

České Vysoké Učení Technické v Praze  
Fakulta strojní  
Ústav mechaniky, biomechaniky a mechatroniky  
Obor: Mechatronika



# Návrh a zprovoznění nového řídícího systému průmyslového robota Puma 200

DIPLOMOVÁ PRÁCE

Vypracoval: Michael Valášek  
Vedoucí práce: Ing. Martin Nečas, MSc., Ph.D.  
Rok: 2018

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Valášek** Jméno: **Michael** Osobní číslo: **397330**  
Fakulta/ústav: **Fakulta strojni**  
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**  
Studijní program: **Strojní inženýrství**  
Studijní obor: **Mechatronika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Návrh a zprovoznění nového řídicího systému průmyslového robota Puma 200**

Název diplomové práce anglicky:

**Design and commissioning of new control system for industrial robot Puma 200**

Pokyny pro vypracování:

1. Seznamte se s problematikou řízení průmyslových robotů.
2. Navrhněte a implementujte nový řídicí systém pro robota Puma 200 v prostředí Matlab/Simulink.
3. Vytvořte kinematický model robota včetně virtuálního 3D modelu.
4. Proveďte kontrolu přesnosti polohování robota pomocí laser trackeru.
5. Kriticky zhodnoťte dosažené výsledky.

Seznam doporučené literatury:

- [1] Stejskal, V., Valášek M.. Kinematics and dynamics of machinery. New York: M. Dekker, 1996. Mechanical engineering (Marcel Dekker, Inc.), 106. ISBN 0824797310.  
[2] Firemní literatura: Texas Instruments pro procesory TMS320F28379D, drv8711, sprui77a  
[3] Speed Estimation and control of a brushless DC motor (BLDC) using the ADMC401 Encoder Interface Unit (EIU), AN401-19

Jméno a pracoviště vedoucí(ho) diplomové práce:

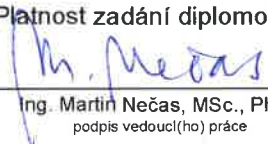
**Ing. Martin Nečas, MSc., Ph.D., odbor mechaniky a mechatroniky FS**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **24.04.2018**

Termín odevzdání diplomové práce: **17.08.2018**

Platnost zadání diplomové práce:

  
Ing. Martin Nečas, MSc., Ph.D.  
podpis vedoucí(ho) práce


  
prof. Ing. Milan Růžička, CSc.  
podpis vedoucí(ho) ústavu/katedry

  
prof. Ing. Michael Valášek, DrSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.


  
Datum převzetí zadání

  
Podpis studenta

## Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze dne 4. června 2018

  
.....  
Michael Valášek

## **Poděkování**

Děkuji Ing. Martinu Nečasovi, MSc., Ph.D. za vedení mé diplomové práce, za laskavé rady, podnětné návrhy a veškerý mi věnovaný čas. Také bych chtěl nesmírně poděkovat svým rodičům za bezednou trpělivost a veškerou podporu při mém studiu, na kterou nikdy nezapomenu. V neposlední řadě velké díky kolegovi Ondrovi Pištorovi, za pomoc při sestavování robota. Dále také děkuji Ing. Jiřímu Volechovi za pomoc při měření s Laser Trackerem.

Michael Valášek



*Název práce:*

**Návrh a zprovoznění nového řídicího systému průmyslového robota Puma 200**

*Autor:* Michael Valášek

*Druh práce:* Diplomová práce

*Vedoucí práce:* Ing. Martin Nečas, MSc., Ph.D.

*Abstrakt:* Tato práce se zabývá zprovozněním šestiosého robota PUMA 200. Jde o prakticky zaměřenou práci s cílem ilustrovat znovupoužití vyřazeného robota a vývoj nového řídicího systému, který pak může být aplikovatelný na libovolný robot. Nejprve je vysvětlena problematika moderních robotů a základ řízení DC motorů. Následně je ukázáno použití konkrétních softwarových prvků v programu MATLAB/Simulink, který byl pro tento úkol zvolen. Výsledkem je konkrétní schéma pro řízení robota vytvořené na platformě C2000 od TEXAS Instruments. Aby byl robot přesně říditelný v prostoru, bylo vypracováno řešení dopředné a inverzní kinematiky. Poslání poloh natočení motorů je následně realizováno PC přes CAN komunikaci. Na závěr je funkční robot testován na polohovou přesnost a opakovatelnost pomocí Laser trackeru a jsou zjištěny jeho vlastnosti.

*Klíčová slova:* Šestiosý robot PUMA 200, TI C2000, Inverzní kinematika, Pozorovatel, Kaskádní řízení, Laser tracker, Přesnost, Opakovatelnost.

*Title:*

**Design and commissioning of new control system for industrial robot Puma 200**

*Author:* Michael Valášek

*Druh práce:* Master's thesis

*Supervisor:* Ing. Martin Nečas, MSc., Ph.D.

*Abstract:* This diploma thesis deals with commissioning of 6 DOF robot PUMA 200. This task is practical oriented with focus on refurbishing discarded robot and on developing a new control system which can be then apply on any robot. At first it is introduced a problems of modern robots and a basic concept of DC motor drive control. Further this thesis contains the description of implementation concrete software elements in program MATLAB/Simulink, which was chosen for this task. The result is concrete scheme for robot control system based on C2000 platform from TEXAS Instruments. For precise control of the robot in 3D coordinates it has been derived forward and inverse kinematic solution. Position information of the motors is then realized by PC through CAN communication protocol. At the end the robot is tested on position precision and repeatability by Laser Tracker and robots properties are evaluated.

*Key words:* 6 DOF robot PUMA 200, TI C2000, Inverse kinematics, Observer, Cascade control, Laser tracker, Precision, Repeatability.

# Obsah

Úvod	1
<b>1 Průmyslové roboty</b>	<b>3</b>
1.1 Šestiosé průmyslové roboty . . . . .	3
1.2 Robot PUMA 200 UNIMATE . . . . .	5
1.3 Servořízení DC motorů . . . . .	7
1.3.1 Fyzikální popis motoru . . . . .	7
1.3.2 Kaskádní řízení . . . . .	9
<b>2 Implementace Hardware a Software</b>	<b>11</b>
2.1 Původní řídicí systém . . . . .	11
2.2 Nový řídicí systém . . . . .	11
2.3 Výběr mikroprocesoru . . . . .	12
2.4 Výběr výkonového modulu . . . . .	14
2.4.1 H-můstek (H-Bridge) . . . . .	15
2.4.2 Zdroj napětí . . . . .	15
2.5 Hardware podpora pro procesory <i>C2000</i> v Simulinku . . . . .	16
2.5.1 SPI komunikace . . . . .	16
2.5.2 Modul ePWM . . . . .	20
2.5.3 Linearizace motorů . . . . .	21
2.5.4 Modul eQEP . . . . .	22
2.6 Výpočet rychlosti . . . . .	23
2.6.1 Zpracování signálu pomocí regresní funkce . . . . .	24
2.6.2 Výpočet pomocí <i>Pozorovatele (Observer)</i> . . . . .	24
2.7 Modul eCAN . . . . .	26

2.8	Zapojení řídicí elektroniky . . . . .	27
2.8.1	Softwarová chyba v Simulinku . . . . .	28
2.8.2	Zapojení všech komponent . . . . .	29
<b>3</b>	<b>Řízení v Simulinku</b>	<b>31</b>
3.1	Kaskádní regulace na MCU1 a MCU2 - přijímač . . . . .	31
3.2	PC - vysílač . . . . .	33
3.3	Virtuální model robota PUMA 200 . . . . .	34
3.4	Naladění PID regulátorů . . . . .	35
<b>4</b>	<b>Kinematika</b>	<b>37</b>
4.1	Popis robota dopřednou kinematikou . . . . .	37
4.2	Inverzní Kinematika . . . . .	38
4.2.1	Popis pohybu robota . . . . .	38
4.2.2	Způsoby řešení inverzní kinematiky . . . . .	39
4.2.3	Analytické řešení inverzní kinematiky robota PUMA . . . . .	40
<b>5</b>	<b>Identifikace robota PUMA 200</b>	<b>45</b>
5.1	Identifikace motorů . . . . .	45
5.2	Identifikace převodových poměrů . . . . .	46
5.2.1	Sférické zápěstí . . . . .	47
5.3	Identifikace konektoru . . . . .	48
5.4	Identifikace konstant robota . . . . .	49
<b>6</b>	<b>Kontrola přesnosti polohování robota pomocí laser trackeru</b>	<b>51</b>
6.1	Experimenty pro ověření přesnosti a opakovatelnosti robota . . . . .	51
6.1.1	Laser tracker . . . . .	52
6.1.2	Měření přesnosti a opakovatelnosti - krychle . . . . .	52
6.1.3	Souřadnicový systém robota . . . . .	53
6.2	Opakovatelnost . . . . .	53
6.3	Přesnost polohování . . . . .	53
6.4	Koordinovaný pohyb . . . . .	55
6.5	Postup kalibrace robota . . . . .	55
6.5.1	Kružnice . . . . .	55

<b>7</b>	<b>Výsledky práce</b>	<b>59</b>
	<b>Závěr</b>	<b>60</b>
	<b>Seznam použitých zdrojů</b>	<b>61</b>
	<b>Příloha</b>	<b>63</b>

# Úvod

V dnešním světě jsou kladeny neustále větší nároky na nižší cenu a s tím související produktivitu práce. Proto také prakticky každá úspěšná velkosériová průmyslová firma vykazuje vysoký stupeň automatizace. K té patří v první linii jednoúčelové automaty a průmyslové roboty. Pochopení a zvládnutí těchto technologií má velký potenciál ve stále se rozšiřujícím odvětví masové automatizace.

V dnešní době je situace na trhu pro začínající či nekomerční subjekt složitá. Pro vytvoření kompletního průmyslového robota je zapotřebí správný design hardware součástí a software algoritmů. Hardware součásti jsou již většinou přístupné a konkurence se liší jen v detailech. Řídící algoritmy vyvíjí na světě pouze několik firem, které mají své know-how pečlivě chráněné. Tím je myšlena především oblast nízkourovňového software pro řízení servomotorů. Zde je možno pořídit hotová řešení od různých firem, ale je možno do řízení motorů nahlédnout jen do té míry, do jaké je to výrobcem umožněno.

Pro vlastní vývoj je tedy velmi vhodné si toto know-how osvojit, neboť umožňuje hlubší náhled do struktury řízení a umožňuje flexibilně upravit řídicí algoritmy dle aktuálních potřeb. Zvládnutí vlastního řízení servomotorů je tudíž velmi žádané. To je zvláště důležité pro vývoj nových řídicích algoritmů a nových robotických operací na univerzitě, tedy FS ČVUT v Praze.

## Motivace

Motivací této práce je vývoj vlastního řídicího systému pro servomotory šestiosého robota, který je nejběžnějším automatizačním prvkem v dnešní době. Pro správnou funkci musí být vybrán adekvátní výpočetní procesor, výkonové prvky a především vytvořen řídicí program, který bude celý systém ovládat. Tento program byl vyvinut v software MATLAB/Simulink, který je vhodný zejména díky své podpoře externího hardware. Tento vyvinutý řídicí systém byl aplikován na průmyslový šestiosý robot PUMA 200 UNIMATE.

Byl vybrán procesor od firmy Texas Instruments, která je plně podporován v software MATLAB/Simulink.

# Cíle práce

Vlastní práce se zabývá naplněním následujících cílů:

1. Seznamte se s problematikou řízení průmyslových robotů.
2. Navrhněte a implementujte nový řídicí systém pro robota Puma 200 v prostředí Matlab/Simulink.
3. Vytvořte kinematický model robota včetně virtuálního 3D modelu.
4. Proveďte kontrolu přesnosti polohování robota pomocí laser trackeru.
5. Kriticky zhodnoťte dosažené výsledky.

# Kapitola 1

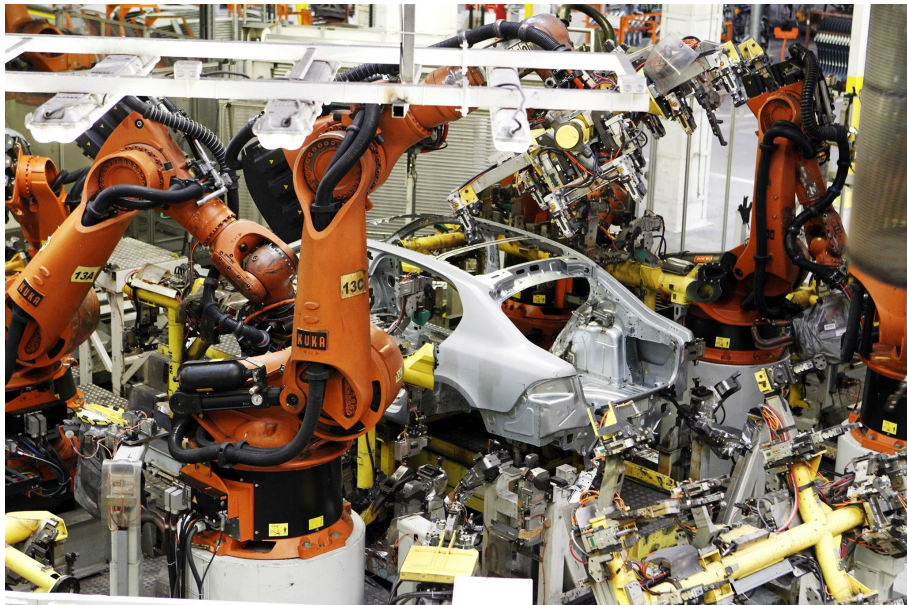
## Průmyslové roboty

Tato kapitola popisuje sériové šestiosé roboty, specificky robota PUMA 200 UMI-MATE a dále obsahuje základní teorii řízení DC servomotorů.

### 1.1 Šestiosé průmyslové roboty

Průmyslové roboty můžeme podle kinematické struktury rozdělit na sériové a paralelní. Šestiosé sériové roboty je možné rozčlenit do několika skupin. Pro toto dělení je důležitá kinematická struktura a druh pohonů.

Nejčastějším zástupcem jsou roboty se sériovou kinematickou strukturou a elektrickými servopohony. Oproti hydraulickým pohonům jsou servopohony efektivnější a lze je jednodušeji řídit.



Obrázek 1.1: Ukázka průmyslových robotů, Škoda Auto [3]

Na Obr. 1.1 je ukázka několika dnešních robotů, které se díky své všestrannosti

osvědčily prakticky u všech velkosériových výrobců jako je například ŠKODA AUTO, Apple Inc. nebo Tesla Inc.

Tento druh šestiosého robota je schopný zastávat funkce jako například sváření, lakování, montáž, lepení, manipulace a další. Typickou předností robotů oproti lidem je pracovat bezchybně v režimu 24/7 a tím navyšovat efektivitu práce.

Na Obr. 1.2, 1.3 je ilustrováno řešení největších výrobců automatizační techniky, kam patří ABB a KUKA.



Obrázek 1.2: Ukázka průmyslových robotů firmy ABB  
[4]



Obrázek 1.3: Ukázka průmyslových robotů firmy KUKA  
[5]



Přestože tyto roboty pochází z dílen různých firem, mají prakticky stejné kinematické uspořádání. Vždy jde o první tři rotační osy a následně sférické zápěstí. Důvod této konfigurace bude vysvětlen v kapitole Kinematika.

## 1.2 Robot PUMA 200 UNIMATE

Pro testy byl vybrán šestiosý robot PUMA 200 od firmy UNIMATE, viz Obr. 1.4. Název PUMA je zkratkou z anglických názvu "Programmable Universal Machine for Assembly" Tento robot byl zvolen díky svým kompaktním rozměrům (bezpečnost) a technickým parametrům, které jsou dostatečné pro vývoj řídicího systému.

Puma 200 byla dříve dodávána v poměrně velkém balení, kde robot tvořil tu menší část. Díky moderní technologii je možné velikost součástí minimalizovat a parametry zůstanou stejné či se dokonce zlepší.



Obrázek 1.4: Uvodní fotografie na manuálu pro PUMA 200. [1]

Robot PUMA 200 je klasický šestiosý manipulátor. První tři osy jsou osazeny DC motory, s výkonem ca. 75 W a úhlovým senzorem (kvadrurní encoder - bude dále vysvětleno), který má 1000 inkrementů na jednu otáčku. Tedy rozlišení tohoto čidla je  $\pm 0.36^\circ$  na otáčku. Tyto motory zajišťují pohyb prvních třech kinematických dvojic a oproti druhým třem kinematickým dvojicím mají elektromechanické brzdy. Druhé tři osy, které manipulují s robotickým zápěstím, jsou osazeny motory menších rozměrů, které jsou umístěny v druhém rameni. Tyto motory mají výkon ca. 35 W a jsou osazeny úhlovým senzorem s 800 inkrementy na otáčku. Tedy rozlišení čidla je  $\pm 0.45^\circ$  na otáčku.

Katalogové parametry robota PUMA 200 jsou [1]:

- Šestiosé rameno s tříosou sférickou dvojicí.
- Maximální dosah 400 *mm* od středu rámu (počátek souřadného systému) až do středu sférické dvojice.
- Maximální zatížení 2.2 *kg*.
- Hmotnost ramene 13.2 *kg*.
- Opakovatelnost  $\pm 0.05$  *mm*.
- Maximální rychlost 1245 *mm/sec* koncového bodu po přímé trajektorii.

Roboty PUMA byly průkopníky využití moderních robotů, kdy byl robot PUMA 200 dokonce použit pro asistenci při operaci mozku viz. Obr. 1.5. Tyto roboty vykazují velice precizní mechanickou konstrukci.



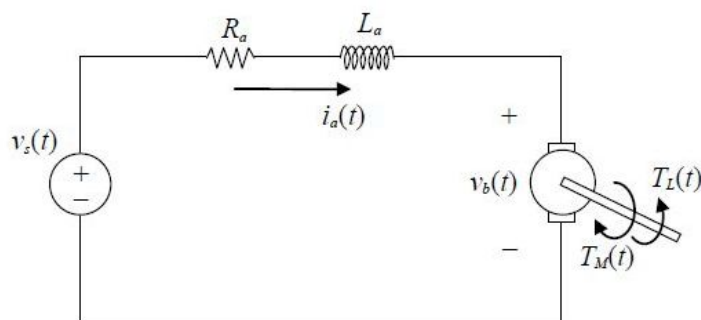
Obrázek 1.5: Operační asistent. [6]

## 1.3 Servořízení DC motorů

Zde jsou popsány principy řízení DC motorů.

### 1.3.1 Fyzikální popis motoru

Na úvod popíšeme rovnice charakterizující DC motor [2]. Tyto motory jsou použity v robotu PUMA 200. Schéma DC motoru s buzením permanentními magnety viz Obr. 1.6.



Obrázek 1.6: Schéma DC motoru.

Pro napětí dle *Kirchhoffova zákona napětí* lze psát následující uzavřenou smyčkovou rovnici (1.1).

$$R_a i_a(t) + L_a \frac{di_a(t)}{dt} + v_b(t) = v_s(t), \quad (1.1)$$

kde  $R_a$  je odpor vinutí motoru,  $i_a$  je proud protékající vinutím,  $L_a$  indukance,  $v_s$  je napájecí napětí a  $v_b$  je zpětné elektromotorické napětí.

Dále rovnice (1.2) popisuje závislost zpětného elektromotorického napětí  $v_b$  na otáčkách  $\omega(t)$  a konstantě  $k_b$ .

$$v_b = k_b \omega(t) \quad (1.2)$$

A nakonec rovnice (1.3) pro závislost kroutivém momentu na proudu vinutím.

$$T_M(t) = k_T i_a(t), \quad (1.3)$$

kde  $T_M$  je daný moment,  $k_T$  momentová konstanta a nakonec proud v cívce  $i_a$ .

Poslední důležitou rovnicí pro popis DC motorů je dynamická rovnice. Popisuje vztah momentů a dynamické odpovědi systému(1.4).

$$J_M \frac{d\omega(t)}{dt} + B_M \omega(t) = T_M(t) - T_L(t), \quad (1.4)$$

kde  $J_M$  je moment setrvačnosti motoru,  $B_M$  je koeficient tření a  $T_L$  je moment zátěže.

Použitím rovnic (1.1), (1.2), (1.3) a (1.4) vyjádříme rovnice (1.5) a (1.6) ve formě:

$$L_a \frac{di_a(t)}{dt} + R_a i_a(t) + k_b \omega(t) = v_s(t) \quad (1.5)$$

$$J_M \frac{d\omega(t)}{dt} + B_M \omega(t) - k_b i_a(t) = -T_L(t) \quad (1.6)$$

Poměr  $L_a/R_a$  je tzv. elektrická časová konstanta a poměr  $J_M/B_M$  je tzv. mechanická časová konstanta.

Elektrická časová konstanta bývá většinou zanedbána vzhledem ke své velikosti oproti mechanické konstantě. Můžeme tedy psát:

$$i_a(t) = \frac{v_s(t)}{R_a} - \frac{k_b}{R_a} \omega(t) \quad (1.7)$$

V kombinaci s (1.6) dostaneme:

$$\frac{\omega(t)}{dt} + \left( \frac{B_M}{J_M} + \frac{k_b^2}{J_M R_a} \right) \omega(t) = -\frac{T_L(t)}{J_M} + \frac{k_b}{J_M R_a} v_s(t) \quad (1.8)$$

Pro stavový popis těchto rovnic je požadovaným vstupem napětí zdroje  $v_s$  a moment zátěže  $T_L$ . Stavové proměnné jsou pak  $x_1(t) = i_a(t)$ ,  $x_2(t) = \omega(t)$  a  $x_3(t) = \theta(t)$ , kde  $d\theta/dt = \omega(t)$ . Nyní lze psát:

$$\dot{x}_1(t) = -\frac{R_a}{L_a} x_1(t) - \frac{k_b}{L_a} x_2(t) + \frac{v_s(t)}{L_a} \quad (1.9)$$

$$\dot{x}_2(t) = \frac{k_b}{J_M} x_1(t) - \frac{B_M}{J_M} x_2(t) - \frac{T_L(t)}{J_M} \quad (1.10)$$

$$\dot{x}_3(t) = x_2(t) \quad (1.11)$$

s výstupem:  $y(t) = x_3(t)$

V maticovém zápisu:

$$\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{A}\mathbf{x}(\mathbf{t}) + \mathbf{B}\mathbf{u}(\mathbf{t}) \quad (1.12)$$

a

$$\mathbf{y}(\mathbf{t}) = \mathbf{c}\mathbf{x}(\mathbf{t}), \quad (1.13)$$

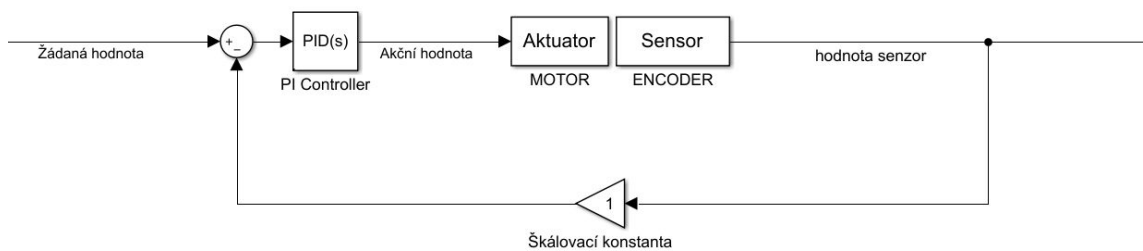
kde

$$\mathbf{x}(\mathbf{t}) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}, \quad \mathbf{u}(\mathbf{t}) = \begin{bmatrix} v_s(t) \\ T_L(t) \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} -R_a/L_a & -k_b/L_a & 0 \\ k_b/J_M & -B_M/J_M & 0 \\ 0 & 1 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1/L_a & 0 \\ 0 & -1/J_M \\ 0 & 0 \end{bmatrix}, \mathbf{c} = [0 \ 0 \ 1]$$

### 1.3.2 Kaskádní řízení

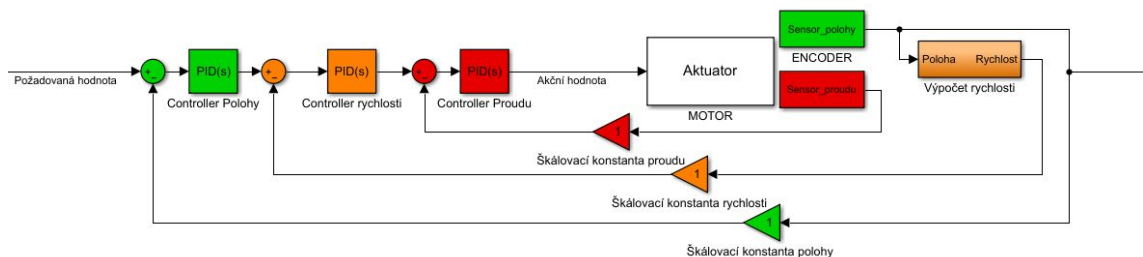
Pro řízení průmyslového robota s DC servopohonu je klíčové zpětnovazební řízení každé ze šesti os viz Obr. 1.7. Pro řízení je třeba mít aktuátor, senzor a kontroler, který řídí požadovanou veličinu v závislosti na regulační odchylce. V tomto případě jde o natočení konkrétní osy motoru, tedy problém řízení polohy.



Obrázek 1.7: Zpětnovazební regulace servomotoru - princip.

Vstupem do takto vytvořeného systému je požadovaná poloha natočení motoru. Od ní je odečtena změřená poloha natočení motoru. Tento rozdíl je následně zpracován v controleru (v našem případě PI regulátor). Dále už následuje přímý pokyn pro aktuátor (v našem případě motor).

V předchozím odstavci byla uvedena základní myšlenka servořízení. V případě reálných systémů se uplatňuje kaskádní regulace pohonů a to především z důvodu snadného ladění z vnitřní smyčky (nejrychlejší) směrem k vnější smyčce (nejpomalejší). Toto uspořádání je vidět na Obr. 1.8.



Obrázek 1.8: Kaskádní regulace DC motoru.

Toto uspořádání zajišťuje postupnou regulaci proudu, rychlosti a polohy.

Proudová smyčka zajistí dostatečně rychlý nárůst proudu a tedy řízení momentu. Rychlostí smyčka, jak již název napovídá, zajišťuje regulaci rychlosti a proto má systém při správném nastavení téměř skokovou odezvu na rychlost. A nakonec smyčka polohová, která zajistí správné natočení a udržení osy v požadové poloze.

Výše popsaný způsob řízení byl aplikován v případě každé ze šesti os robota. Jednotlivé regulátory je třeba naladit pro jednotlivé osy, případně motory, neboť každá z os má jiné dynamické vlastnosti. V našem případě první tři osy mají odlišné motory i encodery na rozdíl od druhých třech os.

Pro správné dosažení bodu a natočení v prostoru musí být dále sestaveno kinematické řešení robota. To zajistí, že pro požadovaný bod a natočení jsou spočteny konkrétní motorové souřadnice tzn. natočení každého motoru. Řešení tohoto problému je podrobně popsáno v kapitole Kinematika.

# Kapitola 2

## Implementace Hardware a Software

Tato kapitola popisuje klíčové prvky navrženého řídicího systému.

### 2.1 Původní řídicí systém

V době vzniku robota PUMA 200 byl pro řídicí systém použit kontroler *LSI-11*, který měl hmotnost *36kg* a rozměry *475mm x 312mm x 590mm* dle [1]. Díky moderní technologii byly možno tyto parametry mnohonásobně zlepšit.

### 2.2 Nový řídicí systém

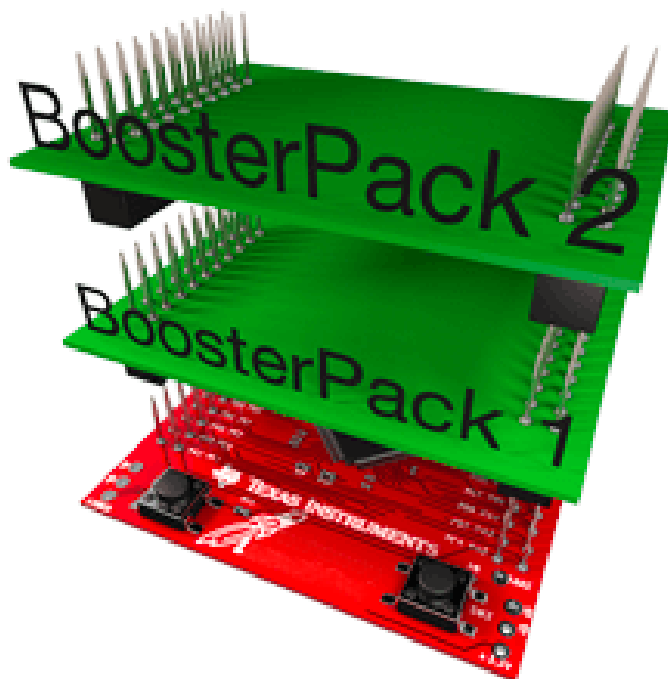
Pro vývoj nové řídicí elektroniky bylo zvoleno prostředí MATLAB/Simulink ve spojení s hardwarem od firmy Texas Instruments. Software MATLAB/Simulink obsahuje přímou podporu pro procesory *C2000*, které jsou přímo určeny pro řízení elektrickým motorů.

Pro vývoj řídicího systému byla zvolena platforma *LaunchPad* od TEXAS Instruments. Podporovanou podskupinou jsou i procesory *C2000*, které lze nalézt i v průmyslových aplikacích (servoměniče *AX5000* Beckhoff). Jejich podpora v Simulinku zajišťuje zprovoznění bez nutnosti hlubší znalosti jazyka *C*.

Tyto vývojové platformy jsou konstruovány tak, aby bylo možné připojení dalších modulů, které tento eco - systém rozšíří o specifické funkce. V tomto případě jde o výkonové moduly pro řízení DC motorů (*H-Bridge*). Prostřednictvím těchto modulů je možné tento systém uzpůsobit nejenom pro řízení motorů, ale i pro komunikaci přes bezdrátové sítě či zobrazování na displayi. Tyto moduly jsou typicky ovládány pomocí signálu posílaných přes standardizované rozhraní *GPIOs* (General-purpose input/output), které můžeme měnit pro určitou funkcionalitu.

LaunchPad platforma pro procesor *TMS320F28027* je zobrazena na Obr. 2.1. Tento

procesor je možné napojit na řadu rozšiřujících modulů. V případě průmyslové aplikace se vyvíjí specifická deska s elektrosoučástkami.



Obrázek 2.1: Ukázka obecného zapojení platformy LaunchPad [7].

## 2.3 Výběr mikroprocesoru

Pro řízení byla původně zvolena platforma s procesorem *TMS320F28069M*, který je přímo určen pro řízení dvou motorů zároveň. Během první fáze vývoje byla zjištěna nová funkcionality pro procesory *C2000* v prostředí Simulink, tzv. *External mode*. Jedná se o konfiguraci, kdy do mikroprocesoru je nahrán program a v reálném čase je možno sledovat procesy probíhající na procesoru. Toto je vhodné například pro přímou kontrolu čtení senzorů.

Procesor *TMS320F28069M* při vyčítání několika grafů na vzorkovací frekvenci *1kHz* nebyl schopen posílat data a muselo se přejít na výkonnější procesory stejné rodiny *C2000*. Jako vhodná náhrada byla zvolena platforma *C2000 Delfino LaunchPad F28379D* dále jen jako *F28379*. Tento procesor je schopen oproti *F28069M* vyčítat více dat a navíc je vybaven možností zpracovávat signály až ze tří senzorů polohy resp. řídit až tři motory zároveň [8]. Na Obr. 2.2 je vidět uspořádání *F28379*.

Zde jsou uvedeny základní parametry *C2000 Delfino LaunchPad F28379D* [9]:

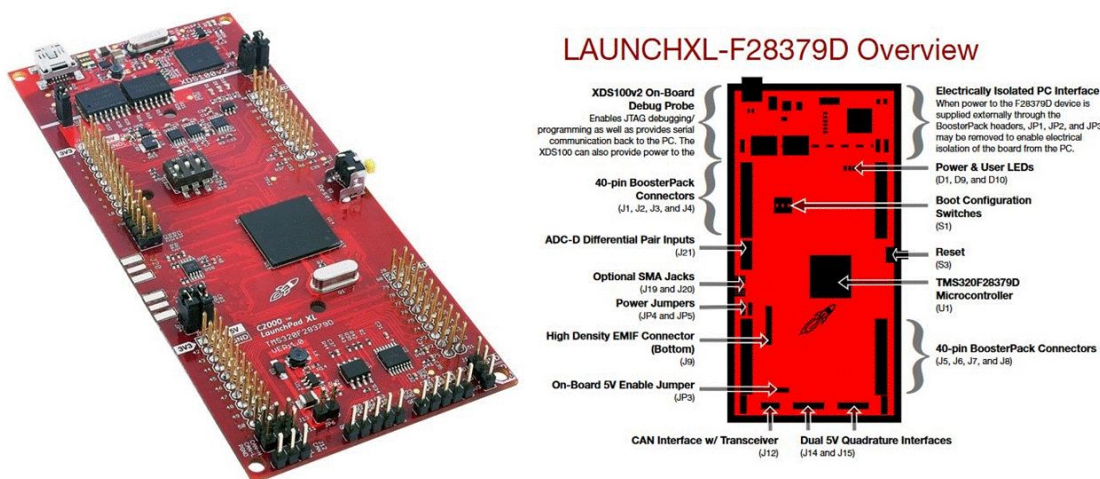
- USB připojení XDS100v2 JTAG pro ladění v reálném čase a flash programming.
- 4x 20-pinové konektory.



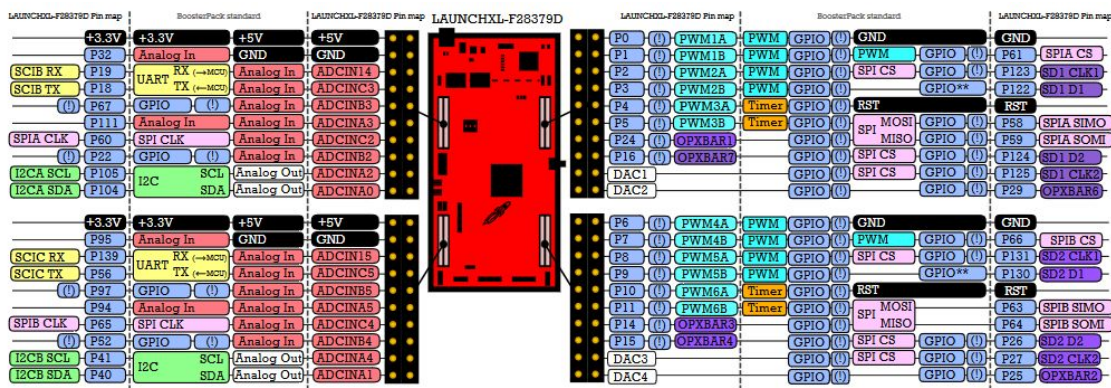
- Programovatelná tlačítka a LED diody.
- TMS320F28379D: 200 MHz dual C28xCPU a dual CLA, 1 MB Flash, 16-bit/12-bit ADCs, comparatory, 12-bit DACs, delta-sigma sinc filtry, HRPWM, eCAP, eQEP, CAN a další funkcionality.
- C2000 Delfino poziční správa TMS320F28379D MCU s možností propojení absolutních encoderů nebo resolverů.
- Podpora připojitelných modulů (*BoosterPacks*).
- Dva naletované encoder konektory plus třetí připojitelný.
- Isolovaný CAN konektor.

Možnost napojení třech sensorů pohybu (Encoder) byla klíčová, neboť pro řízení robota je zapotřebí jen dvou *F28379*. Tyto procesory paradoxně nemají pro takto jednoduché aplikace, jako je řízení více motorů, problém s výkonem (procesor je dokonce dvoujádrový). Avšak právě na vývojové platformě není dostatek výstupů / vstupů, aby mohl být jejich potenciál plně využit. Jak bude dále ukázáno, je zapotřebí konektory správně přepojit, aby mohly být všechny vhodné výstupy použité pro danou aplikaci a navzájem nekolidovaly.

Na následujícím Obr. 2.2 je ukázka platformy *F28379* a na schématu 2.3 je možnost zapojení různých funkcí [8].



Obrázek 2.2: *LaunchPad F28379D* [8], [9].

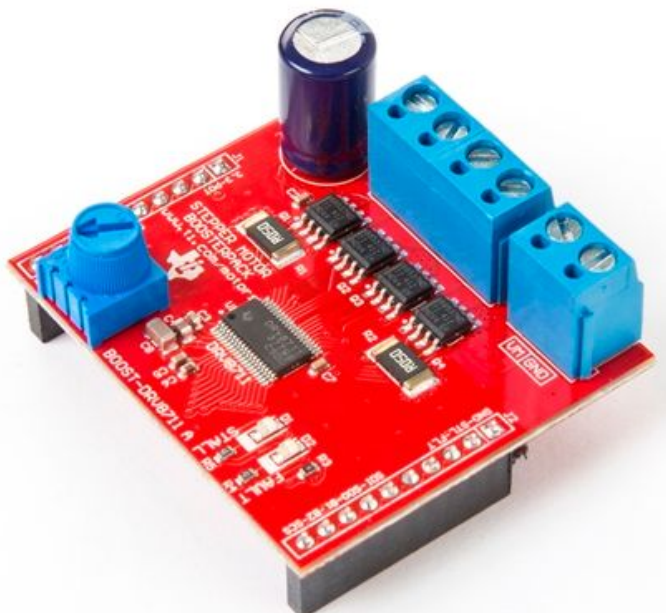


Obrázek 2.3: Schéma možnosti zapojení *LaunchPad F28379D* [8].

## 2.4 Výběr výkonového modulu

Pro zesílení požadovaného signálu a tedy ovládání proudu do motorů bylo již od počátku uvažováno s modulem *Boosterpack DRV8711* dále jen jako *DRV8711* [11] na Obr. 2.4. Primárně jde o výkonový modul pro krokový motor, ale po změně konfigurace lze použít jako dva *H-můstky*.

Tento rozšiřující modul byl vybrán především pro svůj rozsah napájecího napětí až  $52\text{ V}$  a proudovou zátěž. Další vlastností této platformy je plná kompatibility s *F28379* [10]. Jak bude ukázáno, lze tedy použít dva tyto moduly *DRV8711* a spojit je přímo s *F28379*. Tímto je zajištěno řízení až tří os na jeden mikroprocesor. Dále zůstává jeden výstup z modulu *DRV8711* volný pro řízení výkonového prvku jako jsou například brzdy prvních třech os. [1]



Obrázek 2.4: *Boosterpack DRV8711* [11].

Zde jsou uvedeny základní parametry *Boosterpack DRV8711* [11]:

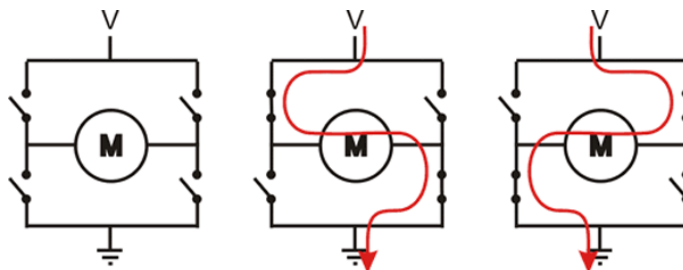
- $8-52V$  vstupní napětí až do trvalého proudu  $4.5A$  na každou ze dvou větví *H-můstku* (hlavní vlastnost).
- Vestavěná funkce mikrokrokování pro velmi jemný pohyb (pro krokové motory).
- SPI komunikace pro nastavení parametrů a vyčítání informací do mikroprocesoru.
- Velmi malé rozměry ( $4.41 \times 5.04 \text{ mm}$ ).
- Plně chráněno proti proudovému přetížení, přehřátí a nízkému napětí.

### 2.4.1 H-můstek (H-Bridge)

Toto zapojení je užíváno pro řízení směru a otáček DC motorů. Obsahuje čtyři výkonové tranzistory, které umožní sepnutí v jednom i druhém směru toku proudu až několikatisíckrát za vteřinu. To nám umožní především dávkovat střední hodnotu napětí na motoru a tím efektivně i proud jeho vinutím.

Na Obr. 2.5 je schéma takového zapojení. Zde je třeba zajistit, aby nikdy nebyly sepnuty tranzistory nad sebou. Tím by došlo ke zkratu a zničení takového můstku. Pokud jsou však křížem sepnuté tranzistory, dochází k toku motorem jedním směrem a opačně druhým.

Dva H-můstky, které jsou na *DRV8711* umožňují řídit až dva DC motory nebo jeden bipolární krokový motor.



Obrázek 2.5: Schéma *H-můstku* [12].

### 2.4.2 Zdroj napětí

Dle manuálu pro PUMA 200 [1], byl vybrán zdroj s obdobnými parametry výkonu. Tedy maximální možný výkon robota  $500W$ . Omezení napětí se odvíjí podle modulu *DRV8711*, který umožňuje maximálně  $52V$ . Tedy s nutnou rezervou byl vybrán zdroj s parametry  $48V, 10A, 500W$ . Napětí  $48V$  je také vhodné díky bezpečnosti práce.

Výkon byl ověřen výpočten:

$$P_{robot} = 3 \times 75 W_{Velky\ motor} + 3 \times 35 W_{Malý\ motor} = 330 W \quad (2.1)$$

Dále je nutné počítat s brzdami prvních třech os, které při maximálním zatížení (sepnutí při  $24V$ ) mají výkon:

$$P_{brzdy} = 40 \Omega \times 0,6 A = 24 W \quad (2.2)$$

Výkon celkem:

$$P_{robot} + P_{brzdy} = 354 W \quad (2.3)$$

A pro manipulační hlavici či další funkční prvky zbývá výkon 146 W.

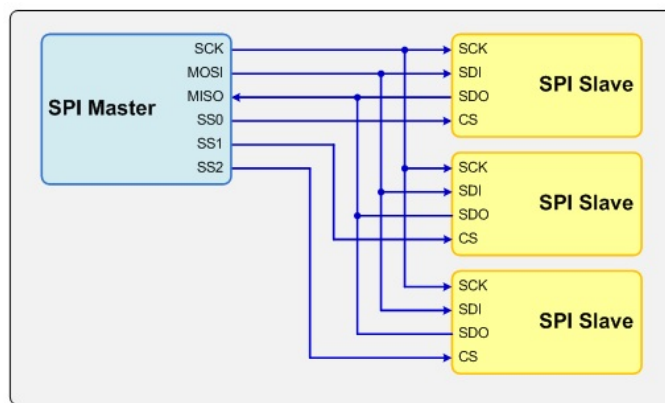
## 2.5 Hardware podpora pro procesory *C2000* v Simulinku

Následně je uveden popis klíčových bloků v programu Simulink, které jsou nutné pro syntézu řízení DC motorů na platformě *C2000*.

### 2.5.1 SPI komunikace

Pro nastavení výkonového modulu *DRV8711* je zapotřebí přepsat jeho registry a to lze přes SPI komunikaci. Následně bude vysvětlena její funkce:

SPI (Serial peripheral interface) je vysokorychlostní synchronní sériová vstupně výstupní komunikace, která umožňuje předem naprogramovaný přenos dat ve volbě od 1 do 16 bitů. Normálně se používá pro komunikaci mezi DSP procesory, externími perifériemi nebo jinými procesory. Tato komunikace umožňuje také posílání dat po jedné lince mezi více zařízeními. V tomto případě je nutné použití Master/Slave módu, přičemž má každé zařízení přiřazen svůj spouštěcí pin. Na Obr. 2.6 je zobrazeno schématické uspořádání pro více zařízení. [13], [15].

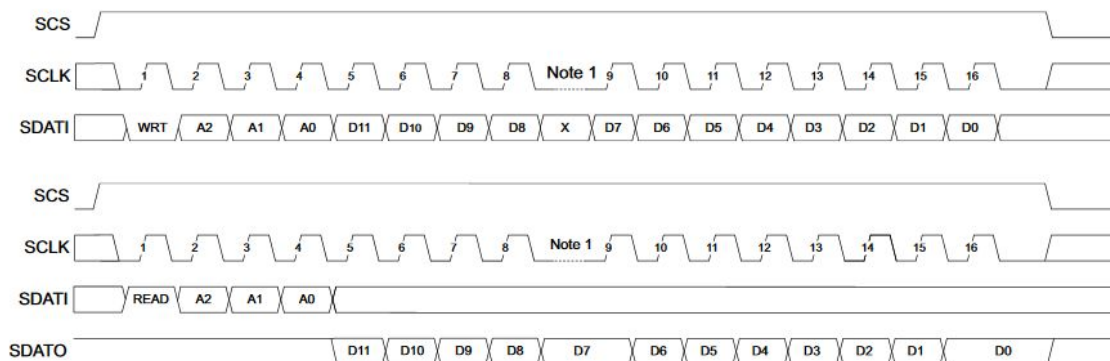


Obrázek 2.6: Schéma SPI komunikace [14].

Průběh komunikace probíhá následovně:

Vysílací zařízení (Master) vyšle pro dané přijímací zařízení (Slave) signál chip select (SCS), tím určí zprávu jemu určenou. Aby byla zajištěna synchronizace posílání zpráv, na jiném kanálu je spuštěn synchronizační signál (sclk). Jde o hodinový signál určité frekvence. Přijímací zařízení při každé vzestupné (možno volit i sestupnou) hraně tohoto signálu přečte na datovém kanálu (SDATI) logickou 1 nebo 0 a tím je určen dílčí bit zprávy.

Celá zpráva má tedy například pro 16 bitů i 16 pulzů synchronizace. Po celém odeslání přejde chip select do logické nuly a zařízení dále neposlouchá. Zápis/čtení komunikace je schématicky vyobrazena na Obr. 2.7 a také změřena na osciloskopu s kanály SDATI, synchronizačního signálu a chip select na Obr. 2.8.

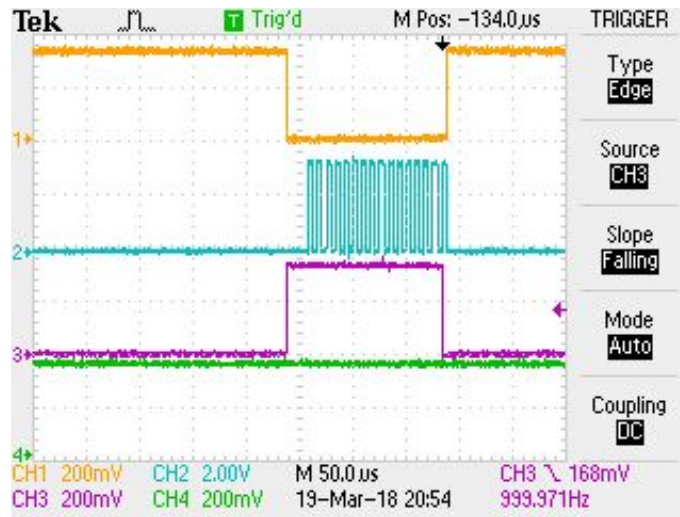


Obrázek 2.7: SPI komunikace, zápis/čtení [13].

Zpráva pro přenastavení potřebných registrů modulu *DRV8711* je složena ze dvou oddělených zpráv. Tyto zprávy musí přepsat 16 bitové registry. Jsou složeny z prvního bitu, který určí, zda-li jde o zápis (logická 0) či čtení (logická 1), tři bity určující adresu registru a nakonec v binární soustavě poslaná zpráva o velikosti 12 bitů.

Pro přeprogramování požadovaného nastavení je zapotřebí přepsat dva registry na adrese *0x010* a *0x000*. První zpráva umožňuje nastavení do módu PWM. Druhá zpráva pak funkci zařízení. Vše ostatní zůstává v továrním nastavení. Na Obr. 2.9





Obrázek 2.8: Záznam SPI komunikace. Zápis do registru 0x000.

je vidět celá struktura možného přeprogramování.

Přepnutí do PWM mode:

0010000100000000

Spuštění *DRV8711*:

0000000000000001

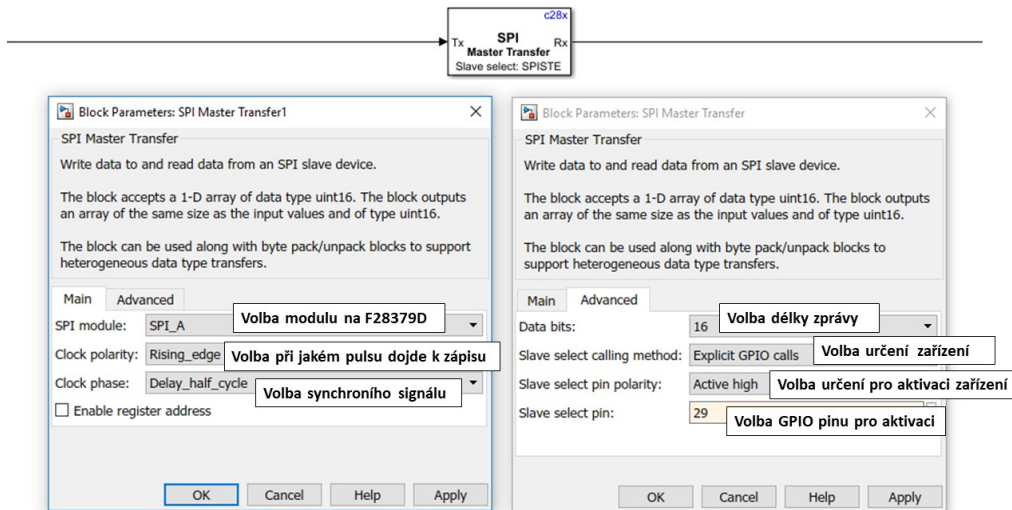
Name	11	10	9	8	7	6	5	4	3	2	1	0	Address Hex	
<b>CTRL</b>	DTIME		ISGAIN		EXSTALL	MODE			RSTEP	RDIR	ENBL	RW	00	
<b>TORQUE</b>	Reserved	SMPLTH			TORQUE								RW	01
<b>OFF</b>	Reserved		PWMMODE		TOFF								RW	02
<b>BLANK</b>	Reserved		ABT		TBLANK								RW	03
<b>DECAY</b>	Reserved	DECMOD			TDECAY								RW	04
<b>STALL</b>	VDIV		SDCNT		SDTHR								RW	05
<b>DRIVE</b>	IDRIVEP		IDRIVEN		TDRIVEP		TDRIVEN		OCPDEG		OCPH		RW	06
<b>STATUS</b>	Reserved				STDLAT	STD	UVLO	BPDF	APDF	BOCP	AACP	OTS	RW	07
Name	11	10	9	8	7	6	5	4	3	2	1	0	Address Hex	

Obrázek 2.9: *DRV8711* Mapa registrů [15].

Nastavení v Simulink bloku **SPI Master Transfer**:

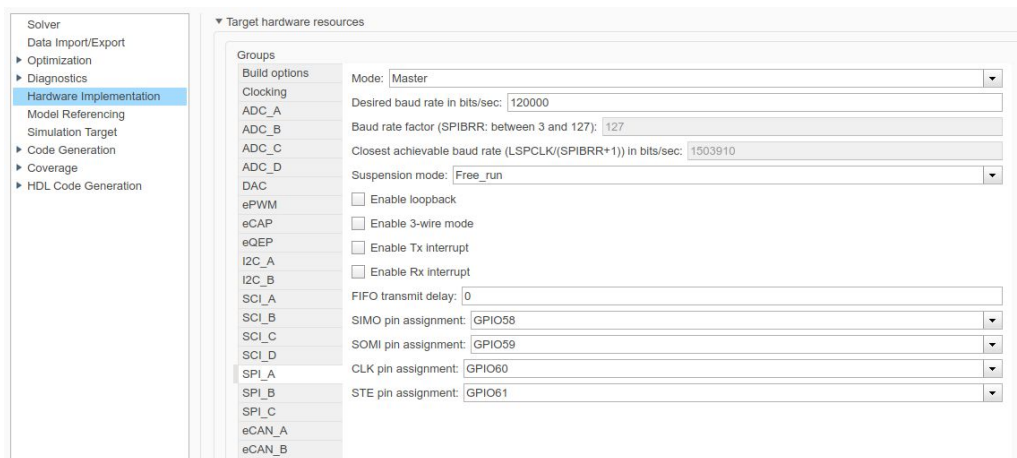
Pro nastavení výkonového modulu *DRV8711* je nutné přepsat registry na tomto chipu. To je umožněno v bloku, který se nazývá **SPI Master Transfer**. Možné volby nastavení jsou vidět na Obr. 2.10.

Zde celé zprovoznění vypadá velmi jednoduše, avšak při prvních testech nebylo prakticky možné modul *DRV8711* vůbec uvést do funkčního stavu. Vše bylo způsobeno nastavením SPI bloku v programu Simulink. Toto bylo zjištěno až po aktualizaci na novou verzi Matlabu. Zde byl přidán nový blok **SPI Master Transfer**. Oproti původní verzi **SPI Transmit** s posíláním zpráv již nebyl problém. Oba bloky jsou identické, avšak jen **SPI Master Transfer** je funkční.



Obrázek 2.10: Nastavení bloku SPI Master Transfer.

Kromě tohoto nastavení bylo nutné i nastavení konkrétních GPIOs výstupů na mikroprocesoru v nastavení *Model Configuration Parameters* v Simulinku. Ty jsou nastaveny tak, aby při nasunutí přídatného modulu *DRV8711*, odpovídaly požadovaným vstupům a bylo možné jejich ovládání viz Obr. 2.11.



Obrázek 2.11: Nastavení SPI parametrů GPIOs výstupů.

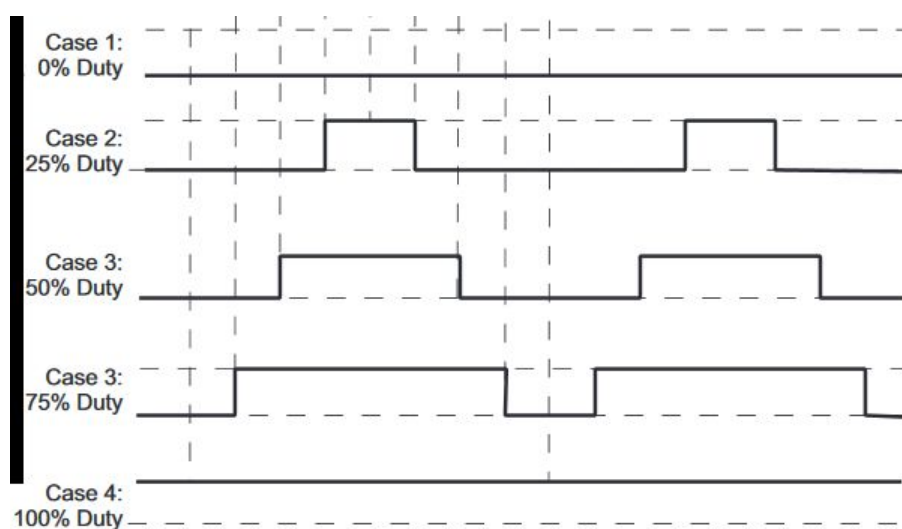
Všechna předchozí nastavení byla pouze programování daných registrů. Další bylo nutno zajistit správné propojení mezi modulem *DRV8711* a procesorem *F28379*, tedy následující zapojení:

- Uzemnění (*GND*);
- Napětí pro procesor VDC *3,3V*;
- Logická 0 (*GND*) pro pin *RESET*;
- Logická 1 (*3,3V*) pro pin *nSLEEP* (uvedení do provozu);
- PWM signály pro řízené napětí na motorech na piny *AIN1*, *AIN2*, *BIN1*, *BIN2* (lo-

gická 0 až 1), volba exkluzivity vstupu 1 nebo 2 určujícího směr rotace ( není možno poslat signál zároveň).

## 2.5.2 Modul ePWM

Pro ovládání otáček DC motorů je nutná schopnost měnit dle potřeby napětí přivedené do motoru. Toto lze uskutečnit různými způsoby, převážně se však používá pulzně šířková modulace (eng.: **P**ulse **W**idth **M**odulation). Tento způsob tzv. modulace napětí je založen na střední hodnotě napětí v čase. Máme-li například  $48V$  a žádané napětí je  $20V$  znamená to, že potřebuje při konstantní frekvenci (frekvence spínání PWM signálu) mít sepnuto  $\frac{20}{48}$  času. Pak nám vychází 41,6% času sepnuté vstupní napětí. Toto schéma lze vidět na Obr. 2.12, pro střidu 0%, 25%, 50%, 75% a 100%.



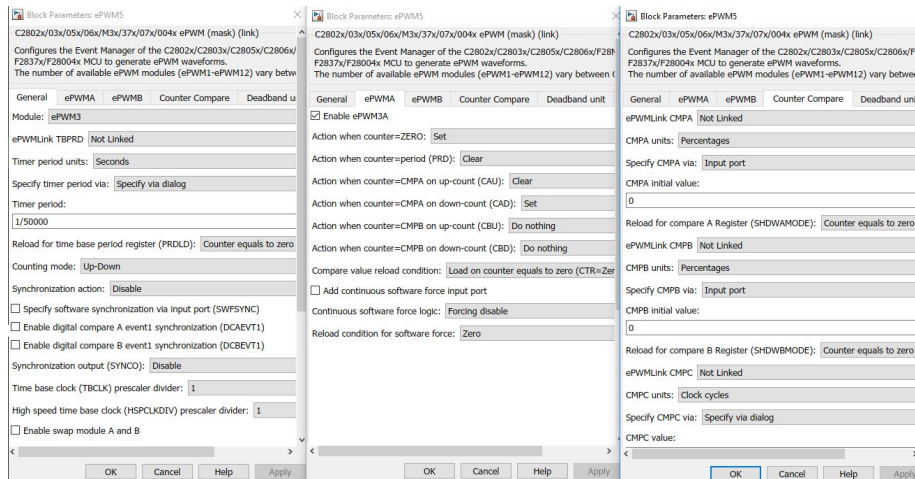
Obrázek 2.12: Schéma PWM signálu [16].

Vytvoření **PWM** signálu je na procesorem *C2000* v Simulinku k dispozici speciální blok zvaný *ePWM*, který umožňuje parametrizaci interního PWM modulu. Tento blok má velmi pokročilé možnosti nastavení i pro nejnáročnější aplikace, avšak pro účely řízení robota je dostačující základní nastavení se základní PWM frekvencí signálu  $25\text{ kHz}$ . Tato frekvence je nutná pro tichý chod motoru, neboť na nižších frekvencích je slyšet i lidským uchem. Na Obr. 2.13 je vidět toto nastavení.

ePWM blok zajišťuje vytvoření potřebného napětí na vinutí. Tento signál je poslán do budiče jednotlivých H-můstek modulu *DRV8711*.

Již během prvních fází testování bylo zjištěno, že parametry obvodu jsou frekvenčně závislé a vykazují jistou míru nelinearity. Toto zjištění významně přispělo k vysvětlení problému s funkčním nastavením řídicí smyčky (nelinearita aktuátoru). Tento problém byl vyřešen linearizační funkcí, nejedná se tedy o řízení, ale ovládání (proudová zpětnovazební smyčka nebyla realizována).

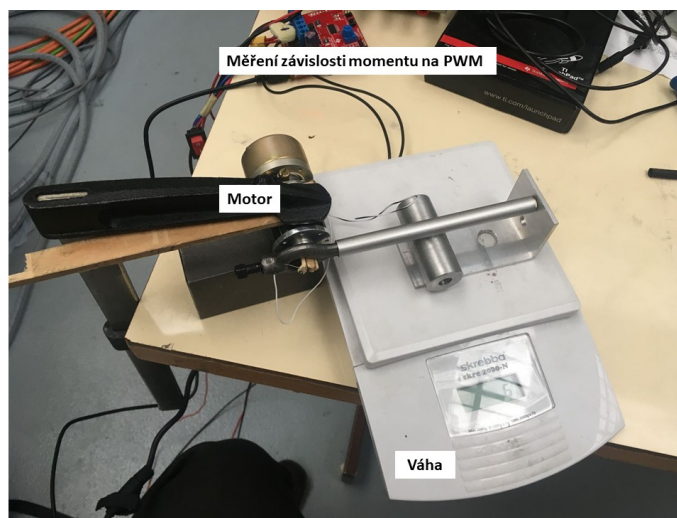




Obrázek 2.13: Základní nastavení ePWM bloku v Simulinku.

### 2.5.3 Linearizace motorů

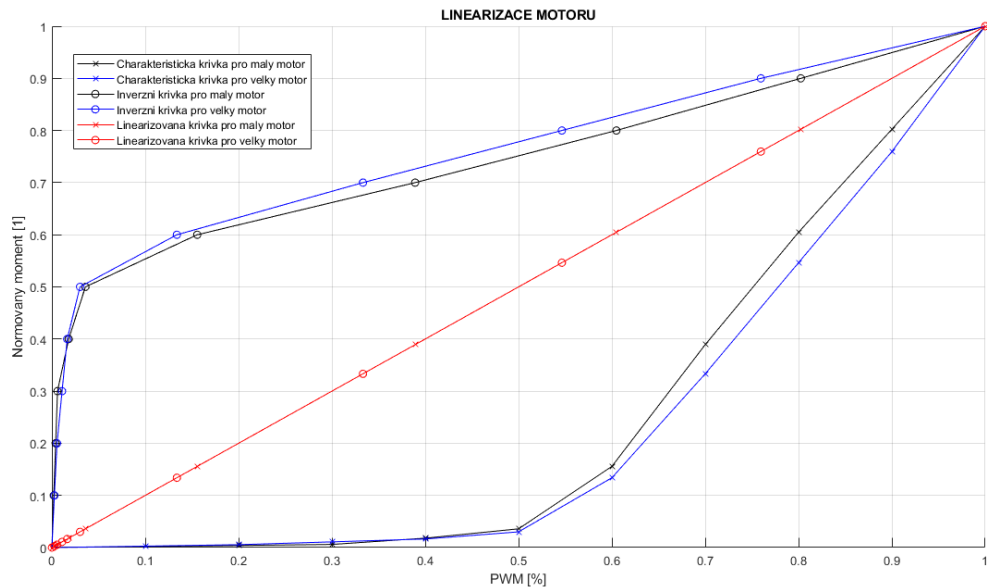
Motory bylo nutné nejprve experimentálně změřit, vzhledem k jejich stáří nejsou žádné parametry veřejně dostupné. Byl staticky změřen moment viz Obr. 2.14 a dynamicky změřeny otáčky bez zátěže pomocí laserové otáčkoměru.



Obrázek 2.14: Experimentální měření statického momentu.

Z měření byly získány následující průběhy viz Obr. 2.15, které ilustrují výše popsání nelineární chování. Na této ilustraci je uvedena i linearizační křivka pro oba motory a výsledný průběh po linearizaci vyneseny červenou barvou.

Tato inverzní funkce byla v Simulinku předřazena před vstup do ePWM bloku. Výsledkem je lineární charakteristika závislosti momentu na PWM vstupu. Tento problém se může zdát být jednoduchý, ale za úspěšným řešením jsou desítky hodin práce a mnoho úvah, proč není zpětnovazební řízení stabilní.

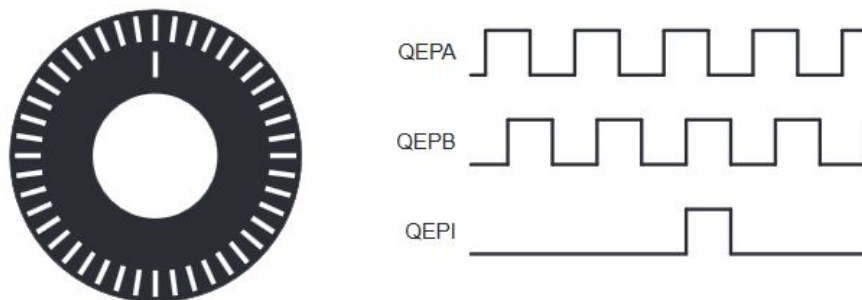


Obrázek 2.15: *Nelinearita řídicího signálu PWM a motoru včetně korekce.*

### 2.5.4 Modul eQEP

Pro přesné řízení polohy os robota je na každý motor připojen senzor polohy - *encoder*, který snímá úhlové natočení osy motoru. V případě řízení DC servomotorů jde o tzv. inkrementální kvadraturní optické encodery.

Konstrukce tohoto senzoru se skládá z oběžného kola, které je pevně spojeno s osou motoru. Na tomto kole jsou natištěny pulzy pro optické snímače, které jsou upevněny na základní desce společně s výpočetní elektronikou. Pro správnou funkčnost encoderu jsou snímače pulzů *A* a *B* oproti sobě posunuty o půl optického pulzu natištěného na skleněném disku. Tím je zajištěno, že reagují vždy s fázovým zpožděním čtvrt pulzu a je možné detekovat čtyřikrát více pozic než je natištěných pulzů. Proto kvadraturní encoder. Optické senzory *A* a *B* reagují na každou náběžnou a sestupnou hranu pulzu. Viz Obr. 2.16



Obrázek 2.16: *Schéma oběžného kola a výčet signálů A a B [17].*

Pro využití encoderů na motorech bylo nutná jejich identifikace. Počty pulzů encoderů jsou odlišné pro první a druhé tři osy robota. Na první tři osy připadá 250 inkrementů, tedy 1000 pulzů na otáčku. Na druhé tři osy připadá jen 200 inkre-

mentů a tedy pouhých 800 pulzů na otáčku. Toto rozlišení je velmi hrubé. Proto se musí výpočet rychlosti, která je nutná pro správnou funkci rychlostní smyčky, filtrovat. Dále je rozebráno v podkapitole *Výpočet rychlosti*.

Pro čtení **QEP** signálu je pro procesory *C2000* v Simulinku speciální blok zvaný *eQEP*, který zajišťuje vyčítání signálu získané na vstupu.

Tento modul má velmi rozsáhlé nastavení, které je možné vidět v připojeném programu. Modul je nastaven tak, aby ze vstupních signálů *A* a *B* zapisoval do 32 bitového registru počet aktuálních pulzů.

Pro zprovoznění třetího eQEP modulu bylo nutné přiřadit ještě příslušné GPIOs na *F28379*.

Pro správnou funkci tohoto modulu je nutné připájet na externím paktivém poli správně zapojený **RC** filtr signálu dle manuálu [8]. Po této úpravě modul funguje identicky jako první dva, které mají již připravené konektory s filtry ve spodní části *F28379*.

## 2.6 Výpočet rychlosti

Jak již bylo řečeno velmi hrubé encodery byly problémem právě pro získání signálu rychlosti.

Dle manuálu [17] lze aktuální rychlost vypočítat dvěma způsoby:

- $v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T}$
- $v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T}$

První způsob je běžné určení rychlosti. Změna dráhy (počet pulzů) za čas. Tento výpočet funguje správně při vysokých rychlostech, kdy máme dostatek pulzů. Avšak při velmi malé a při žádné rychlosti již není možné rychlost stanovit, neboť čítač (counter), které měří čas je po příliš dlouhém čase resetován a rychlost tudíž není stanovena.

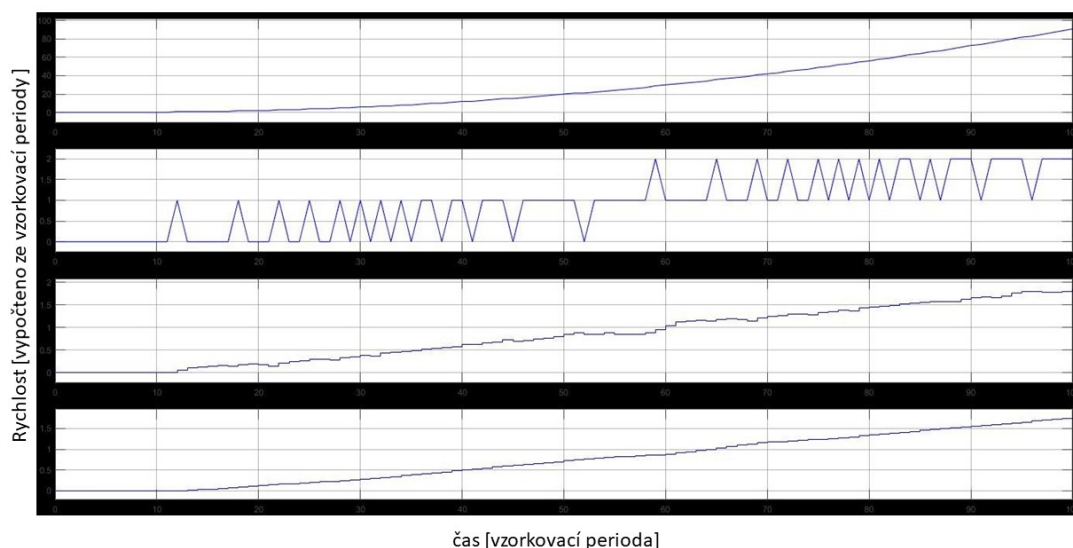
Druhý způsob je určen právě pro pomalé rychlosti. Zde je počítán čas mezi dvěma pulzy a tím je rychlost estimována. Tento výpočet však není přesný pro vysokou rychlost, neboť nastane opačný problém.

Pro pomalý výpočet rychlosti je tento modul bohužel nepoužitelný, neboť z neznámých důvodů nefunguje. Tento případ by rozebírán na fórech podpory Mathworks a Texas Instruments, ale nebyl vyřešen v době zprovoznování robota. Proto bylo nutné přejít na jinou metodu výpočtu rychlosti.

## 2.6.1 Zpracování signálu pomocí regresní funkce

Pro správnou funkcionalitu regulace, musí být rychlostní signál, co nejjemnější, aby nedocházelo k příliš velkým skokům odchylky rychlosti. To lze matematicky vyřešit například průměrováním *moving average*. Avšak signál se o daný počet snímků zpozdí a navíc průměr nerespektuje směrnici trendu.

O něco lepší se ukázalo použití lineární regrese, která oproti průměrování dat respektuje směrnici a zachovává trendy dat s menším zkreslením. Testováním se ukázalo, že při frekvenci rychlostní zpětné vazby  $1\text{ kHz}$ , je nevhodnější výpočet regrese z posledních  $10$  hodnot. Jde o kompromis mezi zpožděním dat a kvalitou signálu. Na Obr. 2.17 je možné vidět vyhlazení dat pomocí regresní funkce.



Obrázek 2.17: Zhora: poloha, rychlost z derivace (velká diskretizační chyba), regrese z 20 hodnot a regrese z 10 hodnot.

## 2.6.2 Výpočet pomocí *Pozorovatele (Observer)*

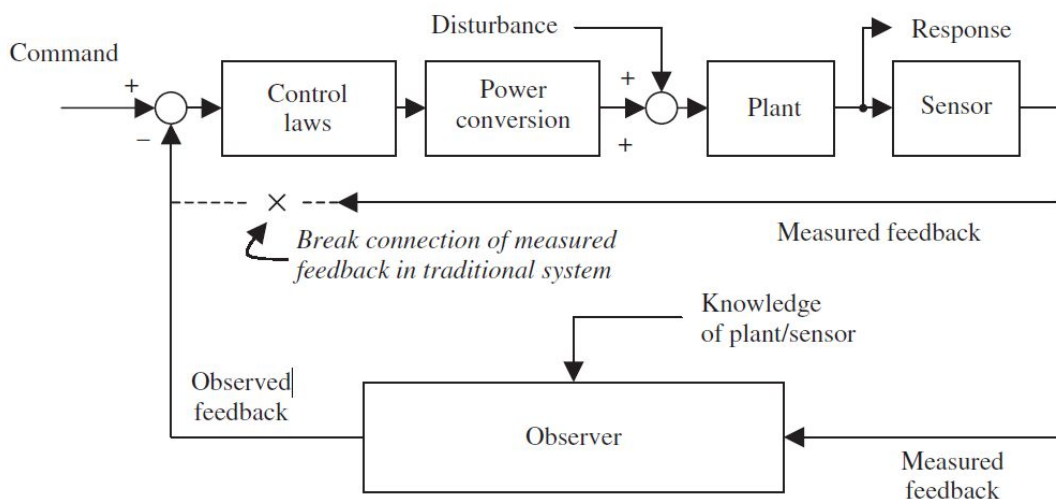
Ještě před naladěním regresní funkce bylo uvažováno i o tzv. *Pozorovateleli* (eng.: *Observer*). Tento algoritmus je vhodný právě pro systémy, kde máme špatná výstupní data, jako například hruhá úhlová data z encoderu.

*Pozorovatel* může být využit jako doplnění či dokonce kompletní náhrada senzorů v systému. To může zlevnit konkrétní systém, neboť senzory a jejich kabely patří k těm nejdražším komponentům. *Pozorovatel* (simulační model reálného systému) je synchronizován s reálným systémem na základě změřené a simulované hodnoty. Stav měřený na simulačním modelu mohou být dokonce přesnější a spolehlivější

než signály získané přímo ze senzoru. Toto platí však v případě dobré shody modelu a reality, současně s kvalitním nastavením synchronizace *Pozorovatele*. To dává inženýrům alternativu k přidání nových senzorů či nahrazení stávajících.

Princip *Pozorovatele* je založen na kombinaci změřeného zpětnovazebního signálu se znalostí komponent systému. Chování systému jde navíc lépe zjistit než pouze čtením zpětnovazebního signálu.

Velkou výhodou *Pozorovatele* je jeho možnost výrazného zvýšení kvality signálu a navíc může zmenšit zpoždění oproti senzorům, které sledují již nastávající jevy [18].



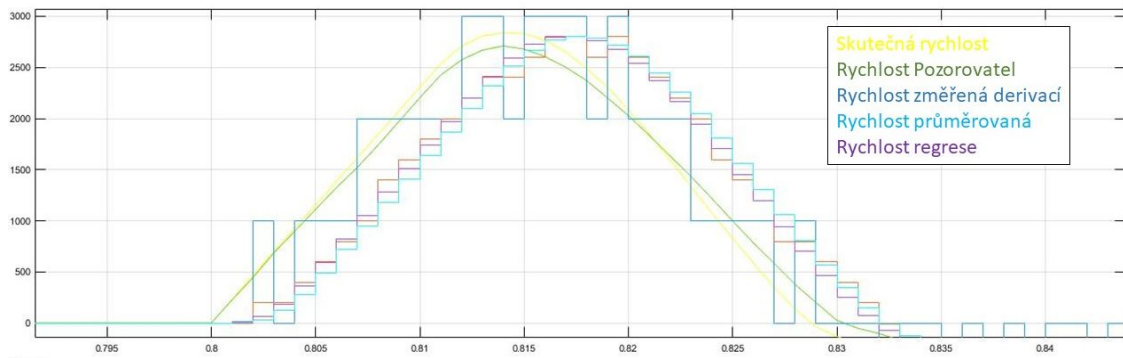
Obrázek 2.18: Schéma Pozorovatele v systému [18].

Použití *Pozorovatele* není všespásné. *Pozorovatel* přináší větší komplexitu do systému a je třeba výpočetního výkonu na jeho realizaci. Také může být méně robustní než fyzické senzory a to především pokud se vlastnosti systému mění během chodu. Avšak ve většině případů přináší *Pozorovatel* výhodu efektivnosti, neboť mohou výrazně snížit celkové náklady systému a zvednout spolehlivost [18].

Na Obr. 2.19 je vidět porovnání různých zpracování signálů. Zde je vidět skutečně velké zlepšení jak ve vyhlazení, tak i ve zpoždění signálu.

*Pozorovatel* byl realizován pro získání simulační hodnoty rychlostního signálu v případě malého motoru. V konečné implementaci nebyl realizován, neboť nebyl dostatek času pro realizaci identifikace jednotlivých motorů. Pro danou funkci je nutné co nejpřesněji identifikovat motory v systému pro každou osu robota, která má svoji dynamiku. Navíc v době implementace nebyl k dispozici plánovaný zdroj napětí, který významně mění vlastnosti chování motorů.

S použitím *Pozorovatele* je počítáno do budoucna ve druhé fázi zlepšení řízení robota.



Obrázek 2.19: Rychlost, malý motor.

## 2.7 Modul eCAN

Tato podkapitola není obsahem diplomové práce. Byla vyvinuta na Fakultě strojní a pro tuto práci použita, aby bylo možné ovládat první tři a druhé tři osy robota synchronizovaně, ze dvou *Launchpadů*. Zde je uvedena pro celkovou ilustraci řízení robota.

CAN (*Controller Area Network*) komunikace používá sériový multimaster komunikační protokol, který dostatečně efektivně přenáší data v reálném čase. Jde o velmi robustní komunikaci s rychlostí až  $1\text{ Mbps}$ . CAN modul je ideální pro signálově hlučná a drsná prostředí jako je automotive či ostatní průmyslová odvětví vyžadující spolehlivou komunikaci [19]. Byl původně vyvinut firmou Bosch.

Jako ostatní moduly podporované v Simulinku pro procesory *C2000* i zde existuje příslušný modul *eCAN*. V případě posílání zpráv s informací o poloze zde komunikuje PC vysílač s MCU *Launchpady F28379*. PC odesílá každou  $1\text{ ms}$  jednu zprávu pro každý *F28379*, který ji dekoduje a distribuuje pro danou osu robota.

V první řadě je odeslána zpráva zapínající zpětnovazební řízení a následné odjištění brzd (v případě *MCU1*). Ve druhé zprávě již přijdou žádané polohy motorů.

Obě zprávy chodí ihned po sobě, proto prakticky nedojde ke zpoždění mezi zařízeními.

Během prvních testů se ukázal velký problém při posílání dat v reálném čase. To mělo za následek nestabilní zpoždění či dokonce chyby v posílaných datech. Proto bylo nutné do komunikace přidat datový akumulátor (*Buffer*). Ten zajišťuje akumulaci posledních  $1000$  zpráv, tedy  $1\text{ s}$  záznamu žádaných poloh. Následně jsou data v pořadí první dovnitř první ven (*FIFO*) přeposílána do řídicí smyčky a tím je zajištěna plynulost pohybu.

Na straně vysílacího PC *Windows 10* je realizováno vysílání dat v taktu jedna milisekunda prostřednictvím zpomalovacího bloku z balíčku *Windows Real Time Target*.

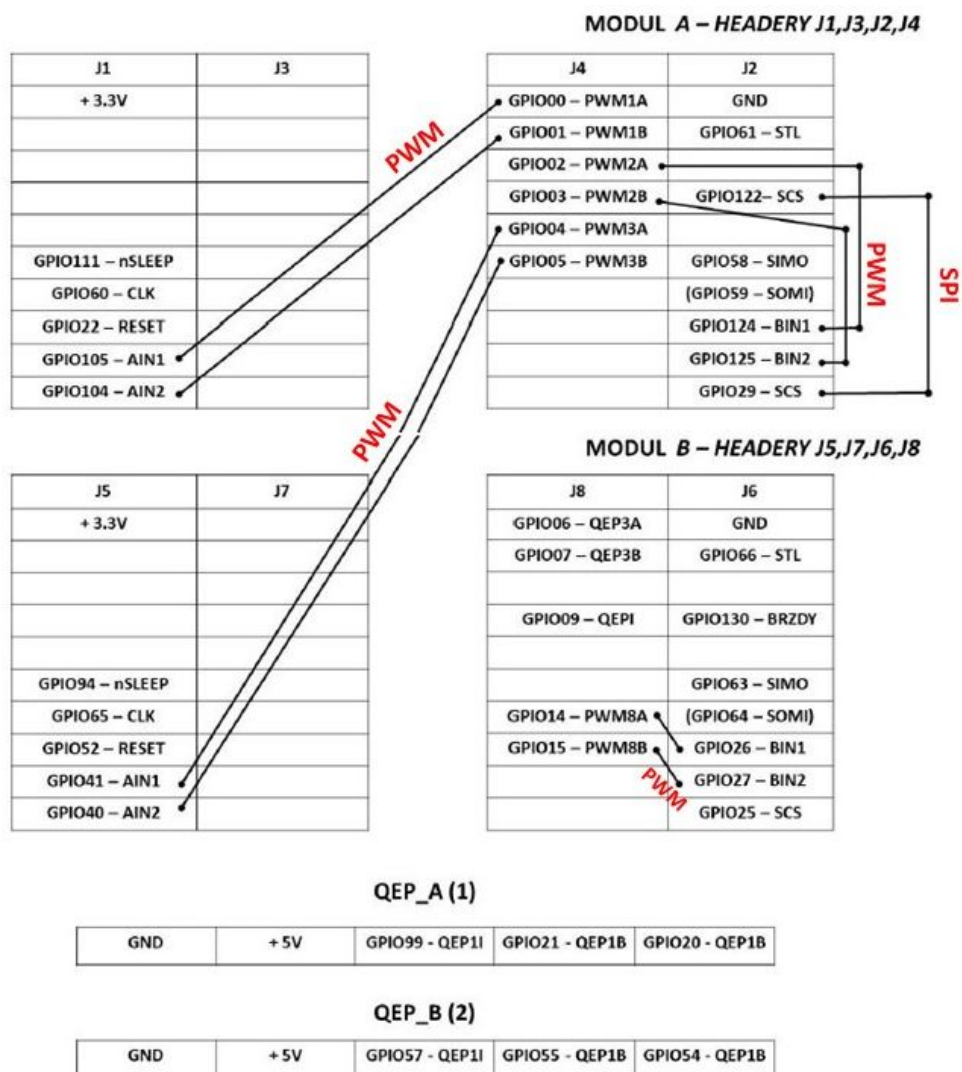
Pro zajištění funkčnosti sběrnice CAN pro *MCU1* a *MCU2*, které jsou v sériovém zapojení, bylo nutné z *MCU1* odpájet  $120\Omega$  terminační odpor. Ten zajišťuje im-



pedanční úpravu sběrnice způsobem, aby bylo zamezeno odrazům signálu na konci sběrnice v průběhu arbitrace.

## 2.8 Zapojení řídicí elektroniky

Pro správné zapojení všech požadovaných funkcí na *F28379* je nutné správně propojit konkrétních vstupy a výstupy. Na výstupních konektorech *F28379* jsou určité GPIOs explicitně přiřazeny určité funkcionalitě a nejde je použít pro jinou funkci. Například jde o moduly generující PWM signál. Ty jsou vždy jen na konektoru *J4* a *J8*. Pro libovolně nastavitelné GPIOs jsou vyhrazeny konektory *J1*, *J2*, *J4*, *J5*, *J6* a *J8* viz Obr. 2.20.

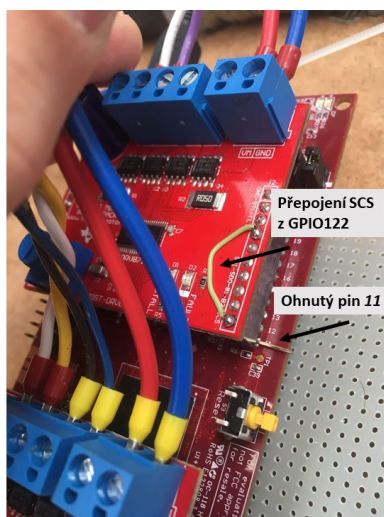


Obrázek 2.20: Zapojení konkrétních Headerů na *F28379*.

## 2.8.1 Softwarová chyba v Simulinku

Při připojení druhého modulu *DRV8711* na modul A *F28379* nastal problém. V nastavení Simulinku nebylo možné přiřadit odpovídající *GPIO29* na pin *11* pro funkci *SCS*, která je nezbytná pro přijetí zprávy komunikace *SPI*. Bez této funkcionality by horní modul A vůbec nebylo možné použít v konfiguraci, která dělá *boostpacky* tak uživatelsky příjemné, díky předem definovanému rozhraní.

Byla kontaktována podpora Texas Instruments a Mathworks. Technická podpora Mathworks chybu potvrdila s tím, že bude nutno počkat do příštího update vestavěných software balíčků (*Support packages for C2000*). Nebylo možné čekat několik měsíců na vyřešení tohoto problému ze strany Mathworks a proto bylo nutné daný pin přepojit z jiného GPIO pinu, specificky *GPIO122*. Na Obr.2.21 je vidět vyřešení daného problému včetně nutného ohnutí pinu *11*, kterému byl původně přiřazen pin *GPIO29* nenastavitelný v modulu *SCIA*.



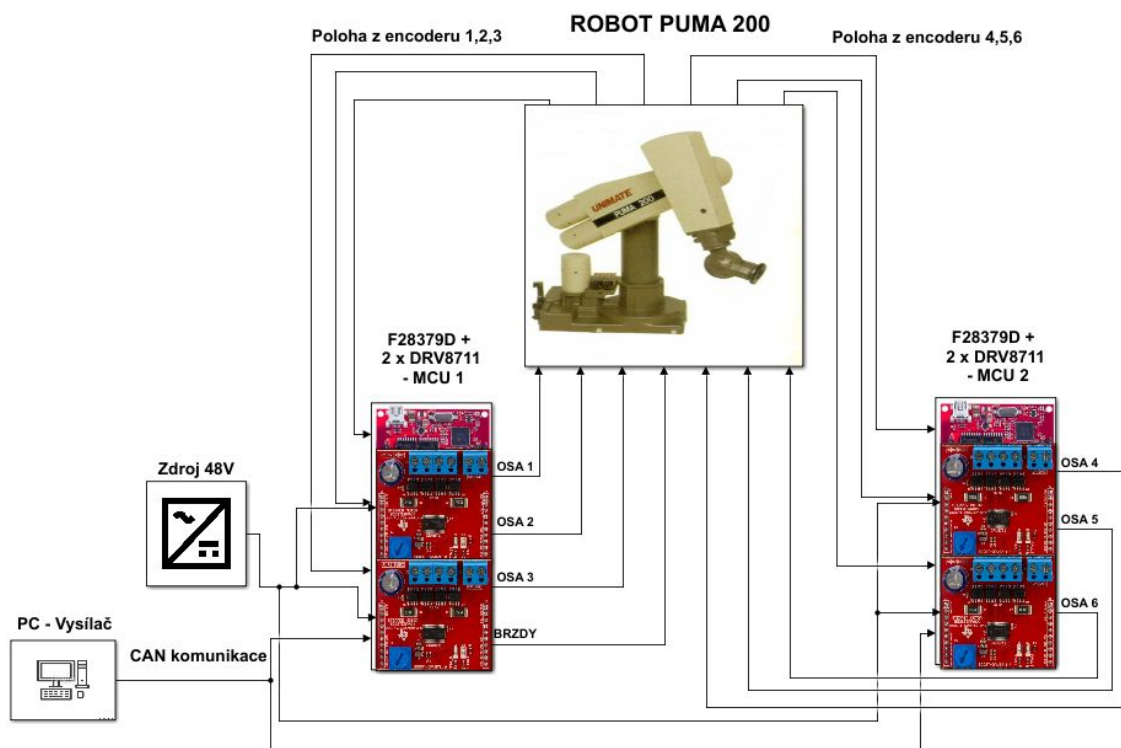
Obrázek 2.21: Vyřešení softwarové chyby přepojením z jiného výstupu.



## 2.8.2 Zapojení všech komponent

Na Obr. 2.22 je uvedeno ilustrační zapojení všech komponent systému v jeden celek: a) PC pro posílání dat programu (trajektorie); b) dva *Launchpady F28379*, který každý má zapojeny dva *DRV8711*, celkem tedy čtyři; c) Zdroj napětí 48V; d) samotný robot PUMA 200.

Mezi všemi komponenty je schématicky zobrazeno propojení. CAN komunikace od PC k *F28379*, napájení pro moduly *DRV8711* a výkonová a zpětnovazební smyčka mezi výkonovou elektronikou a robotem.



Obrázek 2.22: Ilustrační schéma zapojení Hardware.



# Kapitola 3

## Řízení v Simulinku

Tato kapitole popisuje konkrétní schémata řídicích algoritmů v Simulinku.

Řízení robota se skládá ze dvou částí. První část je PC s programem Simulink, která generuje program a následně počítá z inverzní kinematiky natočení konkrétních os robota. Tato data jsou následně posílána přes sběrnici CAN bus do druhé části systému (Hardware), tedy *MCU1* (první Launchpad *F28379*) a *MCU2* (druhý Launchpad *F28379*). Kaskádní regulace na *MCU1* a *MCU2* zajišťuje vykonání natočení os robota.

### 3.1 Kaskádní regulace na *MCU1* a *MCU2* - přijímač

Systém řízení musí být dále rozdělen na dva Launchpady *F28379*, kde první *MCU1* řídí první tři osy včetně brzd a kde druhý *MCU2* řídí druhé tři osy. Obě tyto zařízení jsou připojena na CAN komunikační sběrnici.

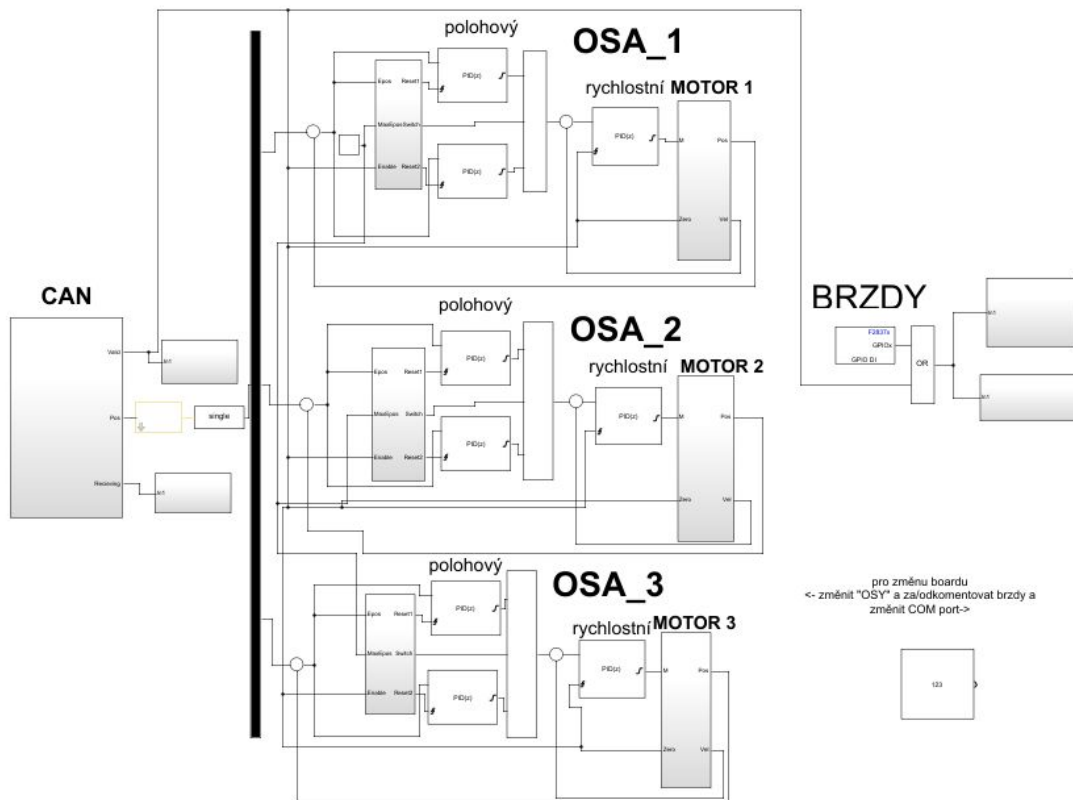
Jak bude dále ukázáno na schématu řízení, celý systém je složen z několika bloků, tvořících dekódování CAN komunikace, uložení dat do vyrovnávacího Bufferu, generování PWM signálů a zpracování QEP signálů. Celý řídicí systém je tvořen souhrnem těchto dvou subsystémů. Tyto programy jsou podrobně k nahlédnutí na příloženém DVD.

Pro ilustraci je na Obr. 3.1 uveden program pro *MCU1* případně pro *MCU2*. Jediným rozdílem jsou jiné parametry nastavení PID regulátorů, linearizace motorů, absence brzd a jiný kód identifikace CAN komunikace.

Toto schéma je ze Simulinku následně přeloženo do programu *C*, který je po kompilaci nahrán na konkrétní MCU. Na schématu 3.1 jsou vidět čtyři okruhy.

První s názvem BRZDY je určen pro odbrždění robota. Tento okruh programu je inicializován CAN komunikací a kdykoliv je CAN komunikace ztracena, brzdy se zabrzdí, aby nedošlo k nekontrolovatelnému pohybu robota.

Druhý až čtvrtý okruh je vyčleněn pro osy 1 až 3 a slouží k řízení os robota. Jsou zde vidět zpětné vazby pod názvem *rychlostní* a *polohová*, které jak nápis napovídá slouží k řízení rychlostní respektive polohové vazby.

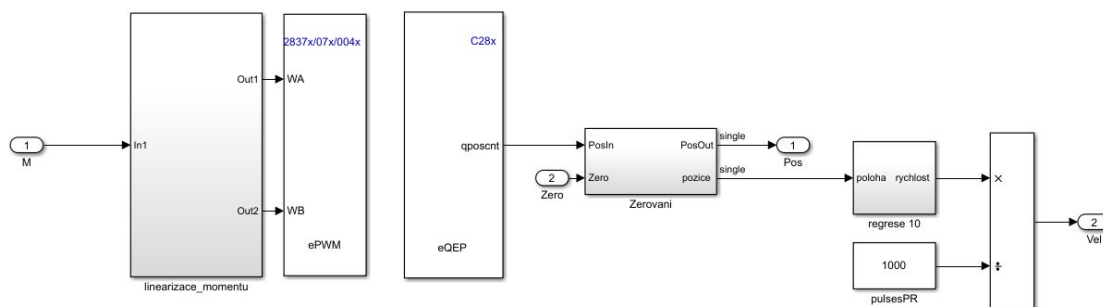


Obrázek 3.1: Program pro MCU1 případně MCU2 v Simulinku

Na Obr. 3.1 je před polohovou vazbou vidět předřazený blok, který dle velikosti rozdílu požadované pozice přepne mezi dvěma PID regulátory. Každý regulátor má nastavenou jinou saturaci výstupního signálu. Tato úprava má za úkol zajistit, dostane-li robot příliš velký požadovaný skok na změnu natočení o více než 400 pulzů (natočení osy motoru o více než 0,4 otáčky), je přepnuto na regulátor s omezením výstupu. Tím je zajištěno, že robot nepřejde velkou rychlostí do požadované polohy a je tím ochráněna mechanická struktura (převodovky, motory) před poškozením. Tato ochrana je jinak nastavena pro druhé tři osy viz schéma v příloze.

Pod *subsystémem MOTOR* jsou bloky, které převedou informaci na PWM signál (ePWM blok) a také odečítají polohu z encoderů (eQEP blok). Tento subsystém je ilustrován na Obr. 3.2. Zde je také vidět blok linearizace, který zajišťuje již zmíněnou linearizaci reakci motorů na požadovaný moment. Dále též vynulování encoderů při zapnutí systému *Zerovani* a také filtrace výstupní rychlosti s nastavením počtu hodnot pro výpočet *regrese*.

Po nahrání tohoto schématu do MCU1 a MCU2 jsou tato zařízení samostatná a kdykoliv jsou napájena je schéma aktivní, vyčkávající na CAN zprávu pro začátek programu.



Obrázek 3.2: Schéma pro řízení motorů ze Simulinku

## 3.2 PC - vysílač

Pro posílání dat do MCU1 a MCU2 je na straně vysílače PC. Zde je spuštěno simulační schéma 3.3, které realizuje inverzní kinematiku a generuje CAN zprávy pro natočení poloh os robota. Toto schéma běží na frekvenci  $1kHz$ .

V tomto schématu se po spuštění simulace musí vybrat daný program pomocí změny konstanty v bloku *Prepinac programu*. Zde byly pro ověření funkčnosti robota vytvořeny dva programy. Za prvné, kreslení kružnice v rovině  $z$  a generátor bodů krychle pro testy přesnosti a opakovatelnosti *Laser trackeru*. Tyto programy v sobě již obsahují offsety souřadnicového systému pro správný začátek programu v prostoru.

Blok Shapery slouží k realizaci přechodových S křivek, při přejezdu z počáteční konfigurace na začátek programu. Inverzní kinematika počítá natočení konkrétních os robota a vstupem pro ni jsou globální souřadnice trajektorie robota. Oba tyto bloky mají pro správný výpočet přiveden offset *Delka griperu*. Zde je nutné zadat správnou velikost griperu v podélné ose úchopné hlavice.

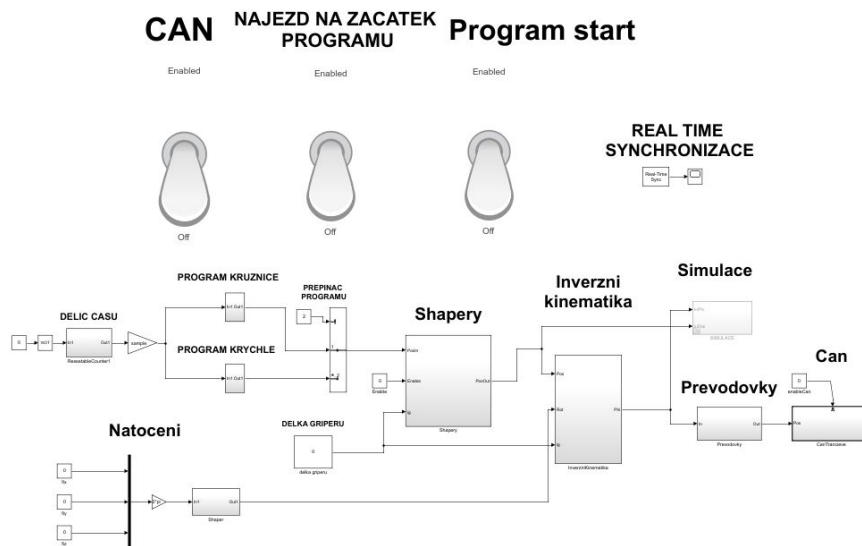
Natočení úchopné hlavice je možné změnit v sekci *Natoceni*, ale pro testování byla orientace vždy v konstantní  $[0 \ 1 \ 0]$ . S natáčením koncového efektoru je počítáno do budoucna.

Blok převodovky upravuje signál pro jednotlivé osy přenásobením konstantami, které přesně odpovídají převodovým poměrům konkrétních os. Více v kapitole identifikace.

Blok *CAN* je samotný vysílač, který kóduje signály do zpráv pro MCU1 a MCU2. Velmi důležitým blokem je *Real-Time Sync*, který zajišťuje zpomalování simulace tak, aby probíhala co nejlépe reálnému času.

Toto schéma je možné použít i pro simulaci a to při odpojení CAN bloku a zapojení bloku *Simulace*.

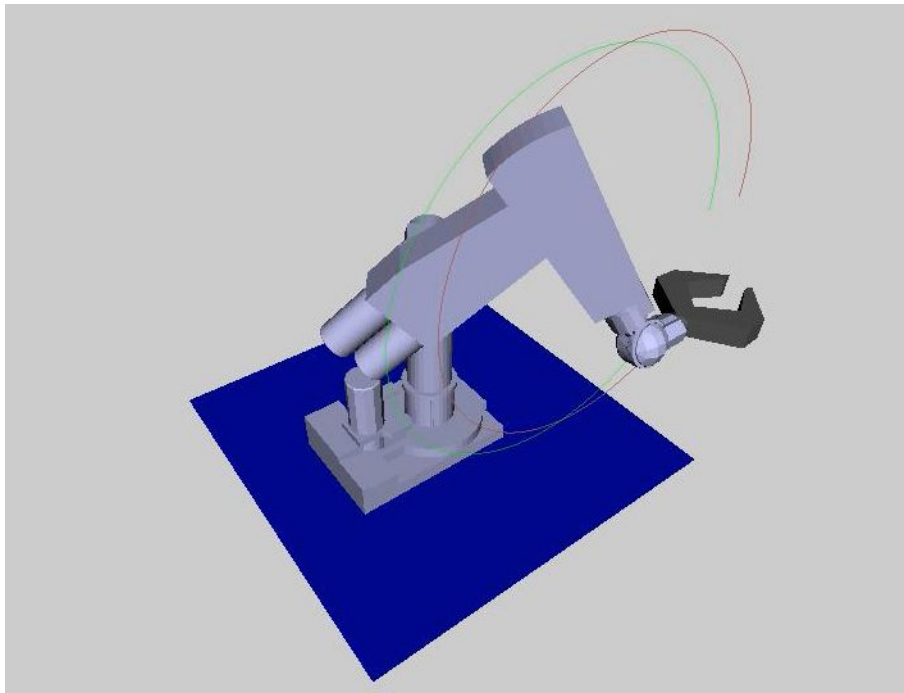
Schéma je uvedeno v příložené dokumentaci.



Obrázek 3.3: Schéma vysílač v Simulinku.

### 3.3 Virtuální model robota PUMA 200

Pro ověření trajektorie a funkčnosti inverzní kinematiky bylo v raných fázích vývoje nutné, aby bylo vše nejdříve vyzkoušeno na virtuálním modelu, který simuluje pohyb skutečného robota. Velkou výhodou je ochrana skutečného robota před fatální chybou a rychlé zjišťování problému oproti přidruženým problémům při testování na Hardware.



Obrázek 3.4: Virtuální model v Simulinku.

Tento model má konkrétní komponenty navrženy nejprve v programu *Autodesk Inventor*. Jejich spojení a sestavení proběhlo v simulačním prostředí Simulink *Virtual Reality Toolbox*. Zde byla otestována především správnost kinematického řešení. Na modelu je dále možné simulovat přejezdy do jiných řešení inverzní kinematiky a testování singularit kinematických dvojic. Model je také přiložen v příloze této práce a je vidět na Obr. 3.4.

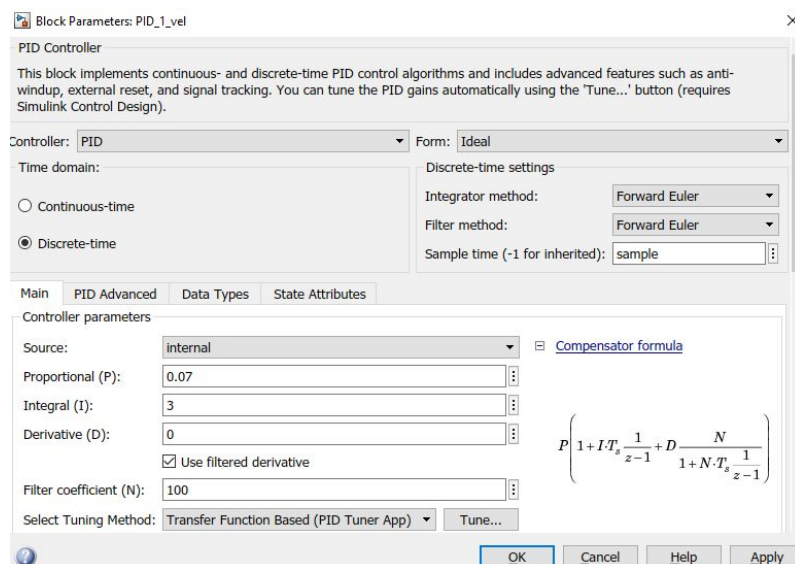
### 3.4 Naladění PID regulátorů

Pro správnou funkci řízení je nutné zpětnovazebnou smyčku opatřit PID regulátory, které zajistí realizaci servořízení. Tento dlouhodobě používaný druh regulátorů je složen ze tří větví, každá má svoji specifickou funkci.

- Složka *P* proporčně zesílí odchylku zpětné vazby (reakce na aktuální stav).
- Složka *I* integruje odchylku zpětné vazby (reakce na historii).
- Složka *D* reaguje na změnu odchylky (predikce).

V případě řízení rychlostní a polohové smyčky jsou použity pouze *PI* větve, které zajistí správné zesílení a eliminaci chyby. Složka *D* zde není použita.

Pro naladění *PI* regulátoru byl aplikován *Ziegler-Nichols* algoritmus [25]. Tedy metoda, která slouží pro základní naladění systému. Dané nastavení v Simulinku je vidět na Obr. 3.5.



Obrázek 3.5: Nastavení PID v Simulinku

Ladění PID regulátoru probíhá následovně: nejprve je velmi opatrně zvyšováno zesílení  $K_u$  než dojde k rozvibrování systému.

Z této hodnoty  $K_u$  se vypočte 45% a tato hodnota je použita jako zesílení  $P$ . Pro složku  $I$  se odečte z grafu odezvy systému délka periody  $T_u$ , tedy časová konstanta. Ta je vydělena hodnotou 1,2. Tato výsledná hodnota je invertována a použita jako zesílení  $I$  [25].

Tedy:  $P = 0,45 \times K_u$  a  $I = T_u/1,2$

Tento způsob opakujeme i pro polohovou smyčku.



# Kapitola 4

## Kinematika

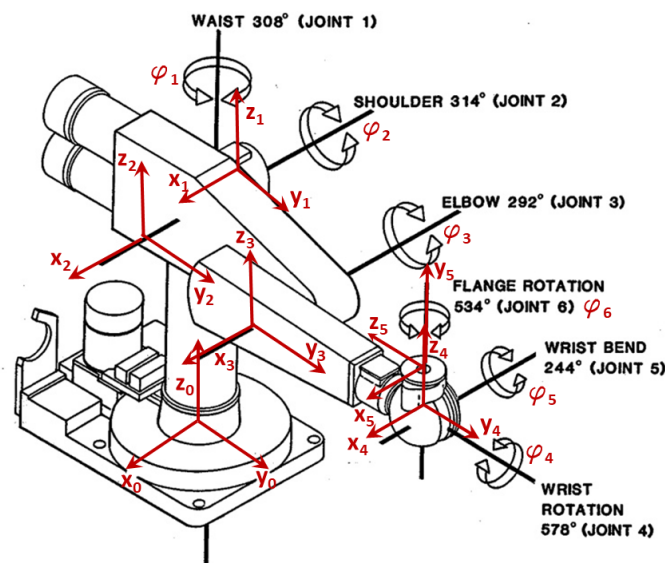
Tato kapitola popisuje kinematický popis robota pro jeho řízení v prostoru.

Kinematický popis robota PUMA 200 lze provést syntetickou geometrií, vektorovou metodou či maticovou metodou. Vzhledem k prostorovému rozložení robota bude použit maticový popis [20], který je výhodný pro 3D popis.

Jde o robota se sériovou kinematickou strukturou. Proto je základem řešení jeho kinematických vlastností popis dopředné kinematické transformace.

### 4.1 Popis robota dopřednou kinematikou

Nejprve je nutné zavést souřadnicové systémy pro popis robota. Na Obr. 4.2 jsou zakresleny souřadnicové systémy každého tělesa umístěné v jednotlivých kinematických dvojicích jednotlivých těles robota.



Obrázek 4.1: Souřadnicové systémy a úhly natočení [1].

Od počátku souřadnicového systému do koncového efektoru (Gripper) je popis realizován výslednou transformační maticí  $\mathbf{T}_{1G}$ , která vznikne násobením dílčích transformačních matic. (4.1).

Každý posuv a natočení je popsán základní příslušnou transformační maticí. V tomto konkrétním případě jsou posunutí - konstantní vzdáleností a natočení - úhlů popisující natočení v kloubech robota. Právě úhly natočení konkrétní osy je žádaný výsledek, kterého je třeba dosáhnout. Z Obr. 4.2 postupně odečteme a dostaneme výslednou transformační matice ( $\mathbf{T}_{1G}$ ) popisující robota přes jeho strukturu [22].

$$\begin{aligned} \mathbf{T}_{1G} = & \mathbf{T}_z(l_1)\mathbf{T}_{\varphi_z}(\varphi_1)\mathbf{T}_x(l_2)\mathbf{T}_{\varphi_x}(\varphi_2)\mathbf{T}_y(l_3)\mathbf{T}_x(l_4)\mathbf{T}_{\varphi_x}(\varphi_3)\mathbf{T}_y(l_5) \\ & \mathbf{T}_{\varphi_y}(\varphi_4)\mathbf{T}_y(l_6)\mathbf{T}_{\varphi_x}(\varphi_5)\mathbf{T}_y(l_7)\mathbf{T}_{\varphi_y}(\varphi_6)\mathbf{T}_x(g_1)\mathbf{T}_y(g_2)\mathbf{T}_z(g_3), \end{aligned} \quad (4.1)$$

kde  $\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6$  jsou úhly rotace v rotačních kloubech robota a  $l_1, l_2, l_3, l_4, l_5, l_6, l_7, g_1, g_2, g_3$  jsou rozměrové konstanty mezi souřadnicovými systémy (geometrické parametry robota).

Takto je popsána dopředná kinematika.

## 4.2 Inverzní Kinematika

Inverzní kinematika popisuje natočení kloubů robota z jeho požadovaného pohybu.

### 4.2.1 Popis pohybu robota

Požadovaný pohyb je popsán fyzikálními souřadnicemi žádaného pohybu koncového efektoru daného posuvem  $x_M, y_M, z_M$  a natočením popsáním *Cardanovými úhly* [20]:

$$\mathbf{T}_{1M} = \mathbf{T}_x(x_M)\mathbf{T}_y(y_M)\mathbf{T}_z(z_M)\mathbf{T}_{\varphi_x}(\varphi_x)\mathbf{T}_{\varphi_y}(\varphi_y)\mathbf{T}_{\varphi_z}(\varphi_z) = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_1 \\ s_{21} & s_{22} & s_{23} & s_2 \\ s_{31} & s_{32} & s_{33} & s_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.2)$$

kde  $s_{ij}$  a  $s_k$  jsou prvky obecné transformační matice.

$$\begin{aligned}
s_{11} &= \cos(\varphi_y) \cos(\varphi_z) \\
s_{12} &= -\cos(\varphi_y) \sin(\varphi_z) \\
s_{13} &= \sin(\varphi_y) \\
s_1 &= x_M \\
s_{21} &= \cos(\varphi_x) \sin(\varphi_z) + \cos(\varphi_z) \sin(\varphi_y) \sin(\varphi_x) \\
s_{22} &= \cos(\varphi_x) \cos(\varphi_z) - \sin(\varphi_y) \sin(\varphi_x) \sin(\varphi_z) \\
s_{23} &= -\cos(\varphi_y) \sin(\varphi_x) \\
s_2 &= y_M \\
s_{31} &= \sin(\varphi_x) \sin(\varphi_z) - \cos(\varphi_x) \cos(\varphi_z) \sin(\varphi_y) \\
s_{32} &= \cos(\varphi_z) \sin(\varphi_x) + \cos(\varphi_x) \sin(\varphi_y) \sin(\varphi_z) \\
s_{33} &= \cos(\varphi_y) \cos(\varphi_x) \\
s_3 &= z_M,
\end{aligned} \tag{4.3}$$

kde  $\varphi_x, \varphi_y, \varphi_z$  jsou *Cardanovy úhly* a  $x_M, y_M, z_M$  jsou souřadnice pro zadaný bod v prostoru, kde chceme mít konečný efektor.

Pohyb robota je popsán rovností polohy konce efektoru a jeho požadovaného pohybu

$$\mathbf{T}_{1G} = \mathbf{T}_{1M}. \tag{4.4}$$

Rovnice pro  $\mathbf{T}_{1G}$  lze dále upravit a konstanty  $l_i$  sečíst, takže vyjde následující transformace.

$$\begin{aligned}
\mathbf{T}_{1G} &= \mathbf{T}_z(l_1) \mathbf{T}_{\varphi_z}(\varphi_1) \mathbf{T}_x(l_{24}) \mathbf{T}_{\varphi_x}(\varphi_2) \mathbf{T}_y(l_3) \mathbf{T}_{\varphi_x}(\varphi_3) \mathbf{T}_y(l_{56}) \\
&\quad \mathbf{T}_{\varphi_y}(\varphi_4) \mathbf{T}_{\varphi_x}(\varphi_5) \mathbf{T}_{\varphi_y}(\varphi_6) \mathbf{T}_x(g_1) \mathbf{T}_y(g_{27}) \mathbf{T}_z(g_3)
\end{aligned} \tag{4.5}$$

Tyto úpravy vychází ze záměnnosti násobení transformačních matic posuvů a rotace a posuvů kolem téže osy, kde

$$\begin{aligned}
l_{24} &= l_2 + l_4 \\
l_{56} &= l_5 + l_6 \\
g_{27} &= g_2 + l_7.
\end{aligned} \tag{4.6}$$

## 4.2.2 Způsoby řešení inverzní kinematiky

Aby bylo možno přesně předepsat bod a natočení koncového efektoru (úchopná hlavice, obráběcí vřeteno), je možných několik postupů.

Tyto postupy se odvíjejí podle uspořádání robota. Pokud jde o tzv. jednoduché uspořádání, je řešitelné analyticky (simple robots). Pokud jde o jiné uspořádání, jedná se o tzv. ne jednoduché roboty (non simple robots) a nelze je řešit analyticky [20].

- Analytické řešení - popis transformačními maticemi nebo geometricky (neprůhledné řešení) [20].
- Tradiční numerické řešení - Newtonova metoda [20].
- Metoda strukturní aproximace [21], [22].
- Metoda diferenciální kinematiky [23].

Skupina tzv. jednoduchých robotů má své analytické řešení a tudíž řešení je s garantovanou přesností a v garantovaném čase výpočtu (přesné a rychlé řešení). Proto již při návrhu robota musí být uvažováno, jak se bude kinematika řešit. To je také důvodem, proč je většina sériových robotů velmi podobná. Jde o první tři rotační osy a sférické zápěstí.

Toto uspořádání umožňuje kinematický řetězec rozříznout právě ve sférické vazbě, rozdělit ji na posuv do sférické vazby a na orientaci kolem ní, a vyřešit rovnice analyticky. Jinak by podobné řešení možné nebylo.

### 4.2.3 Analytické řešení inverzní kinematiky robota PUMA

Řešení je založeno na existenci sférické kinematické dvojice ve struktuře robota. Rovnici (4.4) upravíme tak, aby jednak kinematická transformace robota končila sférickým kloubem a jednak úhel prvního kloubu převedeme na pravou stranu.

Zde je popis takového rozdělení na levou a pravou stranu:

$$\mathbf{T}_{1L} = \mathbf{T}_{1R}, \quad (4.7)$$

kde

$$\mathbf{T}_{1L} = \mathbf{T}_x(l_{24})\mathbf{T}_{\varphi_x}(\varphi_2)\mathbf{T}_y(l_3)\mathbf{T}_{\varphi_x}(\varphi_3)\mathbf{T}_y(l_{56})\mathbf{T}_{\varphi_y}(\varphi_4)\mathbf{T}_{\varphi_x}(\varphi_5)\mathbf{T}_{\varphi_y}(\varphi_6) \quad (4.8)$$

a

$$\mathbf{T}_{1R} = \mathbf{T}_z(-l_1)\mathbf{T}_{\varphi_z}(-\varphi_1)\mathbf{T}_{1M}\mathbf{T}_z(-g_3)\mathbf{T}_y(-g_{27})\mathbf{T}_x(-g_1). \quad (4.9)$$

V místě řezu sférické dvojice musí být radius vektory sobě rovné. Rovnice (4.8) a (4.9) lze zapsat

$$\mathbf{T}_{1L} = \begin{bmatrix} S_{1L} & r_{1L} \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{T}_{1R} = \begin{bmatrix} S_{1R} & r_{1R} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.10)$$

Ze shody

$$\mathbf{r}_{1L} = \mathbf{r}_{1R}, \quad (4.11)$$

kde

$$\mathbf{r}_{1L} = \begin{bmatrix} \mathbf{T}_{L1,4} \\ \mathbf{T}_{L2,4} \\ \mathbf{T}_{L3,4} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_{R1,4} \\ \mathbf{T}_{R2,4} \\ \mathbf{T}_{R3,4} \end{bmatrix} = \mathbf{r}_{1R}. \quad (4.12)$$

Dostaneme tři rovnice pro tři úhly  $\varphi_1, \varphi_2, \varphi_3$ , které jsou pak analyticky řešitelné [20].

$$\begin{aligned}\mathbf{T}_{L1,4} &= l_2 + l_4 \\ \mathbf{T}_{L2,4} &= l_3 \cos(\varphi_2) - l_{56}(\sin(\varphi_2) \sin(\varphi_3) - \cos(\varphi_2) \cos(\varphi_3)) \\ \mathbf{T}_{L3,4} &= l_{56}(\cos(\varphi_2) \sin(\varphi_3) + \cos(\varphi_3) \sin(\varphi_2)) + l_3 \sin(\varphi_2)\end{aligned}\quad (4.13)$$

$$\begin{aligned}\mathbf{T}_{R1,4} &= s_1 \cos(\varphi_1) - g_3(s_{13} \cos(\varphi_1) + s_{23} \sin(\varphi_1)) - g_{27}(s_{12} \cos(\varphi_1) \\ &\quad + s_{22} \sin(\varphi_1)) - g_1(s_{11} \cos(\varphi_1) + s_{21} \sin(\varphi_1)) + s_2 \sin(\varphi_1) \\ \mathbf{T}_{R2,4} &= s_2 \cos(\varphi_1) - g_3(s_{23} \cos(\varphi_1) - s_{13} \sin(\varphi_1)) - g_{27}(s_{22} \cos(\varphi_1) \\ &\quad - s_{12} \sin(\varphi_1)) - g_1(s_{21} \cos(\varphi_1) - s_{11} \sin(\varphi_1)) - s_1 \sin(\varphi_1) \\ \mathbf{T}_{R3,4} &= s_3 - l_1 - g_1 s_{31} - g_3 s_{33} - g_{27} s_{32}\end{aligned}\quad (4.14)$$

Z rovnosti

$$\mathbf{T}_{L1,4} = \mathbf{T}_{R1,4}\quad (4.15)$$

dostaneme

$$A_1 \cos(\varphi_1) + B_1 \sin(\varphi_1) = C_1,\quad (4.16)$$

kde

$$\begin{aligned}A_1 &= s_1 - g_1 s_{11} - g_{27} s_{12} - g_3 s_{13} \\ B_1 &= s_2 - s_{22} g_{27} - g_3 s_{23} - g_1 s_{21} \\ C_1 &= l_2 + l_4.\end{aligned}\quad (4.17)$$

Dále platí:

$$\sin(\varphi_1)^2 + \cos(\varphi_1)^2 = 1\quad (4.18)$$

Z rovnic (4.16), (4.17) a (4.18) dostaneme kvadratickou rovnici pro  $\sin(\varphi_1)$ . Pak z (4.16) určíme  $\cos(\varphi_1)$

$$\begin{aligned}\sin(\varphi_1) &= \frac{B_1 C_1 + \sqrt{B_1^2 C_1^2 - (A_1^2 + B_1^2)(C_1^2 - A_1^2)}}{A_1^2 + B_1^2} \\ \cos(\varphi_1) &= \frac{C_1 - B_1 \sin(\varphi_1)}{A_1}\end{aligned}\quad (4.19)$$

a řešení dokončíme funkcí *atan2*

$$\varphi_1 = \text{atan2}(\sin(\varphi_1), \cos(\varphi_1)).\quad (4.20)$$

V rovnici (4.18) bylo zvoleno znaménko odpovídající konfiguraci RIGHT and ABOVE z Obr. 4.2. Rovnice  $\mathbf{T}_{L2,4}$  a  $\mathbf{T}_{L3,4}$  upravíme a sečteme jejich druhé mocniny.

$$\begin{aligned}\mathbf{T}_{L2,4} &= l_3 \cos(\varphi_2) + l_{56} \cos(\varphi_2 + \varphi_3) \\ \mathbf{T}_{L3,4} &= l_3 \sin(\varphi_2) + l_{56} \sin(\varphi_2 + \varphi_3) \\ \mathbf{T}_{L2,4}^2 + \mathbf{T}_{L3,4}^2 &= (2l_3 l_{56}) \cos(\varphi_3) + l_3^2 + l_{56}^2\end{aligned}\quad (4.21)$$

Odtud

$$\begin{aligned}\cos(\varphi_3) &= \frac{\mathbf{T}_{R2,4}^2 + \mathbf{T}_{R3,4}^2 - l_3^2 - l_{56}^2}{2l_3l_{56}} \\ \sin(\varphi_3) &= \sqrt{1 - \cos(\varphi_3)^2}\end{aligned}\quad (4.22)$$

a řešení dokončíme

$$\varphi_3 = \text{atan2}(\sin(\varphi_3), \cos(\varphi_3)). \quad (4.23)$$

Volba znaménka v (4.22) je pro konfiguraci RIGHT and ABOVE z Obr. 4.2. Konečně rovnice (4.21) jsou dvě lineární algebraické rovnice pro  $\cos(\varphi_2)$  a  $\sin(\varphi_2)$ .

$$\begin{aligned}Det_{\varphi_2} &= l_3^2 + l_{56}^2 + 2l_3l_{56}\cos(\varphi_3) \\ \cos(\varphi_2) &= \frac{\mathbf{T}_{RR2,4}(l_3 + l_{56}\cos(\varphi_3)) + \mathbf{T}_{RR3,4}l_{56}\sin(\varphi_3)}{Det_{\varphi_2}} \\ \sin(\varphi_2) &= \frac{\mathbf{T}_{RR3,4}(l_3 + l_{56}\cos(\varphi_3)) - \mathbf{T}_{RR2,4}l_{56}\sin(\varphi_3)}{Det_{\varphi_2}}\end{aligned}\quad (4.24)$$

$$\varphi_2 = \text{atan2}(\sin(\varphi_2), \cos(\varphi_2)) \quad (4.25)$$

Známe-li první tři úhly  $\varphi_1, \varphi_2, \varphi_3$ , můžeme určit z rovnosti matic směrových kosínů zbývající úhly  $\varphi_4, \varphi_5, \varphi_6$ . Pro užití rovnosti  $\mathbf{S}_{1L} = \mathbf{S}_{1R}$  rovnice (4.7) upravíme na

$$\mathbf{T}_{1LL} = \mathbf{T}_{1RR}, \quad (4.26)$$

kde

$$\begin{aligned}\mathbf{T}_{1LL} &= \mathbf{T}_x(l_{24})\mathbf{T}_{\varphi_x}(\varphi_2)\mathbf{T}_y(l_3)\mathbf{T}_{\varphi_x}(\varphi_3)\mathbf{T}_y(l_{56})\mathbf{T}_{\varphi_y}(\varphi_4)\mathbf{T}_{\varphi_x}(\varphi_5) \\ \mathbf{T}_{1RR} &= \mathbf{T}_z(-l_1)\mathbf{T}_{\varphi_z}(-\varphi_1)\mathbf{T}_{1M}\mathbf{T}_z(-g_3)\mathbf{T}_y(-g_{27})\mathbf{T}_x(-g_1)\mathbf{T}_{\varphi_y}(-\varphi_6),\end{aligned}\quad (4.27)$$

kde

$$\begin{aligned}\mathbf{T}_{LL1,1} &= \cos(\varphi_4) \\ \mathbf{T}_{LL1,2} &= \sin(\varphi_4)\sin(\varphi_5) \\ \mathbf{T}_{LL1,3} &= \cos(\varphi_5)\sin(\varphi_4).\end{aligned}\quad (4.28)$$

$$\begin{aligned}\mathbf{T}_{LL2,1} &= \sin(\varphi_4)(\cos(\varphi_2)\sin(\varphi_3) + \cos(\varphi_3)\sin(\varphi_2)) \\ \mathbf{T}_{LL2,2} &= -\cos(\varphi_5)(\sin(\varphi_2)\sin(\varphi_3) - \cos(\varphi_2)\cos(\varphi_3)) \\ &\quad - \cos(\varphi_4)\sin(\varphi_5)(\cos(\varphi_2)\sin(\varphi_3) + \cos(\varphi_3)\sin(\varphi_2)) \\ \mathbf{T}_{LL2,3} &= \sin(\varphi_5)(\sin(\varphi_2)\sin(\varphi_3) - \cos(\varphi_2)\cos(\varphi_3)) \\ &\quad - \cos(\varphi_4)\cos(\varphi_5)(\cos(\varphi_2)\sin(\varphi_3) + \cos(\varphi_3)\sin(\varphi_2))\end{aligned}\quad (4.29)$$

$$\begin{aligned}
\mathbf{T}_{LL3,1} &= \sin(\varphi_4)(\sin(\varphi_2) \sin(\varphi_3) - \cos(\varphi_2) \cos(\varphi_3)) \\
\mathbf{T}_{LL3,2} &= \cos(\varphi_5)(\cos(\varphi_2) \sin(\varphi_3) + \cos(\varphi_3) \sin(\varphi_2)) \\
&\quad - \cos(\varphi_4) \sin(\varphi_5)(\sin(\varphi_2) \sin(\varphi_3) - \cos(\varphi_2) \cos(\varphi_3)) \\
\mathbf{T}_{LL3,3} &= -\sin(\varphi_5)(\cos(\varphi_2) \sin(\varphi_3) + \cos(\varphi_3) \sin(\varphi_2)) \\
&\quad - \cos(\varphi_4) \cos(\varphi_5)(\sin(\varphi_2) \sin(\varphi_3) - \cos(\varphi_2) \cos(\varphi_3))
\end{aligned} \tag{4.30}$$

$$\begin{aligned}
\mathbf{T}_{RR1,1} &= \cos(\varphi_6)(s_{11} \cos(\varphi_1) + s_{21} \sin(\varphi_1)) + \sin(\varphi_6)(s_{13} \cos(\varphi_1) + s_{23} \sin(\varphi_1)) \\
\mathbf{T}_{RR2,1} &= \cos(\varphi_6)(s_{21} \cos(\varphi_1) - s_{11} \sin(\varphi_1)) + \sin(\varphi_6)(s_{23} \cos(\varphi_1) - s_{13} \sin(\varphi_1)) \\
\mathbf{T}_{RR3,1} &= s_{31} \cos(\varphi_6) + s_{33} \sin(\varphi_6)
\end{aligned} \tag{4.31}$$

$$\begin{aligned}
\mathbf{T}_{RR1,2} &= s_{12} \cos(\varphi_1) + s_{22} \sin(\varphi_1) \\
\mathbf{T}_{RR2,2} &= s_{22} \cos(\varphi_1) - s_{12} \sin(\varphi_1) \\
\mathbf{T}_{RR3,2} &= s_{32}
\end{aligned} \tag{4.32}$$

$$\begin{aligned}
\mathbf{T}_{RR1,3} &= \cos(\varphi_6)(s_{13} \cos(\varphi_1) + s_{23} \sin(\varphi_1)) \\
&\quad - \sin(\varphi_6)(s_{11} \cos(\varphi_1) + s_{21} \sin(\varphi_1)) \\
\mathbf{T}_{RR2,3} &= \cos(\varphi_6)(s_{23} \cos(\varphi_1) - s_{13} \sin(\varphi_1)) \\
&\quad - \sin(\varphi_6)(s_{21} \cos(\varphi_1) - s_{11} \sin(\varphi_1)) \\
\mathbf{T}_{RR3,3} &= s_{33} \cos(\varphi_6) - s_{31} \sin(\varphi_6)
\end{aligned} \tag{4.33}$$

Označíme:

$$\begin{aligned}
\cos(\varphi_{23}) &= \cos(\varphi_2 + \varphi_3) \\
\sin(\varphi_{23}) &= \sin(\varphi_2 + \varphi_3)
\end{aligned} \tag{4.34}$$

Z rovnosti  $\mathbf{T}_{LL2,1} = \mathbf{T}_{RR2,1}$  a  $\mathbf{T}_{LL3,1} = \mathbf{T}_{RR3,1}$  dostaneme

$$\begin{aligned}
AA_1 &= (s_{21} \cos(\varphi_1) - s_{11} \sin(\varphi_1)) \cos(\varphi_{23}) \\
BB_1 &= (s_{23} \cos(\varphi_1) - s_{13} \sin(\varphi_1)) \cos(\varphi_{23}) \\
AA_2 &= -s_{31} \sin(\varphi_{23}) \\
BB_2 &= -s_{33} \sin(\varphi_{23}).
\end{aligned} \tag{4.35}$$

$$\varphi_6 = \text{atan2}(AA_1 + AA_2, BB_2 + BB_1) \tag{4.36}$$

Z rovnosti  $\mathbf{T}_{LL1,1} = \mathbf{T}_{RR1,1}$  a  $\mathbf{T}_{LL3,1} = \mathbf{T}_{RR3,1}$  dostaneme

$$\mathbf{T}_{RR1,1} = \cos(\varphi_6)(s_{11} \cos(\varphi_1) + s_{21} \sin(\varphi_1)) + \sin(\varphi_6)(s_{13} \cos(\varphi_1) + s_{23} \sin(\varphi_1)). \tag{4.37}$$

$$\begin{aligned}
\mathbf{T}_{LL1,1} &= \cos(\varphi_4) \\
\sin(\varphi_4) &= (\sqrt{1 - \cos(\varphi_4)^2})
\end{aligned} \tag{4.38}$$

$$\varphi_4 = \text{atan2}(\sin(\varphi_4), \cos(\varphi_4)) \quad (4.39)$$

A konečně z rovnosti

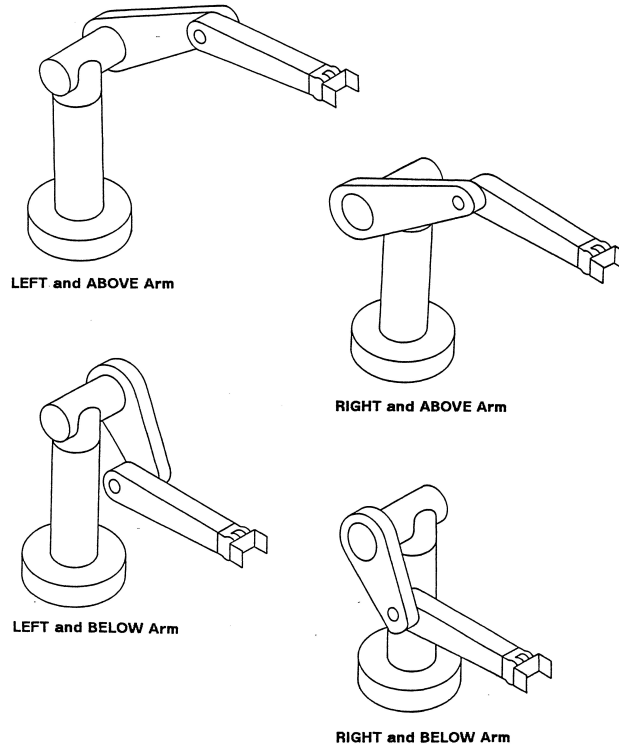
$$\begin{aligned} \mathbf{T}_{RR2,2} &= s_{22} \cos(\varphi_1) - s_{12} \sin(\varphi_1) \\ \mathbf{T}_{RR2,3} &= \cos(\varphi_6)(s_{23} \cos(\varphi_1) - s_{13} \sin(\varphi_1)) - \sin(\varphi_6)(s_{21} \cos(\varphi_1) - s_{11} \sin(\varphi_1)) \end{aligned} \quad (4.40)$$

dostaneme

$$\begin{aligned} \cos(\varphi_5) &= \frac{-\mathbf{T}_{RR2,2} \cos(\varphi_{23}) + \mathbf{T}_{RR2,3} \cos(\varphi_4) \sin(\varphi_{23})}{\text{Det}_{\varphi_5}} \\ \sin(\varphi_5) &= \frac{\mathbf{T}_{RR2,3} \cos(\varphi_{23}) + \mathbf{T}_{RR2,2} \cos(\varphi_4) \sin(\varphi_{23})}{\text{Det}_{\varphi_5}} \end{aligned} \quad (4.41)$$

$$\varphi_5 = \text{atan2}(\sin(\varphi_5), \cos(\varphi_5)) \quad (4.42)$$

Takto je vyřešena inverzní kinematika. Každý z úhlů  $\varphi_i$  má několik řešení, která odpovídají různé konfiguraci robota. Proto je nutné při každé změně konfigurace (i změna kvadrantu), přepínat mezi různými řešeními viz Obr. [24].



Obrázek 4.2: Konfigurace natočení robota [24].



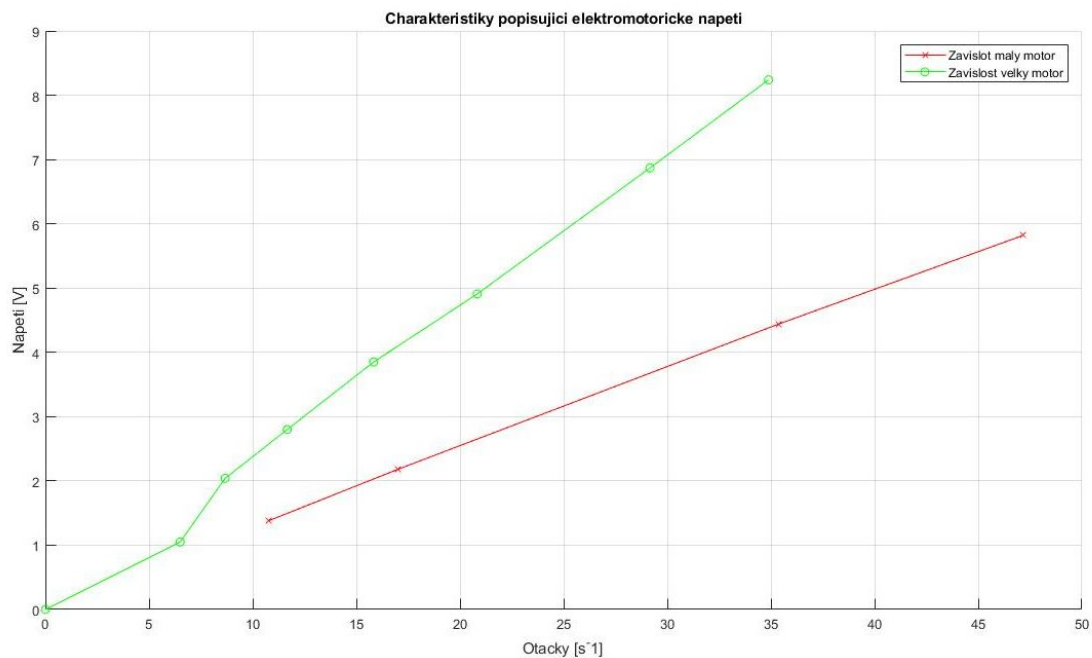
# Kapitola 5

## Identifikace robota PUMA 200

V této kapitole bude vysvětlena identifikace komponent robota.

### 5.1 Identifikace motorů

V první fázi identifikace robota bylo nutné identifikovat motory. K motorům není bohužel dostupná skoro žádná dokumentace, proto byly jejich základní parametry, napětí a proud, kvalifikovaně odhadnuty na základě podobnosti s jinými motory a dle dostupné dokumentace. Dále změřeného ohmického odporu. Také bylo změřeno zpětné elektromotorické napětí na malém a velkém motoru viz. Obr. 5.1.

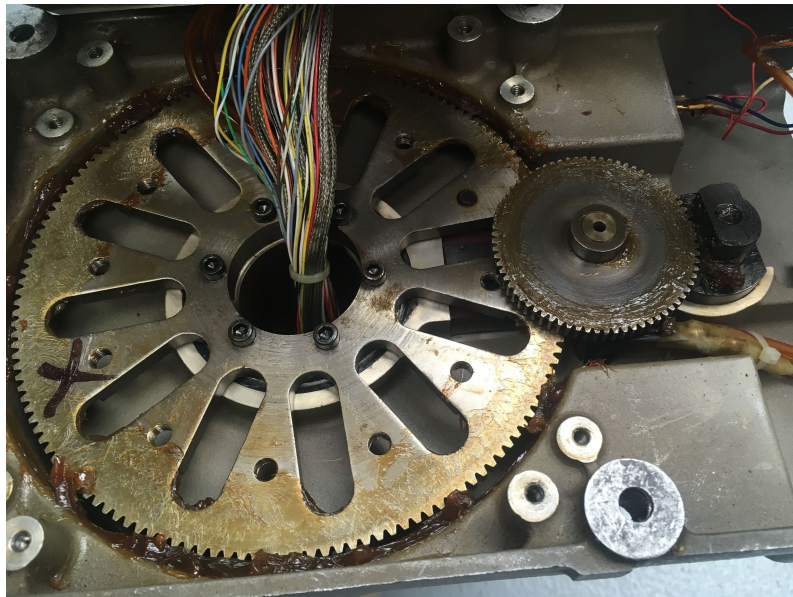


Obrázek 5.1: Závislost elektromotorického napětí experimentálně změřená.

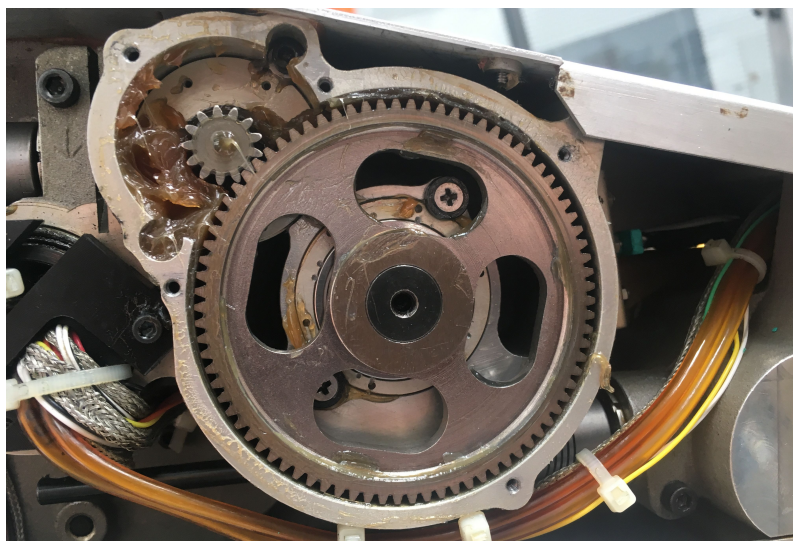
## 5.2 Identifikace převodových poměrů

Pro správnou funkci robota je také nutné určit přesně převodové poměry konkrétních os. Tato měření by bylo možné provést dle počtu pulzů na encoderech při pootočení o dané natočení osy robota. Převodové soukolí bylo vhodné také promazat a seřadit a proto byly převodové poměry určeny přímo jejich rozebráním a určením počtu zubů.

Každé převodové soukolí má jedno excentrické uložení pro vymezení vůlí. Díky tomu lze každou osu přesně nastavit, aby byl pohyb bez vůlí a tedy co nejpřesnější. Na Obr. 5.2 a 5.3 je ukázka dvou převodových soukolí. Přesný převodový poměr je na Obr. 5.6.



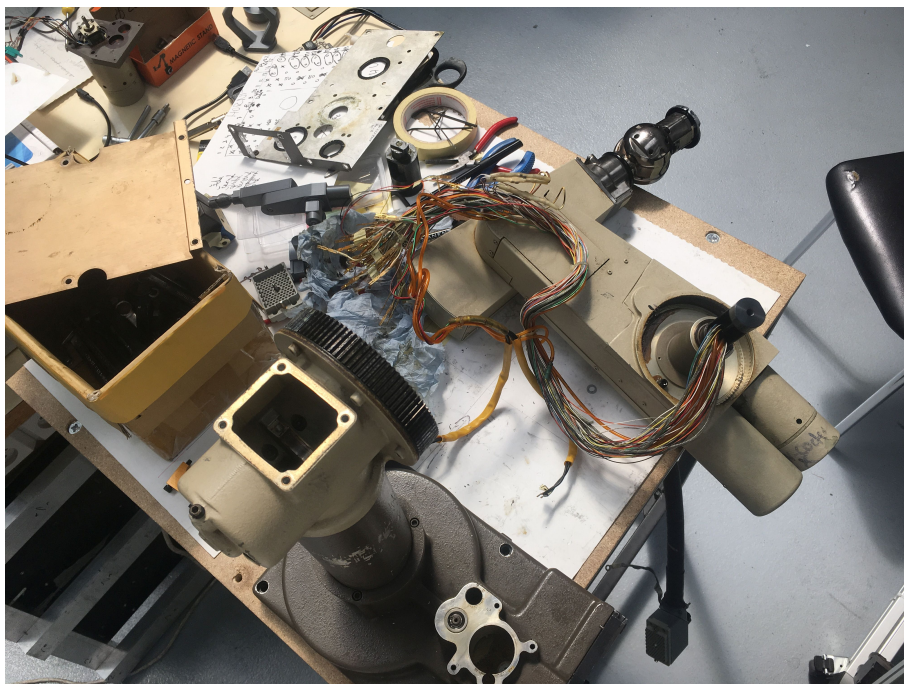
Obrázek 5.2: Převodové soukolí první osy.



Obrázek 5.3: Převodové soukolí druhé osy.

Po prvních testech robota byla zjištěna vůle především na první ose. Excentrické uložení je sice připraveno k vymezení vůle, avšak z neznámých důvodů bylo toto soukolí již dříve maximálně vymezeno a již nebylo možné tento stav zlepšit. Proto došlo k výměně celé součásti. Roboty PUMA 200 byly zakoupeny v páru a druhý poškozený kus měl tuto součást v lepším stavu.

Pro výměnu těchto součástí, muselo dojít k rozpojení konektoru, který, jak je dále ukázáno, má 120 pinů (Obr. 5.8). Dále bylo nutné vytažení všech kabelů a hydraulických rozvodů jak je ilustrováno na Obr. 5.4. Následně byla provedena kompletace robota a znovuzapojení konektoru. Problém s vůlí na první ose se vyřešil a vymezení vůlí na všech převodech bude úkolem dalšího seřízení robota.



Obrázek 5.4: Ukázka rozloženého robota.

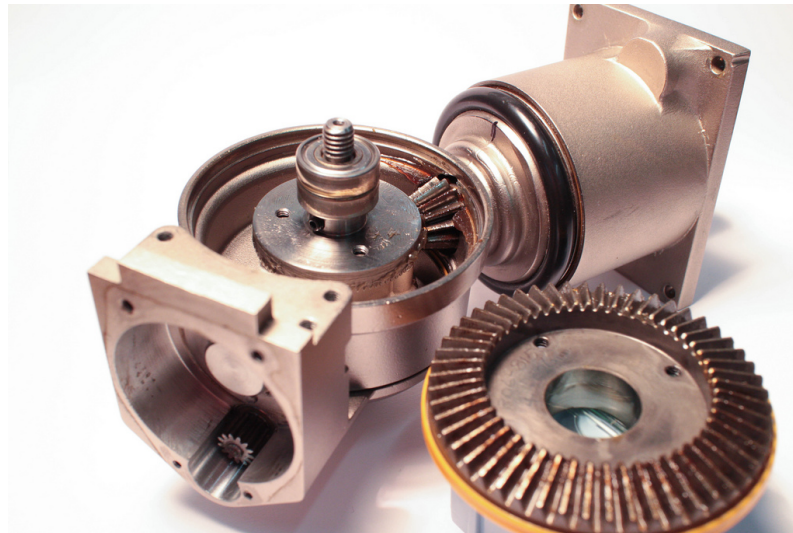
### 5.2.1 Sférické zápěstí

Speciálním prvkem je sférické zápěstí, které je konstruováno pro přenos třech různých rotačních pohybů přes jednu osu otáčení. Tím je dosaženo možnosti umístění pohonů na jedné straně zápěstí a rotačních os na druhé straně. Pro realizaci tohoto pohybu je pátá osa závislá na pohybu čtvrté a šestá osa závislá na páté i na čtvrté ose. Jak je ilustrováno na Obr. 5.6.

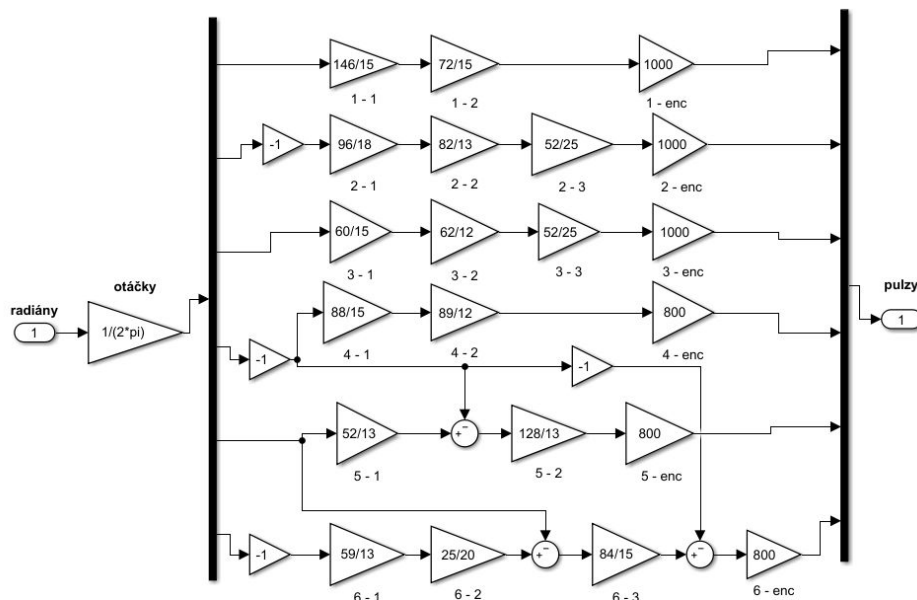
Částečně rozebrané sférické zápěstí je zobrazeno na Obr. 5.5.

Jak již bylo řečeno encodery motorů mají různý počet pulzů na otáčku, takže je nutné tento rozdíl vnést i do výpočtu polohy posílané do motoru. Zde na Obr. 5.6 jsou vidět všechny převodové poměry a poslední blok upravuje počet pulzů na otáčku dané osy.





Obrázek 5.5: Rozebrané sférické zápěstí.

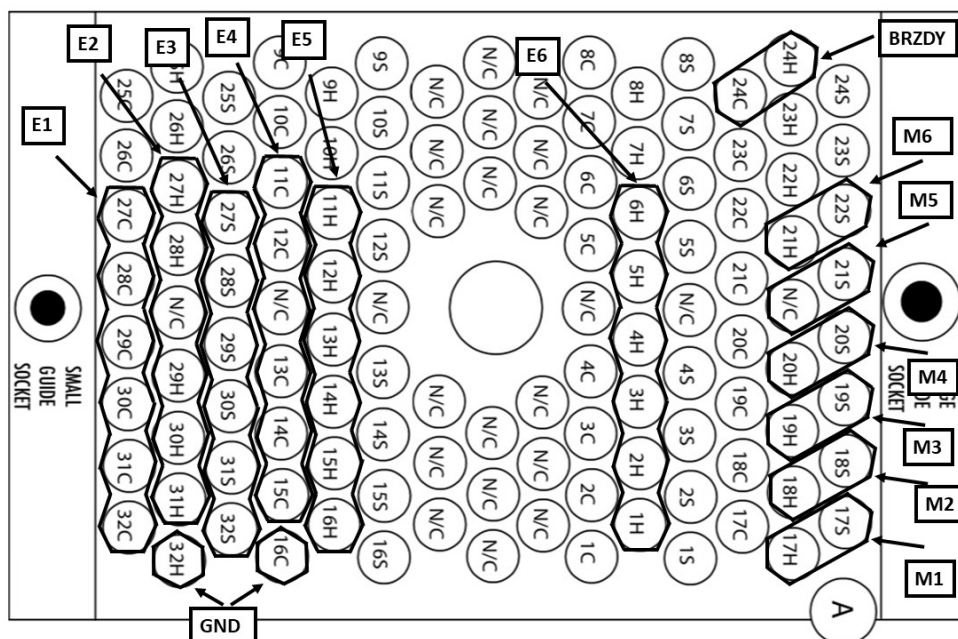


Obrázek 5.6: Schématické zobrazení převodových poměrů a přepočtu pulzů na encoderech.

### 5.3 Identifikace konektoru

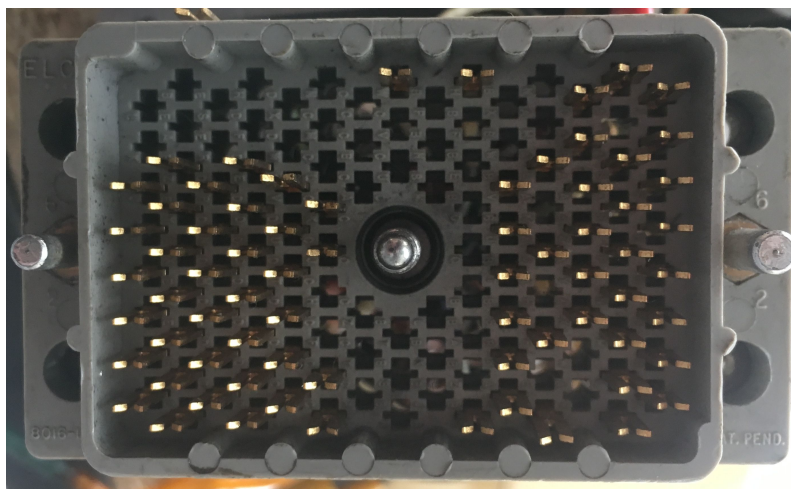
Pro připojení robota k řídicí jednotce je na zadní straně rámu robota konektor **ELCO/EDAC 8016-120-607/608** s možností připojení až 120 pinů. V případě tohoto robota bylo využito 36 pinů pro encodery, 12 pinů pro motory, 2 piny pro brzdy a 2 pro uzemnění. Zbylé piny mohou být použity pro ovládání pneumatiky úchopné hlavice a dalších funkcionalit.

Na Obr. 5.7 je vidět rozmístění konkrétních pinů. Značky  $E_i$  značí zapojení encoderů pro  $i$ -tý motor vždy v pořadí zhora *INDEX signál (27C)*, *GND (28C)*, *B signál (29C)*, *VDC (30C)*, *A signál (31C)*, *TEMP (32C)*, kde druhý kontakt teploměru je



Obrázek 5.7: Schéma konektoru.

připojen k GND pinu *32H* a *16C*. Brzdy jsou zapojeny pro všechny tři osy společně na *24H* a *24C*. Označení vinutí DC motorů je uvedeno na Obr. 5.7 pod  $M_i$ .



Obrázek 5.8: Skutečný konektor.

## 5.4 Identifikace konstant robota

V první fázi identifikace byly zjištěny rozměry robota dle manuálu [1]. Výrobce uvádí pouze konstanty *l1*, *l3* a *l56*. Zbylé konstanty, tedy *l2* a *g27*, musely být pro počáteční identifikaci změřeny experimentálně posuvným měřítkem. Jejich zpřesnění je úkolem kalibrace, která není z časových důvodů obsahem této práce.



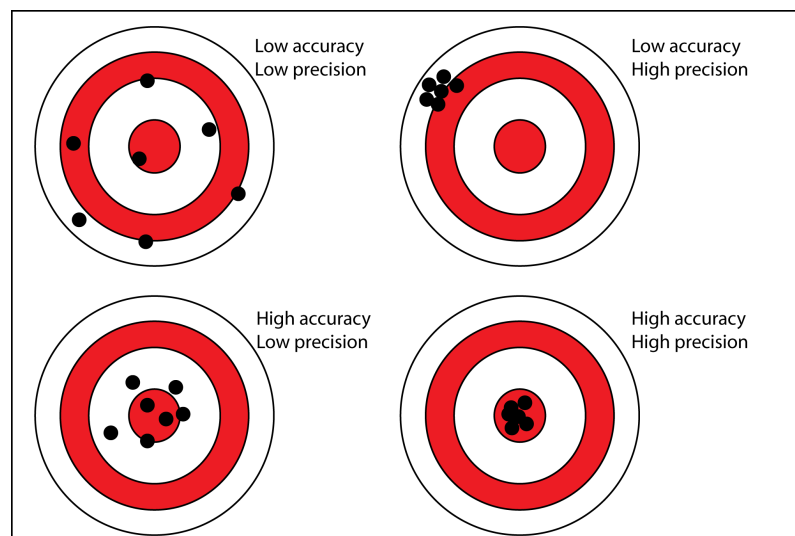
# Kapitola 6

## Kontrola přesnosti polohování robota pomocí laser trackeru

Tato kapitole popisuje ověření polohování pomocí Laser trackeru.

Pro lepší pochopení této kapitoly je nutné znát rozdíl mezi přesností a opakovatelností, které jsou tvoří jedny ze základních vlastností robota.

*Přesnost* popisuje odchylku dosaženého bodu v prostoru od skutečné polohy. Na druhou stranu *opakovatelnost* je odchylka naměřené hodnoty při vícenásobném naměření do jednoho bodu. Tato odlišnost je asi nejlépe ilustrována na Obr. 6.1



Obrázek 6.1: Rozdíl mezi přesností a opakovatelností [26].

### 6.1 Experimenty pro ověření přesnosti a opakovatelnosti robota

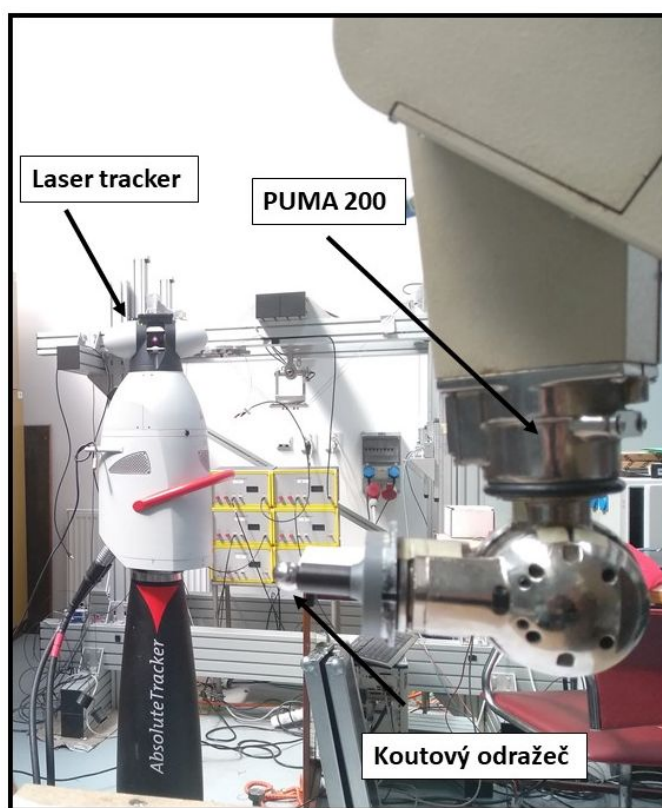
Pro ověření přesnosti a opakovatelnosti byla provedena řada experimentů.

### 6.1.1 Laser tracker

Laser tracker je zařízení určené k přesnému trojrozměrnému bezdotykovému měření pohyblivých předmětů v prostoru. Princip spočívá v kombinaci dvou technologií. Laserové interferometrie pro měření vzdálenosti a servosledování odraženého paprsku koutovým odražečem pro měření azimutu a elevace zrcátka ovládajícího laserový paprsek.

Velkou výhodou tohoto měřicího zařízení je kromě velké přesnosti především schopnost bezkontaktně měřit polohu bodu v prostoru (koutového odražeče), který se může pohybovat s měřeným tělesem [26].

Pro vyhodnocení přesnosti a opakovatelnosti robota PUMA 200 bylo použito měřicí zařízení *Leica AT901B*. Toto zařízení pracuje na výše popsaném princip a jeho parametry jsou pro tuto úlohu více než dostatečné [28] viz. Obr. 6.2.



Obrázek 6.2: Měření pomocí Laser trackeru.

### 6.1.2 Měření přesnosti a opakovatelnosti - krychle

Pro ověření přesnosti a opakovatelnosti polohování byla připravena úloha, při které robot nastaví koncový efektor do bodů tvořící krychli.

Pro tuto operaci byl sestaven program, který rozdělí krychli na 64 bodů, tedy mřížka



4 x 4 x 4 body o hraně krychle 120 mm, tedy 40 mm mezi sousedními body. Robot v těchto bodech vždy 10 s počká, aby byla zajištěna stabilita měření.

### 6.1.3 Souřadnicový systém robota

Pro přesné určení souřadnicového systému robota je nutné provést změření bodů tvořících rovinu, které jsou tvořeny body koncového efektoru při pohybu vždy pouze jedné z os robota. Zde je postačující první a druhá osa a určení středu takto vzniklých kružnic pro určení bodu průniku všech os.

Takto vzniklé roviny je nutné posunout do počátku středu souřadného systému robota. Dále se provede na základě natočení rovin vůči souřadnicovému měřicího přístroje transformace změřených bodů do nově vypočteného souřadnicového systému robota.

Díky této transformaci je možné provést absolutní ověření přesnosti.

## 6.2 Opakovatelnost

Pro ověření opakovatelnosti robot najíždí do 64 bodů  $(x_{K_i}, y_{K_i}, z_{K_i})$  každé krychle  $(K_1, K_2)$  v pořadí v jednom směru a v opačném směru. Tento způsob zjistí opakovatelnost robota.

Abychom mohli vyhodnotit  $e$  rozdíl (chybu) mezi oběmi měřeními, je vypočítána absolutní velikost bodů v prostoru viz (6.1), tedy:

$$e = \sqrt{(x_{K_1} - x_{K_2})^2 + (y_{K_1} - y_{K_2})^2 + (z_{K_1} - z_{K_2})^2} \quad (6.1)$$

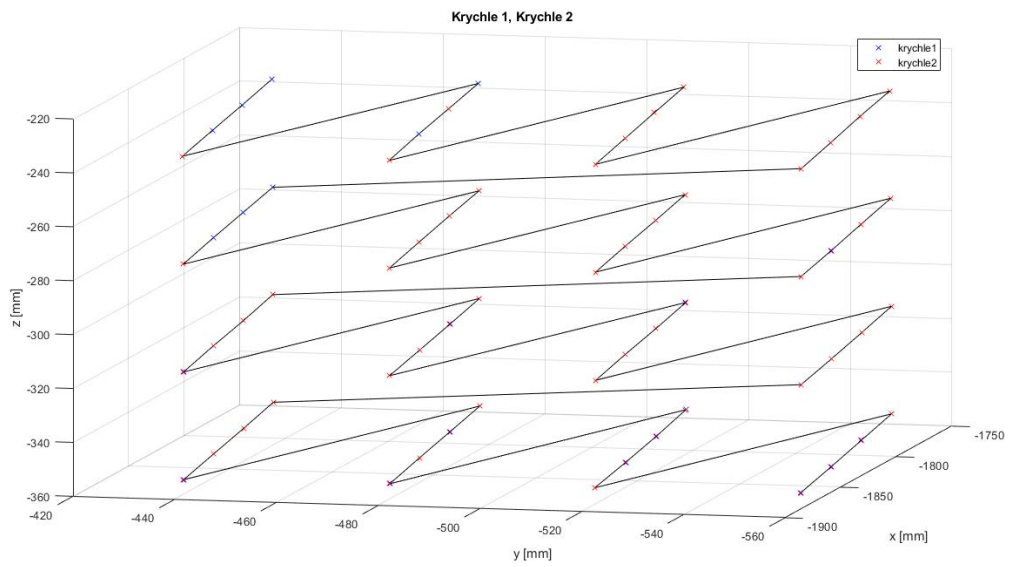
Zobrazení změřených hodnot mezi první krychlí a druhou krychlí je vidět na Obr. 6.3.

a tedy opakovatelnost v poloze 1 až 64 na Obr. 6.4. Její velikost v řádu 0,01 mm je celkem dobrá i ke stáří robota.

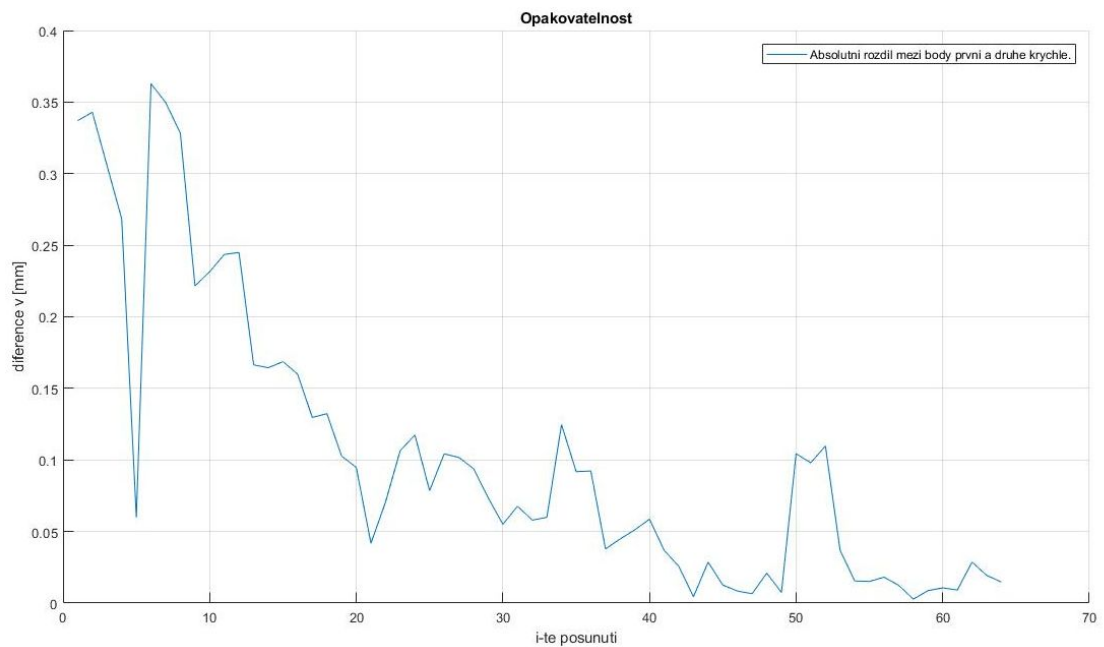
## 6.3 Přesnost polohování

Přesnost polohování je ověřena vzdáleností sousedních bodů v mřížce krychle, které jsou porovnány s konstantním rozdílem bodů tvořící krychli požadovanou programem.

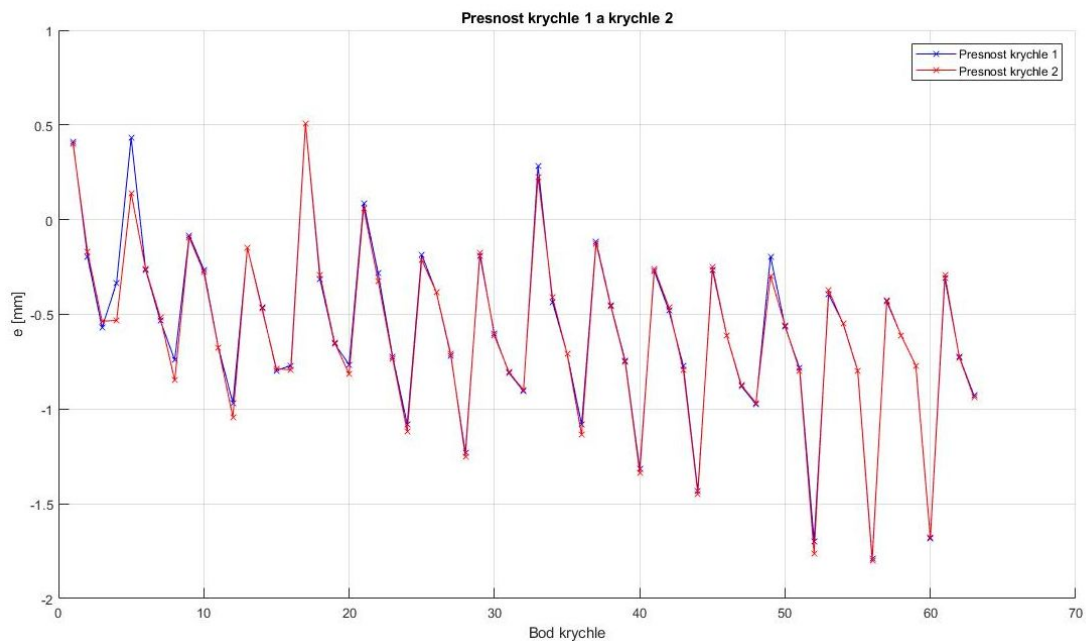
Tedy rozdíl bodů Programu  $|\mathbf{P}_i \mathbf{P}_{i+1}|$ , který je konstantní a je v případě ověření vždy 40 mm. A rozdílem vždy sousedních bodů  $\mathbf{B}$   $|\mathbf{B}_i \mathbf{B}_{i+1}|$ . Tento rozdíl je vidět na Obr. 6.5. Dosažená přesnost kolem 0,5 mm až 1 mm je celkem dobrá.



Obrázek 6.3: Porovnání krychlí v prostoru.



Obrázek 6.4: Opakovatelnost.



Obrázek 6.5: Přesnost polohování.

## 6.4 Koordinovaný pohyb

Pro ukázání funkčnosti robota pro plynulý pohyb byla zvolena úloha **LEV**, která si vytyčuje za cíl nakreslit lva ve znaku ČVUT.

Na Obr. 6.6 je výsledek.

## 6.5 Postup kalibrace robota

Zde je popsána možnost budoucího zlepšení parametrů robota.

Pro přesné určení rozměrů robota je možné pro první přiblížení použít data z manuálu [1]. Výroba má vždy jisté tolerance, každý robot je unikát svými rozměry a sestavením. To má za následek nepřesnosti, které musí být eliminovány právě kalibrací.

### 6.5.1 Kružnice

Pro určení rozměrů robota je vždy spuštěna pouze jedna osa robota, která otáčením tvoří body na kružnici v prostoru. V ideálním případě všechny body leží v jedné rovině a jsou na kružnici o poloměru rovném celému rozměru ramene. Tento postup je opakován pro všechny osy a jejich rozdíly jsou přesné rozměry daných ramen robota.

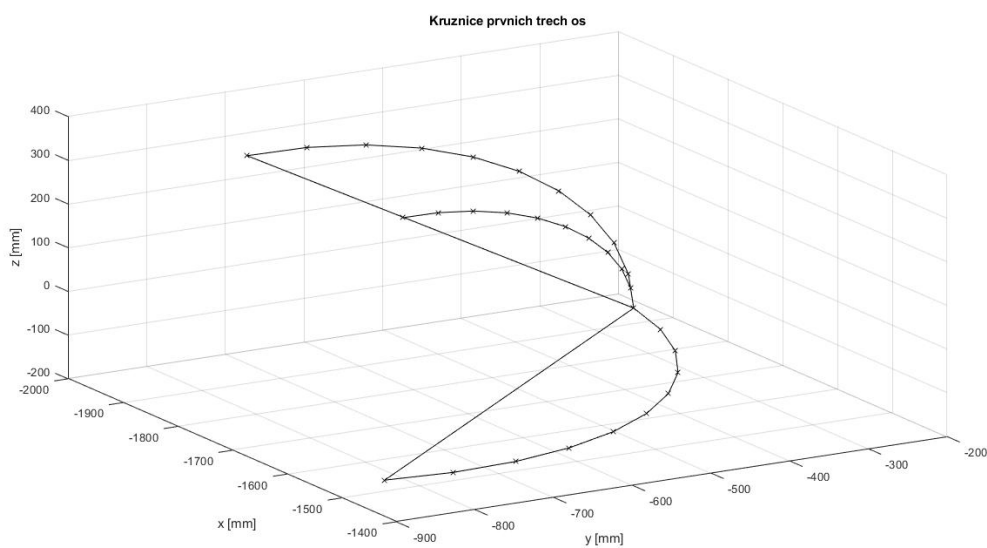


Obrázek 6.6: Lev FS ČVUT.

Naměřené body lze proložit kružnicí, jejíž střed a normála tvoří osu otáčení daného ramene robota. Ta musí být vypočtena metodou, kdy je nejprve z bodů vytvořena rovina (metoda nejmenších čtverců pro rovinu z n bodů), do které jsou promítnuty souřadnice bodů. Dále již řešíme metodou nejmenších čtverců pro získání parametrů rovnice kružnice.

Z rovnic kružnic dále získáme i roviny, které jsou kolmé k rovinám souřadnicového systému robota. Díky tomu jsme následně schopni zjistit posunutí a pootočení vůči souřadnicovému systému měřicího zařízení.

Tento druh kalibrace vychází z předpokladu dokonalých rotací. Na Obr. 6.7 jsou zobrazena naměřená data.



Obrázek 6.7: Ukázka kružnic prvních třech os.



# Kapitola 7

## Výsledky práce

Nový řídicí systém pro robot PUMA 200 byl realizován. Tento řídicí systém je využitelný pro nejrůznější roboty a je otevřený, tedy může být uživateli dále modifikován. Sestavení tohoto systému nebylo jednoduché, objem práce si vyžádal několik stovek pracovních hodin. Výsledkem bylo výrazné zmenšení hmotnosti kontroleru z původních 37 kg na 2 kg.

Vyvinutý řídicí systém je možné použít pro roboty s výkonem 250W na jednu osu. Jádro řídicího systému je možné použít pro libovolné větší roboty, po doplnění výkonové elektroniky výrobcem.

Tento systém může být vylepšen použitím observeru či implementací pokročilých metod interpolace měření z čidel s malým počtem inkrementů.

Podařilo se propojit kinematický model s řídicím modelem robota a implementovat jednoduchý trajektorový plánovač. Tak je možné využít robot pro technologické operace.

Tím je otevřena možnost kalibrace robota a zvýšení přesnosti pohybu. Proto byly provedeny základní experimenty s Laser trackerem, který umožnil ověřit přesnost robota bez kalibrace s nominálním modelem.

Další kroky kalibrace robota nebylo možné provést, neboť zprovoznění řídicího systému zabralo více času než bylo předpokládáno. Tyto kroky jsou popsány na konci kapitoly Kontrola přesnosti polohování robota pomocí Laser trackeru. V krajních polohách vykazuje robot nižší přesnost.

Byla dosažena přesnost  $1\text{ mm}$ , což je obvyklá dobrá hodnota nezkalibrovaného robota. Po kalibraci lze očekávat řádové zlepšení. Opakovatelnost robota je v řádu desetin milimetrů až setin milimetrů, což svědčí o tom, že i starší použitý robot má v dobrém stavu mechanickou část (převodovky a ložiska).

Na robotu by se dalo řešit velké množství dalších úloh, jako je například inverzní dynamika.

# Závěr

Cílem práce bylo využít mechanickou konstrukci robotu PUMA 200 a nahradit jeho řídicí systém současnými prostředky. Navíc je řídicí systém otevřený a použitelný pro další výuku a výzkum na Fakultě strojní ČVUT v Praze.

Nejdříve bylo nutné se seznámit s konceptem původního řídicího systému a možnostmi dnešní náhrady. To je popsáno v kapitole Průmyslové roboty, tím byl naplněn první cíl práce.

Na tomto základě byl vybrán hardware a software nového řídicího systému. Bylo vybráno prostředí pro vývoj MATLAB/Simulink a mikroprocesor TEXAS Instruments *F28379*. To je popsáno v kapitole Hardware a Software implementace. Celý proces zprovoznění byl zdlouhavý a byl provázen překonáváním celé řady překážek. Všechna úskalí se podařilo úspěšně překonat a je tím naplněn druhý cíl práce.

Použití řídicího systému pro konkrétního robota je založeno na znalosti kinematického modelu robota. Kinematický model robota je popsán v kapitole Kinematika. Skládá se z dopředné kinematické transformace robota, která je využita i pro jeho virtuální 3D model a z inverzní kinematické transformace nezbytně nutné pro jeho řízení. Tím je naplněn třetí cíl.

Tento inverzní kinematický model robota byl úspěšně spojen s řídicím systémem a umožňuje programování skutečného pohybu robota. To je popsáno v kapitole Řízení v Simulinku.

Funkčnost řídicího systému je ověřena na reálných pohybech robota, které jsou z vnějšku kontrolovány měřícím systémem Laser tracker. Protože pohyb robota je vždy odlišný od skutečného, je provedeno prvotní měření dosažené přesnosti a popsány další nutné kroky pro jeho zpřesnění. To je předmětem kapitoly Kontrola přesnosti polohování robota pomocí laser trackeru. Tím je naplněn poslední cíl této práce.

Nakonec jsou zhodnoceny výsledky práce s výhledem do budoucna v kapitole Výsledky.



# Seznam použitých zdrojů

- [1] UNIMATE PUMA Series 200, Industrial Robot, May 1984.
- [2] <https://ir.nctu.edu.tw/bitstream/11536/125183/36/509-2.pdf>
- [3] [https://byznys.lidovky.cz/skoda-auto-zbroji-na-vyrobu-novych-suv-letos-prijm-firmy-trhy.aspx?c=A160114\\_170545\\_firmy-trhy\\_pave,lidovky.cz,14.ledna2016](https://byznys.lidovky.cz/skoda-auto-zbroji-na-vyrobu-novych-suv-letos-prijm-firmy-trhy.aspx?c=A160114_170545_firmy-trhy_pave,lidovky.cz,14.ledna2016)
- [4] <http://www.directindustry.de/prod/abb-robotics/product-30265-566107.html>
- [5] <https://red-dot-21.com/p/design-products/industrial-equipment/robots-industrial-equipment/industrial-robot-kr-270-r2700-ultra/?ref=/manufacturer/kuka-roboter-gmbh/&sp=317>
- [6] <http://gelookahead.economist.com/slideshow/the-future-of-surgery/>
- [7] zdroj <https://www.digikey.com/en/product-highlight/t/texas-instruments/launchpad-boosterpack>
- [8] <http://www.ti.com/lit/ug/sprui77a/sprui77a.pdf>
- [9] <http://www.ti.com/tool/LAUNCHXL-F28379D?keyMatch=f28379d&tisearch=Search-EN-Everything>
- [10] <https://dev.ti.com/bpchecker/>
- [11] <http://www.ti.com/tool/BOOST-DRV8711/>
- [12] [http://electronoobs.com/eng\\_circuitos\\_tut1.php](http://electronoobs.com/eng_circuitos_tut1.php)
- [13] <http://www.ti.com/lit/ug/sprueu3a/sprueu3a.pdf>
- [14] <https://m.blog.naver.com/PostView.nhn?blogId=wi3er&logNo=221008438362&categoryNo=0&proxyReferer=&proxyReferer=https%3A%2F%2Fwww.google.com%2F>
- [15] <http://www.ti.com/lit/ds/symlink/drv8711.pdf>
- [16] <http://www.ti.com/lit/ug/sprug04a/sprug04a.pdf>

- [17] <http://www.ti.com/lit/ug/sprug05a/sprug05a.pdf>
- [18] George Ellis, Control System Design Guide, Elsevier Academic Press, San Diego, 2004
- [19] <http://www.ti.com/lit/ug/spru074f/spru074f.pdf>
- [20] V. Stejskal, M. Valasek, Kinematics and Dynamics of Machinery, Marcel Dekker, Inc. New York, 1996
- [21] Kalny, R., Valasek, M.: Continuous path control of non-simple robots, Robotersysteme, 7(1991), pp. 65-72
- [22] Kukula, P.; Valášek, M.: Kinematical Solution by Structural Approximation, In: Computational Kinematics: Proceedings of the 5th International Workshop on Computational Kinematics. Berlin: Springer-Verlag, 2009. pp. 323-330.
- [23] Sciavicco, L., Siciliano, B.: Modeling and control of robot manipulators, McGraw Hill, New York 1996
- [24] Lee C.S.G., Ziegler M.: A Geometric Approach in Solving Inverse Kinematics of PUMA Robots, Ann Arbor, Michigan 48109-1109, 1989
- [25] <http://www.mstarlabs.com/control/znrule.html>
- [26] <http://energysmartohio.com/uncategorized/one-knob-part-4-program-description/>
- [27] [http://www.optical-metrology-centre.com/Downloads/Tech\\_Briefs/TechBrief\\_LaserTracker.pdf](http://www.optical-metrology-centre.com/Downloads/Tech_Briefs/TechBrief_LaserTracker.pdf)
- [28] [http://hds.leica-geosystems.com/downloads123/m1/metrology/AT901/brochures-datasheet/Leica%20Absolute%20Tracker\\_ASME%20Specifications\\_en.pdf](http://hds.leica-geosystems.com/downloads123/m1/metrology/AT901/brochures-datasheet/Leica%20Absolute%20Tracker_ASME%20Specifications_en.pdf)

# Příloha

Na přiloženém DVD je text diplomové práce. V adresáři Program jsou zdrojové kódy a schémata Simulinku a skripty vytvořených programů rozříděné do odpovídajících složek. Dále jsou k práci přiložené zdroje, ze kterých bylo čerpáno.