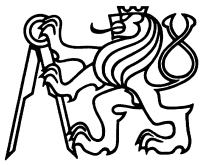
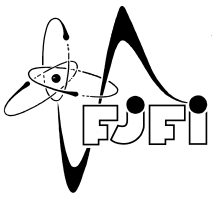


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



Lokalizace mikrorobota z magnetických senzorů

Localization of a microrobot using magnetic sensors

Bakalářská práce

Autor: **Stanislav Novotný**

Vedoucí práce: **doc. Ing. Václav Šmídl, Ph.D.**

Konzultant: **Ing. Martin Juřík**

Akademický rok: 2019/2020

České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská

Katedra: matematiky

Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Stanislav Novotný
Studijní program:	Aplikace přírodních věd
Obor:	Matematické inženýrství
Zaměření:	Matematické modelování
Název práce (česky):	Lokalizace mikrorobota z magnetických senzorů
Název práce (anglicky):	Localization of a microrobot using magnetic sensors

Pokyny pro vypracování:

1. Seznamte se s metodami optimalizace nekonvexních funkcí. Zvláštní pozornost věnujte gradientním metodám s adaptivní volbou kroku. Na jednoduchém příkladě demonstrujte jejich vlastnosti.
2. Seznamte se s daty z magnetické lokalizace mikrorobota. Zvláštní pozornost věnujte modelu reakce senzoru na robota. Diskutujte ostatní možné vlivy na magnetické pole jako případné zdroje poruch.
3. Formulujte úlohu lokalizace mikrorobota jako optimalizační úlohu. Navrhněte ztrátovou funkci a aplikujte již prostudované metody na tuto úlohu. Do úlohy zakomponujte znalost předchozí polohy robota. Diskutujte přesnost výsledku a možné zdroje poruch.
4. Pro zvolené hypotézy o zdrojích poruchy navrhněte jejich kompenzaci a demonstrujte vliv zvoleného řešení na výsledky.

Doporučená literatura:

1. S. Ruder, An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016.
2. M. Jurik, V. Smidl, J. Kuthan, F. Mach, Trade-off Between Resolution and Frame Rate for Visual Tracking of Mini-robots on Planar Surfaces. In 'Proceedings of International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)', 2019.
3. B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman filter. IEEE Aerospace and Electronic Systems Magazine 19(7), 2004, 37–38.

Jméno a pracoviště vedoucího bakalářské práce:

Doc. Ing. Václav Šmídl, Ph.D.
ÚTIA AV ČR, v.v.i., Pod vodárenskou věží 4, 180 00 Praha 8

Jméno a pracoviště konzultanta:

Ing. Martin Juřík
Fakulta elektrotechnická, Katedra teoretické elektrotechniky , Západočeská univerzita v Plzni, Universitní 26, 330 00 Plzeň

Datum zadání bakalářské práce: 31.10.2019

Datum odevzdání bakalářské práce: 7.7.2020

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 23. října 2019

garant oboru

vedoucí katedry

děkan

Poděkování:
Chtěl bych zde poděkovat především svému školiteli docentu Václavu Šmídlovi za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé diplomové práce.

Čestné prohlášení:
Prohlašuji, že jsem tuto práci vypracoval samostatně a uvedl jsem všechnu použitou literaturu.

V Praze dne 7. července 2020 Stanislav Novotný

Lokalizace mikrorobota z magnetických senzorů

Lokalizace mikrorobota z magnetických senzorů

Autor: Stanislav Novotný

Obor: Matematické inženýrsvtí

*Zaměření:*Matematické modelování

Druh práce: Bakalářská práce

Vedoucí práce: Doc. Ing. Václav Šmídl, Ph.D., ÚTIA AV ČR, v.v.i. Pod vodárenskou věží 4 180 00 Praha 8

Konzultant: Ing. Martin Juřík, Fakulta elektrotechnická Katedra teoretické elektrotechniky Západočeská univerzita v Plzni Universitní 26 330 00 Plzeň

Abstrakt: Tato bakalářská práce se zabývá optimalizačními metodami a zaměřuje se konkrétně na spádové. Řeší úlohou lokalizace mikrorobota v magnetickém poli pomocí senzorů. Tato práce obsahuje vytvoření matematického dvourozměrného modelu magnetického pole v okolí robota a následně lokalizaci robota pomocí tohoto modelu. Jsou zde uvedeny experimenty znázorňující chování metod ADAM a RMSProp při konvergenci a jejich následné vyhodnocení. Dále je zde uvedena citlivostní studie modelu.

Klíčová slova: lokalizace mikrorobota, optimalizační metody, model magnetického pole

Localization of a microrobot using magnetic sensors

Localization of a microrobot using magnetic sensors

Author: Stanislav Novotný

Abstract: This bachelor thesis deals with optimization methods and focuses especially on gradient methods. Thesis solves the problem of locating a microrobot in a magnetic field using sensors. This work includes the creation of a mathematical two-dimensional model of the magnetic field around the robot and then the localization of the robot using this model. Experiments showing the behavior of ADAM and RMSProp convergence and their subsequent evaluation are presented. The sensitivity study of the model is also presented here.

Key words: magnetic field model, microrobot localization, optimization methods

Obsah

Úvod **10**

1 Optimalizační metody **11**
1.1 Metody nevyužívající derivace 11
1.1.1 Bisekce 11
1.1.2 Fibonacciovské dělení 12
1.1.3 Metoda zlatého řezu 12
1.2 Newtonova metoda 13
1.3 Metoda největšího spádu 13
1.3.1 RMSProp 14
1.3.2 ADAM 14

2 Data **15**

3 Volba modelu **18**
3.1 Aproximace dat polynomem 1D 18
3.1.1 Metoda nejmenších čtverců 19
3.2 Fyzikální model 1D 20
3.3 Kombinovaný model 1D 22
3.4 Fyzikální model 2D 23
3.5 Kombinovaný model 2D 28

4 Lokalizace **33**

Závěr **37**

Úvod

Všichni optimalizujeme. Každý se snažíme zkrátit si cestu do práce nebo do školy o každý možný metr. Firmy se snaží maximalizovat jejich výrobní procesy. Samotná příroda se snaží dostat systémy do stavů s nejnižší energií. Světelné paprsky létají k našim očím po co nejkratší dráze. V této bakalářské práci se seznámíme s různými optimalizačními technikami, větší pozornost budeme věnovat těm, které využívají gradientní metody.

Budeme je aplikovat na reálný případ lokalizace robota v magnetickém poli. Manipulace s mikrorobotem pomocí magnetů je v současné době objektem mnoha studií. Uplatnění tohoto výzkumu se dá využít například v medicíně, kde by mohli být mikroroboti použiti k přepravě účinné látky do těžko dostupného postiženého místa přímo v pacientově těle nebo například v průmyslu pro jemnou práci. Náš model robota v praxi představuje malá ploška, na které je umístěno 5 magnetek stejně, jako na hrací kostce. Tento magnet se nachází na destičce v magnetickém poli, které je generováno dvěma cívkami. Pod magnetem se nechází senzor skenující magnetické pole.

Bakalářská práce je rozdělena do více částí. Na začátku se seznámíme s teoretickým podkladem k optimalizaci a uvedeme si příklady několika typů optimalizačních technik. Dále se seznámíme s experimentem, který nám poskytl naměřená data. Tyto data využijeme k sestrojení matematického modelu magnetického pole v kapitole Volba modelu. Následně onen model použijeme k lokalizaci mikrorobota, kde budeme řešit jeho přesnost a vnější vlivy způsobující odchylky. Závěr pak věnujeme zhodnocení celkové práce a dosažených výsledků.

Kapitola 1

Optimalizační metody

Optimalizace je důležitý vědecký nástroj. V této kapitole probereme základy optimalizačních metod, jejich využití a příklady využívané v této práci.

Jak již samotný název napovídá, optimalizační metody se využívají k výběru nejvhodnější možné varianty nějakého jevu. Nejprve si určíme náš cíl, čímž může být například profit, energie, čas, zjednodušeně cokoliv, co dokážeme vyjádřit pouhým číslem. Tento náš cíl je spjatý s jistými charakteristikami, které nazveme proměnné. Budeme se snažit najít takové hodnoty těchto proměnných, abychom optimalizovali náš cíl. Proměnné se však také řídí podle určitých podmínek, například výplata naštěstí nemůže být záporná. Hledání tohoto cíle, proměnných a omezení určitého problému se říká modelování. Jakmile máme model, můžeme se pustit do optimalizace. Neexistuje však univerzální způsob optimalizace. Volba správného optimalizačního algoritmu je pouze na nás. Tato volba je kritická, jelikož určuje, jak dobrých výsledků můžeme dosáhnout a pokud vůbec k výsledku dojdeme. Podle [5] matematicky řečeno hledáme maximum nebo minimum nějaké cenové funkce $f : D \to \mathbb{R}$. Extrémy naší cenové funkce se snažíme najít na přípustné množině $\Omega \subseteq D$. Víme, že minimum f je stejné jako maximum $-f$, proto hledáme takové $\hat{x} \in \Omega$, které by splňovalo $f(\hat{x}) \leq f(x), x \in \Omega$, takovéto x nazýváme globálním řešením. Hledání globálního minima však může být příliž obtížný úkol, má smysl se tedy zabývat hledáním lokálního minima, tj. je-li $\hat{x} \in \Omega$ lokálním řešením, pak existuje okolí H bodu $\hat{x}$ takové, že $f(\hat{x}) \leq f(x), x \in \Omega \cap H$. Skutečnost, že se v iteracích blížíme minimu můžeme poznat pomocí Taylorovy věty o aproximaci funkce polynomem z níž poznáme, jestli přírustek funkce f je směrem poklesu. Mnoho tvrzeních je také založeno na tom, zdali-li je funkce f konvexní. U konvexních funkcí platí, že každé jejich lokální minimum je globální minimum. Optimalizační metody můžeme využít v široké škále různých oborů, například ve finanční sféře, či průmyslu.

Nyní si podle [2] uvedeme přehled několika algoritmů, pomocí nichž můžeme najít minimum funkce.

1.1 Metody nevyužívající derivace

V této kapitole se dozvíme něco o metodách bisekce, zlatého řezu a Fibonacciovském dělení, které fungují jak na úlohy jednorozměrné, tak i na úlohy vícerozměrné optimalizace. Tyto metody byly rozvíjeny především v šedesátých letech minulého století. Jejich výhodou je snadná implementace a schonost najít minima nediferencovatelných či nespojitých funkcí. Jsou také rychlejší v tom smyslu, že během iterací nemusíme napočítávat Hessián ani gradient.

1.1.1 Bisekce

Metoda bisekce je založena na následující větě.

Věta 1.1.1. *Nechť pro funkci* $f : \mathbb{R} \to \mathbb{R}$ *na intervalu* $< a, b >$ $(-\infty \leq a \leq b \leq \infty)$ *existuje* $c \in (a, b)$ *takové, že f je klesající na* $< a, c >$ *a rostoucí na* $< c, b >$. *Dále nechť f nabývá minima v* $x* \in < a, b >$. *Pak pro libovolné* $x_1, x_2,$ *kde* $a \leq x_1 \leq x_2 \leq b$ *platí:*

1. *Je-li* $f(x_1) < f(x_2)$, *pak* $x* < x_2$.
2. *Je-li* $f(x_1) > f(x_2)$, *pak* $x* > x_1$.
3. *Je-li* $f(x_1) = f(x_2)$, *pak* $x_1 < x* < x_2$.

Důkaz. Postupujeme sporem, nechť platí předpoklady.

1. Je-li $f(x_1) < f(x_2) \wedge x* \geq x_2$. Pak f je klesající na $< a, x* >$ tj. $f(x_2) < f(x_1)$, což je spor.
2. Je-li $f(x_1) > f(x_2) \wedge x* \leq x_2$. Pak f je rostoucí na $< x*, b >$ tj. $f(x_2) > f(x_1)$, což je spor
3. Je-li $f(x_1) = f(x_2) \wedge (x* \leq x_1 \vee x* \geq x_2)$. Pak buď f je rostoucí na $< x*, b >$ nebo f je klesající na $< a, x* >$, což jsou opět spory s předpoklady.

□

V algoritmu vytváříme posloupnost zmenšujících se intervalů, kde každý takový interval obsahuje hledané minimum. Pro funkci splňující předpoklady předchozí věty postupujeme následovně: V intervalu $I_k =< a_k, b_k >$ zvolíme body $c_k := \frac{a_k+b_k}{2}, d_k := \frac{a_k+c_k}{2}, e_k := \frac{c_k+b_k}{2}$. Využijeme znění věty, tj. porovnáme funkční hodnoty v jednotlivých podintervalech a podle toho se rozhodneme, který interval budeme půlit jako další. Jako konec algoritmu můžeme zvolit například námi požadovanou délku intervalu obsahujícího řešení. Minimum funkce f pak bude představovat střed posledního takto vypočítaného intervalu.

1.1.2 Fibonacciovské dělení

Fibonacciho posloupnost je posloupnost zadaná rekurzivně předpisem

$$F_{n+2} = F_{n+1} + F_n,\ F_0 = 1 = F_1$$

Mějme funkci $f : \mathbb{R} \to \mathbb{R}$ na intervalu $< 0, 1 >$, pro kterou existuje $c \in (0, 1)$ takové, že f je klesající na $< 0, c >$ a rostoucí na $< c, 1 >$. Apriori zvolíme počet členů posloupnosti jako $N \in \mathbb{N}$, tím určíme i požadovanou přesnost odhadu minima. Dále napočítáme body $x_1 = \frac{F_{N-1}}{F_N}$ a $x_2 = \frac{F_{N-2}}{F_N}$. Porovnáme funkční hodnoty v těchto bodech, podle věty 1.1.1 dostaneme $< 0, x_1 >$ nebo $< x_2, 1 >$, tj. interval obsahující minimum funkce f. V tomto intervalu dále napočítáme $x_3 = 0 + \frac{F_{N-3}}{F_N}$, či $x_3 = 1 - \frac{F_{N-3}}{F_N}$. Postup opakujeme, dokud nezískáme x_{N-1}.

1.1.3 Metoda zlatého řezu

U metody Fibonacciho dělení si můžeme všimnout, že pokud chceme změnit přesnost aproximace minima, čili $N \in \mathbb{N}$, musíme znovu napočítat všechny $x_1, ..., x_{N-1}$. Metoda zlatého řezu tento problém odstraní tím, že si zafixujeme hodnotu zkracování intervalu na konstantě $\hat{\gamma} = \lim_{N\to+\infty} \frac{F_{N-1}}{F_N}$. Dělíme tedy náš interval I_k na podintervaly m_k a n_k. Přitom platí, že

$$\frac{m_k}{n_k} = \frac{I_k}{m_k} = \gamma = konst.$$

Využijeme-li toho, že $I_k = m_k + n_k$, dostaneme $\gamma = 1 + \frac{1}{\gamma}$, kladnému řešení této rovnice se říká zlatý řez $\gamma = \frac{1+\sqrt{5}}{2}$. Tato metodu funguje stejně jako metoda bisekce s tím rozdílem, že v intervalu $I_k =< a_k, b_k >$ zvolíme body $c_k := a_k + b_k - d_k,\ kde\ d_k := \frac{b_k-a_k}{\gamma} + a_k$.

1.2 Newtonova metoda

Z [1] již bude zapotřebí, aby naše cílová funkce $f : \mathbb{R} \to \mathbb{R}$ byla dvakrát spojitě diferencovatelná a konvexní v okolí řešení. Tuto metodu využíváme pro hledání kořenů diferencovatelné funkce f, to jest k hledání takového $x \in \mathbb{R}$, pro které $f(x) = 0$. Navýšíme-li však naše nároky na podmínky diferencovatelnosti, můžeme hledat extrémy f, tj. $x \in \mathbb{R}$, pro které $f'(x) = 0$. Pomocí Newtonovy metody sestrojíme posloupnost bodů $(x_k)_{k=0}^{n}$, která bude konvergovat k minimu f, $x_0 \in \mathbb{R}$ bude námi zvolený počáteční bod. Využijeme při tom Taylorův rozvoj funkce f do druhého řádu v bodě x_k

$$f(x_k + t) \approx f(x_k) + f'(x_k)t + \frac{1}{2}f''(x_k)t^2 =: F_{k,t}$$

následující bod $x_{k+1} := x_k + t$ určíme tak, aby byl výše uvedený odhad co nejmenší vzhledem k t. Je-li druhá derivace v bodě x_k kladná, znamená to, že náš odhad je konvexní funkcí. Dále stačí položit derivaci rovnu nule.

$$\frac{d}{dt}F_{k,t} = f'(x_k) + f''(x_k)t = 0 \Rightarrow t = -\frac{f'(x_k)}{f''(x_k)}$$

iterace tedy vypadá

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

Ve vícerozměrném případě bychom dostali

$$x_{k+1} = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

Tato iterace se dá ještě modifikovat přidáním určité délky kroku. Jelikož je výpočet $\nabla^2 f(x_k)^{-1}$ náročný, hledáme obvykle $-\nabla^2 f(x_k)^{-1} \nabla f(x_k) =: A_k$ jako řešení soustavy $\nabla^2 f(x_k) A_k = -\nabla f(x_k)$

1.3 Metoda největšího spádu

Pro účely naší práce budeme pracovat s metodami největšího spádu, které pro svůj chod počítají s gradientem. Jsou to iterativní metody prvního řádu, ve většině případů se optimalizací nedostaneme do exaktního bodu, ve kterém funkce dosahuje lokálního minima kvůli délce kroku. Budeme se muset spokojit s tím, že pokles cenové funkce budeme muset zastavit splněním uspokojivého ukončovacího kritéria. Na druhou stranu jsou tyto metody velice robustní a lehké na implementaci. Podle [7] se budeme se snažit optimalizovat funkci $J(x)$ závislou na parametru $x \in \mathbb{R}^d$ tím, že podnikneme kroky v záporně daném smyslu jejího gradientu $\nabla J(x)$ v určitých bodech. Máme-li totiž $J(x)$ diferencovatelnou na nějakém okolí bodu a, pak $-\nabla J(a)$ je směr nejrychlejšího sestupu v tomto bodě. Symbolem η budeme značit délku jednotlivých kroků. Metoda největšího spádu upravuje parametr x v každém napočítaném kroku následovně:

$$x_t = x_{t-1} - \eta * \nabla J(x_{t-1}) \tag{1.1}$$

implementace v kódu by pak mohla vypadat takto:

```
for i in range(počet_ kroků):
gradient_param = vypočítej_gradient( ztrátová_funkce, data , parametr)
parametr = parametr - délka_kroku * gradient_param
```

Dále si podle [7] uvedeme dvě nejpoužívanější spádové metody.

1.3.1 RMSProp

Metoda největšího spádu často selhává v oblastech, kde spád pro jednu složku parametru funkce je mnohem větší než pro druhou, například v okolí sedlových bodů. RMSProp tento problém řeší pomocí toho, že samotný gradient podělíme odmocninou ze střední hodnoty předchozích gradientů umocněných na druhou. Tím zajistíme, že se budeme stále pohybovat správným směrem s větší rychlostí i v případě, kdy je gradient funkce malý.

$$g_t = \nabla J(x_t)$$
$$E[g^2]_t = \beta * E[g^2]_{t-1} + (1-\beta) * g_t^2$$
$$x_t = x_{t-1} - \eta * \frac{g_t}{\sqrt{E[g^2]_t + \epsilon}}$$

β zde značí pohybový parametr a v praxi se využívá hodnota 0.9, $E[g]_t$ je střední hodnota gradientu, η délka kroku, ϵ se přidává do jmenovatele, aby se zabránilo dělením nulou

1.3.2 ADAM

ADAM vylepšuje RMSProp tím, že přidává střední hodnotu předešlých gradientů, což bychom mohli chápat jako hybnost algoritmu, která se zvětšuje ve směru předešlých gradientů a změnšuje v ostatních. Tím dosáhneme rychlejší konvergence a menších oscilací.

$$m_t = \beta_1 * m_{t-1} + (1-\beta_1) * g_t$$
$$v_t = \beta_2 * v_{t-1} + (1-\beta_2) * g_t^2$$

m_t zde značí již zmíněnou hybnost a v_t je převzaté z RMSProp, β_1 a β_2 jsou pohybové parametry, g_t gradient, jako v předešlé metodě.

Jelikož m_t i v_t nazačátku inicializujeme jako nulové vektory, zůstávají v prvních krocích algoritmu blízké nule. K tomuto fenoménu také napomáhá, pokud jsou β_1 a β_2 blízké 1. Proto tyto vektory upravujeme následujícím způsobem.

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Parametry funkce následně aktualizujeme takto:

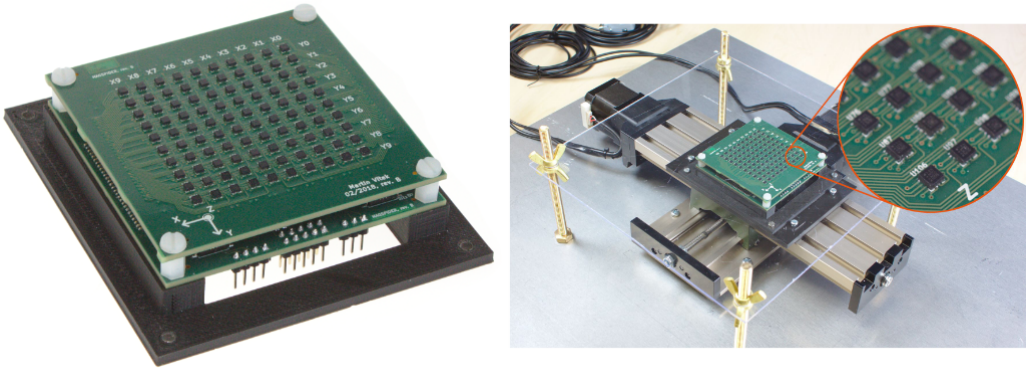
$$\theta_t = \theta_{t-1} - \eta * \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Nejčastěji volíme $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$

Kapitola 2

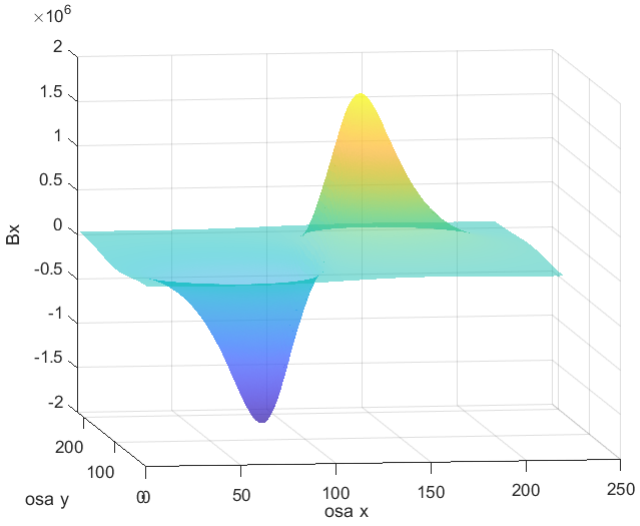
Data

Data nám byla poskytnuta z experimentu, ve kterém se robot nacházel na destičce uvnitře magnetického pole generovaného cívkami. Podle [4] bylo nejdříve snímáno zařízením MagSpider, ale prostorový krok 5 mm mezi senzory spolu s dalšími odchylkami nebyl dostatečně přesný, proto byl sestrojen DerMagTisch, kterým lze velmi jemně pohybovat s krokem 5 mikrometrů. Z tohoto přístroje dostáváme údaje o působení magnetického pole na našeho robota. V souřadnicích $[x, y]$ měříme 3D vektor magnetického pole $[Bx, By, Bz]$, každou z těchto složek představuje matice $220x220$.

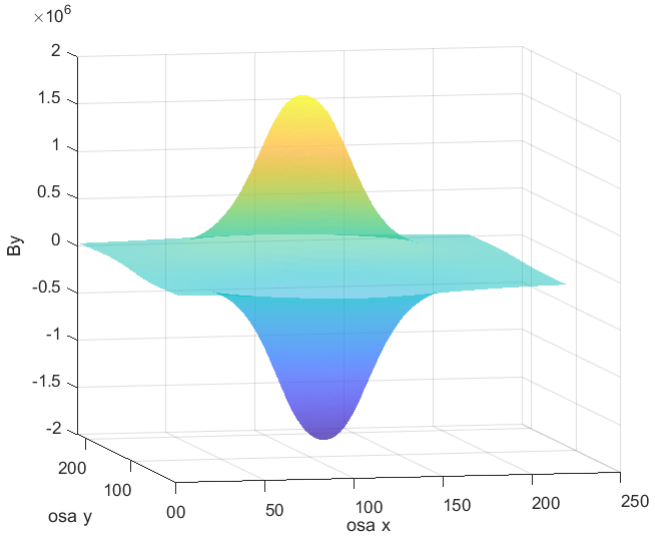


Obrázek 2.1: MagSpider (vlevo) a DerMagTisch (vpravo)

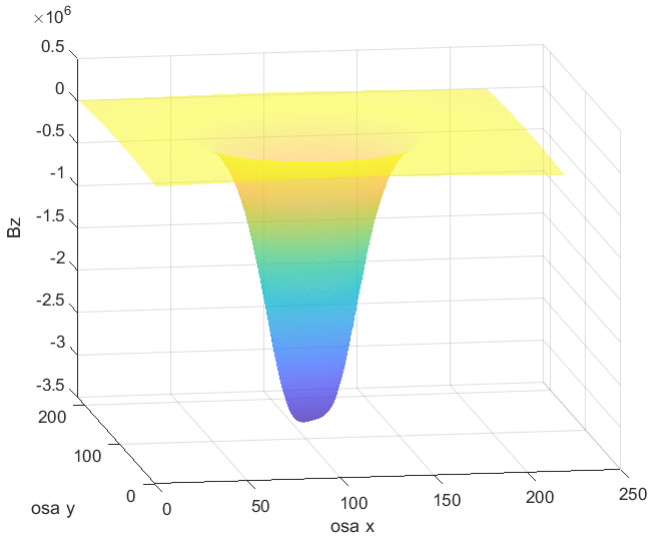
Na obrázcích níže můžeme vidět, jak naše data vypadají.



Obrázek 2.2: Zobrazení závislosti odezvy osy Bx magnetického sensoru na poloze obota vzhledem k sensoru

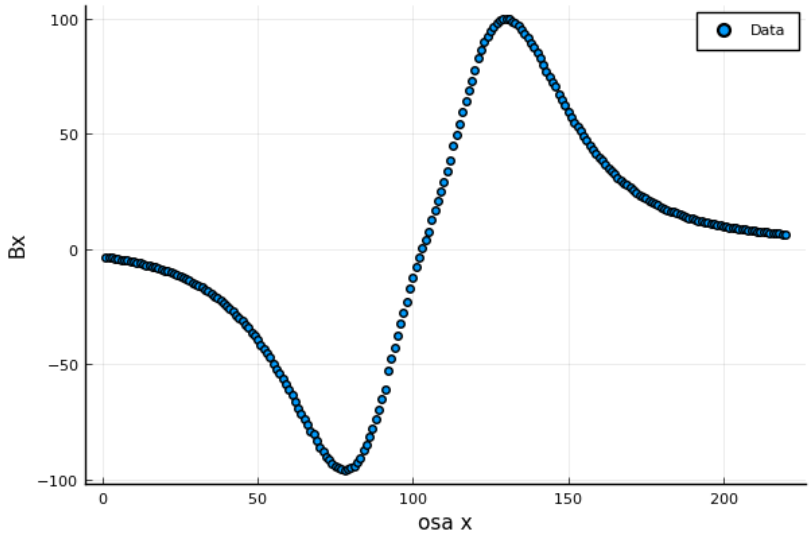


Obrázek 2.3: Zobrazení závislosti odezvy osy By magnetického sensoru na poloze obota vzhledem k sensoru



Obrázek 2.4: Zobrazení závislosti odezvy osy Bz magnetického sensoru na poloze obota vzhledem k sensoru

Můžeme si všimnout, že největší hodnoty dosahují čísel $1.5x10^6$, proto jsme museli data škálovat, abychom předešli případným chybám při mocnění.



Obrázek 2.5: Přeškálovaný řez daty Bx v y=0

Kapitola 3

Volba modelu

Máme naměřená data, jež představuje matice trojrozměrných vektorů. Nejdříve se zaměříme na jednorozměrný případ, kdy si vezmeme průřez dat v ose y. Naším úkolem v této kapitole bude najít křivku, kterou bychom mohli data proložit s uspokojivou přesností. Pracujeme v jazyce Julia ver. 1.4.2, k implementaci gradientních metod využíváme knihovnu Flux.

3.1 Aproximace dat polynomem 1D

Z matematické analýzy známe větu, která nám říká o aproximaci polynomem.

Věta 3.1.1 (Taylorova)**.** *Nechť funkce f má v bodě $a \in \mathbb{R}$ konečnou n-tou derivaci, kde $n \in \mathbb{N}$, dále nechť existuje okolí U_a takové, že platí*

1. *v každém $x \in U_a$ existuje konečná (n-1)-ní derivace f*
2. *v bodě a existuje konečná n-tá derivace funkce f*

Dále nechť $Q(x)$ je polynom stupně $\leq n$, různý od Taylorova polynomu $T_n(x)$ příslušejícího funkci f v bodě a. Pak existuje takové okolí H_a, že

$$| f(x) - T_n(x) | < | f(x) - Q(x) | \text{ pro každé } x \in H_a - \{a\}$$

Důkaz. Uvažujme polynomy T_n a Q ve tvaru

$$T_n(x) = \sum_{k=1}^{n} \alpha_k (x-a)^k \qquad a \qquad Q(x) = \sum_{k=1}^{n} \beta_k (x-a)^k$$

Jelikož se jedná o dva různé polynomy, platí

$$i := min\{\, k \mid \alpha_k \neq \beta_k \,\} \leq n$$

Použijeme vyjádření funkce f pomocí Peanova zbytku $f(x) = T_n(x) + \omega_n(x) * (x-a)^n$. Pro $x \neq a$ dostaneme

$$\left| \frac{f(x) - Q(x)}{(x-a)} \right| = \left| \frac{T_n(x) + \omega_n(x) * (x-a)^n - Q(x)}{(x-a)^i} \right| = \left| \frac{\sum_{k=i}^{n} (\alpha_k - \beta_k) * (x-a)^k + \omega_n(x) * (x-a)^n}{(x-a)^n} \right| =$$

$$= \left| (\alpha_i - \beta_i) + \sum_{k=i+1}^{n} (\alpha_k - \beta_k) * (x-a)^{k-i} + \omega_n(x) * (x-a)^{n-i} \right|$$

Tedy pro x jdoucí k a je

$$\left|\frac{f(x)-Q(x)}{(x-a)^i}\right| \to \left|\alpha_i - \beta_i\right| > 0.$$

Zatímco

$$\left|\frac{f(x)-T_n(x)}{(x-a)^i}\right| \to 0.$$

Z věty o nerovnostech v limitě plyne, že existuje oholí H_a bodu a takové, že

$$\left|\frac{f(x)-Q(x)}{(x-a)^i}\right| > \left|\frac{f(x)-T_n(x)}{(x-a)^i}\right| \text{ pro každé } x \in H_a - \{a\}.$$

Po vynásobení poslední rovnosti kladným $\left|(x-a)^i\right|$ dostaneme tvrzení věty □

Pokusíme se tedy data fitovat pomocí polynomu dostatešně velkého stupně. Zvolíme si například polynom $p(x)$ stupně 8. Při aplikaci gradientní metody na $p(x)$ však narazíme na problém. Maximální hodnota, kterou naše naměřená data obsahují se rovná 100, tuto hodnotu dále mocníme na osmou, v gradientní metodě z tohoto čísla děláme kvadrát. Pro takto velká čísla algoritmus selhvává, protože je obtížné najít krok, který by vedl k řešení úlohy. Aproximace polynomem je však problém, který se dá řešit analyticky pomocí metody nejmenších čtverců, kterou si představíme v následující podkapitole.

3.1.1 Metoda nejmenších čtverců

Vycházet budeme z [3]. Mějme funkci $f(x)$, kterou se snažíme aproximovat polynomem $Q_n(x) = a_0 + a_1x + ... + a_nx^n$. Pro fixní množinu rozdílných bodů $(x_0, x_1, ..., x_M)$ se budeme snažit minimalizovat

$$\left\|f(x) - Q_n(x)\right\| = \sqrt{\sum_{i=0}^{M}[f(x_i) - Q_n(x_i)]^2}$$

přes všechny polynomy stupně nejvýše n. Pokud by byl počet bodů, ve kterých máme naměřená data menší než stupeň polynomu, bylo by možné úlohu řešit pomocí Lagrangeova interpolačního polynomu. Budeme se tedy zajímat pouze o případ, kdy je objem dat, která chceme nafitovat, větší než stupeň polynomu (v našem případě $M = 220 > 8 = n$).

Jelikož je odmocnina ostře rostoucí funkce, můžeme se soustředit pouze na výraz

$$\phi(a_0, ..., a_n) := \sum_{i=0}^{M}[f(x_i) - Q_n(x_i)]^2 = \sum_{i=0}^{M} f^2(x_i) - 2\sum_{k=0}^{n} a_k \sum_{i=0}^{M} x_i^k f(x_i) + \sum_{k=0}^{n}\sum_{j=0}^{n} a_k a_j \sum_{i=0}^{M} x_i^{k+j}$$

Z matematické analýzy víme, že minimum této funkce dostaneme pomocí parciálních derivací, které položíme rovny 0 (čtenář si rozmyslí, že extrém získaný derivací bude opravdu minimum a né maximum, jelikož $\phi(a_0, ..., a_n)$ maxima nenabývá).
Čili

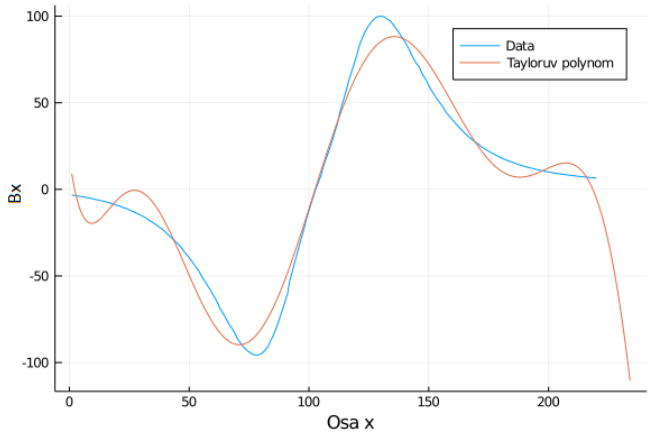
$$\frac{\partial\phi}{\partial a_k} = 0 \Rightarrow \sum_{j=0}^{n} a_j \sum_{i=0}^{M} x_i^{k+j} = \sum_{i=0}^{M} x_i^k f(x_i), \quad k = 0, 1, ..., n$$

Tuto rovnici můžeme přeznačit jako

$$Ax = b, \quad kde\ A \in \mathbb{R}^{n+1\ x\ n+1}, x \in \mathbb{R}^{n+1}, b \in \mathbb{R}^{n+1}$$

Což je soustava n+1 lineárních rovnic, kterou již umíme vyřešit.

Pokud se však podíváme co se děje s polynomem za hranicí našich dat, zjistíme, že fit nesplňuje fyzikální zákony. Jelikož víme, že síla magnetického pole je nepřímo úměrná vzdálenosti od středu působení. Pro Taylorův polynom $t(x)$, kterým jsme data fitovali však platí $\lim_{x\to+\infty} t(x) = \infty$.



Obrázek 3.1: Aproximace polynomem

3.2 Fyzikální model 1D

Z experimentu, ze kterého máme naměřená data víme, že magnetické pole bylo měřeno zařízením DerMagTisch, který se skládá z malých senzorů. Z [6] víme, jak vypadá standardní fyzikální model pro dipól. Tento model je úměrný r^{-4}, kde r je vzdálenost senzoru od robota. Dle toho tedy jako funkci pro fit dat z fyzikálního hlediska volíme

$$F(x) = p_1 * \frac{x}{(x^2 + z^2)^2}, \text{ pro řez osou y v Bx}$$

Víme navíc, že robot se pohybuje po destičce, která je položená rovnoběžně se senzorovou mřížkou, tudíž můžeme proměnnou z považovat za konstantu. Budeme se tedy snažit data nafitovat pomocí polynomu

$$F(x) = p_1 * \frac{x}{(x^2 + p_2)^2}$$

Při výběru gradientní metody, kterou se úlohu budeme snažit vyřešit, budeme postupovat experimentálně. Vybírat budeme mezi RMSProp a ADAM. Zvolíme si maximální počet kroků a po něm vyhodnotíme výsledky. Důležitým ukazatelem konvergence je také ztrátová funkce *loss*, kterou definujeme jako

$$loss(x, y) = \sum_{i=1}^{n} (m_i - y_i)^2$$

kde n je objem dat, x přestavuje n-rozměrný vektor bodů, ve kterých vyhodnocujeme model s parametry p, y je vektor dat, m_i je funkční hodnota modelu v x_i a y_i je hodnota dat na i-té pozici v y. Řídit se

budeme odchylkou danou pomocí residua res, které vypočteme, jako rozdíl hodnoty naměřených dat a funkčních hodnot zvoleného modelu. res je tedy vektor s 220 elementy, ze kterého budeme určovat střední a maximální absolutní hodnotu

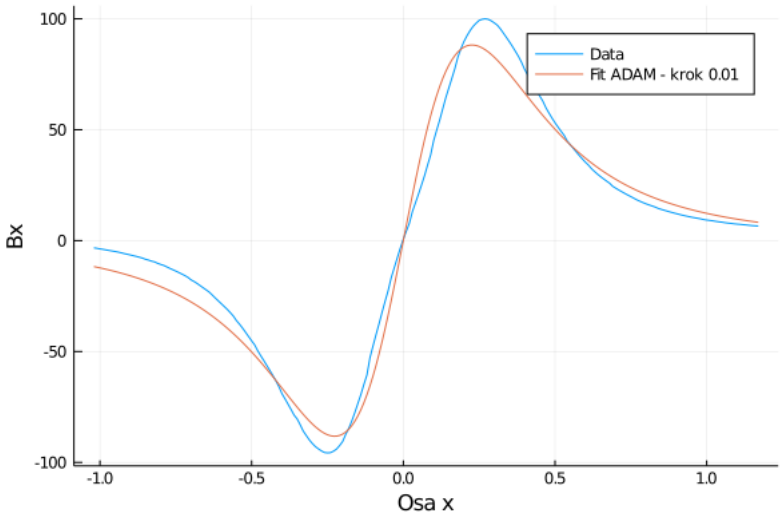
$$MEA = \frac{1}{n}\left|\sum_{i=1}^{n} r_j\right|, \quad MAX = \max_{r_j}\left|r_j\right|$$

kde n je počet prvků v res a $r_j, kde\, j \in (1, ..., n)$ jsou prvky res. MEA, MAX, res i $loss$ obdobně zavádíme i pro dvourozměrné modely.

Určení počátečních koeficientů, tj koeficientů funkce, kterou budeme data aproximovat je úloha hledání globálního minima, která nemá obecné řešení. Mohli bychom to zkoušet různým dělením intervalů, jelikož to ale není cíl naší práce, pomůžeme si programem. Počáteční koeficienty polynomu $Q(x)$ jsme volili blízké těm, které poskytl program MATLAB R2019b, $(p_1, p_2) = (-25, 8, 50, 6)$. V počátečních koeficientech je $MEA = 28.0$ a $MAX = 81.9$

Výběr metody pro 1D fyzikální model Bx				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	1	500	40.0	99.9
	0.1	500	7.01	18.7
	0.01	500	6.87	17.7
	0.001	500	6.82	18.0
	0.0001	500	6.72	18.9
RMSProp	1	500	36.4	93.4
	0.1	500	20.0	62.6
	0.01	500	7.08	19.2
	0.001	500	6.88	17.3
	0.0001	500	6.74	18.7

Hodnota ztrátové funkce se přitom ani v jednom z pokusů nedostala pod hodnotu 14 000. Jako ukončovací podmínku algoritmu jsem nastavil maximální počet kroků 500. Vidíme, že se změnšujícím se krokem bylo dosaženo lepších výsledků. Na obrázku níže je zobrazen fit daty při volbě metody ADAM s krokem 0.01. Výsledné koeficienty měly hodnotu $(p_1, p_2) = (16.36, 0.15)$



Obrázek 3.2: Aproximace fyzikální metodou řezu Bx v y=0

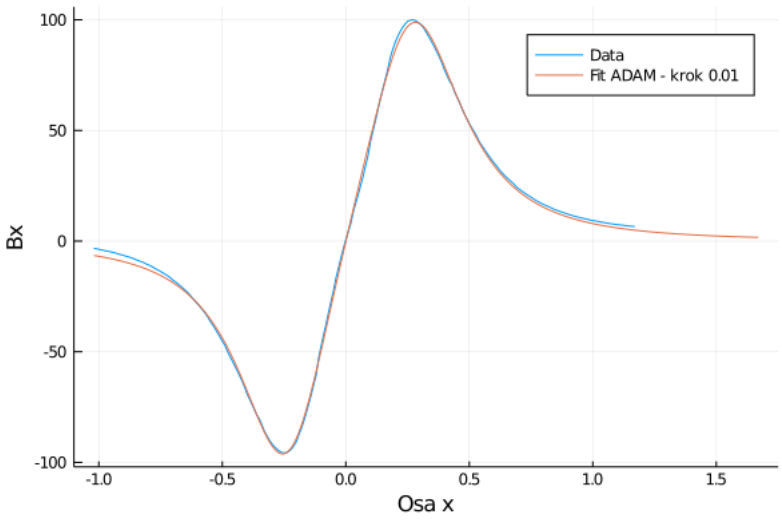
3.3 Kombinovaný model 1D

Myšlenka je taková, že nakombinováním předešlých metod bychom měli dospět k přesnějšímu modelu. Najdeme si vlastní bázi funkcí. Pokusíme se data aproximovat racionální lomenou funkcí. Zvolíme funkci $Q(x) = \frac{f(x)}{g(x)}$, kde $f(x) = p_1 * \frac{x-p_2}{p_3} + p_4$ a $g(x) = (\frac{x-p_5}{p_6})^4 + p_7 * (\frac{x-p_8}{p_9})^3 + p_{10} * (\frac{x-p_{11}}{p_{12}})^2 + p_{13} * \frac{x-p_{14}}{p_{15}} + p_{16}$. Koeficienty p_1 a p_3 atp., které jsou z matematického hlediska zbytečné, jelikož se zkrátí přidáváme, abychom vyzkoušeli, jak se algoritmus bude chovat při větším počtu parametrů. Počáteční hodnoty koeficientů polynomů $f(x)$ a $g(x)$ znovu určíme blízké těm, které dostaneme z MATLAB R2019b při fitu polynomem $Q(x)$ jako $(p_1, ..., p_{16}) = (30, 110, 65, 4, 110, 67, 1, 107, 60, 0.5, 105, 61, 0.5,$
$105, 62, 0.5)$. Je vidět, že $\lim_{x\to+\infty} Q(x) = 0$. Optimální gradientní metodu budeme hledat stejně jako v předešlé metodě.

Výběr metody pro 1D kombinovaný model Bx				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	1	500	40.2	100
	0.1	500	1.40	4.67
	0.01	500	1.46	3.71
	0.001	500	1.45	3.89
	0.0001	500	2.14	7.33
RMSProp	1	500	40.2	100
	0.1	500	40.2	100
	0.01	500	2.76	8.07
	0.001	500	1.52	3.93
	0.0001	500	1.43	3.99

Z tabulky je vidět, že data nejlépe aproximovala metoda ADAM s krokem 0.01. Hodnota ztrátové funkce se držela nad hranicí 600. Konečné parametry z metody ADAM s krokem 0,01 jsou $(p_1, ..., p_{16}) = (25.2, 0.004, 0.116, 0.688, 0.014, 0.43, -1.22, -0.27, 1.25, 0.251, 0.6, 0.72, 1.17, -0.34, 1.38, 0.001)$. Z obrázku níže je patrné, že tento fit již na naše data sedí a odpovídá našim podmínkám i za hranicí dat. Na

jednorozměrném případě jsme tedy demonstrovali, že náš kombinovaný model, který se od fyzikálního liší přidáním volných parametrů a proměnných ve větších mocninách lépe aproximuje data. Tohoto poznatku se tedy budeme dále snažit využít i ve 2D případě.



Obrázek 3.3: Aproximace kombinovanou metodou Bx v y=0

Ve dvourozměrném případě máme k dispozici data v podobě tří matic 220x220, představující působení magnetického pole v osách x,y a z. Hledáme tedy zobrazení 2D $\rightarrow$ 1D, které bodu $[x, y]$ přiřadí příslušnou hodnotu matice dat.

Z věty 2.1.1 víme, že můžeme funkci opět aproximovat dobře na nějakém jejím okolí. Avšak ukázali jsme si, že tento fit není dobrý mimo naměřená data, proto se jím již nebudeme zabývat. Přejdeme rovnou k fyzikálnímu přistupu. Na obrázku níže vidíme, jak vypadají data v Bx složce magnetického pole.

3.4 Fyzikální model 2D

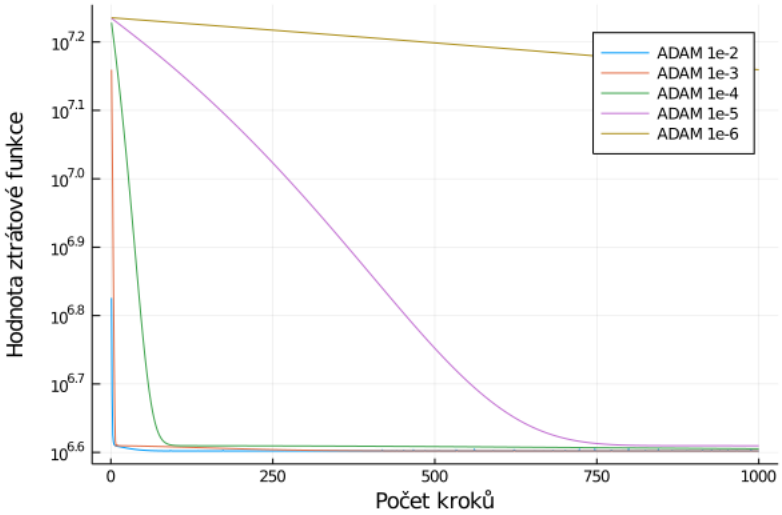
Opět se odkážeme na rovnici (1) v [6] a podle toho zvolíme náš fyzikální model. Začneme složkou Bx. Data budeme aproximovat polynomem

$$F(x, y) = p_1 \frac{x}{(x^2 + y^2 + p_2)^2}$$

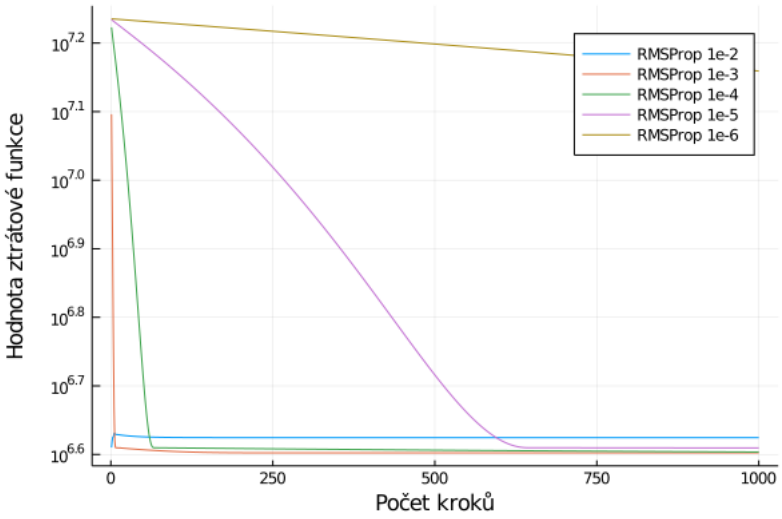
Jako počáteční parametry zvolíme $(p_1, p_2) = (27., 0.2)$. Postup bude stejný jako v předchozím případě. Rozmezí kroků jsem vybral takové, aby byl vidět přechod od příliš velkého kroku k naopak až moc malému.

Výběr metody pro 2D fyzikální model Bx				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	10^{-2}	1000	6.82	51.5
	10^{-3}	1000	6.82	51.5
	10^{-4}	1000	6.96	49.0
	10^{-5}	1000	7.05	47.7
	10^{-6}	1000	10.1	81.7
RMSProp	10^{-2}	1000	6.80	59.0
	10^{-3}	1000	6.84	51.2
	10^{-4}	1000	6.91	46.8
	10^{-5}	1000	7.05	47.7
	10^{-6}	1000	10.1	81.7

V tomto případě vidíme, že nejlépe našla parametry metoda RMSProp s krokem 10^{-4}. Hodnota ztrátové funkce neklesla pod 4e6, výpočet trval cca 70 sekund.



Obrázek 3.4: Porovnání konvergence metody ADAM pro Bx



Obrázek 3.5: Porovnání konvergence metody RMSProp pro Bx

Na obrázcích výše můžeme vidět, jakým způsobem se obě metody blížily k minimu ztrátové funkce. Metoda ADAM i RMSProp na tom byly podobně.

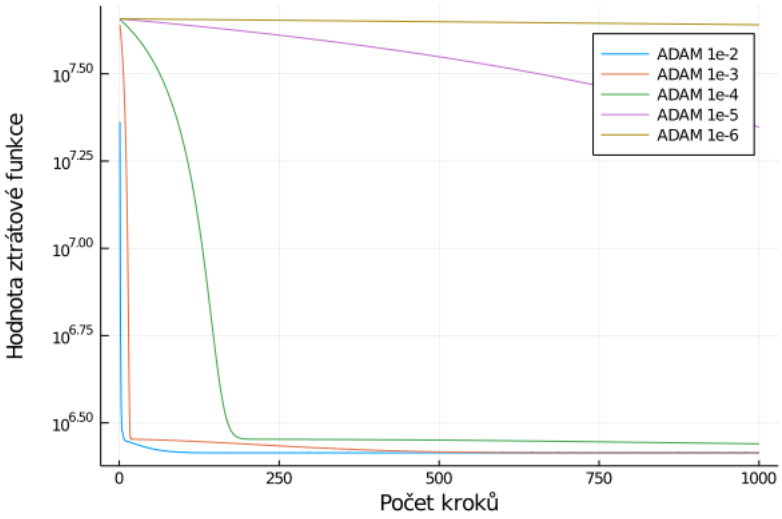
Složku By budeme obdobně aproximovat pomocí

$$G(x, y) = p_1 \frac{x}{(x^2 + y^2 + p_2)^2}$$

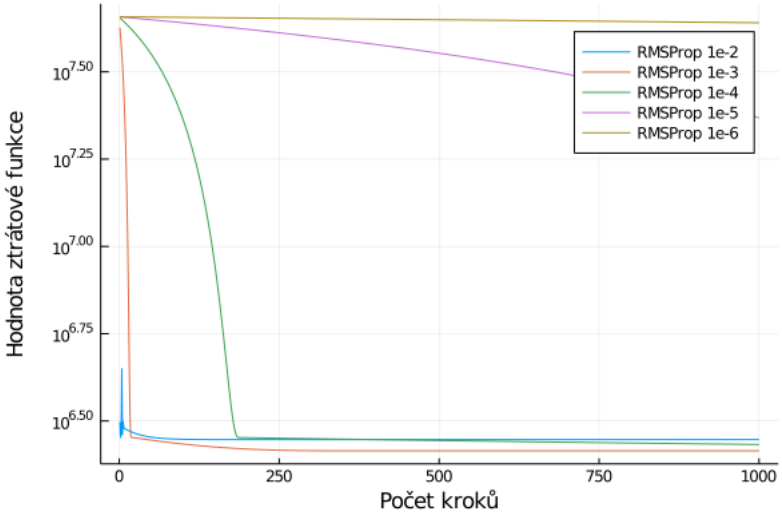
Jako počáteční parametry nyní zvolíme $(p_1, p_2) = (21., 0.3)$ opět pomocí MATLABu.

Výběr metody pro 2D fyzikální model By				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	10^{-2}	1000	5.34	46.7
	10^{-3}	1000	5.33	47.3
	10^{-4}	1000	4.90	53.1
	10^{-5}	1000	10.1	97.9
	10^{-6}	1000	14.0	134
RMSProp	10^{-2}	1000	5.44	58.2
	10^{-3}	1000	5.33	47.7
	10^{-4}	1000	4.95	52.3
	10^{-5}	1000	10.3	100
	10^{-6}	1000	14.0	134

Nyní nejlépe performovala metoda ADAM s délkou kroku 10^{-4}. Čas výpočtiu byl přibližně stejný jako v předchozí optimalizaci. Ztrátová funkce v tomto případě dosahovala minima v 2,3e6. Z obrázků níže vidíme, že zde byl oproti předešlé optimalizaci krok 1e-5 příliš velký na to, aby v požadovaném maximálním počtu kroků zkonvergoval k lokálnímu minimu.



Obrázek 3.6: Porovnání konvergence metody ADAM pro By



Obrázek 3.7: Porovnání konvergence metody RMSProp pro By

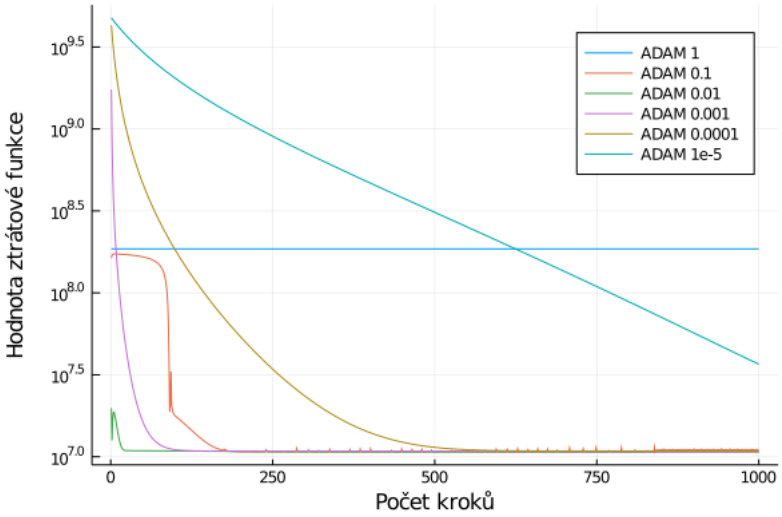
Funkce fitující Bz se bude lišit tím, že v čitateli nám odpadne proměnná.

$$H(x, y) = p_1 \frac{1}{(x^2 + y^2 + p_2)^2}$$

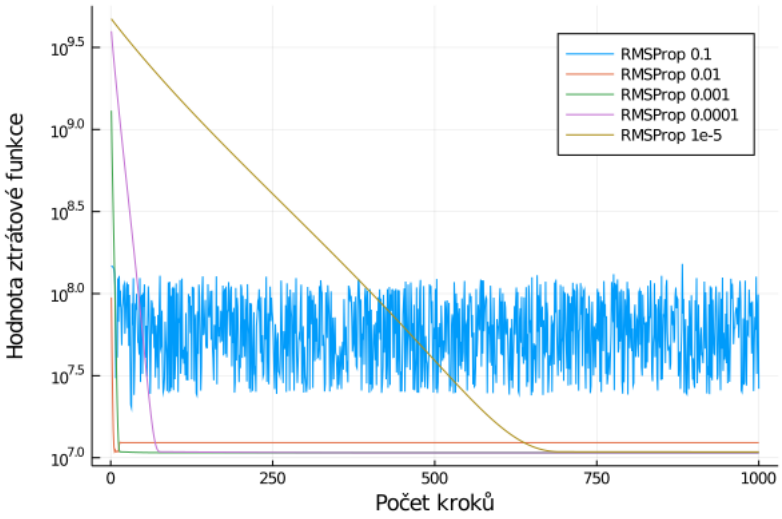
Počáteční hodnoty nastavíme na $(p_1, p_2) = (-4.1, 0.03)$

Výběr metody pro 2D fyzikální model Bz				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	1	1000	23.0	331
	0.1	1000	12.0	118
	0.01	1000	12.0	95.9
	0.001	1000	12.0	95.3
	0.0001	1000	12.3	87.6
	10^{-5}	1000	15.4	300
RMSProp	0.1	1000	19.2	48.7
	0.01	1000	12.5	68.6
	0.001	1000	12.0	100
	0.0001	1000	12.0	96.3
	10^{-5}	1000	12.3	86.8

Ztrátová funkce dosáhla minima 1e7 v čase 70 sekund. Níže můžeme pozorovat mírné oscilce metody ADAM 0.1. Pro stejný krok však RMSProp osciloval mnohem více.



Obrázek 3.8: Porovnání konvergence metody ADAM pro Bz



Obrázek 3.9: Porovnání konvergence metody RMSProp pro Bz

3.5 Kombinovaný model 2D

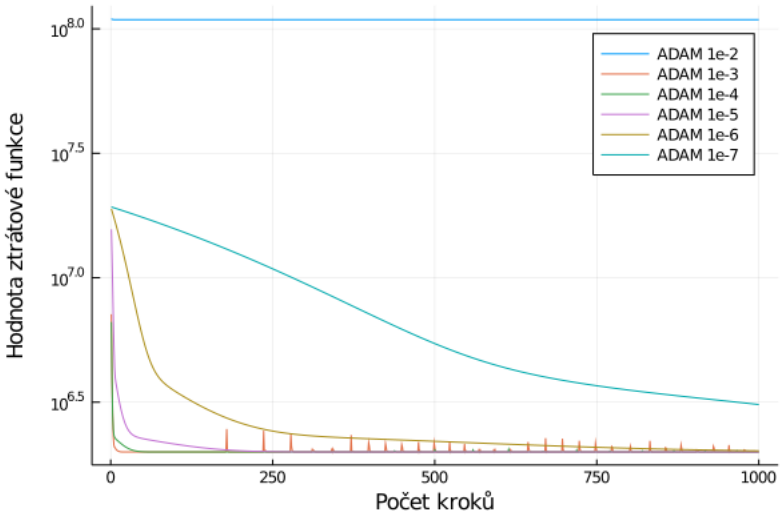
Abychom dosáhli lepšího fitu, tj. menších odchylek, přidáme k polynomu $F(x, y)$ volné parametry a další proměnné x, y ve vyšších mocninách. Zvolíme funkci F jako

$$F(x, y) = p_1 \frac{x}{(p_2x^4 + p_3y^4 + p_4x^2 + p_5y^2 + p_6)^2}$$

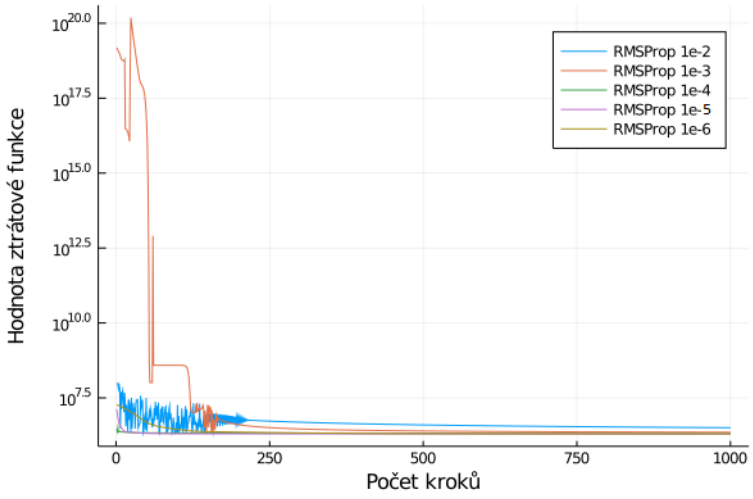
Experimentálně ověříme, jaká gradientní metoda bude nejlepší pro tuto funkci, jejíž vstupové parametry do optimalizace jsme volili $(p_1, ..., p_6) = (0.0036, -0.0127, -0.0154, -0.0073, -0.0088, -0.0017)$ pomocí MatLabu. V počátečních hodnotách máme $MEA = 6.8$ a $MAX = 37.3$. Podmínku ukončení algoritmu jsem se rozhodl zavést pomocí maximálního počtu kroků 1000.

Výběr metody pro 2D kombinovaný model Bx				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	10^{-2}	1000	28.9	190
	10^{-3}	1000	5.68	31.5
	10^{-4}	1000	5.68	30.9
	10^{-5}	1000	5.68	30.9
	10^{-6}	1000	5.68	29
	10^{-7}	1000	6.62	28.8
RMSProp	10^{-2}	1000	6.5	34.4
	10^{-3}	1000	5.93	25.5
	10^{-4}	1000	5.68	33.5
	10^{-5}	1000	5.69	30.3
	10^{-6}	1000	5.68	30.8

Z výsledků můžeme vypozorovat, že obě metody fungovaly špatně při kroku s velikostí 1, jelikož byl až moc velký. Analogicky pak to samé platí při příliš malých krocích. Celkově si metoda ADAM vedla lépe než RMSProp dosahovala lepších výsledků a s průměrným časem výpočtu okolo 60 sekund byla o pár sekund i rychlejší. Na obrázcích níže můžeme porovnat, jakým způsobem se obě metody přibližovaly k lokálnímu minimu.



Obrázek 3.10: Porovnání konvergence metody ADAM u Bx



Obrázek 3.11: Porovnání konvergence metody RMSProp u Bx

Můžeme vidět, že metoda ADAM měla hladší sestup k lokálnímu minimu, zatímco RMSProp u větších délek kroku měla tendenci oscilovat. Jako finální parametry pro náš model jsme vybrali $(p_1, ..., p_6) = (6.5e-3, -4, 9e-3, 0.28, -0.12, -0.14, -0.065, -0.078, -0.016)$, které nám poskytla metoda ADAM s krokem 10^{-6}.

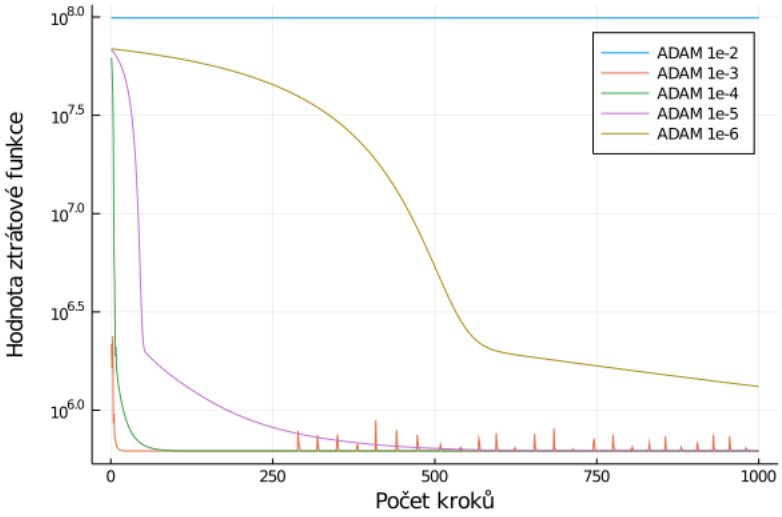
Analogicky jsme upravili i funkci $G(x, y)$ pro fit By složky mag. pole.

$$G(x, y) = p_1 \frac{y}{(p_2 x^4 + p_3 y^4 + p_4 x^2 + p_5 y^2 + p_6)^2}$$

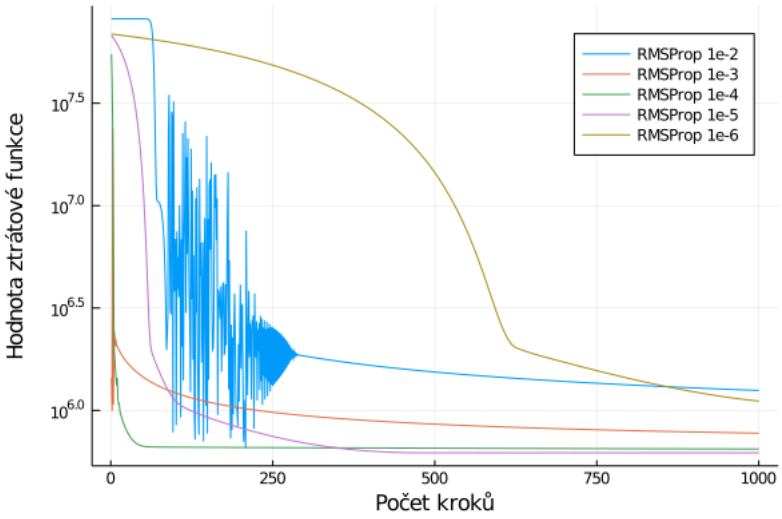
Počáteční parametry funkce G jsme volili jako $(p_1, ..., p_6) = (0.01, 0.001, -0.03, -0.04, -0.02, -0.01)$. V těchto parametrech obrdžíme hodnoty $MEA = 20.8$ a $MAX = 157$

Výběr metody pro 2D kombinovaný model By				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	10^{-2}	1000	26.7	179
	10^{-3}	1000	2.48	24.3
	10^{-4}	1000	2.48	24.3
	10^{-5}	1000	2.48	24.3
	10^{-6}	1000	3.29	31.6
RMSProp	10^{-2}	1000	3.14	38.4
	10^{-3}	1000	2.60	29.8
	10^{-4}	1000	2.54	22.2
	10^{-5}	1000	2.47	24.6
	10^{-6}	1000	3.27	30.9

Zde opět vidíme že největší a nejmenší krok u obou metod již nedává tak dobré výsledky. Nyní jako konečné parametry můžeme vybrat ty, které nám poskytne metoda RMSProp s krokem 10^{-5} $(p_1, ..., p_6) = (9.8e{-}5, -0.0038, 0.0162, -0.0352, -0.0292, -0.01825, -0.0161, -0.0036)$. Obě metody měly průměrný čas výpočtu ve všech krocích 55 sekund. Hodnoty ztrátové funkce se nedostaly pod 620 000.



Obrázek 3.12: Porovnání konvergence metody ADAM u By



Obrázek 3.13: Porovnání konvergence metody RMSProp u By

Z obrázků pozorujeme podobnou tendenci konvergence jako u složky Bx.

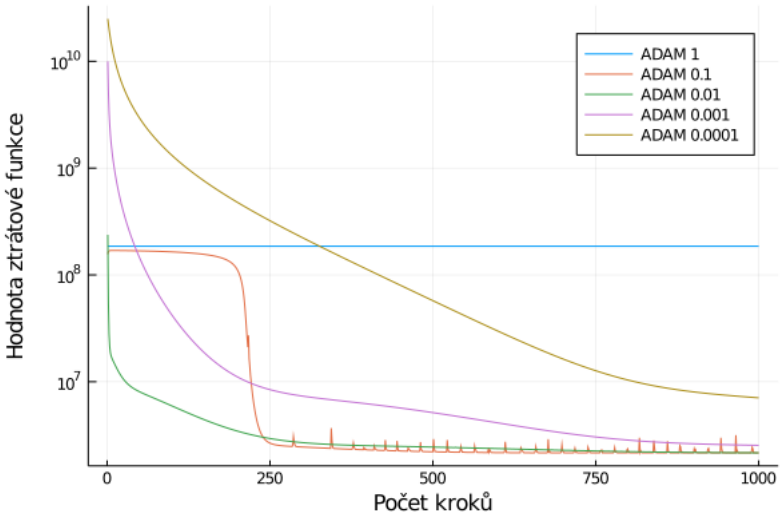
Stejným způsobem vypočítáme parametry i upravené funkce $H(x, y)$ pro složku Bz

$$H(x,y) = p_1 \frac{1}{(p_2x^4 + p_3y^4 + p_4x^2 + p_5y^2 + p_6)^2}$$

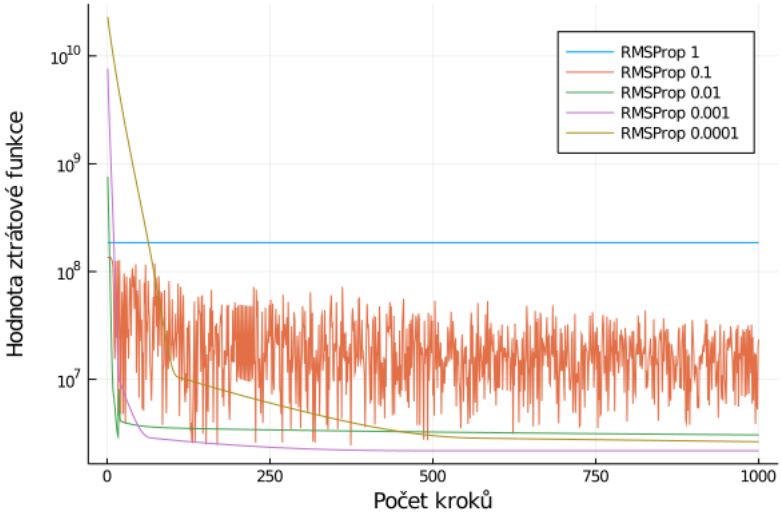
Počáteční parametry zvolíme $(p_1, ..., p_6) = (-5, 8, 3, 0.1, 0.1, 0.03)$ ve kterých měříme hodnoty $MEA = 201$ a $MAX = 5226$. Experimentálně zvolíme metodu, která nám dá nejlepší výsledné parametry funkce H.

Výběr metody pro 2D kombinovaný model Bz				
Název gradientní metody	délka kroku	počet kroků	MEA	MAX
ADAM	1	1000	23	331
	0.1	1000	5.76	21.9
	0.01	1000	5.77	21.8
	0.001	1000	6.21	21.8
	0.0001	1000	8.95	44.2
RMSProp	1	1000	23	331
	0.1	1000	11.5	124
	0.01	1000	6.42	33.3
	0.001	1000	5.79	21.8
	0.0001	1000	6.3	22.4

Zde žádná z metod nedosáhla hodnoty ztrátové funkce menší než $2.16e - 6$. Průměrný čas výpočtu se pohyboval okolo jedné minuty. Jako nejlepší výsledek můžeme zvolit ten, který nám poskytne metoda ADAM s krokem 0.1 $(p_1, ..., p_6) = (-8.11, 10.46, 10.8, 0.349, 0.414, 0.155)$.



Obrázek 3.14: Porovnání konvergence metody ADAM u Bz



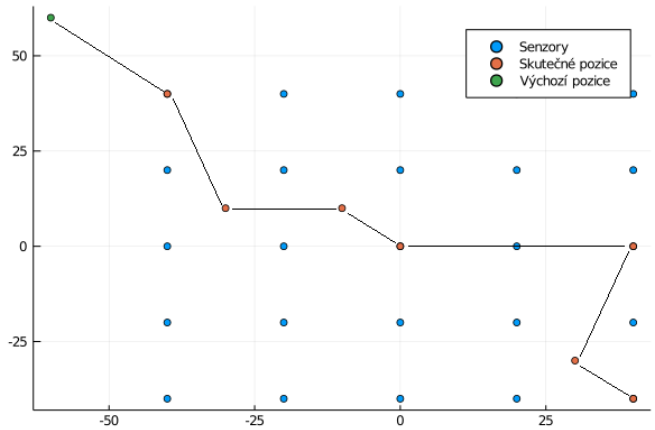
Obrázek 3.15: Porovnání konvergence metody RMSProp u Bz

Porovnáme-li konvergenci RMSProp 0.1 u fyzikálního modelu a kombinovaného, vidíme stejný vzor oscilace.

Kapitola 4

Lokalizace

Máme hotovou volbu modelu, to jest pomocí gradientních metod jsme našli parametry matematických funkcí, které charakterizují působení magnetického pole v okolí našeho robota. Pomocí našeho modelu se teď budeme pokoušet odhadnout pozici robota. Vygenerujeme si syntetická data, která budou představovat mřížku senzorů. Umístíme robota do určité pozice. Každý ze senzorů bude dostávat informaci z magnetického pole o jeho pozici. Optimalizační metodou se pomocí již určeného modelu budeme snažit dospět ke skutečné poloze. Nyní již tedy proměnnou v modelu nepředstavují parametry funkce, ale poloha $[x, y]$. Experiment provedeme na mřížce 5x5 s rovnoměrně rozloženými 25 senzory. Vzdálenost mezi senzory nastavíme na 20 pixelů. Budeme simulovat pohyb robota po určité trajektorii a zkoumat, jak moc se skutečná poloha bude lišit od námi nalezené. Zajímat nás především bude rozdíl mezi fyzikálním a kombinovaným modelem tj. jestli námi určený model bude pro lokalizaci vhodnější. Do pokusu dále zahrneme i šum. Vzdálenost mezi senzorem a polohou robota pozměníme pomocí náhodně vygenerované gaussovské odchylky, kterou přičteme ke snímanému vektoru vzdálenosti. Je-li tedy vektor $\boldsymbol{d}$ rozdíl skutečné polohy mikrorobota a pozice senzoru na mřížce, upravíme ho jako $\boldsymbol{d} + \sigma * \boldsymbol{e}$, kde σ je skalár určující velikost zašumění a $\boldsymbol{e}$ je vektor, jehož složky jsou čísla od 0 do 1 vygenerované pomocí normálního rozdělení. Naměřená data z experimentu musíme interpolovat, abychom dostali výsledek pro jakýkoliv vektor $\boldsymbol{d}$. Ztrátová funkce nyní představuje součet všech tří předešlých ztrátových funkcí s tím rozdílem, že data nám tentokrát poskytuje vlastní vygenerovaná mřížka senzorů. Trajektorie miktorobota je zobrazena níže.



Obrázek 4.1: Trasa robota

Jako první skutečnou pozici robota určíme $a = [-40, 40]$ a výchozí pozici, kterou zadáme do modelu určíme $b = [-60, 60]$, jako gradientní metodu jsou použil ADAM s krokem 0.01 a počet kroků jsem nastavil na 500. Během optimalizačního procesu se nám tedy bude hodnota $[-60, 60]$ blížit k $[-40, 40]$. V dalším kroku nastavíme skutečnou polohu $c = [-30, 10]$ a výchozí b. Proces opakujeme až do posledního bodu. Jako chybu budeme uvažovat eukleidovskou vzdálenost mezi skutečnou pozicí a pozicí, ke které jsme optimalizačním procesem dospěli, tedy

$$Chyba = \sqrt{(x_s - x_n)^2 + (y_s - y_n)^2}$$

kde x_s, y_s značí souřadnice skutečné polohy a x_n, y_n značí souřadnice námi nalezené polohy. Nejdříve se podíváme na výsledky pro oba modely bez zahrnutí šumu.

Odchylka při lokalizaci bez zašumění						
			Kombinovaný model		Fyzikální model	
Výchozí pozice	Skutečná pozice	Metoda	Nalezená pozice	*Chyba*	Nalezená pozice	*Chyba*
$[-60, 60]$	$[-40, 40]$	ADAM 0.01/1e-4	$[-44.3, 43.5]$	5.54	$[-59.6, 60.5]$	28.3
$[-40, 40]$	$[-30, 10]$	ADAM 0.1	$[-42.9, 18.1]$	15.3	$[-48.6, 23.9]$	23.2
$[-30, 10]$	$[-10, 10]$	ADAM 0.1	$[-16.2, 16.2]$	8.8	$[-16.7, 16.6]$	9.41
$[-10, 10]$	$[0, 0]$	ADAM 0.1	$[-1.6, 1.4]$	2.13	$[-0.5, 0.3]$	0.64
$[0, 0]$	$[40, 0]$	ADAM 0.01	$[38.8, 2.67]$	2.93	$[38.8, 2.7]$	2.96
$[40, 0]$	$[30, -30]$	ADAM 0.01	$[37.0, -34.9]$	8.59	$[37.1, -36.6]$	9.69
$[30, -30]$	$[40, -40]$	ADAM 0.01	$[42.4, -42.0]$	3.18	$[41.5, -42.9]$	3.32

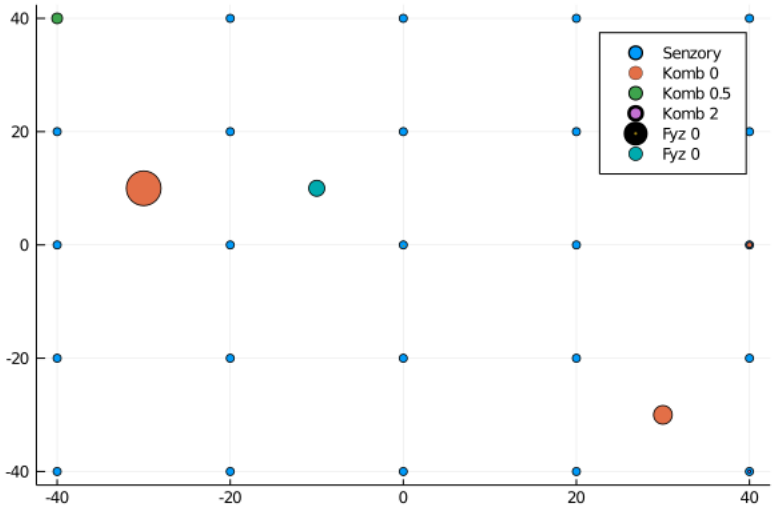
Při každém posouvání robota jsme museli stanovit příslušnou metodu optimalizace, která nejképe konvergovala.Průměrná chyba kombinovaného modelu bez šumu byla 6.64 pixelů, zatímco fyzikálního 11.1. Největší ochylky od přesné pozice jsme měřili na krajích mřížky a mezi senzory, zatímco když se robot nacházel uprostřed pole senzorů, chyba byla nejmenší. To můžeme vysvětlit tím, že s větší vzdáleností mikrorobota od senzorů je také signál snímaný senzory slabší. Chyby jako takové jsou způsobeny nepřestnostmi modelů a interpolací dat.

Odchylka při lokalizaci s $\sigma = 0.5$						
			Kombinovaný model		Fyzikální model	
Výchozí pozice	Skutečná pozice	Metoda	Nalezená pozice	*Chyba*	Nalezená pozice	*Chyba*
$[-60, 60]$	$[-40, 40]$	ADAM 0.01/1e-4	$[-44.2, 43.2]$	5.26	$[-59.5, 60.5]$	28.3
$[-40, 40]$	$[-30, 10]$	ADAM 0.1	$[-42.8, 18.5]$	15.4	$[-47.8, 22.6]$	21.8
$[-30, 10]$	$[-10, 10]$	ADAM 0.1	$[-16.2, 15.9]$	8.58	$[-15.4, 15.2]$	7.47
$[-10, 10]$	$[0, 0]$	ADAM 0.1	$[-1.49, 1.1]$	1.86	$[-1.2, 1.0]$	1.55
$[0, 0]$	$[40, 0]$	ADAM 0.01	$[38.5, 3.0]$	3.42	$[38.9, 1.6]$	1.99
$[40, 0]$	$[30, -30]$	ADAM 0.01	$[37.2, -35.0]$	8.78	$[36.3, -36.4]$	9.00
$[30, -30]$	$[40, -40]$	ADAM 0.01	$[42.4, -42.0]$	3.13	$[41.8, -43.0]$	3.52

Optimalizační metody jsme zachovali stejné jako v předešlém měření. Podobný trend můžeme pozorovat i při přidání slabého šumu s tím rozdílem, že šum v polovině případů nalezenou pozici posunul blíže ke skutečné pozici robota, v druhé polovině naopak.

Odchylka při lokalizaci s $\sigma = 2$						
			Kombinovaný model		Fyzikální model	
Výchozí pozice	Skutečná pozice	Metoda	Nalezená pozice	*Chyba*	Nalezená pozice	*Chyba*
$[-60, 60]$	$[-40, 40]$	ADAM 0.01/1e-4	$[-42.0, 42.4]$	3.10	$[-59.5, 60.5]$	28.3
$[-40, 40]$	$[-30, 10]$	ADAM 0.1	$[-44.3, 18.0]$	16.4	$[-50.3, 26.3]$	26.0
$[-30, 10]$	$[-10, 10]$	ADAM 0.1	$[-18.9, 17.5]$	11.6	$[-22.5, 22.5]$	17.7
$[-10, 10]$	$[0, 0]$	ADAM 0.1	$[-4.3, 2.0]$	4.78	$[4.6, -5.2]$	6.93
$[0, 0]$	$[40, 0]$	ADAM 0.01	$[38.7, 2.2]$	2.59	$[38.9, 2.2]$	2.42
$[40, 0]$	$[30, -30]$	ADAM 0.01	$[37.3, -36.5]$	9.77	$[36.0, -36.9]$	9.17
$[30, -30]$	$[40, -40]$	ADAM 0.01	$[40.6, -41.9]$	1.99	$[44.3, -46.6]$	7.82

Metody pro optimalizaci jsme ponechali stejné. Obdobné výsledky vidíme i při větším šumu. Zde již ale pozorujeme větší odchylku a dá se očekávát že při zvětšujícím se šumu by výsledky byly horší. Na obrázku níže vidíme chyby modelů s příslušným šumem, které v měřených bodech dosahovali nejlepších výsledků. Velikost chyby je přímo úměrná té naměřené. Kombinovaný model byl přesnější než fyzikální, až na výjimku dvou bodů, kde selhal o 1.5 pixelu.



Obrázek 4.2: Nejmenší chyby v pozorovaných bodech

Nyní přidáme k k vektoru $\boldsymbol{d}$ snímajícímu vzdálenost robota od senzoru konstantní odchylku $[1.0, 0.3]$, který má výchylku v jedné složce větší než ve druhé a budeme pozorovat, jak to ovlivní nalezenou polohu od skutečné pro x a y. Šum v tomto případě již uvažovat nebudeme. Konstantní odchylku ještě vynásobíme konstantou c dostatečně velkou tak, aby se to projevilo ve výsledcích. Pokud zachováme optimalizační metody stejné, jako při měření bez šumu v prvním případě, obdržíme výsledky:

Odchylka při lokalizaci s konstantní odchylkou při $c = 20$						
			Kombinovaný model		Fyzikální model	
Výchozí pozice	Skutečná pozice	Metoda	Nalezená pozice	*Chyba*	Nalezená pozice	*Chyba*
$[-60, 60]$	$[-40, 40]$	ADAM 0.01/1e-4	$[-94.3, 104.1]$	83.9	$[-60.5, 60.5]$	29.0
$[-40, 40]$	$[-30, 10]$	ADAM 0.1	$[-61.4, 12.4]$	31.5	$[-67.2, 19.8]$	38.5
$[-30, 10]$	$[-10, 10]$	ADAM 0.1	$[-43.8, 12.7]$	33.9	$[-50.2, 19.0]$	41.2
$[-10, 10]$	$[0, 0]$	ADAM 0.1	$[-41.0, 4.8]$	41.3	$[-56.0, 18.6]$	59.1
$[0, 0]$	$[40, 0]$	ADAM 0.01	$[30.2, -13.0]$	16.3	$[35.4, -18.4]$	19.0
$[40, 0]$	$[30, -30]$	ADAM 0.01	$[85.2, -39.5]$	56.0	$[78.9, -42.9]$	50.6
$[30, -30]$	$[40, -40]$	ADAM 0.01	$[25.4, -53.8]$	20.2	$[28.7, -54.7]$	18.5

Na první pohled jsou odchylky větší. Pozorujeme, že větší konstantní odchylka pro první složku vektoru vzdálenosti způsobila i větší odchylku pro x v nalezené poloze. Měření provedeme ještě jednou, tentokrát s pozměněnou metodou optimalizace pro každou dílčí trasu zvlášť pro kombinovaný i fyzikální model, abychom minimalizovali chybu.

Odchylka při lokalizaci s konstantní odchylkou při $c = 20$						
			Kombinovaný model		Fyzikální model	
Výchozí pozice	Skutečná pozice	Metoda	Nalezená pozice	*Chyba*	Nalezená pozice	*Chyba*
$[-60, 60]$	$[-40, 40]$	ADAM 10^{-7}	$[-60.0, 60.0]$	28.3	$[-60.0, 60.0]$	28.3
$[-40, 40]$	$[-30, 10]$	ADAM 10^{-3}/0.1	$[-44.9, 34.1]$	28.4	$[-44.9, 34.2]$	28.5
$[-30, 10]$	$[-10, 10]$	ADAM 10^{-7}	$[-30.0, 9.9]$	20.0	$[-30.4, 29.5]$	28.3
$[-10, 10]$	$[0, 0]$	ADAM 10^{-7}	$[-10.0, 9.9]$	14.1	$[-10.0, 9.9]$	14.1
$[0, 0]$	$[40, 0]$	ADAM 0.01	$[30.2, -13.0]$	16.3	$[35.4, -18.4]$	19.0
$[40, 0]$	$[30, -30]$	ADAM 10/0.1	$[14.9, -43.3]$	20.1	$[18.9, -48.5]$	21.6
$[30, -30]$	$[40, -40]$	ADAM 10^{-7}	$[29.9, -30.0]$	14.1	$[29.5, -30.4]$	14.2

Snaha zmenšit chybu v tomto případě znamenala nastavit krok optimalizačních metod natolik malý, že se robot nevzdálil z výchozí pozoce. Překvapivý výsledek vidíme u kombinované metody při posunu do posledního bodu, kdy byl lepší ADAM s nezvykle velkým krokem 10.

Závěr

V této bakalářské práci jsme se zabývali optimalizací, nejprve jsme si vysvětlili, co tento pojem znamená, ukázali jsme si několik metod optimalizace a zaměřili jsme se na spádové metody. Seznámili jsme se s daty získanými z experimentu podle [4]. Optimalizační techniky jsme aplikovali na případ lokalizace mikrorobota nacházejícím se v magnetickém poli. Popasovali jsme se s volbou modelu magnetického pole, ve kterém je náš robot umístěn. Při modelování jsme vycházeli z teorie aproximace polynomem, z fyzikální podstaty problému a nakonec zvolili náš vlastní přístup pro přesnější výsledky. Postupovali jsme krok za krokem od jednorozměrného případu, který byl intuitivně lehčí. Po nalezení uspokojivě přesného modelu pro každou dvourozměrnou složku magnetického pole jsme řešili úlohu inverzní, to jest nalezení polohy mikrorobota pomocí našeho modelu a měření ze synteticky vygenerovaných senzorů. Při této lokalizaci jsme ukázali přesnost obou modelů a jejich citlivost na odchylky způsobené šumem a konstantním vychýlením. Potvrdili jsme tím, že náš kombinovaný model lépe aproximoval data.

Problematika lokalizace pomocí optimalizačních metod mě velice zaujala, v budoucnu bych se jí chtěl věnovat intenzivněji ve složitějších případech.

Literatura

[1] Peter Deuflhard. *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, volume 35. Springer Science & Business Media, 2011.

[2] ZDENĚK DOSTÁL and P Bermelijski. Metody optimalizace. *Technická univerzita Ostrava*, 2012.

[3] Eugene Isaacson and Herbert Bishop Keller. *Analysis of numerical methods*. Courier Corporation, 2012.

[4] M. Juřík, J. Kuthan, J. Vlček, and F. Mach. Positioning uncertainty reduction of magnetically guided actuation on planar surfaces. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1772–1778, 2019.

[5] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[6] Valter Pasku, Alessio De Angelis, Guido De Angelis, Darmindra D Arumugam, Marco Dionigi, Paolo Carbone, Antonio Moschitta, and David S Ricketts. Magnetic field-based positioning systems. *IEEE Communications Surveys & Tutorials*, 19(3):2003–2017, 2017.

[7] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.