

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA STROJNÍ

ÚSTAV MECHANIKY, BIOMECHANIKY A MECHATRONIKY  
Odbor mechaniky a mechatroniky



DIPLOMOVÁ PRÁCE

KALIBROVATELNOST MECHANISMŮ PRO KALIBRACI S MODELÝ  
CHYB ČIDEL  
CALIBRABILITY OF MECHANISMS FOR CALIBRATION WITH  
SENSOR ERROR MODELS

AUTOR: Bc. Marek Škorpil  
STUDIJNÍ PROGRAM: Aplikované vědy ve strojním inženýrství  
VEDOUCÍ PRÁCE: prof. Ing. Zbyněk Šika, Ph.D.

Praha, 2022

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Škorpil** Jméno: **Marek** Osobní číslo: **458496**  
Fakulta/ústav: **Fakulta strojní**  
Zadávací katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**  
Studijní program: **Aplikované vědy ve strojním inženýrství**  
Specializace: **Aplikovaná mechanika**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Kalibrovatelnost mechanismů pro kalibraci s modely chyb čidel**

Název diplomové práce anglicky:

**Calibrability of mechanisms for calibration with sensor error models**

Pokyny pro vypracování:

- 1) Seznamte se s kalibrací mechanismů při zohlednění modelů chyb čidel.
- 2) Seznamte se s metodami optimalizace mechanismů pro zlepšení kalibrovatelnosti.
- 3) Sestavte kinematické a kalibrační modely pro příklad rovinného a prostorového mechanismu se zohledněním modelů chyb čidel.
- 4) Zformulujte parametrizaci mechanismů pro jejich optimalizaci.
- 5) Zkoumejte ovlivnění kalibrovatelnosti rozložením kalibračních poloh a rozměry mechanismů při zohlednění modelů chyb čidel.
- 6) Proveďte optimalizaci vybraných mechanismů s kritériem kalibrovatelnosti.

Seznam doporučené literatury:

Z. Šika, V. Hamrle, M. Valášek and P. Beneš, Calibrability as additional design criterion of parallel kinematic machines, Mechanism and Machine Theory, vol. 50, p. 48–63, April 2012.  
T. Skopec, Z. Šika and M. Valášek, Calibration using adaptive model complexity for parallel and fiber-driven mechanisms, Robotica, vol. 34, no. 6, pp. 1416-1435, 2016.  
J. Krivošej, Kinematická kalibrace zohledňující rozložení chyb čidel, Bakalářská práce, ČVUT v Praze, Fakulta strojní, 2016.

Jméno a pracoviště vedoucí(ho) diplomové práce:

**prof. Ing. Zbyněk Šika, Ph.D. odbor mechaniky a mechatroniky FS**

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

\_\_\_\_\_

Datum zadání diplomové práce: **22.04.2022** Termín odevzdání diplomové práce: \_\_\_\_\_

Platnost zadání diplomové práce: \_\_\_\_\_

\_\_\_\_\_  
prof. Ing. Zbyněk Šika, Ph.D.  
podpis vedoucí(ho) práce

\_\_\_\_\_  
prof. Ing. Michael Valášek, DrSc.  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
doc. Ing. Miroslav Španiel, CSc.  
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

## **Poděkování**

Na tomto místě bych chtěl poděkovat všem, kteří mě podporovali při studiu a také těm, kteří mi poskytli podklady pro vypracování této diplomové práce. Zvláště pak děkuji panu prof. Ing. Zbyňku Šikovi, Ph.D. za odborné vedení a konzultování diplomové práce a za cenné rady, díky nimž jsem mohl tuto práci vypracovat. Ještě bych chtěl poděkovat panu Ing. Petru Benešovi, Ph.D., jehož konzultace a nápady byly neocenitelné a výrazně mi dopomohly k dokončení této práce.

## **Prohlášení**

Předkládám tímto k posouzení a obhajobě diplomovou práci zpracovanou na závěr studia na ČVUT v Praze Fakultě strojní.

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

V Praze dne: .....

.....

Podpis

## Anotační list

<b>Jméno autora:</b>	Škorpil Marek
<b>Název diplomové práce:</b>	Kalibrovatelnost mechanismů pro kalibraci s modely chyb čidel
<b>Anglický název:</b>	Calibrability of mechanisms for calibration with sensor error models
<b>Akademický rok:</b>	2021/2022
<b>Obor studia:</b>	Aplikovaná mechanika
<b>Ústav/odbor:</b>	Ústav mechaniky, biomechaniky a mechatroniky Odbor Mechaniky a mechatroniky
<b>Vedoucí diplomové práce:</b>	prof. Ing. Šika Zbyněk Ph. D
<b>Bibliografické údaje:</b>	Počet stran: 78 Počet obrázků: 60 Počet příloh: 1

**Klíčová slova:** Kinematická kalibrace, kalibrovatelnost, chyba čidla, optimalizace, minimalizace, fminsearch, genetické algoritmy, robotické rameno

**Keywords:** Kinematic calibration, calibrability, error of sensor, optimalization, minimalization, fminsearch, genetic algorithm, robotic arm

### Abstrakt:

Kalibrace je proces, při kterém se snažíme zjistit skutečné hodnoty parametrů daného mechanismu, které neznáme. Tato diplomová práce se zabývá dvěma metodami kalibrace, a to s a bez uvažování kompenzace nepřesnosti čidel, které vznikají při měření. Metody porovnáme. Dále budeme mechanismy optimalizovat. Funkcí minimalizace bude kalibrovatelnost. K tomu použijeme metodu fminsearch a genetické algoritmy. Výsledky následně porovnáme a zhodnotíme, zda je výhodné uvažovat s kalibrovatelností jako parametrem již v brzké fázi návrhu mechanismů. Vše provedeme pro 2D mechanismus a 3D mechanismus.

### Abstract:

Calibration is a process in which we find out the true values of parameters of a given mechanism that we do not know. This diploma thesis deals with the two calibration methods, with and without consideration of sensor inaccuracies compensation while we measure. We will compare these methods. Next, we will optimize the mechanisms. The minimization function will be calibrability. For this, we will use the fminsearch method and genetic algorithms. Then, we will compare the results and evaluate whether it is advantageous to consider calibrability as a parameter during early stage of mechanism design. We will do everything for 2D mechanism and 3D mechanism.

## Obsah

1. Kalibrace mechanismů .....	7
1.1 Základní algoritmus kalibrace .....	8
1.2 Kalibrace s uvažováním kompenzace chyb sensorů .....	10
1.3 Kalibrovatelnost .....	13
1.4 Optimalizace .....	16
1.4.1 Fminsearch .....	16
1.4.2 Genetické algoritmy (GA).....	18
2. Rovinný mechanismus .....	20
2.1 Popis modelu.....	20
2.2 Generování dat .....	21
2.3 Kalibrace .....	22
2.4 Výsledky kalibrace.....	25
2.4.1 Chyby všech sensorů $10^{-16}$ .....	26
2.4.2 Chyby všech sensorů $10^{-5}$ .....	28
2.4.3 Zhoršení chyby senzoru $\varphi_{34}$ o jeden a dva řády, ostatní $10^{-5}$ .....	29
2.4.4 Chyba senzoru $\varphi_{34}$ $10^{-3}$ , ostatní $10^{-16}$ .....	31
2.4.5 Porovnání kalibrovatelnosti a směrodatných odchylek.....	32
2.5 Optimalizace .....	34
2.5.1 Chyby měření všech sensorů $10^{-5}$ .....	37
2.5.2 Chyby měření senzoru $\varphi_{34}$ $10^{-3}$ .....	45
2.5.3 Chyby měření všech sensorů $10^{-5}$ – změna cílové funkce .....	51
2.6 Shrnutí výsledků .....	61
3. Prostorový mechanismus .....	63
3.1 Popis.....	63
3.2 Výsledky základní metody kalibrace a optimalizace .....	66
3.3 Výsledky pokročilé metody kalibrace a optimalizace.....	69
3.4 Shrnutí.....	72
Závěr .....	73
Bibliografie .....	76
Seznam obrázků a tabulek.....	77
Přílohy.....	78

# Úvod

Nároky na produktivitu strojů se v dnešním světě neustále zvyšují. To vede k hledání zatím nepoznaných konstrukčních řešení, vývojového procesu návrhu a výroby strojů. Dosud nejrozšířenější koncepce konstrukčního řešení, sériová kinematická struktura, často není dostačující. Pro tyto případy se jeví jako vhodná alternativa použít paralelní kinematické struktury. Výhodou paralelní kinematické struktury jsou především lepší dynamické vlastnosti, které plynou ze zachování vysoké tuhosti mechanismu při nízké hmotnosti. Problémem může být především vůle v kloubech, která vznikne nedostatečnou tuhostí. To nicméně nastává jenom v singulárních polohách a jejich blízkém okolí. Proto se tato metoda využívá v úkonech (např. dokončovací operace), které vyžadují vysokou přesnost polohy, ale už ne tak přesnost momentů nebo sil. Některé nevýhody paralelních mechanismů můžeme potlačit zavedením redundantního počtu pohonů. To klade větší nároky na řízení takového mechanismu. To nicméně použití v praxi nebrání, jelikož řídicí technologie dostupné jsou. Redundantními pohony můžeme řídit tok vnitřních sil a tím vyrušit nežádoucí efekty vůlí v kloubech. Potřebná tuhost mechanismu je zajištěna i v případě, kdy se člen redundantního mechanismu nachází v singulární poloze. Toho dosáhneme pomocí dalšího členu mechanismu, který v singulární poloze nebude. Nutná podmínka je nicméně znalost přesných kinematických rozměrů, například parametrů mechanismu nebo poloh středů kloubů. Nepřesná znalost těchto rozměrů nám způsobí více problémů. Prvním je vyšší mechanické napětí ve struktuře, které vede k rychlejšímu opotřebování stroje. Druhým je působení pohonů mechanismu „proti sobě“, čímž dochází ke ztrátě energie. Třetím, a zároveň nejdůležitějším problémem, je nepřesná poloha koncového nástroje, která může vést až k nesprávnému provedení žádané operace a potenciálnímu znehodnocení výrobního dílu. Pro výpočet kinematických rozměrů proto v některých případech musíme uvažovat i s elastickou deformací. Při přesné znalosti kinematických rozměrů jsou paralelní mechanismy potenciálně velmi přesnými stroji. Této znalosti je nicméně těžké dosáhnout, protože i při nejpřesnější výrobě vždy vznikají odchylky od výkresového rozměru. V současné době proto vzniká mnoho postupů, které se snaží zjistit co nejpřesnější skutečné rozměry stroje. Velice zajímavou variantou se jeví zjišťování rozměrů již sestaveného stroje pomocí kalibrace.

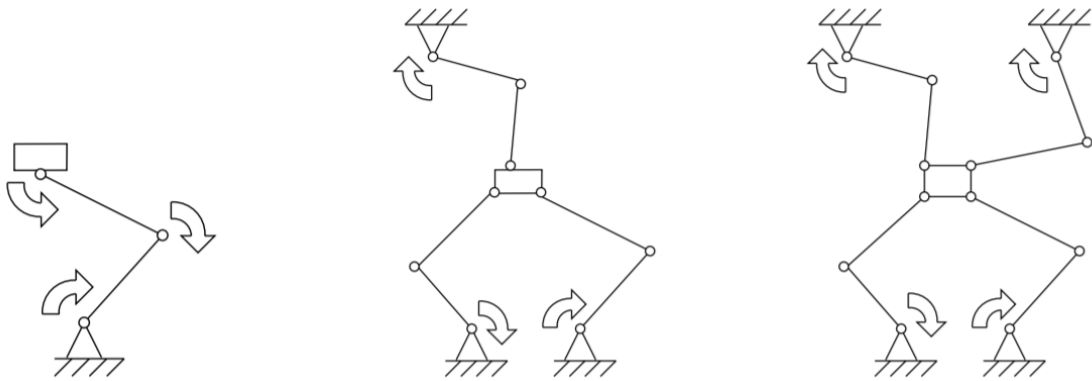
# 1. Kalibrace mechanismů

Kalibrace mechanismů je velice objemné a komplikované téma. Jejím principem je vytvoření takového nástroje, se kterým dokážeme zajistit přesnější požadovanou vlastnost nějaké součásti stroje (většinou pracovního nástroje) a tím zajistit přesnější a lepší funkčnost daného mechanismu/stroje. K tomu budeme potřebovat vytvořit nástroj, v našem případě skript v programu Matlab, který takovou schopnost mít bude. V první části diplomové práce se seznámíme s teorií, která je podkladem k vytvoření tohoto nástroje. Rozdělena bude na dvě části. První se bude zabývat kalibrací mechanismu bez uvažování kompenzace chyb sensorů a bude i základem pro část druhou. Ve druhé části již budeme uvažovat s kompenzacemi chyb sensorů. Kalibrace v praxi je však velice složitý proces. Vstupů, které na mechanismus mají vliv a tím způsobí určitou nepřesnost výsledku, je velké množství (počínaje již zmíněnými chybami nebo offsety čidel, nepřesností z výroby, gravitací, teplotní roztažností apod.). Zahrnutí všech vlivů do kalibrace je proto téměř nemožné a v této práci se budeme zabývat pouze offsety čidel.

Základní metodou kalibrace u strojů je polohování výkonného členu v celém pracovním prostoru do přesně známých poloh. Při tom měříme polohy pohonů a další souřadnice (např. souřadnice v rotačních či posuvných vazbách). Z těchto naměřených poloh pak vytvoříme přeurčenou soustavu lineárních rovnic. Neznámými jsou v této soustavě kinematické rozměry. Jak už bylo řečeno v úvodu, tak u redundantních mechanismů můžeme využít znalost poloh více pohonů a tím pádem přesněji získat polohu end-efektoru. [1]

Pokud máme zadán mechanismus, jehož počet sensorů je vyšší než počet stupňů volnosti, můžeme takový mechanismus označit jako takzvaně samo-kalibrovatelný. Čidla takového mechanismus nám v každé poloze naměří více hodnot, než kolik jich je nezbytně potřeba pro jednoznačné určení polohy. Taková čidla můžeme označit jako redundantní a pomocí nich vypočítáme kinematické rozměry mechanismu. Velkým významem samo-kalibrace je skutečnost, že se daný mechanismus může kalibrovat bez zásahu člověka a při fungování tohoto mechanismu. Můžeme tedy za běhu stroje „opravovat“ hodnoty parametrů a kompenzovat měnící se podmínky působící na stroj (např. teplota, deformace). Tím samozřejmě zpřesňujeme danou operaci, což přináší významnou úsporu času. Může to vést například k menšímu počtu zmetků, které vyrobíme. Samo-kalibrace se dá využít bez

pomoci vnějšího artefaktu. Tyto vlastnosti jsou pro výrobu velice významné a je vhodné se při návrhu nového stroje zamyslet a zauvažovat o řešení, jež samo-kalibraci obsahuje. Návrhové typy mechanismů, které jsme zmínili v úvodu, můžeme vidět na následujícím obrázku. [1]



Obr. 1: Sériová struktura

Paralelní struktura

Redundantní paralelní struktura

V této práci se budeme zabývat kalibrací dvou mechanismů. První mechanismus bude v rozměru roviny (rameno robota) obsahem druhé kapitoly. Druhý mechanismus, kterému je věnovaná třetí kapitola, bude prostorový (také rameno robota). Ukážeme si návrh mechanismů před kalibrací a po kalibraci. Porovnáme obě varianty a popíšeme si jejich benefity. Popíšeme si pojem „kalibrovatelnost“ a následně se zamyslíme nad tím, zda by se tento pojem měl stát významným členem při počátečním návrhu nového stroje.

## 1.1 Základní algoritmus kalibrace

Jak již bylo napsáno výše, v této části se budeme zabývat kalibrací, ve které nebudeme uvažovat kompenzaci chyb čidel. Budeme jí nazývat základní. Ve výpočtu budeme uvažovat pouze chyby parametrů. Tím může být například lišící se délka ramena robota ve skutečnosti oproti rozměru ve výkresové dokumentaci. Tuto metodu popsal G. Stengele a můžeme se s ní seznámit v [2], kde je popsána. Základem kalibrace je prozkoumání kinematických smyček, které jsou obsaženy v kinematické struktuře mechanismu. Smyčkami rozumíme takové cesty, které popisují pohyb mechanismu nebo jeho části a vycházejí a končí ve stejném bodě. Tyto kinematické smyčky můžeme nalézt v každém mechanismu. Mohou procházet a často také prochází koncovým pracovním bodem neboli end efektozem. Smyčky popisujeme pomocí rovnic kinematických vazeb pro danou polohu

$$\mathbf{f}(\mathbf{d}, \mathbf{s}, \mathbf{v}) = \mathbf{0}, \quad (1.1.1)$$



kde  $\mathbf{d}$  označuje parametry mechanismu,  $\mathbf{s}$  měřené vstupní hodnoty v posuvných a rotačních vazbách a  $\mathbf{v}$  popisuje výstupní naměřené hodnoty end efektoru. Jelikož máme přeúčtenou soustavu, a tedy více rovnic než neznámých, použijeme k řešení modifikovanou Newtonovu metodu upravenou pro přeúčtený systém nelineárních rovnic, která splňuje rovnice kinematických vazeb (1.1.1) pro rozsáhlé množství naměřených pozic mechanismu. Počet naměřených pozic mechanismu označme  $j = 1, \dots, n$ . Rovnici (1.1.1) přepíšeme do tvaru

$$\mathbf{F}(\mathbf{d}, \mathbf{S}, \mathbf{V}) = \mathbf{0}, \quad (1.1.2)$$

kdy  $j$ -tá pozice je z (1.1.2) popsána jako  $\mathbf{f}_j(\mathbf{d}, \mathbf{s}_j, \mathbf{v}_j) = 0$ . Vstupní informace do rovnice (1.1.2) si můžeme představit jako sloupcové matice ve tvaru  $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n]^T$ ,  $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n]^T$  a  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]^T$ .

Budeme vycházet z toho, že rozměry  $\mathbf{d}$  jsou pro všechny pozice mechanismu neměnné. Zároveň si zavedeme parametr  $\bar{\mathbf{d}}$ , který bude představovat rozměry ve výkresové dokumentaci. Skutečné rozměry, které naměříme, se od výkresových rozměrů budou lišit z nám známých důvodů. Newtonovu metodu kalibrace pak popíšeme Taylorovým polynomem ve tvaru

$$\mathbf{F}(\bar{\mathbf{d}}, \mathbf{S}, \mathbf{V}) + \mathbf{J}_d \partial \mathbf{d} + \dots = \mathbf{0}, \quad (1.1.3)$$

kde  $\mathbf{J}_d$  označuje Jacobiho matici. Jacobiho matice představuje parciálně derivované rovnice kinematických vazeb podle parametrů  $\mathbf{d}$ . Rovnici (1.1.3) pak můžeme přepsat do tvaru

$$\mathbf{J}_d \partial \mathbf{d} = -\mathbf{F}(\bar{\mathbf{d}}, \mathbf{S}, \mathbf{V}) = \partial \mathbf{r}, \quad (1.1.4)$$

kdy  $i$ -tou iterací této rovnice můžeme napsat ve tvaru

$$\partial \mathbf{d}_i = (\mathbf{J}_{di}^T \mathbf{J}_{di})^{-1} \mathbf{J}_{di}^T \partial \mathbf{r}_i. \quad (1.1.5)$$

$\partial \mathbf{d}_i$  představuje opravu parametru  $\mathbf{d}$ .  $\mathbf{J}_{di}$  je Jacobiho matice a  $\partial \mathbf{r}_i$  reprezentuje vektor odchylek vypočítaný při použití kalibrovaných rozměrů  $\mathbf{d}_i$  z předchozí iterace. Pro počáteční odhad v nulté iteraci nám mohou posloužit rozměry z výkresové dokumentace. Novou hodnotu rozměrů následně vypočítáme následující rovnicí ve tvaru

$$\mathbf{d}_{i+1} = \mathbf{d}_i + \partial \mathbf{d}_i. \quad (1.1.6)$$

Tento postup opakujeme, dokud kalibrace nezkonverguje.

## 1.2 Kalibrace s uvažováním kompenzace chyb senzorů

V této části diplomové práce popíšeme kalibraci při uvažování chyb senzorů. Čerpat budeme z, kterou popsali J. M. Hollerbach a kolegové. Popsána je v [3]. Metoda je implicitní. To znamená, že nebudeme rozlišovat mezi vnitřními a vnějšími naměřenými hodnotami.

Kinematický model mechanismu popíšeme ve tvaru

$$\mathbf{f}(\mathbf{x}, \mathbf{p}) = \mathbf{0} \quad \mathbf{f}: R^k \times R^n \rightarrow R^m, \quad (1.2.1)$$

kde  $\mathbf{x} \in R^k$  je vektor proměnných pohybu a  $\mathbf{p} \in R^n$  je vektor kalibrovaných parametrů. Vektor  $\mathbf{x}$  obsahuje například jak měřené hodnoty v kloubech nebo polohu end efektoru, tak i malé ale neznámé odchylky posunutí nebo vůle. Pro zajištění uzavřenosti kinematických smyček u této metody požadujeme, aby derivace  $\mathbf{f}$  měla hodnotu  $m$ , tedy  $\text{rank}(\partial \mathbf{f} / \partial \mathbf{x}) = m$ , a aby  $k \geq m$ . Dalším předpokladem je použití pouze naměřených poloh mechanismu. Nezměřené pozice z kinematických rovnic vyloučíme a nadále je nebudeme uvažovat a ani s nimi pracovat.

Nyní zjistíme z různých poloh mechanismu vektor  $\mathbf{x}$ . Nechť  $\mathbf{x}_i$  je hodnota  $\mathbf{x}$  v  $i$ -té iteraci. Pak píšeme, že  $\mathbf{x}_i = \bar{\mathbf{x}}_i + \hat{\mathbf{x}}_i$ . Přesnou hodnotu označíme  $\bar{\mathbf{x}}_i$  a hodnotu chyby měření (takzvaná zašuměná data) označíme jako  $\hat{\mathbf{x}}_i$ . Parametry mechanismu jsou samozřejmě konstantní ve všech pozicích. Zavedeme si také počáteční (výkresovou) hodnotu  $\bar{\mathbf{p}}$  a chybu od této hodnoty  $\hat{\mathbf{p}}$ , která vznikne například při výrobě. Psát proto budeme  $\mathbf{p} = \bar{\mathbf{p}} + \hat{\mathbf{p}}$ . Rovnici (1.2.1) pak přepíšeme pro  $i$ -tou iteraci do tvaru

$$\mathbf{f}(\mathbf{x}_i, \mathbf{p}) = \mathbf{f}(\bar{\mathbf{x}}_i + \hat{\mathbf{x}}_i, \bar{\mathbf{p}} + \hat{\mathbf{p}}) = \mathbf{0}, \quad \mathbf{i} = \mathbf{1}, \dots, \mathbf{N} \quad (1.2.2)$$

kde  $N$  je počet naměřených poloh mechanismu.

Cílem této kalibrace je zpřesnění kinematického modelu. Toho dosáhneme zlepšením hodnoty  $\hat{\mathbf{p}}$ . Ze všech hodnot  $\hat{\mathbf{p}}$  a  $\hat{\mathbf{x}}_i$  vybereme takové, které splňují vazbové rovnice a vybereme nejpravděpodobnější kombinaci, která může nastat. Toho docílíme statisticky pomocí Gaussova šumu  $\boldsymbol{\eta}$ , který má střední hodnotu rovnou nule, a kovariance  $\boldsymbol{\Sigma}$ , jejíž hustota pravděpodobnostní funkce je úměrná  $e^{-\boldsymbol{\eta}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta} / 2}$ . Maximalizace této funkce je pak to samé, co minimalizace  $\boldsymbol{\eta}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}$ . Dále budeme předpokládat, že chyby v naměřených polohách mechanismu  $\hat{\mathbf{x}}_i$  a chyby parametrů  $\hat{\mathbf{p}}$  jsou navzájem nezávislé. Gaussov šum se střední hodnotou rovnou nule a kovariancemi popisujeme

$Var(\hat{\mathbf{x}}_i) = E(\hat{\mathbf{x}}_i \hat{\mathbf{x}}_i^T) = \Sigma_x$  a  $Var(\hat{\mathbf{p}}_i) = \Sigma_p$ . Maximalizaci pravděpodobnosti pak můžeme psát ve tvaru

$$\chi^2 = \sum_{i=1}^N \hat{\mathbf{x}}_i^T \Sigma_x^{-1} \hat{\mathbf{x}}_i + \hat{\mathbf{p}}^T \Sigma_p^{-1} \hat{\mathbf{p}}. \quad (1.2.3)$$

Tato rovnice zároveň splňuje rovnici (1.2.3). Můžeme vidět, že nulové střední hodnoty je dosaženo poměrně jednoduše. Známou chybu měření odečteme, zatímco neznámou chybu můžeme považovat za parametr. Pro zjednodušení budeme uvažovat kovarianční matice  $\Sigma_x$  stejnou pro všechny iterace  $i$  a budeme jí proto psát v tomto tvaru.

Problém minimalizace funkce můžeme napsat jako Lagrandian v následujícím tvaru

$$\mathcal{L} = \chi^2 + \sum_{i=1}^n \lambda_i^T \mathbf{f}(\bar{\mathbf{x}}_i + \hat{\mathbf{x}}_i, \bar{\mathbf{p}} + \hat{\mathbf{p}}), \quad (1.2.4)$$

kde  $\lambda_i$  značí Lagrangeovy multiplikátory. První derivací rovnice (1.2.4) získáme extrémální rovnice pro minimalizaci. Pokud bychom řešili rovnici Newtonovou metodou, potřebovali bychom i druhé derivace  $\mathbf{f}$  a řešení by konvergovala kvadraticky. Tento postup nám usnadní řešení tak, že nebudeme muset získat druhou derivaci  $\mathbf{f}$ , ale bude postačovat pouze první derivace. Důsledkem nicméně bude pomalejší (lineární) konvergence. Podmínky konvergence obou postupů jsou stejné. To může být dokázáno.

Zaveďme normalizované odchylky proměnných  $\mathbf{y}_i \in R^k$  a  $\mathbf{q} \in R^n$ , které mají jednotkové kovariance  $Var(\mathbf{y}_i) = E$  a  $Var(\mathbf{q}) = E$ . Dále určíme, že

$$\hat{\mathbf{x}}_i = \Sigma_x^{1/2} \mathbf{y}_i, \quad \hat{\mathbf{p}} = \Sigma_p^{1/2} \mathbf{q}, \quad (1.2.5)$$

kde  $\Sigma_x^{1/2}$  a  $\Sigma_p^{1/2}$  jsou diagonální matice, jejíž členy jsou směrodatné odchylky, a prvky  $\hat{\mathbf{x}}_i$  a  $\hat{\mathbf{p}}$  jsou vzájemně nezávislé (platí v běžných případech).  $1/2$  představuje samozřejmě druhou odmocninu. Rovnice (1.2.5) substituujeme do (1.2.3) a dostáváme tvar

$$\chi^2 = \sum_{i=1}^N \mathbf{y}_i^T \mathbf{y}_i + \mathbf{q}^T \mathbf{q}, \quad (1.2.6)$$

kde si můžeme všimnout, že  $\mathbf{y}_i$  a  $\mathbf{q}$  jsou bezrozměrné. Počáteční odhady zvolíme  $\mathbf{y}_i = \mathbf{0}$  a  $\mathbf{q} = \mathbf{0}$  a snažíme se nalézt takové opravy  $\Delta \mathbf{y}_i$  a  $\Delta \mathbf{q}$ , abychom minimalizovali následující rovnici

$$\sum_{i=1}^N (\mathbf{y}_i + \Delta \mathbf{y}_i)^T (\mathbf{y}_i + \Delta \mathbf{y}_i) + (\mathbf{q} + \Delta \mathbf{q})^T (\mathbf{q} + \Delta \mathbf{q}) \quad (1.2.7)$$

při splnění vazbových podmínek

$$\mathbf{J}_{y_i} \Delta \mathbf{y}_i + \mathbf{J}_{q_i} \Delta \mathbf{q}_i = -\mathbf{f}(\bar{\mathbf{x}}_i + \Sigma_{\mathbf{x}}^{1/2} \mathbf{y}_i, \bar{\mathbf{p}} + \Sigma_{\mathbf{p}}^{1/2} \mathbf{q}), \quad (1.2.8)$$

$$\mathbf{i} = 1, \dots, N$$

kde  $\mathbf{J}_{y_i}$  a  $\mathbf{J}_{q_i}$  jsou matice parciálních derivací, jejichž tvar obdržíme pomocí následujících dvou rovnic.

$$\mathbf{J}_{y_i} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} (\bar{\mathbf{x}}_i + \Sigma_{\mathbf{x}}^{1/2} \mathbf{y}_i, \bar{\mathbf{p}} + \Sigma_{\mathbf{p}}^{1/2} \mathbf{q}) \Sigma_{\mathbf{x}}^{1/2} \quad (1.2.9)$$

$$\mathbf{J}_{q_i} = \frac{\partial \mathbf{f}}{\partial \mathbf{p}} (\bar{\mathbf{x}}_i + \Sigma_{\mathbf{x}}^{1/2} \mathbf{y}_i, \bar{\mathbf{p}} + \Sigma_{\mathbf{p}}^{1/2} \mathbf{q}) \Sigma_{\mathbf{p}}^{1/2} \quad (1.2.10)$$

Naměřené a normalizované chyby  $\mathbf{y}_i$  sepíšeme do vektoru  $\mathbf{y} = (\mathbf{y}_1^T \dots \mathbf{y}_N^T)^T$  o velikosti  $Nk \times 1$ . Rezidua funkce  $\mathbf{f}(\mathbf{x}_i, \mathbf{p})$  sepíšeme podobně do jednoho vektoru  $\mathbf{F}$  o velikosti  $Nm \times 1$ . Kritérium minimalizace popsanou rovnicemi (1.2.7) a (1.2.8)

$$(\mathbf{y} + \Delta \mathbf{y})^T (\mathbf{y} + \Delta \mathbf{y}) + (\mathbf{q} + \Delta \mathbf{q})^T (\mathbf{q} + \Delta \mathbf{q}), \quad (1.2.11)$$

kde chceme minimalizovat  $\Delta \mathbf{y}$  a  $\Delta \mathbf{q}$  a zároveň musíme vyhovět vazbovým podmínkám

$$\mathbf{J}_y \Delta \mathbf{y} + \mathbf{J}_q \Delta \mathbf{q} = -\mathbf{F}, \quad (1.2.12)$$

kdy matice Jacobiánů  $\mathbf{J}_y$  a  $\mathbf{J}_q$  jsou stejné s rovnicí (1.2.8). Nyní využijeme

QR-dekompozici. Principem QR-dekompozice je substituce  $\mathbf{QR} = \mathbf{J}_y^T$  do rovnice (1.2.11), jejíž tvar po úpravě je

$$\mathbf{Q}^T \Delta \mathbf{y} + \mathbf{D} \Delta \mathbf{q} = -\mathbf{R}^T \mathbf{F}, \quad (1.2.13)$$

kde  $\mathbf{D} = \mathbf{R}^{-T} \mathbf{J}_q$ . Matice  $\mathbf{Q}$  je matice ortogonální ( $\mathbf{Q}^T = \mathbf{Q}^{-1}$ ) o velikosti  $Nk \times Nm$ , kdy  $k \geq m$ . Dále můžeme napsat, že matice  $\mathbf{Q}'$  má rozměr  $Nk \times N(k - m)$  a zároveň platí, že  $(\mathbf{Q}\mathbf{Q}')$  je také matice ortogonální.

Minimalizační funkci (1.2.11) můžeme přepsat do tvaru

$$\|\mathbf{Q}^T (\mathbf{y} + \Delta \mathbf{y})\|^2 + \|\mathbf{Q}'^T (\mathbf{y} + \Delta \mathbf{y})\|^2 + \|\mathbf{q} + \Delta \mathbf{q}\|^2, \quad (1.2.14)$$

kteřou můžeme následně zjednodušit. Člen  $\mathbf{Q}^T (\mathbf{y} + \Delta \mathbf{y})$  se nenachází v rovnicích vazeb, a proto platí:

$$\mathbf{Q}^T (\mathbf{y} + \Delta \mathbf{y}) = \mathbf{0}. \quad (1.2.15)$$

Rovnici (1.2.13) vynásobíme členem  $\mathbf{Q}^T \mathbf{y}$ , přeuspořádáme a získáme

$$\mathbf{Q}^T (\mathbf{y} + \Delta \mathbf{y}) = \mathbf{Q}^T \mathbf{y} - \mathbf{D} \Delta \mathbf{q} - \mathbf{R}^T \mathbf{F}. \quad (1.2.16)$$

Do rovnice (1.2.14) následně substituujeme (1.2.15) a (1.2.16) a získáme tvar kritéria minimalizace v podobě

$$\|\mathbf{Q}^T \mathbf{y} - \mathbf{D} \Delta \mathbf{q} - \mathbf{R}^T \mathbf{F}\|^2 + \|\mathbf{q} + \Delta \mathbf{q}\|^2. \quad (1.2.17)$$

Soustavu těchto rovnic řešíme pomocí metody nejmenších čtverců v lineárním systému, kdy budeme opět minimalizovat. Tentokrát budeme minimalizovat členy

$\mathbf{Q}^T \mathbf{y} - \mathbf{D} \Delta \mathbf{q} - \mathbf{R}^T \mathbf{F} = \mathbf{0}$  a  $\mathbf{q} + \Delta \mathbf{q} = \mathbf{0}$ . Řešení hledáme ve tvaru

$$\begin{pmatrix} \mathbf{D} \\ \mathbf{I} \end{pmatrix} \Delta \mathbf{q} = \begin{pmatrix} \mathbf{D} \Delta \mathbf{q} - \mathbf{R}^T \mathbf{F} \\ -\mathbf{q} \end{pmatrix}. \quad (1.2.18)$$

Rovnice (1.2.16) a (1.2.17) vynásobíme maticí  $(\mathbf{Q}\mathbf{Q}^T)$  z obou stran zleva, sloučíme do jedné rovnice a získáme tvar

$$\mathbf{y} + \Delta \mathbf{y} = \mathbf{Q}(\mathbf{Q}^T \mathbf{y} - \mathbf{D} \Delta \mathbf{q} - \mathbf{R}^T \mathbf{F}). \quad (1.2.19)$$

Tento tvar rovnice je finální a získáme z něj opravu měřených parametrů  $\Delta \mathbf{y}$ . Z rovnice (1.2.18) získáme pro změnu opravu kalibrovaných parametrů  $\Delta \mathbf{q}$ .

Postup metody ve výpočetním programu bude následující. Nejprve si vypočítáme Jacobiho matice  $\mathbf{J}_{y_i}$  a  $\mathbf{J}_{q_i}$  a rezidua funkce  $\mathbf{f}(\mathbf{x}_i, \mathbf{p})$ , které jsme označili dříve  $\mathbf{F}$ . Následně zavedeme QR-dekompozici ve tvaru

$$\mathbf{Q}_i \mathbf{R}_i = \mathbf{J}_{y_i}^T. \quad (1.2.20)$$

Zavedeme matici  $\mathbf{D}$ , která se vypočítá v každém iteračním kroku  $i$  způsobem

$$\mathbf{D}_i = \mathbf{R}_i^{-1} \mathbf{J}_{q_i}. \quad (1.2.21)$$

Dále vypočítáme matice  $\mathbf{E}$ , jejíž hodnotu vypočítáme v iteračním kroku  $i$  podle

$$\mathbf{E}_i = \mathbf{Q}_i^T \mathbf{y}_i - \mathbf{R}_i^{-T} \mathbf{f}(\bar{\mathbf{x}}_i + \sum_{\mathbf{x}}^{1/2} \mathbf{y}_i, \bar{\mathbf{p}} + \sum_{\mathbf{p}}^{1/2} \mathbf{q}) \quad (1.2.22)$$

Abychom vypočítali odchylku  $\Delta \mathbf{q}$ , sepíšeme rovnice do soustavy ve tvaru

$$\begin{pmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_N \\ \mathbf{I} \end{pmatrix} \Delta \mathbf{q} = \begin{pmatrix} \mathbf{E}_1 \\ \vdots \\ \mathbf{E}_N \\ -\mathbf{q} \end{pmatrix} \quad (1.2.23)$$

Odkud  $\Delta \mathbf{q}$  s pomocí metody nejmenších čtverců vypočítáme. Dalším a zároveň posledním krokem je výpočet odchylky proměnných  $\Delta \mathbf{y}_i$ . Ty získáme pomocí

$$\mathbf{y}_i + \Delta \mathbf{y}_i = \mathbf{Q}_i(\mathbf{E}_i - \mathbf{D}_i \Delta \mathbf{q}). \quad (1.2.24)$$

Následuje pouze přepis zlepšených hodnot chybových proměnných  $\mathbf{y}_i = \mathbf{y}_i + \Delta \mathbf{y}_i$  a  $\mathbf{q} = \mathbf{q} + \Delta \mathbf{q}$ . Tento postup opakujeme do té doby, dokud nám metoda nezkonverguje.

### 1.3 Kalibrovatelnost

Kalibračním procesem získáme pro daná data jedinečné řešení. Řešení je funkční pro rozsáhlou oblast počátečního odhadu parametrů iteračního řešení Newtonové metody. Při kalibraci reálných strojů bylo zjištěno, že některé parametry určené z různých kalibračních

měření vykazují výrazné rozdíly. Důvodem je vzájemné působení zhoršené podmíněnosti lineární soustavy vypočítané iterací Newtonovou metodou, chyby způsobené zjednodušením modelu a chyby měření. [4] Pro zavedení pojmu kalibrovatelnost musíme více porozumět závislosti mezi prostorem parametrů a prostorem kalibračních výsledků.

Singulární (SVD) rozklad matice soustavy [5], která se nachází v Newtonově metodě, je hlavní krok při mapování prostoru parametrů následující

$$\mathbf{J}_i^{-T} \mathbf{J}_i = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T. \quad (1.3.1)$$

Matice  $\mathbf{U}_i$  a  $\mathbf{V}_i$  jsou matice ortogonální (platí  $\mathbf{U}_i^{-1} = \mathbf{U}_i^T$  a  $\mathbf{V}_i^{-1} = \mathbf{V}_i^T$ ). Matice  $\mathbf{S}_i$  je maticí diagonální, kde na diagonále jsou sestupně poskládaná singulární čísla. Rovnici (1.1.5) můžeme díky singulární rozkladu přepsat do tvaru

$$\mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T \delta \mathbf{d}_i = \mathbf{J}_i^T \delta \mathbf{r}_i. \quad (1.3.2)$$

Zavedeme si vektor pomocných proměnných  $\mathbf{y}_i$

$$\mathbf{y}_i = \mathbf{U}_i \mathbf{V}_i^T \delta \mathbf{d}_i, \quad (1.3.3)$$

jehož prvky vypočítáme z rovnice

$$\mathbf{S}_i \mathbf{y}_i = \mathbf{U}_i^T \mathbf{J}_i^T \delta \mathbf{r}_i. \quad (1.3.4)$$

Pokud je hodnota matice zmenšena o  $r$  (matice singulárních čísel má posledních  $r$  hodnot nulových), pak posledních  $r$  prvků vektoru  $\mathbf{y}_i$  označujeme jako volné parametry řešení. Toto řešení nám může nahradit řešení jedinečné. Oblast odpovídající této části  $\mathbf{y}_i$  označujeme jako oblast necitlivosti. Její mapování provedeme následujícím způsobem:

1. Uvažujeme dostatečný počet iterací Newtonovy metody pro získání řešení.
2. Posledních  $r$  prvků vektoru  $\mathbf{y}_i$  považujeme za volné parametry, které nebudeme počítat přímo. Zbylé parametry vypočítáme standardně.
3. Sestavíme cílovou funkci pro hledání volných parametrů (např. pro průměrné hodnoty součtu absolutních vypočítaných chyb).
4. Cílovou funkci minimalizujeme pomocí genetických algoritmů.

V této kapitole si zavedeme pojem kalibrovatelnost. Kalibrovatelnost nám označuje závislost mezi citlivostí parametrů a podmíněností Jacobiho matice  $\mathbf{J}_d$ . Jako cílové kritérium návrhu mechanismu kalibrovatelnost definoval až Šika a spol. v [5]. Jako cílové kritérium pro návrh redundantních mechanismů byla použita v [1].

Řešení budeme aplikovat pro základní kalibraci popsána v kapitole 1.1, respektive pokročilou kalibraci popsanou v 1.2. Bylo zjištěno, že se parametry mechanismů při různé podmíněnosti kalibrace mohou výrazně odlišovat. To si ukážeme v následující kapitole na příkladu dvou-dimenzionálního robotického ramene a tří-dimenzionálního mechanismu. Z předchozích rovnic (1.3.1) – (1.3.4) vidíme, že je kvalita kalibrace určena kvalitou Jacobiho matice  $\mathbf{J}_d$ . Vypovídající hodnotou Jacobiho matice je její podmíněnost. Parametr kalibrovatelnost proto zavedeme ve tvaru

$$C_{\text{základní}} = \text{cond}(\mathbf{J}_{di}^T \mathbf{J}_{di}) \quad (1.3.5)$$

Tento tvar rovnice budeme používat pro kalibraci základní. Jacobiho matice  $\mathbf{J}_{di}$  získáme pomocí parciální derivací vazbových rovnic podle parametrů.

Pro výpočet kalibrovatelnosti pokročilé kalibrace použijeme matici  $\begin{pmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_N \\ \mathbf{I} \end{pmatrix}$ , kterou můžeme vidět v rovnici (1.2.23). Postup, jak takovou matici získáme, je v předešlé kapitole popsán. Matice  $\mathbf{D}$  je popsána v rovnici (1.2.21) a získáme jí pro  $N$  poloh mechanismu.  $\mathbf{D}$  zapíšeme do matice pod sebe a na konec vložíme jednotkovou matici  $\mathbf{I}$ , která má počet řádků i sloupců stejný, jako je počet parametrů. Tvar, v jakém získáme kalibrovatelnost pro metodu pokročilou, můžeme vidět v následující rovnici.

$$C_{\text{pokročilá}} = \text{cond}\left(\begin{pmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_N \\ \mathbf{I} \end{pmatrix}^T \begin{pmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_N \\ \mathbf{I} \end{pmatrix}\right) \quad (1.3.6)$$

Kalibrovatelnost je bezrozměrné číslo, jehož hodnota popisuje citlivost přesnosti kalibračních rozměrů a přesnost koncového efektoru s ohledem na přesnost měření senzoru. [6] Popisuje tedy, jak kvalitně je daný problém určený. Čím je kalibrovatelnost lepší, tím bychom měli získávat lepší kinematické rozměry kalibrací daného mechanismu a následně získat přesnější řešení výsledného bodu. Podmíněnost je definována jako podíl největšího a nejmenšího singulárního čísla. [7] V našem případě se tedy snažíme získat hodnotu kalibrovatelnosti co možná nejnižší. Čím nižší hodnota, tím méně by měly ovlivňovat chyby parametrů mechanismu (např. chyby senzorů, vůle, změna teploty a tuhosti) hledané rozměry mechanismu a ve výsledku i polohu end-efektoru. Špatně podmíněná matice způsobí větší nepřesnosti v řešení i při malých chybách na vstupu. Jacobiho matice je závislá na rozměrech mechanismu, počtu a volbě měřených souřadnic. Závisí také na volbě použitých poloh a mění se s krokem iterace. Pokud chceme

porovnávat výsledky, musíme zavést neměnnou síť bodů pracovního prostoru, ve kterém můžeme hodnoty porovnávat.

Kalibrovatelnost může rozšířit stávající kritéria optimalizace mechanismů. Můžeme jej použít v případě, že chceme získat lépe kalibrovatelný stroj, který by teoreticky měl být „jednodušší“ na ovládání, a dosáhnout tak lepších hodnot koncového efektoru.

## 1.4 Optimalizace

V této části diplomové práce si ukážeme metody, jakými budeme optimalizovat zadaný mechanismu. Využijeme k tomu celkem dvě metody. Nejprve budeme optimalizovat pomocí funkce *fminsearch*, která je již plně definovaná v optimalizačním toolboxu v Matlabu. Následně použijeme metodu genetických algoritmů, která je definovaná v Matlabu také, a to v globální optimalizačním toolboxu. V této kapitole si popíšeme princip obou metod a důvod, proč použijeme obě.

### 1.4.1 Fminsearch

První si popíšeme metodu *fminsearch*. Tato lokální metoda je nelineární řešič, který hledá minimum funkce specifikovanou jako  $\min_x f(x)$ , kde  $x$  je vektor nebo matice a  $f$  je funkce, která vrací skalár. Předpis je

$$x = \text{fminsearch}(\text{fun}, x0, \text{options}), \quad (1.4.1)$$

kde *fun* je námi definovaný problém, *x0* jsou počáteční podmínky a *options* jsou parametry metody, kdy má řešič přestat pracovat (např. minimální změna cílové funkce, maximální počet iterací). [8]

*Fminsearch* používá Nelder-Mead algoritmus určený pro simplexovou metodu poprvé publikovanou již v roce 1965. Tuto metodu můžeme použít pro jednu nebo více proměnných k řešení neomezeného problému bez použití derivací. [9] V našem případě sice budeme mít problém omezený, nicméně omezení vepíšeme již do základní funkce *fun*. Algoritmus používá simplex o velikosti  $n+1$  bodů pro  $n$ -rozměrný vektor  $x$ . Na začátku algoritmus vypočítá simplex v okolí počátečního odhadu  $x_0$ . [8]

Postup metody [8] je následující:

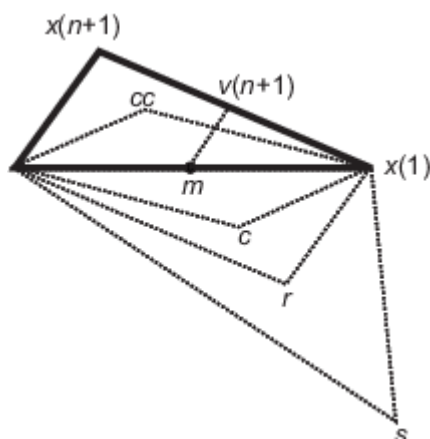
1. Označíme si seznam bodů stávajícího simplexu jako  $x(i)$ ,  $i=1, \dots, n+1$



2. Seřadíme body simplexu od nejnižších funkčních hodnot  $f(x(1))$  po nejvyšší  $f(x(n+1))$ . Algoritmus v každém kroku iterace vypustí aktuálně nejhorší bod  $x(n+1)$  a zapíše místo něj do simplexu jiný bod. Výpočet nového bodu je popsán v 3-6. Pokud nastane případ popsán v bodě 7, změní se všechny body  $n$ , jejichž hodnota je větší než  $f(x(1))$ .
3. Vygenerujeme bod  $r = 2m - x(n+1)$ , kde  $m = \sum x(i)/n$ ,  $i = 1, \dots, n$  a vypočítáme funkci  $f(r)$ .
4. Pokud  $f(x(1)) \leq f(r) < f(x(n))$ , zapíšeme si  $r$  a ukončíme iteraci. Pokud se tak stane, vypíše se nám *reflect*.
5. Pokud  $f(r) < f(x(1))$ , vypočítáme expanzní bod  $s$  a vypočítáme  $f(s)$ 
  - a) pokud  $f(s) < f(x(r))$ , zapíšeme  $s$ , ukončíme iteraci a vypíše se *expand*
  - b) v jiném případě zapíšeme  $r$ , ukončíme iteraci a vypíše se *reflect*.
6. Pokud  $f(r) \geq f(x(n))$ , provedeme kontrakci mezi  $m$  a  $x(n+1)$  nebo  $r$ , podle toho, který bod má menší hodnotu funkce.
  - a) Pokud  $f(r) < f(x(n+1))$ , pak platí  $r > x(n+1)$ , vypočítáme  $c = m + (r - m)/2$  a vypočítáme  $f(c)$ . Pokud bude platit  $f(c) < f(r)$ , zapíšeme si  $c$ , ukončíme iteraci a vypíše se *contract outside*.
  - b) Pokud  $f(r) \geq f(x(n+1))$ , vypočítáme  $cc = m + (x(n+1) - m)/2$  a vypočítáme  $f(cc)$ . Pokud  $f(cc) < f(x(n+1))$ , přijmeme  $cc$ , ukončíme iteraci a vypíše se *contract inside*.
  - c) V jiném případě pokračujeme do bodu 7.
7. Vypočítáme  $n$  bodů Pokud  $v(i) = x(1) + x(x(i) - x(1))/2$  a vypočítáme  $f(v(i))$ ,  $i = 2, \dots, n+1$ . Simplex v další iteraci má hodnoty v bodě  $x(i), v(2), \dots, v(n+1)$ . Program vypíše *shrink*.

Na obrázku můžeme vidět možné konfigurace uzlů vypočítané pomocí *fminsearch*, jehož postup je popsán výše, a také možný simplex. Původní simplex je označen tučně. Metoda iteruje do té doby, dokud nejsou splněny námi zadaná kritéria (*options*).

[8]



Obr. 2: Změna bodu simplexu v *fminsearch*

### 1.4.2 Genetické algoritmy (GA)

Genetické algoritmy jsou druhá metoda, kterou použijeme v naší práci. Je to metoda, která může řešit jak omezené, tak neomezené problémy. Řešení je na základě biologických jevů, které se objevují běžně přírodě, jako je evoluce a přírodní výběr. Genetické algoritmy jsou globální metoda. Princip genetických algoritmů spočívá v modifikaci náhodně vybraných jedinců. Každý jedinec představuje jedno řešení daného problému. V každém kroku algoritmus náhodně vybere jedince z aktuální populace (soubor všech jedinců=populace). Použije je jako „rodiče“ a vytvoří z nich další generaci „lepší“ jedinců. Nová generace se tvoří 3 metodami. Křížením, mutací a reprodukci. [10]

Optimalizovat problém genetickými algoritmy můžeme v tom případě, kdy definovaný problém není dobře řešitelné běžnými optimalizačními algoritmy. To může být i v případě, kdy funkce předmětu optimalizace není spojitá, nediferencovatelná nebo je velice nelineární. Optimalizace ukončíme v případě, kdy dosáhneme námi požadované kvality řešení, které udává takzvané fitness funkce, nebo počtu generací, které jsme ochotni vypočítat. Fitness funkce nám vyjadřuje kvalitu řešení každého jedince. Lepší jedince určíme srovnáním jejich fitness funkcí. [10]

Postup [11] [12] genetických algoritmů je následující:

1. Iniclace – náhodný výběr populace  $P_0$
2. Zhodnocení fitness funkce všech jedinců
3. Výběr a podržení si nejlepších jedinců
4. Výběr rodičů z nejlepších jedinců

5. Generování nových jedinců (dětí) pomocí křížení, mutace, reprodukce
6. Výpočet zdatnosti (fitness funkce) nových jedinců.
7. Při splnění požadované hodnoty fitness funkce nebo maximálního počtu jedinců konec optimalizace. Pokud ne, zpět do bodu 3.
8. Nejlepší řešení = jedinec s nejlepší fitness funkcí

Možností, jak vygenerujeme nové jedince (generaci) jsou celkem tři. Mutace, křížení a reprodukce. Křížením algoritmus náhodně zkombinuje vlastnosti „rodičů“ a vytvoří „dítě“. Při mutaci náhodně změním některou vlastnost (gen) jedince. Při reprodukci pouze zkopírujeme rodiče.

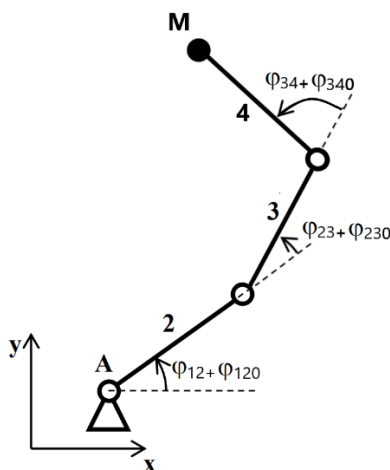
Od klasických algoritmů (např. *fminsearch*) se genetické algoritmy liší ve dvou hlavních skutečnostech. Genetické algoritmy vytváří generace, z nichž jedinec, který splňuje nejlépe fitness funkci, je nejlepší řešení. Následně vygenerují další populaci náhodně. Klasické metody generují jediný bod pro každou iteraci a pro další zlepšení vychází právě z tohoto vypočítaného bodu. Genetické algoritmy proto mohou nalézt lepší řešení díky náhodné změně některého z parametrů optimalizace, kdežto klasické algoritmy pracují striktně deterministicky a mohou nalézt pouze lokální řešení a ne globální. Závisí to především na počátečních podmínkách. Není ale překvapující, že genetické algoritmy mají i negativa. Mezi ty patří především potřeba vypočítat větší počet cílových funkcí pro úplné zkonvergování řešení a tím pádem delší výpočetní čas. Dále také genetické algoritmy mohou ale zároveň nemusí konvergovat k lokálnímu nebo globálnímu řešení. Klasické algoritmy většinou konvergují k lokálnímu řešení lépe. Z tohoto důvodu je vhodné kombinovat obě řešení, kdy si nejprve zjistíme nadějnou oblast globálního řešení pomocí genetických algoritmů a následně využijeme klasické algoritmy s počáteční podmínkou získanou z globálních algoritmů. [12]

## 2. Rovinný mechanismus

V této kapitole si představíme rovinný mechanismus robotického ramene, se kterým budeme pracovat. Nejprve si model popíšeme. Pak si ukážeme generování dat potřebných pro kalibraci, které budou reprezentovat hodnoty získané ze skutečného měření reálného mechanismu. Tato data budou vstupními hodnotami jak pro základní, tak i pokročilou kalibraci. Popíšeme vytvoření Jacobiho matice a kovariančních matic. Následně vytvoříme dva kalibrační modely. První bude popisovat kalibraci robotického ramene bez uvažování kompenzace chyb senzorů. V druhém již budeme kalibrovat s chybami senzorů. Výsledky obou metod graficky zobrazíme a následně je také porovnáme. Práci s robotickým rovinným ramenem zakončíme optimalizací. K tomu využijeme popsané metody `fminsearch` a genetické algoritmy

### 2.1 Popis modelu

Ověření funkčnosti kalibrace a optimalizace kalibrovatelnosti si vyzkoušíme na jednoduchém 2D modelu. Jak již bylo zmíněno, zadaným mechanismem je rameno robota. Robot se skládá z celkem tří ramen, které jsou spojené rotačními vazbami. První rameno je rotační vazbou uchyceno k základnímu rámu mechanismu. To představuje bod A na obrázku Na konci posledního ramene si můžeme představit end efektor (bod M), například nástroj k uchycení nějakého předmětu. Kinematické schéma robotického ramene má tři stupně volnosti a můžeme ho vidět na následujícím obrázku.



Obr. 3: Kinematické schéma robotického ramene

Model popisujeme v kartézské soustavě souřadnic  $x$  a  $y$ . Počáteční bod tedy označíme jako A a jeho vzdálenost od počátku je  $[x_A, y_A]$ . Rozměry ramen označíme popořadě  $l_2$ ,  $l_3$  a  $l_4$ . Poloha koncového bodu neboli end efektoru, který je označený jako

M, je popsána obdobně jako u bodu A, tedy  $[x_M, y_M]$ . Relativní úhly natočení ramen popíšeme postupně jako  $\varphi_{12}$ ,  $\varphi_{23}$  a  $\varphi_{34}$ . Každý úhel má offset, který označíme  $\varphi_{120}$ , respektive  $\varphi_{230}$  a  $\varphi_{340}$ . Tyto offsety představují malé natočení měřicího senzoru oproti ideálnímu stavu. Vzniknou například při montáži senzorů.

## 2.2 Generování dat

Pro generování dat využijeme inverzní metodu. Musíme si zadat pracovní prostor, ve kterém chceme, aby rameno pracovalo. Pro jednoduchost budeme uvažovat čtverec o velikosti 1x1 m. Pracovní prostor rozdělíme NxM body s konstantní vzdáleností mezi sebou a vznikne nám grid. Vzdálenost každého bodu gridu od počátku nám reprezentují souřadnice, ve kterých chceme znázornit koncový bod M mechanismu. Celkem budeme mít tedy NxM souřadnic bodu M. Dále si definujeme K-úhlů natočení čtvrtého ramena vůči základnímu rámu mechanismu, který označíme  $\varphi_{14}$ .

Vznikne nám matice NxMxK, která popisuje různé natočení posledního ramene a polohu bodu M. Pro všechny tyto polohy zjistíme polohy celého mechanismu. V našem případě budeme mít tedy celkem 125 poloh mechanismu (5x5x5), ve kterém potřebujeme vypočítat hodnoty úhlů  $\varphi_{12}$ ,  $\varphi_{23}$  a  $\varphi_{34}$ . K tomu nám pomůžou následující rovnice, které musí být splněny pro zaručení správného fungování mechanismu.

$$x_C = x_M - l_4 \cos(\varphi_{14}) \quad (2.2.1)$$

$$y_C = y_M - l_4 \sin(\varphi_{14}) \quad (2.2.2)$$

$$(l_3 \sin(\varphi_{23}))^2 + ((l_2 + l_3 \cos(\varphi_{14}))^2 = (x_C - x_A)^2 + (y_C - y_A)^2 \quad (2.2.3)$$

$$x_C = x_A + l_2 \cos(\varphi_{12}) + l_3 \cos(\varphi_{12} - \varphi_{23}) \quad (2.2.4)$$

$$y_C = y_A + l_2 \sin(\varphi_{12}) + l_3 \sin(\varphi_{12} - \varphi_{23}) \quad (2.2.5)$$

$$\varphi_{34} = \varphi_{14} - \varphi_{12} - \varphi_{23} \quad (2.2.6)$$

Rovnice (2.2.1) až (2.2.6) si upravíme a vypočítáme velikosti úhlů  $\varphi_{12}$ ,  $\varphi_{23}$  a  $\varphi_{34}$  pro všechny polohy mechanismu.

Hodnoty parametrů jsme museli zadat tak, aby byly vazbové rovnice (2.2.1) až (2.2.5) splněny. Trefit se do hodnot, aby mechanismus fungoval tak, jak má, bylo záludné, ale po několika pokusech jsme získali hodnoty, které můžeme vidět v následující tabulce. Bude

zajímavé sledovat, jak se tyto hodnoty budou v průběhu kalibrací měnit a kam se mechanismus posune v prostoru.

<i>Parametr</i>	<i>Hodnota</i>
$x_A$	-1.0
$y_A$	-1.0
$l_2 = l_3 = l_4$	1.0
$\varphi_{120} = \varphi_{230} = \varphi_{340}$	0.0

Tabulka 1: Počáteční parametry

Data, která získáme tímto postupem, následně zachybuje pomocí v Matlabu již definovaného příkazu „randn“ [13] a vytvoříme z absolutně přesných dat hodnoty, které bychom získali při měření reálného stroje. Příkaz „randn“ vygeneruje náhodná reálná čísla podle normálního rozložení Gaussova statistického modelu, který má rovnici

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \quad (2.2.7)$$

kde střední hodnota reálné proměnné  $x$  se rovná nule a odchylka  $x$  je jedna. Velikost 1000 vygenerovaných hodnot se nejčastěji pohybuje od -3 do 3. Hodnoty následně musíme převést na velikost takového řádu, aby nám korektně reprezentovala zachybování daných parametrů. Parametry délkové zachybuje chybou v řádu  $10^{-3}$  a úhlové parametry v řádu  $10^{-4}$ . Takto zachybovaná data budou simulovat reálné hodnoty s nepřesnostmi měření podle normálního rozložení Gaussova statistického modelu a budeme je používat pro kalibrace. Tato data získáme ze skriptu „Generovani\_dat.m“, který je přiložen v příloze.

## 2.3 Kalibrace

Základem kinematické kalibrace je závislost mezi parametry mechanismu  $d = [x_A; y_A; l_2; l_3; l_4; \varphi_{120}; \varphi_{230}; \varphi_{340}]$ , souřadnicemi pohonů  $\varphi_{12} = \varphi_{12}(t)$ ,  $\varphi_{23} = \varphi_{23}(t)$ ,  $\varphi_{34} = \varphi_{34}(t)$  a polohou koncového bodu mechanismu  $M$  označenou  $x_M = x_M(t)$  a  $y_M = y_M(t)$ . Získání těchto hodnot jsme si ukázali v předchozí podkapitole. V této podkapitole si definujeme potřebné vztahy pro výpočet kalibrací. Definujeme Jacobiho matice parametrů a čidel a kovarianční matice.

Pro zadaný mechanismus známe dvě vazbové rovnice ve tvaru

$$\begin{aligned} x_A + l_2 \cos(\varphi_{12} + \varphi_{120}) + l_3 \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) \\ + l_4 \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} \\ + \varphi_{340}) - x_M = 0 \end{aligned} \quad (2.3.1)$$

$$y_A + l_2 \sin(\varphi_{12} + \varphi_{120}) + l_3 \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) \quad (2.3.2) \\ + l_4 \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} \\ + \varphi_{340}) - y_M = 0$$

Tyto dvě rovnice parciálně zderivujeme podle parametrů  $x_A$ ,  $y_A$ ,  $l_2$ ,  $l_3$ ,  $l_4$ ,  $\varphi_{120}$ ,  $\varphi_{230}$ ,  $\varphi_{340}$  a výsledky zapíšeme do matice, která je ukázaná v následující tabulce. Pro přehlednost jsme zápis uvedli v transponovaném tvaru. Sloupec  $f1$  představuje parciální derivace rovnice (2.3.1) a  $f2$  představuje parciální derivace rovnice (2.3.2). Tuto matici označíme  $J_q$  a nazýváme jí Jacobiho maticí parametrů. Jacobiho matici můžeme vypočítat v jednoduchém případě analyticky nebo můžeme použít v Matlabu příkaz „Jacobian“.

	$f1$	$f2$
$\frac{\partial}{\partial x_A}$	1	0
$\frac{\partial}{\partial y_A}$	0	1
$\frac{\partial}{\partial l_2}$	$\cos(\varphi_{12} + \varphi_{120})$	$\sin(\varphi_{12} + \varphi_{120})$
$\frac{\partial}{\partial l_3}$	$\cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230})$	$\sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230})$
$\frac{\partial}{\partial l_4}$	$\cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$	$\sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$
$\frac{\partial}{\partial \varphi_{120}}$	$-l_2 \cdot \sin(\varphi_{12} + \varphi_{120}) - l_3 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) + l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$	$l_2 \cdot \cos(\varphi_{12} + \varphi_{120}) + l_3 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) + l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$
$\frac{\partial}{\partial \varphi_{230}}$	$-l_3 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) - l_4 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$	$l_3 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) + l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$
$\frac{\partial}{\partial \varphi_{340}}$	$-l_4 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$	$l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$

Tabulka 2: Jacobiho matice  $J_q$

Pro výpočet základní kalibrace, kterou jsme popsali v kapitole 1.2, máme všechny potřebné vstupy, a proto už zbývá tento postup použít. Jacobiho matice dosadíme do rovnice (1.1.5) a vypočítáme opravu parametrů  $\mathbf{d}$ . Následně opravu parametrů přičteme

k parametrům, jak popisuje rovnice (1.1.6). Tento postup opakujeme, dokud nedosáhneme dostatečné přesnosti.

Všechny následující informace, které si v této podkapitole ukážeme, jsou potřeba k výpočtu kalibrace s uvažováním kompenzace nepřesností senzorů, která je popsána v kapitole 1.2. Již jsme si ukázali Jacobiho matici  $J_q$  výše. Tu použijeme ji ve stejném tvaru. Dále definujeme Jacobiho matice čidel  $J_y$ . Matice  $J_y$  vznikne pomocí parciálního zderivování vazbových rovnic podle měřených veličin, kterými jsou  $x_M$ ,  $y_M$ ,  $\varphi_{12}$ ,  $\varphi_{23}$  a  $\varphi_{34}$ . Tvar této matice můžeme vidět v následující tabulce. Použili jsme stejný princip popisu jako v Tabulce 1.

	$f1$	$f2$
$\frac{\partial}{\partial x_M}$	1	0
$\frac{\partial}{\partial y_M}$	0	1
$\frac{\partial}{\partial \varphi_{12}}$	$-l_2 \cdot \sin(\varphi_{12} + \varphi_{120}) - l_3 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) + l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$	$l_2 \cdot \cos(\varphi_{12} + \varphi_{120}) + l_3 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) + l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$
$\frac{\partial}{\partial \varphi_{23}}$	$-l_3 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) - l_4 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$	$l_3 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230}) + l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$
$\frac{\partial}{\partial \varphi_{34}}$	$-l_4 \cdot \sin(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$	$l_4 \cdot \cos(\varphi_{12} + \varphi_{120} + \varphi_{23} + \varphi_{230} + \varphi_{34} + \varphi_{340})$

Tabulka 3: Jacobiho matice  $J_y$

Dále si zavedeme kovarianční matici jak parametrů, tak měřených veličin. Kovarianční maticí parametrů  $\Sigma_p$  bude diagonální matice, kdy prvky diagonály budou mít velikost rozdílu skutečné hodnoty parametru a navržené hodnoty parametru. V našem případě to bude pro délkové parametry ( $x_A$ ,  $y_A$ ,  $l_2$ ,  $l_3$ ,  $l_4$ ) řádově  $10^{-3}$  a pro úhlové parametry ( $\varphi_{120}$ ,  $\varphi_{230}$ ,  $\varphi_{340}$ )  $10^{-4}$ . Jak již bylo zmíněno dříve, tento rozdíl je vytvořený pomocí příkazu *randn*.

Potřebujeme si ještě definovat kovarianční matici pro měřené parametry  $\Sigma_x$ . Budeme předpokládat vzájemnou nezávislost čidel, a proto kovarianční matice měřených



parametrů bude mít také tvar diagonální matice, kde prvky na diagonále budou směrodatné odchylky daného čidla na druhou. Kovarianční matici musíme pokaždé upravit tak, aby odpovídala řádu přesnosti čidel a jejich směrodatným odchylkám. Ve výpočtu použijeme příkaz *cov*, který prvky kovarianční matice vytvoří. [14]

## 2.4 Výsledky kalibrace

V této části si ukážeme výsledky kalibrací pro různě definované zachybování chyb senzorů čidel. Zjistíme, zda námi vytvořené skripty fungují správně a dle předpokladů a ukážeme si, kterou metodou získáme lepší řešení. Vyzkoušíme několik různých případů pro obě metody kalibrace a porovnáme jejich výsledky. Budeme porovnávat především to, jak se zkalibrované parametry přiblížily skutečným a jaká vznikla rezidua vazbových rovnic (2.3.1) - (2.3.2) při odečtení přesných a zkalibrovaných hodnot.

Prvním případem vyzkoušíme funkčnost skriptů a to tak, že budeme uvažovat dokonale přesná čidla. Nepřesnosti jejich měření budou v řádu  $10^{-16}$ , což je v Matlabu numerická nula. V tomto případě očekáváme, že výsledné parametry po skončení obou kalibrací budou totožné nebo se budou limitně blížit nule.

Ve druhém případě už definujeme nepřesnost senzorů tak, aby měly na výsledky vliv. Rozhodli jsme se nastavit nepřesnost pro všechny senzory stejného řádu, a to  $10^{-5}$ . Takovou velikost chyb jsme zvolili z toho důvodu, že reprezentuje hodnoty, kterých je možné dosáhnout v reálném případě.

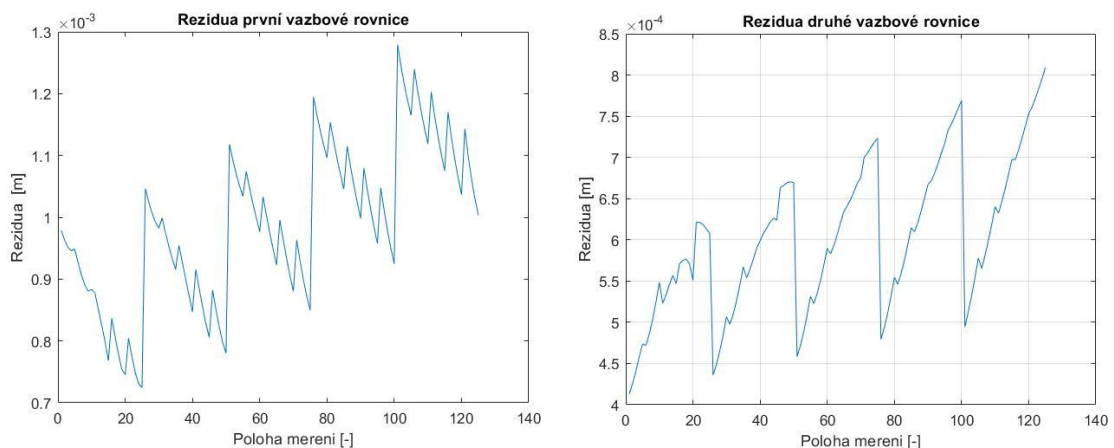
Dále zvolíme nepřesnosti senzoru stejné jako v případě druhém až na jeden senzor. Tím bude senzor pro měření úhlu  $\varphi_{34}$ , jehož řád přesnosti jsme změnili z  $10^{-5}$  na  $10^{-3}$ , respektive  $10^{-4}$ . Toto čidlo tedy měří méně přesněji než ostatní čidla. Poslední příklad bude pro nepřesnost čidla  $\varphi_{34}$   $10^{-3}$  a ostatní zachybujeme  $10^{-16}$ .

Předpokládáme, že v dalších případech již vypořádáme rozdíly mezi skutečnými a zkalibrovanými hodnotami. Zároveň uvidíme, která metoda kalibrace je výhodnější. Se zhoršující se chybovostí čidla  $\varphi_{34}$  očekáváme, že se bude zvětšovat rozdíl mezi přesností obou metod kalibrace. V posledním případě by pak měl být vidět největší vliv tohoto čidla  $\varphi_{34}$  na výsledné řešení.

Než přejdeme k výsledkům, musíme si ještě připomenout, že u dat, která chceme porovnávat, musíme zachybovat senzory čidel a parametrů naprosto stejnými hodnotami. Pokud bychom to neudělali, nemělo by smysl data porovnávat, protože bychom v podstatě porovnávali jiné řešení daného mechanismu. Proto jsme parametry mechanismu zachybovali úplně na začátku příslušnými chybami a následně je nakopírovali do všech výpočtů. Chybovost sensorů musíme určit pro každý případ zvlášť. Skripty pro kalibrace nalezneme v příloze pod názvem Kalibrace\_zakladni“ a „Kalibrace\_pokrocila“.

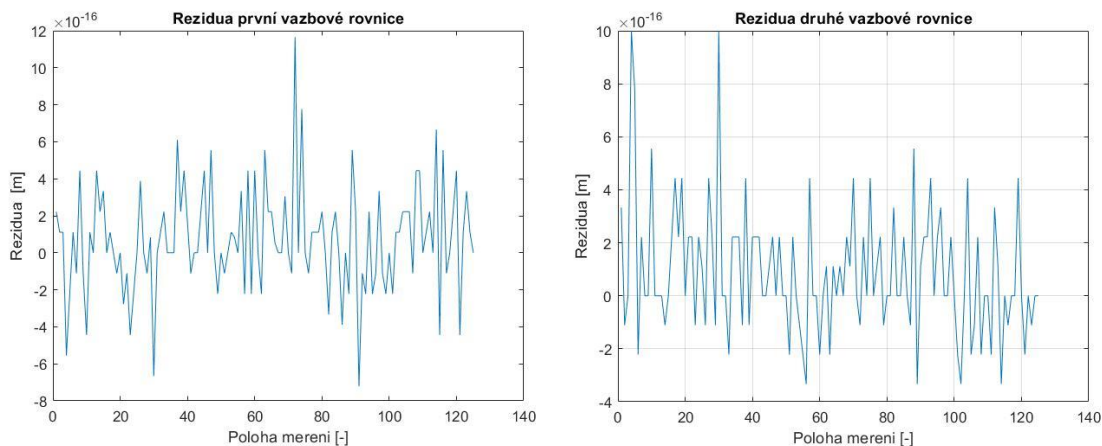
### 2.4.1 Chyby všech sensorů $10^{-16}$

Začneme výsledky pro chyby sensorů rovné nule. Ověříme si tím funkčnost našich skriptů. Na následujících obrázcích můžeme vidět rezidua vazbových rovnic před začátkem kalibrací. Je zřejmé, že jsou rezidua v řádu  $10^{-3}$ - $10^{-4}$ . To jsme předpokládali, jelikož to odpovídá řádu našeho zachybování parametrů a toto je znázornění ještě před spuštěním kalibrace. Pro všechny další případy, které si ukážeme, budou tyto grafy téměř totožné, a proto je už zmiňovat nebudeme.

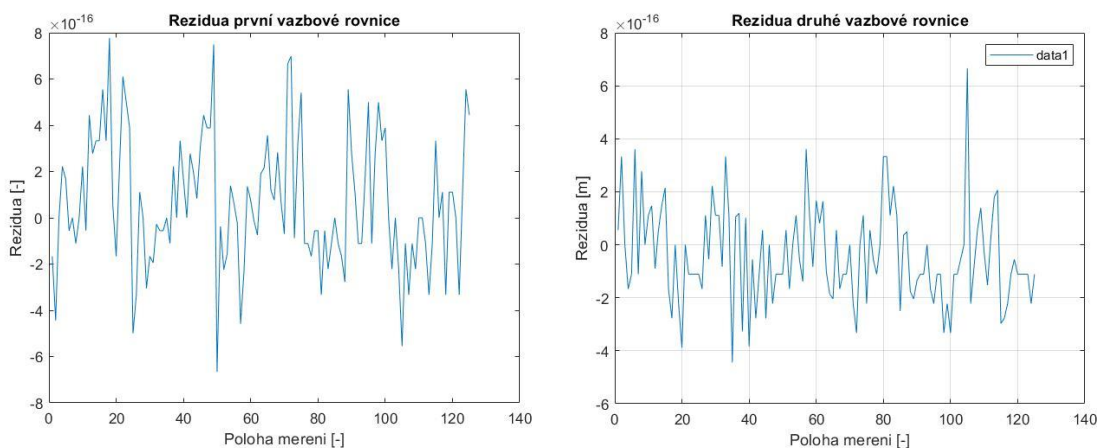


Obr. 4: Rezidua vazbových rovnic před kalibrací

Přichází na řadu výsledky po kalibrování. Jako první můžeme vidět graf reziduí vazbových rovnic po základní kalibraci a hned pod nimi po kalibraci s uvažováním kompenzace nepřesností sensorů. Můžeme vidět, že obě metody dospěli k námi předpokládanému řešení, a vzniklá rezidua mají velikost numerické nuly, tedy  $10^{-16}$ . U obou metod jsme k těmto výsledkům potřebovali celkem čtyři iterace.



Obr. 5: Rezidua vazbových rovnic základní kalibrace s uvažováním chyb senzorů  $10^{-16}$



Obr. 6: Rezidua vazbových rovnic pokročilé kalibrace s uvažováním chyb senzorů  $10^{-16}$

V následující tabulce můžeme vidět rozdíl parametrů skutečných od parametrů zjištěných kalibracemi. Výsledky v této části dopadly znovu tak, jak jsme předpokládali. U obou kalibrací jsou nula. Parametry obou metod nám tedy zkonvergovaly ke skutečným parametrům.

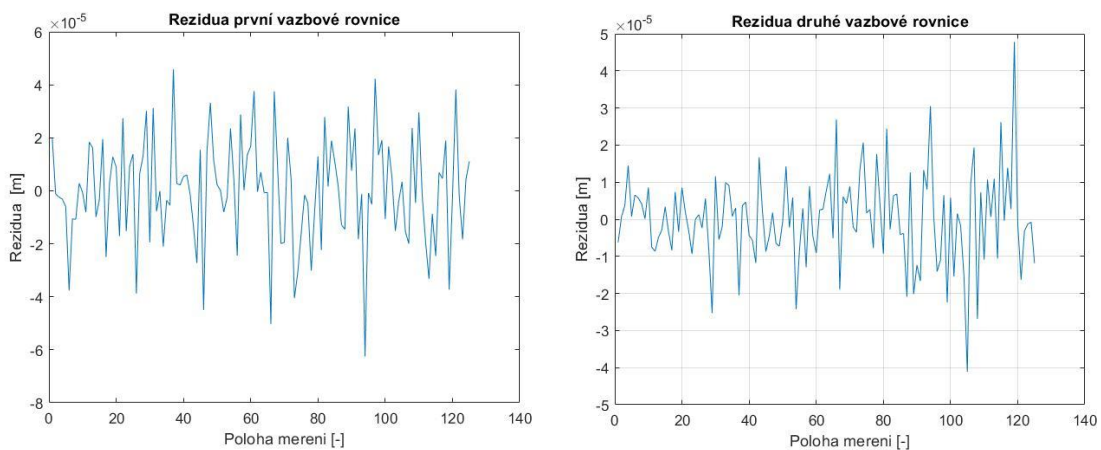
Parametr	Základní kalibrace	Pokročilá kalibrace
$\Delta x_A$ [m]	1.1102e-16	1.1102e-16
$\Delta y_A$ [m]	1.1105e-16	3.3307e-16
$\Delta \varphi_{120}$ [rad]	1.0399e-16	4.2325e-17
$\Delta l_2$ [m]	0	0
$\Delta \varphi_{230}$ [rad]	1.2952e-16	4.6756e-18
$\Delta l_3$ [m]	0	2.2204e-16
$\Delta \varphi_{340}$ [rad]	5.2601e-17	6.9762e-17
$\Delta l_4$ [m]	2.2204e-16	2.2204e-16

Tabulka 4: Porovnání přesných a zkalibrovaných parametrů při chybě senzorů  $10^{-16}$

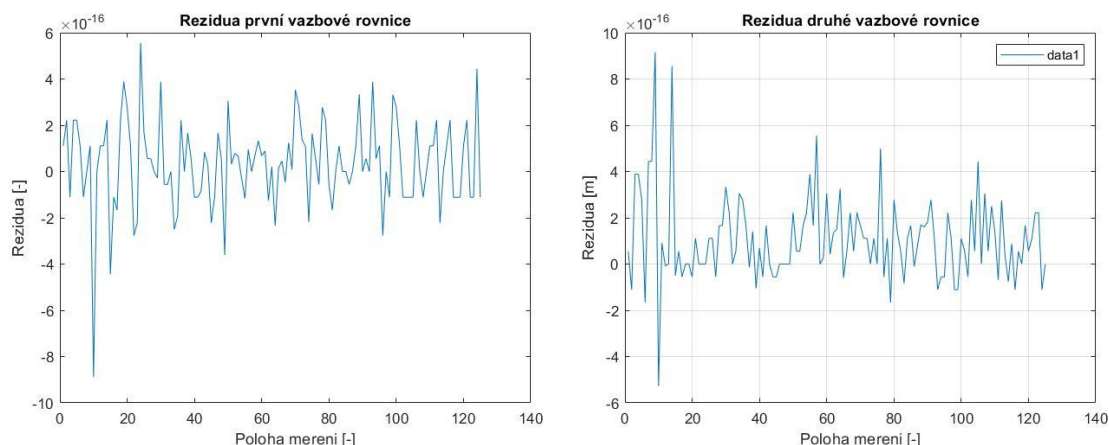
## 2.4.2 Chyby všech senzorů $10^{-5}$

Metody nyní ověříme se zachybovanými senzory v řádu nepřesnosti  $10^{-5}$  u všech senzorů. Níže můžeme vidět porovnání vazbových rovnic obou metod kalibrace od skutečných hodnot. Jak jsme očekávali, rezidua základní kalibrace se pohybují v řádu chyby senzorů. Těchto hodnot dosáhneme již při třetí iteraci. Při dalších iteracích se výsledky nemění a graf je neměnný. U kalibrace s uvažováním kompenzace nepřesnosti senzorů se rezidua posunou podobně jako v předchozím případě na hodnotu numerické nuly. Jak můžeme vidět, oproti základní kalibraci můžeme opravou chyb senzorů splnit výrazně lépe vazbové podmínky a můžeme se pravděpodobně i lépe přiblížit skutečné poloze end efektoru.

Po změně kovarianční matice druhá metoda zabrala přibližně 300 iterací, to je velice vysoký nárůst. V takto jednoduchém výpočtu to není příliš podstatné, nicméně u optimalizování pomocí genetických algoritmů očekáváme dlouhý výpočetní čas. Proto by se možná v případě dalšího používání těchto mnou vytvořených skriptů vyplatilo tyto skripty optimalizovat pro lepší využití výpočetní síly. V případě složitějších mechanismů by se měl tento faktor vzít také v potaz. Nárůst výpočtového času je způsoben především změnou směrodatných odchylek v kovarianční matici. Je proto tyto matice důležité dobře a kvalitně definovat, protože rychlost výpočtu výrazně ovlivňují.



Obr. 7: Rezidua vazbových rovnic základní kalibrace s uvažováním chyb senzorů  $10^{-5}$



Obr. 8: Rezidua vazbových rovnic pokročilé kalibrace s uvažováním chyb senzorů  $10^{-5}$

Následuje porovnání parametrů mechanismu pro obě metody, které uvádíme v tabulce níže. Jak můžeme z tabulky vyzorovat, v některých parametrech je pokročilá kalibrace přesnější, v některých podobná a v některých mírně horší. Tento výsledek je v souladu se zjištěním [15], kdy také u rovinného mechanismu byly získány podobné výsledky.

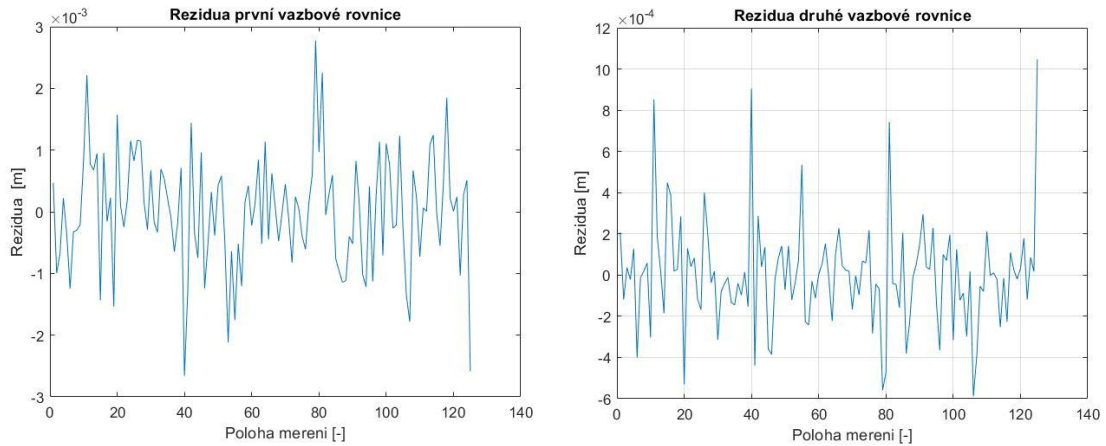
Parametr	Základní kalibrace	Pokročilá kalibrace
$\Delta x_A$ [ $\mu\text{m}$ ]	7.6815	2.6409
$\Delta y_A$ [ $\mu\text{m}$ ]	4.8917	4.7754
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	1.4395	1.7595
$\Delta l_2$ [ $\mu\text{m}$ ]	1.2767	0.9330
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	0.0979	1.4660
$\Delta l_3$ [ $\mu\text{m}$ ]	5.6949	0.6763
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	3.4310	1.9039
$\Delta l_4$ [ $\mu\text{m}$ ]	3.1917	1.8692

Tabulka 5: Porovnání přesných a zkalibrovaných parametrů při chybě senzorů  $10^{-5}$

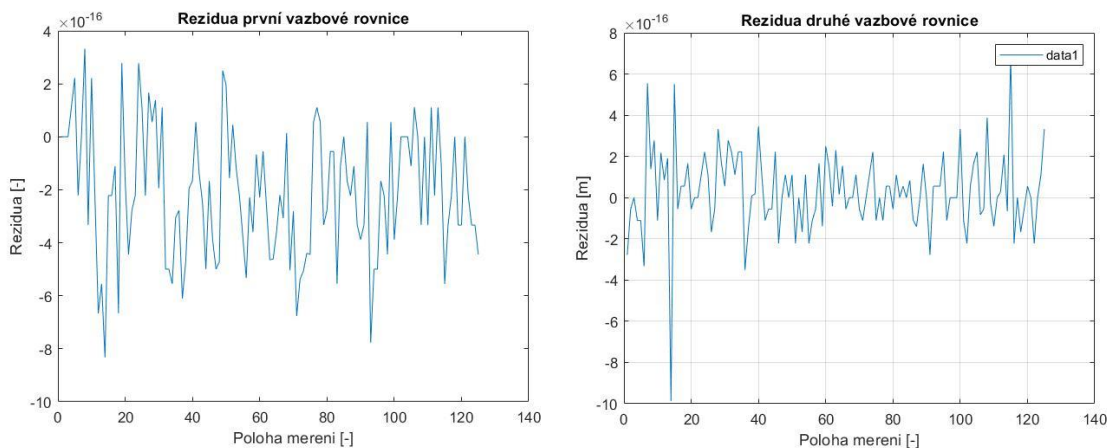
### 2.4.3 Zhoršení chyby senzoru $\varphi_{34}$ o jeden a dva řády, ostatní $10^{-5}$

V této části si ukážeme výsledky pro zhoršení čidla  $\varphi_{34}$ . Snížíme přesnost tohoto senzoru o jeden a následně o dva řády oproti ostatním čidlům, tedy na řád  $10^{-3}$ , respektive  $10^{-4}$ . Chyby ostatních čidel zachováme v řádu  $10^{-5}$ . Uvidíme, zdali tento příklad již potvrdí lepší vlastnosti pokročilé kalibrace. Obrázky jsou ukázány pro zachybování  $\varphi_{34}$  řádem  $10^{-4}$ . Níže můžeme znovu vidět rozdíl vazbových rovnic mezi skutečným mechanismem a mechanismem po kalibraci. U základní kalibrace se tyto rezidua pohybují v řádu  $10^{-3}$ - $10^{-4}$ .

V porovnání s předchozím případem se rezidua vazbových rovnic očekávaně zhoršila. U pokročilé kalibrace jsme předpokládali kompenzaci těchto nepřesností senzorů i při zhoršení některého čidla. To se vyplnilo a rezidua vazbových rovnic jsou rovna numerické nule i v tomto případě, jak můžeme vidět na obr. 10.



**Obr. 9: Rezidua vazbových rovnic základní kalibrace s uvažováním chyby senzoru  $\Phi_{34} 10^{-3}$**



**Obr. 10: Rezidua vazbových rovnic pokročilé kalibrace s uvažováním chyby senzoru  $\Phi_{34} 10^{-3}$ , ostatní  $10^{-5}$**

Teď už nám zbývá akorát porovnat změnu parametrů, abychom ověřili vlastnosti daných metod kalibrace. Porovnání obou metod znovu vychází lépe pro pokročilou metodu. Tentokrát je ve všech parametrech viditelné výrazné zlepšení jak pro zachybování  $\Phi_{34}$  řádem  $10^{-3}$ , tak i  $10^{-4}$ . Ve většině parametrů byla zaznamenána změna k lepšímu o dva, respektive o jeden řád. Ani jeden parametr se nezhoršil.

Znovu je velice důležité správné určení přesnosti čidel a správné definování kovariančních matic, protože to má významný vliv na výsledek celé metody.

Parametr	$\varphi_{34} 10^{-3}$		$\varphi_{34} 10^{-4}$	
	Základní kalibrace	Pokročilá kalibrace	Základní kalibrace	Pokročilá kalibrace
$\Delta x_A$ [ $\mu\text{m}$ ]	200.8194	5.0620	0.1023	0.0077
$\Delta y_A$ [ $\mu\text{m}$ ]	259.4526	68.0387	0.3362	0.0461
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	128.3754	0.8266	0.1945	0.0087
$\Delta l_2$ [ $\mu\text{m}$ ]	274.6754	3.3254	0.0334	0.0019
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	118.3909	0.7874	0.1449	0.0007
$\Delta l_3$ [ $\mu\text{m}$ ]	144.8728	2.2500	0.1971	0.0013
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	152.3961	94.8574	0.1836	0.0100
$\Delta l_4$ [ $\mu\text{m}$ ]	82.6053	70.7411	0.3306	0.0395

Tabulka 6: Porovnání přesných a zkalibrovaných parametrů při chybě senzoru  $\varphi_{34} 10^{-3}$  a  $10^{-4}$ , ostatní  $10^{-5}$

#### 2.4.4 Chyba senzoru $\varphi_{34} 10^{-3}$ , ostatní $10^{-16}$

Pro úplnost si stručně ukážeme, co se stane, pokud všechny senzory budou mít chybu měření v řádu  $10^{-16}$  a jeden senzor, v našem případě znovu  $\varphi_{34}$  bude mít chybu řádu  $10^{-3}$ . Tento případ je kombinací případů, které jsme si již ukázali, proto předpokládáme, že výsledné hodnoty budou také kombinací.

Rezidua vazbových rovnic základní kalibrace se v tomto případě pohybují v řádu  $10^{-3}$  a u pokročilé v řádu  $10^{-16}$ . To jsme již mohli vidět v předchozích případech, a proto si výsledné grafy nebudeme zobrazovat.

Parametr	Základní kalibrace	Pokročilá kalibrace
$\Delta x_A$ [ $\mu\text{m}$ ]	4.6705	0
$\Delta y_A$ [ $\mu\text{m}$ ]	991.5147	0
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	12.2517	0
$\Delta l_2$ [ $\mu\text{m}$ ]	113.7645	0
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	97.9584	0
$\Delta l_3$ [ $\mu\text{m}$ ]	133.7641	0
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	60.3422	57.0403
$\Delta l_4$ [ $\mu\text{m}$ ]	895.2517	0

Tabulka 7: Porovnání přesných a zkalibrovaných parametrů při chybě senzoru  $\varphi_{34} 10^{-3}$ , ostatní  $10^{-16}$

Jak můžeme vidět, v tomto případě získáváme z obou kalibrací naprosto odlišné výsledky. Rozdíly mezi skutečnými a výkresovými parametry jsou u některých parametrů základní kalibrace až v milimetrech, kdežto po pokročilé metodě mají kalibrované parametry rozměr přesných až na offset  $\phi_{340}$ . To jsme mohli po předchozích případech očekávat, protože tento offset je s tímto úhlem spjat. Výsledné změny parametrů jsou v tomto případě po pokročilé kalibraci výrazně přesnější a tím se nám potvrzují lepší vlastnosti této metody kalibrace.

Dále je zajímavé pozorovat, že výsledné rozdíly parametrů u základní kalibrace nabývají větších hodnot než v případě, kdy většina senzorů měla nepřesnosti v řádu je  $10^{-5}$  a senzor  $\phi_{34}$  byl zachybován stejným řádem chyby. Můžeme uvažovat, zdali se zvětšujícím se rozdílem chyb čidel, rostou také rozdíly parametrů skutečné a kalibrované hodnoty. Je možné, že právě tato skutečnost bude mít větší vliv na výsledný rozdíl parametrů než samotný řád nepřesnosti měření chyb senzorů. To nabízí myšlenku zjistit, zdali je tento trend opakující se a existuje obecná závislost mezi těmito jevy. To může být předmětem zkoumání jiné samostatné práce.

#### **2.4.5 Porovnání kalibrovatelnosti a směrodatných odchylek**

V této části kapitoly porovnáme hodnoty kalibrovatelnosti příkladů, které jsme zkalibrovali a jejichž výsledky jsme si ukázali výše pro obě metody kalibrace.

U základní kalibrace předpokládáme, že se hodnota kalibrovatelnosti mění pouze se změnou chyb senzorů, protože jak víme, kalibrovatelnost mechanismu v základní metodě můžeme ovlivnit pouze změnou Jacobiho matice  $J_q$ , ve které se vyskytují pouze parametry mechanismu. To, jak můžeme vidět v následující tabulce, bylo splněno. Pro všechny případy kalibrovatelnost vyšla 416.

Zajímavější výsledky jsme získali z pokročilé kalibrace. Jak můžeme vidět, hodnoty kalibrovatelnosti se výrazně odlišují jak od základní kalibrace pro všechny případy, tak se liší i v závislosti na změně nepřesnosti čidel. Kalibrovatelnost se výrazně mění s rostoucími chybami měření senzorů. Nicméně je otázka, jestli to je způsobené změnou řádu chyby nebo změnou hodnot odchylek pomocí „randn“. Nemůžeme ale nevidět, že se zvětšujícím se rozdílem řádu chyb těchto senzorů se kalibrovatelnost rapidně zvyšuje. Pro všechny senzory zachybované řádem  $10^{-16}$  vyšla 416, nicméně pro stejný řád chyby senzorů až na jeden senzor, jehož přesnost byla  $10^{-3}$ , vyšla kalibrovatelnost 1,3411e+27.



Metoda	Chyby senzorů [rad]	Kalibrovatelnost [-]
<b>Základní</b>	Všechny $10^{-16}$	416.2
	$\varphi_{34} - 10^{-3} - \text{ostatní } 10^{-16}$	415.3
	Všechny $10^{-5}$	416.3
	$\varphi_{34} - 10^{-3} - \text{ostatní } 10^{-5}$	416.4
	$\varphi_{34} - 10^{-4} - \text{ostatní } 10^{-5}$	416.3
<b>Pokročilá</b>	Všechny $10^{-16}$	416.4
	$\varphi_{34} - 10^{-3} - \text{ostatní } 10^{-16}$	1.3411e+27
	Všechny $10^{-5}$	1447.5
	$\varphi_{34} - 10^{-3} - \text{ostatní } 10^{-5}$	137958.0
	$\varphi_{34} - 10^{-4} - \text{ostatní } 10^{-5}$	3253.1

**Tabulka 8: Kalibrovatelnost příkladů**

Jak jsme mohli předpokládat, chyby senzorů hrají významnou roli u kalibrovatelnosti mechanismů, a jejich správné zavedení a zároveň správné definování kovariančních matic, je velice důležitým bodem pro obdržení správných výsledků.

Při zkoušení jsem narazil na zajímavý poznatek. Hodnota kalibrovatelnosti se mění minimálně se změnou řádu chyb čidel (stejně hodnoty zachybování, jiný řád), do té doby, dokud se nepřiblížíme velikostem parametrů. To je pro mě zajímavé zjištění a určitě stojí za bližší prozkoumání a pro někoho to může být inspirace pro další výzkum.

Ještě si ukážeme směrodatné odchylky senzorů čidel pro tyto příklady. Je zřejmé, že velikost všech směrodatných odchylek odpovídá řádům chyb, které jsme definovali. Směrodatné odchylky jsou stejné pro základní i pokročilou metodu daného příkladu.

Čidlo	Chyby senzorů $10^{-16}$	$\varphi_{34} - 10^{-3} - \text{ostatní } 10^{-16}$	Všechny $10^{-5}$	$\varphi_{34} - 10^{-3} - \text{ostatní } 10^{-5}$	$\varphi_{34} - 10^{-4} - \text{ostatní } 10^{-5}$
$x_M$	1.0358e-16	9.4909e-17	1.0543e-05	1.1023e-05	1.0759e-05
$y_M$	9.6165e-17	9.7254e-17	1.1015e-05	9.4616e-06	9.8755e-06
$\varphi_{12}$	9.4299e-17	9.0366e-17	1.0081e-05	9.3534e-06	1.0145e-05
$\varphi_{23}$	9.8880e-17	9.5778e-17	9.1548e-06	1.0437e-05	1.0759e-05
$\varphi_{34}$	9.5967e-17	9.4340e-04	1.0471e-05	9.7068e-04	9.9845e-05

**Tabulka 9: Směrodatné odchylky čidel všech příkladů kalibrací**

Tímto bych uzavřel téma kalibrace. Dle mého názoru jsme si velice pečlivě zdokumentovali, jak dané metody kalibrace fungují a ukázali jsme si několik příkladů, ve kterých jsme zjistili, jak vytvořené skripty pro danou metodu pracují.

## 2.5 Optimalizace

V této části diplomové práci si ukážeme optimalizované výsledky mechanismu, kdy optimalizovanou funkcí bude kalibrovatelnost. V posledním případě cílovou funkci změním. Z předchozích případů, na kterých jsme si zkoušeli kalibraci, si vybereme pouze dva. Některé výpočetní úkony jsou velice zdlouhavé a nejsme schopni obsáhnout všechny případy. Dále nemá smysl si ukazovat případ s absolutní přesností senzorů, protože jsme v takovém případě vypočítali také nulovou odchylku zkalibrovaných parametrů od skutečných a nemůžeme v tomto případě parametry zlepšit, což je mimo jiné to, co očekáváme od optimalizace. Proto si ukážeme případy, kde jsme chybu senzorů zavedli.

Nejprve budeme optimalizovat mechanismus, ve kterém jsou všechny chyby senzorů v řádu  $10^{-5}$ . Druhý příklad bude se stejnou chybou senzoru až na senzor  $\phi_{340}$ . Nepřesnost tohoto senzoru je v řádu  $10^{-3}$ .

Vyzkoušíme celkem tři metody optimalizace. První bude optimalizace pomocí *fminsearch*, kdy výsledek tohoto řešení bude velice záležet na zadaných počátečních hodnotách parametrů. Je proto pravděpodobné, že touto metodou zjistíme pouze lokální řešení a ne globální.

Globální řešení bychom mohli obdržet pomocí optimalizace genetickými algoritmy, která bude následovat jako další. Nevýhoda této metody je bohužel několikanásobně delší výpočetní čas (záleží na podmínkách ukončení optimalizace). Dále se také může stát, že optimální řešení (globální) daného problému se nemusí vůbec nalézt. Genetické algoritmy díky náhodnému výběru velice rychle naleznou nadějnou oblast potenciálního minima fitness funkce, nicméně ke globálnímu minimu konvergují pomaleji. Proto si u této metody zavedeme maximální počet generací, které chceme vypočítat. Počet generací zvolíme podle času výpočtu a tvaru křivky, která bude aproximovat vypočtené hodnoty cílové funkce.

V posledním případě použijeme pravděpodobně nejlepší postup optimalizace. Použijeme totiž jak genetické algoritmy, tak i *fminsearch*. Nejdříve vypočítáme minimum pomocí genetických algoritmů. Toto řešení nám ukáže nadějnou oblast, ve které by se mohlo vyskytovat globální minimum optimalizovaného parametru. Výsledky si uložíme a použijeme je jako počáteční hodnoty v optimalizaci pomocí *fminsearch*. Tento postup by měl být nejlepší a výsledná hodnota kalibrovatelnosti by již mohla nabývat globálního minima.

Ještě si musíme zavést omezující podmínky, ve kterých budeme hledat optimální řešení genetickými algoritmy. Hranice, které jsme zvolili, jsou ukázané v následující tabulce. Podmínky jsme zvolili velice volně a neměli by nás omezovat v možnosti nalezení optimálního řešení. V případě reálného mechanismu je potřeba se na ně zaměřit více a určit je podle umístění, kde se mechanismus bude nacházet, nebo například zástavbového prostoru. V našem případě je pravděpodobné, že se k hranicím takto definovaných intervalů vůbec nepřiblížíme a výsledné optimální řešení bude „někde uprostřed“.

<i>Parametr</i>	<i>Interval</i>
$x_A$	(-5; 5)
$y_A$	(-5; 5)
$l_2 = l_3 = l_4$	(0.1; 3)
$\varphi_{120} = \varphi_{230} = \varphi_{340}$	(0; $2\pi$ )

**Tabulka 10: Interval parametrů pro genetické algoritmy**

Dále si ukážeme, jak jsme definovali optimalizační funkce a za jakých podmínek dojde k ukončení výpočtu. Pro optimalizaci *fminsearch* jsme použili příkaz následujícím způsobem:

$$[x, fval, exitflag, output] = \text{fminsearch}(@x) \quad (2.5.1)$$

$$\text{cilova\_zakladni\_fminsearch}(x, \text{Poloha\_M}, P\_sk, \text{Chyba\_cidel}, KOV), p0, \text{options})$$

Na levé straně rovnice je zobrazen výstup, který obdržíme po konci optimalizace, kde  $x$  jsou parametry,  $fval$  minimalizovaná hodnota cílové funkce,  $exitflag$  značí, jak byla optimalizace ukončena a  $output$  vypíše vlastnosti optimalizace (počet vypočtených generací, počet vypočtených funkcí, ...). Na pravé straně máme definován optimalizační příkaz *fminsearch*. Inputem tohoto příkazu je cílová funkce, kterou chceme optimalizovat, v tomto případě *cilova\_zakladni\_fminsearch*,  $p0$ , kde jsou uloženy počáteční parametry a

*options*. Pomocí *options* popisujeme, jak má optimalizace probíhat a vlastnosti pro její ukončení.

Podmínky pro ukončení optimalizace jsme definovali tímto způsobem:

```
options = optimset('Display','iter','Tolx',1e-8,'TolFun',... (2.5.2)
... 1e-4,'MaxIter',10000,'PlotFcns',@optimplotfval)
```

Optimalizace pomocí *fminsearch* jsem se rozhodl ukončit v případě, že bude změna parametrů menší než  $10^{-8}$ , změna optimalizované funkce menší než  $10^{-4}$  nebo dosáhneme-li daného počtu iterací. Počet iterací upravujeme podle toho, jakou metodou zrovna optimalizujeme a jak vypadá graf cílové funkce. Budeme se ale pohybovat v řádů stovek iterací.

Příkaz pro genetické algoritmy vypadá následovně:

```
[x, fval, exitflag, output] = (2.5.3)
= ga(@(x) cilova_zakladni_ga(x, Poloha_M, P_sk, Chyba_cidel, KOV),
length(p0), [], [], [], [], LoB, UpB, [], [], options)
```

kde *ga* označuje příkaz pro genetické algoritmy. Význam ostatních prvků je stejný jako v předchozím *fminsearch*. Místo počátečních parametrů jsme definovali celkový počet optimalizovaných parametrů pomocí *length(p0)* (celkem 8) a určili spodní hranici hodnot parametrů jako *LoB* a horní jako *UpB*. Tyto hodnoty souhlasí s tabulkou 10. Předpis pro ukončení výpočtu jsou následující:

```
options = optimoptions('ga','MaxGenerations',100,' (2.5.4)
MaxStallGenerations',Inf,'PlotFcn',@gaplotbestfun,'Display','iter')
```

Maximální počet generací (*MaxGeneration*) jsem pro optimalizaci základní kalibrace zvolil 100. Pro optimalizace pokročilé kalibrace z důvodu výrazně delšího výpočtu jsem zvolil 20 generací. Na grafech následně uvidíme, zdali metody již konvergují nebo bude potřeba počet generací upravit. Velikost populace bude 50 jedinců pro všechny případy, což je doporučená hodnota při počtu proměnných větších než pět.

Ještě si připomeneme skutečnost, že pro porovnávání výsledků musíme mít data zachybovaná naprosto stejně pro všechny tři metody optimalizace. Proto si data chyb jak parametrů, tak i čidel vytvoříme úplně na začátku a dále je budeme již pouze kopírovat. Teď už jsme si představili všechny potřebné parametry pro optimalizaci a můžeme přejít k zobrazení výsledků.

## 2.5.1 Chyby měření všech senzorů $10^{-5}$

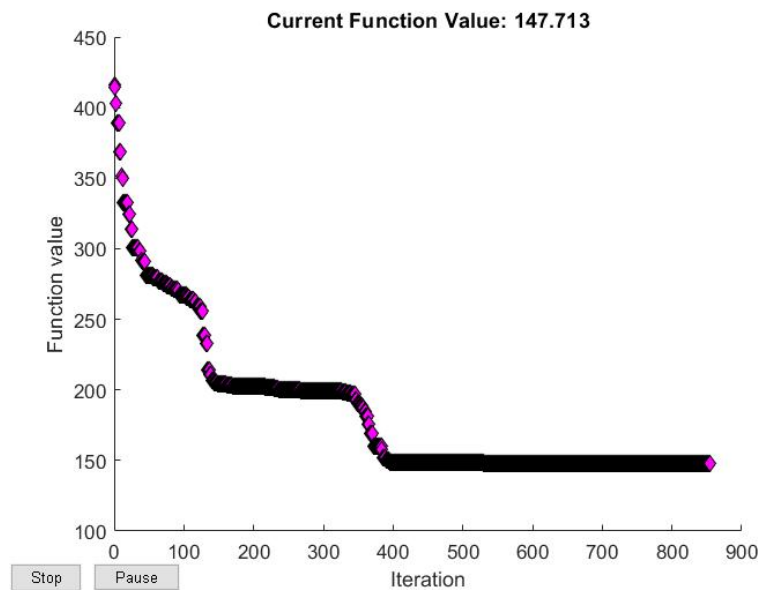
První případ, který budeme uvažovat, je zachybování nepřesnosti senzorů v řádu  $10^{-5}$ . Data jsou zachybovaná stejně jako v kapitole 2.4.2. Pro všechny metody, které si ukážeme, jsme si vygenerovali naprosto stejné odchylky parametrů i nepřesnosti chyb čidel. Ukážeme si samozřejmě postupný vývoj optimalizované funkce, tedy kalibrovatelnosti. Výsledky optimalizace porovnáme s výsledky samotných kalibrací. Uvidíme, jestli a jak jsme se přiblížili optimalizovanými parametry ke skutečným.

### 2.5.1.1 Optimalizace základní metody kalibrace

Jako první si ukážeme výsledky pro optimalizaci základní metody kalibrace. Připomeneme si ještě, že kalibrovatelnost pro metodu základní kalibrace určujeme pomocí podmíněnosti Jacobiho matice, kterou získáme z parciálně zderivovaných vazbových rovnic podle parametrů.

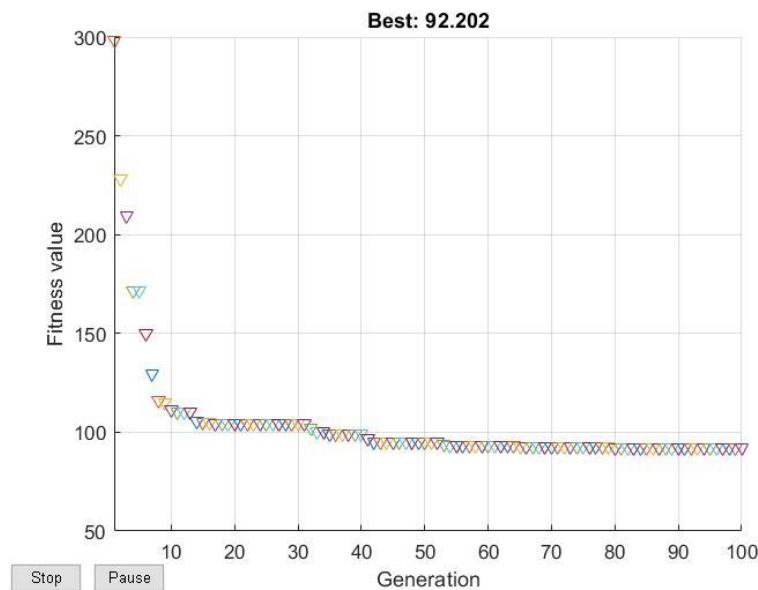
Jako první budeme optimalizovat pomocí *fminsearch*. Na následujícím grafu můžeme vidět vývoj kalibrace pro tuto metodu. Pro všechny optimalizace pomocí *fminsearch*, musí platit, že kalibrovatelnost v počátku je rovna kalibrovatelnosti, kterou jsme zjistili v samostatné kalibraci. Ověříme tím, že optimalizujeme správně určený mechanismus a neudělali jsme chybu například ve vkládání vstupních dat.

Jak můžeme vidět, to bylo v tomto případě splněno. Optimalizace má tedy stejné vstupní hodnoty jako základní kalibrace pro zachybování nepřesnosti čidel o řádu  $10^{-5}$ . Optimalizovaná funkce konverguje k hodnotě přibližně 147,7 a to již v okolí čtyřsté iterace. Minimální hodnotu kalibrovatelnosti, kterou jsme pro tento případ obdrželi, je 147,7. To je oproti počáteční hodnotě (416,3) zlepšení o 64 %. Tuto hodnotu bereme zatím pouze jako lokální minimum a v porovnání ostatními metodami optimalizace uvidíme, zdali to nebude i globální minimum.



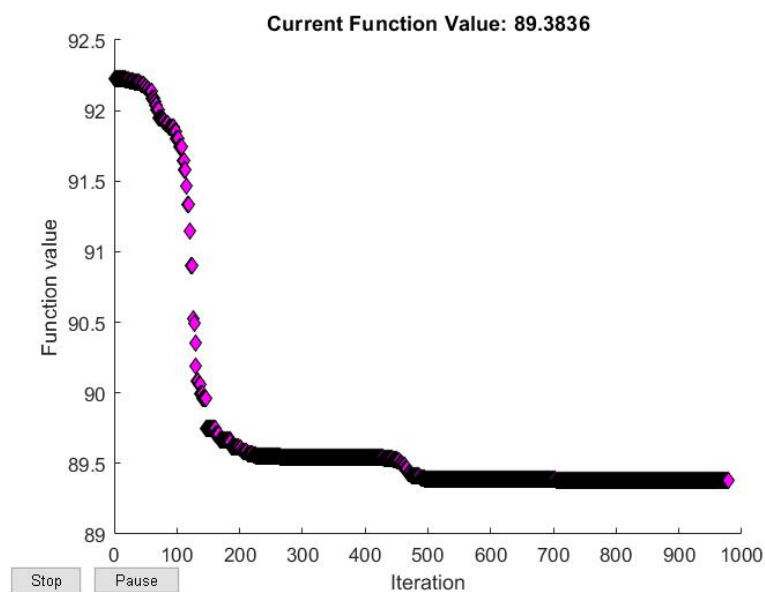
**Obr. 11: Vývoj kalibrovatelnosti –základní kalibrace – chyba senzorů  $10^{-5}$  – fminsearch**

Teď už přejdeme k optimalizaci pomocí genetických algoritmů, jejichž výsledek je znázorněn na dalším obrázku. V první generaci genetické algoritmy vypočítaly kalibrovatelnosti o velikosti 230,6. Jelikož ale víme, jak pracují genetické algoritmy (především na základě náhodného výběru) je zřejmé, že je to pouze náhoda a pokud bychom chtěli pustit tento výpočet znovu, počáteční hodnota by se určitě změnila. Jak ale můžeme vidět z grafu, kalibrovatelnost jsme znovu úspěšně minimalizovali. Ubrali jsme dalších 50 jednotek a po optimalizaci pomocí genetických algoritmů jsme získali hodnotu kalibrovatelnosti 92,2. To je velice pěkný výsledek a později uvidíme, jak se promítne do rozdílů parametrů. Pro optimalizace pomocí genetických algoritmů jsme potřebovali přibližně 60 generací. Tato hodnota je však pouze orientační. Víme, že genetické algoritmy konvergují k finální hodnotě pomalu. To nám potvrdilo dalších 40 generací, kdy se finální hodnota kalibrace nezměnila ani o jedno procento. Z toho důvodu očekávám, že jsme získali globální minimum nebo se mu budeme alespoň výrazně přibližovat.



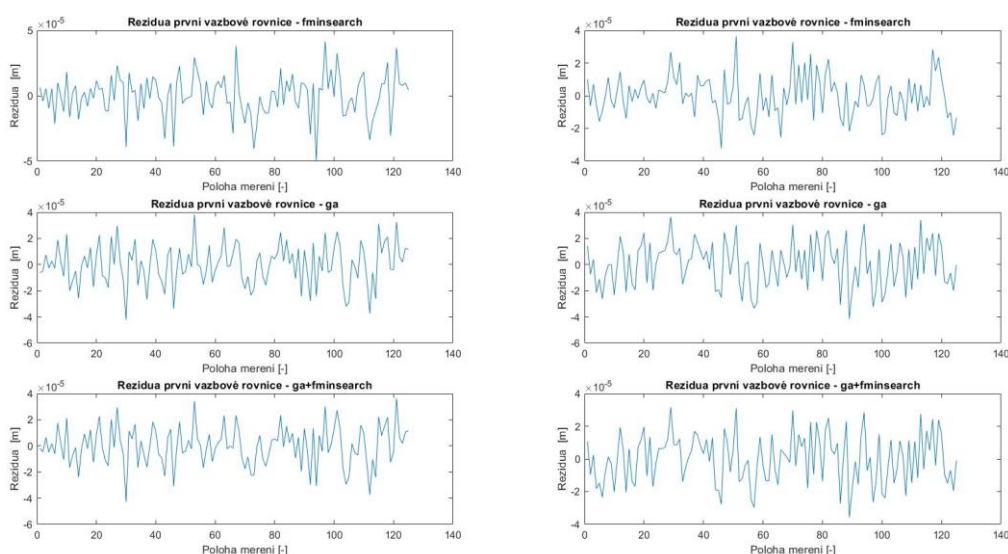
Obr. 12: Vývoj kalibrovatelnosti –základní kalibrace – nepřesnosti  $10^{-5}$  – genetické algoritmy

Zda jsme již získali globální minimum ověříme pomocí opětovného použití *fminsearch*. Tentokrát ale počátečním hodnotám parametrů zadáme takovou velikost, aby odpovídaly parametrům zjištěným v nejlepším pokusu optimalizace pomocí genetických algoritmů. Po nahrání těchto hodnot můžeme vidět, že se minimální hodnota kalibrovatelnosti posunula o téměř 3 jednotky, a to na 89,4. Není to tak výrazný posun, ale stále je to posun o více skoro tři procenta. Z tohoto malého posunu můžeme soudit, že jsme již s velkou pravděpodobností našli minimum globální. Vývoj kalibrovatelnosti pro tuto metodu můžeme také vidět na dalším obrázku.



Obr. 13: Vývoj kalibrovatelnosti –základní kalibrace – chyba senzorů  $10^{-5}$  - genetické algoritmy + *fminsearch*

Ukážeme si také rezidua vazbových rovnic. Ty můžeme vidět na následujícím obrázku. Jak jsme očekávali, rezidua jsou podobné jako v samostatné základní kalibraci, a to o velikosti námi definovaného řádu chyb čidel, v tomto případě  $10^{-5}$ .



Obr. 14: Rezidua vazbových rovnic – základní kalibrace – chyba senzorů  $10^{-5}$

Zbývá nám porovnat změny parametrů optimalizovaných hodnot a skutečných hodnot. Doufejme, že optimalizace zafunguje, a rozdíl optimalizovaných parametrů od skutečných bude menší než v případě obyčejné kalibrace. V následující tabulce toto porovnání společně se zjištěnou kalibrovatelností můžeme vidět.

Parametr	Základní kalibrace	fminsearch	Globální algoritmy	GA + fminsearch
$\Delta x_A$ [ $\mu\text{m}$ ]	2.6409	2.8120	1.5582	0.7537
$\Delta y_A$ [ $\mu\text{m}$ ]	4.7754	5.7994	3.9033	5.7901
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	1.7595	0.2365	0.1028	1.9834
$\Delta l_2$ [ $\mu\text{m}$ ]	0.9330	0.1432	2.0809	3.4400
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	1.4660	0.4702	0.3251	1.4994
$\Delta l_3$ [ $\mu\text{m}$ ]	0.6763	1.6603	2.0990	1.8801
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	1.9039	3.7504	1.3364	1.1689
$\Delta l_4$ [ $\mu\text{m}$ ]	1.8692	5.1179	2.5614	4.5329
Kalibrovatelnost	416.3	147.7	92.2	89.4
Průměrný rozdíl parametru [ $\mu$ -]	3.4631	2.4987	1.7458	2.6 310

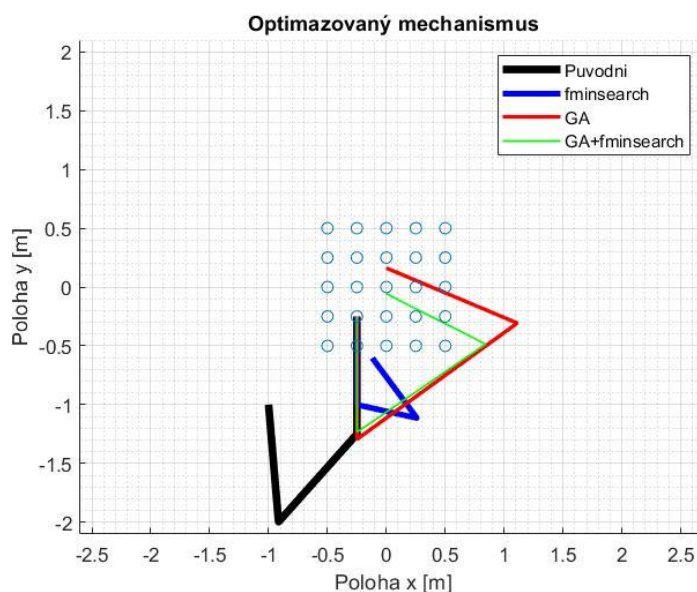
Tabulka 11: Porovnání metod optimalizace a základní kalibrace – chyba senzorů  $10^{-5}$



Pokud se podíváme na výsledné rozdíly parametrů metod optimalizace, z prvotního pohledu není vidět výrazné zlepšení. Pokud ale zprůměrujeme chyby všech senzorů, rozdíl se již projeví. Z tohoto porovnání vychází k mému překvapení nejlépe optimalizace pomocí genetických algoritmů. Pokud bychom ale porovnávali parametry samostatně, nelze vybrat metodu, která by byla lepší. Změny parametrů jsou v některých případech výrazně lepší, zároveň jsou ale v některých případech i výrazně horší, případně jsou podobné. Průměrný rozdíl parametru se po optimalizaci zlepšil pro metodu *fminsearch* o 28 % v porovnání samotné kalibrace. Metodou genetických algoritmů dosáhneme zlepšení o téměř 49,8 % a metodou GA a *fminsearch* o 24 %.

Jak už jsem zmínil, k mému překvapení nejlépe dopadly genetické algoritmy. Očekával jsme spíše poslední metodu GA a *fminsearch*, že dopadne nejlépe, ale při porovnání můžeme zjistit, že se rozdíl parametrů zhoršil.

Porovnali jsme si optimalizaci základní metody a jako poslední si ukážeme porovnání optimalizovaných návrhů mechanismu pro všechny tři metody společně s počátečním. Černá čára představuje původní mechanismus, modrá představuje optimalizaci pomocí *fminsearch*, červená pomocí genetických algoritmů a zelená pomocí spojení obou předchozích metod. Tento popis bude pro každý následující obrázek stejný. Bod mechanismu A (označili jsme v kapitole 2.1.) se se snižující kalibrovatelností v tomto případě posouvají blíže středu pracovního prostoru. To jsme mohli předpokládat.

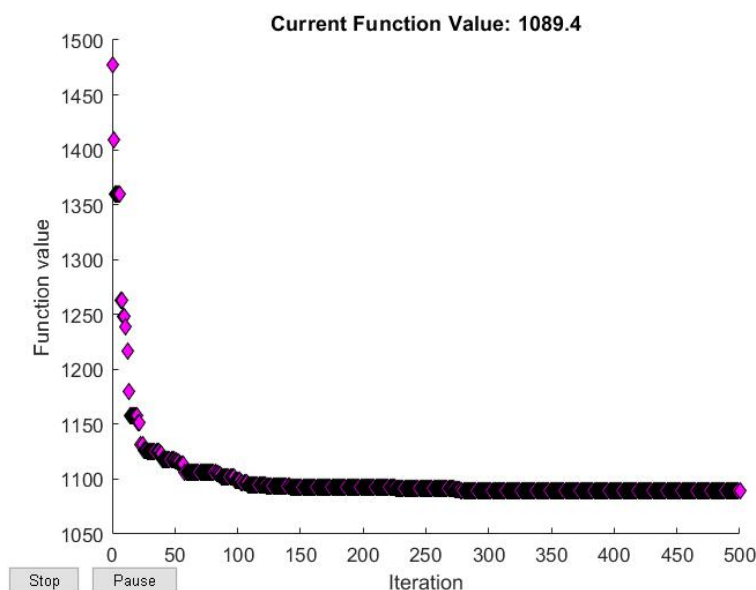


Obr. 15: Optimalizované návrhy mechanismu– základní kalibrace – chyba senzoru  $10^{-5}$

### 2.5.1.2 Optimalizace pokročilé metody kalibrace

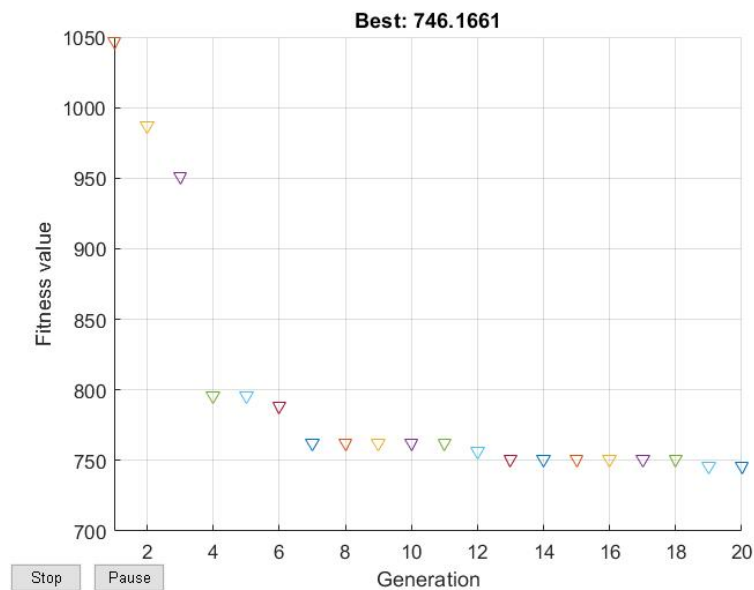
V této části si ukážeme výsledky z optimalizace pomocí pokročilé metody kalibrace. Postup bude stejný jako u předchozího příkladu. Nejprve zjistíme výsledky optimalizace pomocí `fminsearch`, následně pomocí genetických algoritmů a zakončíme to použitím obou metod.

Postupný vývoj kalibrovatelnosti pro metodu `fminsearch` můžeme vidět na následujícím grafu. Jak můžeme pozorovat, cílovou funkci jsme znovu výrazně zmenšili, a to z 1447,5 na 1089,4. Podobně jako v případě optimalizace pomocí základní kalibrace, i v tomto případě došlo k výraznému snížení, a to o přibližně o 25 %. Podle grafu to vypadá, že jsme našli lokální minimum a optimalizace zkonvergovala. Proto jsme jí po 500 iteracích ukončili. Další výraznější pokles již při vyšším počtu iterací neočekávám, přesto to nevylučuji. Globální minimum v tomto případě pravděpodobně stejně nenalezneme, a proto se přesuneme na globální metodu.



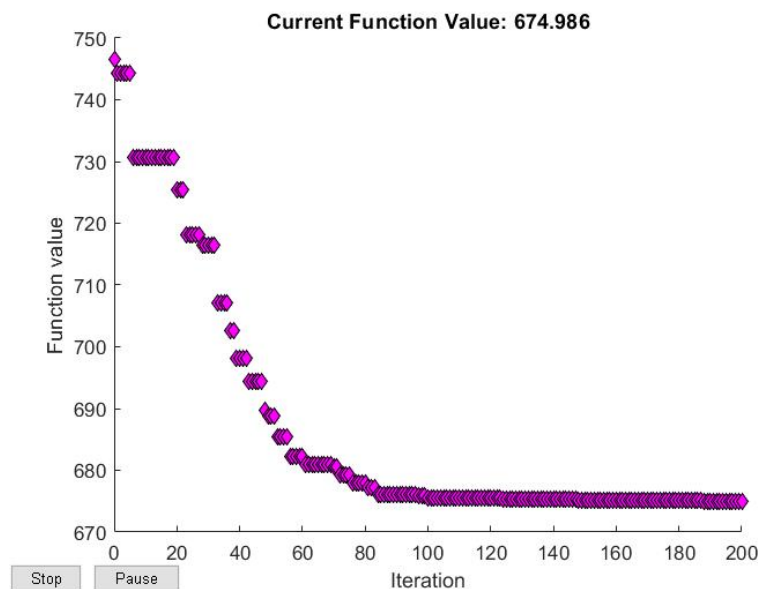
Obr. 16: Vývoj kalibrovatelnosti –pokročilá kalibrace – chyba senzorů  $10^{-5}$  – `fminsearch`

Jak jsme předpokládali, tak globální minimum kalibrovatelnosti jsme určitě nenalezli. K tomuto zjištění se stačí podívat na výsledky z optimalizace pomocí genetický algoritmů, kdy i prvotní odhad genetických algoritmů měl menší cílovou funkci (1048,1). Následně jsme jí ještě optimalizovali na 746,2. Vypadá to, že by křivka mohla konvergovat okolo hodnoty 740, nicméně k tomuto tvrzení 20 generací nestačí. Stálo by za to vyzkoušet vyšší počet generací. Nicméně uvidíme, co s tím ještě udělá kombinace obou metod.



Obr. 17: Vývoj kalibrovatelnosti –pokročilá kalibrace – chyba senzorů 10<sup>-5</sup> -genetické algoritmy

Jak můžeme vidět na dalším obrázku, kalibrovatelnost jsme poslední metodou ještě snížili o dalších 72 bodů na 675. Tato metoda, zdá se již zkonvergovala a výsledek optimalizace považujeme za hodnotu globálního minima.



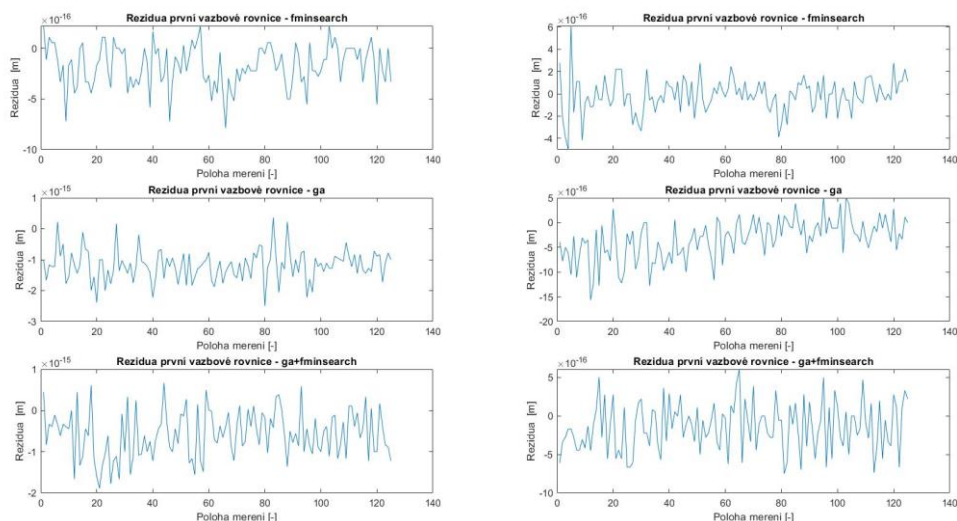
Obr. 18: Vývoj kalibrovatelnosti –pokročilá kalibrace – chyba senzorů 10<sup>-5</sup> – genetické algoritmy +fminsearch

V tabulce můžeme vidět, jak se projevila optimalizace na změně rozdílů skutečných a zkalibrovaných hodnot parametrů.

Parametr	Pokročilá kalibrace	fminsearch	Globální algoritmy	GA + fminsearch
$\Delta x_A$ [ $\mu\text{m}$ ]	2.609	6.0936	10.1997	6.6500
$\Delta y_A$ [ $\mu\text{m}$ ]	4.7754	1.7691	9.6353	11.2175
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	1.7595	1.0030	4.0376	3.8620
$\Delta l_2$ [ $\mu\text{m}$ ]	0.9330	1.5004	10.4543	15.8071
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	1.4660	1.8155	1.6531	0.3141
$\Delta l_3$ [ $\mu\text{m}$ ]	06763	2.3324	10.1304	14.7940
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	1.9039	3.7488	6.2397	2.1992
$\Delta l_4$ [ $\mu\text{m}$ ]	1.8692	1.4123	7.1650	6.9632
Kalibrovatelnost	1447.5	1089.4	746.1661	674.9862
Průměrný rozdíl parametru [ $\mu$ -]	2.0030	2.4594	7.4394	7.7259

Tabulka 12: Porovnání metod minimalizace pokročilé kalibrace – chyba senzoru  $10^{-5}$

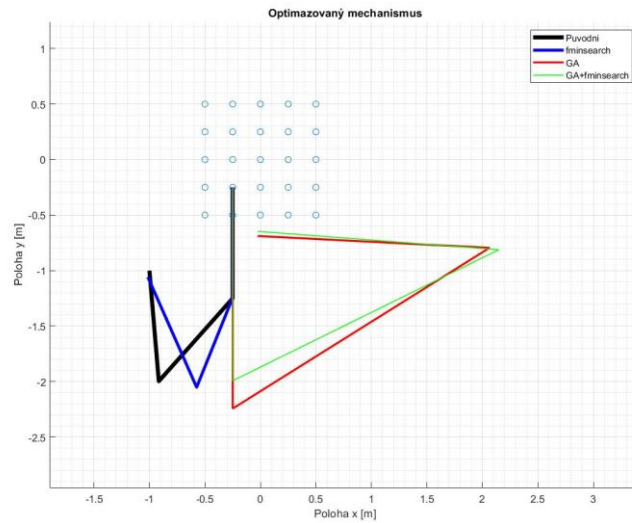
Jak můžeme vidět kalibrovatelnost optimalizacemi jsem sice snížili, nicméně se nám v tomto případě rozdíly parametrů výrazně zhoršily, a to téměř čtyřikrát pro poslední metodu. Grafy reziduí vazbových rovnic vidíme na dalším obrázku a podobně jako při samostatné pokročilé kalibraci nabývají hodnoty řádu  $10^{-15}$  pro všechny optimalizace.



Obr. 19: Rezidua vazbových rovnic – pokročilá kalibrace – chyba senzorů  $10^{-5}$

Znázornění optimalizovaných mechanismů pro metody optimalizace vidíme níže. Podobně jako v předchozím příkladě, mechanismus po optimalizaci *fminsearch* zůstal v blízkosti původního návrhu mechanismu. Po použití genetických algoritmů se bod

mechanismu A posunul znovu blíže ke středu pracovního prostoru. Použití *fminsearch* s počátečními hodnotami z GA už výrazný vliv nemělo.



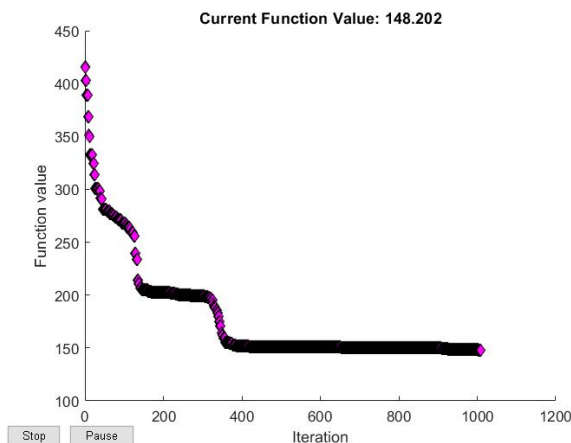
Obr. 20: Optimalizované návrhy mechanismu – pokročilá kalibrace – chyba senzoru  $10^{-5}$

## 2.5.2 Chyby měření senzoru $\varphi_{34} 10^{-3}$

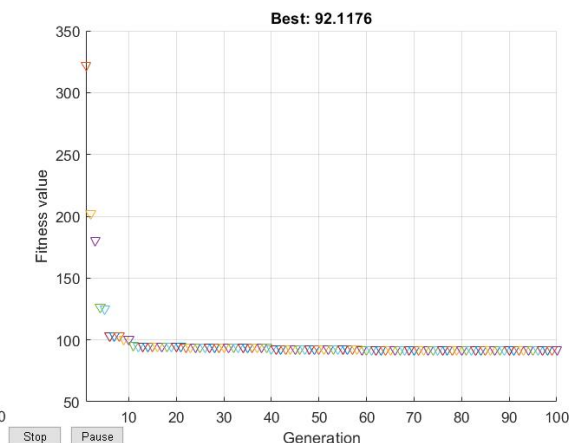
V této části si ukážeme výsledky optimalizace pro zachybování senzorů v řádu  $10^{-5}$  až na  $\varphi_{34}$ . Tento senzor zachybujeme náhodně v řádu  $10^{-3}$ .

### 2.5.2.1 Optimalizace základní metody kalibrace

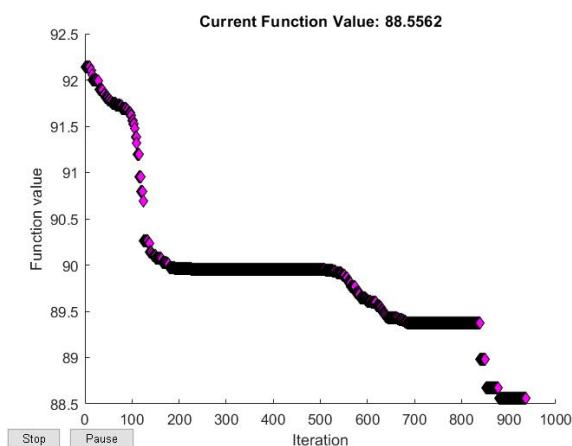
Nejprve si znovu ukážeme optimalizaci základní metody kalibrace. Na následujících třech obrázcích můžeme vidět postupnou minimalizaci kalibrovatelnosti danými metodami. Jak můžeme vidět, grafy i výsledné hodnoty jsou velice podobné jako v případě popsaném v kapitole 2.5.1.1. Výsledné hodnoty kalibrovatelnosti se liší maximálně v jednotce procenta. To pro nás ale nemůže být překvapivé, když si uvědomíme, že Jacobiho matice je pro oba případy totožně definovaná. Počáteční hodnota kalibrovatelnosti byla 416,4 a postupně jsme ji optimalizací dokázali snížit (148,2 - *fminsearch*, 92,4 - genetické algoritmy) na 89,4 (*GA+fminsearch*).



Obr. 22: Vývoj kalibrovatelnosti –základní kalibrace chyba senzoru  $\varphi_{34} 10^{-3}$  – fminsearch

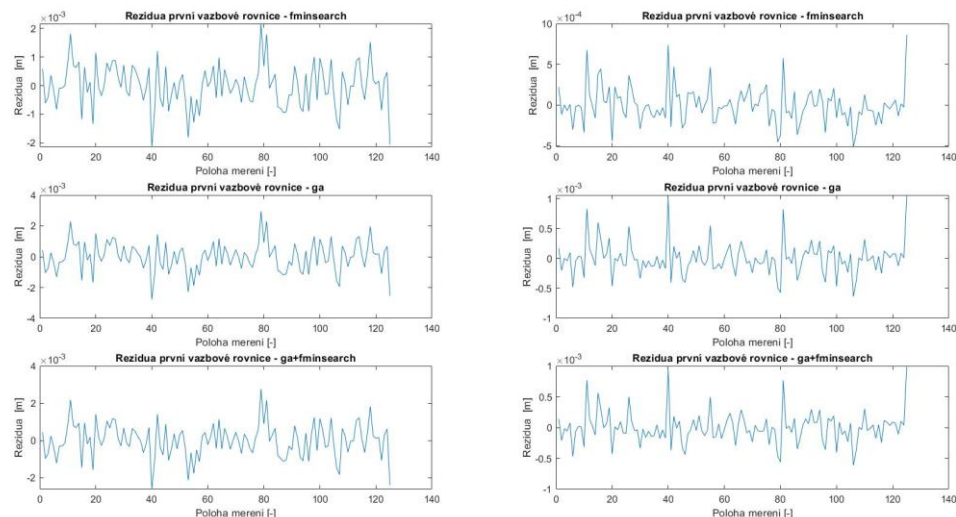


Obr. 21: Vývoj kalibrovatelnosti – základní kalibrace chyba senzoru  $\varphi_{34} 10^{-3}$  – GA + fminsearch



Obr. 23: Vývoj kalibrovatelnosti – základní kalibrace – chyba senzoru  $\varphi_{34} 10^{-3}$  – GA + fminsearch

Dále si ukážeme rezidua vazbových rovnic. Očekáváme, že budou vypadat podobně jako v případě samostatné kalibrace a jak můžeme vidět, tento předpoklad byl splněn. Rezidua vazbových rovnic jsou u všech metod optimalizace v řádu  $10^{-3}$  až  $10^{-4}$  podobně jako při samotné kalibraci se stejným zachybováním.



Obr. 24: Rezidua vazbových rovnic – základní kalibrace – chyba senzoru  $\phi_{34} 10^{-3}$

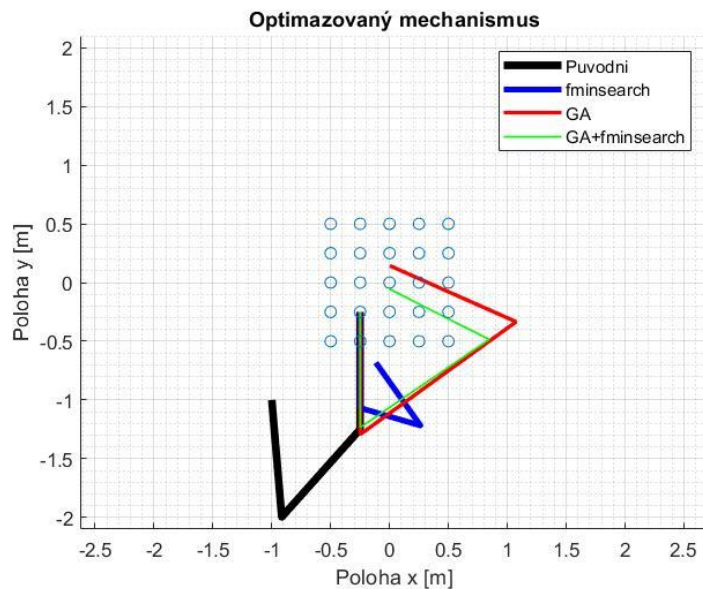
Níže si ukážeme, jak se změnila rozdíly parametrů. Podobně jako dříve, i v této metodě jsme zoptimalizovali cílovou funkci kalibrace. Na změně rozdílu parametrů se to projevilo pozitivně, nicméně k mému překvapení vyšla v tomto ohledu nejlépe optimalizace první (fminsearch), a to o téměř 30 % oproti zbylým dvěma optimalizacím. Oproti samotné kalibraci jsme zaznamenali zlepšení o 66 %.

Parametr	Základní kalibrace	fminsearch	Globální algoritmy	GA + fminsearch
$\Delta x_A$ [ $\mu\text{m}$ ]	200.8194	86.6795	145.7699	105.6813
$\Delta y_A$ [ $\mu\text{m}$ ]	259.4526	272.1380	215.9174	221.4365
$\Delta \phi_{120}$ [ $\mu\text{rad}$ ]	128.3754	63.1359	256.9790	266.7864
$\Delta l_2$ [ $\mu\text{m}$ ]	274.6754	147.0627	51.7898	65.1800
$\Delta \phi_{230}$ [ $\mu\text{rad}$ ]	118.3909	17.1595	80.7051	105.9057
$\Delta l_3$ [ $\mu\text{m}$ ]	144.8728	11.6958	196.4995	179.6985
$\Delta \phi_{340}$ [ $\mu\text{rad}$ ]	152.3961	104.4827	6.5713	8.4999
$\Delta l_4$ [ $\mu\text{m}$ ]	82.6053	160.9173	84.4104	103.8681
Kalibrovatelnost	416.4	148.2	92.1	88.6
Průměrný rozdíl parametru [ $\mu$ -]	170.1985	107.9089	129.8303	132.1320

Tabulka 13: Porovnání metod minimalizace – základní kalibrace – chyba senzoru  $\phi_{34} 10^{-3}$  – ostatní  $10^{-5}$

Poslední věc, která nám zbývá ukázat si v této metodě je, jaké schéma mechanismu získáme po optimalizacích. Můžeme to vidět na následujícím obrázku. Barevně jsou

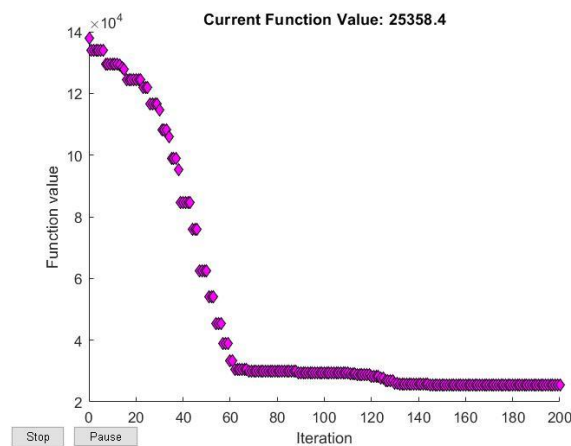
mechanismy popsané stejně jako v předchozích grafech. Jak vidíme, mechanismy po optimalizaci zaujaly téměř totožnou polohu jako mechanismy při optimalizaci základní kalibrace, kdy jsme zachybovali čidla řádem chyby  $10^{-5}$ . To znovu není žádným překvapením, jestliže změna chyby čidla nemá v této metodě vliv na Jacobiho matici.



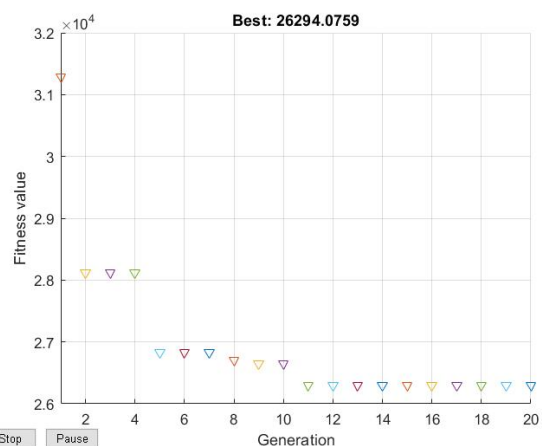
Obr. 25: Optimalizované návrhy mechanismu – základní kalibrace – chyba senzoru  $\varphi_{34} 10^{-3}$

### 2.5.2.2 Optimalizace pokročilé metody kalibrace

Teď si ukážeme výsledky optimalizace pokročilé metody pro tento případ. Stejně jako v předchozích případech i tady můžeme na následujících obrázcích vidět postupný vývoj minimalizace kalibrovatelnosti danými metodami.

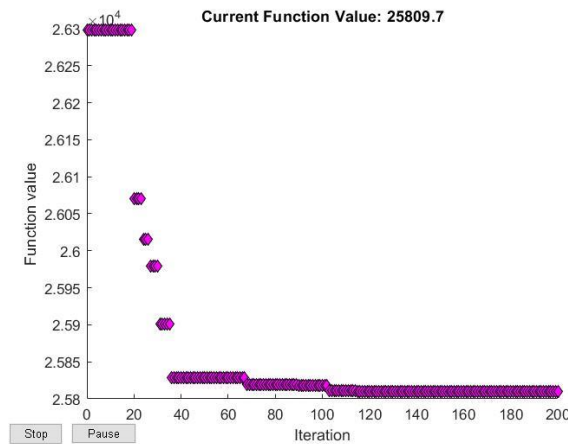


Obr. 27: Vývoj kalibrovatelnosti – pokročilá kalibrace  
chyba senzoru  $\varphi_{34} 10^{-3}$  - fminsearch



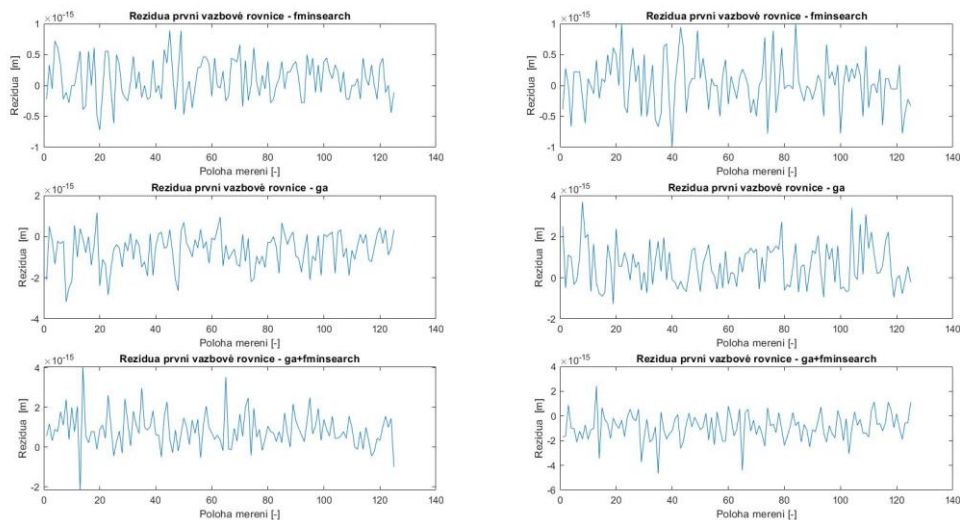
Obr. 26: Vývoj kalibrovatelnosti – pokročilá kalibrace  
chyba senzoru  $\varphi_{34} 10^{-3}$  – genetické algoritmy





Obr. 28: Vývoj kalibrovatelnosti – pokročilá kalibrace – chyba senzoru  $\varphi_{34} 10^{-3}$  – GA + fminsearch

K mému překvapení jsme v tomto případě získali nejnižší hodnotu kalibrovatelnosti z první optimalizace pomocí *fminsearch*, a to 25358,4. Je zajímavé, že genetické algoritmy nenašli řešení lepší. Způsobeno to může být malým počtem vypočtených generací. Dle grafu to však vypadá, že se metoda zkonvergovala u stávající hodnoty a z tohoto důvodu jsme větší počet generací nepočítali. Následné použití *fminsearch* už na výslednou hodnotu velký vliv nemělo. Rezidua vazbových rovnic, která můžeme vidět na následujícím obrázku, znovu nabývají řádu  $10^{-16}$ . Jak můžeme pozorovat, to se děje ve všech kalibracích pokročilou metodou.



Obr. 29: Grafy reziduí optimalizace pokročilé kalibrace – chyba senzoru  $\varphi_{34} 10^{-3}$

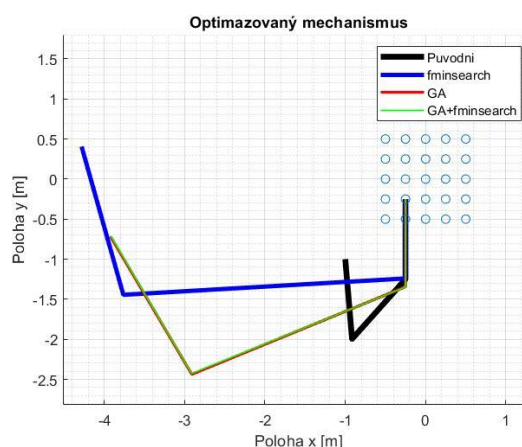
V další tabulce znovu vidíme porovnání rozdílů senzorů. Stejně jako u optimalizace základní metody kalibrace, i v tomto případě jsme získali nejlepší zlepšení pro optimalizaci pomocí *fminsearch*. Jak ale vidíme, průměrné hodnoty jsou od samostatné kalibrace o

třetinu horší. Zbylé dvě metody optimalizace jsou na průměrnou chybu parametru horší více než dvakrát.

Parametr	Pokročilá kalibrace	Fminsearch	Globální algoritmy	GA + fminsearch
$\Delta x_A$ [ $\mu\text{m}$ ]	5.0620	4.0411	6.6097	4.6455
$\Delta y_A$ [ $\mu\text{m}$ ]	68.0387	102.3873	272.5224	242.8114
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	0.8266	1.7296	36.6770	30.4112
$\Delta l_2$ [ $\mu\text{m}$ ]	3.3254	15.7147	85.7839	71.9292
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	0.7874	2.0447	14.6420	13.0267
$\Delta l_3$ [ $\mu\text{m}$ ]	2.2500	11.2677	23.3978	15.7764
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	94.8574	102.2677	124.6858	119.5992
$\Delta l_4$ [ $\mu\text{m}$ ]	70.7411	88.2574	93.1534	97.0229
Kalibrovatelnost	137958.0	25358.4	26294.1	25809.6
Průměrný rozdíl parametru [ $\mu$ -]	30.7361	41.0259	82.1840	74.4028

Tabulka 14: Porovnání metod minimalizace a pokročilé kalibrace – chyba senzoru  $\varphi_{34} 10^{-3}$

Na posledním obrázku vidíme mechanismus pro všechny optimalizace. Tentokrát se bod A pro GA a pro GA+fminsearch posunul výrazně doleva až k hraniční podmínce, kterou jsme definovali. Zároveň se také prodloužilo druhé rameno. Tyto změny nastaly pravděpodobně proto, že se metody snaží minimalizovat změnu úhlu  $\varphi_{34}$  pro různé polohy mechanismu.



Obr. 30: Optimalizované návrhy mechanismu – pokročilá kalibrace – chyba senzoru  $\varphi_{34} 10^{-3}$

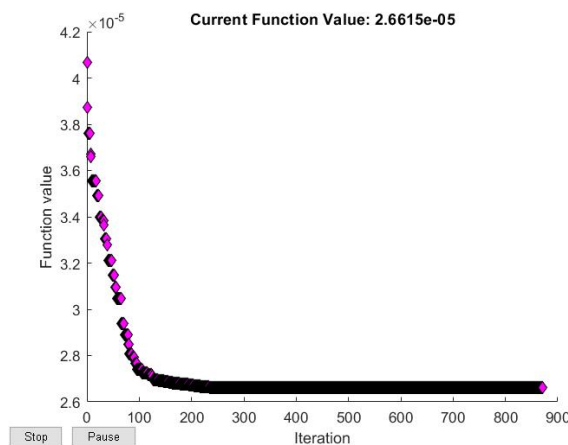
### 2.5.3 Chyby měření všech senzorů $10^{-5}$ – změna cílové funkce

V této části si ukážeme poslední případ použití optimalizace rovinného mechanismu. Tím bude stejný případ jako v kapitole 2.5.1, nicméně tentokrát budeme optimalizovat jinou cílovou funkci. Cílovou funkcí již nebude kalibrovatelnost, ale součet rozdílů skutečných hodnot parametrů a zkalibrovaných ve tvaru

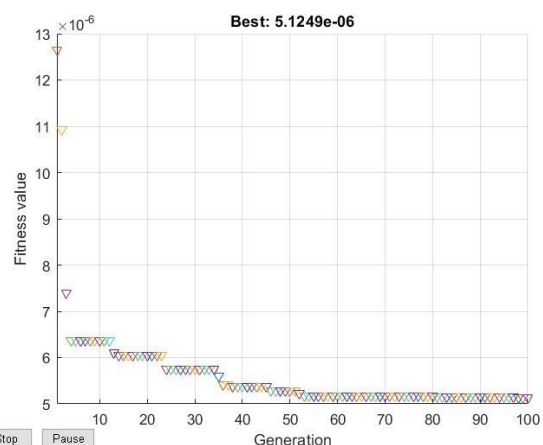
$$F = \Delta x_A + \Delta y_A + \Delta \varphi_{12} + \Delta l_2 + \Delta \varphi_{23} + \Delta l_3 + \Delta \varphi_{34} + \Delta l_4. \quad (2.5.5)$$

#### 2.5.3.1 Optimalizace základní metody kalibrace

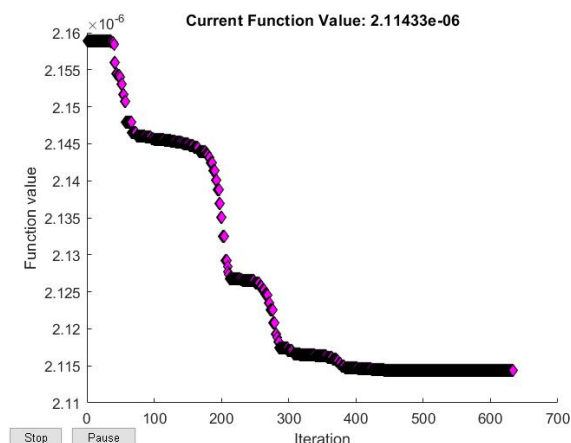
Jako první si znovu ukážeme optimalizace základní metodou. Výsledné grafy ze všech tří optimalizací můžeme vidět na následujících obrázcích, kde cílová funkce podobně jako v předchozích případech zkonvergovala. V tabulce, kde si porovnáme změny parametrů bude zajímavé sledovat, jak výrazně se optimalizace podepsala na výsledných hodnotách změn parametrů. Suma odchylek parametrů byla na počátku rovna  $4,06727e-5$ . Po optimalizaci pomocí `fminsearch` se změnila na  $2,6615e-5$ . To je zlepšení o 35 %. Lokální minimum jsme ale nenalezli. Nižších hodnot jsme dosáhli pomocí genetických algoritmů, které nám optimalizovali cílovou funkci na hodnotu  $5,1249e-6$ . To je velice razantní pokles o 87 %. Jak se to projeví na samostatných parametrech bude rovněž zajímavé sledovat. O dalších téměř 50 % jsme snížili optimalizovanou funkci pomocí `fminsearch` s počátečními podmínkami z GA, a to na  $2,1143e-6$ .



Obr. 32: Optimalizace sumy –základní kalibrace chyba senzorů  $10^{-5}$  - `fminsearch`



Obr. 31: Optimalizace sumy – základní kalibrace chyba senzorů  $10^{-5}$  - ga



Obr. 33: Optimalizace sumy – základní kalibrace – chyba senzorů  $10^{-5}$  – ga + fminsearch

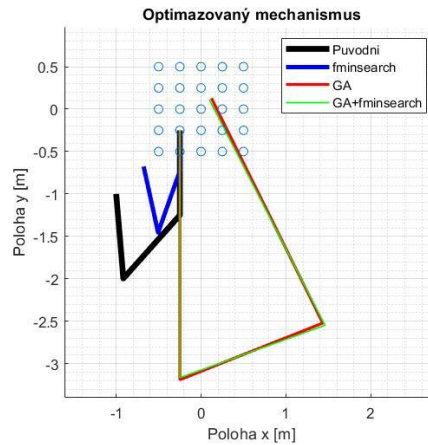
Jak vidíme, rozdíly parametrů se díky optimalizovaným metodám zlepšili, a to pro poslední dvě optimalizace více než desetkrát. Jak ale můžeme pozorovat, zároveň se s tím zvýšila hodnota kalibrovatelnosti, a to téměř čtyřikrát z 416,3 na 1531,7. Pro prvotní optimalizaci pomocí *fminsearch* se nicméně kalibrovatelnost snížila na 399,5.

Parametr	Základní kalibrace	Fminsearch	Globální algoritmy	GA + fminsearch
$\Delta x_A$ [ $\mu\text{m}$ ]	7.6815	0.2352	0.0097	0.0005
$\Delta y_A$ [ $\mu\text{m}$ ]	4.8917	8.6520	0.0276	0.0000
$\Delta\phi_{120}$ [ $\mu\text{rad}$ ]	1.4395	1.5347	0.2845	0.2932
$\Delta l_2$ [ $\mu\text{m}$ ]	1.2767	0.0000	0.0485	0.0000
$\Delta\phi_{230}$ [ $\mu\text{rad}$ ]	0.0979	0.0162	0.0091	0.0018
$\Delta l_3$ [ $\mu\text{m}$ ]	5.6949	4.0211	1.1454	1.1876
$\Delta\phi_{340}$ [ $\mu\text{rad}$ ]	3.4310	5.4384	0.6332	0.6313
$\Delta l_4$ [ $\mu\text{m}$ ]	3.1917	6.7173	0.0005	0.0000
Kalibrovatelnost	416.3	399.5	1522.6	1531.7
Průměrný rozdíl parametru [ $\mu$ -]	3.4631	3.3269	0.2698	0.2643

Tabulka 15: Porovnání metod minimalizace (změna FF) - základní kalibrace – chyba senzoru  $10^{-5}$

Rezidua vazbových rovnic jsou srovnatelná s optimalizací stejného zachybování chyb ale jiné cílové funkce, a proto si je nebudeme ukazovat. Schéma optimalizovaných mechanismů už ano, a to hned na dalším obrázku. Bod A se posunul znovu ke středu pracovního prostoru podobně jako dříve, nicméně tentokrát se prodloužily ramena  $l_2$  a  $l_4$  na

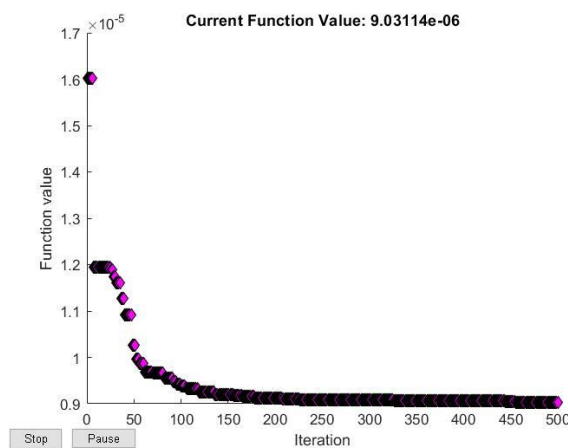
skoro 3 metry. To je mezní hodnota délky ramene, kterou jsme nastavili a je možné, že by se ramena prodloužila ještě více bez těchto podmínek.



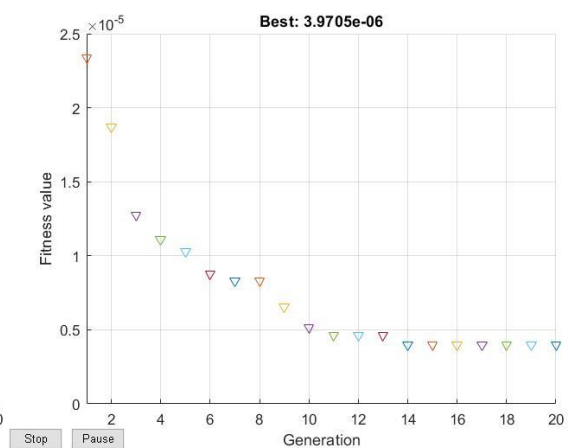
Obr. 34: Optimalizované návrhy mechanismu – základní kalibrace – chyba senzoru  $\varphi_{34} 10^{-3}$

### 2.5.3.2 Optimalizace pokročilé metody kalibrace

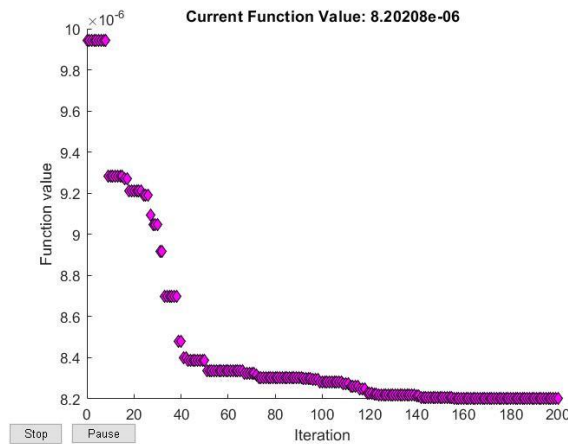
V této části si ukážeme optimalizaci pokročilé metody pro cílovou funkci zadanou jako sumu rozdílů parametrů. Jak můžeme vidět z následujících grafů, cílovou funkci jsme znovu úspěšně minimalizovali. První optimalizace pomocí fminsearch získala hodnotu  $9,03114e-5$ . Je zajímavé sledovat, že už prvotní generace genetických algoritmů zjistila hodnotu cílové funkce výrazně menší a to  $3,9705e-5$ . Znovu tedy vidíme, že fminsearch nenašel globální minimum, ale pouze lokální. Při použití fminsearch s počátečními podmínkami, které jsme získali z nejlepšího výpočtu fitness funkce z předchozí metody, jsme získali ještě nižší hodnotu, a to  $8,2020e-6$ . Tato hodnota by už globální minimum mohla být.



Obr. 36: Optimalizace sumy – pokročilá kalibrace chyba senzorů  $10^{-5}$  – fminsearch



Obr. 35: Optimalizace sumy – pokročilá kalibrace chyba senzorů  $10^{-5}$  – GA



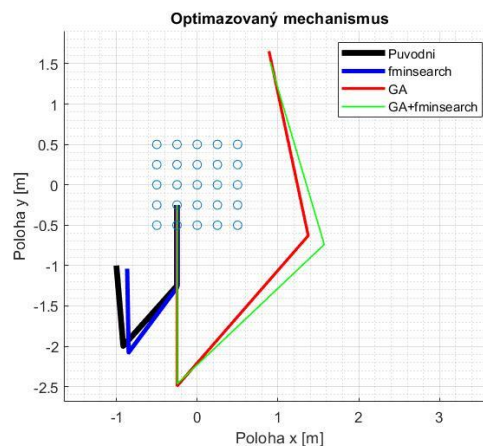
Obr. 37: Optimalizace sumy – pokročilá kalibrace – chyba senzorů  $10^{-5}$  – GA + *fminsearch*

Změny reziduí si ukazovat nebudeme, protože jsou znovu stejné jako v případě příkladu se stejnými chybami a jiné optimalizované funkce. Jak můžeme vidět všechny metody optimalizace výrazně snížily rozdíly parametrů. První metodou se průměrná odchylka parametrů snížila téměř na polovinu oproti odchylce ze samostatné kalibrace. Ještě přesnější výsledky parametrů získáme po optimalizaci pomocí genetických algoritmů i následného použití *fminsearch*. Zvýšila se ale pro změnu kalibrovatelnost na 1676,6. Genetické algoritmy snížili průměrnou odchylku o více než 50 % vůči *fminsearch*. Na to použitá poslední metoda znovu zlepšila výsledné hodnoty, a to o dalších 40 %. Oproti samostatné kalibraci jsou tyto hodnoty tedy výrazně přesnější, a to až téměř desetkrát přesnější pro GA + *fminsearch* oproti samotné kalibraci. Kalibrovatelnost se pro obě tyto metody snížila na 980 respektive 904. V této optimalizaci můžeme s jistotou určit, že nejlepší je použití metodu poslední, tedy spojení genetických algoritmů a *fminsearch*.

Parametr	Pokročilá kalibrace	Fminsearch	Globální algoritmy	GA + fminsearch
$\Delta x_A$ [ $\mu\text{m}$ ]	2.609	0.4772	0.0131	0.0041
$\Delta y_A$ [ $\mu\text{m}$ ]	4.7754	4.0131	0.1431	0.0355
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	1.7595	1.4178	0.2395	0.0099
$\Delta l_2$ [ $\mu\text{m}$ ]	0.9330	0.0322	0.9905	0.9077
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	1.4660	0.4366	0.7918	0.4473
$\Delta l_3$ [ $\mu\text{m}$ ]	06763	0.1162	1.0176	0.8936
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	1.9039	0.4722	0.3924	0.0103
$\Delta l_4$ [ $\mu\text{m}$ ]	1.8692	2.0666	0.3825	0.0761
Kalibrovatelnost	1447.5	1676.6	979.1	904.1
Průměrný rozdíl parametru [ $\mu$ -]	2.0030	1.1290	0.4963	0.2980

Tabulka 16: Porovnání metod minimalizace sumy – pokročilá kalibrace – chyba senzoru  $10^{-5}$

Naposledy si ukážeme, kam se nám mechanismus posunul pro vypočítané optimalizace. Tentokrát se bod mechanismu A pro druhou a třetí optimalizaci posunul nahoru a doprava od pracovního prostoru. Mechanismus po první optimalizaci se od původního výrazně neposunul.



Obr. 38: Optimalizované návrhy (suma) mechanismu – pokročilá kalibrace – chyba senzoru  $10^{-5}$

### 2.5.3.3 Změna chyb parametrů i čidel

Jak jsme mohli vidět v tabulce 16, rozdíly parametrů se při optimalizaci velice zlepšily. Toto zlepšení je nicméně tak velké, že je v případě simulace reálného případu krajně nepravděpodobné. Je možné, že se optimalizace „přizpůsobila“ námi zadaným chybám. Vyzkoušíme proto, jaké výsledky obdržíme, pokud zachováme kovarianční

matice a zkalibrujeme optimalizovaný mechanismus s jinými chybami parametrů i čidel (řádově stejnými). Tyto výsledky porovnáme s původním mechanismem, kde budou chyby čidel a parametrů stejné jako v případě optimalizovaného mechanismu. Jako optimalizovaný mechanismu jsem zvolil ten, u kterého se zlepšily rozdíly parametrů nejvíce, tedy po optimalizaci pomocí genetických algoritmů a fminsearch. V tabulce 17 můžeme vidět výsledky, které vyšly po zprůměrování pěti různých případů zachybování parametrů a čidel.

Parametr	Původní mechanismus		Optimalizovaný mechanismus	
	Základní kalibrace	Pokročilá kalibrace	Základní kalibrace	Pokročilá kalibrace
$\Delta x_A$ [ $\mu\text{m}$ ]	4.3795	6.7248	2.9282	1.8584
$\Delta y_A$ [ $\mu\text{m}$ ]	3.9013	2.7258	14.2653	1.6019
$\Delta\phi_{120}$ [ $\mu\text{rad}$ ]	2.1079	4.1531	3.9251	4.8726
$\Delta l_2$ [ $\mu\text{m}$ ]	3.9928	1.9323	16.8438	1.9291
$\Delta\phi_{230}$ [ $\mu\text{rad}$ ]	3.1449	4.1161	0.7754	3.0317
$\Delta l_3$ [ $\mu\text{m}$ ]	4.3903	2.1228	7.1849	1.4893
$\Delta\phi_{340}$ [ $\mu\text{rad}$ ]	5.0244	4.7013	0.4221	3.2500
$\Delta l_4$ [ $\mu\text{m}$ ]	4.4359	4.2319	8.7726	2.7272
Kalibrovatelnost	416.3	1477.7	1533.2	903.9
Průměrný rozdíl parametru	3.9221	3.8385	6.8896	2.5950

Tabulka 17: Porovnání kalibrace pro pět různých zachybování původního a optimalizovaného mechanismu

Jak můžeme vidět, zlepšení rozdílu parametrů není po změně zachybování již tak výrazné, jako v tabulce 15 a 16. V případě základní kalibrace se rozdíly parametrů dokonce u optimalizovaného mechanismu zhoršily. Zhoršení je výrazné, a to téměř na dvojnásobek oproti základní kalibraci původního mechanismu. U pokročilé kalibrace rozdíly parametrů po optimalizaci vyšly lépe pro optimalizovaný mechanismus, a to o téměř 34 %.

#### 2.5.3.4 Změna chyb čidel pro novou iteraci/generaci

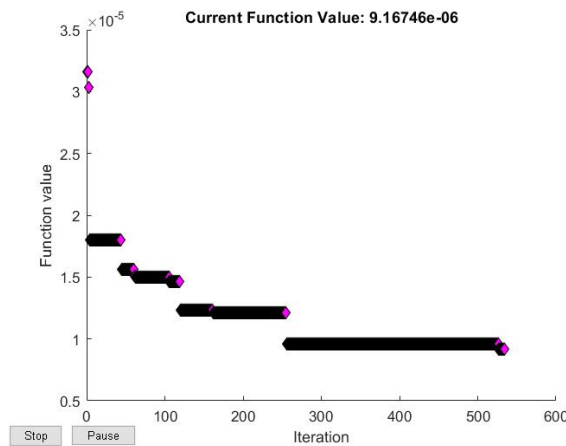
V posledním případě rovinného mechanismu si ukážeme, jaké výsledky obdržíme po optimalizaci, když budeme generovat pokaždé nové chyby čidel senzorů pro novou iteraci/generaci pomocí příkazu „randn“. Chyby nicméně budou nabývat stejného řádu jako v dřívějším případě, a to  $10^{-5}$ . Tímto budeme nejpřesněji simulovat skutečné měření.



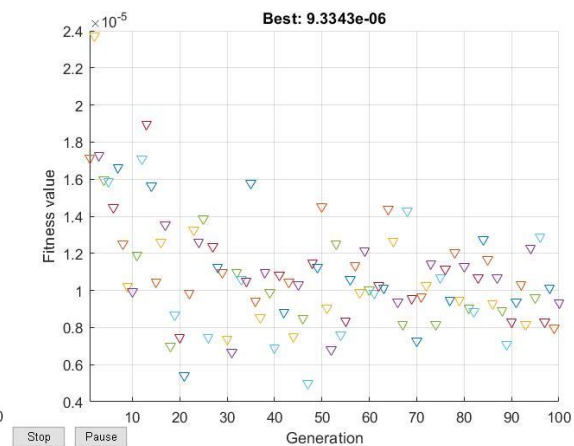
Chyby parametrů a kovarianční matice ponecháme pro celý výpočet stejné a budou mít stejné hodnoty jako v případě 2.5.3.3.

#### 2.5.3.4.1 Optimalizace základní metody kalibrace

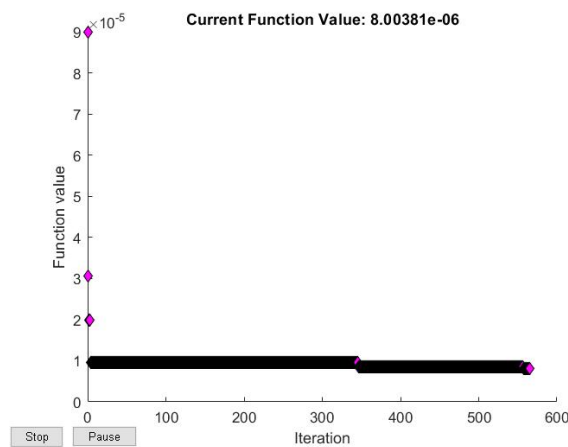
Jako první si ukážeme optimalizace základní kalibrace. Na následujících obrázcích můžeme vidět vývoj optimalizované funkce pro všechny tři metody optimalizace.



Obr. 40: Optimalizace sumy – základní kalibrace chyba senzorů  $10^{-5}$  – fminsearch



Obr. 39: Optimalizace sumy – základní kalibrace chyba senzorů  $10^{-5}$  – GA



Obr. 41: Optimalizace sumy – základní kalibrace – chyba senzorů  $10^{-5}$  – GA + fminsearch

Jak můžeme vidět, tentokrát grafy mají jiný tvar než v předchozích případech optimalizace. Nevidíme již postupné snižování cílové funkce, které připomínalo tvar pro optimalizaci pomocí fminsearch. Vidíme spíše skokové změny. Nejlepší hodnota, které jsme dosáhli touto optimalizací dosáhli, je  $9,1675 \times 10^{-6}$ . Při optimalizaci pomocí genetických algoritmů jsme získali hodnotu cílové funkce nižší, a to  $4,9771 \times 10^{-6}$ . Jak ale můžeme vidět, výsledná hodnota fitness funkce generace se v tomto případě častokrát i zvýší oproti dřívější. Pokud se ale podíváme na graf jako celek, vidíme, že se výsledné hodnoty

optimalizované funkce nacházejí v užším pásu. Pro následné použití *fminsearch* můžeme znovu pozorovat skokové změny až na hodnotu  $8,0038e-6$ .

Toto různé chování, které pozorujeme v grafech, je způsobené chybou čidel, která se pro každou iteraci nebo generaci mění. U *fminsearch* je skoková změna z toho důvodu, že pro dané chyby mechanismu většinou nebyly nalezeny lepší hodnoty cílové funkce a byla zapsána původní hodnota. U genetických algoritmů z důvodu jiné chyby můžeme nalézt i vyšší hodnotu nejlepšího jedince dané populace.

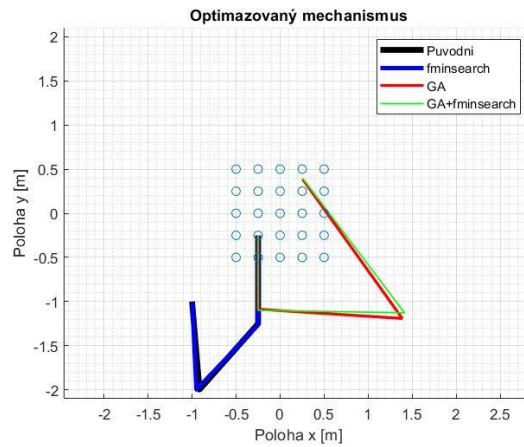
V tabulce 18 porovnáme parametry pro nejlepší hodnoty cílové funkce. Jak můžeme vidět optimalizací jsme rozdíly zkalibrovaných a skutečných parametrů výrazně zlepšili. Prvotní optimalizace je lepší o 68,5 % oproti samotné kalibraci. Zlepšení genetických algoritmů je nejvýraznější, a to o 82 % oproti kalibraci. Optimalizace kombinací genetických algoritmů a *fminsearch* získala srovnatelný výsledek jako samotné použití *fminsearch*.

<b>Parametr</b>	<b>Základní kalibrace</b>	<b>Fminsearch</b>	<b>Globální algoritmy</b>	<b>GA + fminsearch</b>
$\Delta x_A$ [ $\mu\text{m}$ ]	7.6815	0.8405	0.0368	0.6998
$\Delta y_A$ [ $\mu\text{m}$ ]	4.8917	1.6065	0.2432	0.8252
$\Delta\varphi_{120}$ [ $\mu\text{rad}$ ]	1.4395	1.4480	0.1236	1.5125
$\Delta l_2$ [ $\mu\text{m}$ ]	1.2767	2.0227	2.6467	0.2685
$\Delta\varphi_{230}$ [ $\mu\text{rad}$ ]	0.0979	0.7975	0.2367	0.0399
$\Delta l_3$ [ $\mu\text{m}$ ]	5.6949	0.6670	0.3507	2.7307
$\Delta\varphi_{340}$ [ $\mu\text{rad}$ ]	3.4310	1.2111	1.1761	0.7821
$\Delta l_4$ [ $\mu\text{m}$ ]	3.1917	0.1540	0.1632	1.1452
Kalibrovatelnost	416.3	427.0	262.6	253.1
Průměrný rozdíl parametru [ $\mu$ -]	3.4631	1.0934	0.6221	1.0005

Tabulka 18: Porovnání metod minimalizace sumy – základní kalibrace – chyba senzoru  $10^{-5}$

Grafy reziduí vazbových rovnic byly podobné jako v jiných případech optimalizace základní kalibrace, a proto si je nebudeme ukazovat. Velikost reziduí je v řádu  $10^{-5}$ . Ukážeme si ale, jak vypadají schémata mechanismu po optimalizacích. Vidíme to na následujícím obrázku. Mechanismus pro optimalizace pomocí *fminsearch* je téměř totožný s původním mechanismem. U zbylých metod optimalizace se bod A mechanismu posunul

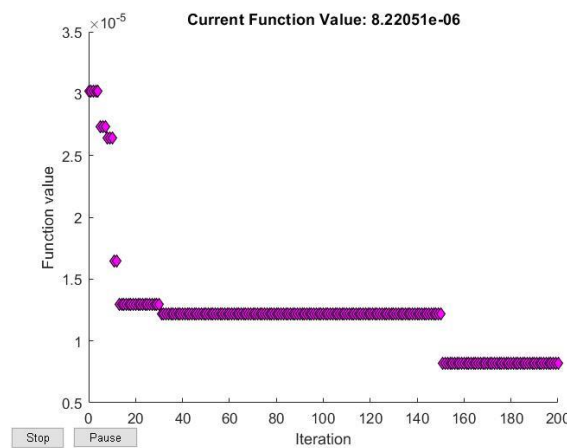
mírně nahoru a doprava od středu pracovního prostoru. To jsme viděli již dříve a není to pro nás překvapivé.



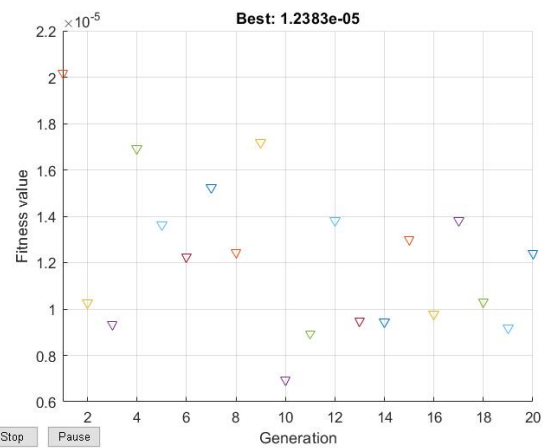
Obr. 42: Optimalizované (suma) návrhy mechanismu – základní kalibrace – chyba senzoru  $10^{-5}$

#### 2.5.3.4.2 Optimalizace pokročilé metody kalibrace

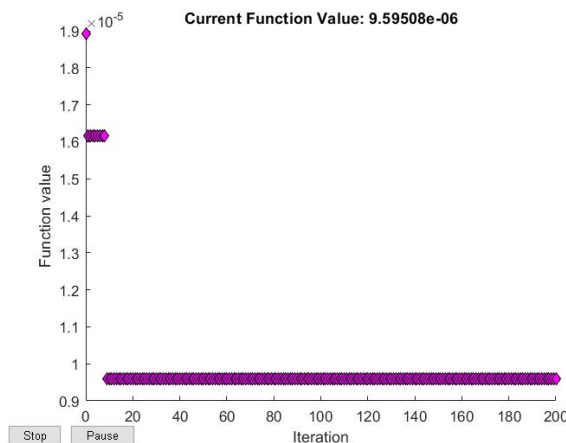
V této části si ukážeme výsledky optimalizace pokročilé metody kalibrace se změnou chyb senzorů v každé nové iteraci nebo generaci. Obdrželi jsme vizuálně podobné grafy jako v případě optimalizace základní metody kalibrace. Nejlepší hodnoty výsledné funkce jsme získali pro optimalizace pomocí genetických algoritmů, a to  $6,9293e-6$ . První metodou optimalizace jsme získali nejlepší hodnotu cílové funkce  $8,2205e-6$  a poslední metodou  $9,5951e-6$ . Podobně jako v předchozím případě jsou i v tomto případě výsledné hodnoty optimalizace pomocí fminsearch pro obě metody srovnatelné.



Obr. 44: Optimalizace sumy – pokročilá kalibrace chyba senzorů  $10^{-5}$  – fminsearch



Obr. 43: Optimalizace sumy – pokročilá kalibrace chyba senzorů  $10^{-5}$  – GA



Obr. 45: Optimalizace sumy – pokročilá kalibrace – chyba senzorů  $10^{-5}$  – GA + fminsearch

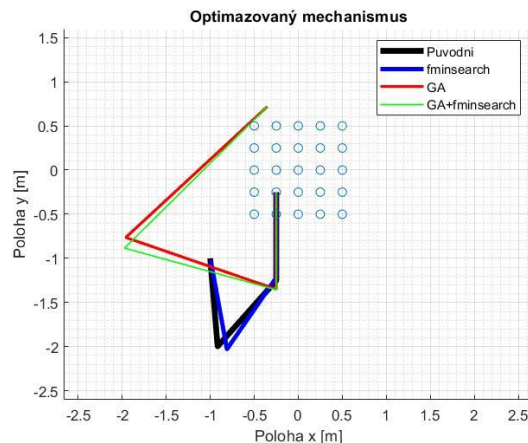
V tabulce můžeme vidět porovnání výsledků všech tří metod optimalizace a také samotné kalibrace. Nejlépe znovu dopadla optimalizace pomocí genetických algoritmů. Rozdíly parametrů se v tomto případě zlepšily v průměru o téměř 57 %. Pro první metodu jsme zaznamenali zlepšení o 49 % a pro třetí o 40 %. Tyto zjištění jsou v souladu s grafy, které jsme si již popsali výše. Kalibrovatelnost se pro všechny metody výrazně neliší.

Parametr	Pokročilá kalibrace	Fminsearch	Globální algoritmy	GA + fminsearch
$\Delta x_A$ [ $\mu\text{m}$ ]	2.609	0.5375	1.7050	1.9548
$\Delta y_A$ [ $\mu\text{m}$ ]	4.7754	1.0135	0.4201	1.2118
$\Delta\phi_{120}$ [ $\mu\text{rad}$ ]	1.7595	1.2736	0.3395	0.1946
$\Delta l_2$ [ $\mu\text{m}$ ]	0.9330	1.7278	0.0841	2.0021
$\Delta\phi_{230}$ [ $\mu\text{rad}$ ]	1.4660	0.9082	0.8332	0.6606
$\Delta l_3$ [ $\mu\text{m}$ ]	06763	0.5027	0.8476	2.6575
$\Delta\phi_{340}$ [ $\mu\text{rad}$ ]	1.9039	1.8741	1.1809	0.3814
$\Delta l_4$ [ $\mu\text{m}$ ]	1.8692	0.3830	1.5187	0.5324
Kalibrovatelnost	1447.5	1394.3	1479.7	1524.6
Průměrný rozdíl parametru [ $\mu$ -]	2.0030	1.0276	0.8662	1.1994

Tabulka 19: Porovnání metod minimalizace sumy – pokročilá kalibrace – chyba senzoru  $10^{-5}$

Grafy reziduí si ukazovat nebudeme, protože znovu nabývají hodnot jako v předešlých optimalizacích pro pokročilou metodu kalibrace. Jsou tedy v řádu  $10^{-16}$ . Ukážeme si, jak vypadají optimalizované mechanismy. Vidíme, že po první metodě optimalizace zůstal mechanismus téměř stejný jako mechanismus původní. Pro zbylé

metody optimalizace se mechanismus posunul a bod A se nachází nad pracovním prostorem.



Obr. 46: Optimalizované (CF – suma) návrhy mechanismu – pokročilá kalibrace – chyba senzoru  $10^{-5}$

## 2.6 Shrnutí výsledků

V této kapitole jsme si ukázali výsledky pro základní i pokročilou metodu kalibrace pro různě definované velikosti chyb čidel senzorů. Vypočítali jsme pro dané zachybování mechanismu kalibrovatelnost. Zjistili jsme, že pro základní kalibraci se kalibrovatelnost s jinou chybou senzorů nemění. To nebylo překvapení, protože v Jacobiho matici, která se používá pro výpočet kalibrovatelnosti pro základní metodu kalibrace, se chyby senzorů nevyskytují. Hodnotu pro kalibrovatelnost základní metody jsme zjistili 416,3.

Stejně zachybování čidel jsme použili pro pokročilou kalibraci. Nejprve jsme vyzkoušeli funkčnost skriptu se zachybováním senzorů v řádu  $10^{-16}$ , což je numerická nula v Matlabu. Hodnota kalibrovatelnosti vyšla 416,4, což je stejný výsledek jako v případě základní kalibrace. Vytvořený skript zafungoval podle předpokladů. Následně jsme provedli pokročilou kalibraci také pro zbylé případy zachybování. Pro zachybování senzorů v řádu  $10^{-5}$  vyšla kalibrovatelnost 1447,5. Pokud jsme zhoršili přesnost jednoho senzoru o jeden řád, kalibrovatelnost vyšla 3253,1. Při zhoršení přesnosti totožného senzoru o další řád pak byla kalibrovatelnost 137958. Z toho můžeme usuzovat, že se zhoršujícím se rozdílem chyb senzorů je hůře definovaná matice pro výpočet kalibrovatelnosti, a proto její podmíněnost vyjde vyšší. To se potvrdilo, když jsme určili chybu všech senzorů nulovou kromě jednoho, jehož zachybování bylo v řádu  $10^{-3}$ . V tomto případě kalibrovatelnost vyšla v řádu  $10^{+27}$ . Tímto jsme si ukázali výsledky kalibrace a je čas přejít k optimalizaci.

Vybrali jsme dva případy zachybování. Ty jsme optimalizovali pomocí *fminsearch*, genetických algoritmů a spojením obou předchozích metod. Vybrali jsme případ pro zachybování všech senzorů v řádu  $10^{-5}$  a pro zachybování jednoho senzoru v řádu  $10^{-3}$  a zbylých  $10^{-5}$ .

Pro první případ zachybování jsme zjistili nejmenší hodnotu kalibrovatelnosti u optimalizace základní kalibrace pomocí genetických algoritmů a *fminsearch*. Zlepšení bylo 88 % na 89,4. Nejlepšího průměrného rozdílu parametrů jsme nicméně dosáhli u genetických algoritmů a výsledky byly o téměř třetinu lepší než po optimalizaci pomocí GA+*fminsearch*. U všech optimalizací nicméně byla zjištěna nižší průměrná hodnota rozdílu parametrů než u kalibrace. U pokročilé kalibrace jsme také snížili hodnotu kalibrovatelnosti, a to o 54 %. Průměrná chyba rozdílu parametrů ale vzrostla na téměř čtyřnásobek.

Při optimalizaci druhého případu zachybování jsme obdrželi podobné výsledky. Nejlepší hodnotu kalibrovatelnosti u optimalizace základní metody jsme našli pro genetické algoritmy s *fminsearch*. Nejlepšího snížení průměrného rozdílu jsme dosáhli u optimalizace pomocí samotného *fminsearch*. U optimalizace pokročilé metody jsme překvapivě získali nejlepší hodnotu kalibrovatelnosti pro *fminsearch*. To mohlo být způsobené menším počtem generací u genetických algoritmů. Průměrný rozdíl parametrů ale i v tomto případě ve všech metodách vzrostl oproti samotné kalibraci. Z tohoto důvodu jsme vyzkoušeli změnit cílovou funkci na sumu rozdílů parametrů skutečných a zkalibrovaných. Výsledek pro tuto změnu je popsán v odstavci níže.

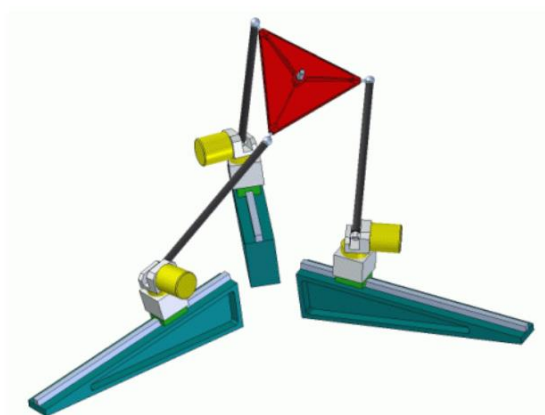
Nejvíce přiblížit se reálnému měření se můžeme v případě, kdy se budou chyby senzorů měnit s každou novou naměřenou hodnotou. Přepsali jsme proto skript a vyzkoušeli optimalizace se změněnou cílovou funkcí a měnicími se chybami senzorů pro každou iteraci optimalizace. Pro optimalizaci základní kalibrace jsme pro nejlepší metodu (GA) zjistili zlepšení průměrného rozdílu o 82 % a kalibrovatelnosti se snížila o 37 % na 262,6. U optimalizace pokročilé metody vyšel nejlepší průměrný rozdíl u genetických algoritmů. Snížení bylo o 57 %. Zároveň ale mírně vzrostla kalibrovatelnost, a to o 2 %.

### 3. Prostorový mechanismus

V této kapitole si ukážeme prostorový mechanismus, který budeme optimalizovat tak, jako jsme to vyzkoušeli na mechanismu rovinném. S tímto mechanismem pracoval již Hamrle [1]. My si vybereme jednu část tohoto mechanismu a pokusíme se jí optimalizovat. Nejprve si ale daný mechanismus popíšeme a ukážeme si postup kalibrace, který Hamrle použil. Nebudeme si tentokrát vypisovat všechny potřebné informace. To z toho důvodu, že už je tento mechanismus složitější než ten předchozí, data nabudou na objemu a nebylo by to příliš přehledné. Všechna použitá data nicméně budou uložena v příloze ve složce „3D\_mechanismus“.

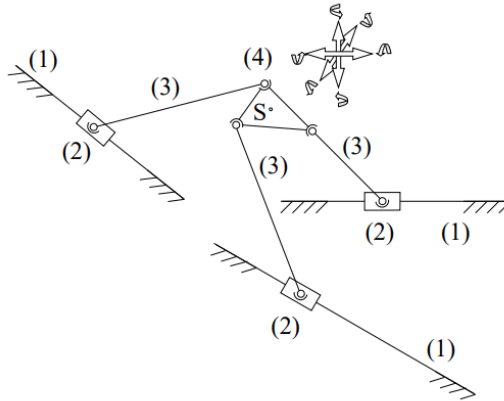
#### 3.1 Popis

Pro naši práci jsme vybrali mechanismus, který se nachází na následujícím obrázku. Tento mechanismus se nazývá RedCaM (Redundanz Calibration Machine) a má šest stupňů volnosti. Fungování stroje je založeno na paralelní kinematice, která využívá princip redundantního měření. To, jak jsme si již dříve naznačili, by mělo vést k lepším hodnotám kalibrovatelnosti, a tedy k přesnějším hodnotám end efektoru. Stroj má trojúhelníkovou platformu, jejíž polohu a natočení v prostoru se snažil Hamrle zjistit. S rámem je tato platforma spojena třemi nohami. Zjišťujeme celkem 9 hodnot při měření a jelikož má mechanismus 6 stupňů volnosti, je tento systém trojnásobně redundantní. Hamrle pro mechanismus ukázal tři návrhy řešení. V našem případě si vybereme návrh s posuvnými vazbami a z něho ještě „vyjmeme“ jedno rameno, které budeme optimalizovat.



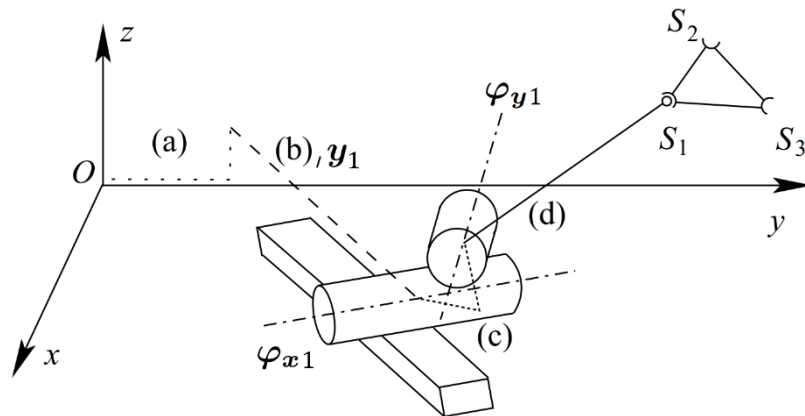
Obr. 47: RedCaM posuvný [5]

Základní kinematické schéma mechanismu je na následujícím obrázku. Jak jsme již zmínili, skládá se z trojúhelníkové platformy, tří nohou a tří posuvných lineárních vedení. Nohy jsou dvěma rotačními vazbami (na obrázku jako sférické) spojené s vozíky, které jsou na lineárním vedení, a na opačném konci nohy je připojena platforma pro měření. Měřenými parametry jsou v tomto případě poloha vozíku na vedení posuvné vazby a úhly rotace mezi nohou a vozíkem. Tímto způsobem získáme devět měřených veličin.



Obr. 48: Kinematické schéma měřícího stroje RedCaM s posuvnými vazbami [5]

Pro náš výpočet použijeme pouze jednu nohu tohoto mechanismu. Kinematické schéma jedné nohy můžeme vidět na obrázku níže.



Obr. 49: Kinematické schéma jedné nohy pro RedCaM s posuvnou vazbou [5]

Transformační matice mezi souřadnicovými systémy pro nohu jsou ve tvaru:

$$\mathbf{T}_{Oa} = \mathbf{T}_{\varphi_z}(\varphi_{zp})\mathbf{T}_{\varphi_y}(\varphi_{yp})\mathbf{T}_x(x)\mathbf{T}_{\varphi_y}(-\varphi_{yp})\mathbf{T}_{\varphi_z}(-\varphi_{zp}) \quad (3.1.1)$$

$$\mathbf{T}_{ab} = \mathbf{T}_x(x_d)\mathbf{T}_y(y_d)\mathbf{T}_{\varphi_x}(\varphi_{xd})\mathbf{T}_{\varphi_y}(\varphi_{yd})\mathbf{T}_{\varphi_z}(\varphi_{zd}) \quad (3.1.2)$$

$$\mathbf{T}_{bc} = \mathbf{T}_y(y_t)\mathbf{T}_z(z_t)\mathbf{T}_{\varphi_y}(\varphi_{yt})\mathbf{T}_{\varphi_z}(\varphi_{zt})\mathbf{T}_{\varphi_x}(\varphi_{xt}) \quad (3.1.3)$$

$$\mathbf{T}_{cd} = \mathbf{T}_x(x_c)\mathbf{T}_y(y_c)\mathbf{T}_z(z_c) \quad (3.1.4)$$



$$\mathbf{r}_{dS_1} = [0, 0, 0, 1], \quad (3.1.5)$$

$$\mathbf{T}_{Oa} = \mathbf{T}_y(y_p)\mathbf{T}_z(z_p)\mathbf{T}_{\varphi_y}(\varphi_{yp})\mathbf{T}_{\varphi_z}(\varphi_{zp})$$

kde prvky v závorkách jsou délky posunutí podél dané osy, respektive úhel natočení okolo dané osy.

Popis celé transformace, tedy popis pohybu z prvního souřadnicového systému do koncového bodu, je popsán následujícím způsobem:

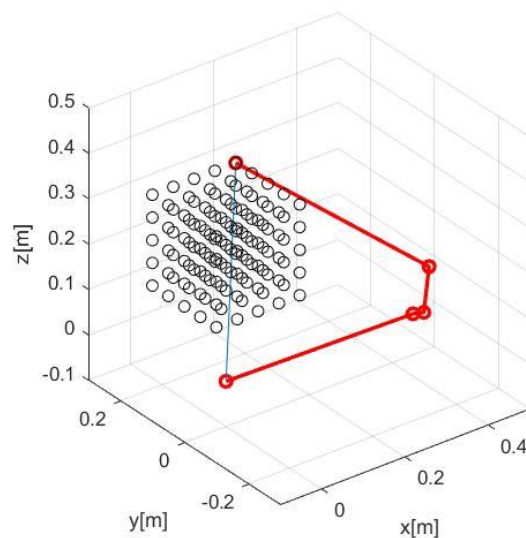
$$\mathbf{r}_{OS_1} = \mathbf{T}_{Oa} \mathbf{T}_{ab} \mathbf{T}_{bc} \mathbf{T}_{cd} \mathbf{r}_{dS_1} \quad (3.1.6)$$

Pro generování dat využijeme inverzní kinematiku. Definujeme si pracovní prostor, který bude mít tvar kvádru. V něm vytvoříme grid o 5x5x5 bodech. Celkem tedy budeme mít 125 poloh mechanismu, které budeme používat ke kalibraci a následné optimalizaci. Všechny tyto polohy musí splňovat rovnice (3.1.1) - (3.1.6). Pracovní prostor a počáteční mechanismus můžeme vidět na dalším obrázku.

Parametrů, které budeme kalibrovat pro část posuvného mechanismu, je celkem 13, a jsou následující:

$$P = [\varphi_{zp}, \varphi_{yp}, x_d, y_d, \varphi_{xd}, \varphi_{yd}, y_t, y_z, \varphi_{yt}, \varphi_{zt}, x_c, y_c, z_c]$$

Měřené souřadnice pro nohu jsou  $S = [x, \varphi_x, \varphi_x]$



**Obr. 50: Počáteční mechanismus**

Pro optimalizaci prostorového mechanismu si nebudeme ukazovat výsledky pro všechny metody a zachybování čidel jako v případě rovinného mechanismu. Ukážeme si samozřejmě výsledky ze základní i pokročilé kalibrace a následně budeme mechanismus optimalizovat metodou fminsearch a genetickými algoritmy, kdy nejprve cílová funkce bude kalibrovatelnost a následně suma rozdílů parametrů skutečných a zkalibrovaných ve tvaru

$$\mathbf{F} = \Delta \varphi_{zp} + \Delta \varphi_{yp} + \Delta x_d + \Delta y_d + \Delta \varphi_{xd} + \Delta \varphi_{yd} + \quad (3.1.7) \\ + \Delta y_t + \Delta y_z + \Delta \varphi_{yt} + \Delta \varphi_{zt} + \Delta x_c + \Delta y_c + \Delta z_c$$

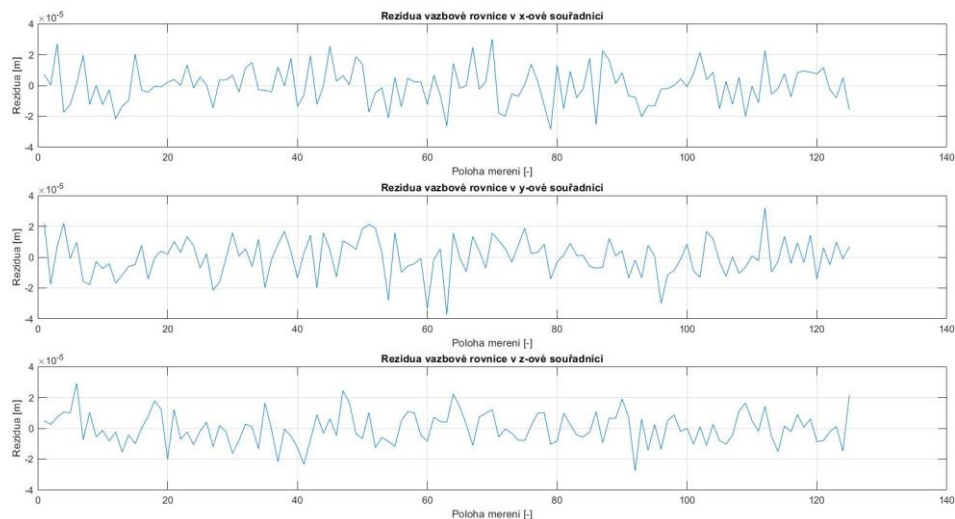
Parametry zachybujeme stejně jako v rovinném mechanismu, a to pomocí „randn“. Velikost zachybování bude pro všechny parametry v řádu  $10^{-3}$ . Pro všechny výsledky si uložíme jedna data zachybování parametrů. Čidla zachybujeme také pomocí „randn“, a to o velikosti  $10^{-5}$  pro všechny senzory. Zachybování čidel i parametrů bude pro všechny ukázané výsledky totožné.

Kovarianční matice parametrů určíme znovu jako rozdíl skutečných hodnot parametru a výkresových. Kovarianční matici čidel definujeme pomocí příkazu „cov“, který nám vypočítá kovarianci zachybování daného čidla. Obě matice budou znovu diagonální.

## 3.2 Výsledky základní metody kalibrace a optimalizace

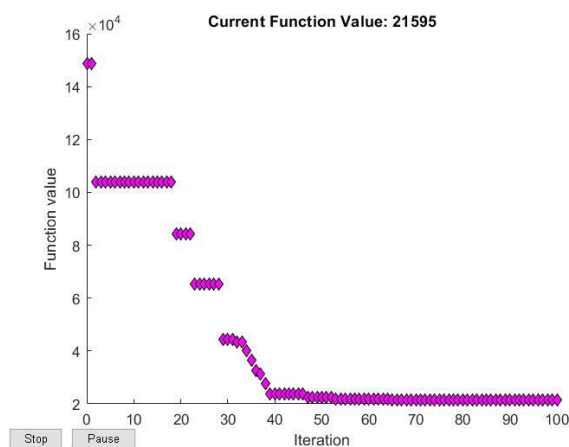
V této části si ukážeme výsledky, které jsme získali ze základní kalibrace, optimalizace metodou fminsearch a genetickými algoritmy, kdy optimalizovaná funkce je kalibrovatelnost, a optimalizace, kdy cílovou funkcí je suma rozdílů parametrů definovaná rovnicí (3.1.7).

Nejprve si ukážeme splnění vazbové rovnice. Jak můžeme vidět na následujícím obrázku, rezidua vazbové rovnice ve všech třech souřadnicích (x, y, z) jsou pro základní kalibraci v řádu  $10^{-5}$ . Tento výsledek je v řádu námi zadaných chyb senzorů a jak již víme z předchozí kapitoly, tento výsledek není překvapující. Graf pro optimalizaci splnění vazbových rovnic ukazovat nebudu, jelikož je téměř totožný.

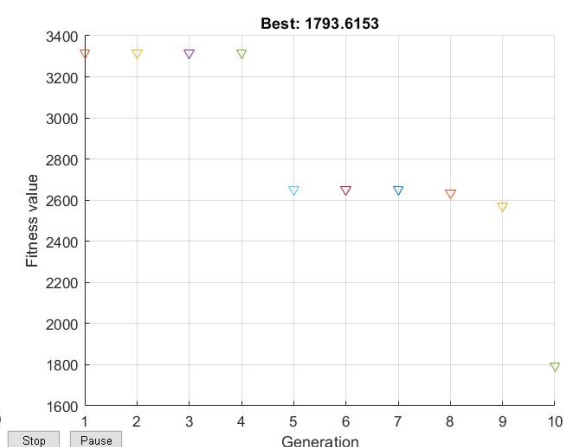


**Obr. 51: Rezidua vazbové rovnice prostorového mechanismu– základní kalibrace**

Vývoj cílové funkce můžeme vidět na následujících grafech. V horní části můžeme vidět optimalizace cílové funkce zadané jako kalibrovatelnost. Pod nimi se nachází grafy vývoje pro cílovou funkci zadanou jako suma rozdílů parametrů. V obou případech je levý obrázek pro optimalizaci pomocí *fminsearch* a napravo pro genetické algoritmy.

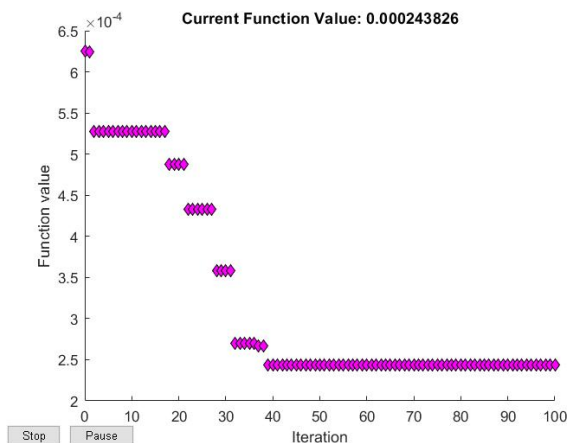


**Obr. 53: Vývoj CF (kalibrovatelnost) - základní kalibrace – *fminsearch***

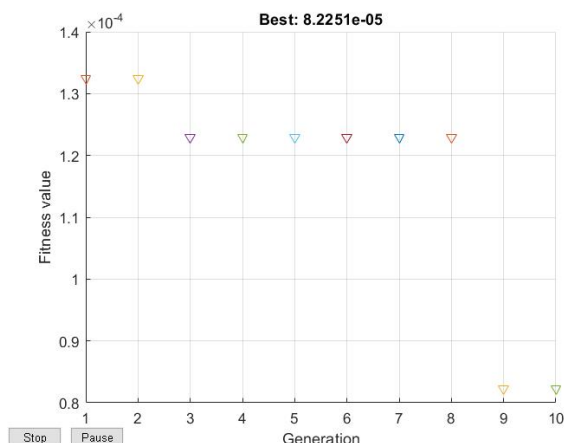


**Obr. 52: Vývoj CF (kalibrovatelnost) - základní kalibrace – GA**

Hodnota kalibrovatelnosti, kterou jsme vypočítali po samotné základní kalibraci je 148870,1 a jak můžeme vidět, je to počáteční hodnota optimalizace pomocí *fminsearch*. Po provedení této optimalizace jsme získali hodnotu kalibrovatelnosti 21594,9, což je snížení o 85 %. Po provedení optimalizace genetickými algoritmy jsme obdrželi hodnotu 1793,6. To je oproti samotné kalibraci snížení o více než 98 %.



Obr. 55: Vývoj CF (suma) - základní kalibrace – fminsearch



Obr. 54: Vývoj CF (suma) - základní kalibrace –GA

Optimalizace cílové funkce jako suma rozdílů parametrů proběhla také v pořádku. Po první optimalizaci jsme snížili cílovou funkci o 61 % a kalibrovatelnost jsme vypočítali 21706,2. To je podobný výsledek jako v případě optimalizace kalibrovatelnosti. Pro optimalizaci genetickými algoritmy jsme získali pro nejlepší výsledek hodnotu kalibrovatelnosti 113740,4 a snížení cílové funkce je o 86 %.

V tabulce níže můžeme vidět porovnání rozdílu parametrů pro základní kalibraci a její optimalizace. Pro jednoduchost ukazujeme pouze vybrané parametry. Po samotné kalibraci jsme obdrželi průměrnou chybu rozdílu parametru o velikosti 48,0898 mikronu. Pro všechny optimalizace jsme získali přesnější hodnoty.

Parametr	Základní kalibrace	CF – Kalibrovatelnost		CF – Suma rozdílů parametrů	
		Fminsearch	GA	Fminsearch	GA
$\Delta \varphi_{zp}$ [μrad]	13.1258	8.1556	1.8891	12.8728	2.0337
$\Delta x_d$ [μm]	25.4376	5.4559	11.5546	7.9437	4.4696
$\Delta \varphi_{yd}$ [μrad]	52.9293	6.6720	10.0504	18.1268	2.6991
$\Delta z_t$ [μm]	2.1300	0.8509	9.6910	7.5056	2.3822
$\Delta \varphi_{yt}$ [μrad]	101.5097	70.7704	13.2001	2.1779	32.7642
$\Delta x_c$ [μm]	123.9539	33.9644	16.3978	7.2328	2.6910
$\Delta y_c$ [μm]	7.8102	2.1217	6.3617	0.2783	1.9735
Kalibrovatelnost	148870.1	21594.9	1793.6	21706.2	113740.4
Průměrný rozdíl parametru [μ-]	48.0898	18.7818	7.2426	18.7559	6.3270

Tabulka 20: Porovnání výsledků základní kalibrace a optimalizace pro prostorový mechanismus

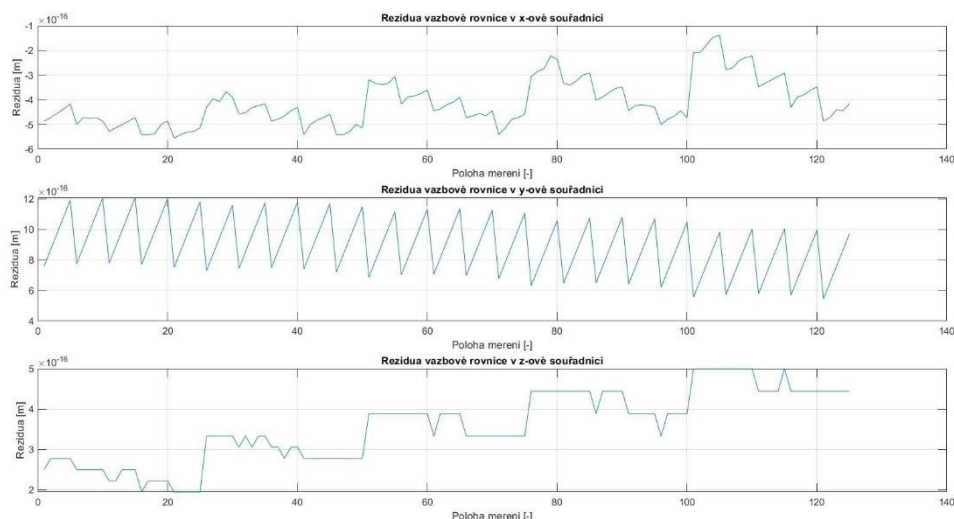
Po optimalizaci cílové funkce jako je kalibrovatelnost, jsme obdrželi pro metodu *fminsearch* průměrnou chybu parametru 18,7818 mikronu a pro genetické algoritmy 7,2426 mikronu. V procentuálním vyjádření to je zlepšení oproti samotné kalibraci o téměř 61 % pro *fminsearch* a o 84 % při použití genetických algoritmů.

Při optimalizaci cílové funkce jako sumy rozdílů jsme získali průměrnou chybu parametru 18,7559 mikronu pro *fminsearch*. To je velice podobná hodnota jako u optimalizace kalibrovatelnosti pomocí *fminsearch*. Odůvodňují si to tím, že lokální řešení pro obě metody má podobně definované hodnoty parametrů. Po porovnání parametrů obou metod jsme zjistili, že obě metody zoptimalizovaly parametry na téměř totožné hodnoty. Genetickými algoritmy jsme zlepšily průměrný rozdíl parametru na 6,3270 mikronu. To představuje zlepšení o téměř 87 % oproti samotné kalibraci.

### 3.3 Výsledky pokročilé metody kalibrace a optimalizace

V této části si ukážeme výsledky pro kalibraci a následnou optimalizaci za použití pokročilé kalibrace pro obě metody optimalizace.

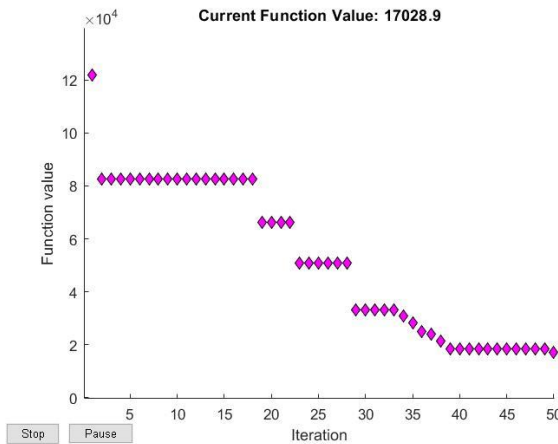
Jako první můžeme vidět splnění vazbových rovnic po samotné kalibraci. Vidíme, že dochází ke kompenzaci chyb čidel a rezidua rovnic jsou pro všechny tři souřadnice v řádu  $10^{-16}$ . Podobné hodnoty získáme i při optimalizacích.



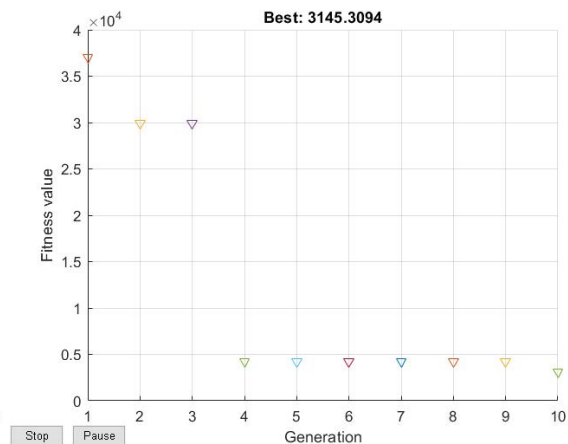
Obr. 56: Rezidua vazbové rovnice prostorového mechanismu– pokročilá kalibrace

Na následujících grafech můžeme vidět vývoj cílové funkce. Podobně jako v předchozí kapitole, i teď jsou horní dva obrázky pro optimalizaci cílové funkce zadané

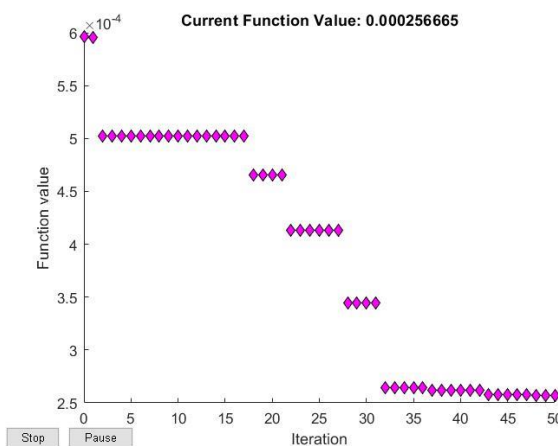
jako kalibrovatelnost, kdy na levé straně je metoda *fminsearch* a na pravé genetické algoritmy. Obrázky níže jsou pro optimalizaci sumy rozdílů parametrů.



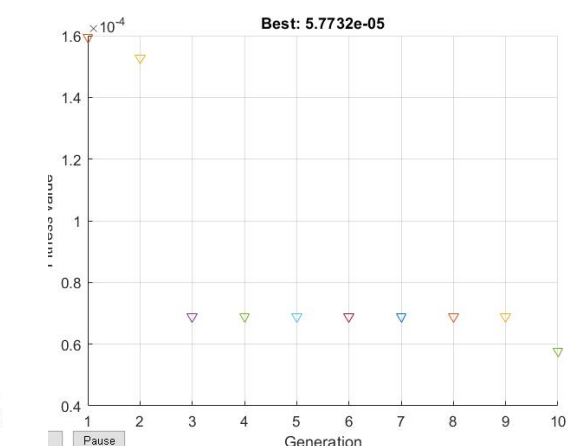
Obr. 58: Vývoj CF (kalibrovatelnost) - pokročilá kalibrace – fminsearch



Obr. 57: Vývoj CF (kalibrovatelnost) – pokročilá kalibrace – GA



Obr. 60: Vývoj CF (suma) - pokročilá kalibrace – fminsearch



Obr. 59: Vývoj CF (suma) - pokročilá kalibrace – GA

Jak můžeme vidět, cílové funkce všech metod jsme úspěšně optimalizovali. U optimalizace pomocí genetických algoritmů je možné, že metoda ještě nezkonvergovala a při větším počtu generací bychom zjistili ještě lepší hodnoty cílových funkcí.

Výsledky, které jsme získali pro optimalizaci pokročilé kalibrace pomocí *fminsearch*, jsou pro obě cílové funkce podobné. Kalibrovatelnost pro první cílovou funkci vyšla 17029,4, což je zlepšení o 86 %. Pro cílovou funkci jako sumy jsme získali hodnotu kalibrovatelnosti 18709,8. To je zlepšení o 85 %. Genetickými algoritmy jsme pro optimalizace kalibrovatelnosti zjistili hodnotu 3145,3, což je zlepšení o 97 %. Při

optimalizaci sumy rozdílů parametrů jsme nezaznamenali tak výrazné snížení kalibrovatelnosti, nicméně snížení o 61 % na hodnotu 47496,5 je přesto vysoké.

V tabulce 21 vidíme porovnání výsledků, který jsme získali. Ukážeme si znovu pouze některé parametry. Samotnou kalibraci jsme dostali průměrnou odchylku skutečných a zkalibrovaných parametrů 45,4714 mikronu. To je o 4,7 % lepší hodnota, než jsme pro stejné zachybování čidel i parametrů obdrželi ze základní metody kalibrace. Zjistili jsme tím, že pro kalibraci tohoto mechanismu získáme přesnější hodnotu z pokročilé kalibrace. Dále můžeme vidět, že hodnoty pro optimalizaci pomocí *fminsearch* jsou pro obě cílové funkce podobné. Průměrný rozdíl parametru jsme vylepšili o téměř 60 % pro cílovou funkci jako kalibrovatelnost a pro cílovou funkci sumy o 57 %. Podobné výsledky jsou dle mého názoru znovu způsobeny tím, že metoda zkonvergovala k podobným hodnotám parametrů. Jak můžeme vidět, tak jsme tím získali pouze lokální řešení.

Pro optimalizace pomocí genetických algoritmů jsme výsledky získali už rozdílné. Pro minimalizaci kalibrovatelnosti jsme vypočítali hodnotu průměrné chyby rozdílu na 5,2274 mikronu. Zlepšení oproti kalibraci je o necelých 89 %. Ještě nižší hodnotu průměrné rozdíly jsme vypočítali pro druhou cílovou funkci, a to 4,4410 mikronu, což je zlepšení o 90 %.

Parametr	Pokročilá kalibrace	FF-kalibrovatelnost		FF-suma rozdílů parametrů	
		Fminsearch	GA	Fminsearch	GA
$\Delta \varphi_{zp}$ [μrad]	5.3382	0.7609	0.4066	4.0067	0.9450
$\Delta x_d$ [μm]	25.0017	6.3517	1.7474	8.2894	6.0692
$\Delta \varphi_{yd}$ [μrad]	52.7975	10.2259	0.3571	15.2844	2.3998
$\Delta y_z$ [μm]	2.1283	1.8236	7.7622	7.7110	10.1967
$\Delta \varphi_{yt}$ [μrad]	74.4115	54.1796	4.6021	53.9443	3.1243
$\Delta x_c$ [μm]	125.9718	37.1080	9.0640	38.3497	1.3917
$\Delta y_c$ [μm]	8.1935	1.7759	2.9259	2.2432	5.1519
Kalibrovatelnost	121814.0	17029.4	3145.3	18709,8	47496.5
Průměrný rozdíl parametru [μ-]	45.8714	18.5422	5.2274	19.7435	4.4410

Tabulka 21: Porovnání výsledků pokročilé kalibrace a její optimalizace

### 3.4 Shrnutí

V této kapitole jsme si popsali zadaný prostorový mechanismus, který jsme následně zkalibrovali. Zjistili jsme, že je kalibrovatelnost vypočítaná pomocí pokročilé kalibrace lepší oproti základní o 18 %. Číselně vyšla 148870,1 pro základní kalibraci a 121814 pro pokročilou kalibraci. Průměrný rozdíl parametrů skutečných a zkalibrovaných byl také lepší pro pokročilou kalibraci, a to o 4,7 %.

Optimalizací základní kalibrace pro obě cílové funkce jsme vypočítali velice podobné hodnoty. Je to pravděpodobně způsobené podobným lokálním řešením. Pro optimalizaci kalibrovatelnosti *fminsearch* vyšla hodnota cílové funkce 21594,9 a pro optimalizaci sumy 21706,2. To je zlepšení oproti samotné kalibraci o přibližně 85 %. Velikosti průměrného rozdílu jsou také velice podobné, a to 18,7818 respektive 18,7559 mikronu. Oproti základní kalibraci je vylepšení hodnot o přibližně 61 %.

Dále jsme optimalizovali mechanismus genetickými algoritmy. Výsledné hodnoty kalibrovatelnosti nabývají hodnoty 1793,6, kdy byla cílová funkce kalibrovatelnost, a 113740,4, kdy byla cílová funkce suma rozdílů parametrů. Zlepšení to je o 98 % v prvním případě a téměř 24 % v druhém případě. To je veliký rozdíl. Očekával bych, že se tento rozdíl projeví na průměrném rozdílu parametru. Pro optimalizaci sumy nicméně vyšla hodnota 6,3270. To je menší hodnota, než jsme získali při optimalizaci kalibrovatelnosti (7,2426). V obou případech je to nicméně významné zlepšení. V procentech je to zlepšení o 87 % pro nižší hodnotu průměrného rozdílu parametru a 84 % pro vyšší.

Po optimalizaci pokročilé kalibrace *fminsearch* jsme pro obě cílové funkce znovu získali podobné výsledky. Kalibrovatelnost se v prvním případě cílové funkce zlepšila na hodnotu 17028,9. To je zlepšení o 86 %. V případě druhé cílové funkce jsme zjistili kalibrovatelnost 18709,8. To je zlepšení o 85 %. Průměrné rozdíly parametrů se v obou případech také zlepšily, a to o 60 % respektive 57 %. Pro genetické algoritmy s cílovou funkcí kalibrovatelnost jsme získali hodnotu kalibrovatelnosti 3145,3. To představuje zlepšení o 97 %. Pro genetické algoritmy s cílovou funkcí sumy byla kalibrovatelnost 47496,5. To je oproti předchozí metodě výrazně horší výsledek. Oproti samotné kalibraci je to nicméně zlepšení o 57 %. Průměrné rozdíly se pro obě cílové funkce také zlepšily a to o 89 % na 5,2274 mikronu pro první cílovou funkci a pro druhou cílovou funkci o 90 % na 4,410 mikronu.



## Závěr

Cílem této diplomové práce bylo seznámit se s problémem kalibrace mechanismů a dvěma metodami, kterými můžeme kalibraci uskutečnit. První metodou byla kalibrace mnou označena jako základní, kterou jsem si popsal v kapitole 1.1. Druhou metodou byla mnou označená pokročilá kalibrace, která je popsána v kapitole 1.2. Dále jsme si zavedli pojem kalibrovatelnost a popsal metody *fminsearch* a genetické algoritmy, které jsme použili při optimalizaci.

V druhé kapitole jsme se zaměřili na rovinný mechanismus, kterým bylo rameno robota. Definovali jsme všechny hodnoty, které jsme potřebovali pro následující výpočty. Poté jsme si představili související výsledky výpočtů. Nejprve jsme ukázali výsledky pro základní i pokročilou metodu kalibrace s různě definovanými velikostmi chyb čidel sensorů. Vypočítali jsme pro dané zachybování mechanismu kalibrovatelnost. Zjistili jsme, že pro základní kalibraci se kalibrovatelnost s jinou chybou sensorů nemění. To nebylo překvapení, protože v Jacobiho matici, která se používá pro výpočet kalibrovatelnosti pro základní metodu kalibrace, se chyby sensorů nevyskytují. U pokročilé kalibrace se kalibrovatelnost již pro různé zachybování mění. S rostoucím rozdílem řádu zachybování čidel se horšila i podmíněnost matice, kterou jsme využili pro výpočet kalibrovatelnosti, a proto rostla kalibrovatelnost. Pokud jsme porovnali zlepšení rozdílu parametrů skutečných a zkalibrovaných, pro pokročilou kalibraci ve většině případů vycházely lepší výsledky.

Další částí druhé kapitoly jsou výsledky optimalizace, pro níž jsme vybrali dva případy zachybování chyb sensorů. První zachybování bylo v řádu  $10^{-5}$  pro všechny senzory. V druhém případě byly senzory zachybované také v řádu  $10^{-5}$  až na jeden, jehož přesnost jsme zhoršili o dva řády. Pro oba případy zachybování jsme v nejlepší metodě optimalizace základní kalibrace snížili kalibrovatelnost o přibližně 88 % (GA+*fminsearch*). V obou případech jsme také snížili i průměrný rozdíl skutečných a zkalibrovaných parametrů, a to o 24 %, respektive 34 %. Optimalizace pokročilé metody kalibrace proběhla úspěšně. Kalibrovatelnost jsme v obou případech dle předpokladů minimalizovali, a to o přibližně 54 % pro první zachybování (GA+*fminsearch*) a 72 % pro druhé (*fminsearch*). Co nebylo podle předpokladů, byly výsledky průměrných rozdílů parametrů, které oproti kalibraci a předpokladu vzrostly. V prvním případě vzrostla hodnota pro průměrný rozdíl parametru pro nejlepší metodu optimalizace na téměř

čtyřnásobek. V druhém případě zachybování byl nárůst menší a to o 35 % oproti samotné kalibraci. Proč jsme obdrželi takovéto výsledky a průměrné rozdíly parametrů se násobně zvýšily, si nedokážu vysvětlit. Bude proto potřeba další výzkum tohoto problému, zda-li jsou výsledky validní nebo jsme udělali někde chybu.

Kvůli těmto neočekávaným výsledkům jsme se rozhodli změnit cílovou funkci na sumu rozdílů parametrů skutečných a zkalibrovaných. Zároveň jsme se chtěli přiblížit více měření ve skutečnosti. Proto jsme chyby senzorů měnili s každou novou naměřenou hodnotou. Pro optimalizaci základní kalibrace jsme pro nejlepší metodu (GA) zjistili zlepšení průměrného rozdílu parametru o 82 % a kalibrovatelnosti o 37 %. U optimalizace pokročilé metody jsme našli nejlepší průměrný rozdíl znovu u genetických algoritmů. Zlepšení bylo v tomto případě 57 %. Zároveň ale mírně vzrostla kalibrovatelnost, a to o 2 %. Tímto jsme splnili zadané body pro tuto práci pro rovinný mechanismus.

Porovnání metod optimalizace mezi sebou a určení, která metoda je nejlepší, je nicméně komplikované. Získali jsme často velice rozdílné výsledky pro různé metody a jednoznačný závěr se nám nepovedlo stanovit. Dle mého názoru jsme nejlepších hodnot nejčastěji dosáhli optimalizací pomocí genetických algoritmů. Pokud bychom tedy měli zvolit pouze jednu metodu optimalizace, zvolil bych pravděpodobně tuto. Její nevýhodou je nicméně výpočetní čas, který je násobně delší oproti *fminsearch*.

Ve třetí kapitole jsme si ukázali prostorový mechanismus. Ten jsme si popsali a definovali jsme vše, co bylo potřeba pro kalibraci a následnou optimalizaci. Pro optimalizaci jsme zvolili dvě metody, a to *fminsearch* a genetické algoritmy. Cílovou funkci jsme nejprve definovali jako kalibrovatelnost a následně jako součet rozdílů skutečných a zkalibrovaných parametrů. Pro kalibraci jsme zjistili lepší hodnotu kalibrovatelnosti pokročilé kalibrace o 18 % oproti základní. Průměrný rozdíl parametrů byl také lepší pro pokročilou kalibraci, a to o 4,7 %.

Optimalizace základní kalibrace obou cílových funkcí zjistila velice podobné hodnoty pro *fminsearch*. Zlepšení oproti samotné kalibraci bylo přibližně 85 %. Průměrné chyby se také zlepšily, a to přibližně o 61 %. Dále jsme optimalizovali genetickými algoritmy. Kalibrovatelnost jsme při této metodě optimalizace zlepšili o 98 %, kdy byla cílová funkce kalibrovatelnost, a o 24 %, kdy byla cílová funkce suma rozdílů parametrů. Očekával bych,

že se tento rozdíl projeví na průměrném rozdílu parametru. Zlepšení průměrného rozdílu parametru je v první případě 84 % a ve druhém 87 %. Po optimalizaci *fminsearch* jsme pro obě cílové funkce znovu získali podobné výsledky. Kalibrovatelnost se v obou případech cílové funkce zlepšila přibližně o 50 %. Průměrné rozdíly parametrů se zlepšily přibližně o 60 %. Genetickými algoritmy s cílovou funkcí kalibrovatelnosti jsme dosáhli zlepšení o 97 %. Pro genetické algoritmy s cílovou funkcí sumy bylo zlepšení kalibrovatelnosti menší, a to 57 %. Průměrné rozdíly se pro obě cílové funkce také zlepšily přibližně o 90 %. Tím jsme splnili zadané body pro prostorový mechanismus.

Z mého pohledu byly splněny všechny body, které jsme měly pro tuto diplomovou práci zadané. Pro určení, jestli má být kalibrovatelnost uvažovaná jako parametr při počátečním návrhu mechanismu, jsme nenalezli jednoznačnou odpověď. V některých případech jsme po optimalizaci zlepšili požadované vlastnosti mechanismu, ovšem v některých nikoli. Výsledky velice záležely na tom, jak jsme daný příklad definovali. Z tohoto důvodu se neodvážuji dělat konečné závěry. Pro jednoznačnou odpověď proto problematika kalibrovatelnosti vyžaduje další výzkum. Skripty v Matlabu, které jsme vytvořili, jsou uloženy v příloze. Z důvodu velikosti jsem byl nucen odstranit naměřená data pro optimalizace. V případě jakékoli navazující práce je proto potřeba mechanismus optimalizovat a tato data vytvořit.

# Bibliografie

- [1] HAMRLE, Vojtěch. Kalibrovatelnost a její použití pro návrh paralelních kinematických struktur. Jugoslávských partyzánů 1580/3 160 00 Praha 6 - Dejvice Česká republika, 2009. Disertační práce. ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE Fakulta strojní.“. *Vedoucí práce Prof. Ing. Šika Zbyněk Ph. D.*
- [2] G. Stengele, Cross Hueller Specht Xperimental, a machining center with new hybrid kinematics, in: R. Neugebauer (Ed.), Development methods and application experience of parallel kinematics, IWU FhG, Chemnitz, 2002, pp. 609–627
- [3] J. M. Hollerbach, C. W. Wampler a . A. Tatsuo, „An Implicit Loop Method for Kinematic Calibration and Its Application to Closed-Chain Mechanisms,“ IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, pp. 710-724, October 1995
- [4] D. Daney, I.Z. Emiris, Kinematic calibration of the Gough platform, Proceedings of the 2001 IEEE International Conference on Robotics & Automation, Seoul
- [5] ŠIKA, Zbyněk, Vojtěch HAMRLE, Michael VALÁŠEK a Petr BENEŠ. Calibrability as additional design criterion of parallel kinematic machines. Mechanism and Machine Theory [online]. 2012, 2011(50), 16 [cit. 2022-08-25]. Dostupné z: <https://www.sciencedirect.com>
- [6] Tomáš Skopec, Zbyněk Šika a Michael Valášek. Calibration using adaptive model complexity for parallel and fiber-driven mechanisms. Robotica [online]. July 10, 2015, 10 July 2015, 2015, 1416–1435 [cit. 2022-08-25]. Dostupné z: doi:10.1017/S02635747150004
- [7] Cond: Condition number of matrix. The MathWorks, Inc. [online]. 2022 [cit. 2022-08-25]. Dostupné z: <https://www.mathworks.com/help/symbolic/cond.html>“.
- [8] Fminsearch Algorithm. The MathWorks, Inc. [online]. Natick, Massachusetts, USA, 2022 [cit. 2022-08-25]. Dostupné z: <https://www.mathworks.com/help/optim/ug/fminsearch-algorithm.html>
- [9] JEFFREY C. LAGARIAS, JAMES A. REEDS, MARGARET H. WRIGHT, AND PAUL E. WRIGHT. CONVERGENCE PROPERTIES OF THE NELDER–MEAD SIMPLEX METHOD IN LOW DIMENSIONS. SIAM Journal on Optimization. 1998, December 1998(Vol. 9), 7. Dostupné z: doi:.S1052623496303470.
- [10] Find global minima for highly nonlinear problems. The MathWorks, Inc. [online]. Natick, Massachusetts, USA, 2022 [cit. 2022-08-25]. Dostupné z: <https://www.mathworks.com/discovery/genetic-algorithm.html>
- [11] What Is the Genetic Algorithm?. The MathWorks, Inc. [online]. Natick, Massachusetts, USA, 2022 [cit. 2022-08-25]. Dostupné z: <https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html>
- [12] How the Genetic Algorithm Works. The MathWorks, Inc. [online]. Natick, Massachusetts, USA, 2022 [cit. 2022-08-25]. Dostupné z: <https://www.mathworks.com/help/gads/how-the-genetic-algorithm-works.html>
- [13] Randn: Normally distributed random numbers. The MathWorks, Inc. [online]. 2022 [cit. 2022-08-25]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/randn.html>
- [14] Cov: Covariance. The MathWorks, Inc. [online]. 2022 [cit. 2022-08-25]. Dostupné z: <https://www.mathworks.com/help/matlab/ref/cov.html>
- [15] KRIVOŠEJ, Jan. Kinematická kalibrace zohledňující rozložení chyb čidel. Praha, 2016. Bakalářská práce. FS ČVUT. Vedoucí práce Ing. Petr Beneš Ph.D..
- [16] M. Valášek a V. Stejskal, Kinematics and dynamics of machinery, 270 Madison

## Seznam obrázků a tabulek

Obr. 1: Sériová struktura	Paralelní struktura	Redundantní paralelní struktura ....	8
Obr. 2: Změna bodu simplexu v <i>fminsearch</i> .....			18
Obr. 3: Kinematické schéma robotického ramene .....			20
Obr. 4: Rezidua vazbových rovnic před kalibrací.....			26
Obr. 5: Rezidua vazbových rovnic základní kalibrace s uvažováním chyb sensorů $10^{-16}$ .....			27
Obr. 6: Rezidua vazbových rovnic pokročilá kalibrace s uvažováním chyb sensorů $10^{-16}$ .....			27
Obr. 7: Rezidua vazbových rovnic základní kalibrace s uvažováním chyb sensorů $10^{-5}$ .....			28
Obr. 8: Rezidua vazbových rovnic pokročilá kalibrace s uvažováním chyb sensorů $10^{-5}$ .....			29
Obr. 9: Rezidua vazbových rovnic základní kalibrace s uvažováním chyby senzoru $\varphi_{34} 10^{-3}$ .....			30
Obr. 10: Rezidua vazbových rovnic pokročilá kalibrace s uvažováním chyby senzoru $\varphi_{34} 10^{-3}$ , ostatní $10^{-5}$ .....			30
Obr. 11: Vývoj kalibrovatelnosti –základní kalibrace – chyba sensorů $10^{-5}$ – <i>fminsearch</i> .....			38
Obr. 12: Vývoj kalibrovatelnosti –základní kalibrace – nepřesnosti $10^{-5}$ – genetické algoritmy ...			39
Obr. 13: Vývoj kalibrovatelnosti –základní kalibrace – chyba sensorů $10^{-5}$ - genetické algoritmy + <i>fminsearch</i> .....			39
Obr. 14: Rezidua vazbových rovnic – základní kalibrace – chyba sensorů $10^{-5}$ .....			40
Obr. 15: Optimalizované návrhy mechanismu– základní kalibrace – chyba senzoru $10^{-5}$ .....			41
Obr. 16: Vývoj kalibrovatelnosti –pokročilá kalibrace – chyba sensorů $10^{-5}$ – <i>fminsearch</i> .....			42
Obr. 17: Vývoj kalibrovatelnosti –pokročilá kalibrace – chyba sensorů $10^{-5}$ -genetické algoritmy			43
Obr. 18: Vývoj kalibrovatelnosti –pokročilá kalibrace – chyba sensorů $10^{-5}$ – genetické algoritmy + <i>fminsearch</i> .....			43
Obr. 19: Rezidua vazbových rovnic – pokročilá kalibrace – chyba sensorů $10^{-5}$ .....			44
Obr. 20: Optimalizované návrhy mechanismu– pokročilá kalibrace – chyba senzoru $10^{-5}$ .....			45
Obr. 21: Vývoj kalibrovatelnosti – základní kalibrace      chyba senzoru $\varphi_{34} 10^{-3}$ – GA + <i>fminsearch</i> .....			46
Obr. 22: Vývoj kalibrovatelnosti –základní kalibrace .....			46
Obr. 23: Vývoj kalibrovatelnosti – základní kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ – GA + <i>fminsearch</i> .....			46
Obr. 24: Rezidua vazbových rovnic – základní kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ .....			47
Obr. 25: Optimalizované návrhy mechanismu– základní kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ .....			48
Obr. 27: Vývoj kalibrovatelnosti – pokročilá kalibrace .....			48
Obr. 26: Vývoj kalibrovatelnosti – pokročilá kalibrace .....			48
Obr. 28: Vývoj kalibrovatelnosti – pokročilá kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ – GA + <i>fminsearch</i> .....			49
Obr. 29: Grafy reziduí optimalizace pokročilá kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ .....			49
Obr. 30: Optimalizované návrhy mechanismu– pokročilá kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ .....			50
Obr. 31: Optimalizace sumy – základní kalibrace .....			51
Obr. 32: Optimalizace sumy –základní kalibrace .....			51
Obr. 33: Optimalizace sumy – základní kalibrace – chyba sensorů $10^{-5}$ – ga + <i>fminsearch</i> .....			52
Obr. 34: Optimalizované návrhy mechanismu– základní kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ .....			53
Obr. 36: Optimalizace sumy – pokročilá kalibrace.....			53
Obr. 35: Optimalizace sumy – pokročilá kalibrace.....			53
Obr. 37: Optimalizace sumy – pokročilá kalibrace – chyba sensorů $10^{-5}$ – GA + <i>fminsearch</i> .....			54
Obr. 38: Optimalizované návrhy (suma) mechanismu– pokročilá kalibrace – chyba senzoru $10^{-5}$ .			55
Obr. 40: Optimalizace sumy – základní kalibrace .....			57
Obr. 39: Optimalizace sumy – základní kalibrace .....			57
Obr. 41: Optimalizace sumy – základní kalibrace – chyba sensorů $10^{-5}$ – GA + <i>fminsearch</i> .....			57
Obr. 42: Optimalizované (suma) návrhy mechanismu – základní kalibrace – chyba senzoru $10^{-5}$ .			59
Obr. 43: Optimalizace sumy – pokročilá kalibrace.....			59
Obr. 44: Optimalizace sumy – pokročilá kalibrace.....			59
Obr. 45: Optimalizace sumy – pokročilá kalibrace – chyba sensorů $10^{-5}$ – GA + <i>fminsearch</i> .....			60
Obr. 46: Optimalizované (CF – suma) návrhy mechanismu – pokročilá kalibrace – chyba senzoru $10^{-5}$ .....			61

Obr. 47: RedCaM posuvný .....	63
Obr. 48: Kinematické schéma měřicího stroje RedCaM s posuvnými vazbami.....	64
Obr. 49: Kinematické schéma jedné nohy pro RedCaM s posuvnou vazbou .....	64
Obr. 50: Počáteční mechanismus .....	65
Obr. 51: Rezidua vazbové rovnice prostorového mechanismu– základní kalibrace.....	67
Obr. 52: Vývoj CF (kalibrovatelnost) - základní .....	67
Obr. 53: Vývoj CF (kalibrovatelnost) - základní .....	67
Obr. 54: Vývoj CF (suma) - základní kalibrace –GA .....	68
Obr. 55: Vývoj CF (suma) - základní.....	68
Obr. 56: Rezidua vazbové rovnice prostorového mechanismu– pokročilá kalibrace .....	69
Obr. 57: Vývoj CF (kalibrovatelnost) – pokročilá .....	70
Obr. 58: Vývoj CF (kalibrovatelnost) - pokročilá.....	70
Obr. 59 Vývoj CF (suma) - pokročilá .....	70
Obr. 60: Vývoj CF (suma) - pokročilá.....	70
Tabulka 1: Počáteční parametry.....	22
Tabulka 2: Jacobiho matice $J_q$ .....	23
Tabulka 3: Jacobiho matice $J_y$ .....	24
Tabulka 4: Porovnání přesných a zkalibrovaných parametrů při chybě senzorů $10^{-16}$ .....	27
Tabulka 5: Porovnání přesných a zkalibrovaných parametrů při chybě senzorů $10^{-5}$ .....	29
Tabulka 6: Porovnání přesných a zkalibrovaných parametrů při chybě senzoru $\varphi_{34} 10^{-3}$ a $10^{-4}$ , ostatní $10^{-5}$ .....	31
Tabulka 7: Porovnání přesných a zkalibrovaných parametrů při chybě senzoru $\varphi_{34} 10^{-3}$ , ostatní $10^{-16}$ .....	31
Tabulka 8: Kalibrovatelnost příkladů.....	33
Tabulka 9: Směrodatné odchylky čidel všech příkladů kalibrací .....	33
Tabulka 10: Interval parametrů pro genetické algoritmy .....	35
Tabulka 11: Porovnání metod optimalizace a základní kalibrace – chyba senzorů $10^{-5}$ .....	40
Tabulka 12: Porovnání metod minimalizace pokročilé kalibrace – chyba senzoru $10^{-5}$ .....	44
Tabulka 13: Porovnání metod minimalizace – základní kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ – ostatní $10^{-5}$ .....	47
Tabulka 14: Porovnání metod minimalizace a pokročilé kalibrace – chyba senzoru $\varphi_{34} 10^{-3}$ .....	50
Tabulka 15: Porovnání metod minimalizace (změna FF) - základní kalibrace – chyba senzoru $10^{-5}$ .....	52
Tabulka 16: Porovnání metod minimalizace sumy – pokročilá kalibrace – chyba senzoru $10^{-5}$ .....	55
Tabulka 17: Porovnání kalibrace pro pět různých zachybování původního a optimalizovaného mechanismu .....	56
Tabulka 18: Porovnání metod minimalizace sumy – základní kalibrace – chyba senzoru $10^{-5}$ .....	58
Tabulka 19: Porovnání metod minimalizace sumy – pokročilé kalibrace – chyba senzoru $10^{-5}$ .....	60
Tabulka 20: Porovnání výsledků základní kalibrace a optimalizace pro prostorový mechanismus	68
Tabulka 21: Porovnání výsledků pokročilé kalibrace a její optimalizace.....	71

## Přílohy

1. DP-2022-Skorpil\_Marek-priloha.zip