účely této práce použijeme jako kontrolní faktor pro naše analytické řešení.

Pro práci v toolboxu už budeme potřebovat data robotického systému, jako jsou délky ramen, hmotnosti článků, polohy těžišť a momenty setrvačnosti každého ramene. Tyto parametry pro prvotní simulace byly použity setrvačné parametry získané z 3D modelu zhotoveným v CADu. [14] K tomu přidáme ještě DH tabulku, kterou už máme sestavenou z analytického řešení. Zde ji budeme zavádět následujícím způsobem:

$$\text{dhparams} = \begin{bmatrix} l_0 & 0 & 0 & 0 \\ l_1 & 0 & 0 & 0 \\ l_2 & 0 & 0 & 0 \end{bmatrix} \quad (4.43)$$

Pro rotace v ose  $\mathbf{z}$  použijeme nuly, protože zde zavádíme počáteční stav systému.

Nyní je zapotřebí definovat základní strukturu robotu. Ta je v toolboxu označována jako **rigidBodyTree**. Nejdůležitější informace této funkce je, kolik článků systém obsahuje. Tyto části pak jednu po druhé definujeme. Následně specifikujeme klouby, které spojují jednotlivé části. Zde je nutno uvést jestli se jedná o kloub otočný, posuvný či fixovaný. V tuto chvíli už jen nastavíme natočení kloubů podle DH tabulky a propojíme jednotlivá ramena. Kód, který provádí tyto činnosti je následující:



```

%definuje strukturu robota:
planar_3DOF = rigidBodyTree('DataFormat','row', 'MaxNumBodies', 3);

body1 = rigidBody('body1'); %specifikuje pevné těleso
jnt1 = rigidBodyJoint('jnt1','revolute'); %natočení v kloubu
body2 = rigidBody('body2'); %2. rameno
jnt2 = rigidBodyJoint('jnt2','revolute'); %natočení v kloubu
body3 = rigidBody('body3'); %3. rameno
jnt3 = rigidBodyJoint('jnt3','revolute');%natočení v kloubu

%nastaví natočení podle DH tabulky
setFixedTransform(jnt1,dhparams(1,),'dh');
setFixedTransform(jnt2,dhparams(2,),'dh');
setFixedTransform(jnt3,dhparams(3,),'dh');

%propojíme ramena
body1.Joint = jnt1;
body2.Joint = jnt2;
body3.Joint = jnt3;

```

Obr. 4.3: Struktura robotického systému v toolboxu

Po přípravě této fáze přejdeme k nastavení hmotnosti, umístění těžiště a momentů setrvačnosti pro každé rameno. Nakonec je nutné znovu propojit všechny části a určit gravitační vektor.

```

%Nastavení momentů a těžiště:
body1.Mass = m1;
body1.CenterOfMass = [xt1 yt1 zt1];
body1.Inertia = [Jxx1 Jyy1 Jzz1 Jyz1 Jxz1 Jxy1];

body2.Mass = m2;
body2.CenterOfMass = [xt2 yt2 zt2];
body2.Inertia = [Jxx2 Jyy2 Jzz2 Jyz2 Jxz2 Jxy2];

body3.Mass = m3;
body3.CenterOfMass = [xt3 yt3 zt3];
body3.Inertia = [Jxx3 Jyy3 Jzz3 Jyz3 Jxz3 Jxy3];

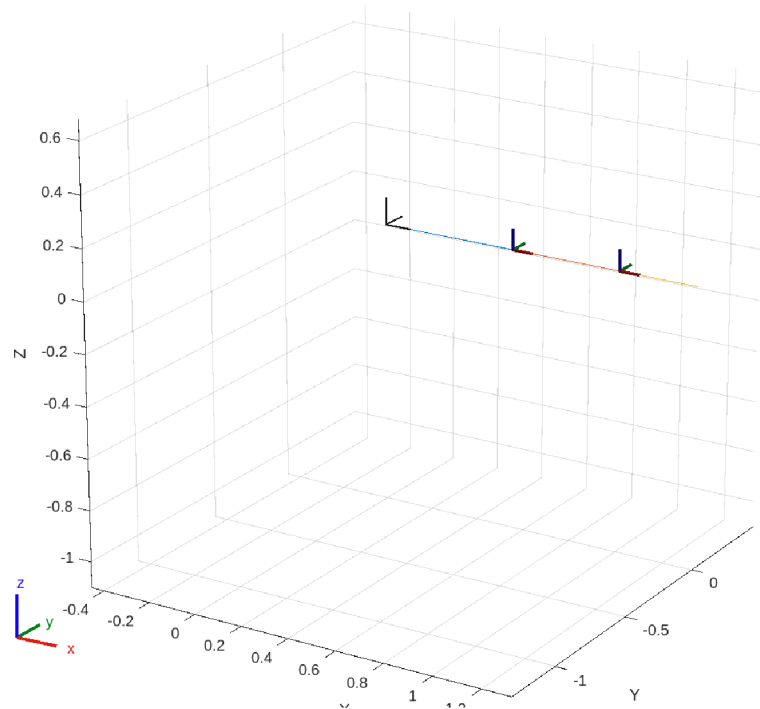
%propojení jednotlivých částí
addBody(planar_3DOF,body1,'base');
addBody(planar_3DOF,body2,'body1');
addBody(planar_3DOF,body3,'body2');

planar_3DOF.Gravity = [0 0 -9.81];

```

Obr. 4.4: Nastavení hodnot pro jednotlivá ramena

V tento moment máme připravený model robotu v toolboxu. Můžeme si ještě zobrazit, jakým stylem vypadají články robota v home pozici. To provedeme pomocí příkazu `show()`.



Obr. 4.5: Home pozice planárního robotu

## 4.4 Simulační porovnání

Vrhněme se tedy na ověření správnosti analytického řešení. Jak jsem již zmínil dříve, budeme pracovat v prostředí Matlab Simulink. Přenesme teď tedy obě řešení do tohoto prostředí a porovnejme je.

Pro analytické řešení použijeme blok **Matlab function**, kam nahrajeme naše řešení. Jelikož máme všechny matice vyjádřené symbolicky, nemusíme řešení vkládat celé a stačí nám pouze matice pro výpočet výsledného momentu (tedy  $\mathbf{H}$ ,  $\mathbf{C}$  a  $\mathbf{G}$ ). Ovšem na začátek programu musíme přidat všechny hodnoty, které jsou potřeba k výpočtu. Vstupem do tohoto bloku budou jednotlivé konfigurace, rychlosti a zrychlení pro každý článek robotu. Výstupem pak vypočtený moment.

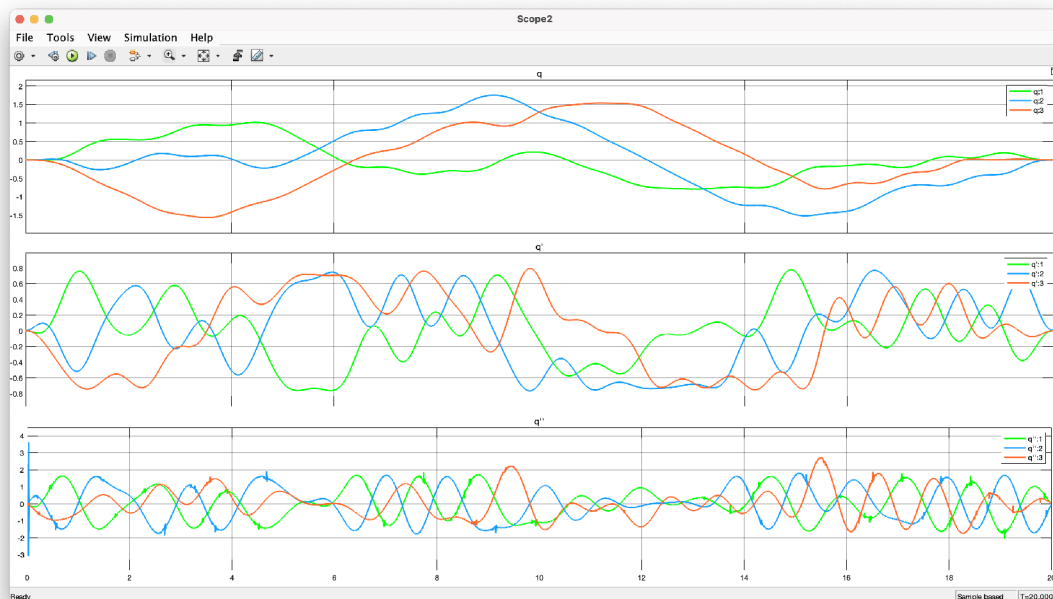
Pro simulaci řešení v Robotics system toolbox použijeme jeden z bloků, které nabízí simulink knihovna tohoto toolboxu. Tento blok se nazývá **Inverse Dynamics**. Jako vstupy do tohoto bloku jsou opět jednotlivé konfigurace, rychlosti a zrychlení. Do bloku se pak jako jediná věc zadává název vytvořeného modelu, neboli model vytvořený funkcí **rigidBodyTree**. Výstupem tohoto bloku je poté vypočtený moment.

Poslední věc, nezbytná pro simulaci, je trajektorie, po které bude robot provádět pohyb. Bez této trajektorie není možné odsimulovat chování systému. K tomuto máme dvě možnosti, tou první je použití bloku z knihovny toolboxu. Tento blok se nazývá **Polynomial trajectory**. Ten nám dle zadaných parametrů vygeneruje požadovaně dlouhou trajektorii. Více o tomto bloku je možné se dočíst zde [4]. Druhou možností je programově zhotovit strukturu a použít blok **From workspace**.

V této struktuře musí jako první hodnota být údaj o vzorkování (tedy časová proměnná) a následně údaje o konfiguracích, rychlostech či zrychleních.

Já použiji vytvořené struktury, které jsem zhotovil z dat z měření na robotu. O měření se budeme bavit později v této práci, proto to zde nebude rozebráno. Zhotovení struktur je k nahlédnutí v příloze 1. Simulační schéma, na kterém je prováděno následující srovnání a je k dispozici jako příloha 2.

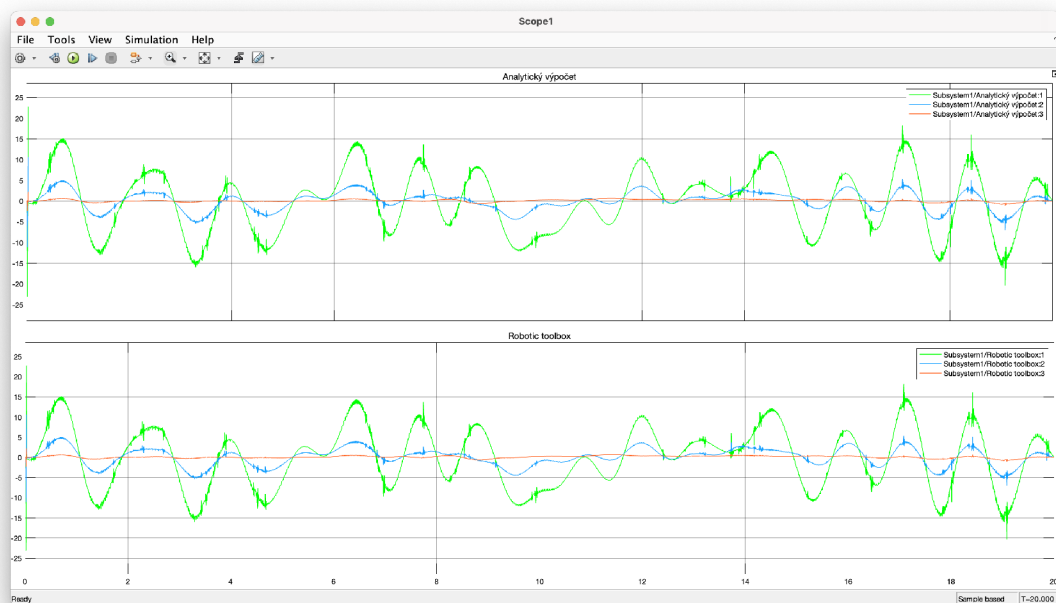
Na obrázku 4.6 je vidět vstupní trajektorie, která je použita pro zobrazení a porovnání výsledků.



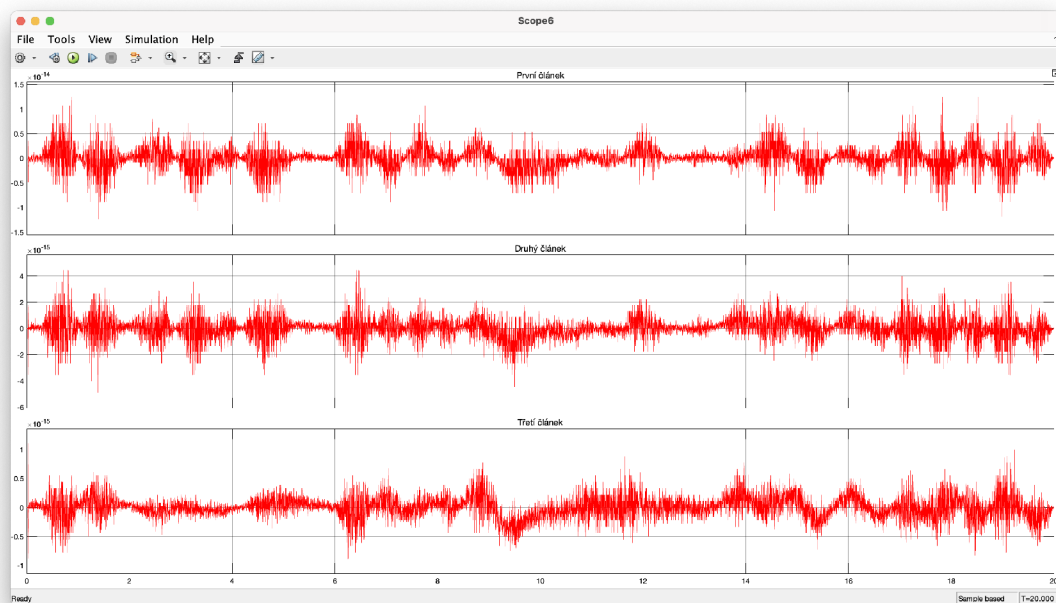
Obr. 4.6: Vstupní trajektorie pro ověření výsledků

Obrázek 4.7 nám ukazuje, jak vypadají výsledné momenty jak z analytického, tak z toolbox řešení. Tyto dva grafy nám neřeknou nic o tom, zda jsou modely shodné a tedy jestli lze říci, že je naše analytické řešení správné. Nyní je nutné si ještě zobrazit diferenci těchto dvou výsledků. Tím si zobrazíme, zda se výsledky shodují. Teoreticky by tento rozdíl měl být nulový, ale toho prakticky nejde dosáhnout. Pojďme tedy porovnat rozdíl na obrázku 4.8.

Při pohledu na tento graf pro nás není až tak důležité, jak rozdíly vycházejí. Pro nás nejdůležitější informací je, v jakých řádech se rozdíl pohybuje. V našem porovnání se rozdíly pohybují v řádech  $10^{-14}$  u prvního ramene a  $10^{-15}$  u druhého a třetího ramene. Z těchto dat nyní lze s jistotou prohlásit, že analytické řešení je správné a lze s ním pracovat dále.



Obr. 4.7: Vypočtené momenty z analytického výpočtu a toolbox modelu



Obr. 4.8: Rozdíl výsledných momentů

## 5 Identifikace

V této kapitole se budeme zabývat identifikací dynamických parametrů. Důležitost identifikace dynamických parametrů spočívá v tom, že umožňuje lépe porozumět chování manipulátoru v reálném prostředí a vytvořit efektivnější řídicí algoritmy. Přesné znalosti parametrů umožňují dosáhnout přesnějších a robustnějších řídicích strategií, což vede k vyšší přesnosti a výkonu manipulátoru v různých aplikacích. Problematiku identifikace lze rozdělit do následujících kroků. [23]

### 5.1 Tvorba dynamického modelu

Nejprve je třeba určit přesný dynamický model daného systému. Ten je pro identifikaci velmi důležitý. Tuto problematiku již máme probranou z kapitoly 4.2.

### 5.2 Vektor koeficientů

Dynamické rovnice obvykle obsahují nelineární vztahy, které vycházejí z interakcí mezi polohami, rychlostmi a zrychleními. Nicméně, u robotických manipulátorů jsme schopni nalézt určitou linearitu mezi jednotlivými parametry, což umožňuje přepisovat nelineární rovnice do lineární a tím výrazně zjednodušit proces identifikace.

Nyní lze přepsat rovnici dynamiky do lineární podoby následujícím způsobem:

$$u = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \rightarrow Y(q, \dot{q}, \ddot{q})\beta = u \quad (5.1)$$

kde:

- $Y$  je matice regresorů
- $\beta$  je vektor dynamických koeficientů

Dynamické koeficienty jsou tvořeny kombinacemi dynamických parametrů (těch je v obecnosti 10 pro každý člen robotu) a jejich stanovení není jednoznačné. Provádět identifikaci přímo dynamických parametrů by také bylo možné. Ovšem musíme vzít v úvahu, že tyto parametry jsou často lineárně závislé, a tudíž by se do identifikace projevíly vícekrát a zhoršily by se tak výsledky. Snaha u volby koeficientů je tedy kladena na lineární nezávislost mezi jednotlivými koeficienty. Je klíčové nalézt

správnou kombinaci parametrů, protože špatně zvolené koeficienty mohou vést k chybným výsledkům, bez ohledu na další práci. Možností, jak je získat existuje více, v této práci je použita dekompozice, která je implementována v BIRDy. [24] [15] Problematiku nalezení koeficientů pomocí jiných metod je možnost dohledat v [10] [22]

## 5.3 Identifikační experiment a trajektorie

Identifikační trajektorie jsou klíčovým prvkem pro správné provedení identifikace. Tyto trajektorie jsou pečlivě navrženy tak, aby vybudily všechna chování daného systému. Zároveň nám musí poskytnout dostatečné množství použitelných dat, abychom na nich mohli postavit další procedury. Experiment by měl obsahovat všechny možné scénáře pro všechny osy. To zahrnuje například změny rychlosti pohybu, variabilitu zrychlení pohybu, periodické opakování nebo trajektorie, které simulují pracovní proces, který bude daný manipulátor vykonávat. Zároveň by měl brát v úvahu možná omezení pohybu, jako jsou koncové body pracovního prostoru. Je i vhodné pohybovat se různými rychlostmi kolem těchto bodů, protože chování kolem nich by mohlo být v určitých případech kritické.

U tohoto experimentu je velmi důležité i samotné měření výstupních dat. Je nutné zvolit ideální vzorkovací frekvenci. Ta by měla být dostatečně rychlá, abychom mohli zachytit veškeré změny. Zde se bavíme o frekvencích menších než řád  $10^{-3}$ .

Společně s identifikační trajektorií je více než vhodné naměřit i trajektorii validační. Tu použijeme po samotné identifikaci, abychom ověřili správnost získaných výsledků. Je totiž jasné, že identifikované parametry budou vycházet mnohem přesněji na trajektorii, na které byly identifikovány.

Lze tedy říci, že návrh takovéto trajektorie je klíčovým úspěchem samotné identifikace. Pojdme se podívat na to, jak se taková trajektorie navrhuje.

### 5.3.1 Návrh identifikační trajektorie

Pro tvorbu excitační trajektorie v identifikačním experimentu se často využívá Fourierova řada, která umožňuje aproximovat periodickou funkci pomocí součtu sinusových a kosinových funkcí. Tento matematický nástroj poskytuje flexibilitu v návrhu trajektorií a umožňuje generovat různé typy pohybů, které jsou vhodné pro identifikaci dynamických koeficientů robotu. Ostatně je to odborně popsáno ve vědeckých pracích [26] nebo [25].

Pro tvorbu excitační trajektorie se používá konečná Fourierova řada, kde je počet harmonických funkcí omezen na určitý počet  $N$ . Tato aproximace má poté následující tvar:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^N (a_n \cdot \cos(n\omega t) + b_n \cdot \sin(n\omega t)) \quad (5.2)$$

kde:

- Kde  $f(t)$  je periodická funkce s periodou  $T$

- $\omega$  je úhlová frekvence
- $a_0, a_n, b_n$  jsou koeficienty Fourierovy řady

## 5.4 Identifikační metody

Identifikační metody nám slouží k odhadu dynamických koeficientů na základě experimentálních dat. V dnešní době již existuje spousta metod, od základních až po metody založených na neuronových sítích. Některé využívají otevřené smyčky, jiné smyčky uzavřené. U určitých robotických systémů může nastat situace, kdy některá z metod není vhodná a není možné ji efektivně použít. Toto omezení vyplývá z individuálních charakteristik jednotlivých metod, které je nutné pečlivě zvážit.

My se v této kapitole podrobněji podíváme na tři metody, které budou použity v této práci a u kterých budeme rozebírat výsledky.

### 5.4.1 Metoda nejmenších čtverců

Asi nejnámější a hojně používanou metodou je metoda nejmenších čtverců, v mnoha literaturách označována LS (least squares). Tuto metodu řadíme mezi statistické metody. Principem této metody je minimalizace součtu čtverců rozdílu mezi naměřenými a predikovanými hodnotami, což nám umožňuje optimální odhad koeficientů modelu.

Tato metoda má nevýhodu v tom, že není robustní vůči šumu a odchylkám v datech.

Abychom se mohli pomocí této metody dostat na požadovaný vektor  $\beta$  je nutné mít k dispozici naměřené hodnoty momentu a sestavenou matici regresorů. Vzorec, kterým se jednotlivé prvky vektoru  $\beta$  pak vypadá následovně:

$$\beta = (Y^T Y)^{-1} Y^T u \quad (5.3)$$

kde:

- $\beta$  vektor identifikovaných koeficientů
- $Y$  je matice regresorů
- $u$  naměřený moment

Výsledný vektor koeficientů lze aplikovat na vzorec pro inverzní dynamiku 5.1 a dopočítat momenty. Po identifikaci je vždy žádoucí kontrola výsledku. V tomto případě se porovnají naměřené momenty s těmi vypočtenými. [15]



## 5.4.2 Metoda vážených nejmenších čtverců

Metoda vážených nejmenších čtverců (Weighted least squares, WLS) je metoda obdobná metodě nejmenších čtverců s tím rozdílem, že zde vstupují do výpočtu váhy, které uvádějí spolehlivost dat. Toto je velmi nápomocné v případech, kdy máme data s různou mírou přesnosti a chceme, aby byla více vážená přesnější měření než ta malá. Do vzorce je tedy přidána matice  $\mathbf{W}$ , která přiřazuje váhy jednotlivých měření. Tato matice je obvykle diagonální, při čemž se na ní nachází jednotlivé váhy. Většinou jsou váhy přímo úměrné inverzi rozptylu nebo variaci měření, a tedy měření s menším rozptylem obdrží vyšší váhu než měření s rozptylem větším. Toto je většinou zajišťováno statistickými metodami. Je zde možnost nastavovat váhy i ručně, ale tato varianta je nerealizovatelná pro měření obsahující větší množství dat. V praxi je výběr váhové matice často kompromisem mezi přesností a robustností. Vzorec pro výpočet vektoru koeficientů je pak následující:

$$\beta = (Y^T W Y)^{-1} Y^T W u \quad (5.4)$$

kde:

- $\beta$  vektor identifikovaných parametrů
- $Y$  je matice regresorů
- $W$  je diagonální váhová matice
- $u$  naměřený moment

Ve valné většině případů je jasné, že tato metoda bude vycházet přesněji než klasická metoda nejmenších čtverců. Nevýhodou může být, že při špatné volbě váhové matice budou výsledky hodně nepřesné a je tedy výhodné ověření shody těchto dvou metod. Pokud WLS vychází hůře než LS, je na místě přehodnotit váhovou matici do přesnější podoby a identifikaci udělat znovu. [15]

## 5.4.3 ML

Metoda ML (zkratka z Maximum Likelihood) je statistická metoda, jejíž princip je maximalizace pravděpodobnostního rozdělení mezi naměřenými daty a predikcí modelu. Jednodušeji řečeno, cílem této metody je nalézt hodnoty parametrů, které nejlépe vystihují naměřená data.

Pravděpodobnostní rozdělení je reprezentováno normálním rozdělením, což vede na minimalizaci kvadratického rozdílu mezi naměřenými daty a predikcí. Obecně to lze popsat vztahem:

$$\hat{\beta}_{\text{ML}} = \arg \max_{\beta} p(y | \beta) \quad (5.5)$$

kde  $p(y | \beta)$  je pravděpodobnost, že byla naměřena právě daná data  $y$  za předpokladu, že jsou správně predikována modelem s parametry  $\beta$ .



Uvažujme tedy pravděpodobnostní funkci jako:

$$p(y|\beta) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\tau_i(\beta) - y_i)^2}{2\sigma^2}\right)$$

Logaritmus této funkce je:

$$\log p(y|\beta) = -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^N (\tau_i(\beta) - y_i)^2$$

Maximalizace této pravděpodobnostní funkce se tedy rovná minimalizaci výrazu:

$$\sum_{i=1}^N (\tau_i(\beta) - y_i)^2$$

Výpočet parametrů lze pak přepsat do následujícího tvaru:

$$\hat{\beta}_{ML} = \arg \min_{\beta} \sum_i \epsilon_i^T G_i \Sigma_i^{-1} G_i^T \epsilon_i$$

kde

- $\hat{\beta}_{ML}$  je odhadnutý vektor koeficientů
- $\epsilon_i = \tau_i - y_i$  jsou chyby mezi naměřenými daty  $y_i$  a predikcemi  $\tau_i$
- $G_i$  jsou matice regresorů
- $\Sigma_i$  je kovarianční matice chyb

Někoho může mást, že ačkoliv se jedná o maximalizaci funkce, je ve finálním vzorci minimalizace funkce. To je z důvodu, že se hledá maximum odhadu při zohlednění negativního logaritmu funkce.

Tato metoda poskytuje odhady dynamických parametrů s různou přesností v závislosti na dostupných datech a modelových předpokladech. Záleží tedy velmi na kvalitě přesnosti modelu. [20] [19]

## 5.5 Vyhodnocení identifikace

Vyhodnocení identifikace je klíčovým krokem při výběru nejideálnější metody pro konkrétní aplikaci. Použitím vhodné metody lze získat nejpřesnější a nejspolehlivější data, podle kterých se dá následně navrhnout řízení. Je proto důležité mít určitou metriku pro porovnání výsledků. Touto metrikou pro nás může být například RMSE. RMSE je běžně používaná metrika pro nalezení kvantitativní míry chyby

predikce mezi naměřenými daty a daty, které dostaneme pomocí identifikace. Obecně se nechá tato metrika vyjádřit vzorcem:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (5.6)$$

kde:

- $N$  je počet pozorování
- $y_i$  je skutečná hodnota
- $\hat{y}_i$  vypočtená (predikovaná) hodnota

Nížší hodnota RMSE naznačuje lepší shodu mezi predikcemi a skutečnými daty. Obecně se tedy vyplatí aplikovat na identifikační data více metod, následně je vyhodnotit a vybrat z nich nejpřesnější data.

Pro výpočet RMSE je použita implementována funkce `rmse()` v Matlabu, která je schopna vrátit tuto hodnotu jak pro každou osu odděleně, nebo pouze jednu celkovou hodnotu pro všechny osy dohromady.

## 5.6 Benchmark for Identification of Robot Dynamics

Benchmark for Identification of Robot Dynamics, zkráceně a dále používáno jako **BIRDy** je identifikační nástroj určený pro zjednodušení procesu identifikace dynamických parametrů. Nabízí matematické podklady k realizaci více než 20 identifikačních metod. S odborným přístupem se jedná o silný nástroj, který je aplikovatelný na libovolné robotické struktury vytvořené uživatelem. Byl vytvořený Quentinem Leboutetem, je volně dostupný a podléhá licenci, která umožňuje jeho volné použití [24]. Tento benchmark vznikl pro účely článku, který je dostupný zde [15].

Sám benchmark má v sobě zabudované tři ukázkové příklady robotů se šesti stupni volnosti. Na těchto příkladech je možnost vyzkoušet implementované identifikační metody a analyzovat jejich reakci v prostředí Matlab. Ze začátku práce s ním je toto více než přínosné. Benchmark je komplexní systém obsahující rozsáhlou sadu podprogramů, které generují významné množství užitečných dat.

Pro využití tohoto benchmarku na jiné robotické systémy je zapotřebí vytvořit určité struktury. Tyto struktury benchmarku definují, s jakým systémem pracuje (stupně volnosti, délky ramen, setrvačné parametry, atd.), předávají mu data z měření a popisují trajektorii, po které robot provádí pohyb. K vytvoření těchto struktur nejlépe poslouží již vytvořené příklady, protože sám autor nikde nepopisuje, jak struktury vytvořit. Rozeberme si nyní jednotlivé struktury detailněji.

### 5.6.1 Robot description file

V tomto souboru v podstatě vytváříme popis robotického systému, obdobně jako v robotic toolboxu. Definujeme zde například hodnoty z DH tabulky, informace o kloubech, ramenech, hmotnostech a setrvačných parametrech. Důležitá hodnota je hodnota vzorkovací frekvence, která se musí shodovat s vzorkovací frekvencí, kterou máme z identifikačního měření. Jen je třeba dávat pozor, že v tomto souboru se nastavuje převrácená hodnota, tedy v mém případě se vzorkovací frekvence přepíše z  $2 * 10^{-4}$  na  $5 * 10^3$ . Dále zde jsou implementovány různé modely tření, přičemž je většina z nich ve fázi rozpracování. Já pro veškeré identifikace použil hojně používaný model tření **ViscousCoulomb**, tedy viskózní Columbovo tření. Velká výhoda spočívá v tom, že je lineární v porovnání s jinými modely tření, jak je psáno zde [7] nebo v [16].

Tento soubor příkládám k nahlédnutí jako přílohu 3.

### 5.6.2 Struktura reálného robotu

Tato struktura benchmarku říká, jak se chová reálný robot, na kterém provádíme identifikaci. Tento soubor vytvoříme z reálných dat naměřených během identifikačního experimentu. Z něj máme všechna data uložena v databázovém souboru CSV. Matlab nám umožňuje z CSV souboru udělat klasický struct a data si v něm pak upravit, což je v tomto případě potřeba. Výsledná struktura musí obsahovat informace o vzorkování, požadované trajektorii, naměřené polohy, rychlosti, zrychlení a moment. Tato data poskládáme do struktury dle předlohy z ukázkových příkladů a pojmenujeme ji **data\_planar\_3DOF\_real**. Pokud máme takto vytvořenou strukturu, vložíme ji do složek benchmarku a v hlavním programu nastavíme proměnnou **experimentOnTrueRobot** na True, což mu řekne, že má tuto strukturu po spuštění v souborech najít a použít ji.

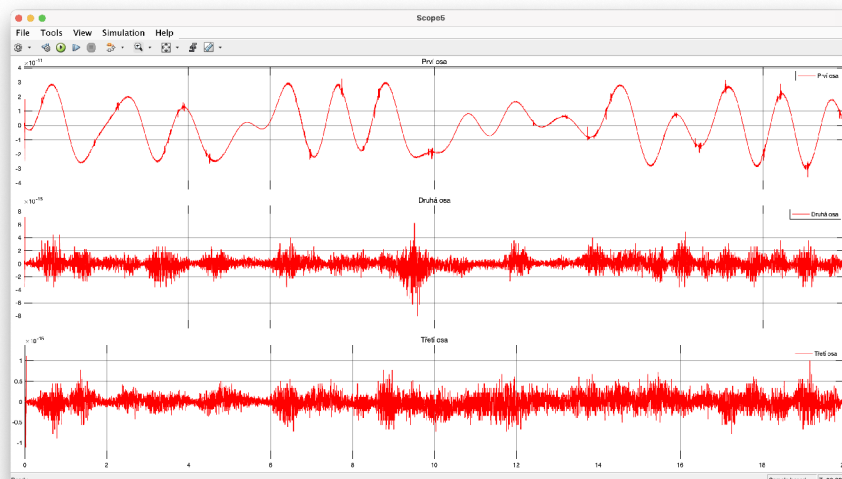
### 5.6.3 Data trajektorie

Tato struktura obsahuje pouze data o tom, jak vypadá excitační trajektorie, kterou jsme použili pro měření. Nachází se v ní tedy čtyři řádky: údaje o vzorkování, polohy, rychlosti a zrychlení jednotlivých ramen. Po vytvoření je třeba vložit do souborů benchmarku, obdobně jako strukturu reálného robotu. Je důležité dávat pozor, že po každé změně trajektorie a tedy změně této datové struktury je třeba v hlavním programu změnit proměnnou **regenerateTrajectory** na true, aby se identifikace provedla na nových datech. Po prvním spuštění je možnost tuto proměnnou znovu vrátit do false, protože je zbytečné stejnou trajektorii generovat stále dokola. Při rozsáhlejších robotických strukturách by to identifikaci zbytečně zdržovalo.

### 5.6.4 Hlavní program

Když máme všechny předchozí struktury vytvořené, přejdeme do hlavního programu benchmarku. Než přejdeme k samotné identifikaci, zkontrolujeme, zda jsme

všechny description soubory udělali správně. Jednoduše nezadáme žádnou identifikační metodu a necháme benchmark, aby nám symbolicky vygeneroval vektor koeficientů  $\beta$ . Do tohoto vektoru následně vložíme proměnné, které jsme zadávali při tvorbě analytického modelu (momenty setrvačnosti, hmotnosti, délky ramen a polohy těžišť). Když potom dosazený vektor pronásobíme maticí regresorů, měli by nám vycházet stejné hodnoty momentů, jako z analytického řešení. Lépe řečeno se stejně malou chybou.



Obr. 5.1: Rozdíl momentů spočítaných pomocí BIRDy a analytického řešení

Na obrázku výše lze vidět, že rozdíly mezi BIRDy a analytickým řešením jsou menší než  $10^{-11}$ , což nám potvrzuje, že jsme všechno do benchmarku zadali správně.

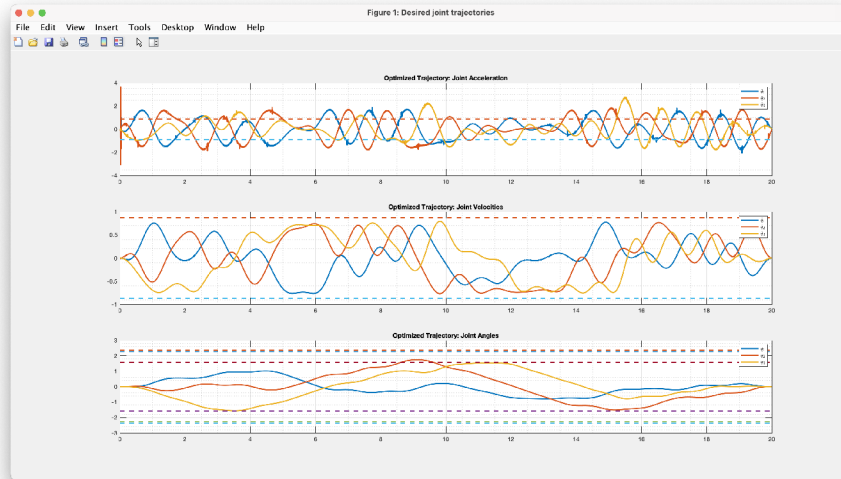
V benchmarku máme k dispozici si zobrazit námi vloženou trajektorii. Jelikož ji vkládáme jako description soubor, je velmi důležité zkontrolovat jeho správnost, abychom předešli zbytečným chybám vzniklým špatnou interpretací výsledků. Trajektorie je zobrazena na obrázku 5.2.

V tento moment máme jistotu, že všechny námi vložené soubory, které jsou potřebné pro identifikaci jsou správné a můžeme přejít k samotné identifikaci. Nastavme tedy v hlavním programu zmíněné identifikační metody, tedy OLS (klasické nejmenší čtverce), WLS (vážené nejmenší čtverce) a ML (maximum likelihood).

V momentě, kdy benchmark identifikaci dokončí, dostaneme 3 soubory s výsledky, pro každou metodu zvlášť. V těchto souborech se nachází kromě vektoru koeficientů i informace o nastavení benchmarku, aby uživatel věděl, za jakých podmínek identifikace proběhla.

## 5.7 Porovnání a vyhodnocení výsledků identifikace

Následně provedeme porovnání a vyhodnocení jednotlivých metod identifikace. Pro zajištění lepší validity výsledků je důležité provést ověření na validační trajektorii,



Obr. 5.2: Trajektorie vložená do BIRDy

což minimalizuje riziko zkreslení reality. Vykonání vyhodnocení na validační trajektorii je důležité, neboť vyhodnocení na identifikační trajektorii by mohlo poskytnout zdánlivě přesnější výsledky než ve skutečnosti jsou.

Vypočítáme si pomocí koeficientů  $\beta$  a matice regresorů  $Y$  výsledné momenty všech metod. Toto vyhodnocení zajišťuje program, který jsem přiložil k nahlédnutí v příloze 7. Vypočítané momenty v porovnání s naměřeným momentem jsou zobrazeny na obrázku 5.3.

Porovnání momentů bylo pak realizováno na validační trajektorii, viz obrázek 5.4. Z těchto hodnot je vypočtena hodnota RMSE, která je vidět na tabulce 5.1.

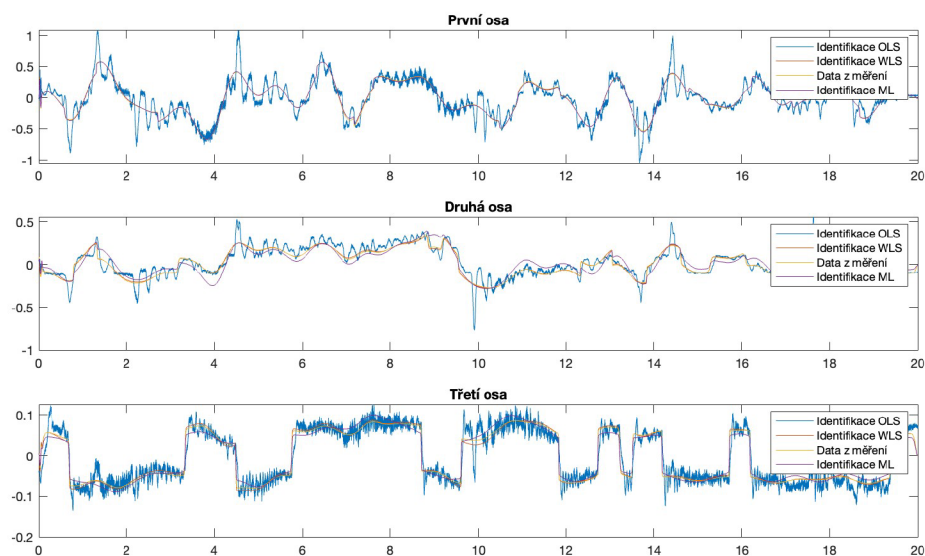
	<i>OLS</i>	<i>WLS</i>	<i>ML</i>
První osa	0.0924	0.0924	0.0924
Druhá osa	0.0538	0.0538	0.0653
Třetí osa	0.0194	0.0190	0.0203

Table 5.1: Výsledky RMSE

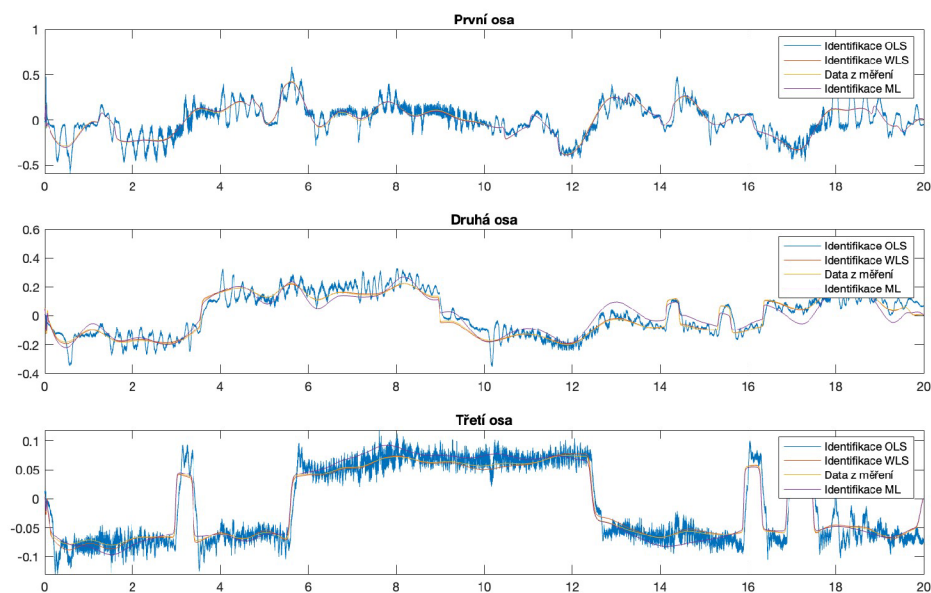
Výsledky ukazují, že metoda WLS dosáhla nejlepších výsledků s nejnižší průměrnou hodnotou RMSE pro všechny osy, následovaná metodou WLS, ML a OLS. Existuje několik faktorů, které mohou přispět k lepším výsledkům metod WLS ve srovnání s metodami OLS a ML.

Jedním je, že rozptyl chyb v datech není konstantní. Metoda WLS je schopna efektivně řešit tento problém prostřednictvím vážených odhadů, což může vést k přesnějším odhadům parametrů modelu. [24]

Důležité je také vzít v úvahu komplexnost modelu a dostupnost naměřených dat. Metody, jako je ML, mohou být citlivější na nedostatky v naměřených datech nebo na nevhodné specifikace modelu, což může vést k horším výsledkům. Naopak, metoda WLS je robustnější vůči těmto problémům. [24]



Obr. 5.3: Výsledky momenty spočítány s výsledků identifikace v porovnání s naměřeným momentem (identifikační trajektorie)



Obr. 5.4: Výsledky momenty spočítány s výsledků identifikace v porovnání s naměřeným momentem (validační trajektorie)



## 6 Řízení

Řízení robotických manipulátorů je klíčovým prvkem v oblasti robotiky, který se zabývá navrhováním algoritmů a strategií pro řízení pohybu a chování robotických manipulátorů. Základním cílem řízení robotických manipulátorů je dosažení požadovaného pohybu či stavu manipulátoru vzhledem k danému úkolu nebo okolnostem. To zahrnuje kontrolu polohy, rychlosti, zrychlení a síly působící na manipulátor. Porozumění principům a metodám řízení manipulátorů je klíčové pro úspěšnou implementaci robotických systémů a maximalizaci jejich výkonu a efektivity. V této kapitole si představíme dvě hojně používané metody.

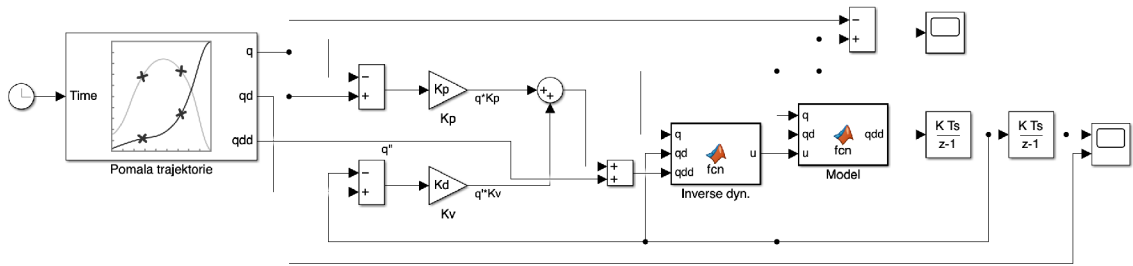
### 6.1 Momentové řízení

Tato metoda se řadí mezi metody, které pro regulaci používají inverzní dynamiky. Princip této metody spočívá v tom, že systém (v mnoha případech nelineární) je zapojen do kaskády s jeho inverzní funkcí. nutno podotknout, že inverze není nikdy dokonalá a proto tato metoda vyžaduje zpětnovazební smyčku, aby měla prostor na řešení těchto chyb.

$$\begin{aligned} u &= M(q)\{\ddot{q} + K_v(\dot{q} - \dot{q}^*) + K_p(q - q^*)\} + C(q, \dot{q})\dot{q} + G(q) \\ &= D^{-1}(q, \dot{q}, (\ddot{q} + K_v(\dot{q} - \dot{q}^*) + K_p(q - q^*))) \end{aligned} \quad (6.1)$$

Výpočet tohoto řízení je dán vztahem 6.1, kde  $K_p$  a  $K_v$  jsou zesílení regulátoru a  $D^{-1}$  označuje inverzní dynamickou funkci,  $\dot{\mathbf{q}}, \mathbf{q}$  jsou vstupní (referenční) rychlosti a polohy, a  $\dot{\mathbf{q}}^*, \mathbf{q}^*$  jsou spočítané rychlosti a polohy.

Dle rovnice 6.1 je pak sestaveno schéma, které se nachází na obrázku 6.1.

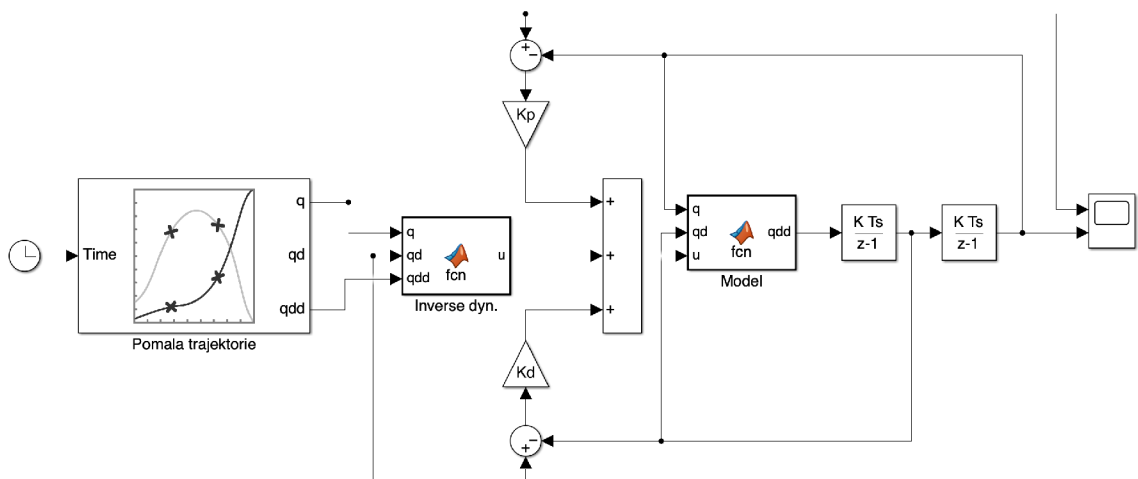


Obr. 6.1: Schéma momentového řízení

Výpočet inverze dynamiky zde probíhá v každém kroku simulace, proto je důležité mít výpočet co nejvíce zjednodušený, aby zbytečně nezatěžoval výpočetní algoritmus. Tato metoda je náchylná na odlišnosti a nepřesnosti v používaných datech. [21]

Model s řízením je k dispozici v příloze 9. Pro tvorbu trajektorie je využit blok **jtraj**, který je vytvořen v [21].

## 6.2 Dopředné řízení



Obr. 6.2: Schéma dopředného řízení

Dopředné řízení, neboli feedforward řízení je založeno na předpokladu, že většina nežádoucích chyb je známa předem a může být eliminována. Tato metoda využívá modely dynamiky manipulátoru k výpočtu ideálních akčních zásahů, a proto je nutné, aby byl model co nejvíce přesný. Výpočty akčních zásahů se vypočítávají na základě požadovaných trajektorií a dynamických vlastností manipulátoru. Regulator pro feedforward řízení je dán vztahem:

$$\begin{aligned} u &= M(q) * \ddot{q} + C(q, \dot{q}) * \dot{q} + G(q) + (K_v * (\dot{q} - \dot{q}^*) + K_p * (q - q^*)) \\ &= D^{-1}(q, \dot{q}, \ddot{q}) + K_v * (\dot{q} - \dot{q}^*) + K_p * (q - q^*) \end{aligned} \quad (6.2)$$



kde  $K_p$  a  $K_v$  jsou zesílení regulátoru a  $D^{-1}$  označuje inverzní dynamickou funkci,  $\dot{\mathbf{q}}, \mathbf{q}$  jsou vstupní (referenční) rychlosti a polohy, a  $\dot{\mathbf{q}}^*, \mathbf{q}^*$  jsou spočítané rychlosti a polohy. [21]

Dle rovnice 6.2 je sestaveno simulační schéma, které je zobrazeno na obrázku 6.2.

Lze tedy říct, že toto řízení se skládá z dopředné složky a ze zpětnovazební složky (viz obr. 6.2). Tato zapojení se kombinují pro dosažení vyšší stability a přesnosti. Zároveň zpětnovazební smyčka umožňuje korigovat chyby a odchylky od požadovaných trajektorií, které nemohou být předvídaný nebo zahrnutý do modelu dynamiky. [21]

Simulační schéma je přiloženo v příloze 10.

### 6.3 Experimentální porovnání metod řízení

V této části práce se pokusíme co nejobektivněji porovnat výše uvedené dvě metody. Za účelem objektivního porovnání navrheme 2 trajektorie, přičemž jedna bude díky své dynamice reprezentovat relativně pomalý pohyb robotu. Druhá trajektorie je navržena, tak aby dostatečně vybudila dynamiku robotu a představovala tak "náročnější" úkol pro navržené regulátory. Návrh a generování trajektorie byly realizovány s využitím nástroje **Polynomial trajectory**. Jedná se o nástroj z knihovny **Robotic System Toolbox**, který umožňuje navrhovat trajektorii definováním waypointů a následnou časovou parametrizaci. Jednotlivé waypointy jsou proloženy polynomiální funkcí, aby tvořili trajektorii. Blok umožňuje nastavení počáteční konfigurace a rychlosti jednotlivých os systému. Tato nastavení jsou v obou variantách nastaveno do nuly.

Vygenerovaná trajektorie vypadá následujícím způsobem:

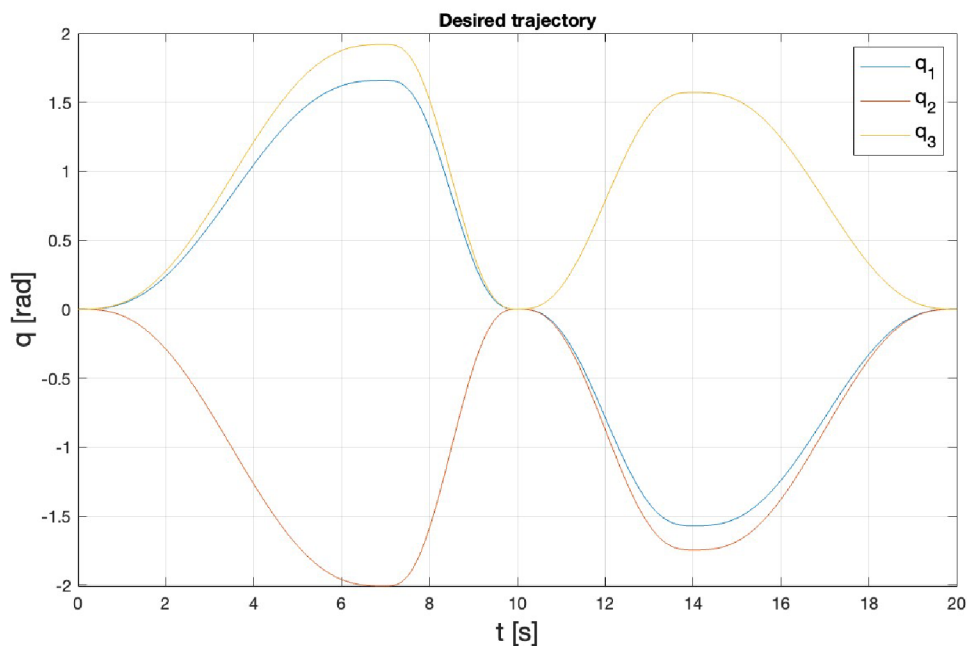
Jelikož chceme využít výsledky identifikační metody pro účely simulace řízení, je potřeba upravit dynamickou rovnici

$$u = M(q)\ddot{q} + C(q, \dot{q})\dot{q}$$

na

$$u = \hat{M}(q)\ddot{q} + \hat{C}(q, \dot{q})\dot{q}$$

kde matice  $\hat{M}$  a  $\hat{C}$  jsou sestaveny z estimovaných dynamických parametrů získaných estimací váženou metodou nejmenších čtverců. Tento přepis je použit pro výpočet přímé i inverzní dynamiky, respektive v Matlab funkcích použitých v simulacích řízení. Nyní lze vytvořit hmotnostní matici a matici Coriolisových sil v závislosti na výsledcích identifikace. Pro všechny následující experimenty využijí výsledky metody WLS.



Obr. 6.3: Trajektorie vygenerovaná blokem Polynomial trajectory

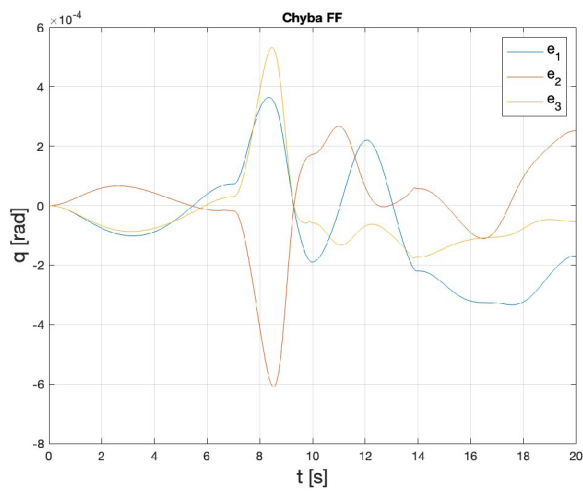
Maximální rychlost trajektorie [stupeň/s]		
Osa	Pomalá	Rychlá
1	59.3750	89.0625
2	71.8750	107.8125
3	68.7500	103.1250

Obr. 6.4: Maximální rychlosti u trajektorií

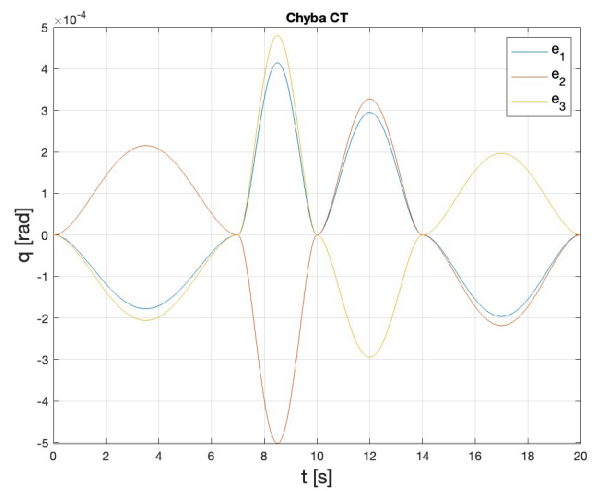
### 6.3.1 Simulační porovnání bez regulátorů

Tento experiment zkoumá chybu, která vzniká v důsledku výpočtu inverzní a následně přímé dynamiky. Pro obě trajektorie jsou následně vypočítány maximální chyby sledování trajektorie na jednotlivých osách.

Z grafů 6.5 a analýzy tabulky 6.7 vyplývá, že při absenci akčního zásahu vykazuje momentové řízení menší chyby. Při pomalé trajektorii je pozorována výrazně menší odchylka než při rychlejší trajektorii, a to bez ohledu na použitou metodu. Nicméně, metoda feedforward projevuje vyšší náchylnost k chybám při rychlých změnách trajektorie, což se projevuje ve zvýšených odchylkách.

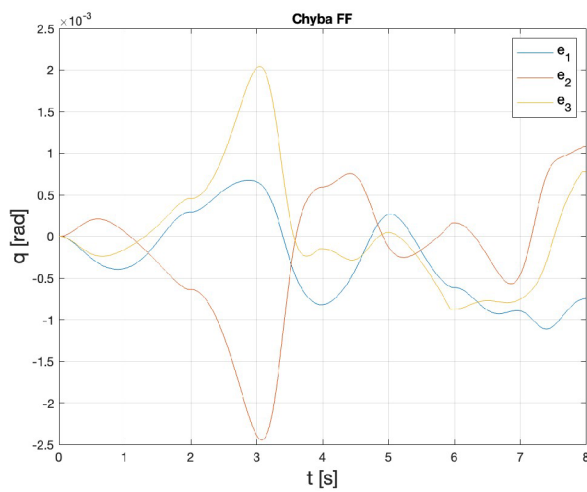


(a) Feedforward řízení

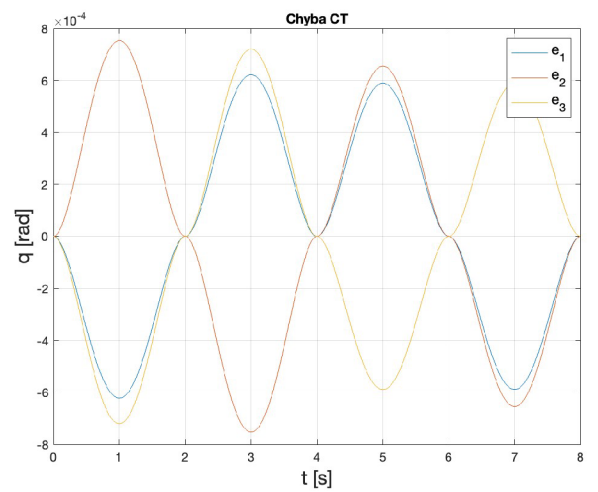


(b) Momentové řízení

Obr. 6.5: Odchyly od požadované trajektorie při pomalém pohybu



(a) Feedforward řízení



(b) Momentové řízení

Obr. 6.6: Odchyly od požadované trajektorie při rychlém pohybu

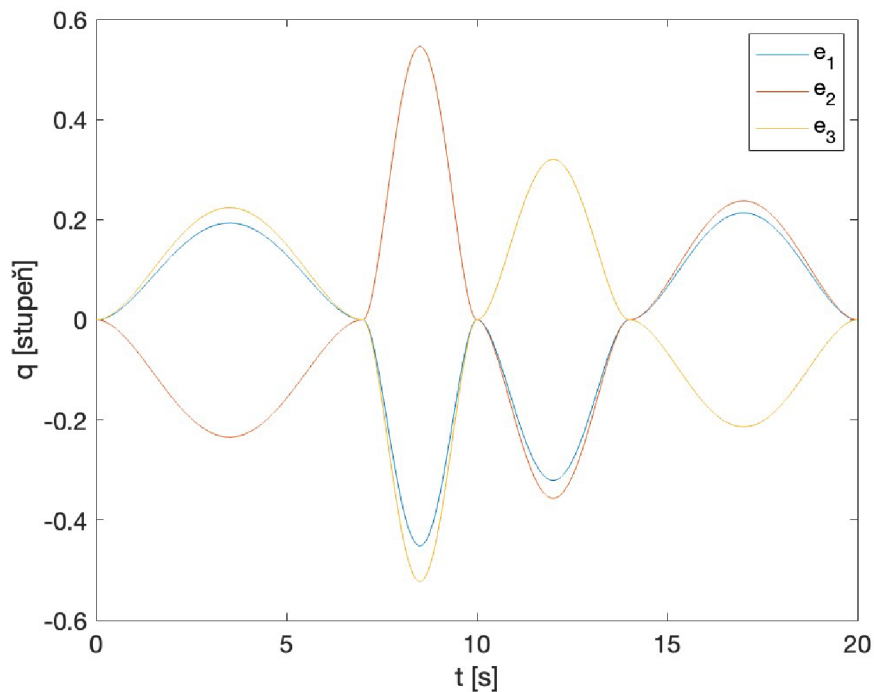
Maximální chyba sledování trajektorie [rad]						
Bez regulátorů	Computed torque			Feedforward		
	Osa	Pomalá trajektorie	Rychlá trajektorie	Osa	Pomalá trajektorie	Rychlá trajektorie
	1	4.1456e-04	6.2194e-04	1	3.6386e-04	0.0011
2	5.0184e-04	7.5287e-04	2	6.0947e-04	0.0024	
3	4.8002e-04	7.2014e-04	3	5.3159e-04	0.0020	

Obr. 6.7: Chyby na jednotlivých osách (bez regulátorů)

### 6.3.2 Polohové řízení na reálném systému

Jako první experiment na reálném systému bude provedeno polohové řízení. Pro realizaci řízení na reálný robotický systém je použito otevřené řídicí prostředí s automatickým generováním kódu, který převádí Matlab/Simulink simulační modely na kód v jazyce C/C++ používaný v systému PLC. Více o tomto prostředí se lze dočíst v [14].

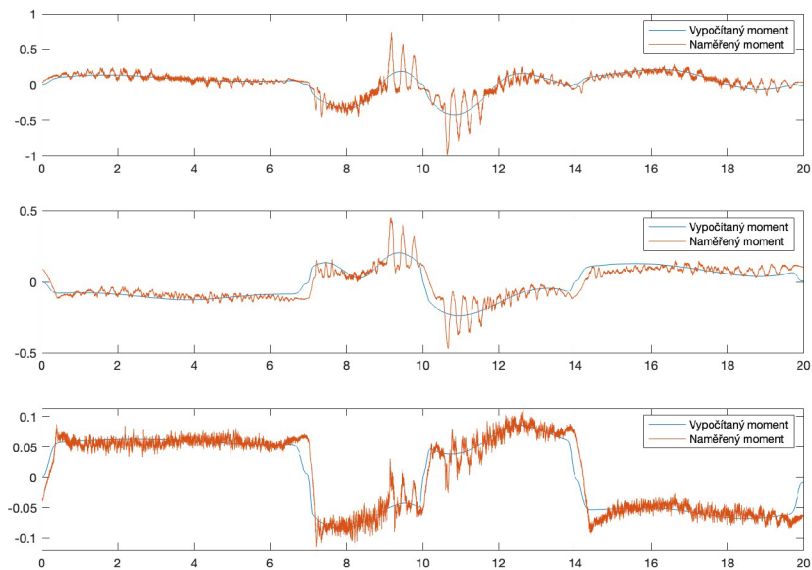
Na systém aplikujeme pomalou trajektorii, zmíněnou v 6.3, za pomoci automatického generování kódu, zmíněného výše. Porovnání naměřených momentů z tohoto experimentu a vypočtených momentů, které jsou spočítány z naměřené trajektorie nám poskytnou validaci lineárního přepisu. Experimentální měření je vzorkováno frekvencí 2 ms. Pro nastavení regulátorů v systému je využita kaskádní regulace od výrobce, kde jsou jednotlivé regulátory naladěny autotuningem. Jako první nejprve porovnáme průběhy regulačních odchylek.



Obr. 6.8: Průběh regulačních odchylek z polohového řízení

Průběhy z obrázku 6.8 nám dávají přehled o kvalitě regulace. Z toho můžeme usoudit, že tato metoda poskytuje přijatelné výsledky a můžeme porovnat průběhy momentů.

Z porovnání momentů z obrázku 6.9 můžeme vyhodnotit, že lineární přepis vzhledem k vektoru koeficientů je dostatečně přesný a aplikovatelný na experimenty na reálném systému.



Obr. 6.9: Porovnání naměřeného momentu a vypočítaného momentu po lineárním přepisu

### 6.3.3 Dopředné řízení na reálném systému

Jako první v těchto experimentech byla provedena sada měření, na základě kterých se zvolila nejlepší kombinace zesílení pro regulátory. Regulátory byly voleny jako optimální kompromis mezi kmitáním jednotlivých os, šumu v rychlostech a minimální odchylky měřených poloh vůči polohám požadovaným. Ty jsou tedy zvoleny:

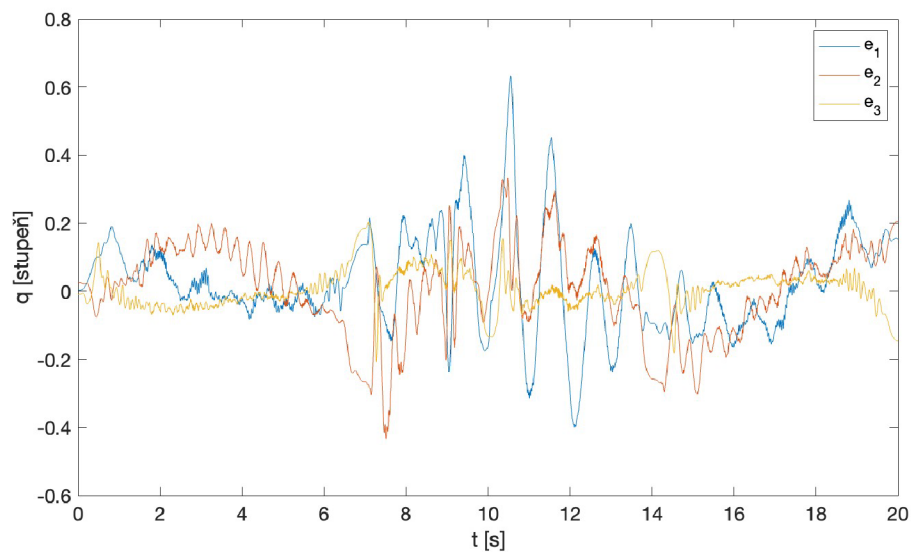
$$K_p = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 15 & 0 \\ 0 & 0 & 15 \end{bmatrix} \quad K_d = \begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$$

Pro realizaci řízení na reálném systému je znovu použito otevřené řídicí prostředí a automatické generování kódu [14]. Vzorkujeme frekvencí  $400 \mu\text{s}$ . Průběh regulační odchylky na všech osách je zobrazen na obrázku 6.10.

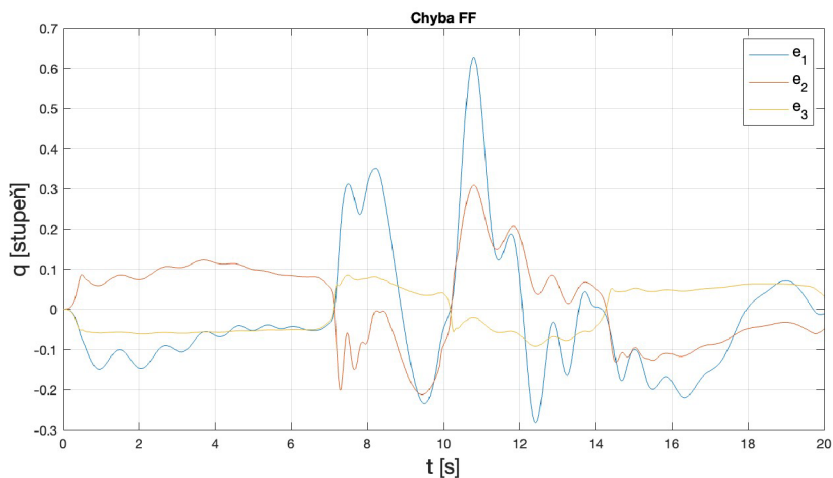
Jelikož se na reálném systému projevují mechanické jevy, jako jsou například vlivy převodovek motorů, nelze nikdy dosáhnout tak přesné regulace, jako to může být u simulací. Nicméně výsledná chyba je v přijatelných mezích. Pokusme se nyní spočítat průměrnou chybu momentu (vypočítaný moment - naměřený moment), kterou následně využijeme do simulačního modelu. Postup pro její výpočet je rozepsán zde [9].

Tuto chybu aplikujeme na všechny koeficienty a provedeme simulaci. Do simulace zanášíme i stejné nastavení regulátorů. Na obrázku 6.11 se nachází průběh regulační odchylky po aplikaci chyby na koeficienty.

Z vizuálního porovnání je možné vysledovat, že nyní máme model dopředného řízení, který mnohem více odpovídá reálnému systému.



Obr. 6.10: Průběh regulačních odchylek



Obr. 6.11: Průběhy regulačních odchylek s chybou na koeficientech, získány pomocí simulace

### 6.3.4 Momentové řízení na reálném systému

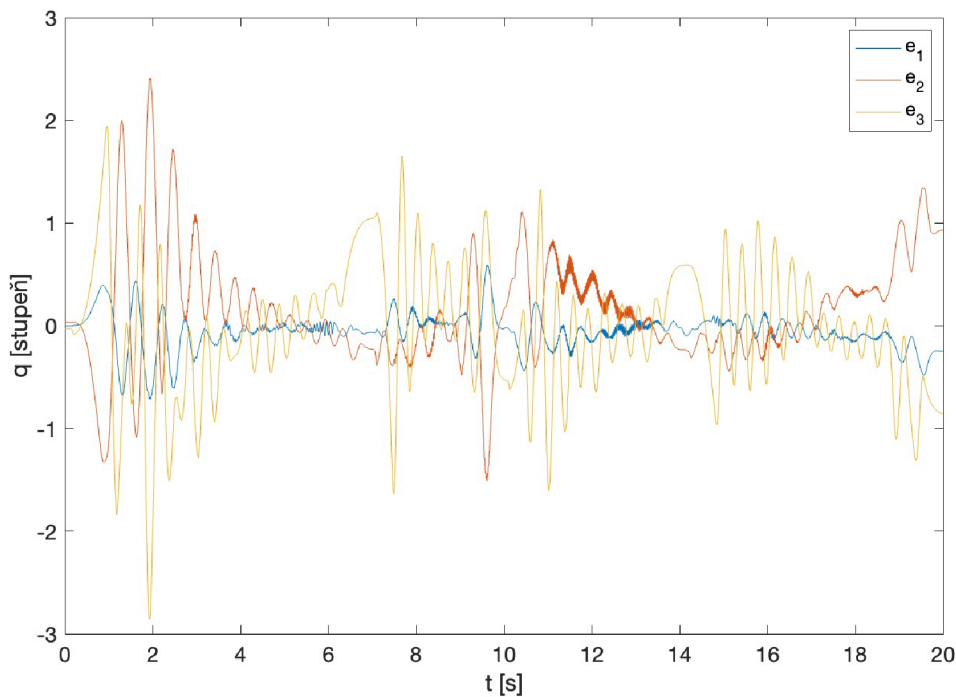
Nyní obdobně jako u dopředného řízení realizujeme computed torque řízení na reálný systém robotu. Nalezení optimálních zesílení regulátorů bylo realizováno experimentálně. Pro tuto metodu vypadají následovně:

$$K_p = \begin{bmatrix} 250 & 0 & 0 \\ 0 & 250 & 0 \\ 0 & 0 & 300 \end{bmatrix} \quad K_d = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

Experiment byl vzorkován frekvencí 2 ms. Průběh regulačních odchylek je zobrazen na obrázku 6.12. Oproti feedforward řízení je u této metody daleko větší šum na naměřených rychlostech a nachází se tedy ve velké míře i na zrychleních. To ve velké míře ubírá na kvalitě měření a tedy hlavně výsledků.

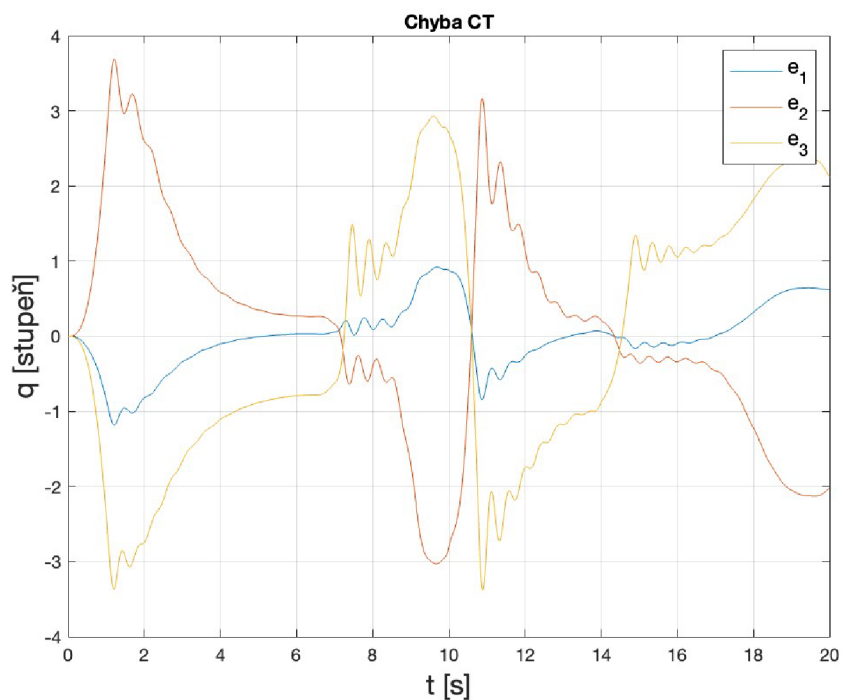
V návaznosti na zpřesnění simulačního modelu jsem se pokusil vyjádřit průměrnou chybu z naměřených dat dle postupu popsaného zde [9]. Tuto chybu jsem následně zavedl na všechny koeficienty. Průběh regulační odchylky po zavedení těchto chyb je zobrazen na obrázku 6.13.

Po zavedení této chyby je vidět, že se simulační model mnohem více přiblížil reálnému systému. Nicméně je na několika místech, že je regulační odchylka vyšší, než která byla změřena. Nejspíš by byla zapotřebí další analýza pro zpřesnění modelu, ale i tak teď lépe odpovídá reálnému robotu, na kterém měření probíhala.



Obr. 6.12: Průběh regulačních odchylek





Obr. 6.13: Průběh regulačních odchylek s chybou na koeficientech, získány pomocí simulace

### 6.3.5 Zhodnocení

Řízení bylo ověřeno jak simulačně tak na reálném systému. Zároveň bylo provedeno zavedení nepřesnosti do modelu, které jak je z výsledku simulace patrné, lépe charakterizuje chování reálného systému. Takto upravené modely lze využít pro další potřeby.

Metoda computed torque prokázala mnohem větší náchylnost na šum v datech. Ty by se dali eliminovat přidáním robustní funkce do již vytvořeného modelu, která by kompenzovala šum, který vzniká u experimentu.

Oproti dopřednému řízení je momentová metoda mnohem více náchylná na nepřesnosti v modelu. To by také mohla vyřešit přidaná robustní funkce, která by pomohla regulátoru.



## 7 Závěr

Jako první v této práci jsme si představili technologii 3D tisku ve stavebnictví pomocí robotických manipulátorů. Tato kapitola je velmi důležitá pro pochopení motivaci pro tvorbu a vývoj těchto robotických systémů.

Následně je detailně probrána kinematika, tedy analýza robotu za pomoci Denavit-Hartenbergovy konvence, a dynamika. Zde je rozebrána problematika ohledně maticového odvození rovnic dynamiky. Za pomoci informací získaných v této kapitole je zhotoven přesný analytický popis planárního robotu se třemi stupni volnosti. Ten je následně porovnán s bloky Robotic System Toolboxu, které jsou dostupné v prostředí Matlab Simulink. Toto srovnání, jak jsem již zmínil dříve, je velmi důležitý pro kontrolu správnosti vytvořeného modelu. Bez správného modelu se nelze posunout dále k problematice identifikace a řízení. Chyba zde vychází ve velmi malých rádech, tudíž je model dostatečně přesný pro pokračování v práci.

Navazující část této práce se věnuje samotné identifikaci dynamických parametrů. Na tuto proceduru je používán Benchmark (BIRDy). Zde bylo nejprve důležité pochopit jeho funkčnost, protože je velmi rozsáhlý a nabízí spoustu možností. Po seznámení s problematikou byl zhotoven model planárního robotu. Na vytvořené trajektorii byly naměřeny všechna potřebná data, jako jsou například jednotlivé polohy, rychlosti a momenty na všech osách. Následná identifikace setrvačných parametrů byla provedena s využitím nástroje BIRDy. Za pomoci něho bylo navíc získáno symbolické vyjádření jednotlivých matic, ze kterých je tvořena dynamická rovnice. Z těchto dat byly spočítány výsledné momenty na všech osách. Všechny výsledky jsou porovnány s naměřenými momenty a je výpočtem RMSE vyjádřena jejich přesnost. Zde je nutné podotknout, že benchmark při každé identifikaci může poskytnout jinou variaci vektoru koeficientů a tedy i výsledných analytických matic. Bylo tedy nutné před každým ověřením výsledků zkontrolovat, zda je vše aktuální.

V poslední sekci této práce jsou za pomoci maticového vyjádření z BIRDy a vektoru koeficientů zhotoveny dvě metody řízení, tedy momentové a dopředné řízení. Obě tyto metody jsou založeny na přesné znalosti dynamického modelu a dávají uspokojivé výsledky. Pro validaci využití identifikovaných parametrů pro účely řízení jsou obě metody experimentálně měřeny na reálném systému, kde jsou nalezeny optimální zesílení pro regulátory. Na výsledcích experimentů jsou následně zpřesněny simulační modely tak, aby odpovídaly reálnému systému a mohly být použity pro další experimenty.

Hlavním cílem této práce byla identifikace dynamických parametrů, který byl splněn. Zároveň bylo provedeno otestování použití estimovaných parametrů pro účely řízení založených na modelu. Testování dalších přístupů řízení a důkladná

analýza byla bohužel nad časový rámec této práce, ale výsledky mohou sloužit jako základ pro navazující práci.

## Seznam literatury:

- [1] Nasa plánuje tisknout domy pomocí robotů, první by mohl stát na měsíci do roku 2030, . URL <https://zahranicni.hn.cz/c1-66219430-nasa-planuje-tisknout-domy-pomoci-robotu-prvni-by-mohl-stat-na-m>
- [2] 3d star: 3d tisk ve stavebnictví a architektuře, . URL <https://publi.cz/download/publication/2590?online=1>.
- [3] Industrial robot cost declines should trigger tipping points in demand. URL <https://ark-invest.com/articles/analyst-research/industrial-robot-cost-declines/>.
- [4] Generate polynomial trajectories through waypoints - simulink. [https://www.mathworks.com/help/robotics/ref/polynomialtrajectory.html?s\\_tid=doc\\_ta](https://www.mathworks.com/help/robotics/ref/polynomialtrajectory.html?s_tid=doc_ta).
- [5] World robotics 2023 report: Asia ahead of europe and the americas, . URL <https://ifr.org/ifr-press-releases/news/world-robotics-2023-report-asia-ahead-of-europe-and-the-americas>.
- [6] World robotics report: “all-time high” with half a million robots installed in one year, . URL <https://ifr.org/ifr-press-releases/news/wr-report-all-time-high-with-half-a-million-robots-installed>.
- [7] Chapter 8. models of friction. URL [https://www.mogi.bme.hu/TAMOP/robot\\_applications/ch07.html#ch-8.3](https://www.mogi.bme.hu/TAMOP/robot_applications/ch07.html#ch-8.3).
- [8] Místo cihel a zedníků nastupují 3d tiskárny a nové materiály. URL <https://www.materialtimes.com/tema/misto-cihel-a-zedniku-nastupuji-3d-tiskarny-a-nove-materialy.html>.
- [9] Mean absolute difference. URL [https://en.wikipedia.org/wiki/Mean\\_absolute\\_difference](https://en.wikipedia.org/wiki/Mean_absolute_difference).
- [10] Linear parametrization and identification of robot dynamics. URL [http://www.diag.uniroma1.it/deluca/rob2\\_en/05b\\_LinearParametrizationIdentification.pdf](http://www.diag.uniroma1.it/deluca/rob2_en/05b_LinearParametrizationIdentification.pdf).

- [11] J.J. Craig. *Introduction to Robotics: Mechanics & Control*. Introduction to robotics : Mechanics & control / John J ... Craig. Addison-Wesley Publishing Company, 1986. ISBN 9780201103267. URL <https://books.google.cz/books?id=uDNSAAAAMAAJ>.
- [12] Jacques Denavit and Richard S. Hartenberg. Standard denavit-hartenberg convention, 1964. URL [https://www.researchgate.net/figure/Standard-Denavit-Hartenberg-convention-22\\_fig4\\_326106790](https://www.researchgate.net/figure/Standard-Denavit-Hartenberg-convention-22_fig4_326106790). Zobrazeno z: ResearchGate.
- [13] Sebastian Echeandia and Patrick Wensing. Numerical methods to compute the coriolis matrix and christoffel symbols for rigid-body systems. 10 2020.
- [14] Daniel Kajzr, Tomas Myslivec, and Josef Cernohorsky. An open plc-based robot control system for 3d concrete printing. *Robotics*, 12(4), 2023. ISSN 2218-6581. doi: 10.3390/robotics12040096. URL <https://www.mdpi.com/2218-6581/12/4/96>.
- [15] Quentin Leboutet, Julien Roux, Alexandre Janot, Julio Rogelio Guadarrama-Olvera, and Gordon Cheng. Inertial parameter identification in robotics: A survey. *Applied Sciences*, 11(9):4303, May 2021. doi: 10.3390/app11094303.
- [16] Ananya Mathu and Syeda Roquiya Badr. Dynamic characteristic of combined viscous and coulomb damping under varying nature of material and coefficient of friction. *Journal of Physics: Conference Series*, 1706(1):012179, dec 2020. doi: 10.1088/1742-6596/1706/1/012179. URL <https://dx.doi.org/10.1088/1742-6596/1706/1/012179>.
- [17] MathWorks. Simplify expressions - matlab simulink, 2023. URL <https://www.mathworks.com/help/releases/R2023a/symbolic/simplify.html>. [Online; accessed 22-February-2024].
- [18] MathWorks. Robotics system toolbox - matlab simulink, Bez data. URL [https://uk.mathworks.com/products/robotics.html?s\\_tid=A0\\_PR\\_info](https://uk.mathworks.com/products/robotics.html?s_tid=A0_PR_info). [Online; přístup dne 22. únor 2024].
- [19] M.M. Olsen and H.G. Petersen. A new method for estimating parameters of a dynamic robot model. *IEEE Transactions on Robotics and Automation*, 17(1): 95–100, 2001. doi: 10.1109/70.917088.
- [20] Swevers J. Verdonck W. Olsen, M. M. Maximum likelihood identification of a dynamic robot model: Implementation issues. *The International Journal of Robotics Research*, 21(2), 2002.
- [21] Corke Peter. *Robotics, vision and control fundamental algorithms in MATLAB*. Springer International Publishing, 2017.
- [22] T. Popule. *Řízení Pohybu víceosých robotických manipulátorů*. DSpace at University of West Bohemia, 2013.

- [23] Jan Swevers, Walter Verdonck, and Joris De Schutter. Dynamic model identification for industrial robots. *Control Systems, IEEE*, 27:58 – 71, 11 2007. doi: 10.1109/MCS.2007.904659.
- [24] Tum-Ics. Tum-ics/birdy: Benchmark for identification of robot dynamics. URL <https://github.com/TUM-ICS/BIRDy?tab=readme-ov-file>.
- [25] Nd Vuong and Marcelo Jr. Dynamic model identification for industrial robots. *Acta Polytechnica Hungarica*, 6, 12 2009.
- [26] Wenxiang Wu, Shiqiang Zhu, Xuanyin Wang, and Huashan Liu. Closed-loop dynamic parameter identification of robot manipulators using modified fourier series. *International Journal of Advanced Robotic Systems*, 9(1):29, Jan 2012. doi: 10.5772/45818.
- [27] V. Záda. *Robotika: Matematické Aspekty analýzy a řízení*. Technická univerzita v Liberci, 2012.

## Seznam příloh

- Příloha 1: Zhotovený program pro výpočet analytického řešení, vytvoření modelu v Robotics System Toolbox a obsluha BIRDy
- Příloha 2: Simulační model s porovnáním analytického, toolbox a BIRDy řešení + porovnání rozdílů těchto metod
- Příloha 3: Model planárního robotu do BIRDy (robot description file)
- Příloha 4: Struktura reálného robotu pro BIRDy
- Příloha 5: Data o reálné trajektorii pro BIRDy
- Příloha 6: Program pro vyhodnocení výsledků identifikace
- Příloha 7: Funkce pro vyčtení vektoru koeficientů
- Příloha 8: Model momentového řízení
- Příloha 9: Model feedforward řízení
- Příloha 10: Data naměřená z momentového řízení
- Příloha 11: Data naměřená z dopředného řízení