VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

OPTIMALIZACE ROZMĚRŮ ČTYŘČLENNÉHO MECHANISMU PRO DOSAŽENÍ POŽADOVANÉ TRAJEKTORIE

OPTIMIZATION OF FOUR-BAR MECHANISM GEOMETRY FOR THE PREDESCRIBED TRAJECTORY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE — LUKÁŠ KORYTÁR

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. ROBERT GREPL, Ph.D.

SUPERVISOR

BRNO 2015

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky
Akademický rok: 2014/15

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Lukáš Korytár

který/která studuje v **bakalářském studijním programu**

obor: **Základy strojního inženýrství (2341R006)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Optimalizace rozměrů čtyřčlenného mechanismu pro dosažení požadované trajektorie

v anglickém jazyce:

Optimization of four-bar mechanism geometry for the predescribed trajectory

Stručná charakteristika problematiky úkolu:

Tato práce se bude zabývat optimalizační úlohou hledání parametrů mechanismu s cílem realizovat požadovanou trajektorii koncového efektoru při zadané struktuře mechanismu a pohybu akčního členu. Základním příkladem pro studium bude čtyřčlenný mechanismus, který umožňuje realizaci široké škály trajektorií. Pro optimalizaci budou využity metody dostupné v prostředí MATLAB.

Cíle bakalářské práce:

1) Seznamte se s metodou řešení přímé a inverzní úlohy kinematiky pomocí matic.
2) Vykreslete různé výsledné trajektorie pracovního bodu čtyřčlenného mechanismu v závislosti na rozměrech mechanismu a poloze pracovního bodu.
3) Vyhledejte a prostudujte literaturu zabývající se optimalizací mechanismu s ohledem na výslednou trajektorii. Analyzujte použitou kriteriální funkci, optimalizační algoritmy a aplikační oblast.
4) Navrhněte a implementujte kriteriální funkci pro porovnání aktuální trajektorie vypočítané kinematickým modelem a zadané cílové trajektorie.
5) Pro čtyřčlenný mechanismus a zadanou realizovatelnou trajektorii testujte konvergenci optimalizační metody k cílové trajektorii. Zhodnoťte vlastnosti této základní úlohy s ohledem na použitou optimalizační metodu (použijte také některý z tzv. evolučních algoritmů (GA, Diferenciální evoluce), typ kriteriální funkce a volbu počátečního odhadu parametrů (sledujte citlivost konvergence na různý počáteční odhad).

Seznam odborné literatury:

- J. A. Cabrera, A. Simon, M. Prado, (2002) Optimal synthesis of mechanisms with genetic algorithms, Mechanism and Machine Theory 37, 1165-1177
- Valášek, M.: Mechatronika, Vydavatelství ČVUT 1995
- Grepl, R.: Modelování mechatronických systémů v Matlab/SimMechanics, BEN, 2007
- Grepl, R. Kinematika a dynamika mechatronických systémů, Nakladatelství CERM, ISBN: 978-80-214-3530-8

Vedoucí bakalářské práce: doc. Ing. Robert Grepl, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2014/15.

V Brně, dne 27. 11. 2014

L.S.

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA STROJNÍHO INŽENÝRSTVÍ DĚKANÁT

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan

Abstrakt

Práce se zabývá optimalizací délek ramen čtyřčlenného mechanismu typu RRRR pomocí maticové kinematiky. Srovnává a popisuje metody k tomu dostupné v prostředí MATLAB.

Dále je navržen optimalizační algoritmus, využívající funkci *fmincon,* pro niž realizuje prvotní odhad algoritmus Monte-Carlo. Pro navržený algoritmus jsou definovány základní požadavky na trajektorii, kterou je možno přesně kopírovat.

Abstract

This thesis deals with the optimization of arm lengths of four-bar RRRR mechanism using matrix kinematics. It compares and describes the methods, which are available in MATLAB software environment for this purpose.

Consequently, an optimization algorithm using fmincon function is designed for which Monte-Carlo algorithm performs the first estimates. Elementary requirements for the proposed algorithm are devised in order to accurately duplicate the predescribed trajectory.

Klíčová slova

Čtyřčlenný mechanismus, MATLAB *optimization toolbox*, Genetický algoritmus, Metoda Monte-Carlo, Maticová kinematika

Keywords

Four-bar mechanism, MATLAB *optimization toolbox*, Genetic algorithm, Monte-Carlo method, Matrix kinematics

Bibliografická citace

KORYTÁR, L. *Optimalizace rozměrů čtyřčlenného mechanismu pro dosažení požadované trajektorie.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2015. 36 s. Vedoucí bakalářské práce doc. Ing. Robert Grepl, Ph.D.

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval sám na základě svých znalostí, snahy, dovedností, rad a pokynů vedoucího bakalářské práce. Uvedl jsem veškeré použité podklady a literaturu.

V Brně dne: ……. …….

Lukáš Korytár

Obsah

1. Úvod

Stroje dnes využívají nepřeberné množství mechanismů, které jim zajišťují převod translačních či rotačních pohybů na složitější křivky. Jedním z nich je čtyřčlenný mechanismus. Ten převádí rotační pohyb krajních členů na obecný rovinný pohyb prostředního, čímž nám dovoluje dosáhnout rozličných trajektorií. Můžeme jej najít v zavěšení zadního kola horského bicyklu, mechanismu automobilových stěračů, kopírovacích zařízeních a zavěšení kol automobilu.

Tato práce se zabývá optimalizací rozměrů čtyřčlenného mechanismu pro dosažení námi zvolené trajektorie. Úloha je zadána ve 2D prostoru a řešena maticovou kinematikou. Pro optimalizaci budou využity metody dostupné v prostředí MATLAB, které dále srovnám z hlediska rychlosti i přesnosti.

Úspěšně syntetizovaný mechanismus musí splňovat proporční vyváženost, kopírovat křivku v zadané toleranci a otáčet se v zadaném intervalu akčního úhlu. Zde použité algoritmy jsou schopny optimalizovat pouze pro určitou množinu křivek na daném intervalu. Proto budou navrženy omezení a kritéria aplikačních oblastí, pro něž řešení konverguje.

2. Rešerše

2.1. Typy čtyřčlenných mechanismů a jejich popis

Čtyřčlenný mechanismus je nejjednodušší pohyblivý uzavřený mechanismus. Skládá se ze čtyř těles spojených rotační vazbou. Dvě protilehlá tělesa vykonávají rotační pohyb a jsou upevněné k základovému tělesu, takzvanému rámu. Zbylé, čtvrté, spojuje dva rotující členy a koná obecný rovinný pohyb.

Dle Grashofových podmínek dělíme čtyřčlenné mechanismy do pěti kategorií[1]:

Případ	Kritérium	Nejkratší rameno	Kategorie
1	$s + l < p + q$	Rám	Dvouklikový
2	$s + l < p + q$	Rotující člen	Klikovahadlový
3	$s + l < p + q$	Spojující člen	Dvouvahadlový
4	$s + l = p + q$	Kterékoliv	Paralelogram
5	$s + l > p + q$	Kterékoliv	Trojvahadlový

s … nejkratší rameno
l … nejdelší rameno
p … délka jednoho ze středně dlouhých ramen
q … délka jednoho ze středně dlouhých ramen

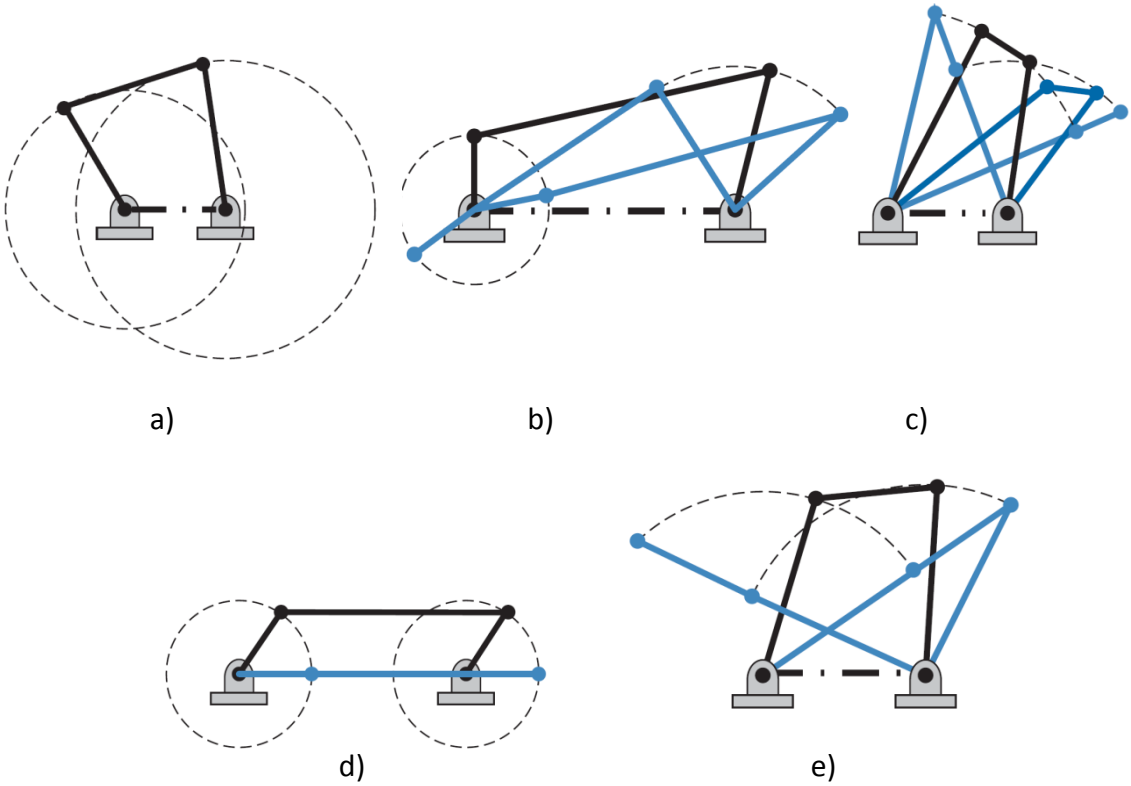
Dvouklikový mechanismus: Jeho nejkratším ramenem je rám. Pokud plynule rotuje jedno z jeho upevněných ramen, rotuje i to druhé. Rotující členy jsou schopny dosáhnout otočení o plných 360°.

Klikovahadlový mechanismus: Nejkratším ramenem je rotující člen (klika). Pokud jím začneme spojitě otáčet, vahadlový člen bude oscilovat mezi svými krajními polohami (vyznačeny modře na obr. 2.1-b). Příkladem takovéhoto mechanismu je například stěrač čelního skla automobilu. Zatímco motor rotuje s klikou, stěrač se pohybuje tam a zpět.

Dvouvahadlový mechanismus: Zde je oběma upevněným členům umožněn pohyb pouze v omezeném akčním úhlu. Naopak nejkratší spojovací člen smí konat rotaci v plném rozsahu.

Paralelogram: Tento mechanismus má dle definice stejnou hodnotu součtu nejkratšího ramene s nejdelším jako součet obou středně dlouhých ramen. Na jeho principu pracují mechanická kopírovací zařízení nebo různé druhy stojanů či držáků.

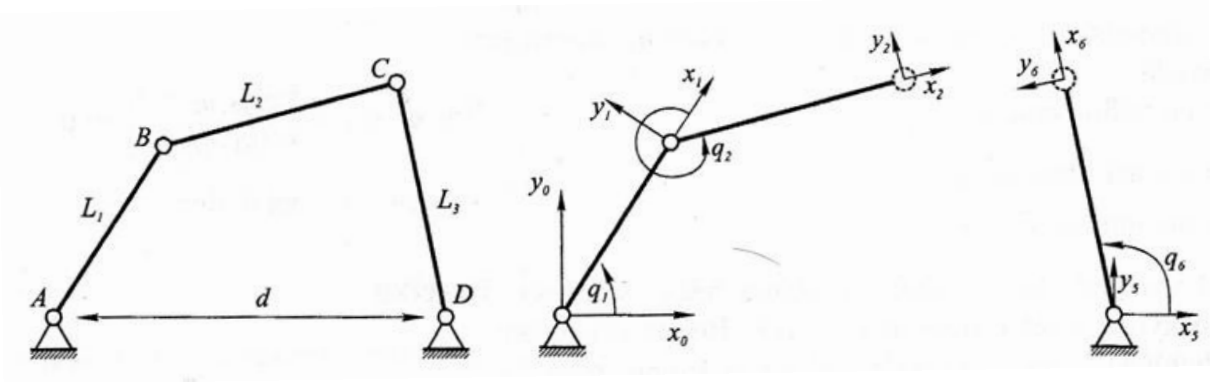
Trojvahadlový: Nemá požadavek na nejkratší rameno. Je velmi podobný dvojvahadlovému mechanismu, pouze s tím rozdílem, že jeho spojovací člen není schopen vykonat rotaci o celých 360°. Do této skupiny lze zařadit i Wattův přímovod užívaný pro zavěšení kol automobilů.



Obr. 2.1 Mechanismus a) dvouklikový; b) klikovahadlový; c) dvouvahadlový; d) paralelogram; e) trojvahadlový [1]

2.2. Řešení přímé úlohy kinematiky pomocí matic

Čtyřčlenný mechanismus má jeden stupeň volnosti. Jako parametr volíme q_1. Úhly q_2, q_6 určují polohu ramen L_2, L_3, přičemž jsou oba funkcí q_1. Nyní si zvolíme sledovaný bod P v souřadném systému „2“. Jeho polohový vektor je potom $\boldsymbol{r_2^P}$.



Obr. 2.2 Schéma čtyřčlenného mechanismu [2]

Pokud zvolíme:

$$\boldsymbol{r_2^P} = \begin{pmatrix} P_x \\ P_y \end{pmatrix} = \mathbf{0} \tag{2.1}$$

Pak dokážeme z transformačních matic vyjádřit dvě použitelné rovnice pro směry *x, y:*

$$\boldsymbol{T_{20} r_2^P - T_{50} T_{65} r_2^P} = \begin{pmatrix} L_2 c_{12} + L_1 c_1 - L_3 c_6 - d \\ L_2 s_{12} + L_1 s_1 - L_3 s_6 \\ 0 \\ 0 \end{pmatrix} = \mathbf{0} \tag{2.2}$$

$$\boldsymbol{U} = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f_1(\boldsymbol{q}) \\ f_2(\boldsymbol{q}) \end{pmatrix}, kde\ \boldsymbol{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_6 \end{pmatrix} \tag{2.3}$$

Z rovnice (2.2) plynou nelineární rovnice, z nichž nelze explicitně vyjádřit q_2, q_6. Proto je nutné k výpočtu použít iteračního procesu pomocí analytického Jakobiánu.

$$\boldsymbol{U_k} = f(\boldsymbol{q_k})$$

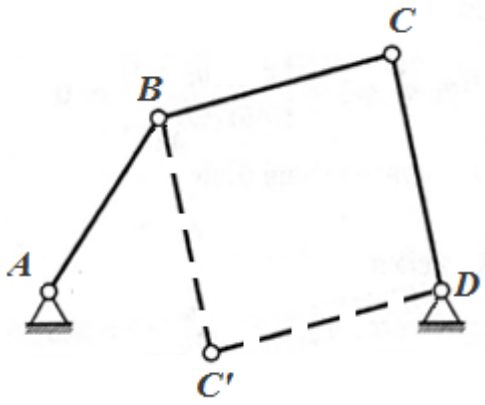
$$\boldsymbol{q_k} = \boldsymbol{q_j} + \boldsymbol{J^{-1}}(\boldsymbol{U_0} - \boldsymbol{U_k}), kde\ \boldsymbol{U_0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.4}$$

$$k = j + 1$$

Do procesu (2.4) je nutné dosadit počáteční odhad. Pokud lze mechanismus zkompletovat pro zadaný parametr q_1, konverguje vektor $\boldsymbol{U_k}$ v řádech desítek kroků pod toleranci 10^{-3} a matice $\boldsymbol{q_k}$ obsahuje polohové úhly ramen. Úloha může mít žádné, jedno nebo dvě řešení. To může vést k dezinterpretaci funkčního mechanismu. Podmínky řešitelnosti pro dané q_1 vycházejí z geometrie mechanismu:

$$\begin{aligned} L_2 + L_3 &\geq l \\ L_3 - L_2 &\leq l \\ L_2 - L_3 &\leq l \end{aligned} \tag{2.5}$$

l *je vzdálenost počátků souřadných systémů* |15|



Obr. 2.3 Nejednoznačný mechanismus pro dané q_1

Podrobné řešení kinematické úlohy je popsáno v [2].

2.3. Optimalizační algoritmy

Tyto algoritmy hledají takové délky mechanismu, aby sledovaný bod P kopíroval při zadaném rozsahu akčního úhlu q_1 co nepřesněji zadanou trajektorii. Přesnost se počítá na základě kriteriální funkce (viz. 4.1.1. a 4.1.2.). Výstupem optimalizace je matice řešení $\boldsymbol{X}$:

$$\boldsymbol{X} = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ d \\ P_x \\ P_y \end{pmatrix} \tag{2.6}$$

2.3.1. MATLAB funkce *fminsearch*

Funkce *fminsearch* je součástí nástroje *Optimization tool* programu MATLAB. Dokáže minimalizovat námi zvolenou nelineární funkci více proměnných, která nemá žádná omezení, a to pomocí simplexového algoritmu. Umí si poradit s nespojitou funkcí, pokud se nespojitost nenachází v blízkosti řešení. *Fminsearch* vyhledává pouze lokální minimum, takže přesnost záleží také na počátečním odhadu. Obě MATLAB funkce, které zde popisuji, mají velkou citlivost řešení na počáteční odhad. Jeho změnou, která je o 5 řádů níže, než konečný výsledek, dostáváme diametrálně jiné řešení. Podrobný popis algoritmu lze najít v [5].

Pro zahájení výpočtu tedy potřebuje počáteční odhad $\boldsymbol{X_{in0}}$. Ten lze získat některým z algoritmů popisovaných v kapitolách 2.3.3. a 2.3.4. Zápis funkce vypadá následovně:

```
[x,fval,exitflag] = fminsearch(fun,x0,options)
```

Obr. 2.4 Ukázka volání funkce fminsearch

Do argumentu funkce postupně vkládáme optimalizovanou funkci (fun), počáteční odhad řešení $(x0)$, nastavení $(options)$. Do výstupu se nám ukládá optimalizované řešení (x), hodnota funkce v tomto bodě $(fval)$, a impuls $(exitflag)$, jakým byla optimalizace ukončena.

2.3.2. MATLAB funkce *fmincon*

Tuto funkci také najdeme v *Optimization tool* programu MATLAB. Iteračním procesem minimalizuje nelineární funkci o více proměnných, která však může obsahovat sadu podmínek. Funguje pouze pro spojité funkce, které mají spojitou první derivaci. K výpočtu je nutný počáteční odhad $\boldsymbol{X_{in0}}$. Podrobný popis algoritmu lze najít v [6]. Zápis funkce:

```
[x,fval,exitflag] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)

 output =

         iterations: 1
          funcCount: 46
     constrviolation: 0
           stepsize: 1.3691e-10
          algorithm: 'interior-point'
      firstorderopt: 9.0631e+03
       cgiterations: 35
            message: 'Local minimum possible. Constraints satisfied
```

Obr. 2.5 Ukázka volání funkce fmincon a popis jeho výstupu

Výpis $(output)$ informuje o průběhu a výstupu funkce. Nás zajímá, zda nebyla porušena omezení $(constrviolation)$ a zda bylo dosaženo minima $(message)$. V argumentech funkce se kromě parametrů zmiňovaných v 2.3.1. objevují následující podmínky pro řešení:

$$Lineární\ nerovnosti: \boldsymbol{A \cdot x \leq b} \tag{2.11}$$

$$Lineární\ rovnosti: \boldsymbol{Aeq \cdot x = beq} \tag{2.12}$$

$$Spodní\ a\ horní\ hranice: \boldsymbol{lb \leq x \leq\ ub} \tag{2.13}$$

$$Nelineární\ podmínky\ rovnosti\ a\ nerovnosti\ (nonlcon):$$

$$\boldsymbol{c(x) \leq 0, ceq = 0}, funkce\ vracející\ vektor \tag{2.14}$$

Z uživatelského hlediska má *fmincon* velkou výhodu díky tomu, že smíme řešení omezit podmínkami. Tyto podmínky nejsou terminální. Algoritmus je smí porušit ve velmi malém, implicitně definovaném rozsahu. Jsme schopni definovat proporční závislosti v mechanismu a tím zaručit jeho realizovatelnost. Dále je možné přímo určit délku jednotlivých ramen - omezit oblast hledání, což má dva důsledky:

- Možnost zhoršení přesnosti nalezeného mechanismu, pokud výrazně omezíme prohledávanou oblast a tím eliminujeme „lepší řešení“
- Podstatné zkrácení výpočetního času omezením prohledávané oblasti na zadaný region

2.3.3. Metoda Monte Carlo

Monte Carlo (MC) nachází uplatnění při řešení jedno i vícerozměrných integrálů, systémů lineárních rovnic či parciálních diferenciálních rovnic. MC řeší úlohy pomocí mnohokrát opakovaných náhodných pokusů. V dnešní době se využívají pseudonáhodná čísla generovaná číslicovým počítačem. Podstatou řešení je sestrojení pravděpodobnostní úlohy, která má stejný charakter jako původní úloha. Ucelená teorie a příklady použití jsou zmíněny v [3].

V našem případě bude postup následující:

- Vygenerujeme určitou množinu ***M*** matic řešení ***X***, která budou složena ze pseudonáhodných čísel na zadaném intervalu
- Vypočteme řešitelnost ***X*** pro zadaný interval akčního úhlu
- Pakliže je mechanismus ***X*** řešitelný, zhodnotíme jeho trajektorii z hlediska kriteriální funkce
- Řešení ***X*** s nejnižší hodnotou kriteriální funkce (nejméně odlišnou trajektorií) vybereme jako nejvhodnější

MC dosahuje při naší aplikaci a velikosti množiny ***M*** o 2-3 řády větší hodnoty kriteriální funkce než metody MATLAB *optimization toolboxu* (2.3.1., 2.3.2.). Proto bude použita spíše jako počáteční odhad pro tyto iterativní metody, než jako konečný výsledek.

2.3.4. Genetický algoritmus

Genetický algoritmus (GA) je heuristická metoda, která využívá principů přirozené selekce, známých z evoluční biologie. Používá se při numerické optimalizaci, strojovém učení, tvorbě modelů (ekonomických, sociálních, populačních) a obecně k řešení matematických problémů, pro něž neexistuje exaktní algoritmus.

GA je sofistikovanější algoritmus založený také na generování náhodných čísel na zadaném intervalu. Tento algoritmus je časově náročnější než MC a má několik volitelných parametrů. Při optimalizaci dosahuje podobné úspěšnosti jako MC.

Algoritmus je možno regulovat následujícími faktory:

- Počet generací (opakování cyklu)
- Velikost populace (počet chromozomů)
- „Crossover rate“ (pravděpodobnost výběru chromosomu pro „zkřížení“)
- „Mutation number“ (pravděpodobnost výběru genu pro „mutaci“)

Popis algoritmu[4]:

- **Inicializace:** Začneme stvořením náhodné populace ***M*** o několika chromozomech, kde chromozom je jedna sada řešení ***X*** $(L_1, L_2, L_3, d, P_x, P_y)$. Element chromozomu se nazývá gen.

- **Zhodnocení:** Spočteme, které chromozomy populace (***X***) jsou řešitelné a u těch následně vypočítáme i jejich přesnost. Ty nejpřesnější mají nejvyšší pravděpodobnost dostat své geny do příští populace.
- **Selekce:** Výběr chromozomů pro další generaci se nazývá selekce. Spočteme vhodnost $(fitness)$, pravděpodobnost výběru $(prob)$ a kumulativní pravděpodobnosti (c_prob). Vygenerujeme množinu (o velikosti populace) náhodných čísel (R) a ta nám určí, který chromozom bude vybrán.

$$fitness_i = \frac{1}{1+fk_i}, kde\ fk_i\ je\ i-tá\ hodnota\ kriteriální\ funkce \quad (2.7)$$

$$prob = \frac{fitness_i}{\sum_{i=1}^{n} fitness_i} \quad (2.8)$$

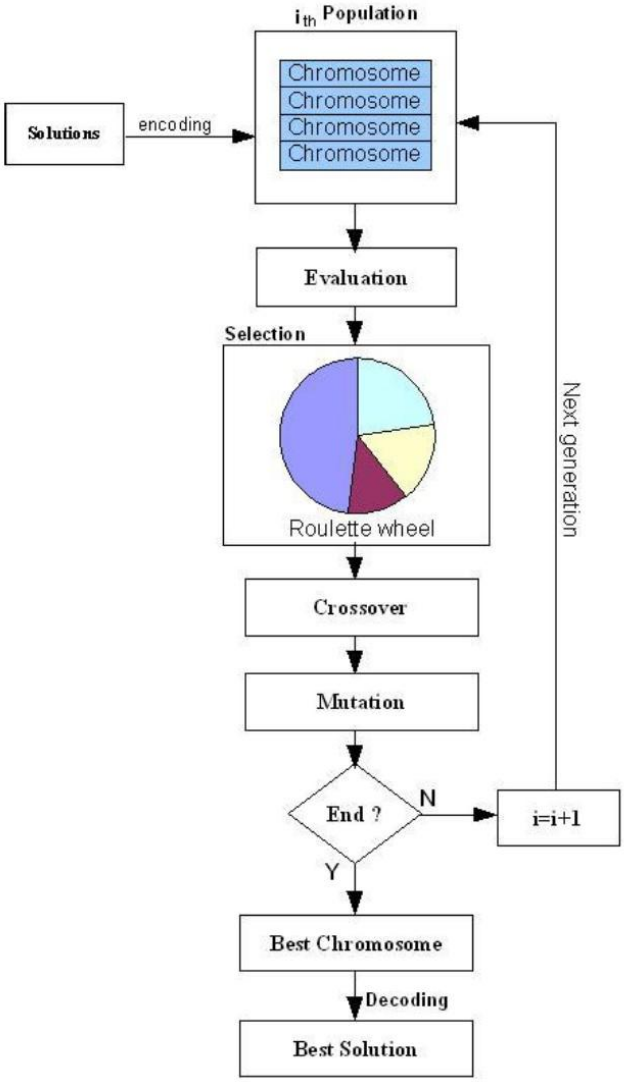
$$c\_prob(k) = \sum_{i=1}^{j} fitness_i \quad (2.9)$$

$$\left[c_{prob(j)} < R(k) < c_{prob(j+1)}\right] \quad (2.10)$$

$$=> (i+1)\ chromosom\ bude\ zvolen$$

- **Zkřížení (crossover):** Několik chromosomů (***X***), zvolených pro další generaci, si vymění své geny (záleží na „Crossover rate“).
- **Mutace**: Mutace vnáší do algoritmu další náhodný prvek. Náhodně zvoleným genům se přiřadí náhodná hodnota. Počet zmutovaných genů záleží na „Mutation number“. Algoritmus dále pokračuje v části „Zhodnocení“.

Po zopakování n-cyklů (generací), nebo dosažení určité přesnosti se vybere nejlepší hodnota ze všech řešení.



Obr. 2.4 Schéma genetického algoritmu [4]

3. Formulace problému

Úkolem je otestovat pro čtyřčlenný mechanismus a zadanou realizovatelnou trajektorii konvergenci optimalizační metody k cílové trajektorii. Hodnotit se bude zejména:

- Časová náročnost
- Přesnost
- Citlivost na počáteční odhad

Dále je nutné podle výsledků předchozí fáze:

- Nalézt ideální nastavení parametrů ovlivňujících výpočet
- Pokusit se co nejpřesněji optimalizovat základní typy trajektorií
- Vyslovit podmínky na trajektorii, pro niž lze úspěšně optimalizovat mechanismus

4. Řešení a výsledky

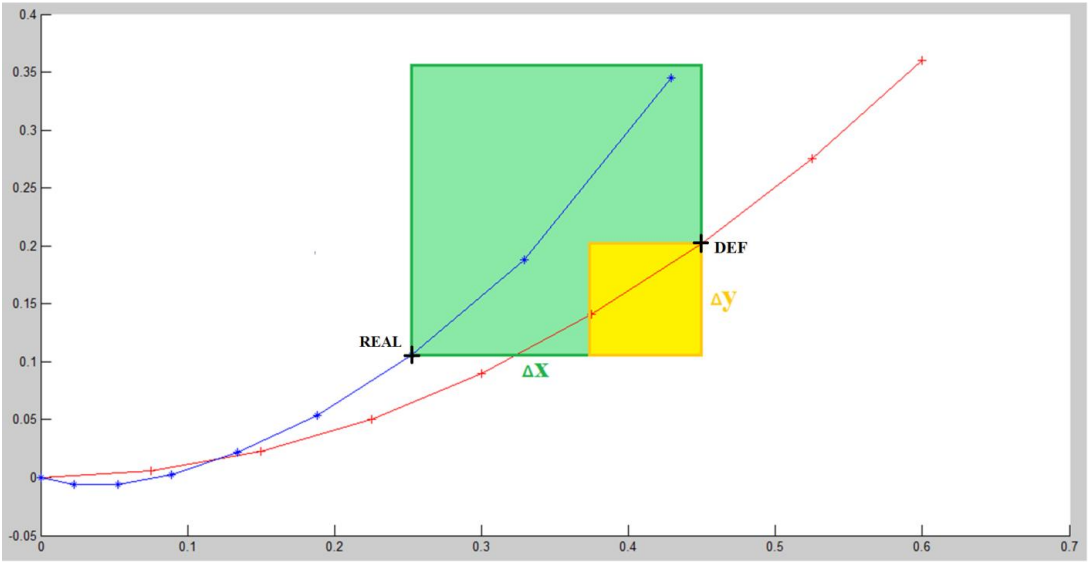
4.1. Analytická část

Při procesu optimalizace bude použit matematický aparát na hodnocení, úpravu, diskretizaci trajektorie. Dále je nutné rozpoznat řešitelnost mechanismu a vytvořit náhodná čísla na daném intervalu. Tyto operace lze výhodně provést v prostředí MATLAB.

4.1.1. Kriteriální funkce MNČ

Metodou nejmenších čtverců (MNČ) spočteme sumu kvadrátů odchylek bodů definované trajektorie od bodů té reálné. Tuto hodnotu můžeme považovat za jakési měřítko kvality – hodnotu kriteriální funkce (KF) vygenerované trajektorie a následně podle ní trajektorie porovnávat. Výhoda MNČ tkví v tom, že zohledňuje odchylku čtverce na jedné ose, i když na druhé (kolmé) je minimální. Pokud bychom například sčítali pouhé obsahy obdélníků, mohlo by se stát, že jedna složka bude nulová a tím vyruší podstatnou druhou složku.

$$Err_{MNČ} = \sum_{i=1}^{n}[(x_i^{DEF} - x_i^{REAL})^2 + (y_i^{DEF} - y_i^{REAL})^2] \tag{4.1}$$



Obr. *4.1 Schéma metody nejmenších čtverců* ($Err_{MNČ} = 0{,}2326$)

4.1.2. Kriteriální funkce MNČ s penalizací poměrů ramen

Jednoduchá MNČ nijak neomezuje délky ramen mechanismu. Takto generované mechanismy jsou teoreticky pořádku, konvergují k řešení rychleji, ale mohou mít prakticky nerealizovatelné rozměry. Proto se jeví jako vhodné zavést penalizace, jež zvyšují hodnotu kriteriální funkce (KF), pokud je porušena některá z podmínek:

$$P_{MECH} = \begin{cases} 0, & P_{MECH} < t \\ P_{MECH}, & P_{MECH} \geq t \end{cases} \qquad P_{ARM} = \begin{cases} 0, & P_{ARM} < u \\ P_{ARM}, & P_{ARM} \geq u \end{cases} \tag{4.2}$$

$$P_{MECH} = \frac{max\ (|L_1|, |L_2|, |L_3|, |d|)}{min\ (|L_1|, |L_2|, |L_3|, |d|)} \tag{4.3}$$

$$P_{ARM} = \frac{max\ (|P_x|, |P_y|)}{min\ (|L_1|, |L_2|, |L_3|, |d|)} \tag{4.4}$$

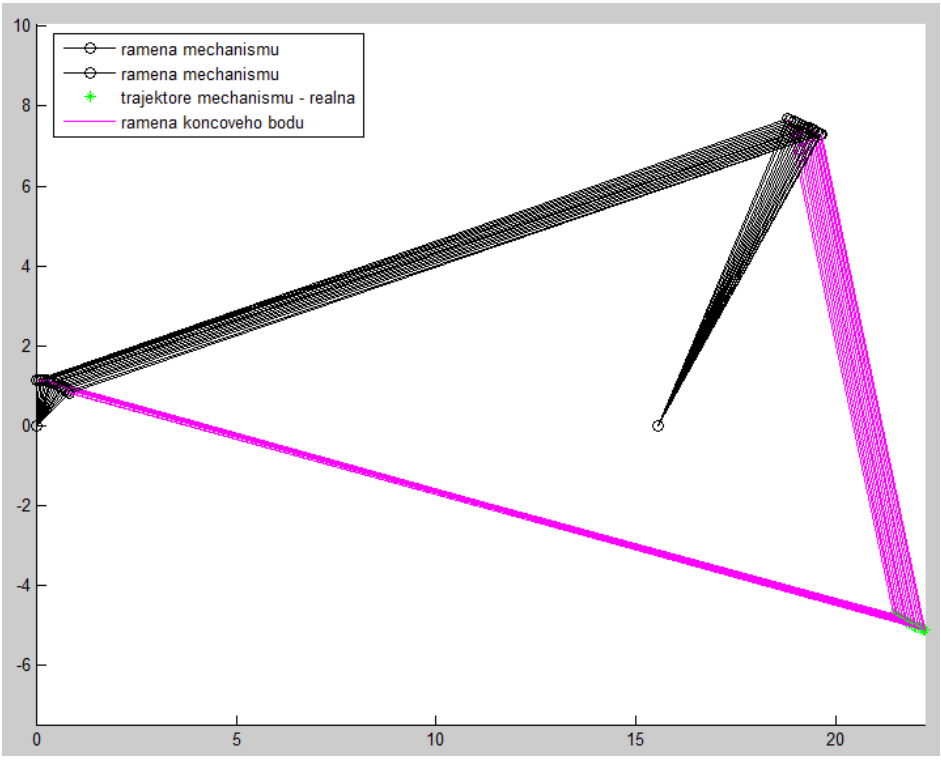
$$P_{MECH} \ldots poměr\ nejdelšího\ k\ nejkratčímu\ ramenu\ čtyřčlenného\ mechanismu$$

$$P_{ARM} \ldots poměr\ nejdelšího\ ramene\ (složky\ v\ souřadnicovém\ systému\ 2)\ sledovaného\ bodu\ ku\ nejkratšímu\ ramenu\ mechanismu$$

Konstanty t, u vyjadřují maximální dovolené poměry. Pokud je nejkratší rameno maximálně desetinásobkem toho nejdelšího, pak mechanismus opticky splňuje podmínky realizovatelnosti. Čtyřčlen s velkými rozdíly rozměrů koná při vykreslování trajektorie velmi nepatrný pohyb vůči svým rozměrům (viz obr 4.2). Takto je potom reálný mechanismus náchylný k chybám. Proto doporučuji nastavit t, u v rozmezí 5-10. Kompletní kriteriální funkce má tvar:

$$Err_{CELK} = Err_{MNČ} + c_1 \cdot P_{MECH} + c_2 \cdot P_{ARM} \tag{4.5}$$

Kriteriální funkce (4.5) zaručuje vyšší pravděpodobnost, že minimum funkce nalezneme pro realizovatelný mechanismus. Konvergence řešení záleží na rozsazích poměrů (čím větší, tím lepší) a na nastavení konstant c_1, c_2. Potřebujeme-li striktně dodržet poměry v mechanismu, doporučuji nastavit konstanty tak, aby hodnota KF při porušení jednoho z pravidel nemohla klesnout pod mez tolerované chyby úspěšného řešení. Je efektivní použít tuto KF v kombinaci s optimalizačním algoritmem *fminsearch*. K penalizační metodě jsem čerpal z [8].



Obr. 4.2 Příliš velké poměry ramen ($P_{MECH} = 10{,}86;\ P_{ARM} = 17{,}18$)

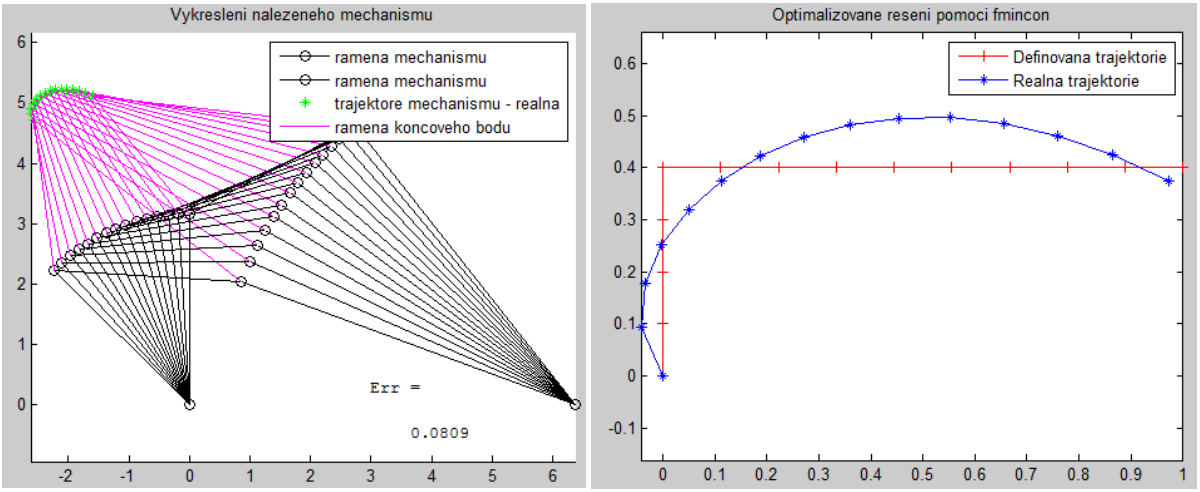
4.1.3. Zavedení trajektorie do relativního souřadnicového systému

Vzhledem k tomu, že nás u trajektorie zajímá zejména její tvar a nikoliv poloha vůči mechanismu (globálnímu souřadnému systému), je výhodné zavést počátek trajektorie do samostatného relativního souřadného systému. V tomto systému bude porovnávána s předdefinovanou dráhou pomocí kriteriální funkce.

$$x_{i,REL}^{P} = x_{i,ABS}^{P} - x_{1,ABS}^{P} \tag{4.6}$$

$$y_{i,REL}^{P} = y_{i,ABS}^{P} - y_{1,ABS}^{P} \tag{4.7}$$

Touto úpravou zvětšíme množinu řešení, která nám vyhovují, tedy zrychlíme a zpřesníme výpočet.



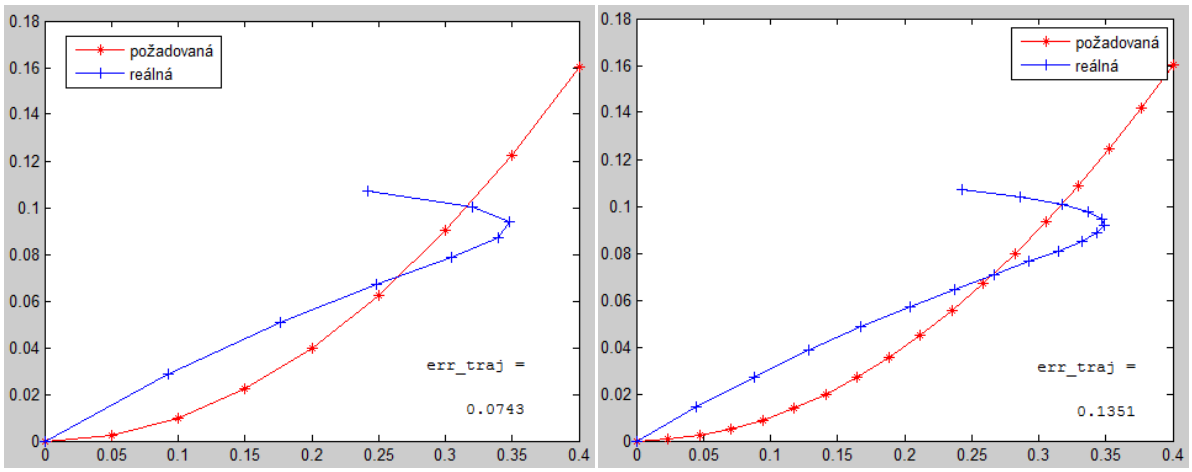
a) *b)*

Obr. 4.3 a) Mechanismus; b) Požadovaná a reálná trajektorie zavedená do počátku souřadnic

4.1.4. Diskretizace trajektorie koncového bodu P

Mechanismus opisuje reálnou, spojitou křivku. Abychom ji ale mohli porovnat, musíme dosadit do vzorce 4.1, který vyžaduje pouze body. Proto je nutné úlohu diskretizovat – rozdělit trajektorii na body, kterými sledovaný bod P prochází. Tím se částečně vzdáváme přesnosti, ale numerické řešení tento postup vyžaduje.

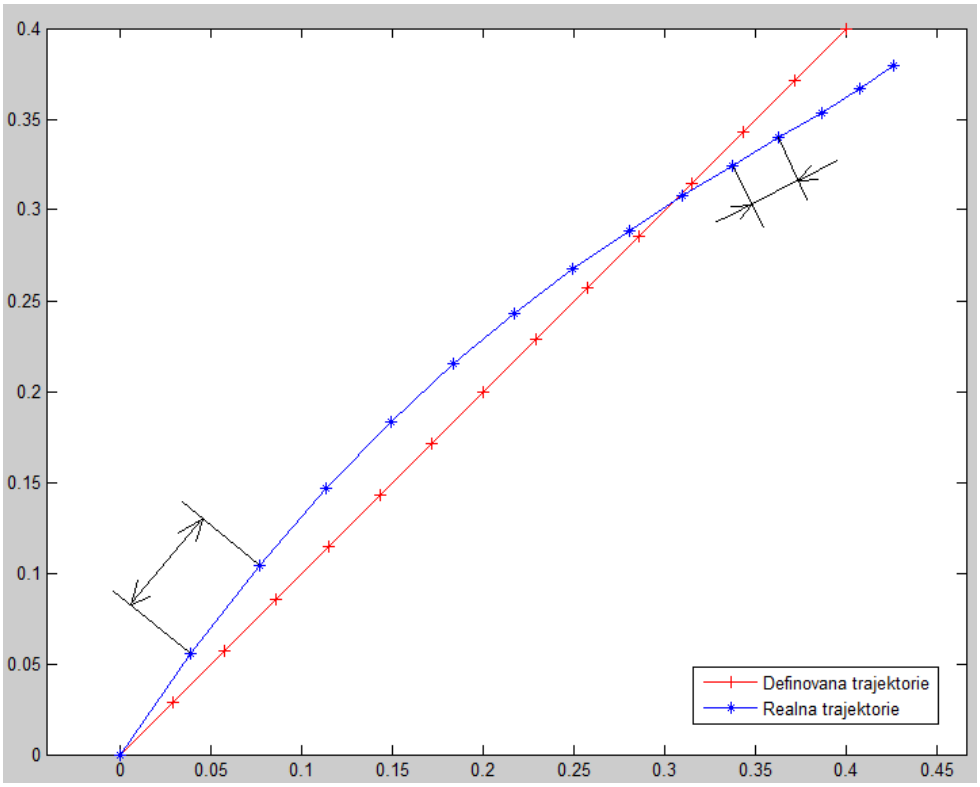
Je nutné zvolit „dostatečné“ množství průchozích bodu tak, aby přesnost vyhovovala aplikaci. Porovnáváme-li jedno řešení při různém počtu průchozích bodů, bude mít křivka rozdělená na více intervalů i větší chybu. Jak je vidět z obr. 4.4, chyba trajektorie roste pro dané řešení úměrně s počtem bodů.



a) *b)*

Obr. 4.4 a) 9 průchozích bodů ($Err_{MNČ} = 0{,}0743$)*; b) 15 průchozích bodů* ($Err_{MNČ} = 0{,}1351$)

Nejjednodušší diskretizace dosáhneme, rozdělíme-li akční úhel q_1 na $(n-1)$ intervalů, než chceme průchozích bodů. Pro tuto množinu stejně velkých úhlových intervalů následně spočteme pomocí přímé kinematiky polohu bodu P. Množina průběžných poloh bodu P pak vystihuje dráhu reálného mechanismu. V ideálním případě rozdělíme trajektorii na několik stejně dlouhých částí. Reálně tomu tak není, neboť závislost velikosti úhlového intervalu na uražené vzdálenosti bodu P není lineární (obr. 4.5).



Obr. 4.5 Rozdíl délky intervalu na trajektorii pro stejný rozsah akčního úhlu

4.1.5. Zaručení řešitelnosti mechanismu

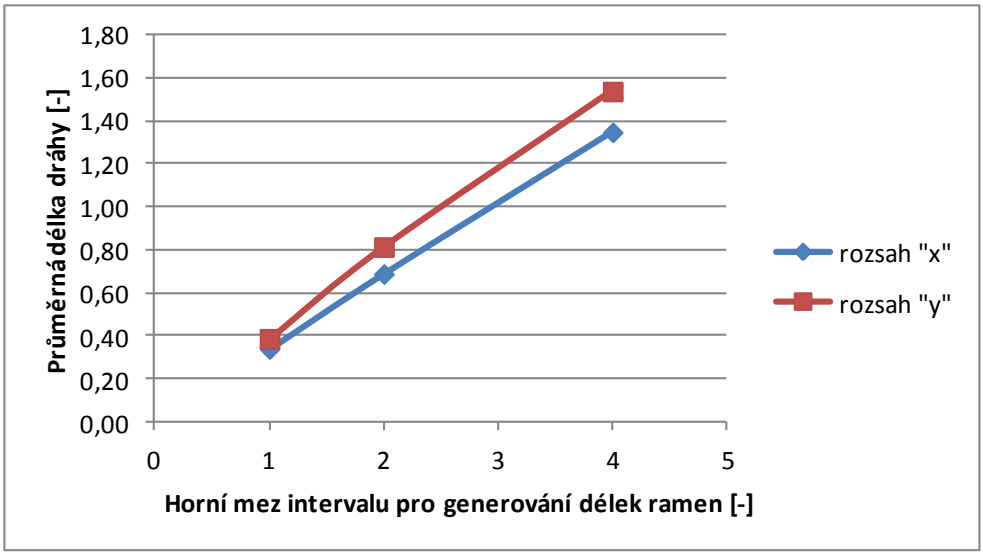
Řešitelnost mechanismu lze zaručit buď analyticky z podmínek (2.5) nebo numericky pomocí iteračního procesu (2.4). Zmíněný proces konverguje (je schopen sestavit mechanismus zadaných délek pro pevně daný úhel q_1) v řádech jednotek až desítek kroků. Pokud tedy při while-cyklu nekonverguje během 100 iterací pod určenou toleranci (u trajektorií na intervalu $\langle 0{,}1\rangle$ je dostatečná tolerance chyby sestavení 0,001), je pro tento úhel q_1 mechanismus neřešitelný.

4.1.6. Tvorba náhodného čísla na daném intervalu

V algoritmu MC i GA generujeme pseudonáhodná čísla. V prostředí MATLAB je to obvykle pomocí příkazu *rand*. Ten generuje čísla v intervalu $\langle 0; 1\rangle$, což je při naší aplikaci nevýhodné. Potřebovali bychom rozsah náhodných čísel (=rozsah velikosti ramen) přizpůsobit velikosti trajektorie. Pro vygenerování náhodného čísla z na intervalu $\langle a; b\rangle$ použijeme tento zápis:

$$z = a + b \cdot rand \tag{4.8}$$

Experimentálně jsem zjistil, že trajektorie produkovaná mechanismem, jehož délky ramen jsou $\langle 0; 1\rangle$ a akční úhel $\frac{\pi}{4}$ mívá rozsah $\Delta x = 0{,}34;\ \Delta y = 0{,}39$. Zvětšíme-li interval, z něhož generujeme náhodné délky ramen, bude se alespoň do hodnoty $\langle 0; 4\rangle$ chovat lineárně (viz obr. 4.6). Pak tedy máme možnost upravovat velikosti odhadovaných délek ramen v závislosti na velikosti trajektorie, což nám umožňuje lepší prvotní odhady, tudíž i optimalizované výsledky.



Obr. 4.6 Závislost velikosti trajektorie na intervalu odhadovaných náhodných čísel

4.2. Optimalizační část

4.2.1. Prvotní odhad pro optimalizaci řešení a jeho realizace

MC a GA mají parametry, při nichž dosahují různé hodnoty KF a také různou dobu výpočtu:

- MC – počet pokusů
- GA – počet generací

Tyto hodnoty se pokusím experimentálně zjistit na příkladu optimalizace pro trajektorii $y = 0{,}5 \cdot sin(5x)$ při akčním úhlu $q_1 = \langle \frac{\pi}{2}; \frac{\pi}{4} \rangle$ a devíti průchozích bodech. Výpočet bude probíhat na počítači s procesorem Intel Core i3 M380 2,53 GHz.

Měřit čas v prostředí MATLAB lze funkcemi *tic-toc* a *cputime*. Vzhledem k tomu, že *cputime* (čas který strávila centrální procesorová jednotka výpočtem) nenavrací konzistentní hodnoty na různých operačních systémech[7], použiji k hodnocení *tic-toc*. Je nutné zmínit, že pokud slouží MC nebo GA k prvotnímu odhadu pro funkce *optimization toolboxu*, nezáleží úspěch optimalizace pouze na tom, jak velká je chyba odhadu, ale také na tom jak „podobné“ jsou si křivky, či zda „vedou“ odhadnuté rozměry mechanismu na požadované minimum funkce.

Prvotní odhad bude využit pro rychlé přiblížení k řešení. Jakmile se začne křivka $Err_{MNČ} = f(parametr)$ snižovat strmost klesání, je efektivnější využít *fminsearch/fmincon*. Budu proto odhadovat parametry pro MC, GA na základě polohy tohoto bodu. Všechny programy a funkce jsou připojeny na CD.

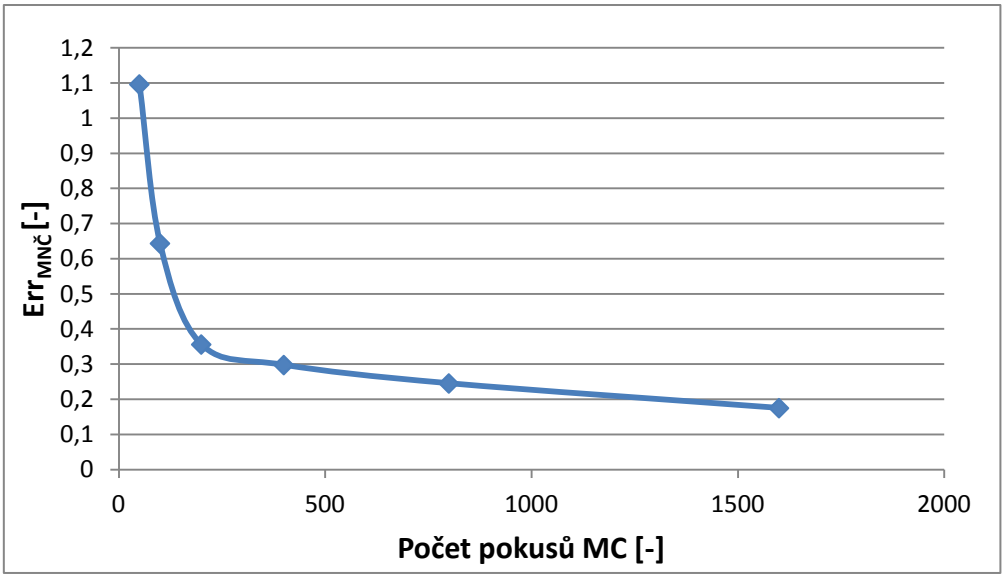
4.2.1.1. MC

Při experimentu bylo opakováno vždy 10 měření pro 50, 100, 200, 400, 800 a 1600 pokusů metody MC. Z každého z 10 měření bylo vybráno nejpřesnější řešení a ze všech dohromady pak stvořen aritmetický průměr. Tento průměr chyby nejlepšího řešení pak byl zanesen do grafů v závislosti na počtu pokusů a času nutném pro jejich výpočet. Křivky $Err_{MNČ} = f(t)$, $Err_{MNČ} = f(n - pokusů)$ mají hyperbolický charakter. Závislost $t = f(n - pokusů)$ je téměř striktně lineární, proto mají grafy 4.7-a, b stejnou vypovídající hodnotu.

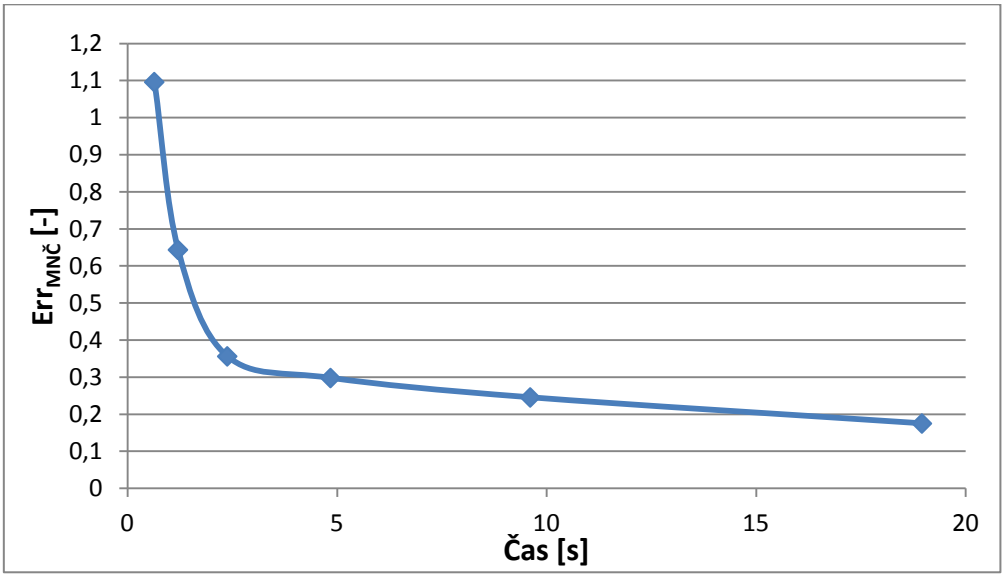
Z 4.7-a vyplývá, že u 200 pseudonáhodných pokusů se začneme blížit na hranici efektivnosti tohoto algoritmu. Nejlepší průměrná chyba byla podle očekávání pro největší počet pokusů ($Err_{MNČ} = 0{,}1750$). Nejnižší chyba během celého měření byla $Err_{MNČ} = 0{,}0637$, což je nepřesné, protože pro takto dlouhou trajektorii odpovídá přesnému řešení řád chyby $Err_{MNČ} \cong 10^{-3}$. Podle těchto dat následně odhadneme ideální množství pokusů pro tuto metodu, abychom dosáhli co možná největší přesnosti při největší úspoře výpočetního času.

Pro použití algoritmu MC jako prvotního odhadu volím 200 pokusů a přesnost, kterou po něm požaduji je 0,3 (pokud nedosáhne této přesnosti, bude se opakovat).

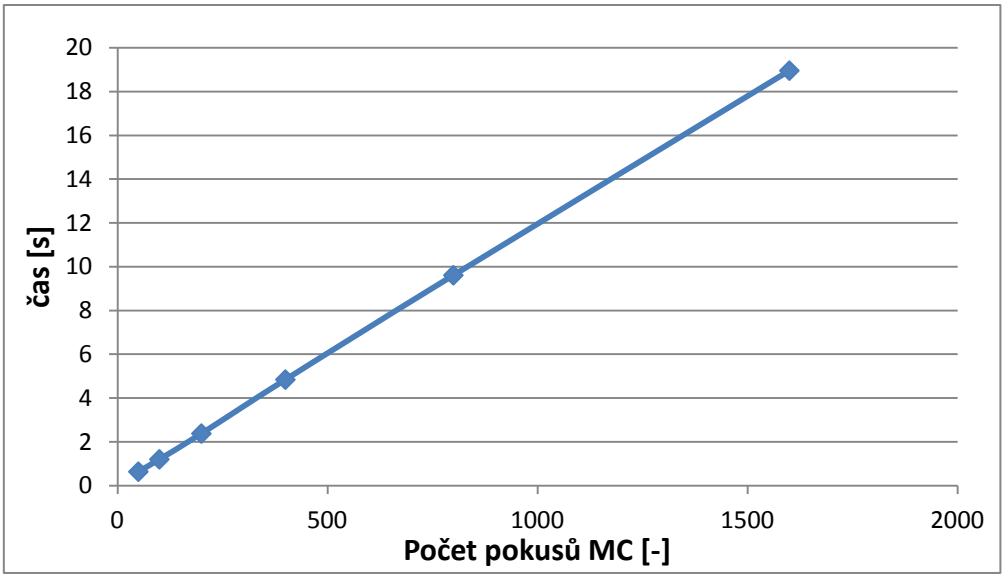
Algoritmus je přiložen na CD jako *mc_mereni.m*.



Obr. 4.7 - a) Závislost průměrné chyby nejlepších řešení na počtu pokusů algoritmu MC, který tato řešení generoval



Obr. 4.7 - b) Závislost průměrné chyby nejlepších řešení na době výpočtu algoritmu MC, který tato řešení generoval



Obr. 4.7 - c) Závislost průměrného výpočetního času na počtu pokusů MC

4.2.1.2. GA

Pro experiment s GA jsem nastavil tyto parametry:

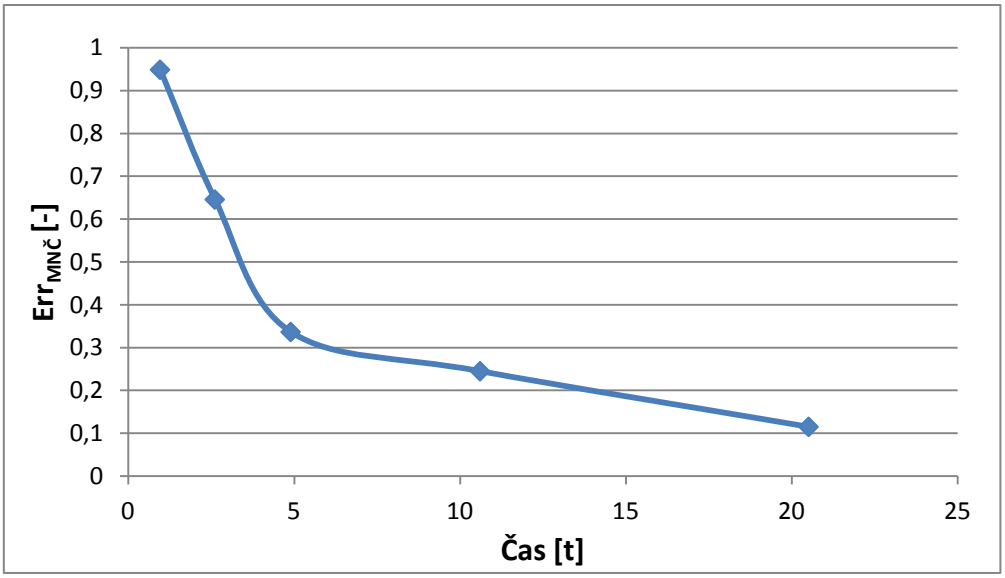
- Velikost populace = 50
- Pravděpodobnost zkřížení = 0,2
- Pravděpodobnost mutace = 0,1
- Počet generací bude hledaným parametrem

Dále jsem zvyšoval počet generací od 1, kdy by výsledky měly přibližně odpovídat algoritmu MC, až po 40 (1, 5, 10, 20, 40). Měření jsem opakoval 10x pro každý případ. Z každé poslední generace bylo vybráno nejlepší řešení a jejich suma zprůměrována. Průměrná hodnota nejlepších řešení figuruje v závislostech na výpočetním čase a počtu generací na obr 4.8-a, b. Křivky $Err_{MNČ} = f(t)$, $Err_{MNČ} = f(n - generací)$ mají hyperbolický charakter. Závislost $t = f(n - generací)$ je téměř lineární. Charakteristiky mají podobný tvar jako u MC.

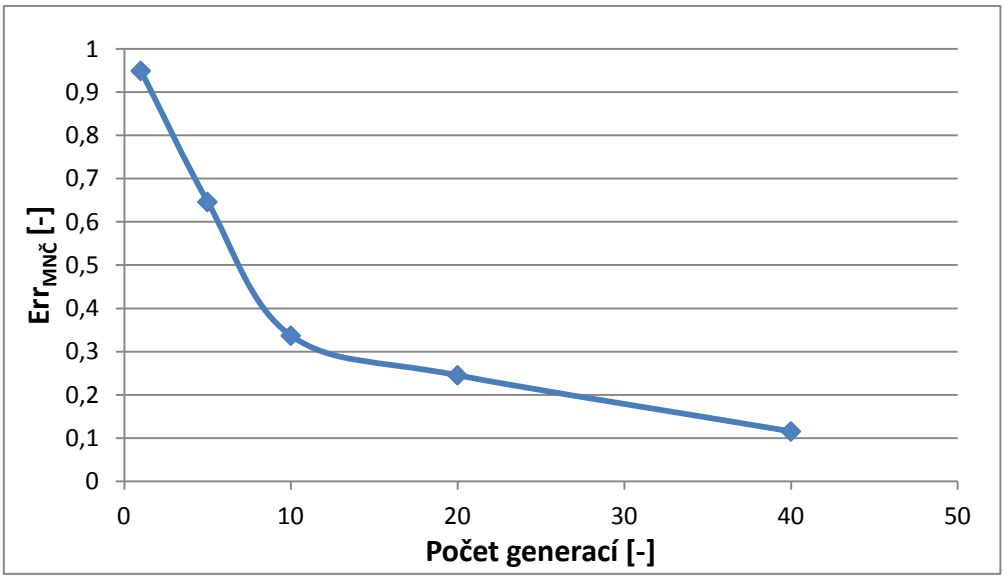
Touto metodou jsme schopni dosáhnout průměrné přesnosti $Err_{MNČ} = 0{,}1153$, kdy je ovšem výpočet zdlouhavý a neefektivní. Nejnižší chyba byla evidována dle předpokladu pro největší počet generací (výpočetní čas). Nejvyšší zaznamenaná přesnost byla $Err_{MNČ} = 0{,}0584$. Až na průměrnou přesnost jsou hodnoty obdobné jako u MC.

Pro použití algoritmu GA jako prvotního odhadu volím 10 generací, které mají dosáhnout přesnosti 0,3 (Pokud funkce nedosáhne této přesnosti, bude se opakovat).

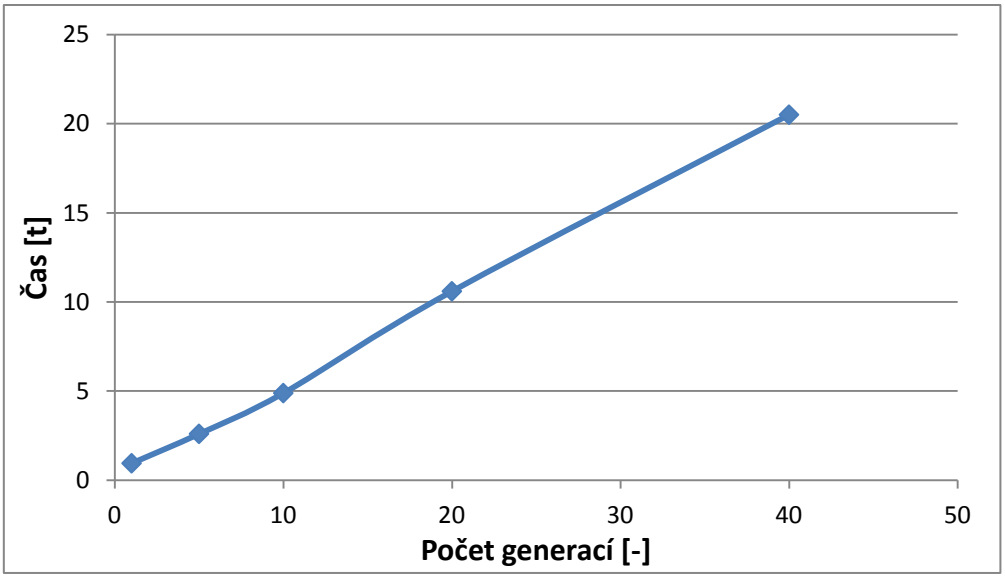
Algoritmus je přiložen na CD jako ga_*mereni.m*.



Obr. 4.8 – a) Průměrná chyba nejlepších řešení v poslední generaci v závislosti na průměrné době jejich optimalizace



Obr. 4.8 – b) Průměrná chyba nejlepších řešení v poslední generaci v závislosti na počtu generací

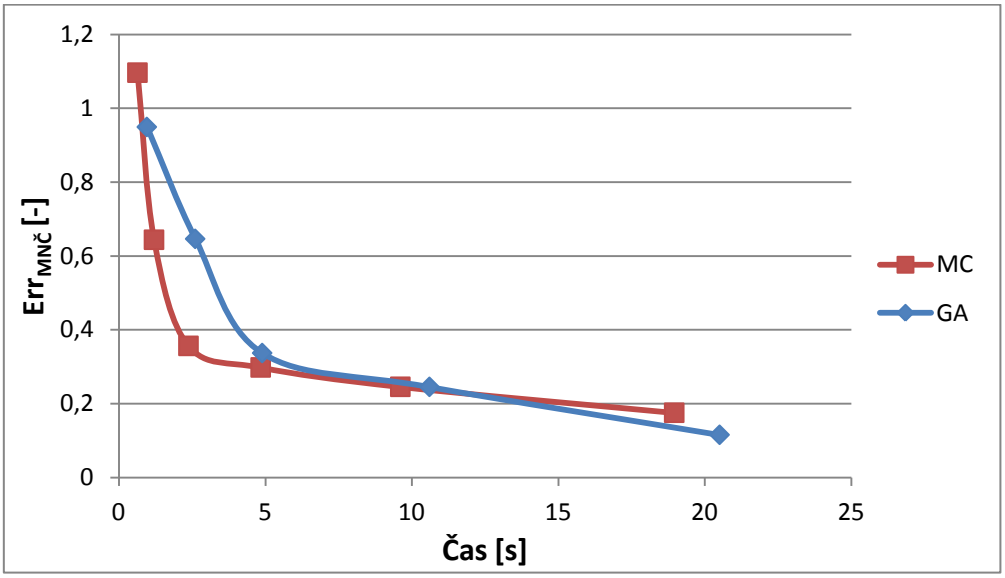


Obr. 4.8 – c) Průměrná doba optimalizace v závislosti na počtu generací

4.2.1.3. Srovnání MC a GA

Pro srovnání MC a GA nám poslouží závislost jejich přesnosti na výpočetním čase. Z obr. 4.9 vyplývá:

- GA je schopen dosáhnout při výpočtech přesahujících 15 s vyšší přesnosti (o 30% pro 20 s)
- MC dojde k přesnějšímu výpočtu při rychlostech do 5 s (o 45% pro 2,5 s)
- V rozmezí 5-15 s je efektivita algoritmů téměř stejná



Obr. 4.9 Porovnání přesnosti GA oproti MC

Při hledání počátečního odhadu vítězí algoritmus MC, protože dříve než za 14 s není GA průměrně schopen vygenerovat lepší výsledek. Navíc MC vyprodukuje za 2,4 s hodnotný

výsledek, jenž by trval GA minimálně dvojnásobnou dobu. MC je mnohem jednodušší algoritmus.

Optimalizace pomocí těchto algoritmů je stále nedokonalá, potřebovali bychom přesnost o řád nižší - okolo 0,001.

4.2.2. Optimalizace pomocí *optimization toolboxu*

Počáteční odhady dosahují chyby těsně pod $Err_{MNČ} = 0{,}3$, již je nutno dále optimalizovat o 2 řády pro uspokojivé řešení. To bude provedeno pomocí funkcí MATLAB *optimization toolboxu*. Otázkou je, který z algoritmů dokáže dosáhnout větší přesnosti. Testování zmíněných funkcí bude probíhat při stejné trajektorii a podmínkách jako v předešlé kapitole 4.2.1. Experiment má velmi hrubý informativní charakter, co se týče časového odhadu, protože nebyl opakován a také kvůli procesům na pozadí počítače, které nelze dokonale eliminovat.

4.2.2.1. Fminsearch v kombinaci s MNČ a penalizacemi

V tomto případě je využita KF zmíněná v 4.1.2. Podmínky v ní užité (4.2) mají vliv pouze na hodnotu KF a tak nepřímo ovlivňují poměry v mechanismu při optimalizaci. V mém experimentu platí, že pokud je chyba trajektorie příliš vysoká, ovlivní ji poměry v mechanismu jen minimálně. Chyby pramenící z poměrů začnou nabírat na váze, až když se hodnota KF sníží. Muže se tedy stát, že optimalizace nalezne minimum funkce i pro disproporční mechanismus.

Velmi důležitá je volba konstant c_i pro KF tak, aby ovlivnily celkovou chybu ve vhodné hladině. Já je volím $c_1 = c_2 = 1{,}667 \cdot 10^{-4}$, aby při překročení tolerovaného poměru mechanismu $t = u = 6$, dosáhly alespoň hodnoty 0,001. Tak bude zaručena nemožnost konvergence řešení disproporčního mechanismu pod tuto mez.

Při optimalizaci byl zvolen vzorek řešitelných počátečních odhadů (nástřelů), které byly vygenerovány pomocí MC. Bylo dosaženo následujících hodnot:

- Počet vyhodnocovaných nástřelů: 119
- Čas: 1007,9176 s
- Průměrný čas: 8,4699 s
- Průměrná přesnost odhadu: 5,6486
- Průměrná přesnost optimalizovaného řešení: 0,7088
- Nejlepší řešení: $2{,}6849 \cdot 10^{-3}$, P_{ARM} = 11,5930, P_{MECH} = 3,1302
- Korelační koeficient: 0,1023

$$r_{xy} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{s_x \cdot s_y} \tag{4.9}$$

$$[x_i, y_i] \ldots souřadnice\ i - tého\ bodu$$

$$\bar{x}, \bar{y} \ldots aritmetický\ průměr\ \ souřadnic\ v\ daném\ směru$$

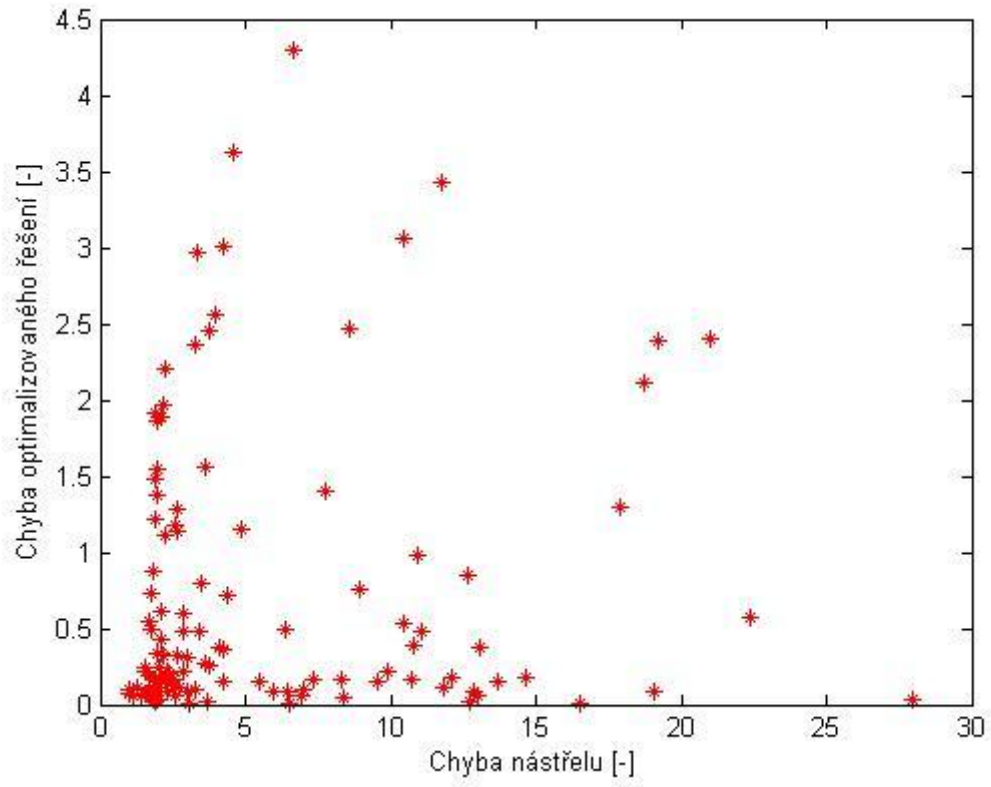
$$s_x, s_y \ldots směrodatná\ odchylka\ \ souřadnic\ v\ daném\ směru$$

Fminsearch vylepšil průměrné řešení o 87,5% oproti prvotnímu odhadu. Nejlepší má chybu $Err_{CELK} = 2{,}6849 \cdot 10^{-3}$. Uvědomíme-li si, že hodnota penalizace v tomto případě dosahuje $1{,}9326 \cdot 10^{-3}$, jedná se o akceptovatelné řešení. Na druhou stranu vidíme disproporční mechanismus, který byl nalezen pro minimum KF, přestože nesplňuje zvolené podmínky realizovatelnosti. Možnost porušení podmínek však skýtá výhodu tohoto přístupu, obzvláště když máme možnost obětovat disproporci za přesnost.

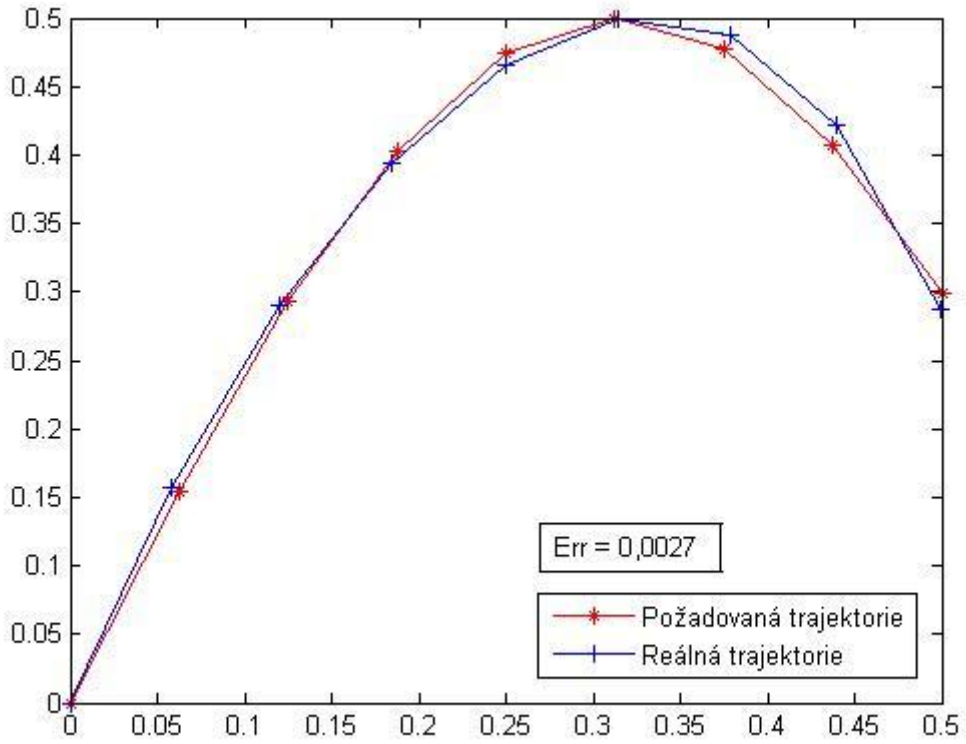
Podle koeficientu korelace (4.9)[9] dokážeme exaktně určit, zda je přesnost optimalizace závislá na počátečním odhadu. Aplikujeme-li jej na množinu bodů z obr. 4.10, dosahuje hodnoty 0,1023, což ukazuje na velmi slabou přímkovou závislost s kladnou směrnicí. Zajímavé jsou četné úspěšné optimalizace ($Err_{MNČ} < 1$) nepřesných počátečních odhadů. Za vysokou hustotu dobrých řešení v levém dolním rohu je zodpovědné spíše velké množství nástřelů v oblasti $\langle 0; 0{,}5 \rangle$, jejichž počet se s chybou nástřelu snižuje.

Ze zmíněného lze vyvozovat existenci jiných proměnných, na nichž závisí přesnost optimalizovaného řešení. Může to být tvar trajektorie, respektive její křivost.

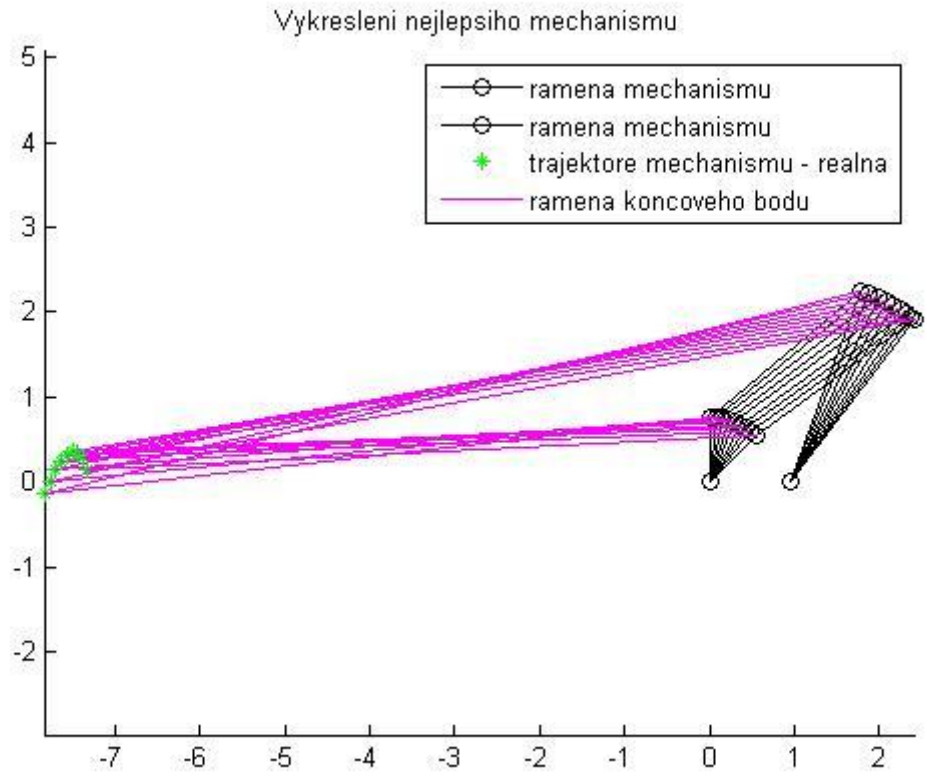
Algoritmus je přiložen na CD jako *fmins_mereni.m*.



Obr. 4.10 Závislost přesnosti počátečního odhadu na přesnost řešení fminsearch



Obr. 4.11 Přesné řešení fminsearch



Obr. 4.12 Disproporční mechanismus získaný pomocí fminsearch

4.2.2.2. Fmincon v kombinaci s MNČ

Fmincon využívá jako KF metodu MNČ. Její oblast hledání minima je omezena (viz 2.3.2.) tak, abychom výpočet zrychlili a upřesnili. Já jsem volil tyto podmínky:

$$0 \leq g * \min(|L_1|, |L_2|, |L_3|, |d|) - \max(|L_1|, |L_2|, |L_3|, |d||P_x||P_y|) \qquad (4.10)$$

$$g \ldots maximální\ tolerovaný\ poměr$$

$$\{|L_1|, |L_2|, |L_3|, |d|, |P_x|, |P_y|\} \leq h \qquad (4.11)$$

$$h \ldots maximální\ tolerovaný\ rozměr$$

(4.10) je variací podmínek v 4.1.2. Rozdílem mezi nimi je, že se zde omezuje oblast hledání, takže musí být přímo respektována. Je zapsána v podobě nelineární podmínky jako další *mfile*. V mém experimentu nemusí optimalizace docílit konkrétních parametrů, proto volím obecně $g = 6, h = 8$. Maximální tolerovaný rozměr pro naši trajektorii na intervalu $\langle 0; 0{,}5 \rangle$ dostačuje několikanásobně a maximální poměr opticky splňuje realizovatelný mechanismus.

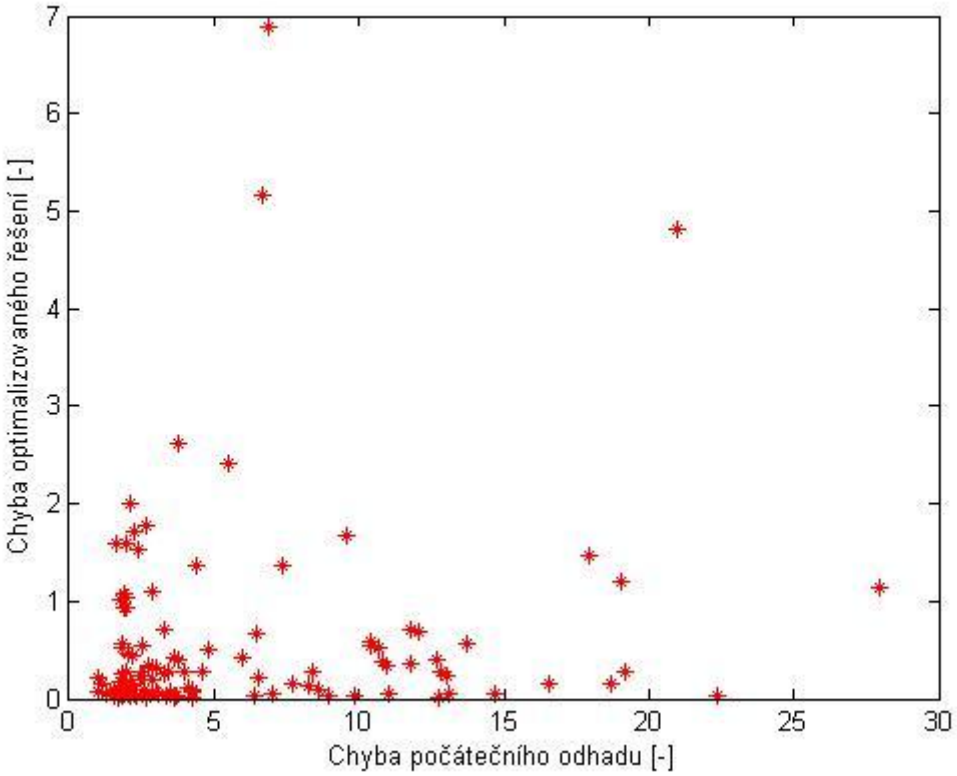
Při optimalizaci jsem zvolil vzorek řešitelných počátečních odhadů, které byly vygenerovány pomocí MC. Na výstupu dosáhl algoritmus následujících hodnot:

- Počet vyhodnocovaných nástřelů: 119
- Celkový čas: 418,5550 s
- Průměrný čas: 3,5173 s
- Průměrná přesnost nástřelu: 5,6486
- Průměrná přesnost optimalizovaného řešení: 0,5562
- Nejlepší řešení: 0,0129; P_{ARM} = 2,6753; P_{MECH} = 1,3892
- Korelační koeficient: 0,1655

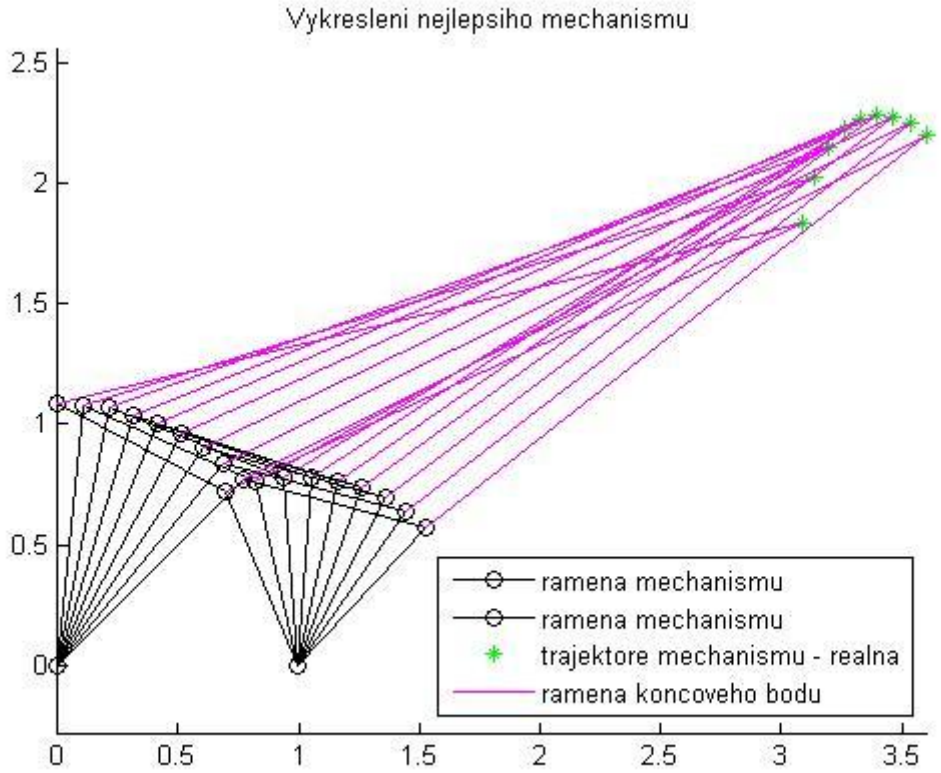
Vidíme, že v průměru dokázal *fmincon*+MNČ zpřesnit výsledek o jeden řád od prvotního odhadu. Nejlepší řešení se stále pohybuje o řád výše oproti ideálnímu řešení. Tato trajektorie je však v poměru k dále řešeným trajektoriím obtížně optimalizovatelná. Za povšimnutí stojí velmi krátký průměrný čas optimalizace 3,5173 s na jedno řešení, což je způsobeno omezením prohledávaného prostoru pomocí již zmíněných podmínek.

Hodnota korelačního koeficientu 0,1655 (o 61,78% vyšší než u *fminsearch*+MNČ s penalizacemi) nám říká, že je zde velmi slabá přímková závislost, která, pokud existuje, má kladnou směrnici. O rozložení bodů v grafu a důsledcích platí to samé co v předchozí kapitole.

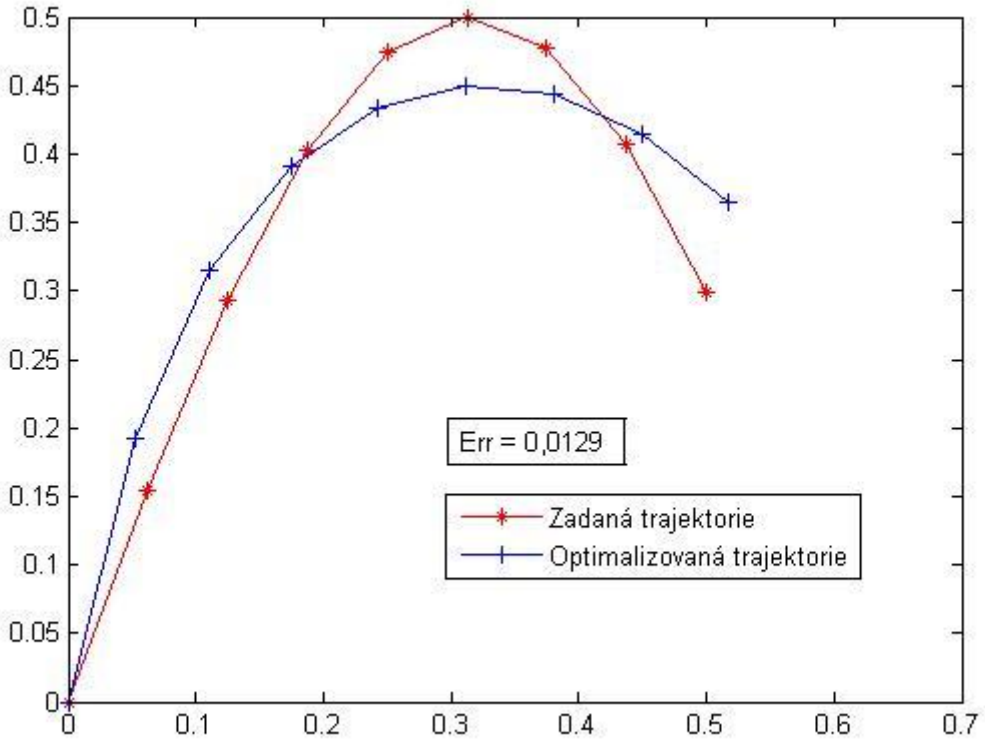
Algoritmus je přiložen na CD jako *fminc_mereni.m*.



Obr. 4.13 Závislost přesnosti počátečního odhadu na přesnost řešení fmincon+MNČ



0br. 4.14 Proporčně vyvážený mechanismus



Obr. 4.15 Nepřesné řešení fmincon+MNČ, lze zlepšit uvolněním podmínek či jiným počátečním odhadem

4.2.2.3. Srovnání fmincon+MNČ a fminsearch+MNČ s penalizacemi

Při stejné množině počátečních odhadů a popsaném nastavení můžeme o MATLAB funkcích říci:

- *Fmincon*+MNČ je asi 2x rychlejší při vyhodnocování
- Průměrná chyba *fmincon*+MNČ je o 21,5% nižší
- Ani jeden kombinovaný algoritmus nedokázala dokonale splnit zadání
    - *Fmincon*+MNČ navrhl nedostatečně přesnou trajektorii při splnění požadavků na proporčnost mechanismu
    - *Fminsearch*+MNČ s penalizacemi došel k velmi dobré trajektorii, přičemž ale nesplnil požadavky na proporčnost
- Nemůžeme říci, že přesnost optimalizace u obou kombinovaných algoritmů záleží pouze na přesnosti počátečního odhadu
- Optimalizace oběma kombinovanými algoritmy sníží hodnotu KF v průměru o 1 řád, u nejlepších řešení dosahujeme vylepšení o 2-3 řády, kdy $Err_{MNČ} \cong 10^{-3}$

4.2.3. Aplikační oblasti – možnosti úlohy a její omezení

V této kapitole nastíním možnost, jak optimalizovat mechanismus pro různé trajektorie, jejich parametry pro úspěšnou optimalizaci a nastavení optimalizačního algoritmu.

4.2.3.1. Popis univerzálního algoritmu

V optimalizačním algoritmu figuruje MC (nastaveno dle 4.2.1.1.) jako nástřel, neboť dosahuje rychle uspokojivých hodnot. MC hledá ve while-cyklu počáteční odhad, dokud nedosáhne hodnoty $Err_{MNČ} = 0{,}3$ která je definována uživatelem (obvykle empiricky zjištěná hranice možností). Pokud takto přesný odhad nenajde ani po 6 opakováních, volí se za počáteční odhad nejlepší řešení z posledního cyklu.

Nástřel řešení putuje do algoritmu *fmincon* v kombinaci s MNČ, který je nastaven dle kapitoly 4.2.2.2. Ten se pokusí o optimalizaci mechanismu pod nastavenou hodnotu tolerance. Přesné řešení se sice pohybuje okolo $Err_{MNČ} \cong 10^{-3}$, ale to funguje jen za splnění předpokladů z následující kapitoly (4.2.3.2.). Jinak je nastavení tolerance individuální.

V případě neúspěšné optimalizace se pokračuje v cyklu dalším nástřelem a novou optimalizací.

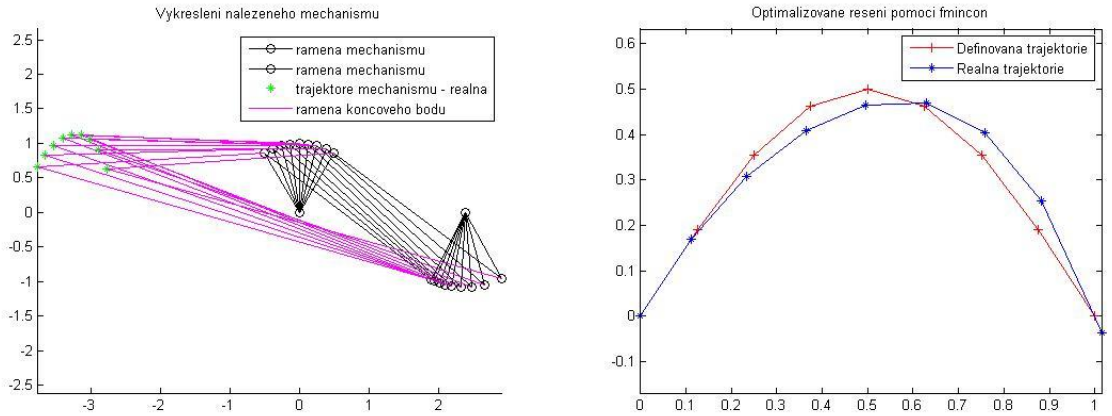
4.2.3.2. Omezení trajektorie pro optimalizaci

Základní nároky na trajektorii, u níž požadujeme dostatečnou přesnost:

- Je definována prostou funkcí
- Nemění tvar z konvexní na konkávní
- Nemění podstatně svou křivost (samotný poloměr křivosti nemá sám o sobě vliv, pokud dodržíme ostatní pravidla)
- Je v rozmezí $\langle 0; 1 \rangle$, a to z důvodu:
    - Nastavení tolerancí
        - Pro přesné řešení
        - Pro sestavení mechanismu (přímá kinematika (2.4) počítá matici $\boldsymbol{q_k}$, dokud hodnoty v matici $\boldsymbol{U_k}$ nejsou menší než 10^{-3})
    - Nastavení mezí velikosti mechanismu u algoritmu *fmincon*
- Má rovnoměrně distribuované body po délce

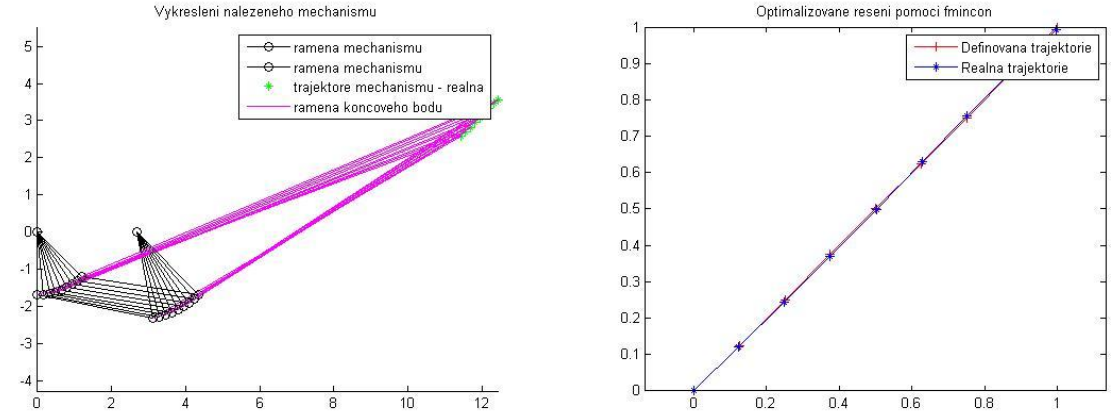
Akční úhel pro takto definovanou trajektorii má rozsah okolo $\frac{\pi}{4}$ ale jeho počátek může být definován různě. V následujících ukázkách jsem volil takové akční úhly, které po iteračním procesu metodou pokus-omyl umožnily nejlepší shodu trajektorie.

4.2.3.3. Optimalizované trajektorie



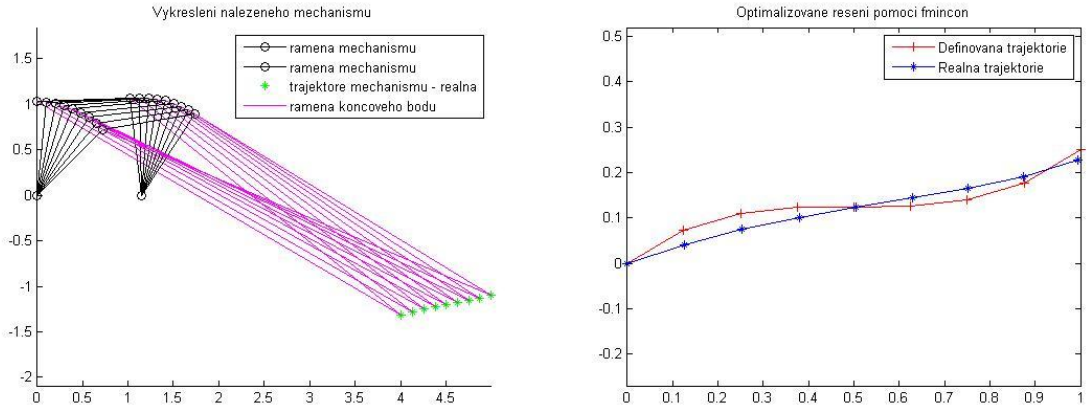
Obr. 4.16 Sinus není prostá funkce a podstatně mění svou křivost. Po 370 s iteračního výpočtu vychází opticky podobná křivka.

$$Err_{MNČ} = 12{,}93 \cdot 10^{-3}, q_1 = \langle \frac{2\pi}{3}; \frac{\pi}{3} \rangle,$$
$$(L_1 = 0{,}9996;\ L_2 = 3{,}0182;\ L_3 = 1{,}0850;\ d = 2{,}3849;\ P_x = -5{,}4849;\ P_y = -2{,}1475)$$



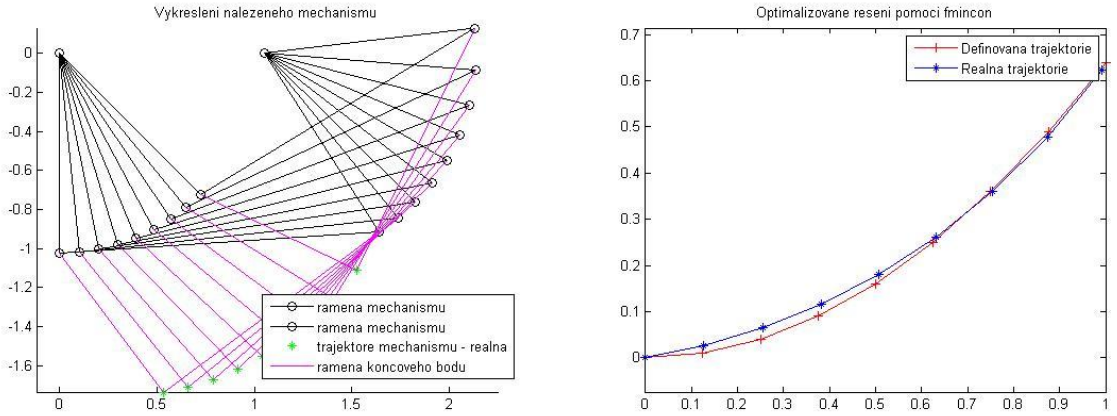
Obr. 4.17 Přímková dráha splňuje požadavky, algoritmus ji dokáže kopírovat s malou chybou.

$$Err_{MNČ} = 0{,}23 \cdot 10^{-3}, q_1 = \langle -\frac{\pi}{2}; -\frac{\pi}{4} \rangle,$$
$$(L_1 = 1{,}6905;\ L_2 = 3{,}1945;\ L_3 = 2{,}3666;\ d = 2{,}6988;\ P_x = 7{,}1492;\ P_y = 6{,}4624)$$



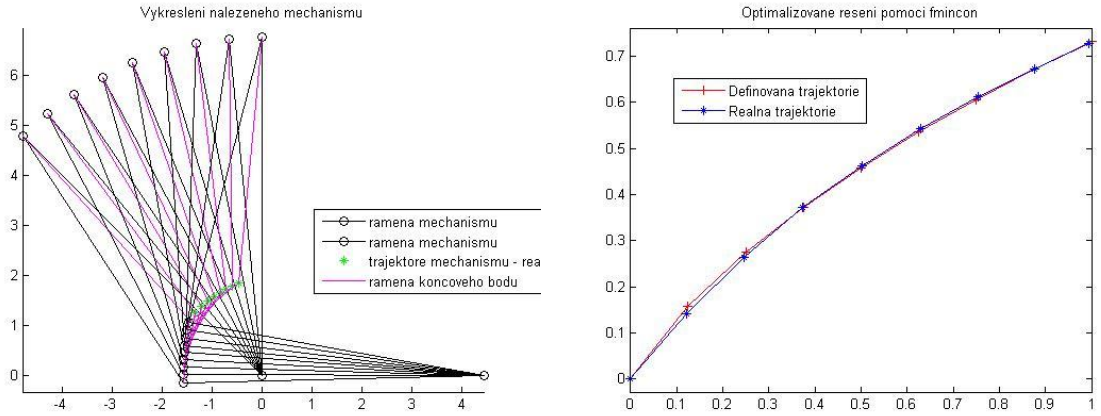
Obr. 4.18 Kubická parabola mění svůj tvar z konkávního na konvexní. V této situaci nedokáže algoritmus uspokojivě optimalizovat. Nízká hodnota KF odpovídá dobré synchronizaci v x-ových souřadnicích.

$$Err_{MNČ} = 4{,}48 \cdot 10^{-3}, q_1 = \langle \frac{\pi}{2}; \frac{\pi}{4} \rangle,$$
$$(L_1 = 1{,}0277;\ L_2 = 1{,}0277;\ L_3 = 1{,}0726;\ d = 1{,}1494;\ P_x = 2{,}8823;\ P_y = -2{,}4948)$$



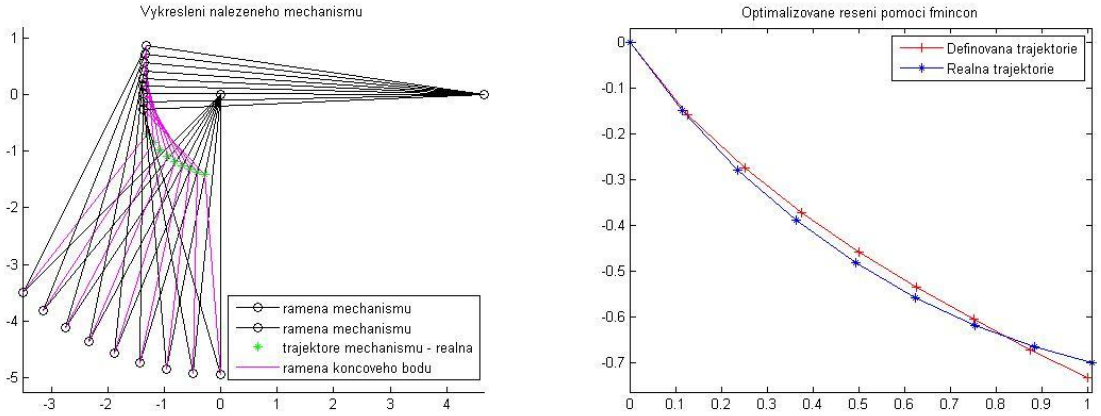
Obr. 4.19 Kvadratická parabola v 1. kvadrantu, jejíž osa je rovnoběžná s osou y

$$Err_{MNČ} = 2{,}80 \cdot 10^{-3}, q_1 = \langle -\frac{\pi}{2}; -\frac{\pi}{4} \rangle,$$
$$(L_1 = 1{,}0230;\ L_2 = 1{,}6441;\ L_3 = 1{,}0887;\ d = 1{,}0479;\ P_x = -1{,}1621;\ P_y = -0{,}7454)$$



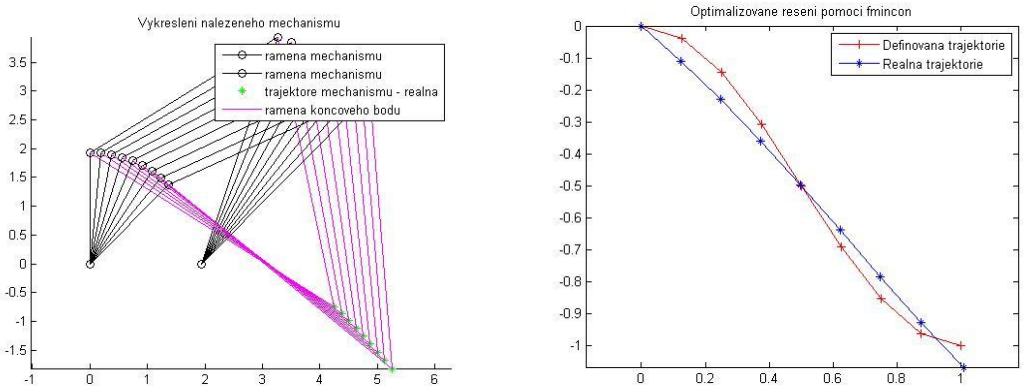
Obr. 4.20 Kvadratická parabola v 1. kvadrantu, jejíž osa je rovnoběžná s osou x.

$$Err_{MNČ} = 0{,}63 \cdot 10^{-3}, q_1 = \langle \frac{3\pi}{4}; \frac{\pi}{2} \rangle,$$
$$(L_1 = 6{,}7697;\ L_2 = 5{,}8915;\ L_3 = 6{,}0251;\ d = 4{,}4528;\ P_x = -1{,}0031;\ P_y = 0{,}7787)$$



Obr. 4.21 Kvadratická parabola ve 4. kvadrantu, jejíž osa je rovnoběžná s osou x.

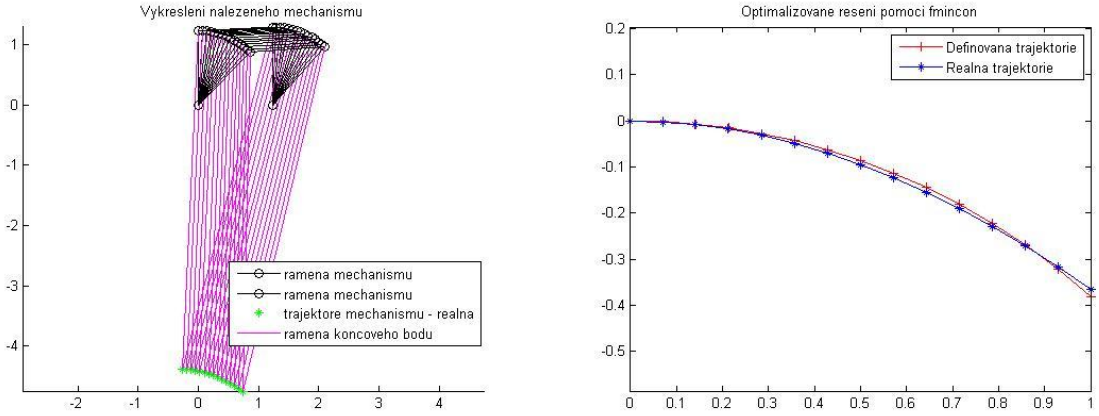
$$Err_{MNČ} = 3{,}29 \cdot 10^{-3}, q_1 = \langle -\frac{3\pi}{4}; -\frac{\pi}{2} \rangle,$$
$$(L_1 = 4{,}9464;\ L_2 = 4{,}8790;\ L_3 = 6{,}0344;\ d = 4{,}6577;\ P_x = -1{,}3977;\ P_y = -0{,}7141)$$



Obr. 4.22 Kosinusoida mění stejně jako kubická parabola konkávnost na konvexnost.

$$Err_{MNČ} = 28{,}13 \cdot 10^{-3}, q_1 = \langle \frac{\pi}{2}; \frac{\pi}{4} \rangle,$$

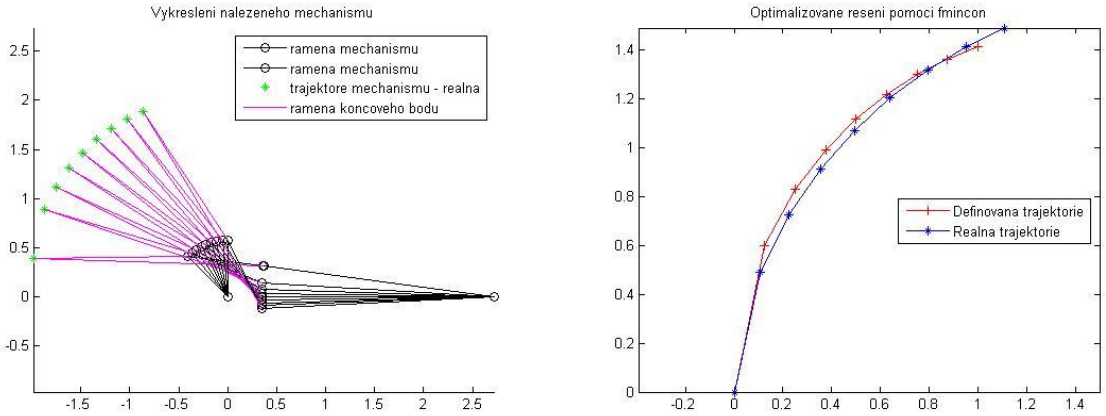
$$(L_1 = 1{,}9351;\ L_2 = 3{,}8384;\ L_3 = 4{,}1614;\ d = 1{,}9351;\ P_x = -1{,}6160;\ P_y = -4{,}5123)$$



Obr. 4.23 Konkávní část kružnice v 4. kvadrantu.

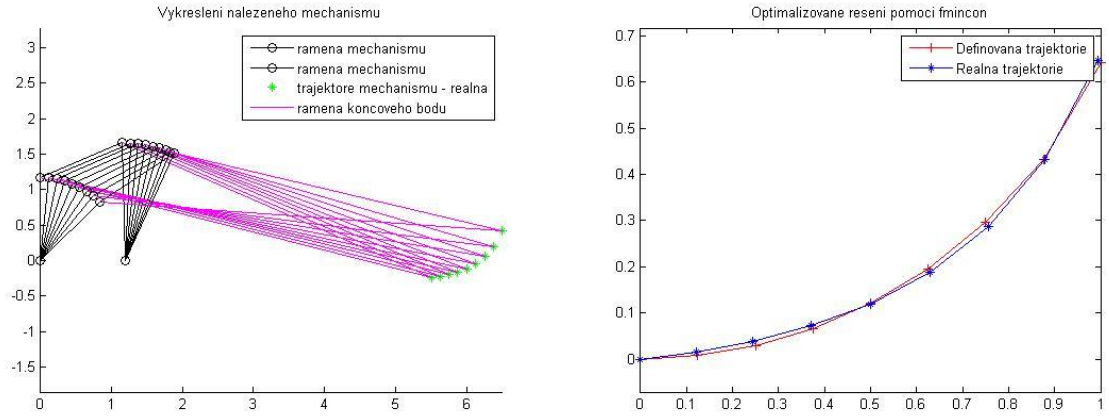
$$Err_{MNČ} = 0{,}91 \cdot 10^{-3}, q_1 = \langle \frac{\pi}{2}; \frac{\pi}{4} \rangle,$$

$$(L_1 = 1{,}2289;\ L_2 = 1{,}2289;\ L_3 = 1{,}2893;\ d = 1{,}2319;\ P_x = -1{,}7631;\ P_y = -5{,}5913)$$



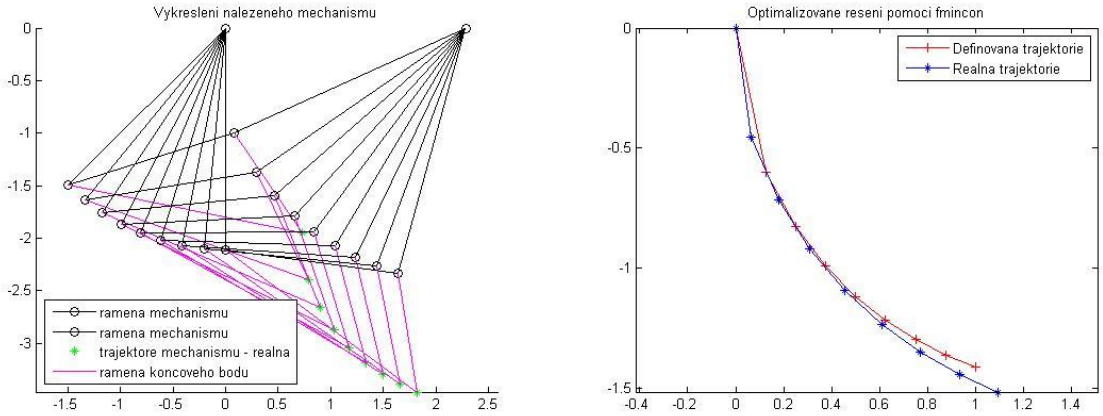
Obr. 4.24 Konkávní část kružnice v 1. kvadrantu. Nepravidelná distribuce bodů způsobuje vysokou chybu.

$$Err_{MNČ} = 62{,}11 \cdot 10^{-3}, q_1 = \langle \frac{\pi}{2}; \frac{\pi}{4} \rangle,$$
$$(L_1 = 0{,}5784;\ L_2 = 0{,}7787;\ L_3 = 2{,}3748;\ d = 2{,}7187;\ P_x = -2{,}3355;\ P_y = -0{,}1940)$$



Obr. 4.25 Konvexní část kružnice v 1. kvadrantu.

$$Err_{MNČ} = 0{,}48 \cdot 10^{-3}, q_1 = \langle \frac{\pi}{2}; \frac{\pi}{4} \rangle,$$
$$(L_1 = 1.1764;\ L_2 = 1.2489;\ L_3 = 1.6591;\ d = 1.1920;\ P_x = 3.2857;\ P_y = -3.4244)$$



Obr. 4.26 Konkávní část kružnice ve 4. kvadrantu. Nepravidelná distribuce bodů.

$$Err_{MNČ} = 88{,}02 \cdot 10^{-3}, q_1 = \langle -\frac{3\pi}{4}; -\frac{\pi}{2} \rangle,$$

$$(L_1 = 2{,}1142;\ L_2 = 1{,}6512;\ L_3 = 2{,}4176;\ d = 2{,}2827;\ P_x = 0{,}3281;\ P_y = -1{,}0983)$$

5. Závěr

V rešeršní části (2.2.) byla dle zadání okrajově popsána problematika kinematické úlohy optimalizace délek čtyřčlenného mechanismu pro dosažení definované trajektorie pomocí matic. Ukázala se nutnost iteračního procesu, proto byl k řešení využit program MATLAB.

V další kapitole práce (2.3.) uvádím zkoumané optimalizační algoritmy, jejich principy a uživatelskou stránku. Tyto algoritmy následně prošly testem, který zohledňoval časovou náročnost, přesnost a u funkcí MATLAB *optimization toolboxu* i citlivost na počáteční odhad. Zjištění dokázala, že chyba tohoto odhadu nehraje zásadní roli ve výsledku optimalizace.

Na základě nasbíraných dat byl vytvořen a nastaven algoritmus, který je schopen generovat rozměry čtyřčlenného mechanismu pro dosažení požadované trajektorie. Ten bohužel nedokázal uspokojivě optimalizovat mechanismus tak, aby kopíroval všechny druhy trajektorií. Proto úspěšnou optimalizaci je nutné, aby trajektorie splňovala požadavky na „tvar“.

Například trajektorie, která se opakuje ve smyčce, je zajímavým námětem pro další práci. K tomu zajisté přispěje odkrytí všech parametrů majících na optimalizaci vliv.

6. Literatura

[1] MYSZKA, David H. *Machines and mechanisms* —4th ed. Dayton: Patience Hall, 2011, 576 stran. ISBN-13: 978-0-13-215780-3

[2] GREPL, Robert. *Kinematika a dynamika mechatronických systémů*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2007, 158 s. ISBN 978-80-214-3530-8.

[3] SLABEJOVÁ, Danica. *Metoda Monte Carlo* [online]. 2007 [cit. 2015-05-15]. Bakalářská práce. Masarykova univerzita, Přírodovědecká fakulta. Vedoucí práce Gejza Wimmer. Dostupné z: <http://is.muni.cz/th/151336/prif_b/>.

[4] HERMAWANTO, Denny. *Genetic Algorithm for Solving Simple Mathematical Equality Problem* [online]. 2013 [cit. 2015-05-15]. Indonesian Institute of Sciences (LIPI). Dostupné z: <http://arxiv.org/ftp/arxiv/papers/1308/1308.4675.pdf >.

[5] Mathworks, Inc. MATLAB Help Files - *Unconstrained Nonlinear Optimization Algorithms*. MATLAB.com. [online]. 1.5.2015 [cit. 2015-5-15]. Dostupné z: http://www.mathworks.com/help/optim/ug/unconstrained-nonlinear-optimization-algorithms.html

[6] Mathworks, Inc. MATLAB Help Files - *Constrained Nonlinear Optimization Algorithms*. MATLAB.com. [online]. 1.5.2015 [cit. 2015-5-15]. Dostupné z: http://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html

[7] Mathworks, Inc. MATLAB Help Files – *Overview of Performance Timing Functions*. MATLAB.com. [online]. 1.5.2015 [cit. 2015-5-18]. Dostupné z: http://www.mathworks.com/help/MATLAB/MATLAB_prog/analyzing-your-programs-performance.html

[8] ALHAJJ , Ahmed Saeed Mohsen Alhajj. *Path synthesi sof four-bar mechanism using harmony optimization* [online]. 2011 [cit. 2015-05-22]. Bachelor Thesis. Department of Mechanical Engineering National Institute of Technology Rourkela. Under the Guidance of Prof. J. Srinivas. Dostupné z: http://ethesis.nitrkl.ac.in/2409/1/the_final_thesis.pdf

[9] KARPÍŠEK, Zdeněk. *Matematika IV*. 4., přeprac. vyd. Brno: Akademické nakladatelství CERM, 2014, 171 s. ISBN 978-80-214-4858-2.