

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra kybernetiky

## Kinematická kalibrace a optimalizace pohybu průmyslového robotu

**Matěj Vetchý**

Školitel: Ing. Vladimír Smutný, Ph.D.  
Květen 2023



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vetchý** Jméno: **Matěj** Osobní číslo: **491939**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra kybernetiky**  
Studijní program: **Kybernetika a robotika**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Kinematická kalibrace a optimalizace pohybu průmyslového robotu**

Název bakalářské práce anglicky:

**Kinematic Calibration and Motion Optimization of Industrial Manipulator**

Pokyny pro vypracování:

1. Seznamte se s Robotickým operačním systémem ROS a navrženým pracovištěm robota.
2. Seznamte se s metodami kinematické kalibrace a navrhnete postup pro kalibraci za pomoci měřicího přístroje Leica AT 403.
3. Kalibraci provedte a data vyhodnoťte.
4. Pro použitý robot se sedmi stupni volnosti analyzujte plánované pohyby a navrhnete algoritmus, který pro zadanou operaci určí optimální polohu sedmého stupně volnosti. Zohledněte technické parametry použitého mechanismu.
5. Provedte simulace, a pokud to bude možné i pokusy na skutečném robotu, a výsledky vyhodnoťte.

Seznam doporučené literatury:

- [1] Albert Nubiola, Ilian A. Bonev: Absolute calibration of an ABB IRB 1600 robot using a laser tracker, Robotics and Computer-Integrated Manufacturing, Volume 29, Issue 1, 2013, ISSN 0736-5845, <https://doi.org/10.1016/j.rcim.2012.06.004>.
- [2] Hayati, S.; Mirmirani, M. Improving the absolute positioning accuracy of robot manipulators. J. Field Robot. 2010, 2, 397–413.
- [3] Pamanes G., J. Alfonso, Wenger, Philippe, Zapata D., Jose Luis: Motion Planning of Redundant Manipulators for Specified Trajectory Tasks in Advances in Robot Kinematics: Theory and Applications, 2002, Springer, ISBN 978-94-017-0657-5
- [4] Santolaria, Jorge & Conte, Javier & Ginés, Manuel. (2012). Laser tracker-based kinematic parameter calibration of industrial robots by improved CPA method and active retroreflector. The International Journal of Advanced Manufacturing Technology. 66. 10.1007/s00170-012-4484-6.
- [5] Klanke, Stefan & Lebedev, Dmitry & Haschke, Robert & Steil, Jochen & Ritter, Helge. (2006). Dynamic Path Planning for a 7-DOF Robot Arm. IEEE International Conference on Intelligent Robots and Systems. 3879 - 3884. 10.1109/IROS.2006.281798.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Vladimír Smutný, Ph.D. robotické vnímání CIIRC**

Jméno a pracoviště druhého(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **17.12.2021**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Vladimír Smutný, Ph.D.  
podpis vedoucí(ho) práce

prof. Ing. Tomáš Svoboda, Ph.D.  
podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Poděkování

Chtěl bych vyjádřit upřímné díky svému vedoucímu práce za neustálou podporu a trpělivost, kterou mi projevoval během celého procesu. Vaše neocenitelná podpora mi byla velkou oporou.

Speciální díky si zaslouží kolega V. Jelinek za pomoc s měřením a zpracováním výsledků kalibrace.

Nakonec děkuji kolegům z CIIRC za seznámení s prostředím ROS.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 26. května 2023

.....

podpis autora práce

## Abstrakt

Tato práce se skládá ze dvou částí. V první části se zaměřujeme na kalibraci kinematiky robotické sestavy, která se skládá z robota YASKAWA GP 180-120 a lineárního pojezdu, na který je robot přidělán. Robotická sestava je určena k automatickému stavění zdí. Úkolem kalibrace je zjistit, zda je přesnost robota dostatečná pro úlohu, kterou plní a pokud tomu tak není, zjistit, jak je možné chybu snížit kalibračním modelem sestavy.

Kalibrací jsme zjistili chybu robota a se zákazníkem jsme zhodnotili, že chyba robota je dostatečná pro úlohy, které vykonává.

Připojením robota k pojezdu se robotická sestava stala redundantní, tedy disponuje nadbytečným stupněm volnosti.

Druhá část práce se zabývá optimalizací pohybu robota na pojezdu. Úkolem je zrychlit stavbu zdi pohybem robota na pojezdu a zároveň se vyhýbat singulárním polohám.

Pro plánování pohybu robota po pojezdu jsme vyvinuli optimalizační metodu, která využívá jako proměnné redundantní souřadnice robota na sedmé ose. Po odstranění redundance je pak možné analyticky vypočítat inverzní kinematiku. Plánování pohybu jsme rozdělili na plánování robota běžnými metodami a pohyb robota na sedmé ose.

Optimalizací se povedlo zrychlit pohyby při stavbě zdi o 12%.

**Klíčová slova:** kalibrace, optimalizace, inverzní kinematika, singularity robota, redundantní robot

**Školitel:** Ing. Vladimír Smutný, Ph.D.  
Robotické vnímání CIIRC  
Praha  
Jugoslávských partyzánů 1580

## Abstract

This work consists of two parts. The first part focuses on the kinematic calibration of a robotic system comprising a YASKAWA GP 180-120 robot and a linear axis to which the robot is attached. The robotic system is designed for automatic wall construction. The objective of the calibration is to determine, whether the robot's accuracy is sufficient for its task and, if not, to identify how the calibration of the system model can reduce the error.

Through calibration, we identified the robot's accuracy and, in collaboration with the customer, assessed that the accuracy is acceptable for the performed tasks.

By connecting the robot to the linear axis, the robotic system became redundant, meaning it has an excess degree of freedom.

The second part addresses the optimization of the robot's motion on the linear axis. The goal is to accelerate wall construction through the robot's movement on the linear axis while avoiding singular positions.

For the planning of the robot's motion on the linear axis, we developed an optimization method utilizing the redundant seventh axis as variables. After removing the redundancy, it is then possible to analytically calculate the inverse kinematics. The motion planning was divided into planning the robot using conventional methods and the motion of the robot on the seventh axis.

Through optimization, the movements of brick laying have been accelerated by 12%.

**Keywords:** calibration, optimization, inverse kinematics, robot singularities, redundant robot

**Title translation:** Kinematic Calibration and Motion Optimization of Industrial Manipulator

# Obsah

<b>1 Úvod</b>	<b>1</b>		
1.1 Úvod do kalibrace	1		
1.2 Úvod do optimalizační části	1		
1.3 State of the Art	2		
<b>2 Použitý hardware</b>	<b>3</b>		
2.1 Limity	3		
2.2 Denavitova-Hartenbergova notace	3		
2.3 Inverzní kinematická úloha	5		
2.4 Pracovní oblast	6		
<b>3 Měření</b>	<b>7</b>		
3.1 Příprava měření	7		
3.1.1 Maximální úhel reflektorů	7		
3.1.2 Kolizní model	8		
3.1.3 Kalibrační sady	8		
3.2 Měření	11		
3.2.1 Transformace laser tracker-svět	11		
<b>4 Zpracování výsledků měření</b>	<b>13</b>		
4.1 Filtrace naměřených dat	13		
4.2 Model	13		
4.3 Kalibrace	14		
4.3.1 Výsledky	15		
4.3.2 Závěr	15		
<b>5 Optimalizace</b>	<b>17</b>		
5.1 Metody k nalezení extrémů funkce	17		
5.2 Kritéria	18		
5.2.1 Čas	18		
5.2.2 Vzdálenost od singulárních poloh	18		
5.2.3 Vzdálenost od limitních poloh	20		
5.2.4 Vzdálenost od kolize	20		
5.3 Relativní poloha	20		
5.4 Limitní polohy na pojezdu	20		
5.5 Modifikace inverzní kinematické úlohy	21		
<b>6 Implementace optimalizace</b>	<b>25</b>		
6.1 Optimalizační metody	25		
6.2 Ztrátová funkce	26		
6.2.1 Vytvoření cesty	26		
6.2.2 Výpočet kritérií	27		
6.3 Komunikace	27		
<b>7 Experimentální výsledky</b>	<b>29</b>		
7.1 Řezy kritérií	30		
7.1.1 Konvekčnost ztrátové funkce	30		
7.1.2 Vliv pohybu na pojezdu na rychlost pokládání cihel	31		
<b>7.2 Průběh optimalizace</b>	<b>31</b>		
<b>7.3 Optimalizace celé zdi</b>	<b>34</b>		
7.3.1 Vliv optimalizace na čas položení cihly	34		
7.3.2 Vliv optimalizace na vzdálenost od singularit	35		
7.3.3 Statistika	35		
<b>8 Závěr</b>	<b>45</b>		
<b>Literatura</b>	<b>47</b>		
<b>Seznam zkratk</b>	<b>49</b>		

## Obrázky

2.1 Souřadnicové soustavy DH notace na modelu robota. Klouby jsou otočeny na $10^\circ$ , robot stojí v souřadnici 0.1 m. ....	5	7.5 Řez kritérií teoretického času pro nabrání první dvojice cihel a jejich napětění. ....	35
2.2 Řez pracovní oblasti bodu P s označenou oblastí s nevhodnou konfigurací. [7] .....	6	7.6 Řez kritéria teoretického času pro pohyb z levé pomocné polohy levé palety do pravé pomocné polohy pravé palety. ....	36
3.1 Reflektor na rotačním stolku. ....	8	7.7 Průběh pozic robota na pojezdu časově neomezené optimalizace kombinací metod differential evolution a Nelder-Mead. Referenční minimum se nachází v -0.28 m, -0.46 m, -0.65 m. ....	36
3.2 Nárýs a půdorys reálné klece (černě) a klece v kolizního modelu (červeně). [mm] .....	9	7.8 Průběh kritérií během časově neomezené optimalizace kombinací metod differential evolution a Nelder-Mead. Legenda kritérií je na obrázku 7.9. ....	37
3.3 Transformace při natačení reflektorů k laser trackeru. ....	9	7.9 Legenda kritérií .....	37
3.4 Znázornění veličin pro výpočet kloubů 5, 6 při natačení k laser trackeru z vektoru $\mathbf{t}_L^P$ . ....	10	7.10 Průběh pozic robota na pojezdu časově neomezené optimalizace metodou Nelder-Mead. ....	38
3.5 Vizualizace transformací od laser trackeru k reflektorům. ....	11	7.11 Průběh kritérií během časově neomezené optimalizace metodou Nelder-Mead. Legenda kritérií je na obrázku 7.9. ....	38
5.1 Znázornění vektorů $\mathbf{b}_{i-1}$ a $\mathbf{r}_{i-1}$ , $e$ . [11] .....	19	7.12 Průběh pozic robota na pojezdu časově omezené optimalizace metodou Nelder-Mead. ....	39
5.2 Řez pracovní oblasti robota ve výšce $z$ , s příkladem zobrazení úsečky relativní polohy na sedmé ose. [m]	21	7.13 Průběh kritérií během časově omezené optimalizace metodou Nelder-Mead. Legenda kritérií je na obrázku 7.9. ....	39
5.3 Aproximace pracovní oblasti robota 2.2 křivkami. ....	22	7.14 Průběh pozic robota na pojezdu časově omezené optimalizace metodou differential evolution. ...	40
5.4 Výskyt řešení IK v prostoru $q_4$ a $q_6$ (vlevo). Řešení IK splňující podmínku 2.1 (vpravo). ....	23	7.15 Průběh kritérií během časově omezené optimalizace metodou differential evolution. Legenda kritérií je na obrázku 7.9. ....	40
6.1 Struktura rozhraní komunikace služby pro optimalizaci pohybu po pojezdu. ....	27	7.16 Průběh pozic robota na pojezdu časově omezené optimalizace kombinací metod differential evolution a Nelder-Mead. ....	41
7.1 Řez součet kritérií pro pohyb ze startovní polohy a nabrání první dvojice cihel. ....	31		
7.2 Řez součtu kritérií pro nabrání první dvojice cihel a jejich napětění. ....	32		
7.3 Řez součtu kritérií pro napětění první dvojice cihel a jejich položení do zdi. ....	33		
7.4 Řez součtu kritérií pro nabrání dvojice cihel s pořadovým číslem 74 a přesunem do pravé pomocné polohy pravé palety. ....	34		



7.17 Průběh kritérií během časově omezené optimalizace kombinací metod differential evolution a Nelder-Mead. Legenda kritérií je na obrázku 7.9.....	41
7.18 Histogram vzdálenosti robota od os poloh pokládání vrchních pěti řad zdi. ....	42
7.19 Histogram poloh robota na pojezdu během stavby celé zdi. ..	42
7.20 Histogram poloh robota na pojezdu během stavby celé zdi. ..	43
7.21 Histogram poloh robota na pojezdu během nanášení pěny. ....	43

## Tabulky

2.1 Označení souřadnicových systémů.	3
2.2 Limity robota MOTOMAN GP180-120 [7] .....	4
2.3 DH notace robota MOTOMAN GP180-120. ....	5
2.4 DH notace pro robota s externím pojezdem. ....	6
3.1 Maximální relativní úhel reflektoru.	7
4.1 Modifikovaná DH notace sestavy	14
4.2 Statistika odchylek požadovaných poloh od reálných po eliminaci chyby měření. ....	15
4.3 Chybové parametry DH notace zkalibrovaného robota. ....	15
4.4 Statistika odchylek požadovaných poloh od reálných zkalibrovaného robota. ....	15
7.1 Schodištvé pořadí nabírání cihel z palety. Pohled na palety z boku. Robot stojí vlevo. ....	29
7.2 Výsledky experimentů optimalizace první dvojice cihel. ....	33



# Kapitola 1

## Úvod

V této práci budeme pracovat s průmyslovým šestiosým robotem YASKAWA GP 180-120, který je připevněn na lineární pojezd. Tato soustava bude sloužit k automatickému stavění zdí. V první části této práce se budeme zabývat kalibrací celé této sestavy. V druhé části se poté budeme zabývat optimalizací cesty robota a jeho jízdy na pojezdu. Celý projekt je implementovaný v prostředí ROS2.

### 1.1 Úvod do kalibrace

Cílem této části je změřit kinematickou chybu robota a v případě, nedostačné přesnosti, zkalibrovat parametry.

Robotická kalibrace je důležitou oblastí v oblasti robotiky a automatizace. Jde o proces měření a upřesnění parametrů robotického modelu ke zlepšení přesnosti robota. Jedním z běžných přístupů k robotické kalibraci je použití matematických modelů k popisu kinematiky a dynamiky robota a následné odhadování parametrů těchto modelů na základě naměřených dat.

Kinematické parametry hlavně popisují délky ramen robota a relativní orientaci sousedních os. Dynamické parametry poté popisují hmotnost ramen a kloubů, a vnitřní tření. [1] Kalibrace také může sloužit k opravě chyb způsobených opotřebením součástek.[2] V této práci se budeme hlavně zabývat kinematickou kalibrací.

Přesnost robota lze rozdělit na dvě kategorie, přesnost dojezdu do polohy (accuracy) a opakovatelnost (repeatability/precision). U opakovatelnosti jde o rozptyl výsledných poloh při opakovaném najíždění do stejné polohy. Na druhou stranu přesnost je vzdálenost dosažených poloh od očekávané polohy.

Výrobce udává parametry robota, mezi které patří i jeho přesnost a opakovatelnost. Tyto hodnoty však nemusejí být přesné, kvůli nedokonalosti výroby jednotlivých částí, či dostačující pro některé úlohy.

### 1.2 Úvod do optimalizační části

Robot během své funkce přejíždí mezi několika polohami, nabírá cihlu/y na paletě, nanáší pěnu na cihlu/y, pokládá cihlu/y na určené místo ve stěně.

Následně se vrací k paletě k další iteraci. Robot se při své práci pohybuje také v blízkosti singularit. Úkolem této práce je vytvořit ROS službu (service), která vrátí optimální polohy robota na pojezdu pro definované polohy.

Singularity jsou konfigurace, ve kterých nemá Jakobián plnou hodnotu. Některé metody pro vyhnutí se singulárním polohám zahrnují přidání redundantních stupňů volnosti nebo použití technik plánování trajektorie, které se vyhnou singulárním konfiguracím.

Výskyt singulárních poloh uvnitř pracovního prostoru robota je problém pro plánování trajektorie a řízení robota. Při konfiguracích manipulátoru blízko singulární polohy jsou pro relativně malé posuny koncového efektoru potřeba velké pohyby kloubů. [3]

Šestiosý robot má 6 stupňů volnosti (DOF). Připojení robota na pojezd přidá této sestavě 1 DOF, celková sestava má 7 DOF. Úloha pohybu 3D objektů ve 3D prostoru má 6 DOF, translace v x, y, z a rotace v ose x, y, z (roll, pitch, yaw). Jelikož má robotická sestava více stupňů volnosti než úloha, kterou vykonává, má inverzní kinematická úloha nekonečně mnoho řešení. Robot, který má více stupňů volnosti než je počet stupňů volnosti úlohy, kterou vykonává, se nazývá redundantní.

Cílem této části je využít redundantnosti robotické soustavy k vyhnutí se singulárním polohám a jízdou sedmé osy snížit čas uložení cihly do zdi.

## 1.3 State of the Art

Metoda použitá v [4] představuje použití laser trackeru k přesnému změření polohy chapadla robota. Metoda navrhuje připevnění reflektorů pro laser tracker a měření pohybu robota po předem definované trajektorii. Změřená data se poté porovnávají s předpokládanou trajektorií. Vzniklá chyba je použita k aktualizaci parametrů kinematického modelu robota.

ROS2 je nová verze ROSu (Robot Operating System). ROS umožňuje efektivní tvorbu složitých robotických systémů. ROS2 používá nový komunikační middleware s názvem Data Distribution Service (DDS). DDS umožňuje rychlejší a spolehlivější komunikaci mezi různými částmi robotického systému, což je důležité pro aplikace v reálném čase. Další výhodou ROSu 2 je větší modularita, které umožňuje tvorbu jednotlivých částí systému a jejich integraci ke zbytku systému. ROS2 podporuje více programovacích jazyků, včetně C++, Python a Rust. [5]

## Kapitola 2

### Použitý hardware

V této kapitole se seznámíme s robotickou sestavou použitou k automatické stavbě zdí. Sestava obsahuje šestiosého robota MOTOMAN GP180-120 od firmy YASKAWA, tento robot je připojen na externí pojezd vyvinutý firmou KM Robotics. Na koncový efektor je připevněno chapadlo přizpůsobeno k uchopení speciálně upravených cihel od firmy Wienerberger.

Jako přesné laserové zařízení k měření polohy chapadla máme k dispozici zařízení Leica Absolute Tracker AT403, dále jen laser tracker. Přesnost tohoto zařízení při měření polohy reflektorů je  $\pm 15 \mu\text{m} + 6 \mu\text{m}/\text{m}$ . Laser tracker měří až do vzdálenosti 320 m. [6]

V tabulce vidíme označení důležitých souřadnicových systémů.

Souřadnicový systém	Název v ROSu	Matematické označení
svět	base_link	W
robot	robot_base_link	R
DH notace 5 - P		P
laser tracker		L
koncový efektor	robot_tool0	E

**Tabulka 2.1:** Označení souřadnicových systémů.

### 2.1 Limity

Tabulka limitů jednotlivých kloubů samotného robota vidíme v tabulce 2.2.

Chapadlo robota obsahuje senzory připojené kabely. Kabely omezují součet otočení čtvrtého a šestého kloubu následovně

$$|q_4 + q_6| \leq \frac{3\pi}{4}. \quad (2.1)$$

### 2.2 Denavitova-Hartenbergova notace

Denavitova-Hartenbergova (DH) notace se používá k popisu kinematiky sériového manipulátoru. DH notace popisuje relaci, mezi dvěma osami po

Číslo kloubu	Název	Rozsah kloubu [m]	Maximální rychlost [m/s]
0	B	+1.203/-1.132	0.422

Číslo kloubu	Název	Rozsah kloubu [°]	Maximální rychlost [°/s]
1	S	± 180	125
2	L	+76/-60	115
3	U	+90/-86	125
4	R	± 360	182
5	B	± 130	175
6	T	± 360	265

**Tabulka 2.2:** Limity robota MOTOMAN GP180-120 [7]

sobě jdoucích kloubů v robotickém rameni, pomocí čtyř parametrů. Těmito parametry jsou

- $\Theta$  - úhel rotace kolem osy  $z$ ,
- $d$  - translace v ose  $z$ ,
- $a$  - translace v ose  $x$ ,
- $\alpha$  - rotace kolem osy  $x$ .

Pravidlem je, že osa  $z$  každého kloubu souhlasí s kladnou orientací rotace/translace daného kloubu. K výpočtu transformace mezi osami  $i$  a  $i + 1$  se k příslušné proměnné přičte kloubová souřadnice, k  $\Theta$  pro rotační kloub,  $d$  pro translační kloub. Transformace mezi základnou a koncovým efektozem robota lze vypočítat pomocí vzorce

$$\mathbf{T}_n^1 = \mathbf{T}_{z1} \mathbf{T}_{x1} \dots \mathbf{T}_{xn} \mathbf{T}_{zn}. \quad (2.2)$$

$$\mathbf{T}_{zi} = \begin{bmatrix} \cos \Theta_i & -\sin \Theta_i & 0 & 0 \\ \sin \Theta_i & \cos \Theta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{T}_{xi} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.3)$$

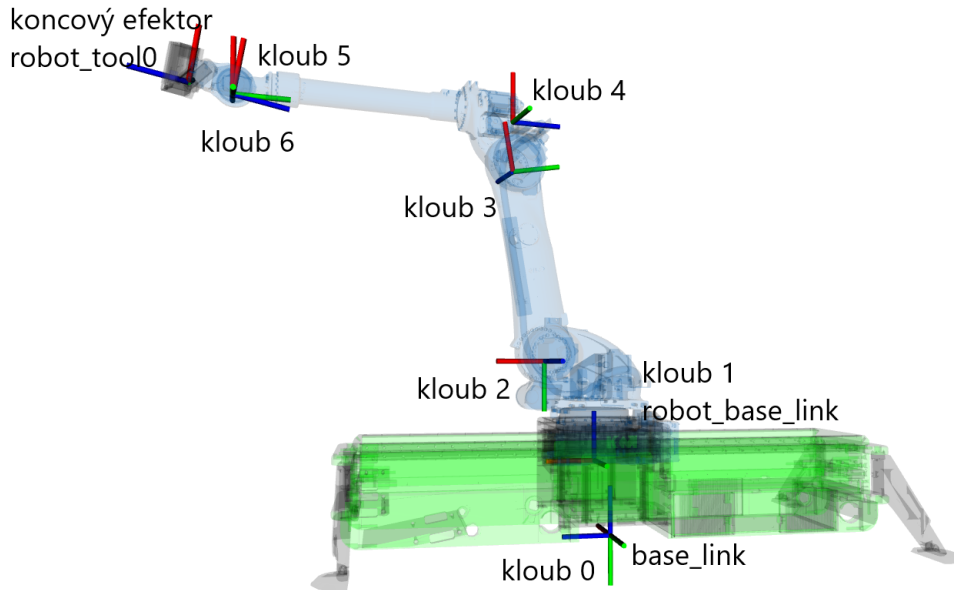
Výčíslené parametry DH notace robota MOTOMAN vidíme v tabulce 2.3, výsledná transformace odpovídá transformaci od počátku robota ke koncovému efektoru. Souřadnicové osy vytvořené DH notací vidíme na obrázku 2.1.

DH notaci pro celou robotickou sestavu vidíme v tabulce 2.4. Aby tato transformace odpovídala transformaci od počátku robotické soustavy ke koncovému efektoru je potřeba ji zleva vynásobit transformací

$$\mathbf{T}_0^W = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.4)$$

$i$	$\Theta[^\circ]$	$d[\text{m}]$	$a[\text{m}]$	$\alpha[^\circ]$
1	0	0.650	0.325	$-\frac{\pi}{2}$
2	$-\frac{\pi}{2}$	0	1.15	$\pi$
3	0	0	0.3	$-\frac{\pi}{2}$
4	0	-1.59	0	$\frac{\pi}{2}$
5	0	0	0	$-\frac{\pi}{2}$
6	0	0.225	0	$\pi$

**Tabulka 2.3:** DH notace robota MOTOMAN GP180-120.



**Obrázek 2.1:** Souřadnicové soustavy DH notace na modelu robota. Klouby jsou otočeny na  $10^\circ$ , robot stojí v souřadnici 0.1 m.

## 2.3 Inverzní kinematická úloha

Inverzní kinematická úloha (IK) slouží k výpočtu kloubových souřadnic, které docílí požadované polohy koncového efektoru. Používá se k přepočtu trajektorie robota z kartézských souřadnic do souřadnic kloubových, které již robot může vykonávat. IK pro robota MOTOMAN má pro některé polohy až 8 řešení. Z těchto řešení nás nezajímají ta, kde poloha je za robotem, viz obrázek 2.2.

Zbývající dvě řešení lze rozpoznat podle posledních tří kloubových souřadnic. Pátý kloub má podstatně větší motor na jedné straně, to vytváří na této straně větší tzv. "bouli". Poloha této boule závisí na rotaci čtvrtého kloubu. IK je implementována tak, aby vracela jedno řešení s boulí nahoře a jedno s boulí směřující dolů. Změna konfigurace se provádí podle vzorců

$$q_4^k = q_4 \pm \pi, \quad (2.5)$$

$$q_5^k = -q_5, \quad (2.6)$$

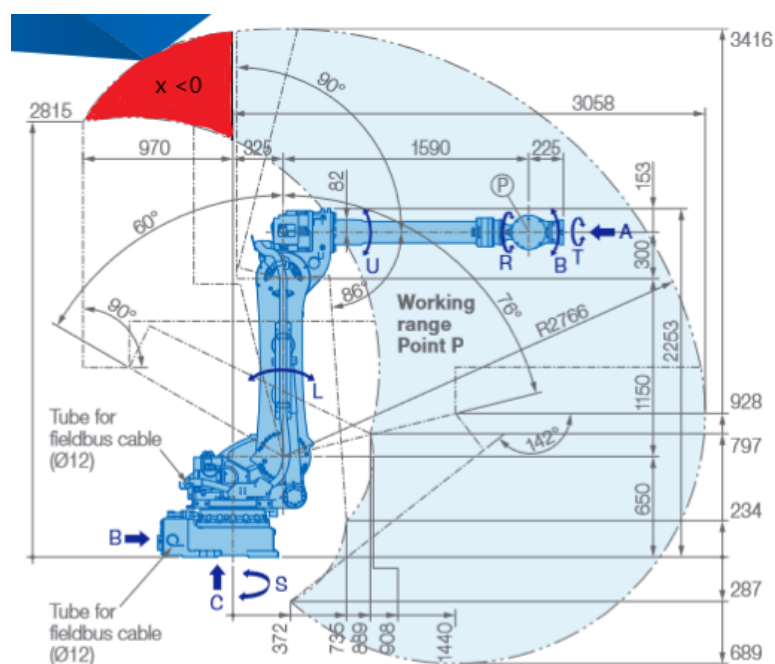
i	$\Theta$ [°]	d [m]	a [m]	$\alpha$ [°]
0	0	0	0	$\frac{\pi}{2}$
1	$\frac{\pi}{2}$	1.106	0.325	$-\frac{\pi}{2}$
2	$-\frac{\pi}{2}$	0	1.15	$\pi$
3	0	0	0.3	$-\frac{\pi}{2}$
4	0	-1.59	0	$\frac{\pi}{2}$
5	0	0	0	$-\frac{\pi}{2}$
6	0	0.225	0	$\pi$

**Tabulka 2.4:** DH notace pro robota s externím pojezdem.

$$q_6^k = q_6 \mp \pi. \quad (2.7)$$

## 2.4 Pracovní oblast

Pracovní oblast koncového efektoru šestiosého robota je složité znázornit v dvourozměrném prostoru. Můžeme však zobrazit řez tímto šesti dimenzionálním prostorem. Pomocí prvních tří kloubů můžeme přesně určit polohu bodu P. Polohu bodu P také můžeme přesně určit z polohy koncového efektoru. Můžeme tak určit zda poloha je přípustná. Leták robota [7] obsahuje řez pracovní oblasti pro polohu bodu P 2.2.



**Obrázek 2.2:** Řez pracovní oblasti bodu P s označenou oblastí s nevhodnou konfigurací. [7]



# Kapitola 3

## Měření

V této kapitole se zaměříme na přípravu měřených poloh, které laser trackerem změříme.

### 3.1 Příprava měření

Před samotným měřením je potřeba připravit měřené polohy. Tyto polohy musejí být laser trackerem změřitelné. Změřitelnost zaručíme správným natočením reflektorů vůči laser trackeru a absencí překážek v prostoru mezi reflektory a laser trackerem.

Důležitým faktorem je, aby robot do něčeho nenarazil, tudíž připravíme kolizní model prostředí, ve kterém se robot bude při měření pohybovat. Následně je potřeba připravit algoritmus pro měření jednotlivých bodů.

#### 3.1.1 Maximální úhel reflektorů

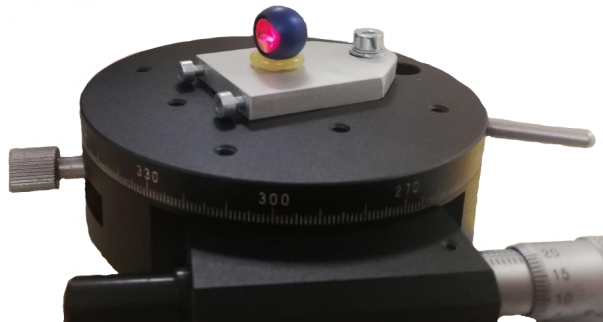
Provedeme experiment, při kterém budeme hledat maximální úhel natočení reflektoru od směru k laser trackeru, kdy nám laser tracker reflektor ještě změří.

Reflektor připevníme na rotační stůl. Stůl postavíme tak, aby byl reflektor ve stejné výšce jako senzor laser trackeru. Laser tracker jsme postavili přibližně 180 cm od reflektoru. Reflektor na rotačním stolku vidíme na obrázku 3.1

Provedli jsme dvě nezávislá měření. Při druhém měření jsme otočili rotační stůl o 180°. Naměřené hodnoty vidíme v tabulce 3.1. Obě měření mají stejný výsledek 36.5°. Pro měření reflektorů na chapadle budeme uvažovat maximální úhel za 30°.

	1. měření	2. Měření
Maximální úhel [°]	91	270
Minimální úhel [°]	18	199
Střed [°]	54.5	234.5
Rozsah [°]	36.5	36.5

**Tabulka 3.1:** Maximální relativní úhel reflektoru.



Obrázek 3.1: Reflektor na rotačním stolku.

### ■ 3.1.2 Kolizní model

Samotný robot a jeho pojezd svůj kolizní model již mají. Při měření se však robot bude nacházet v kleci a je proto nutné tuto klec přidat do kolizního modelu. Na obrázku 3.2 vidíme schéma a rozměry reálné klece a klece implementované do kolizního modelu.

Klec v kolizním modelu je značně menší z důvodů nepřesného postavení robota do středu klece. Takto máme vyhrazený čtverec 130x100 cm pro umístění robota, aby kolizní model zabránil kolizím.

### ■ 3.1.3 Kalibrační sady

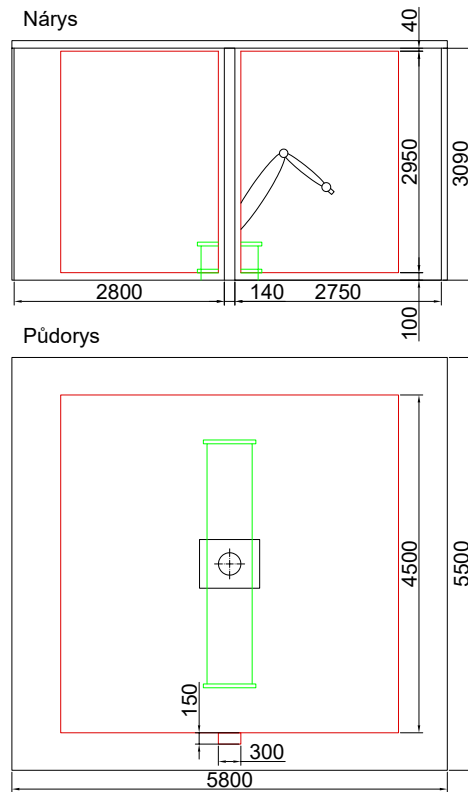
Jednou z měřicích sad bude kartézský součin kloubů. Budeme chtít měřit chybu robota při nabírání a pokládání cihel. Chceme také vědět, jak vyklonění robota z pojezdu ovlivní chybu. To budou další měřené sady. Na jednu stranu chapadla robota připevníme tři reflektory. Měřené body musíme připravit tak, aby tyto reflektory byly laser trackerem změřitelné.

### ■ Kartézský součin kloubových souřadnic

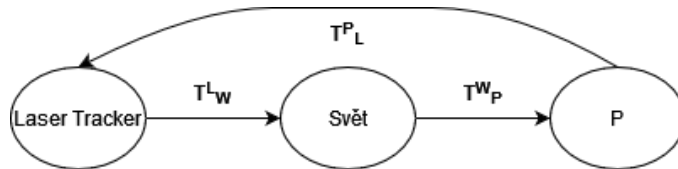
Pro zpřesnění parametrů DH notace využijeme kartézského součinu kloubových souřadnic. Množství měřených bodů lze snadno regulovat počtem kombinací pro každý kloub. Kartézským součinem získáme rovnoměrné rozložení poloh v prostoru kloubových souřadnic.

Pokud chceme dělat kartézský součin všech šesti kloubů nebudeme mít zaručeno, že reflektory jsou laser trackerem vidět. Bude proto vhodnější dělat pouze kombinace prvních čtyř kloubů a použít další dva na natočení strany chapadla s reflektory k laser trackeru. Je také potřeba zkontrolovat, že robot není v kolizi sám se sebou či okolím a také pokud mezi reflektory a laser trackerem není překážka.

Pro natočení reflektorů k laser trackeru potřebujeme znát transformaci laser tracker-svět  $\mathbf{T}_W^L$ . Jak tuto transformaci získat si řekneme v 3.2.1. Pomocí DH notace a kombinace prvních čtyř kloubů vytvoříme transformaci  $\mathbf{T}_P^W$ .



**Obrázek 3.2:** Nárys a půdorys reálné klece (černě) a klece v kolizního modelu (červeně). [mm]



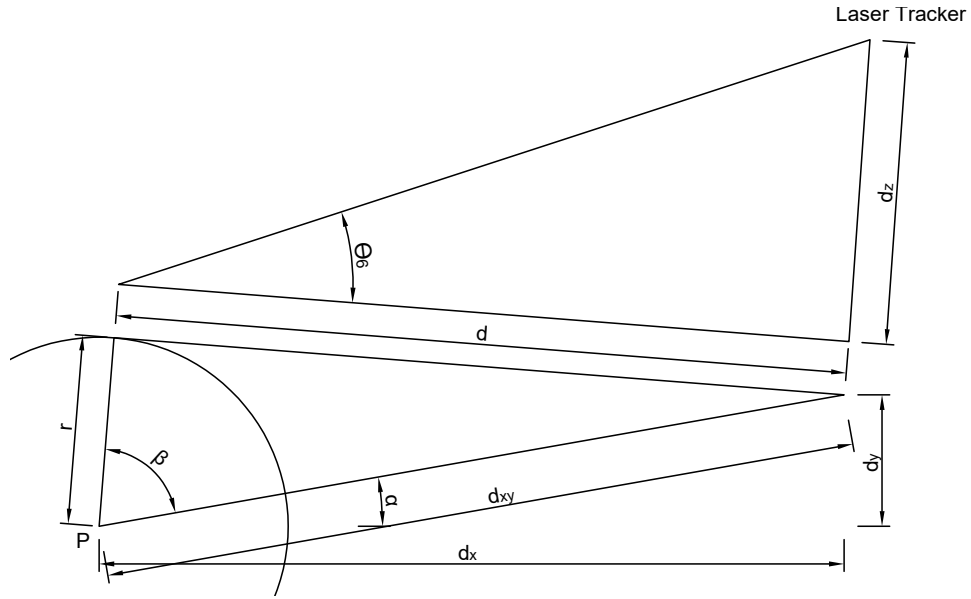
**Obrázek 3.3:** Transformace při natáčení reflektorů k laser trackeru.

Natočení kloubů 5, 6 lze získat z vektoru z bodu P k laser trackeru  $\mathbf{t}_L^P$ . Tato translace je součástí transformační matice

$$\mathbf{T}_L^P = \mathbf{T}_W^P \mathbf{T}_L^W = \begin{bmatrix} \mathbf{R}_L^P & \mathbf{t}_L^P \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.1)$$

$$\mathbf{t}_L^P = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad (3.2)$$

Kloubem 5 ( $\Theta_5$ ) srovnáme reflektory do roviny ve které se nachází laser tracker. Kloubem 6 ( $\Theta_6$ ) nasměrujeme reflektory k laser trackeru.  $r = 250$  mm je vzdálenost bodu P od roviny reflektorů na chapadle.  $d_{xy}$  je vzdálenost laser trackeru od P v rovině os  $x, y$ .



**Obrázek 3.4:** Znázornění veličin pro výpočet kloubů 5, 6 při natáčení k laser trackeru z vektoru  $\mathbf{t}_L^P$ .

$$d_{xy} = \sqrt{d_x^2 + d_y^2}, \quad (3.3)$$

$$d = \sqrt{d_{xy}^2 - r^2}, \quad (3.4)$$

$$\alpha = \frac{\pi}{2} + \arctan \frac{d_x}{d_y}, \quad (3.5)$$

$$\beta = \arccos \frac{r}{d_{xy}}, \quad (3.6)$$

$$\Theta_5 = \alpha \pm \beta, \quad (3.7)$$

$$\Theta_6 = \pm \left( \frac{\pi}{2} + \arccos \frac{d_z}{d} \right). \quad (3.8)$$

V případě, že se laser tracker již v ose pátého kloubu nachází nebo-li  $d_{xy} \leq r$ , můžeme kloub 5 zvolit libovolně. Zvolíme proto úhly  $\Theta_5 = 0$  a  $\Theta_6 = 0$  nebo  $\Theta_6 = \pi$  podle toho, na které straně se laser tracker nachází.

### ■ Dataset pro polohy ve zdi, palety a kolmé roviny k ose pojezdu

Chceme vědět chybu, kterou robot má během pracovního procesu pokládání cihel. Vytvoříme proto polohy v kartézských souřadnicích, simulující pokládání a nabírání cihel. Pro výklon z pojezdu vytvoříme polohy v rovině kolmé k ose pojezdu, v různých výškách a vzdálenostech od pojezdu. Vybereme pouze ty, které jsou v pracovní oblasti robota. K vyhnutí se singularitě posuneme tyto polohy z osy robota o 10 cm blíže k laser trackeru. Pro všechny tyto polohy vypočítáme IK v obou konfiguracích. Polohy zdi a palety budeme měřit s robotem postaveným v nulových souřadnicích pojezdu. Polohy v rovině

paralelní k ose pojezdu budeme měřit ve třech souřadnicích pojezdu a to -1 m, 0 m, 1 m.

## 3.2 Měření

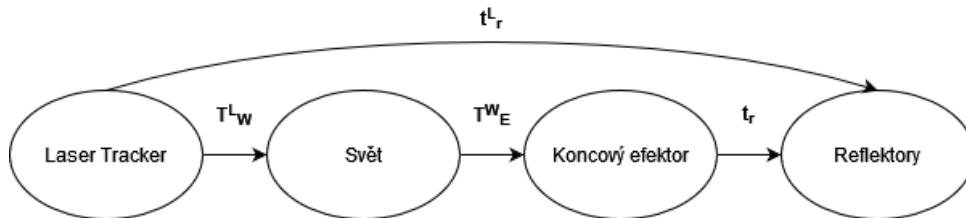
Před zahájením měření je potřeba zkontrolovat přesnost kolizního modelu a popřípadě jej upravit, aby odpovídal realitě. Je potřeba umístit laser tracker tak, aby relativní natočení os reflektorů vůči laser trackeru bylo menší než  $30^\circ$  pro libovolnou měřenou polohu. Pro splnění této podmínky stačí postavit laser tracker do osy pojezdu ve vzdálenosti 3,9 m od klece. Na chapadlo umístíme tři reflektory. Každému reflektoru odhadneme translaci z koncového efektoru  $\mathbf{t}_r$ . Reflektory připevníme i na nepohyblivou část světa, například klec či stěnu pro kontrolu případné změny polohy laser trackeru.

Pro měření všech připravených poloh, nejdříve pošleme robota do polohy a laser trackerem změříme tři reflektory na chapadle. Po každých 50 změřených poloh také zkontrolujeme konstantní polohu laser trackeru změřením reflektorů na stěně.

Celkem jsme naměřili 527 kalibračních poloh.

### 3.2.1 Transformace laser tracker-svět

Laser tracker potřebuje pro automatické měření znát přibližnou polohu reflektorů. V diagramu 3.5 vidíme jak polohu reflektorů získat, prvně ze samotného měření laser trackeru  $\mathbf{t}_r^L$ , a také z transformací laser tracker-svět, DH notace robota, a odhadu poloh reflektorů na chapadle. Jedinou neznámou transformací zde je transformace laser tracker-svět.



Obrázek 3.5: Vizualizace transformací od laser trackeru k reflektorům.

Změříme laser trackerem polohu reflektorů v některých polohách robota. Pro tyto polohy získáme vektory .

$$\mathbf{t}_{ri}^W = \mathbf{T}_E^W \mathbf{t}_{ri}, \quad (3.9)$$

Transformaci  $\mathbf{T}_W^L$  získáme ze

$$\mathbf{p}_i = \mathbf{t}_{ri}^W, \quad \mathbf{q}_i = \mathbf{t}_{ri}^L, \quad (3.10)$$

$$\mathbf{R}^*, \mathbf{t}^* = \arg \min_{\mathbf{R}, \mathbf{t}} \sum_i \|\mathbf{R}\mathbf{p}_i + \mathbf{t} - \mathbf{q}_i\|^2. \quad (3.11)$$

Dle [8] je řešením

$$\bar{\mathbf{q}} = \frac{1}{N} \sum_i^N \mathbf{q}_i, \quad \bar{\mathbf{p}} = \frac{1}{N} \sum_i^N \mathbf{p}_i, \quad (3.12)$$

$$\mathbf{q}'_i = \mathbf{q}_i - \bar{\mathbf{q}}, \quad \mathbf{p}'_i = \mathbf{p}_i - \bar{\mathbf{p}}, \quad (3.13)$$

$$\mathbf{U}\mathbf{S}\mathbf{V}^T = \sum_i \mathbf{p}'_i \mathbf{q}'_i{}^T, \quad (3.14)$$

$$\mathbf{R}^* = \mathbf{V}\mathbf{U}^T, \quad (3.15)$$

$$\mathbf{t}^* = \bar{\mathbf{q}} - \mathbf{R}^* \bar{\mathbf{p}}. \quad (3.16)$$

$$\mathbf{T}_{\mathbf{W}}^{\mathbf{L}} = \begin{bmatrix} \mathbf{R}^* & \mathbf{t}^* \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (3.17)$$

## Kapitola 4

### Zpracování výsledků měření

V této kapitole se budeme zabývat zpracováním naměřených dat. Vytvoříme kinematický model robota. Z naměřených dat vypočítáme chybu robota. Do modelu přidáme parametry, které budeme kalibrovat. Budeme se zabývat pouze základy kalibrace, podrobnější informace jsou k dispozici v dokumentu [9], který byl vytvořen pro firmu KM Roboticks.

Veškerá měření poloh reflektorů na stěně byla konstantní, není proto nutné implementovat korekci.

#### 4.1 Filtrace naměřených dat

Z naměřených dat nejprve vyloučíme ty, kde laser tracker nezměřil některý z reflektorů.

U zbylých dat zkontrolujeme, zda je každý reflektor změřen správně. Spočítáme relativní vzdálenosti středů jednotlivých reflektorů. Tyto vzdálenosti by měly být konstantní. Odfiltrujeme ta data, pro která se tyto vzdálenosti vzdalují od průměru o více než 0.3 m. Laser tracker se mohl zaměřit na jiný reflektor nebo špatně změřit správný reflektor.

Odfiltrovali jsme 116 poloh. Zbylých 411 poloh použijeme ke kalibraci.

#### 4.2 Model

Pro vytvoření modelu využijeme transformace laser tracker-svět  $\mathbf{T}_{\mathbf{W}}^{\mathbf{L}}$ , DH notaci robota  $\mathbf{T}_E^{\mathbf{W}}$ , transformaci  $\mathbf{t}_r$ , viz diagram 3.5.

Z důvodu paralelnosti os druhého a třetího kloubu, použijeme pro druhý kloub modifikovanou DH notaci. Kalibrace parametrů tak bude numericky stabilnější. Modifikované DH notaci pro rotační kloub vidíme v tabulce 4.1, kde  $\beta$  označuje rotaci kolem osy  $y$ .

Je důležité vybrat vhodné chybové parametry pro kalibraci tak, aby byly vzájemně nezávislé.

Pro eliminaci chyby měření do modelu přidáme chybové parametry transformací  $\mathbf{T}_{\mathbf{W}}^{\mathbf{L}}$  a  $\mathbf{t}_r$ .

1. 6 parametrů - Translační a rotační vektor transformace svět-laser tracker.

$i$	$\Theta[\text{rad}]$	$d[\text{m}]$	$a[\text{m}]$	$\alpha[\text{rad}]$	$\beta[\text{rad}]$
0	0	0	0	$\frac{\pi}{2}$	
1	$\frac{\pi}{2}$	1.106	0.325	$-\frac{\pi}{2}$	
2	$-\frac{\pi}{2}$		1.15	0	$\pi$
3	$\pi$	0	0.3	$-\frac{\pi}{2}$	
4	0	-1.59	0	$\frac{\pi}{2}$	
5	0	0	0	$-\frac{\pi}{2}$	
6	0	0.225	0	$\pi$	

**Tabulka 4.1:** Modifikovaná DH notace sestavy

2. 9 parametrů - Translační vektor pro každý reflektor na chapadle  $\mathbf{t}_r$ .
3. 7 parametrů - Poloha robota na pojezdu  $d_0$ , tento parametr je potřeba kalibrovat z důvodu absence přímého čtení kloubu 0 v měřených polohách.

Po eliminaci chyby měření ke stávajícím chybovým parametrům přidáme do modelu také chybové parametry DH notace robota.

4. 5 parametrů - Offset kloubových souřadnic  $\theta$  kloubů 1 až 5 .
5. 7 parametrů - Náklon mezi osami  $\alpha_0 - \alpha_5, \beta_2$ .
6. 7 parametrů - Délky ramen  $a_1 - a_5, d_3, d_4$ .

### 4.3 Kalibrace

Ke kalibraci parametrů využijeme metodu nejmenších čtverců. Minimalizujeme funkci

$$\sum_{i=1}^n \left| e(\mathbf{T}_L^W(\mathbf{x})h(\mathbf{t}_{r,i}^L)) - e(\mathbf{T}_E^W(\mathbf{x}, \mathbf{q}_i)h(\mathbf{t}_r(\mathbf{x}))) \right|^2, \quad (4.1)$$

kde

- $\mathbf{t}_{r,i}^L$  -  $i$ -tý změřený bod laser trackerem,
- $\mathbf{q}_i$  - Kloubové souřadnice robota při měření  $i$ -tého bodu,
- $\mathbf{x}$  - Vektor kalibrovaných parametrů.

Transformaci  $\mathbf{T}_L^W(\mathbf{x})$  a  $\mathbf{t}_r(\mathbf{x})$  vytvoříme přímo z kalibrovaných parametrů. Transformaci  $\mathbf{T}_E^W(\mathbf{x})$  vypočítáme z DH notace v kloubových souřadnicích vyčtených z robota. DH notaci modifikujeme kalibrovanými parametry. Funkce  $e$  převání homogenní souřadnice na souřadnice euklidovské. Funkce  $h$  převání euklidovské souřadnice na souřadnice homogenní.



### 4.3.1 Výsledky

Předpokládáme, že chyba robota má normální rozdělení. Parametry, které nás mohou zajímat jsou průměrná chyba, maximální chyba,  $3.4\sigma$  - chyba do které se vejde 99.97% hodnot. Změřené body jsme rozdělili na dvě sady. Na první budeme kalibrovat kalibrační parametry. První sada obsahuje polohy kartézského součinu a polohy z roviny paralelní k ose pojezdu. Druhá poslouží k nezávislému zhodnocení. Druhá sada obsahuje polohy zdi.

Výsledek po minimalizaci funkce 4.1 pouze s parametry chyby měření (1-3) vidíme v tabulce 4.2. Chyba reprezentuje chybu nekalibrovaného robota.

sada	průměrná chyba [m]	maximální chyba [m]	$3.4\sigma$ [m]
kalibrační	0.0028	0.0077	0.0062
nezávislá	0.0033	0.0071	0.0069

**Tabulka 4.2:** Statistika odchylek požadovaných poloh od reálných po eliminaci chyby měření.

Pro kalibraci robota přidáme do modelu chybové parametry DH notace (4-6). Chybové parametry DH notace po kalibraci vidíme v tabulce 4.3. Výsledek minimalizace vidíme v tabulce 4.4.

$i$	$\Delta\theta$ [rad]	$\Delta d$ [m]	$\Delta a$ [m]	$\Delta\alpha$ [rad]	$\Delta\beta$ [rad]
0			-0.002	-0.0001	
1	0.0021		-0.0007	-0.0014	
2	0.0023		0.001	-0.0006	-0.0012
3	-0.0004	0.0006	0.0009	0.0003	
4	-0.0016	0.0009	0	-0.0004	
5	0.0004		-0.0002	0.0002	

**Tabulka 4.3:** Chybové parametry DH notace zkalibrovaného robota.

sada	průměrná chyba [m]	maximální chyba [m]	$3.4\sigma$ [m]
kalibrační	0.0009	0.003	0.002
nezávislá	0.0013	0.0025	0.0028

**Tabulka 4.4:** Statistika odchylek požadovaných poloh od reálných zkalibrovaného robota.

### 4.3.2 Závěr

Kalibrací modelu robota se podařilo snížit chybu robota na 40% chyby nekalibrovaného robota. Robotická sestava je určena k hromadné výrobě. Chceme-li, aby každá vyrobená robotická sestava měla nízkou chybu, je nutno kalibrovat každou zvlášť. U robotů k hromadné výrobě je vhodné přemýšlet nad kalibračními parametry, které budou mít všechny vyrobené sestavy společné. Příklad takových parametrů může být prohnutí se pod vlastní vahou. Tyto parametry jsou prozkoumány v dokumentu [9].

#### 4. Zpracování výsledků měření

---

Průměrná chyba nekalibrované sestavy je 0.33 mm. Se zákazníkem jsme zhodnotili, že přesnost nekalibrovaného robota je dostačující pro úlohy, které vykonává.

## Kapitola 5

### Optimalizace

Cílem optimalizační části je využití externího pojezdu pro zrychlení stavby zdi a vyhýbání se singularitám. V této kapitole se budeme zabývat optimalizačním procesem. Zvolíme vhodná kritéria k minimalizaci.

Polohy, které robot musí navštívit, rozdělíme do skupin, ve kterých nechceme, aby se robot pohyboval po pojezdu. Tímto snížíme případnou chybu spojenou s jízdou sedmé osy. Optimalizační funkce bude hledat optimální kloubové souřadnice pro sedmou osu při práci v těchto skupinách. Tímto se zbavíme redundantnosti sestavy a umožní nám to analyticky počítat inverzní kinematiku. Problém můžeme rozdělit na dvě části, plánování pohybu robota a pohyb po sedmé ose. Abychom mohli polohy přepočítat do souřadnic robota, musejí být definovány ve světových souřadnicích. Z poloh vytvoříme cestu, po které chceme, aby se robot pohyboval. Tuto cestu přepočítáme do souřadnic robota. Mezi polohami skupin robot přejíždí po sedmé ose lineární interpolací.

Je třeba, aby se definovala poloha robota na začátku cesty. Tato poloha je konstantní a optimalizací jí nezměníme. Pro každou polohu v cestě vypočítáme ztrátová kritéria.

#### 5.1 Metody k nalezení extrémů funkce

Existuje široká škála optimalizačních gradientních metod, které se používají k nalezení minima funkcí. Tyto metody jsou obzvláště užitečné pro konvexní funkce, kde existuje pouze jediné globální minimum. Nevýhodou je, že gradientní metody potřebují k nalezení minima derivace funkce. Námi minimalizovaná funkce nemusí být diferencovatelná.

V kapitole 7.1.1 zjistíme, že námi optimalizovaná funkce nemusí být konvexní. Při hledání minima nekonvexní funkce je často možné narazit pouze na lokální minimum. Lokální minimum je bod, ve kterém je hodnota funkce nejnižší v okolí daného bodu, avšak může být vyšší než hodnota globálního minima.

Existuje několik technik, které se snaží překonat problém lokálních minim. Jednou z nich je použití různých počátečních bodů a opakování optimalizačního procesu, aby se minimalizovala pravděpodobnost uvíznutí v lokálním minimu. Další možností je předřazení optimalizačními algoritmy, které zkoumají různé body v prostoru a mají vyšší pravděpodobnost objevení globálního

minima. Tyto metody jsou však výpočetně náročně a potřebují mnoho iterací k nalezení globálního minima.

Opakování optimalizačního procesu se nehodí v případě, kdy je optimalizace časově omezená. Metody prohledávající celý prostor potřebují znát intervaly proměnných ve kterém mají optimum hledat.

## 5.2 Kritéria

Výběr správných kritérií optimalizace je stěžejní pro zlepšení funkcionality robota. Hlavním kritériem je v našem případě minimalizace času, ostatní kritéria napomáhají řízení robota.

### 5.2.1 Čas

Pohybem robota na sedmé ose můžeme ovlivnit rychlost vykonávání jeho činnosti. Reálná doba pohybu robota je určena během výpočtu trajektorie. Tento výpočet je však velmi časově náročný pro implementaci do optimalizačního procesu. Levně však můžeme vypočítat čas teoretický. Tento čas vypočítáme jako

$$\sum_s \max_i \left( \frac{\Delta q_i}{v_{qi, max}} \right) \quad i = 0, \dots, 6, \quad (5.1)$$

kde  $v_{qi, max}$  je maximální rychlost  $i$ -tého kloubu a  $\Delta q_i$  označuje posun v  $i$ -tém kloubu mezi po sobě jdoucími polohami na cestě.

### 5.2.2 Vzdálenost od singulárních poloh

Singulární polohy manipulátoru se nacházejí tam, kde je jakobián robota singulární. Jakobián robota vypočítáme z DH notace jako

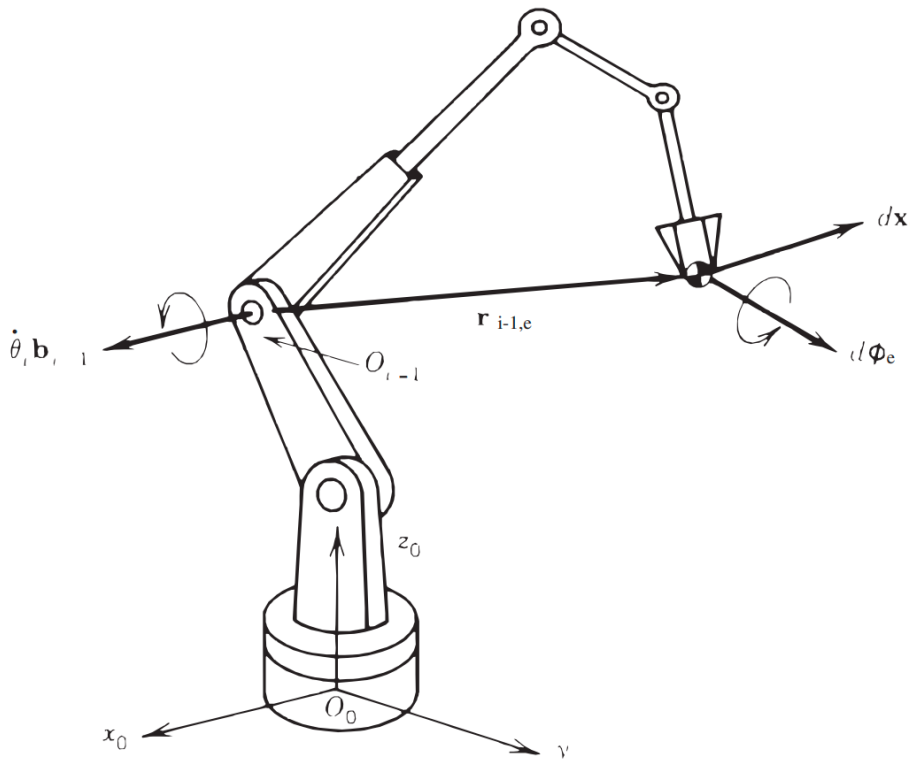
$$J = [J_1 | J_2 | \dots | J_6], \quad (5.2)$$

$$J_i = \begin{bmatrix} \mathbf{b}_{i-1} \times \mathbf{r}_{i-1,e} \\ \mathbf{b}_{i-1} \end{bmatrix}, \quad (5.3)$$

$$\mathbf{b}_{i-1} = \mathbf{R}_1^0(q_1) \dots \mathbf{R}_{i-1}^{i-2}(q_{i-1}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (5.4)$$

$$\mathbf{r}_{i-1, e} = e(\mathbf{T}_1^0 \dots \mathbf{T}_7^6 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}) - e(\mathbf{T}_1^0 \dots \mathbf{T}_{i-1}^{i-2} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}), \quad (5.5)$$

kde pro rotační kloub platí rovnice 5.3. Okolo vektoru  $\mathbf{b}_{i-1}$  se otáčí  $i-1$  kloub. Vektor  $\mathbf{r}_{i-1, e}$  reprezentuje vektor mezi  $i$ -tým kloubem a koncovým efektem. Funkce  $e$  převání homogenní souřadnice na souřadnice euklidovské.[10]



**Obrázek 5.1:** Znázornění vektorů  $\mathbf{b}_{i-1}$  a  $\mathbf{r}_{i-1, e}$ . [11]

Vizualizaci vektorů  $\mathbf{b}_{i-1}$  a  $\mathbf{r}_{i-1}$  vidíme na obrázku 5.1. Sloupec jakobiánu pro pojezd je konstantní a tudíž ho nemusíme počítat.

Existují různé způsoby, jak zjistit, zda je čtvercová matice singulární.

- Determinant je roven 0.
- Nemá plnou hodnotu  $\Leftrightarrow \text{rank} < n$  pro matice  $n \times n$
- Podmíněnost matice - Singulárním rozkladem (SVD) zjistíme singulární čísla matice. Matice je singulární, jestliže alespoň jedno singulární číslo matice je rovno 0.

Z výpočtu jakobiánu 5.6 vidíme, že jakobián je rozměrově rozdělen na dvě části, na translační a rotační část. Pro nepřekrývání rozměrů při výpočtu je nezbytné rozdělit výpočet na translační a rotační část. Výpočet determinantu nečtvercové matice není možný. Pro numerické výpočty použijeme podmíněnost matice.

$$J = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \cdots & \frac{\partial x}{\partial q_6} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \cdots & \frac{\partial y}{\partial q_6} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \gamma}{\partial q_1} & \frac{\partial \gamma}{\partial q_2} & \cdots & \frac{\partial \gamma}{\partial q_6} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{m}}{\partial \text{rad}} \\ \frac{\partial \text{rad}}{\partial \text{rad}} \end{bmatrix} \quad (5.6)$$

### 5.2.3 Vzdálenost od limitních poloh

Dalším problémem jsou polohy, kdy jeden nebo více kloubů dosáhne svých mechanických limitů, brání dalšímu pohybu v určitém směru. Proto přidáme vzdálenost od limitních poloh kloubů do kriteriální funkce. Vzdálenost vypočítáme jako

$$\min_i \min \left( \frac{|q_i - q_{i, \min}|}{|q_{i, \max}| + |q_{i, \min}|}, \frac{|q_i - q_{i, \max}|}{|q_{i, \max}| + |q_{i, \min}|} \right), \quad i = 1, \dots, 6. \quad (5.7)$$

Kritéria rozdělíme rozměrově na robota a pojezd.

### 5.2.4 Vzdálenost od kolize

Cesta koncového efektoru je předem definována. Pohyb robota po pojezdu má omezený vliv na schopnost vyhýbat se kolizím. Jedinou proměnnou je orientace boule kloubu 5. Vzdálenost od kolizí se do optimalizačního procesu, kvůli výpočetní náročnosti nehodí.

Pro zajištění bezkolizního pohybu je třeba vybrat vhodnou cestu a konfiguraci kloubu 5.

## 5.3 Relativní poloha

IK je implementována pouze pro šestiosý robot. Požadovanou polohu koncového efektoru je potřeba definovat v souřadnicích robota. Polohy máme definované v souřadnicích světa. Přepočítání polohy ze souřadnic světa do souřadnic robota je následující:

$$x_R = x_W - q_0 \quad (5.8)$$

$$z_R = z_W - 0.456 \quad (5.9)$$

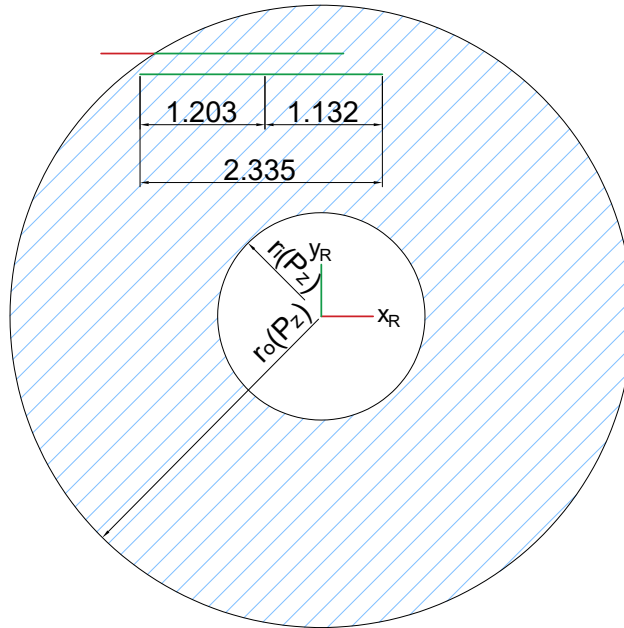
Kde  $q_0$  reprezentuje polohu robota na pojezdu, 0.456 reprezentuje výšku pojezdu.

Pohyb robota po pojezdu přemísťuje relativní polohu v souřadnicích robota po přímce paralelní s osou pojezdu. Na této přímce se délkou pojezdu vymezi úsečka o délce pojezdu, viz obrázek 5.2.

## 5.4 Limitní polohy na pojezdu

Na některé polohy robot nemusí dosáhnout z celého pojezdu. Pro každou skupinu poloh zjistíme interval ve kterém se polohu robota může pohybovat k zajištění dosažitelnosti všech poloh ve skupině. K zajištění dosažitelnosti poloh využijeme pracovní oblasti robota. Pro polohy skupiny zjistíme odpovídající polohu bodu P v souřadnicích robota. Ze  $z$  souřadnice bodu P určíme poloměry mezikruží. Pojezd vymezi úsečku paralelní s osou pojezdu o délce pojezdu. Poloha je dosažitelná na části úsečky uvnitř mezikruží pracovní oblasti.

Příklad vidíme na obrázku 5.2. Poloměry  $r_o$ ,  $r_i$  jsou proměnnými  $z$  souřadnice bodu P. Limitní polohy externí osy pro skupinu poloh získáme jako průnik intervalů úseček uvnitř mezikruží příslušných poloh skupiny. Jestliže intervaly poloh skupiny mají prázdný průnik nelze na všechny dosáhnout z jedné polohy na pojezdu. Jestliže některá poloha má dvě nespojitě části úsečky uvnitř mezikruží, jedná se o polohu příliš blízko středu pojezdu a takové polohy nejsou přípustné.



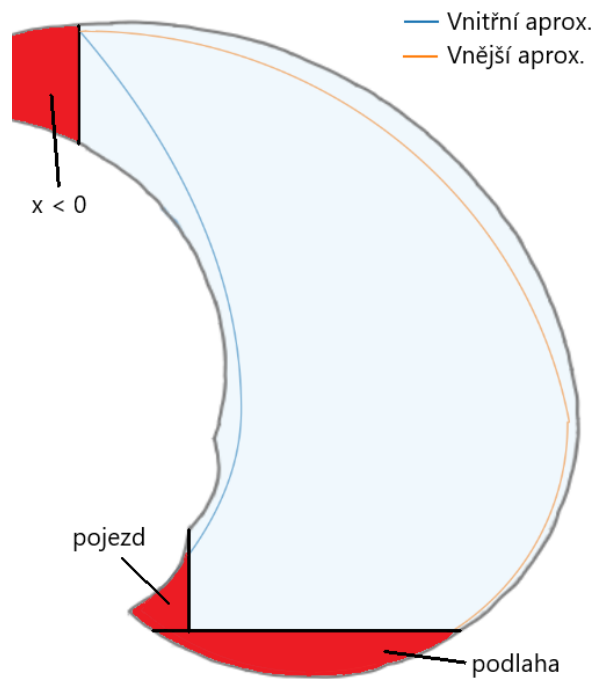
**Obrázek 5.2:** Řez pracovní oblasti robota ve výšce  $z$ , s příkladem zobrazení úsečky relativní polohy na sedmé ose. [m]

K získání poloměrů  $r_o$ ,  $r_i$  aproximujeme pracovní oblast robota křivkami, viz obrázek 5.3:

- Venkovní horní kružnice :  $(z - 0.37)^2 + (r_o - 0.0035)^2 = 9.18$
- Venkovní spodní kružnice:  $(z - 0.9148)^2 + (r_o - 1.429)^2 = 2.475$
- Vnitřní horní parabola:  $r_i = -0.379(z - 1)^2 + 1$
- Vnitřní spodní parabola:  $r_i = -0.173(z - 1)^2 + 1$

## 5.5 Modifikace inverzní kinematické úlohy

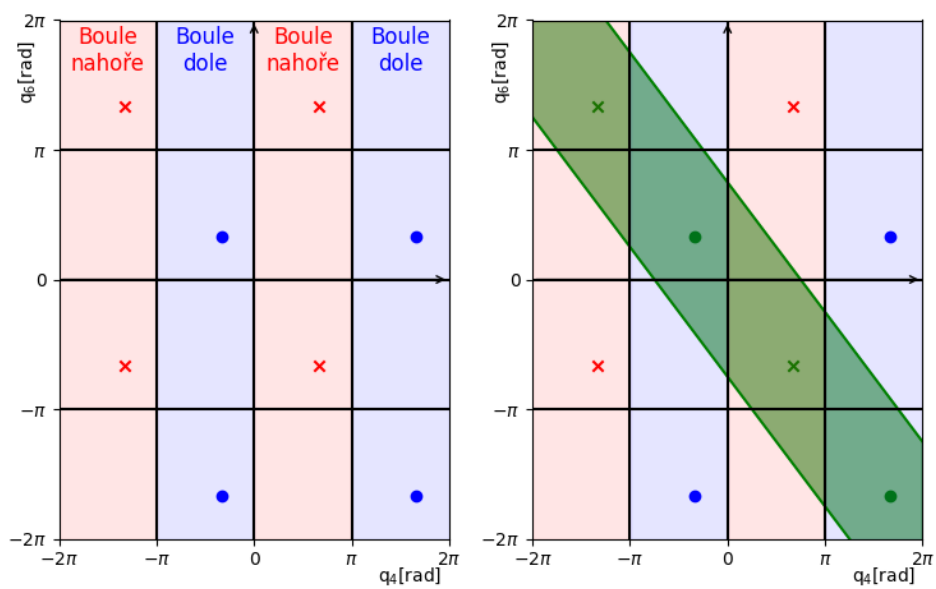
Implementovaná IK vrací pouze 2 řešení v intervalu  $(-\pi, \pi)$ , v celém rozmezí kloubů je těchto řešení více. Zobrazení všech možných řešení pro jednu polohu vidíme v grafu 5.4(vlevo). V celém rozmezí pro klouby  $q_4$  a  $q_6$  existuje 8 řešení. Chceme aby plánovaná cesta byla spojitá. Během plánování cesty v



**Obrázek 5.3:** Aproximace pracovní oblasti robota 2.2 křivkami.

jedné konfiguraci může při výjezdu z intervalu  $(-\pi, \pi)$  přeskočit řešení a robot změni konfiguraci. Aby cesta byla spojitá je potřeba zvolit nejbližší řešení k předchozí poloze robota ze všech 8 řešení. Z těchto 8 řešení splňují podmínku pro kabely 2.1 pouze 4 viz 5.4(vpravo).





**Obrázek 5.4:** Výskyt řešení IK v prostoru  $q_4$  a  $q_6$  (vlevo). Řešení IK splňující podmínku 2.1 (vpravo).



## Kapitola 6

### Implementace optimalizace

V této kapitole se budeme zabývat implementací celého projektu. Vytvoříme ROS2 službu, která bude schopna komunikovat s ostatními uzly v prostředí ROS2. Službu budeme implementovat v programovacím jazyce Python. Implementujeme ztrátovou funkci  $n$  proměnných. Pro tuto funkci vybereme vhodnou optimalizační metodu k nalezení minima.

Před začátkem optimalizace omezíme prostor proměnných podle kapitoly 5.4. Zkontrolujeme, zda jsou konfigurace pátého kloubu splnitelné. Může se stát, že požadavek na konfigurace není splnitelný, v tom případě vrátíme chybu. Také zkontrolujeme, že startovní poloha je ze souřadnice `start_extaxis` dosažitelná.

#### 6.1 Optimalizační metody

Pro nalezení minima v optimalizačním problému můžeme využít knihovnu `scipy`, konkrétně její modul `optimize`. V rámci tohoto modulu můžeme použít funkci `minimize`, která nám umožňuje provést optimalizaci s různými metodami. Jednou z těchto metod je metoda Nelder-Mead (NM). [12]

Metoda NM je jednoduchá a robustní metoda přímého prohledávání (direct search), pro hledání minima funkcí více proměnných, která nepotřebuje derivace funkce. Metody přímého prohledávání mohou skončit v lokálním minimu. Abychom minimalizovali toto riziko, je důležité vhodně vybrat počáteční řešení. Pro získání takového počátečního řešení můžeme využít metodu differential evolution (DE). [13]

Metoda DE je optimalizační algoritmus, který prohledává celý prostor a má schopnost odhadnout polohu optimální řešení. DE nezaručuje že se vždy najde optimální řešení. Pro velký prostor může tato metoda vyžadovat velký počet iterací, než dosáhne přesného odhadu.

Kombinací metod NM a DE v rámci knihovny `scipy` můžeme dosáhnout efektivního hledání minima ztrátové funkce. Používáním metody NM a správným výběrem počátečního řešení z metody DE můžeme minimalizovat riziko uvíznutí v lokálním minimu a zároveň dosáhnout optimálního řešení.

Metoda DE při prohledávání prostoru využívá náhodnosti. Pro zajištění opakovatelnosti výsledků je také vhodné nastavit buď počáteční řešení nebo konkrétní hodnotu pro `seed` randomizeru. Tím dosáhneme reprodukovatelným



### 6.2.2 Výpočet kritérií

Kritéria počítáme pro celou cestu až na konstantní startovní polohu. Jakobián vypočítáme z DH notace užitím rovnice 5.2. K výpočtu podmíněnosti translační a rotační části jakobiánu využijeme funkci `numpy.linalg.cond`. Teoretický čas vypočítáme užitím rovnice 5.1. Vzdálenost od limit je v limitě rovna 0. Abychom mohli vzdálenost od limit přidat do ztrátové funkce, budeme přidávat převrácenou hodnotu.

Pro kritéria podmíněnosti a vzdálenosti budeme minimalizovat nejhorší případ po celé cestě. Počítáme s maximální hodnotou podmíněnostmi a minimální vzdáleností od limitů.

Ztrátová funkce vrací součet

$$\sum_{i=1}^5 v_i k_i \quad (6.1)$$

kde  $v_i$  je váha  $i$ -tého kritéria a  $k_i$  je  $i$ -té kritérium. Abychom nesčítali kritéria s různými rozměry, váhy kritérií mají převrácené rozměry. Například váha pro kritérium teoretického času s rozměrem  $s$  má rozměr  $1/s$ .

## 6.3 Komunikace

V ROSu se ke komunikaci s externí službou často využívá přizpůsobené rozhraní. Toto rozhraní je definováno pomocí souboru typu `.srv`, který specifikuje strukturu požadavku (request) a odpovědi (response) pro danou službu.

```
geometry_msgs/PoseStamped[] ps_list
bool[] lin_interpolation
int32[] ps_group_sizes
float64 start_extaxis

#optional
string[] configuration
float64[] joint_state_seed
float64[] criteria_coeficients
float64[] max_joint_speeds
int32 max_feval
int32 interpolation_points

--

bool minimization_status
string status_message
float64[] solutions
```

**Obrázek 6.1:** Struktura rozhraní komunikace služby pro optimalizaci pohybu po pojezdu.

- ■ `ps list` - List poloh cesty, které má robot navštívit. Tyto polohy musejí být definovány v souřadnicích světa. První poloha reprezentuje polohu, kde robot skončil v minulé iteraci.
- `lin interpolaton` - List o délce počtu skupin. True pokud chceme skupiny propojit lineárním pohybem nebo-li kartézskou interpolací.
- `ps group sizes` - List o délce počtu skupin poloh reprezentující počty poloh přiřazené ke každé skupině. Součet se musí rovnat délce `ps list - 1`
- `start extaxis` - Poloha robota na pojezdu v první poloze (`ps_list[0]`).
- `configuration` - Požadované konfigurace boule (U, D, -) v polohách.
- `start joint states seed` - Kloubové souřadnice robota v počáteční poloze.
- `criteria coeficients` - váhy jednotlivých kritérií ve ztrátové funkci.
  1. Teoretický čas
  2. Podmíněnost translační části jakobiánu
  3. Podmíněnost rotační části jakobiánu
  4. Vzdálenost od limit kloubů robota
  5. Vzdálenost od limit pojezdu
- `max joint speeds` - Maximální rychlosti při výpočtu kritéria teoretického času.
- `max feval` - Maximální počet kroků optimalizace (evaluací ztrátové funkce), omezující délku optimalizace.
- `interpolation points` - Počet interpolovaných poloh během jízdy mezi skupinami poloh.
- ■ `minimization status` - True pokud minimalizace skončila úspěšně. False jestliže minimalizace skončila předčasně (nenalezla minimum) či s chybou.
- `status message` - Zpětná vazba minimalizace.
- `solutions` - Doporučené polohy robota na pojezdu při práci ve skupinách.

## Kapitola 7

### Experimentální výsledky

V této kapitole budeme zkoumat výsledky a proces optimalizace. Vizualizujeme řezy kritériální funkce a průběh hledání minima. Zjistíme vliv optimalizace na zrychlení pokládání cihel do zdi.

Experimenty budeme provádět s následujícím nastavením. Před začátkem stavby robot stojí v nulových kloubových souřadnicích. Na pravé straně pojezdu (ve směru kladné orientace jízdy) má k dispozici dvě palety. Obě palety jsou ve stejné vzdálenosti od pojezdu. Na levé straně pojezdu robot pokládá cihly do zdi. Na každé paletě je 80 cihel (4x4x5, šířka x hloubka x výška). Palety nazveme levou a pravou z pohledu, kde robot stojí za paletami. Robot je svým chapadlem schopen nabrat až dvě cihly najednou. Rozměry jedné cihly jsou 250 mm, 300 mm, 250 mm (šířka, délka, výška).

Každá paleta má jednu pomocnou příjezdovou polohu a dvě odjezdové, pro levou a pravou dvojici cihel. Prvně se odeberou cihly z palety pravé a následně z palety levé. Z palety se odebere nejdříve pravá dvojice cihel a při další iteraci levá dvojice. Z důvodu kolize s cihlami v nižší řadě se nabírají cihly ve schodištvém pořadí, viz tabulka 7.1.

1	3	6	10
2	5	9	14
4	8	13	17
7	12	16	19
11	15	18	20

**Tabulka 7.1:** Schodištvé pořadí nabírání cihel z palety. Pohled na palety z boku. Robot stojí vlevo.

Před položením cihly do zdi se nanese pěna. Tryska pěny je připevněna na čelní stranu pojezdu. Nanášení se provádí pohybem před tryskou. Tento pohyb se opakuje dvakrát. Poprvé se cihla zvlhčí a podruhé se nanese pěna z trysky. Nanášení pěny končí v pomocné odjezdové poloze.

Zeď se staví ve vzdálenosti 2 m od pojezdu. Zeď je 16 cihel dlouhá a 10 cihel vysoká. Každá sudá řada je posunuta doleva o polovinu šířky cihly. Každá dvojice cihel má příjezdovou polohu a odjezdovou polohu. Příjezdová poloha se nachází 35 cm blíže k pojezdu, 35 cm vpravo a 20 cm výše. Odjezdová poloha se nachází 35 cm blíže k pojezdu. Stavba zdi probíhá postupně po

řadách.

V jednom cyklu robot začíná v nulové poloze, pro první pár cihel, či odjezdové poloze minulé dvojice cihel. První skupina poloh reprezentuje nabrání cihly. Skupina obsahuje příjezdovou polohu u palety, polohu dvojice cihel v paletě a odjezdovou polohu. Pro levou dvojici cihel přidáme i pravou odjezdovou polohu. K vyhnutí se kolize s pravou paletou přidáme pro levou paletu extra skupinu obsahující pravou odjezdovou polohu pravé palety.

Další skupina reprezentuje nanášení pěny. Skupina obsahuje polohy před a za tryskou, tyto polohy jsou započítány dvakrát. Nakonec obsahuje odjezdovou polohu.

Poslední skupina reprezentuje pokládání cihly do zdi. Skupina obsahuje příjezdovou polohu k místu ve zdi, místo ve zdi a odjezdovou polohu.

Váhy všech kritérií ponecháme na hodnotě 1. Pro lineární pohyb mezi skupinami vytvoříme 5 interpolovaných bodů.

## 7.1 Řezy kritérií

Použití řezů funkcí více proměnných je nesmírně užitečné při vizualizaci těchto funkcí. Řezy nám poskytují grafickou reprezentaci funkce, což usnadňuje identifikaci extrémů. Tímto způsobem můžeme lépe porozumět vlastnostem funkce.

Vytvoříme dvourozměrné řezy pro dvě skupiny poloh. Do kritérií započítáme pouze polohy ze dvou příslušných skupin. Osy omezíme dosahem robota na polohy ve skupině, viz kapitola 5.4. Hodnotu kritérií vypočítáme každých 10 cm v rozmezí limit.

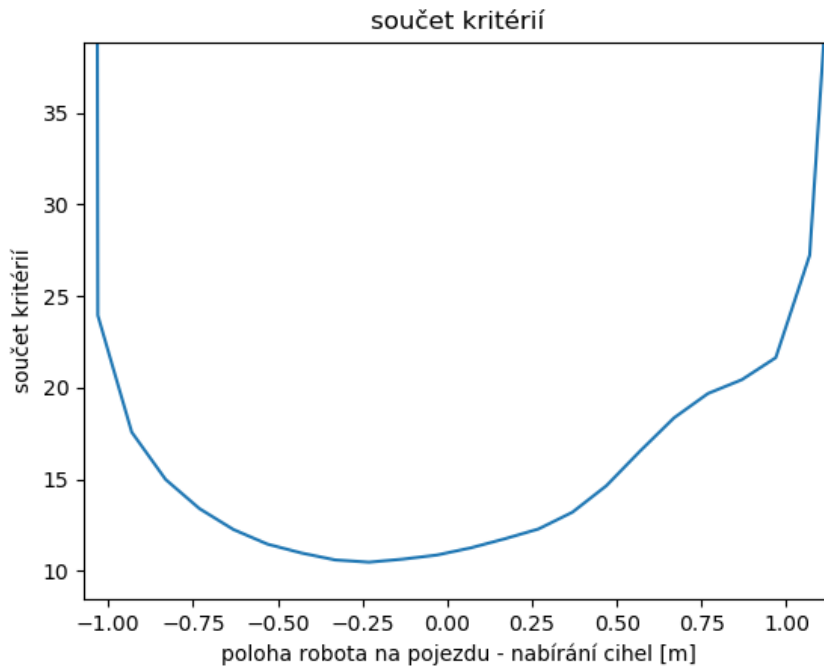
### 7.1.1 Konvexnost ztrátové funkce

Prvně se podíváme na řezy položení první dvojice cihel do zdi. Řez ze startovní polohy je jednorozměrný. Řezy součtu kritérií pro první dvojici cihel vidíme na obrázcích 7.1, 7.2, 7.3. Ze součtů 7.2, 7.3 odečteme kritérium od limit pojezdu. Ze součtu pro 7.3 odečteme také kritérium vzdálenosti od limit robota. Tyto kritéria jsou shora neomezené a ve vizualizaci zastíní ostatní kritéria.

Z analýzy řezů 7.1, 7.2 a 7.3 by se mohlo zdát, že použití metody DE předřazené metodě NM není nezbytné. Nicméně je klíčové si uvědomit, že ačkoli tyto funkce vykazují konvexní řezy, neznamená to automaticky, že jsou konvexní ve všech proměnných. Může se stát, že funkce má lokální konvexní oblasti nebo lokální extrémy v různých částech svého definičního oboru. Například řez pro dvojici cihel s pořadovým číslem 74 má pro nabírání cihel a jízdy k pravé pomocné poloze pravé palety řez 7.4, který má alespoň dva lokální extrémy. V takovém případě je vhodné použít kombinaci metod, jako je DE a NM, aby bylo dosaženo správné minimum. Z řezu 7.4 jsme opět odečetli kritéria pr vzdálenost od limit.

Rozdělení řezu na vícero lokálních konvexních oblastí je způsobeno kritériem podmíněnosti jakobiánu. Podmíněnost jakobiánu je vysoká pokud robot stojí





**Obrázek 7.1:** Řez součet kritérií pro pohyb ze startovní polohy a nabrání první dvojice cihel.

v ose některé polohy. V případě řezu 7.4 se jedná o jednu z interpolovaných poloh mezi pravou pomocnou polohou levé palety a pravé palety.

### 7.1.2 Vliv pohybu na pojezdu na rychlost pokládání cihel

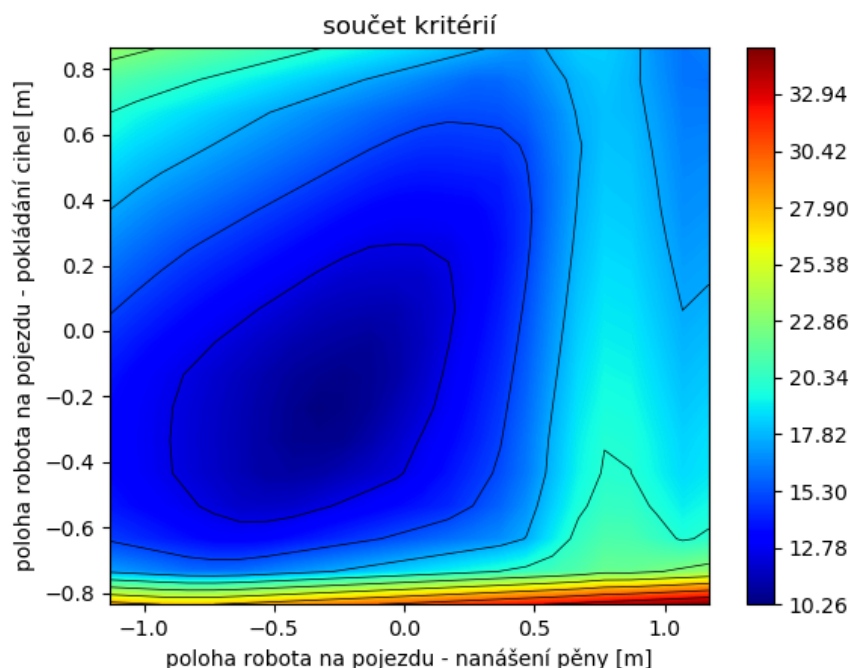
V této sekci se podíváme na kritérium teoretického času. Z řezu 7.5 můžeme říci, že vzdálený pohyb robota po pojezdu je časově velmi drahý. Rychlost pojezdu je malá a při velkých vzdálenostech zpomaluje celý pohyb robota.

Vzdálenost, o kterou se robot může pohnout, aby nezpomalovat celý proces zjistíme experimentem. Vytvoříme řez z levé pomocné polohy levé palety do pravé pomocné polohy pravé palety. Tento pohyb je paralelní s osou pojezdu a dlouhý přibližně 2.8 m. Řez vidíme na obrázku 7.6.

Z řezu 7.6 vidíme, že pro pohyby paralelní s osou pojezdu kritérium teoretického času směřuje robota ve směru pohybu. Snižuje tím relativní vzdálenost poloh pro robota. Délka ujeté vzdálenosti není konstantní a závisí na vzdálenosti od středu pojezdu.

## 7.2 Průběh optimalizace

V této sekci se podíváme na hledání minima ztrátové funkce. Použijeme kombinaci optimalizačních metod Nelder-Mead a differential evolution.



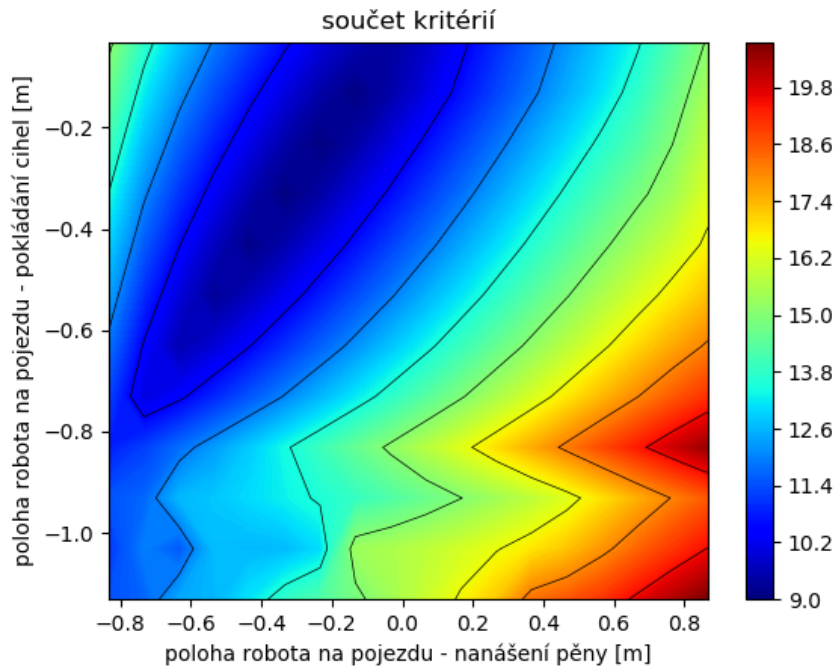
**Obrázek 7.2:** Řez součtu kritérií pro nabrání první dvojice cihel a jejich napěnění.

Pokud chceme optimalizaci provádět online během stavby je třeba optimalizaci časově omezit. Robot postaví 160 cihel za jednu hodinu. Do tohoto času jsou započítána měření před stavbou. Na jednu dvojici cihel připadá 45 s.

Notebooku s procesorem Intel® Core™i7-6600 průměrně vypočítá jeden krok optimalizace za 0.3 s. Délka výpočtu je závislá na počtu poloh, pro které se počítá IK. Z výpočtu jednoho kroku se průměrně 80% času počítá IK. Vymešíme-li maximální čas optimalizace na 15 s, maximální počet evaluací nastavíme na 50. Metodě DE omezíme parametr `popsize` na 3.

Provedeme pět experimentů pro hledání optimálních poloh pro první pár cihel.

1. Provedeme časově neomezenou optimalizaci. Použijeme kombinaci metod DE a NM. Tímto získáme přesnou polohu minima, ke kterému můžeme vztahovat výsledky optimalizací časově omezených.
2. Provedeme časově neomezenou optimalizaci samotnou metodou NM. Uvidíme zda s dostatkem času metoda NM nalezne stejné optimum.
3. Použijeme samotnou metodu NM, které omezíme čas optimalizace. Počáteční polohy zvolíme středy intervalů.
4. Použijeme samostatnou metodu DE, které omezíme čas optimalizace.
5. Metodě NM předradíme metodu DE. Čas optimalizace omezíme a každé metodě přidělíme přibližně polovinu času.



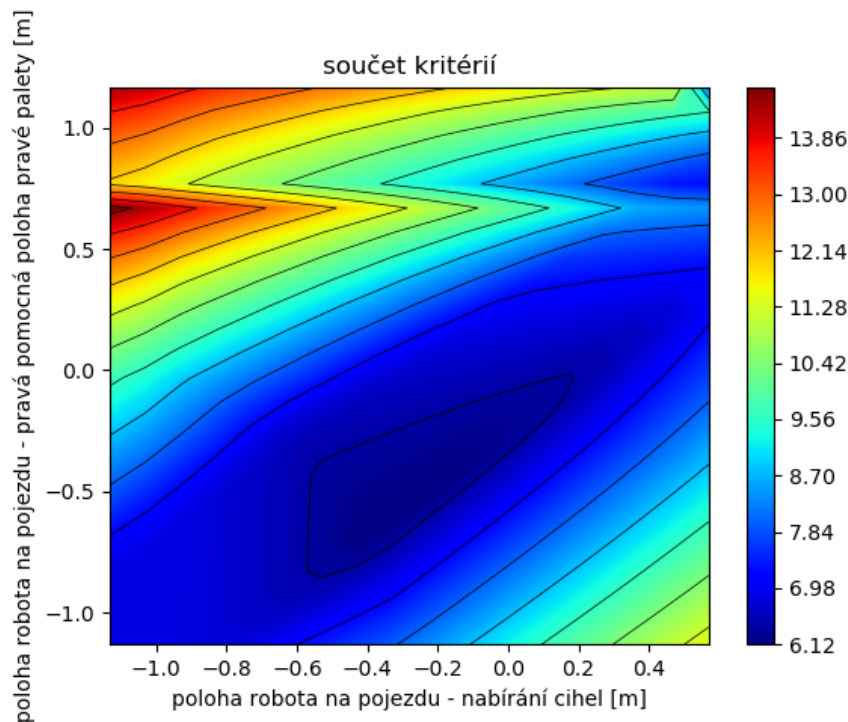
**Obrázek 7.3:** Řez součtu kritérií pro napěnění první dvojice cihel a jejich položení do zdi.

č. exp.	obrázky	ztráta	teor. čas [s]	poloha minima [m]	vzdálenost od ref. minima [m]
1	7.7, 7.8	18.00	3.86	-0.28, -0.52, -0.63	0
2	7.10, 7.10	18.00	3.87	-0.28, -0.52, -0.63	0
3	7.13, 7.13	19.33	4.64	0.25, -0.44, -0.67	0.54
4	7.14, 7.15	18.87	4.5	-0.33, -0.55, -0.62	0.06
5	7.16, 7.17	18.4	3.9	-0.28, -0.55, -0.62	0.04

**Tabulka 7.2:** Výsledky experimentů optimalizace první dvojice cihel.

Výsledné hodnoty experimentů optimalizací vidíme v tabulce 7.2.

Omezené optimalizace skončili předčasně, bez nalezení minima. Samotná metoda NM potřebuje spoustu kroků aby se v nekonvexním prostoru dostala do referenčního minima. Samotná proto není vhodná. Metoda DE se k referenčnímu minimu přiblíží nezávisle na počátečním řešení. Metoda DE je vhodná jestliže postačí být v okolí optima. Jestliže se chceme k optimální hodnotě přiblížit více, je potřeba za DE zařadit metodu NM. Je žádoucí přidat tuto možnost do rozhraní optimalizační služby.



**Obrázek 7.4:** Řez součtu kritérií pro nabírání dvojice cihel s pořadovým číslem 74 a přesunem do pravé pomocné polohy pravé palety.

## 7.3 Optimalizace celé zdi

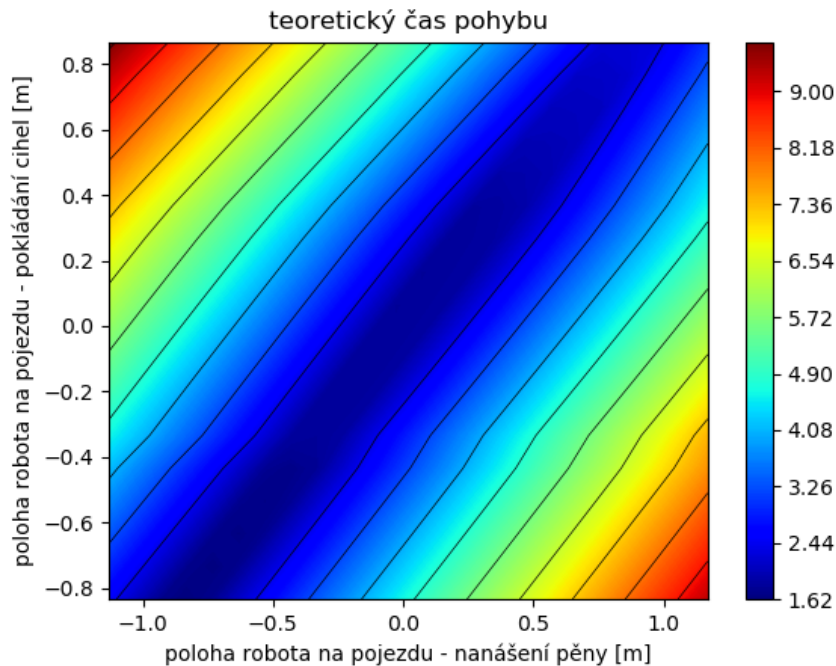
Při dočasném řešení robot stojí v nulové poloze pojezdu při nabírání cihly, nanášení pěny a při pokládání spodních čtyř řad cihel do zdi. Ve vyšších řadách se robota postaví 70 cm od osy kolmé k místu položení cihel směrem do středu pojezdu. To znamená, že pro polohu položení se souřadnicí  $x = 0.5$  m robot při pokládání cihel stojí v  $-0.2$  m. Polohy robota omezíme na interval  $(-1, 1)$  [m]. Tímto způsobem se robot vyhne singularitě.

### 7.3.1 Vliv optimalizace na čas položení cihly

Součástí optimalizace je minimalizovat čas, který trvá robotu postavit zeď. Výsledky s použitím optimalizace budeme porovnávat s dočasným řešením.

Robotem odsimulujeme pokládání páté řady cihel a porovnáme časy s použitím optimalizovaných hodnot a hodnot v dočasném řešení. Robot má různě nastavené rychlosti při jízdě trajektorie interpolované a trajektorie nalezené plánovačem v prostoru (free space planer). Abychom nezahrnuli tyto rozdílné rychlosti do výpočtu, veškeré trajektorie budeme hledat plánovačem v prostoru.

Čas položení páté vrstvy cihel s hodnotami v dočasném řešení je 234.6 s. Čas položení páté vrstvy cihel s optimalizovanými hodnotami je 206.1 s. Procentuální zrychlení je tedy 12.2%.



**Obrázek 7.5:** Řez kritérií teoretického času pro nabrání první dvojice cihel a jejich napěnění.

### 7.3.2 Vliv optimalizace na vzdálenost od singularit

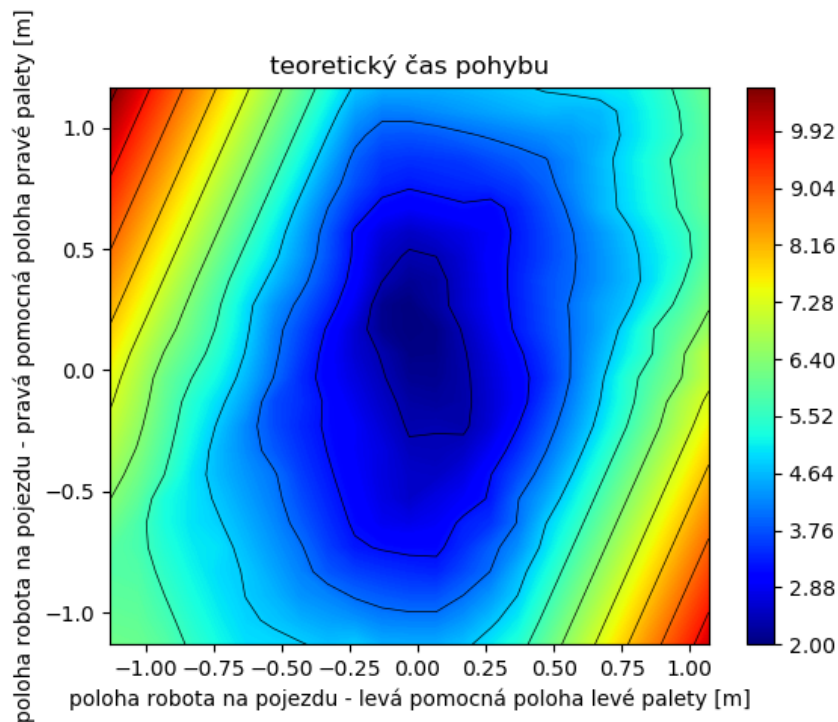
Jedna ze singularit, se kterou se robot může setkat, nastává pokud je kloub 5 v nulové a robot stojí v ose koncového efektoru. K této singularitě se robot může přiblížit při pokládání cihel do horních řad zdi.

Histogram vzdálenosti robota od os cihel vrchních pěti řad zdi vidíme na obrázku 7.18. Robot se od os singulárních poloh vzdaluje nejméně o 0.14 m.

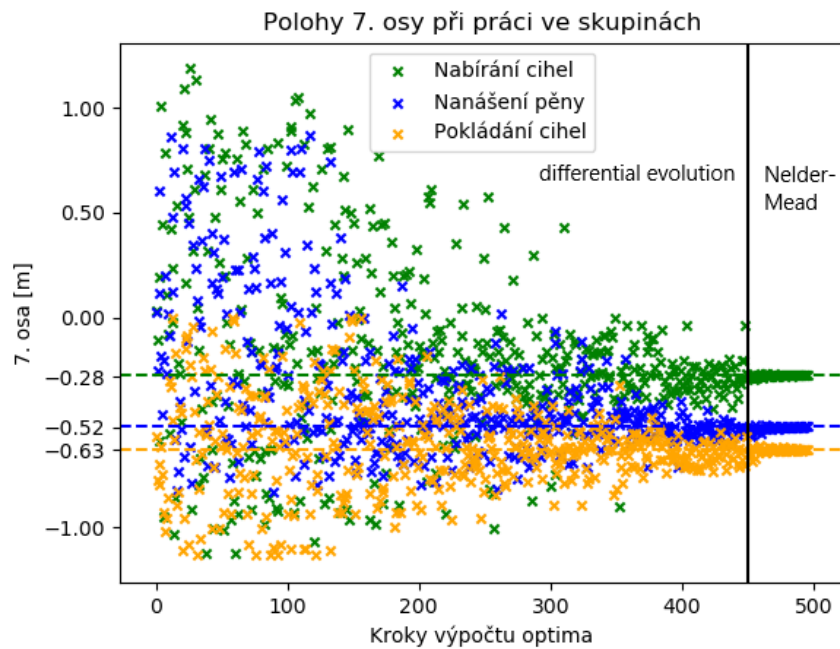
### 7.3.3 Statistika

Optimalizujeme-li celou zeď ujede se na pojezdu delší vzdálenost než při dočasném řešení o 2.5 m. Histogram poloh na pojezdu vidíme na obrázku 7.19. Histogram ujetých vzdáleností vidíme na obrázku 7.20. Z důvodu pomalé rychlosti pohybu robota na pojezdu se robot pohybuje uprostřed pojezdu a nejví sklon k překonávání velkých vzdáleností.

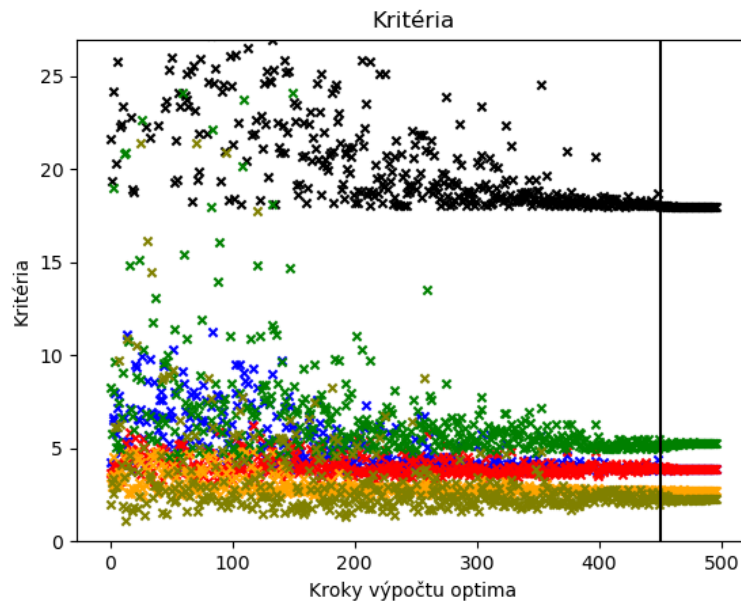
Polohy při nanášení pěny jsou konstantní pro všechny dvojice cihel. Histogram poloh robota při nanášení pěny vidíme na obrázku 7.21.



**Obrázek 7.6:** Řez kritéria teoretického času pro pohyb z levé pomocné polohy levé palety do pravé pomocné polohy pravé palety.



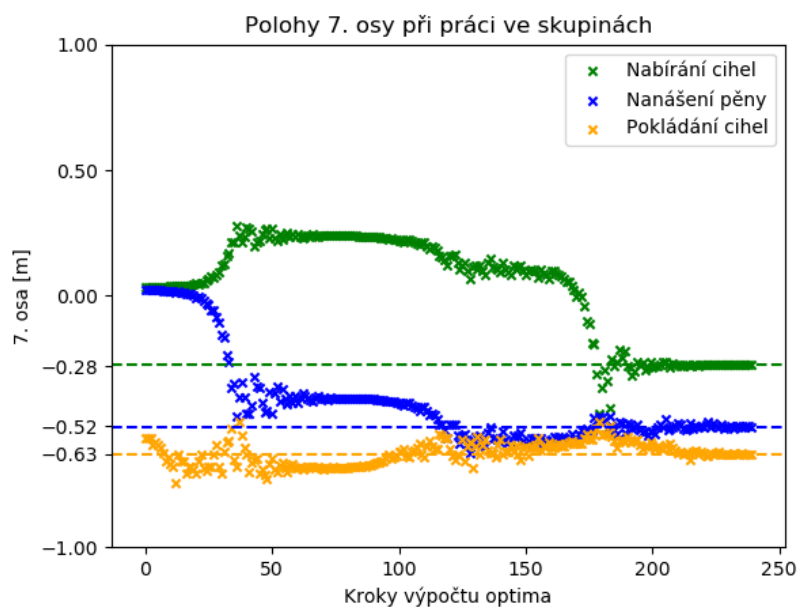
**Obrázek 7.7:** Průběh pozic robota na pojezdu časově neomezené optimalizace kombinací metod differential evolution a Nelder-Mead. Referenční minimum se nachází v -0.28 m, -0.46 m, -0.65 m.



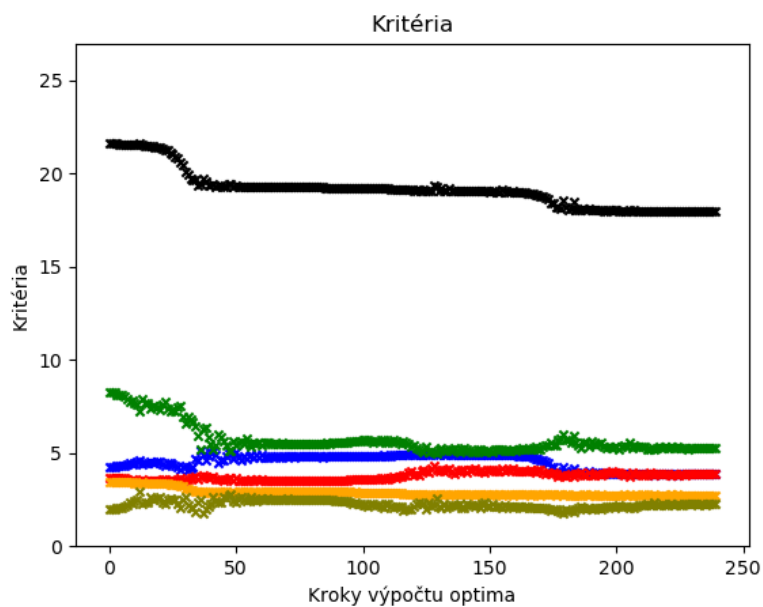
**Obrázek 7.8:** Průběh kritérií během časově neomezené optimalizace kombinací metod differential evolution a Nelder-Mead. Legenda kritérií je na obrázku 7.9.

- × součet kritérií
- × maximální hodnota podmíněnosti translační části jakobiánu po celé cestě
- × maximální hodnota podmíněnosti rotační části jakobiánu po celé cestě
- × teoretický čas pohybu
- × 1/minimální normalizovaná vzdálenost od limit robota po celé cestě
- × 1/minimální normalizovaná vzdálenost od limit pojezdu po celé cestě

**Obrázek 7.9:** Legenda kritérií

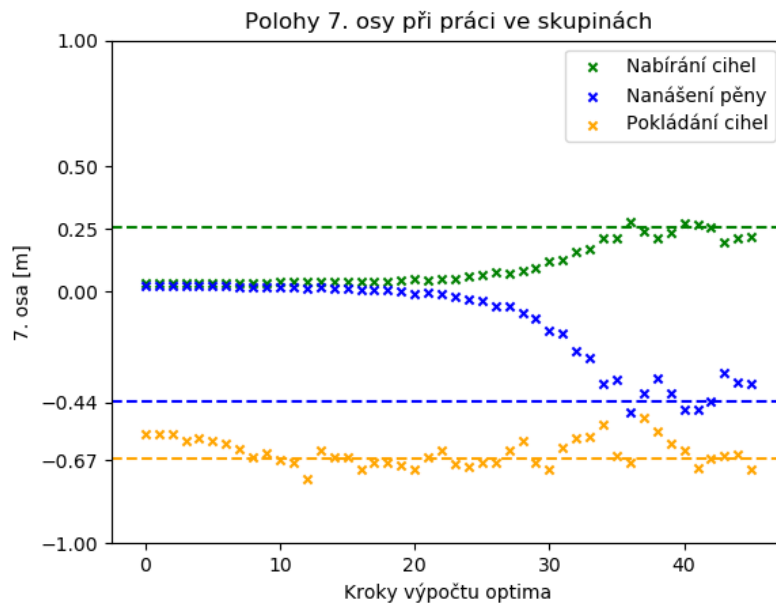


**Obrázek 7.10:** Průběh pozic robota na jezdci časově neomezené optimalizace metodou Nelder-Mead.

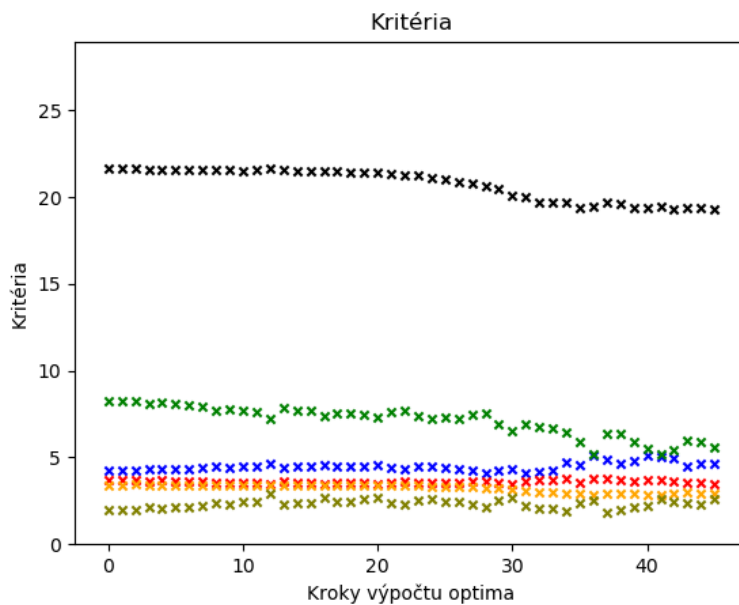


**Obrázek 7.11:** Průběh kritérií během časově neomezené optimalizace metodou Nelder-Mead. Legenda kritérií je na obrázku 7.9.

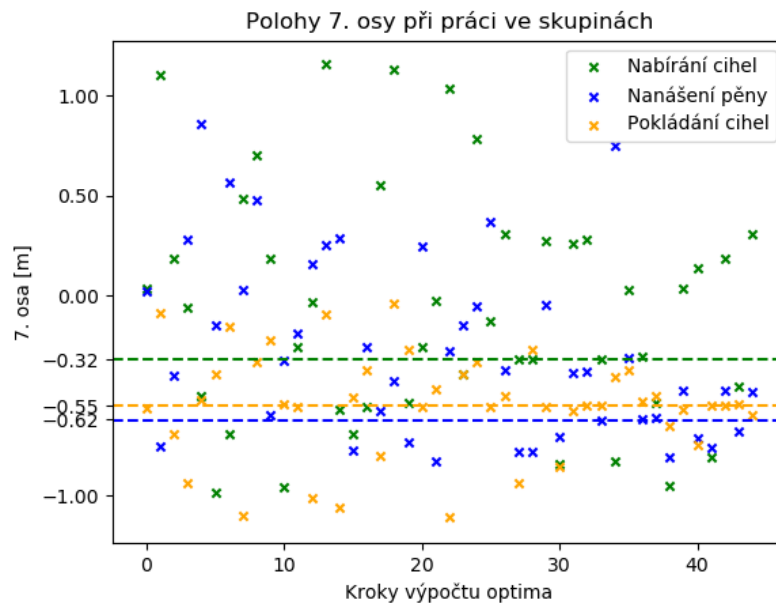




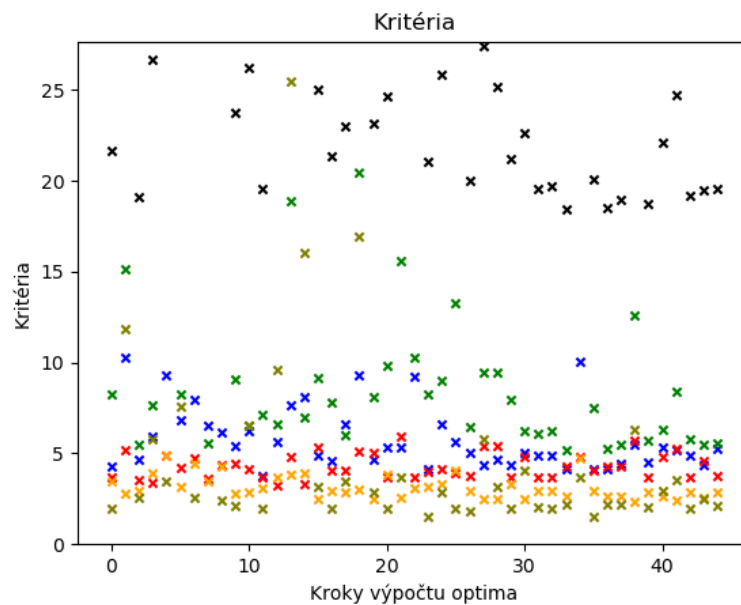
**Obrázek 7.12:** Průběh pozic robota na pojezdu časově omezené optimalizace metodou Nelder-Mead.



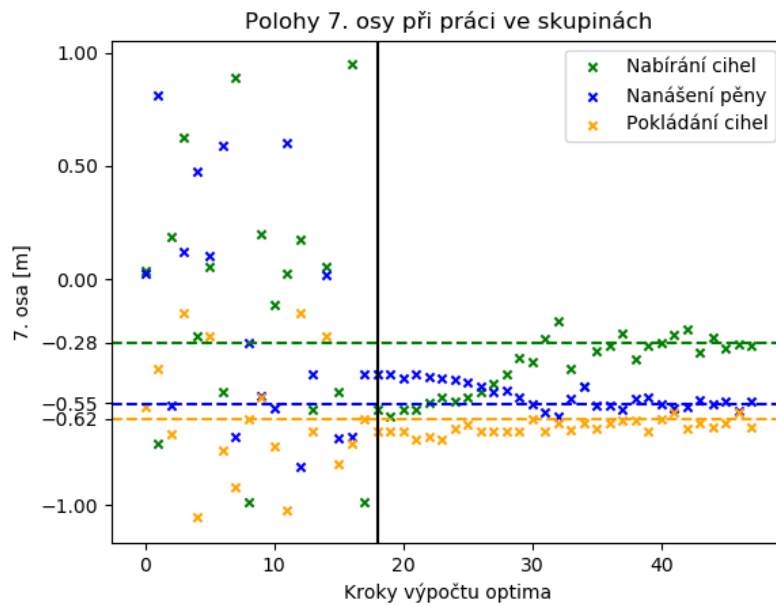
**Obrázek 7.13:** Průběh kritérií během časově omezené optimalizace metodou Nelder-Mead. Legenda kritérií je na obrázku 7.9.



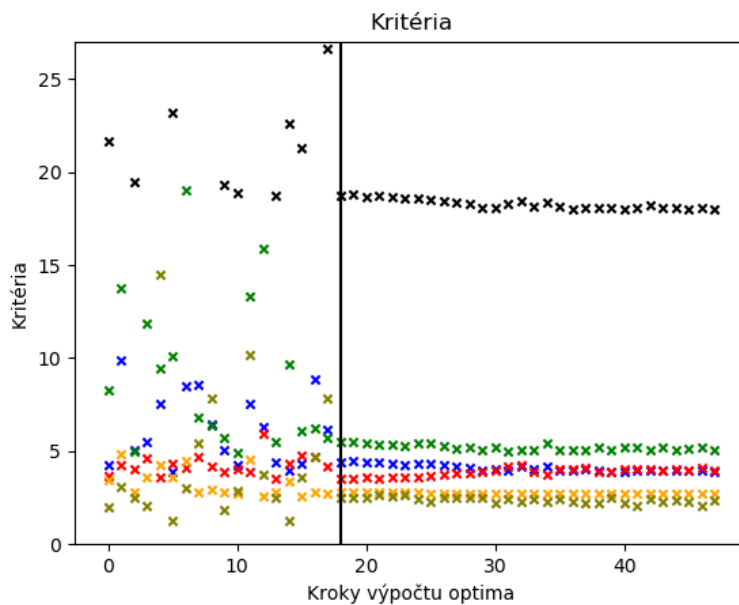
**Obrázek 7.14:** Průběh pozic robota na jezdci časově omezené optimalizace metodou differential evolution.



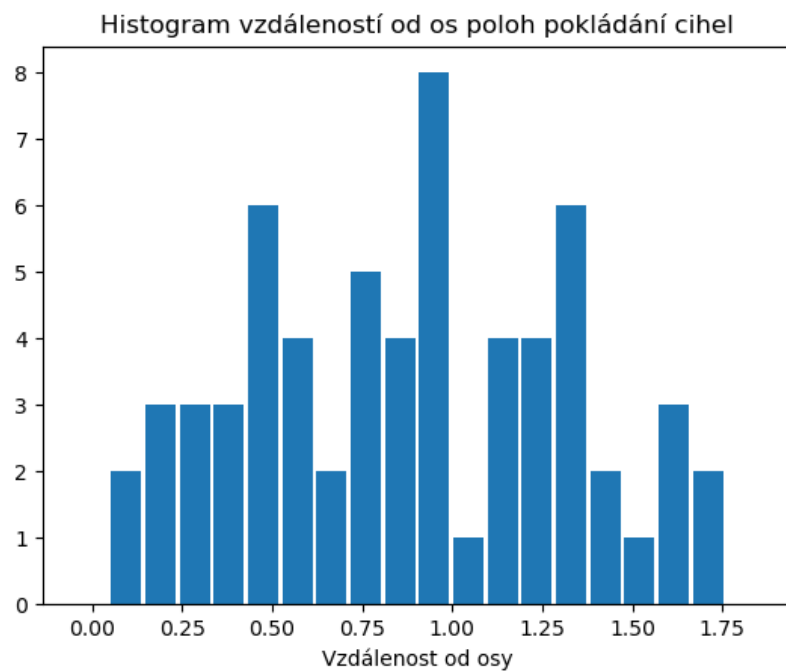
**Obrázek 7.15:** Průběh kritérií během časově omezené optimalizace metodou differential evolution. Legenda kritérií je na obrázku 7.9.



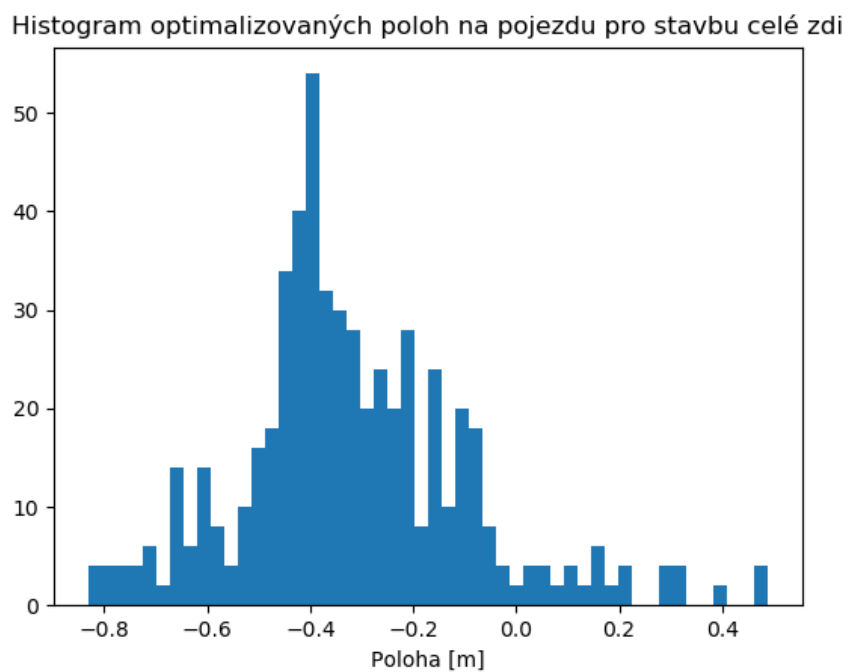
**Obrázek 7.16:** Průběh pozic robota na pojezdu časově omezené optimalizace kombinací metod differential evolution a Nelder-Mead.



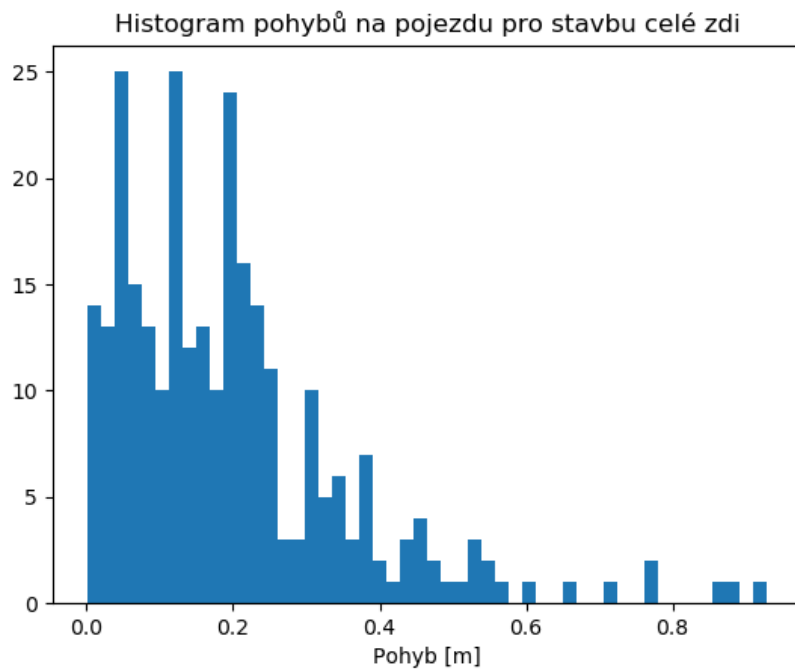
**Obrázek 7.17:** Průběh kritérií během časově omezené optimalizace kombinací metod differential evolution a Nelder-Mead. Legenda kritérií je na obrázku 7.9.



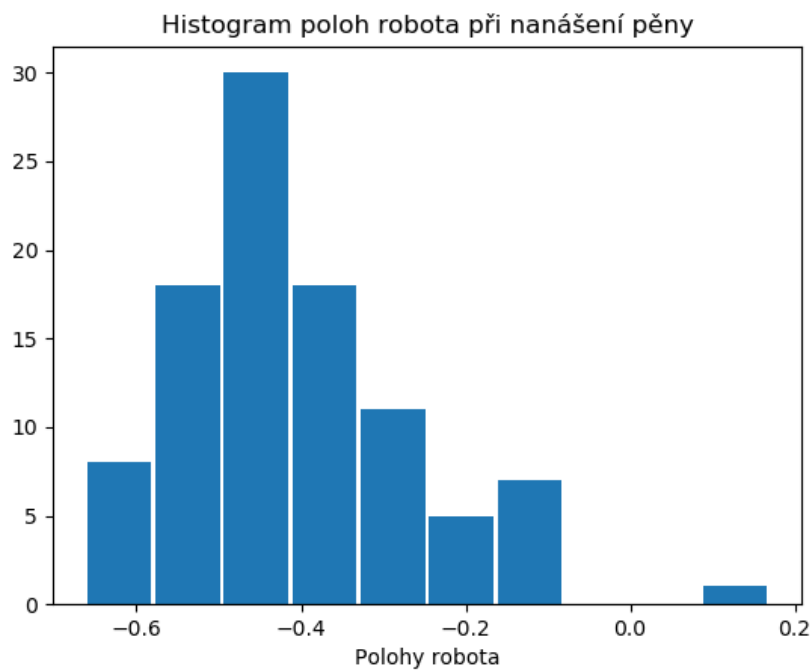
**Obrázek 7.18:** Histogram vzdálenosti robota od os poloh pokládání vrchních pěti řad zdi.



**Obrázek 7.19:** Histogram poloh robota na pojezdu během stavby celé zdi.



**Obrázek 7.20:** Histogram poloh robota na pojezdu během stavby celé zdi.



**Obrázek 7.21:** Histogram poloh robota na pojezdu během nanášení pěny.



## Kapitola 8

### Závěr

Cílem kalibrační části této práce bylo změřit přesnost robota připojeného na pojezdu a provést kalibraci modelu celé sestavy.

Změřili jsme chybu sestavy. Průměrná chyba nezkalibrované sestavy je 0.0033 m. Kalibrací jsme zpřesnili některé parametry sestavy, čímž jsme snížili průměrnou chybu na 0.0013 m. Kalibrační parametrů sestavy jsme snížili chybu robota o 60%.

Se zákazníkem jsme zhodnotili, že přesnost robotické sestavy je dostatečná pro úlohy, které vykonává. Kusová kalibrace modelu robotické sestavy je tedy nadbytečná.

Cílem optimalizační části bylo vytvořit službu v prostředí ROS2, která optimalizuje polohy robota na pojezdu v definovaných polohách cesty. Cílem bylo, aby se robot s pomocí sedmé osy vyhýbal singularitám a zrychlil pokládání cihel.

Vytvořili jsme službu s integrovanou minimalizační metodou a přizpůsobeným rozhraním pro komunikaci s klientem. Pro minimalizaci jsme vybrali proměnné polohy robota ve skupinách, ve kterých nechceme, aby se robot pohyboval na sedmé ose. Tímto snížíme případnou chybu spojenou s jízdou na ose a zmenšíme dimenzi prostoru optimalizace. Po odstranění redundance můžeme kloubové souřadnice robota dopočítat pomocí inverzní kinematiky. Problém se zjednodušil na řízení robota běžnými metodami a řízení sedmé osy optimalizací.

Pro nalezení minima jsme se rozhodli kombinovat metody differential evolution a Nelder-Mead. Metoda Nelder-Mead je známa tím, že může skončit v lokálním minimu, zatímco metoda differential evolution má schopnost prohledávat celý prostor a odhadnout polohu minima. Kombinací těchto dvou metod dosahujeme přibližného minima s menším počtem kroků.

Pokud je naším cílem pouze se přiblížit k minimu, můžeme použít metodu differential evolution samostatně.

Za kritéria k minimalizaci jsme zvolili teoretický čas vykonání cesty, vzdálenost od singularit a vzdálenost od limit kloubů. Vzdálenost od singularit počítáme jako podmíněnost jakobiánu robota.

Optimalizací se zrychlily pohyby pro stavbu zdi o 12.2%. Robot během stavby ujede na pojezdu o 2.5m větší vzdálenost. Kritérium podmíněnosti jakobiánu vzdaluje robota od singularitních poloh.

Vytvořená optimalizační metoda je pouze nástrojem. Je důležité, aby si zákazník sám vyzkoušel různé parametry, aby našel ty, které přesně odpovídají konkrétním požadavkům. Je nezbytné provést vlastní experimenty a testy, aby se dosáhlo optimálních výsledků.

Optimalizační metoda je implementována univerzálně, což znamená, že je navržena tak, aby byla použitelná pro různé procesy a úlohy. Pokud dojde v budoucnu ke změně procesu stavby, není nutné měnit samotnou implementaci metody.

Nevýhodou této metody je využití inverzní kinematiky. Výpočet inverzní kinematiky je pomalý proces a omezuje počet kroků, které optimalizace může provést. Je možné, že metody využívající pouze přímou kinematickou úlohu, mohou najít optimální hodnoty rychleji.





## Literatura

- [1] Bernard, Roger, and S. Albright, eds. Robot calibration. Springer Science & Business Media, 1993.
- [2] Z. Roth, B. Mooring and B. Ravani, "An overview of robot calibration," in IEEE Journal on Robotics and Automation, vol. 3, no. 5, pp. 377-385, October 1987, doi: 10.1109/JRA.1987.1087124.
- [3] Tourassis VD, Ang MH. Identification and Analysis of Robot Manipulator Singularities. The International Journal of Robotics Research. 1992;11(3):248-259. doi:10.1177/027836499201100307
- [4] Santolaria, J., Conte, J. & Ginés, M. Laser tracker-based kinematic parameter calibration of industrial robots by improved CPA method and active retroreflector. Int J Adv Manuf Technol 66, 2087–2106 (2013). <https://doi.org/10.1007/s00170-012-4484-6>
- [5] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. 2016. Exploring the performance of ROS2. In Proceedings of the 13th International Conference on Embedded Software (EMSOFT '16). Association for Computing Machinery, New York, NY, USA, Article 5, 1–10. <https://doi.org/10.1145/2968478.2968502>
- [6] Absolute Tracker AT403, <https://pvp-snk.ru/image/data/01Aicon/04Laser-Tracker-Systems/01Absolute-Tracker-AT403/Absolute-Tracker-AT403.pdf>, Accessed: 2023
- [7] Flyer Robot GP180-120, [https://www.cz.yaskawa.eu.com/products/robots/handling-mounting/productdetail/product/gp180-120\\_705](https://www.cz.yaskawa.eu.com/products/robots/handling-mounting/productdetail/product/gp180-120_705), Accessed: 2023
- [8] K. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PAMI-9, no. 5, pp. 698–700, 1987.
- [9] V. Smutný, V. Jelínek, M. Vetchý, P. Krsek, "Kalibrace kinematiky včetně vlivu zatížení robotu Zedník V 1.0, "Neveřejná publikace Výzkumné zprávy, 2022.

- [10] V. Smutný, Diferenciální kinematika, handout k předmětu Robotika, <http://cmp.felk.cvut.cz/cmp/courses/ROB/roblec/diffkin-notecz.pdf>, Accessed: 2023
- [11] Asada H Slotine J-JE. Robot Analysis and Control. New York N.Y: J. Wiley; 1986.
- [12] Saša Singer and John Nelder (2009) Nelder-Mead algorithm. Scholarpedia, 4(7):2928.
- [13] Kenneth V. Price, Rainer M. Storn, Jouni A. Lampinen, Differential Evolution, A Practical Approach to Global Optimization, ISBN:978-3-540-20950-8, Published: 13 December 2005, <https://doi.org/10.1007/3-540-31306-0>,
- [14] Zhenhua, W., Hui, X., Guodong, C., Rongchuan, S. and Sun, L. (2014), "A distance error based industrial robot kinematic calibration method", Industrial Robot, Vol. 41 No. 5, pp. 439-446. <https://doi.org/10.1108/IR-04-2014-0319>
- [15] Albert Nubiola, Ilian A. Bonev, "Absolute calibration of an ABB IRB 1600 robot using a laser tracker", Robotics and Computer-Integrated Manufacturing, Volume 29, Issue 1, 2013, Pages 236-245, ISSN 0736-5845, <https://doi.org/10.1016/j.rcim.2012.06.004>
- [16] S. Klanke, D. Lebedev, R. Haschke, J. Steil and H. Ritter, "Dynamic Path Planning for a 7-DOF Robot Arm,"2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006, pp. 3879-3884, doi: 10.1109/IROS.2006.281798.



## Seznam zkratk

**DDS** Data Distribution Service

**DE** diferencial evolution

**DH** Denavitova-Hartenbergova

**DOF** stupňů volnosti

**IK** Inverzní kinematická úloha

**NM** Nelder-Mead