

České vysoké učení technické v Praze
Fakulta strojní
Ústav mechaniky, biomechaniky a mechatroniky



Dynamické schopnosti robota zachytit letící předmět ve vesmíru

Diplomová práce

Ondřej Salva

Specializace: Robotika
Vedoucí práce: prof. Ing. Michael Valášek, DrSc.

2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Salva** Jméno: **Ondřej** Osobní číslo: **482402**
Fakulta/ústav: **Fakulta strojní**
Zadávající katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Robotika a výrobní technika**
Specializace: **Robotika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Dynamické schopnosti robota zachytit letící předmět ve vesmíru

Název diplomové práce anglicky:

Dynamic capabilities of a robot to capture a flying object in space

Pokyny pro vypracování:

Výpočet realizovatelnosti zachycení a zpomalení letícího předmětu ve vesmíru s ohledem na dynamické limity robota.

1. Formulace úlohy zachycení letícího předmětu robotem
2. Formulace úlohy určení možného zpomalení robotem na trajektorii v jeho pracovním prostoru
3. Výpočet dynamických schopností robota na trajektorii v jeho pracovním prostoru
4. Návrh vhodné záchytové křivky
5. Výpočet dosažitelného zachycení pro vhodný robot
6. Optimalizace záchytové trajektorie s cílem maximální rychlosti zachytávaného předmětu
7. Zhodnocení možnosti záchytu předmětu

Seznam doporučené literatury:

- Valášek, M.: Dynamic time parametrization of manipulator trajectories, *Kybernetika* 23(1987), 2, 154-174
- Sciacivco, L., Siciliano, B.: *Modeling and control of robot manipulators*, McGraw Hill, 1996
- Stejskal, V., Valasek, M.: *Kinematics and dynamics of machinery*, Marcel Dekker, New York, 1996

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Michael Valášek, DrSc. odbor mechaniky a mechatroniky FS

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

Datum zadání diplomové práce: **27.04.2023**

Termín odevzdání diplomové práce: **14.08.2023**

Platnost zadání diplomové práce: _____

prof. Ing. Michael Valášek, DrSc.
podpis vedoucí(ho) práce

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

doc. Ing. Miroslav Španiel, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a že jsem uvedl v příloženém seznamu veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací, vydaným ČVUT v Praze 1.7.2009.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne

.....
Ondřej Salva

Poděkování

Tímto bych rád poděkoval vedoucímu práce prof. Ing. Michaelu Valáškoví, DrSc. za cenné podněty a rady při řešení diplomové práce.

Další poděkování patří mé rodině a nejbližším za podporu a trpělivost po celou dobu studia.

Ondřej Salva

Název práce:

Dynamické schopnosti robota zachytit letící předmět ve vesmíru

Autor: Ondřej Salva

Studijní program: Robotika a výrobní technika

Specializace: Robotika

Druh práce: Diplomová práce

Akademický rok zhotovení: 2022/2023

Univerzita: Fakulta strojní, ČVUT

Ústav:

Ú12105 - Ústav mechaniky, biomechaniky a mechatroniky

Vedoucí práce: prof. Ing. Michael Valášek, DrSc.

Abstrakt:

Diplomová práce je zaměřena na zhodnocení schopnosti zvoleného robotického manipulátoru zachytit letící předmět ve vesmíru. Pro vybraný manipulátor je vyvinut algoritmus, který dává ucelenou představu o možnosti provedení úlohy a umožňuje následnou optimalizaci parametrů robota a trajektorie jeho pohybu. Je vypočtena realizovatelnost zachycení předmětu pro zvolené parametry manipulátoru. Následně je provedena optimalizace parametrů a výsledky realizovatelného zachycení pro optimalizované parametry jsou porovnány s výsledky původními. Na závěr je správnost výpočetního algoritmu ověřena pomocí simulace pohybu v prostředí Matlab-Simulink.

Klíčová slova:

robotické zachytávání, robot ve vesmíru, dynamické možnosti robota, zachytávání předmětu

Title:

Dynamic capabilities of a robot to capture a flying object in space

Author: Ondřej Salva

Study program: Robotics and Production Technology

Specialization: Robotics

Thesis type: Master's thesis

Academic year: 2022/2023

University: Faculty of Mechanical Engineering, CTU

Department:

U12105 - Department of Mechanics, Biomechanics and Mechatronics

Supervisor: prof. Ing. Michael Valášek, DrSc.

Abstract:

The thesis is focused on analysis of the ability of a chosen robotic manipulator to capture a flying object in space. An algorithm is developed to give us a complex image of the realizability of the task. The algorithm allows further optimization of the trajectory of the end-effector and parameters of the robot. The realizable capturing motion is calculated for chosen parameters of the manipulator. The parameters are then optimized and the results before and after optimization are compared. Finally, the correctness of the developed algorithm is verified by simulation in Matlab-Simulink.

Key words:

Robotic Capture, Robot in Space, Dynamic Capabilities of a Robot, Object Capture

Obsah

1	Úvod	1
2	Cíle práce	2
3	Stav problematiky	3
3.1	Definice problému	3
3.2	Úloha globální dynamiky	4
3.3	Určení časové parametrizace	7
3.3.1	Vyjádření omezení manipulátoru	7
3.3.2	Rychlostní profil	8
3.3.3	Metody řešení časově optimální parametrizace	10
4	Analytické výpočty	12
4.1	Volba záchytové křivky	13
4.2	Inverzní kinematika	15
4.3	Výpočet parciálních derivací	17
4.4	Rovnice dynamiky	20
4.5	Limity pohonů	23
4.6	Shrnutí výpočtu	28
4.6.1	Omezení zrychlení	28
4.6.2	Omezení rychlostí	29
5	Algoritmizace	31
5.1	Princip algoritmu	32
5.2	Numerická integrace	32
5.2.1	Dopředná integrace	33

5.2.2	Zpětná integrace	34
5.3	Výpočet konstantní části profilu	35
5.4	Výpočet limitní křivky rychlosti	38
5.5	Řešení nulového dělení	39
5.6	Vykreslení požadovaných veličin	39
6	Optimalizace	44
6.1	Funkce rychlostního profilu	44
6.2	Surrogate optimalizace	45
6.3	Optimalizace záchytové úsečky	47
6.4	Optimalizace dynamiky	49
6.4.1	Optimalizace momentů pro dosažení zadané rychlosti	49
6.4.2	Optimalizace rozměrů robotu	52
6.5	Zhodnocení	54
7	Ověření výpočtu	56
8	Závěr	58
	Seznam použitých zdrojů	60
	Seznam obrázků	62
	Příloha	64
	Použitý software	88

Kapitola 1

Úvod

Robotické manipulátory mají v dnešní době využití v mnoha oborech, od velkosériových průmyslových výrob po moderní lékařství. Není tedy překvapivé, že svůj účel nacházejí i v oboru vesmírného výzkumu, pro který je přirozená potřeba moderních technologií a který často svými specifickými podmínkami podněcuje další vědecký pokrok.

Při objevování vesmíru je často téměř nemožné svěřit některé úkony člověku, proto je výzkum motivován k sestrojení strojů různých typů, které jsou schopny specifické potřeby splnit. Může se jednat o průzkumná vozítka, tensegrity struktury či klasické sériové roboty, které mohou být umístěny na vesmírných stanicích. Jedním příkladem využití sériového robotu se bude zabývat tato práce.

Motivace

Motivací práce je nalezení způsobu zhodnocení realizovatelnosti zachycení letících předmětů ve vesmíru pomocí robotu umístěného na vesmírné stanici, či plavidle. Předmět letící vesmírem je třeba uchopit a zpomalit takovým způsobem, aby při tomto pohybu nebyla překročena zvolená omezení robotu.

Kapitola 2

Cíle práce

Vlastní práce se zabývá naplněním následujících cílů:

1. Formulace úlohy zachycení letícího předmětu robotem.
2. Formulace úlohy určení možného zpomalení robotem na trajektorii v jeho pracovním prostoru
3. Výpočet dynamických schopností robota na trajektorii v jeho pracovním prostoru
4. Návrh vhodné záchytové křivky
5. Výpočet dosažitelného zachycení pro vhodný robot
6. Optimalizace záchytové trajektorie s cílem maximální rychlosti zachytávaného předmětu
7. Zhodnocení možnosti záchytu předmětu

Kapitola 3

Stav problematiky

Robotické manipulátory jsou vhodným prostředkem k manipulaci s předměty v případech, kdy by požadované úkony byly pro člověka jen těžce splnitelné. V průmyslových výroбах se může jednat o manipulaci s těžkými předměty v nebezpečných prostředích nebo provádění jednoduchých monotónních úkonů po dlouhou dobu ve vysoké frekvenci. Je tedy přirozené, že našly tyto stroje uplatnění i ve vesmírném průmyslu. Příkladem mohou být roboty Canadarm (Obr. 3.1), které jsou již po několik desetiletí využívány k různorodým manipulačním aplikacím ve vesmírném prostředí [1].



Obr. 3.1: Manipulátor na vesmírné stanici [1]

3.1 Definice problému

Tato práce se zabývá případem využití robotického manipulátoru pro zachycení volně letícího předmětu. Předmětem by mohlo být například náradí, nebo vesmírné smetí, které by bylo potřeba odstranit z okolí vesmírné stanice. Cílem je provedení výpočtu, který by poskytoval ucelenou představu o realizovatelnosti úlohy zachycení a zpomalení letícího předmětu pro zvolený robotický manipulátor.

Prvním krokem je úvaha o samotném způsobu realizace zachycení letícího tělesa. Uvažujeme těleso, které se ve vesmíru pohybuje kolem vesmírné stanice (nebo plavidla) rovnoměrným pohybem. K vesmírné stanici (nebo plavidlu) je ukotven robotický manipulátor s omezeným pracovním prostorem. Pro zachycení je nutné, aby se dostatečně velká část trajektorie tělesa nacházela v pracovním prostoru robotu a aby byl na této trajektorii robot schopný provést zachycení tělesa v souladu se svými dynamickými omezeními.

Prvním způsobem realizace zachycení by mohl být případ, kdy by se robot pouze nastavil do pozice, ve které by se jeho end-effector nacházel na trajektorii letícího předmětu podobně jako při chytání míčku do nádoby. Samotné zachycení by poté proběhlo při kontaktu tělesa s end-effectorem. Nevýhodou tohoto přístupu je neplynulé zpomalování tělesa a vzniklý ráz při kontaktu tělesa s end-effectorem. Tímto způsobem by mohlo dojít k poškození manipulátoru nebo tělesa. Dále by takto docházelo k velikému namáhání robotu z důvodu rychlého zpomalení tělesa. Z pohledu dynamických možností tedy toto řešení nepovažujeme za vhodné.

Vhodnějším řešením je plánování pohybu end-effectoru tak, aby byla v momentě zachycení relativní rychlost tělesa vůči end-effectoru nulová a nedocházelo tak k rázu. To znamená zrychlení end-effectoru na rychlost zachytávaného tělesa, provedení zachycení a následné zpomalení. Celý pohyb je nutné naplánovat s ohledem na zvolená omezení robotu. Křivka, po které se pohybuje end-effector, je předem vhodně zvolena. Jedná se o problém hledání realizovatelného pohybu robotu po známé křivce, tedy řešení úlohy globální dynamiky [2].

V této kapitole budou shrnuty nástroje potřebné pro výpočet globální dynamiky a řešení vhodné časové parametrizace. Následně aplikujeme vybrané metody na případ zachycení letícího tělesa ve vesmíru.

3.2 Úloha globální dynamiky

Úloha globální dynamiky slouží k získání přehledu o dynamických možnostech stroje na známé křivce při daných parametrech a omezeních [2]. Prvním krokem výpočtu je popsání polohy end-effectoru v pracovním prostoru.

Předpokládejme parametr p popisující známou trajektorii end-effectoru manipulátoru v kartézských souřadnicích x a y od počátečního do koncového bodu (uvažujeme pouze polohu v rovině). Parametr nabývá v čase rostoucí hodnoty od 0 do p_{max} . Kartézské souřadnice polohy end-effectoru lze sloučit do radiusvektoru R [2],[3]:

$$\mathbf{R}(p) = [x(p), y(p)]^T \quad p \in \langle 0, p_{max} \rangle \quad (3.1)$$

Úkolem je nalézt vhodný časový průběh parametrizace křivky tak, aby byla splněna dynamická omezení manipulátoru [2],[3]:

$$p = p(t) \quad (3.2)$$

Mezi kartézskými (3.1) a kloubovými souřadnicemi $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ platí vztah [2],[3]:

$$\mathbf{R}(p) = \mathbf{K}(\boldsymbol{\theta}(p)) \quad (3.3)$$

Kloubové souřadnice lze poté vyjádřit ze známých kartézských souřadnic pomocí inverzní kinematiky [2],[3]:

$$\boldsymbol{\theta}(p) = \mathbf{K}^{-1}(\mathbf{R}(p)) \quad (3.4)$$

Kromě kinematické závislosti mezi kartézskými a kloubovými souřadnicemi je třeba sestavit rovnice dynamiky. Ty lze obecně vyjádřit ve tvaru [3]:

$$\mathbf{n} = \mathbf{I}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}}^T \mathbf{C}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \mathbf{V}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} + \mathbf{g}(\boldsymbol{\theta}) \quad (3.5)$$

Kde:

\mathbf{n} je n -dimenzionální vektor momentů v pohonech manipulátoru

$\mathbf{I}(\boldsymbol{\theta})$ je (n, n) matice setrvačnosti

$\mathbf{C}(\boldsymbol{\theta})$ je (n, n, n) tenzor Coriolisových a odstředivých sil

$\mathbf{V}(\boldsymbol{\theta})$ je (n, n) matice třecích sil

$\mathbf{g}(\boldsymbol{\theta})$ je n -dimenzionální vektor gravitačních sil

Limity momentů v pohonech jsou uvažovány závislé na úhlové rychlosti [2],[3],[4]:

$$\mathbf{n}^- - \mathbf{F}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \leq \mathbf{n} \leq \mathbf{n}^+ - \mathbf{F}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad (3.6)$$

Časové derivace kloubových souřadnic v rovnicích dynamiky (3.5) lze nahradit parciálními derivacemi podle parametru p a časovými derivacemi parametru \dot{p}, \ddot{p} [2],[3]:

$$\dot{\boldsymbol{\theta}} = \frac{\partial \mathbf{K}^{-1}(\mathbf{R})}{\partial p} \dot{p} = \boldsymbol{\theta}' \cdot \dot{p} \quad (3.7)$$

$$\ddot{\boldsymbol{\theta}} = \frac{\partial \mathbf{K}^{-1}(\mathbf{R})}{\partial p} \ddot{p} + \frac{\partial^2 \mathbf{K}^{-1}(\mathbf{R})}{\partial p^2} \dot{p}^2 = \boldsymbol{\theta}' \cdot \ddot{p} + \boldsymbol{\theta}'' \cdot \dot{p}^2 \quad (3.8)$$

Tyto vztahy je možné dosadit do rovnic dynamiky (3.5). Při zavedení substituce $\dot{p} = d_1$ a $\ddot{p} = d_2$ dostáváme nerovnici [3]:

$$\mathbf{n}^- - \mathbf{g}(\boldsymbol{\theta}) \leq \mathbf{A}d_1^2 + \mathbf{B}d_1 + \mathbf{D}d_2 \leq \mathbf{n}^+ - \mathbf{g}(\boldsymbol{\theta}) \quad (3.9)$$

Kde:

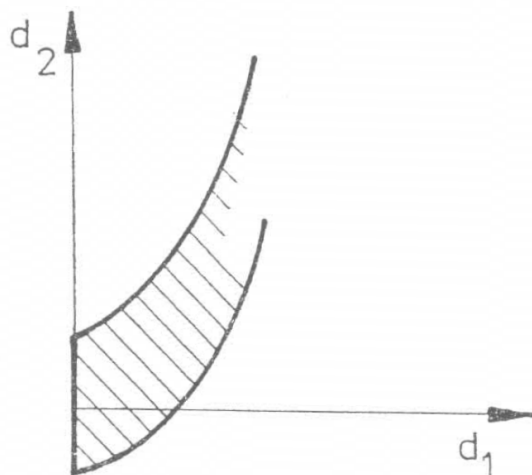
$$\mathbf{A} = \mathbf{I}(\boldsymbol{\theta})\boldsymbol{\theta}'' + (\boldsymbol{\theta}')^T \mathbf{C}(\boldsymbol{\theta})\boldsymbol{\theta}' \quad (3.10)$$

$$\mathbf{B} = (\mathbf{V}(\boldsymbol{\theta}) + \mathbf{F}(\boldsymbol{\theta}))\boldsymbol{\theta}' \quad (3.11)$$

$$\mathbf{D} = \mathbf{I}(\boldsymbol{\theta})\boldsymbol{\theta}' \quad (3.12)$$

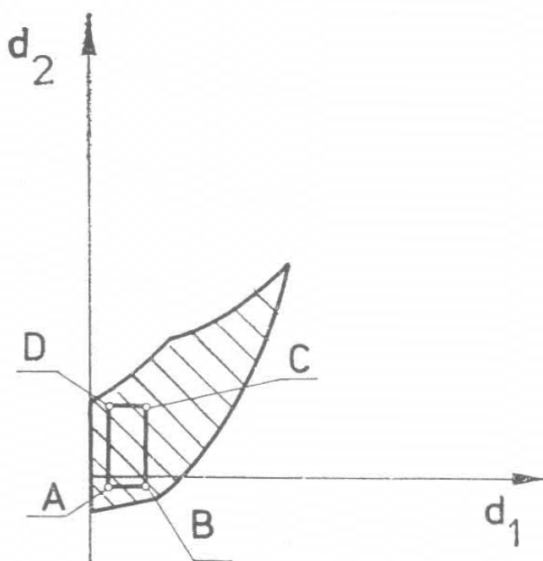
Pro každý pohon zvoleného manipulátoru lze poté v místě křivky daném hodnotou parametru p sestavit diagram realizovatelných hodnot $d_1 - d_2$. Realizovatelná oblast

hodnot je podle (3.9) omezena dvěma kvadratickými rovnicemi, jak je zobrazeno na Obr. 3.2 [3].



Obr. 3.2: Realizovatelná oblast pohonu [3]

Tuto oblast lze sestavit pro každý pohon. Pokud chceme znát realizovatelné hodnoty pro celý manipulátor, je nutné dílčí oblasti sjednotit do jednoho diagramu. Příklad pro RR manipulátor je zobrazen na Obr. 3.3 [3].

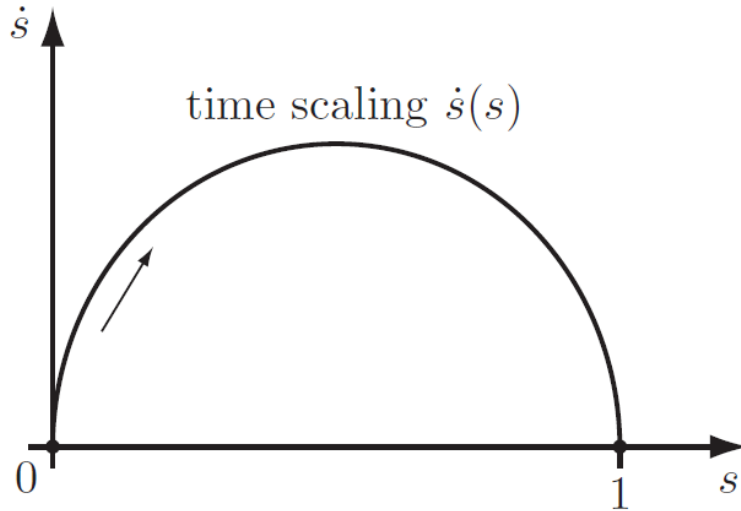


Obr. 3.3: Realizovatelná oblast manipulátoru [3]

Zobrazení je výhodné především v případech, kdy je potřeba zjistit dynamické schopnosti manipulátoru v konkrétních bodech trajektorie. Diagram dává názornou představu o realizovatelných hodnotách d_1, d_2 . Pokud chceme zjistit časový průběh $p(t)$, je vhodné použít metodu na podobném principu, jak je popsáno v následující části práce.

3.3 Určení časové parametrizace

Metoda hledání časové parametrizace využívá zobrazení v diagramu $p - \dot{p}$. Na horizontální ose zobrazujeme pomocí hodnoty parametru p pozici end-effectoru na křivce a na vertikální ose rychlost, která je vázána na hodnotu derivace \dot{p} . Pohyb po křivce je popsán jednosměrný, platí tedy $\dot{p} \geq 0$. V diagramu je hledána závislost, která popisuje průběh rychlosti po křivce, příklad je znázorněn na Obr. 3.4, kde je pro označení parametru použita proměnná s [5]. V tomto případě jde o průběh s nulovými rychlostmi v počátečním a koncovém bodě.



Obr. 3.4: Průběh parametrizace křivky [5]

3.3.1 Vyjádření omezení manipulátoru

K nalezení parametrizace je využita znalost realizovatelných hodnot zrychlení \ddot{p} v jednotlivých bodech křivky. Tyto limity je třeba vyjádřit z omezení manipulátoru podobně, jako bylo provedeno pro momenty pohonů v předchozí metodě [5].

Dynamická omezení uvažujeme stejná jako v globální dynamice [2],[3],[4]:

$$\mathbf{n}^- - \mathbf{F}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \leq \mathbf{n} \leq \mathbf{n}^+ - \mathbf{F}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} \quad (3.13)$$

Vztahy (3.7) a (3.8) dosadíme do rovnic dynamiky (3.5) a obdržíme rovnice [5],[6]:

$$\mathbf{n} = \mathbf{a}(p) \ddot{p} + \mathbf{b}(p) \dot{p}^2 + \mathbf{c}(p) \dot{p} + \mathbf{g}(p) \quad (3.14)$$

Kde:

$$\mathbf{a}(p) = \mathbf{I}(\boldsymbol{\theta}(p)) \boldsymbol{\theta}'(p) \quad (3.15)$$

$$\mathbf{b}(p) = \mathbf{I}(\boldsymbol{\theta}(p)) \boldsymbol{\theta}''(p) + \boldsymbol{\theta}'(p)^T \mathbf{C}(\boldsymbol{\theta}(p)) \boldsymbol{\theta}'(p) \quad (3.16)$$

$$\mathbf{c}(p) = (\mathbf{V}(\boldsymbol{\theta}(p)) + \mathbf{F}(\boldsymbol{\theta}(p))) \boldsymbol{\theta}'(p) \quad (3.17)$$

Dosazením rovnice (3.14) do vztahu (3.13) dostáváme pro každý pohon manipulátoru realizovatelný interval hodnot \ddot{p} [5],[6]:

$$L_i(p, \dot{p}) \leq \ddot{p} \leq U_i(p, \dot{p}) \quad (3.18)$$

Kde:

$$L_i(p, \dot{p}) = \frac{n_i^- \delta_i(p) + n_i^+ (1 - \delta_i(p)) - b_i(p) \dot{p}^2 - c_i(p) \dot{p} - g_i(p)}{a_i(p)} \quad (3.19)$$

$$U_i(p, \dot{p}) = \frac{n_i^+ \delta_i(p) + n_i^- (1 - \delta_i(p)) - b_i(p) \dot{p}^2 - c_i(p) \dot{p} - g_i(p)}{a_i(p)} \quad (3.20)$$

$$\textit{if } a_i(p) > 1 : \delta_i(p) = 1$$

$$\textit{if } a_i(p) < 0 : \delta_i(p) = 0$$

Hodnoty jednotlivých pohonů lze sjednotit do společného intervalu výpočtem maxima spodních omezení, respektive minima horních [5],[6]:

$$L(p, \dot{p}) \leq \ddot{p} \leq U(p, \dot{p}) \quad (3.21)$$

Kde:

$$L(p, \dot{p}) = \max_i \{L_i(p, \dot{p})\} \quad (3.22)$$

$$U(p, \dot{p}) = \min_i \{U_i(p, \dot{p})\} \quad (3.23)$$

Pokud by nastal případ, kdy platí $a_i(p) = 0$, nejsou pro daný pohon definovány limity $L_i(p, \dot{p})$, $U_i(p, \dot{p})$ a je třeba pro dané p zkontrolovat podmínku [5]:

$$n_i^- \leq b_i(p) \cdot \dot{p}^2 + c_i(p) \cdot \dot{p} + g_i(p) \leq n_i^+ \quad (3.24)$$

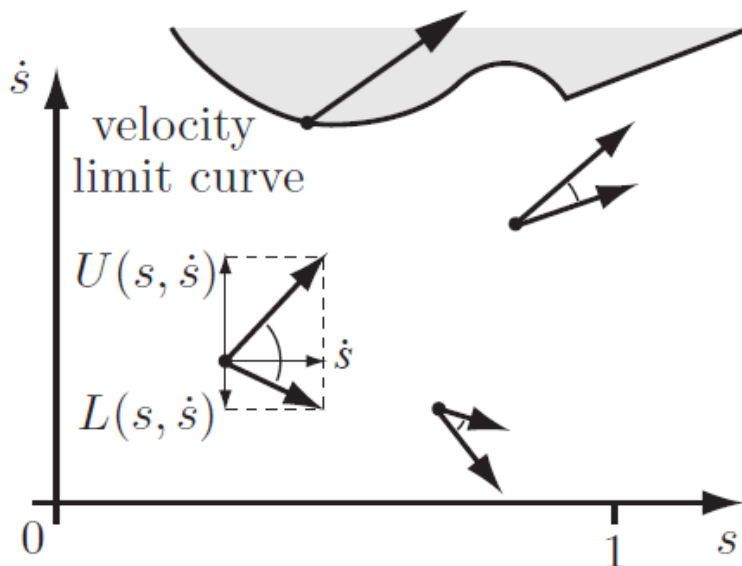
S vyjádřenými hodnotami realizovatelného zrychlení po zadané trajektorii je možné přejít k hledání vhodné časové parametrizace v $p - \dot{p}$ diagramu.

3.3.2 Rychlostní profil

Cílem je nalezení časové parametrizace $p(t)$ pro pohyb po zadané křivce v souladu se zvolenými omezeními. Prostředkem ke zjištění tohoto průběhu je hledání závislosti $\dot{p}(p)$ po délce křivky, tuto závislost nazveme rychlostním profilem [5].

Pohyb po křivce je popsán hodnotami parametru $p \in \langle 0, p_{max} \rangle$. Pro jednoduchost volíme $p_{max} = 1$. Jak již bylo zmíněno, uvažujeme pouze jednosměrný pohyb, na trajektorii tedy platí $\dot{p} \geq 0$. Podmínky pro \ddot{p} vyjadřují realizovatelná zrychlení a nejlépe jsou zobrazeny v konkrétním bodě diagramu. Pomocí hodnot $L(p, \dot{p})$ a $U(p, \dot{p})$ lze pro každý bod křivky s danou rychlostí end-effectoru sestavit realizovatelný kužel, ve kterém se může rychlostní profil nacházet, jak je zobrazeno na Obr. 3.5. Pokud

v diagramu nastane situace, kdy platí $L(p, \dot{p}) > U(p, \dot{p})$, kužel mizí a neexistují akční zásahy, které v tomto stavu udrží manipulátor na zvolené křivce. Část diagramu, kde platí $L(p, \dot{p}) > U(p, \dot{p})$, nazýváme nerealizovatelnou oblastí, na Obr. 3.5 je zobrazena šedou barvou. Hraniční křivku, kde platí $L(p, \dot{p}) = U(p, \dot{p})$, nazveme limitní křivkou rychlosti. Na této limitní křivce je realizovatelný kužel redukován do pouhého vektoru [5].



Obr. 3.5: Realizovatelný kužel v $s - \dot{s}$ diagramu [5]

Pro zajištění realizovatelnosti rychlostního profilu je nutné, aby jeho tečna v každém bodě křivky ležela uvnitř realizovatelného kužele. Při analýze dynamických schopností manipulátoru pro zachycení letícího předmětu je naším cílem nalezení průběhu, kdy by bylo možné zachytit co nejrychleji letící těleso. Tento průběh odpovídá časově optimální parametrizaci, pro kterou je cílem minimalizace kritéria času [5]:

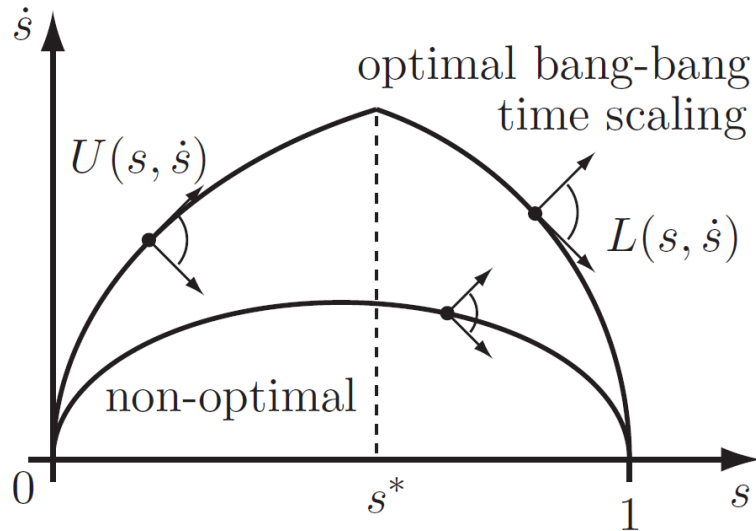
$$T = \int_0^T 1 dt \quad (3.25)$$

Toto kritérium lze převést na hledání průběhu s maximální rychlostí [5]:

$$T = \int_0^T 1 dt = \int_0^T \frac{dp}{\dot{p}} dt = \int_0^1 \frac{dt}{dp} dp = \int_0^1 \frac{1}{\dot{p}(p)} dp \quad (3.26)$$

Při hledání časově optimálního průběhu parametrizace hledáme pohyb, při kterém je manipulátor na limitu svých pohonů. Takový pohyb se v každém bodě diagramu pohybuje tečně k hranici realizovatelného kužele, jak je zobrazeno na Obr. 3.6. Jedná se o bang-bang pohyb s jedním přepínacím bodem mezi zrychlováním a zpomalováním. Pohyb, který není na limitech manipulátoru, je časově neoptimální [5].

Doposud zobrazované profily byly příkladem časově optimální parametrizace s jedním přepínacím bodem, která je použitelná v případě, že se celý profil nachází pod



Obr. 3.6: Časově optimální profil v $s - \dot{s}$ diagramu [5]

limitní křivkou rychlosti. Pokud by došlo k protnutí limitní křivky profilem, dospěl by manipulátor do neřiditelného stavu. V tomto případě by bylo nutné použít metody, které hledají časovou parametrizaci s více přepínacími body mezi akcelerací a decelerací. Příklad takového rychlostního profilu je zobrazen na Obr. 3.7.

3.3.3 Metody řešení časově optimální parametrizace

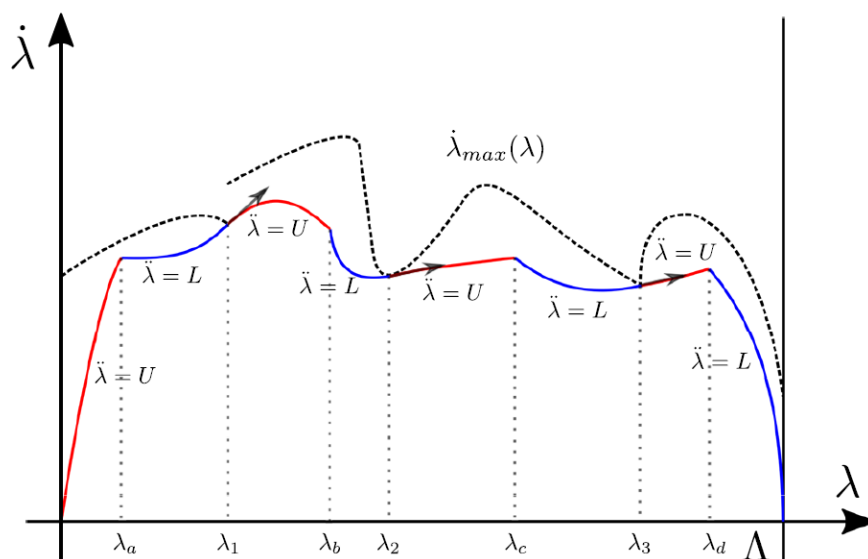
Hledání časové parametrizace, kdy se rychlostní profil pohybuje podél limitní křivky rychlosti je v robotice známým problémem a bylo navrženo několik způsobů jeho řešení. Cílem je nalezení takového profilu, který se k limitní křivce co nejvíce přibližuje a plní tak kritérium minimálního času. Příkladem je průběh zobrazený na Obr. 3.7, kde je možné vidět, že řešením je bang-bang řízení s přepínáním mezi maximální akcelerací a decelerací.

Jednotlivé metody řešení jsou dobře shrnuty v [7]. Prvním způsobem hledání časové parametrizace je využití dynamického programování. Metoda diskretizuje $p - \dot{p}$ diagram a využívá metod dynamického programování pro nalezení průběhu rychlostního profilu [7]. Tento postup je využit v práci [8].

Dalším způsobem je řešení pomocí konvexní optimalizace, která spočívá v rozdělení osy p na N segmentů a převedení původního problému na konvexní optimalizaci $O(N)$ proměnných vázaných pomocí $O(N)$ podmínek [7]. Příkladem takového přístupu je práce [9].

Nejčastějším způsobem řešení tohoto problému byla metoda hledání přepínacích bodů, která ve většině případů využívá přístup numerické integrace. Příkladem jsou práce [10],[11].

Po zhodnocení těchto dosavadních metod byl v práci [7] navržen robustní algoritmus pro řešení časově optimálního řízení, který byl následně zdokonalen v práci [12].



Obr. 3.7: Hledání přepínacích bodů [6]

Rozdílný přístup k problému nabízí práce [6], kde se autoři k nalezení optimální časové parametrizace rozhodli pro využití genetických algoritmů.

V disertační práci [13] je řešen problém časově optimálního řízení robotů s uvažováním dynamiky pohonů.

Kapitola 4

Analytické výpočty

Poznatky z předchozí kapitoly lze aplikovat na řešení zadaného problému. Cílem je provedení výpočtu, který by poskytoval představu o možnostech realizace zachycení a zpomalení letícího předmětu o známé hmotnosti pomocí zvoleného robotického manipulátoru. Výsledky této analýzy by následně bylo možné použít pro simulační, nebo fyzickou realizaci problému.

Uvažujeme, že se letící předmět pohybuje po trajektorii, kterou je ve vesmíru možné aproximovat přímkou. K jeho zachycení je nutné aby se end-effector robotického ramene pohyboval po určitou dobu, která je potřebná pro vykonání zachycení mechanismem end-effectoru, stejnou rychlostí a po stejné trajektorii jako letící předmět. Samotnou trajektorii end-effectoru robotu je tedy vhodné rozdělit na 3 části:

1. Zrychlování z počátečního bodu (nulová rychlost v počátečním bodě)
2. Vykonání zachycení předmětu (úsek s konstantní rychlostí)
3. Zpomalování do koncového bodu (nulová rychlost v koncovém bodě)

Jedná se o hledání rychlostního profilu, který by byl v prvním a posledním úseku časově optimální (hrana realizovatelného kužele je v těchto úsecích v každém bodě tečnou k profilu), což odpovídá profilu s maximální realizovatelnou rychlostí, a v úseku vykonávání zachycení předmětu měl konstantní rychlost rovnou rychlosti zachytávaného předmětu. Je nutné, aby se trajektorie nacházela v pracovním prostoru zvoleného manipulátoru a výsledný rychlostní profil musí vyhovovat zvoleným limitům vybraného manipulátoru, v našem případě budeme uvažovat:

1. Omezení momentů v pohonech
2. Omezení úhlové rychlosti pohonů
3. Omezení úhlového zrychlení pohonů

K původně zadaných omezením momentů je přidána možnost definice omezení úhlových rychlostí a zrychlení. V této práci nebude počítáno s omezením ryvu \ddot{p} , této

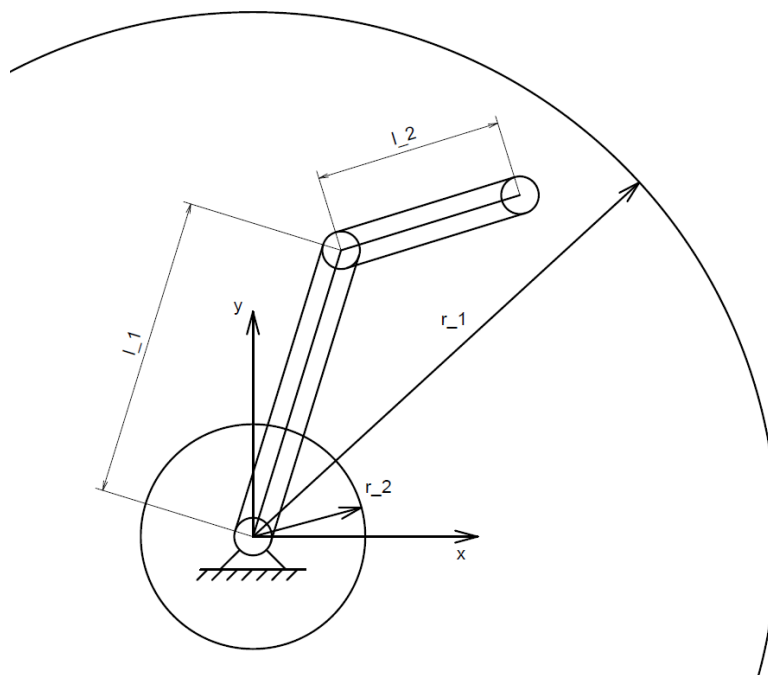
problematice se věnuje práce [13]. V našem případě tedy bude docházet ke skokovým změnám momentů a úhlových zrychlení pohonů.

Výpočet provedeme s uvažováním následujících předpokladů. Polohu robotu vůči trajektorii letícího tělesa lze ovlivnit pohybem plavidla, ke kterému je robot upevněn, nebo umístěním robotu na říditelný mechanismus. Dále uvažujeme, že hmotnost plavidla, ke kterému je robot upevněn, je dostatečně veliká a nedochází tak k ovlivnění jeho polohy pohybem robotu nebo setrvačností zachytávaného tělesa.

4.1 Volba záchytové křivky

Volba křivky, po které se bude end-effector pohybovat, je klíčová pro řešení celého problému. Toto rozhodnutí úzce souvisí s použitým typem manipulátoru, jelikož je křivka omezena tvarem jeho pracovního prostoru. Pro řešení úlohy volíme rovinný RR manipulátor především z důvodu dobré realizovatelnosti analytického výpočtu. V praxi by při zachycování ve vesmíru byla nutná schopnost orientace manipulátoru v prostoru, tuto volbu lze tedy vnímat jako zjednodušení problému, na které by bylo možné navázat obdobným výpočtem pro složitější mechanické struktury. Další možností by bylo skutečné použití RR manipulátoru za předpokladu, že by byl umístěn na říditelném mechanismu, který by umožňoval jeho orientaci v prostoru. V této orientaci by mohla pomoci i případná změna polohy či orientace uvažované vesmírné lodi, či stanice, na které by byl manipulátor umístěn, pokud by ovšem byla v dané situaci realizovatelná. Bylo by však nutné pro RR manipulátor použít minimálně jednu další osu, aby bylo umožněno natočení end-effectoru.

Příklad pracovního prostoru manipulátoru je zobrazen na Obr. 4.1.



Obr. 4.1: Pracovní prostor RR manipulátoru

Pokud neuvažujeme dorazy jednotlivých pohonů, je možné říci, že je pracovní prostor mezikružím s poloměry:

$$\begin{aligned} r_1 &= l_1 + l_2 \\ r_2 &= l_1 - l_2 \end{aligned} \tag{4.1}$$

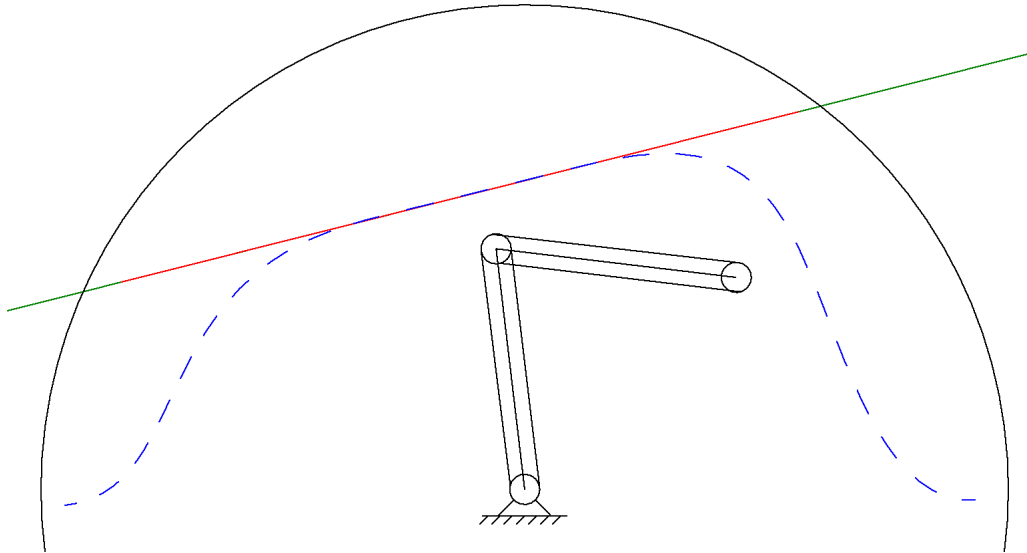
V případě, kdy platí $l_2 \geq l_1$, přechází mezikružím v celou kružnici s poloměrem r_1 . V tomto pracovním prostoru se musí nacházet každý bod křivky, kterou nazveme zachytová, zvolené pro pohyb end-effectoru. V práci uvažujeme možnost ovlivnění polohy robotu vůči trajektorii letícího předmětu. Tato skutečnost umožňuje budoucí optimalizaci trajektorie end-effectoru s cílem dosažení nejpříznivějších výsledků z pohledu zachycení předmětu. Jak již bylo zmíněno v předchozí kapitole, end-effector by na zvolené křivce měl zrychlovat z počátečního bodu, provést zachycení tělesa a zpomalit do koncového bodu. Požadavky pro zachytovou křivku jsou definovány takto:

1. Nachází se v pracovním prostoru manipulátoru.
2. Dostatečně velká část křivky je shodná s trajektorií letícího předmětu.

Dostatečně velkou částí křivky je myšlena úsečka náležící trajektorii letícího tělesa, na níž probíhá zachytávání. Délka této úsečky je závislá na době potřebné pro vykonání zachycení mechanismem end-effectoru. Při návrhu zachytové křivky uvažujeme dva možné přístupy.

První možností je rozdělení křivky na 3 úseky, z nichž by přímce letícího předmětu náležela pouze druhá část, na které je prováděno zachytávání. V prvním a posledním úseku by se end-effector mohl pohybovat po křivkách, jejichž tvar by byl zvolen s ohledem na optimální možnosti zrychlení, respektive zpomalení manipulátoru a bylo by tak možné dosáhnout co nejvyšších možných zachytitelných rychlostí. Při volbě křivky by bylo vhodné dbát na geometrickou spojitost jejích částí, mohly by být použity např. křivky spline, či kruhové oblouky. Příklad takové křivky je zobrazen na Obr. 4.2 modrou barvou, zelená přímka představuje trajektorii letícího tělesa. Tato varianta je optimální z hlediska dosahovaných rychlostí, jedná se však o složitý výpočet, který by byl náročný při plánované optimalizaci trajektorie end-effectoru.

Druhou možností je volba pohybu end-effectoru po úsečce, která celá náleží přímce letícího tělesa. Bylo by na ní tedy provedeno zrychlování end-effectoru, zachycení a následně i zpomalení do koncového bodu. Na Obr. 4.2 je znázorněna červenou barvou. Jedná se o výpočetně příznivou variantu, kterou lze parametricky snadno popsat. S ohledem na plánovanou optimalizaci trajektorie byla pro řešení zvolena právě tato varianta a to především z důvodu časově méně náročných výpočtů. V této oblasti by bylo možné na práci navázat a výpočet praktikovat na složitější křivky, které by mohly dosahovat lepších výsledků z pohledu dosažitelné rychlosti.



Obr. 4.2: Volba záchytové křivky

Volíme tedy úsečku s počátečním bodem $A[a_x, a_y]$ a koncovým bodem $B[b_x, b_y]$. Úsečku lze parametricky popsat radiusvektorem (3.1):

$$\mathbf{R}(p) = [x(p), y(p)]^T \quad p \in \langle 0, 1 \rangle$$

Kde:

$$\begin{aligned} x(p) &= a_x + p \cdot u_x & u_x &= b_x - a_x & p &\in \langle 0, 1 \rangle & (4.2) \\ y(p) &= a_y + p \cdot u_y & u_y &= b_y - a_y \end{aligned}$$

4.2 Inverzní kinematika

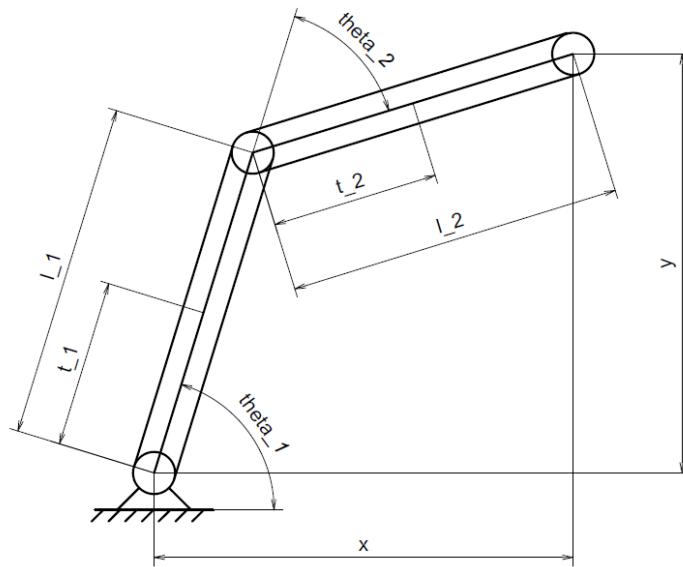
Prvním krokem v řešení globální dynamiky manipulátoru je podle [3] již zmíněné vyjádření inverzní kinematiky RR manipulátoru, tedy vztahu (3.4):

$$\boldsymbol{\theta}(p) = \mathbf{K}^{-1}(\mathbf{R}(p))$$

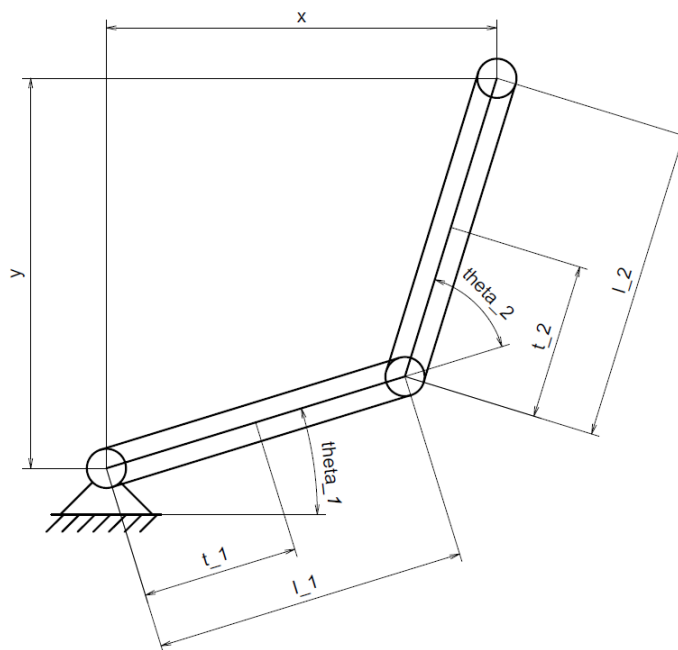
Kde:

$$\boldsymbol{\theta}(p) = [\theta_1(p), \theta_2(p)]^T$$

Pro rovinný RR manipulátor existují v každém zadaném bodě pracovního prostoru dvě realizovatelné konfigurace. Výjimkou jsou pouze okraje pracovního prostoru, na kterých se tyto dvě konfigurace rovnají. Zobrazeny jsou na Obr. 4.3 a 4.4. Rozdíl spočívá ve volbě orientace úhlu θ_2 .



Obr. 4.3: I. konfigurace RR robotu



Obr. 4.4: II. konfigurace RR robotu

Řešení vztahu inverzní kinematiky (3.4) je pro RR manipulátor známým výpočtem a jeho výsledky je možné podle [14] zapsat jako:

$$\theta_2 = \mp \cos^{-1} \left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \right) \quad (4.3)$$

$$\theta_1 = \tan^{-1} \left(\frac{y}{x} \right) \pm \tan^{-1} \left(\frac{l_2 \cdot \sin(\theta_2)}{l_1 + l_2 \cdot \cos(\theta_2)} \right) \quad (4.4)$$

V konfiguraci I se v rovnici pro θ_2 vyskytuje znaménko $-$ a v rovnici pro θ_1 $+$. V případě konfigurace II je tomu naopak. Pro řešení úlohy volíme konfiguraci I. Z důvodu možnosti popsání úhlů v celém intervalu $\langle 0, 2\pi \rangle$ je výhodné úhly popsat pomocí funkce $atan2$ [3]:

$$h = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1l_2} \quad (4.5)$$

$$\theta_2 = atan2\left(-\sqrt{1-h^2}, h\right) \quad (4.6)$$

$$\begin{aligned} \theta_1 = atan2((l_1 + l_2 \cdot \cos(\theta_2)) \cdot y - l_2 \cdot \sin(\theta_2) \cdot x, \\ (l_1 + l_2 \cdot \cos(\theta_2)) \cdot x + l_2 \cdot \sin(\theta_2) \cdot y) \end{aligned} \quad (4.7)$$

Požadované závislosti $\theta_1(p)$, $\theta_2(p)$ jsou následně obdrženy dosazením $x(p)$, $y(p)$ do rovnic (4.5) a (4.7).

4.3 Výpočet parciálních derivací

Pro výpočet časové parametrizace křivky je potřeba vyjádřit prvky matic parciálních derivací θ' a θ'' . Matice jsou zavedeny takto:

$$\theta' = [\theta'_1, \theta'_2]^T = \left[\frac{\partial \theta_1}{\partial p}, \frac{\partial \theta_2}{\partial p} \right]^T \quad (4.8)$$

$$\theta'' = [\theta''_1, \theta''_2]^T = \left[\frac{\partial^2 \theta_1}{\partial p^2}, \frac{\partial^2 \theta_2}{\partial p^2} \right]^T \quad (4.9)$$

Pro výpočet jejich prvků zavádíme v inverzní kinematice náhradní označení argumentů funkce $atan2$:

$$\begin{aligned} \theta_2 &= atan2(y_{arg2}, x_{arg2}) \\ \theta_1 &= atan2(y_{arg1}, x_{arg1}) \\ x_{arg2} &= h \\ y_{arg2} &= -\sqrt{1-h^2} \\ x_{arg1} &= (l_1 + l_2 \cdot \cos(\theta_2))x + l_2 \cdot \sin(\theta_2)y \\ y_{arg1} &= (l_1 + l_2 \cdot \cos(\theta_2))y - l_2 \cdot \sin(\theta_2)x \end{aligned}$$

Nyní vyjádříme parciální derivace úhlů v pohonech podle parametru p pomocí řetězových derivací:

$$\frac{\partial \theta_i}{\partial p} = -\frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \frac{\partial x_{arg,i}}{\partial p} + \frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \frac{\partial y_{arg,i}}{\partial p} \quad (4.10)$$

$$\frac{\partial x_{arg2}}{\partial p} = \frac{\partial x_{arg2}}{\partial h} \frac{\partial h}{\partial p}$$

$$\frac{\partial y_{arg2}}{\partial p} = \frac{\partial y_{arg2}}{\partial h} \frac{\partial h}{\partial p}$$

$$\frac{\partial h}{\partial p} = \frac{\partial h}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial h}{\partial y} \frac{\partial y}{\partial p}$$

$$\frac{\partial x_{arg1}}{\partial p} = \frac{\partial x_{arg1}}{\partial \theta_2} \frac{\partial \theta_2}{\partial p} + \frac{\partial x_{arg1}}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial x_{arg1}}{\partial y} \frac{\partial y}{\partial p}$$

$$\frac{\partial y_{arg1}}{\partial p} = \frac{\partial y_{arg1}}{\partial \theta_2} \frac{\partial \theta_2}{\partial p} + \frac{\partial y_{arg1}}{\partial x} \frac{\partial x}{\partial p} + \frac{\partial y_{arg1}}{\partial y} \frac{\partial y}{\partial p}$$

Veškeré dílčí členy rovnic jsou uvedeny až po výpočtech druhých parciálních derivací. Druhé parciální derivace jsou vyjádřeny následovně:

$$\begin{aligned} \frac{\partial^2 \theta_i}{\partial p^2} &= \frac{\partial}{\partial p} \left(-\frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) \frac{\partial x_{arg,i}}{\partial p} - \frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \frac{\partial^2 x_{arg,i}}{\partial p^2} \\ &+ \frac{\partial}{\partial p} \left(\frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) \frac{\partial y_{arg,i}}{\partial p} + \frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \frac{\partial^2 y_{arg,i}}{\partial p^2} \end{aligned} \quad (4.11)$$

$$\begin{aligned} \frac{\partial}{\partial p} \left(-\frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) &= \frac{\partial}{\partial x_{arg,i}} \left(-\frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) \frac{\partial x_{arg,i}}{\partial p} \\ &+ \frac{\partial}{\partial y_{arg,i}} \left(-\frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) \frac{\partial y_{arg,i}}{\partial p} \end{aligned}$$

$$\begin{aligned} \frac{\partial}{\partial p} \left(\frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) &= \frac{\partial}{\partial x_{arg,i}} \left(\frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) \frac{\partial x_{arg,i}}{\partial p} \\ &+ \frac{\partial}{\partial y_{arg,i}} \left(\frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) \frac{\partial y_{arg,i}}{\partial p} \end{aligned}$$

$$\frac{\partial}{\partial x_{arg,i}} \left(-\frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) = \frac{2x_{arg,i}y_{arg,i}}{(x_{arg,i}^2 + y_{arg,i}^2)^2}$$

$$\frac{\partial}{\partial y_{arg,i}} \left(-\frac{y_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) = \frac{\partial}{\partial x_{arg,i}} \left(\frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) = \frac{y_{arg,i}^2 - x_{arg,i}^2}{(x_{arg,i}^2 + y_{arg,i}^2)^2}$$

$$\frac{\partial}{\partial y_{arg,i}} \left(\frac{x_{arg,i}}{x_{arg,i}^2 + y_{arg,i}^2} \right) = -\frac{2x_{arg,i}y_{arg,i}}{(x_{arg,i}^2 + y_{arg,i}^2)^2}$$

$$\frac{\partial^2 x_{arg2}}{\partial p^2} = \frac{\partial^2 x_{arg2}}{\partial h^2} \left(\frac{\partial h}{\partial p} \right)^2 + \frac{\partial x_{arg2}}{\partial h} \frac{\partial^2 h}{\partial p^2}$$

$$\frac{\partial^2 y_{arg2}}{\partial p^2} = \frac{\partial^2 y_{arg2}}{\partial h^2} \left(\frac{\partial h}{\partial p} \right)^2 + \frac{\partial y_{arg2}}{\partial h} \frac{\partial^2 h}{\partial p^2}$$

$$\frac{\partial^2 h}{\partial p^2} = \frac{\partial^2 h}{\partial x^2} \left(\frac{\partial x}{\partial p} \right)^2 + \frac{\partial h}{\partial x} \frac{\partial^2 x}{\partial p^2} + \frac{\partial^2 h}{\partial y^2} \left(\frac{\partial y}{\partial p} \right)^2 + \frac{\partial h}{\partial y} \frac{\partial^2 y}{\partial p^2}$$

$$\begin{aligned} \frac{\partial^2 x_{arg1}}{\partial p^2} &= \left(\frac{\partial^2 x_{arg1}}{(\partial \theta_2)^2} \frac{\partial \theta_2}{\partial p} + \frac{\partial^2 x_{arg1}}{\partial \theta_2 \partial x} \frac{\partial x}{\partial p} + \frac{\partial^2 x_{arg1}}{\partial \theta_2 \partial y} \frac{\partial y}{\partial p} \right) \frac{\partial \theta_2}{\partial p} + \frac{\partial x_{arg1}}{\partial \theta_2} \frac{\partial^2 \theta_2}{\partial p^2} \\ &+ \left(\frac{\partial^2 x_{arg1}}{\partial x^2} \frac{\partial x}{\partial p} + \frac{\partial^2 x_{arg1}}{\partial \theta_2 \partial x} \frac{\partial \theta_2}{\partial p} \right) \frac{\partial x}{\partial p} + \frac{\partial x_{arg1}}{\partial x} \frac{\partial^2 x}{\partial p^2} \\ &+ \left(\frac{\partial^2 x_{arg1}}{\partial y^2} \frac{\partial y}{\partial p} + \frac{\partial^2 x_{arg1}}{\partial \theta_2 \partial y} \frac{\partial \theta_2}{\partial p} \right) \frac{\partial y}{\partial p} + \frac{\partial x_{arg1}}{\partial y} \frac{\partial^2 y}{\partial p^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 y_{arg1}}{\partial p^2} &= \left(\frac{\partial^2 y_{arg1}}{(\partial \theta_2)^2} \frac{\partial \theta_2}{\partial p} + \frac{\partial^2 y_{arg1}}{\partial \theta_2 \partial x} \frac{\partial x}{\partial p} + \frac{\partial^2 y_{arg1}}{\partial \theta_2 \partial y} \frac{\partial y}{\partial p} \right) \frac{\partial \theta_2}{\partial p} + \frac{\partial y_{arg1}}{\partial \theta_2} \frac{\partial^2 \theta_2}{\partial p^2} \\ &+ \left(\frac{\partial^2 y_{arg1}}{\partial x^2} \frac{\partial x}{\partial p} + \frac{\partial^2 y_{arg1}}{\partial \theta_2 \partial x} \frac{\partial \theta_2}{\partial p} \right) \frac{\partial x}{\partial p} + \frac{\partial y_{arg1}}{\partial x} \frac{\partial^2 x}{\partial p^2} \\ &+ \left(\frac{\partial^2 y_{arg1}}{\partial y^2} \frac{\partial y}{\partial p} + \frac{\partial^2 y_{arg1}}{\partial \theta_2 \partial y} \frac{\partial \theta_2}{\partial p} \right) \frac{\partial y}{\partial p} + \frac{\partial y_{arg1}}{\partial y} \frac{\partial^2 y}{\partial p^2} \end{aligned}$$

Dílčí členy prvních a druhých parciálních derivací v rovnicích (4.10) a (4.11) mají tyto hodnoty:

$$\begin{array}{ll} \frac{\partial x_{arg2}}{\partial h} = 1 & \frac{\partial y_{arg2}}{\partial h} = \frac{h}{\sqrt{1-h^2}} \\ \frac{\partial^2 x_{arg2}}{\partial h^2} = 0 & \frac{\partial^2 y_{arg2}}{\partial h^2} = \frac{1}{(1-h^2)^{\frac{3}{2}}} \\ \frac{\partial h}{\partial x} = \frac{x}{l_1 l_2} & \frac{\partial h}{\partial y} = \frac{y}{l_1 l_2} \\ \frac{\partial^2 h}{\partial x^2} = \frac{1}{l_1 l_2} & \frac{\partial^2 h}{\partial y^2} = \frac{1}{l_1 l_2} \\ \frac{\partial x}{\partial p} = u_x & \frac{\partial y}{\partial p} = u_y \\ \frac{\partial^2 x}{\partial p^2} = 0 & \frac{\partial^2 y}{\partial p^2} = 0 \\ \frac{\partial x_{arg1}}{\partial x} = \cos(\theta_2) l_2 + l_1 & \frac{\partial x_{arg1}}{\partial y} = l_2 \cdot \sin(\theta_2) \\ \frac{\partial^2 x_{arg1}}{\partial x^2} = 0 & \frac{\partial^2 x_{arg1}}{\partial y^2} = 0 \\ \frac{\partial y_{arg1}}{\partial x} = -l_2 \cdot \sin(\theta_2) & \frac{\partial y_{arg1}}{\partial y} = l_1 + l_2 \cdot \cos(\theta_2) \\ \frac{\partial^2 y_{arg1}}{\partial x^2} = 0 & \frac{\partial^2 y_{arg1}}{\partial y^2} = 0 \end{array}$$

$$\frac{\partial x_{arg1}}{\partial \theta_2} = -l_2 (x \cdot \sin(\theta_2) - y \cdot \cos(\theta_2))$$

$$\frac{\partial y_{arg1}}{\partial \theta_2} = -l_2 (y \cdot \sin(\theta_2) + x \cdot \cos(\theta_2))$$

$$\frac{\partial^2 x_{arg1}}{(\partial \theta_2)^2} = -l_2 (x \cdot \cos(\theta_2) + y \cdot \sin(\theta_2))$$

$$\frac{\partial^2 y_{arg1}}{(\partial \theta_2)^2} = -l_2 (y \cdot \cos(\theta_2) - x \cdot \sin(\theta_2))$$

$$\frac{\partial^2 x_{arg1}}{\partial \theta_2 \partial x} = -l_2 \cdot \sin(\theta_2)$$

$$\frac{\partial^2 y_{arg1}}{\partial \theta_2 \partial x} = -l_2 \cdot \cos(\theta_2)$$

$$\frac{\partial^2 x_{arg1}}{\partial \theta_2 \partial y} = l_2 \cdot \cos(\theta_2)$$

$$\frac{\partial^2 y_{arg1}}{\partial \theta_2 \partial y} = -l_2 \cdot \sin(\theta_2)$$

4.4 Rovnice dynamiky

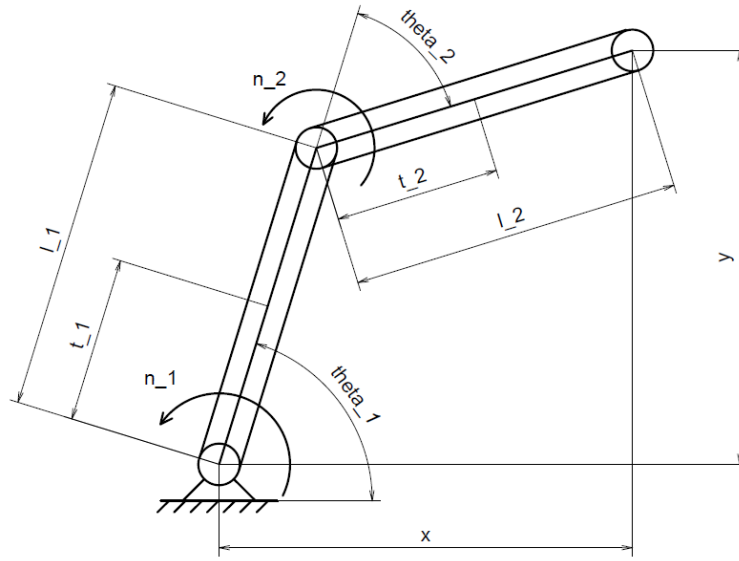
Dalším krokem je vyjádření rovnice dynamiky (3.5) [3]:

$$\mathbf{n} = \mathbf{I}(\boldsymbol{\theta}) \ddot{\boldsymbol{\theta}} + \dot{\boldsymbol{\theta}}^T \mathbf{C}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} + \mathbf{V}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}} + \mathbf{g}(\boldsymbol{\theta})$$

Pro případ manipulátoru ve vesmíru uvažujeme nulovou složku $\mathbf{g}(\boldsymbol{\theta})$. Dále jsou zanedbány třecí síly, tedy i složka $\mathbf{V}(\boldsymbol{\theta})$ je rovna nule. Člen $\dot{\boldsymbol{\theta}}^T \mathbf{C}(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}$ lze pro zjednodušení nahradit tvarem $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$, tím dostáváme hledanou podobu rovnic dynamiky:

$$\mathbf{n} = \mathbf{I}(\boldsymbol{\theta}) \ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \dot{\boldsymbol{\theta}} \tag{4.12}$$

Pro odvození použijeme metodu Langrangeových rovnic II. druhu. Rozměrové parametry manipulátoru jsou zobrazeny na Obr. 4.5. Parametry t_1, t_2 určují polohu těžišť jednotlivých těles. Dále je počítáno s hmotnostmi m_1, m_2 a s momenty setrvačnosti těles vzhledem k těžišti I_{1T1}, I_{2T2} . Do koncového bodu manipulátoru je umístěn hmotný bod o hmotnosti M , který představuje zjednodušení zachyceného tělesa. Hodnotu hmotnosti M lze měnit pro konkrétní polohu na trajektorii v závislosti na tom, jestli již bylo provedeno zachycení.



Obr. 4.5: Rozměry potřebné pro rovnice dynamiky

Langrangeovy rovnice II. druhu jsou dány vztahem [4],[15]:

$$\frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{q}_j} \right) - \frac{\partial E_k}{\partial q_j} = Q_j \quad (4.13)$$

Kde E_k představuje kinetickou energii soustavy, q_j natočení v pohonech θ_1 a θ_2 a Q_j kroutící momenty v pohonech n_1 a n_2 . Kinetická energie je vyjádřena jako:

$$E_k = \frac{1}{2} m_1 v_{1T1}^2 + \frac{1}{2} I_{1T1} \dot{\theta}_1^2 + \frac{1}{2} m_2 v_{1T2}^2 + \frac{1}{2} I_{2T2} (\dot{\theta}_1 + \dot{\theta}_2)^2 + \frac{1}{2} M v^2 \quad (4.14)$$

Kde:

$$\begin{aligned} v_{1T1}^2 &= \dot{x}_{1T1}^2 + \dot{y}_{1T1}^2 \\ v_{1T2}^2 &= \dot{x}_{1T2}^2 + \dot{y}_{1T2}^2 \\ v^2 &= \dot{x}^2 + \dot{y}^2 \end{aligned}$$

K vyjádření kinetické energie je třeba popsat kvadráty rychlostí jednotlivých členů manipulátoru. Polohy end-effektoru a těžišť těles jsou popsány takto:

$$\begin{aligned} x_{1T1} &= t_1 \cdot \cos(\theta_1) \\ y_{1T1} &= t_1 \cdot \sin(\theta_1) \\ x_{1T2} &= l_1 \cdot \cos(\theta_1) + t_2 \cdot \cos(\theta_1 + \theta_2) \\ y_{1T2} &= l_1 \cdot \sin(\theta_1) + t_2 \cdot \sin(\theta_1 + \theta_2) \\ x &= l_1 \cdot \cos(\theta_1) + l_2 \cdot \cos(\theta_1 + \theta_2) \\ y &= l_1 \cdot \sin(\theta_1) + l_2 \cdot \sin(\theta_1 + \theta_2) \end{aligned}$$

Z poloh jsou derivováním vyjádřeny rychlosti ve směrech os:

$$\dot{x}_{1T1} = -t_1 \cdot \sin(\theta_1) \cdot \dot{\theta}_1$$

$$\dot{y}_{1T1} = t_1 \cdot \cos(\theta_1) \cdot \dot{\theta}_1$$

$$\dot{x}_{1T2} = -l_1 \cdot \sin(\theta_1) \cdot \dot{\theta}_1 - t_2 \cdot \sin(\theta_1 + \theta_2) \cdot (\dot{\theta}_1 + \dot{\theta}_2)$$

$$\dot{y}_{1T2} = l_1 \cdot \cos(\theta_1) \cdot \dot{\theta}_1 + t_2 \cdot \cos(\theta_1 + \theta_2) \cdot (\dot{\theta}_1 + \dot{\theta}_2)$$

$$\dot{x} = -l_1 \cdot \sin(\theta_1) \cdot \dot{\theta}_1 - l_2 \cdot \sin(\theta_1 + \theta_2) \cdot (\dot{\theta}_1 + \dot{\theta}_2)$$

$$\dot{y} = l_1 \cdot \cos(\theta_1) \cdot \dot{\theta}_1 + l_2 \cdot \cos(\theta_1 + \theta_2) \cdot (\dot{\theta}_1 + \dot{\theta}_2)$$

Z rychlostí v osových směrech jsou vyjádřeny kvadráty rychlostí:

$$v_{1T1}^2 = \dot{x}_{1T1}^2 + \dot{y}_{1T1}^2 = t_1^2 \cdot \dot{\theta}_1^2 \quad (4.15)$$

$$v_{1T2}^2 = \dot{x}_{1T2}^2 + \dot{y}_{1T2}^2 = l_1^2 \cdot \dot{\theta}_1^2 + t_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 t_2 \cdot \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (4.16)$$

$$v^2 = \dot{x}^2 + \dot{y}^2 = l_1^2 \cdot \dot{\theta}_1^2 + l_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2l_1 l_2 \cdot \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (4.17)$$

Po dosazení (4.15), (4.16) a (4.17) do (4.14) je kinetická energie rovna:

$$E_k = \frac{1}{2} (m_1 t_1^2 + I_{1T1} + m_2 l_1^2 + M l_1^2) \dot{\theta}_1^2 + \frac{1}{2} (m_2 t_2^2 + I_{2T2} + M l_2^2) (\dot{\theta}_1 + \dot{\theta}_2)^2 + (m_2 l_1 t_2 + M l_1 l_2) \cdot \cos(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (4.18)$$

Vyjádříme derivace potřebné pro sestavení Lagrangeových rovnic:

$$\frac{\partial E_k}{\partial \dot{\theta}_1} = (m_1 t_1^2 + I_{1T1} + m_2 l_1^2 + M l_1^2) \dot{\theta}_1 + (m_2 t_2^2 + I_{2T2} + M l_2^2) (\dot{\theta}_1 + \dot{\theta}_2) + (m_2 l_1 t_2 + M l_1 l_2) \cos(\theta_2) (2\dot{\theta}_1 + \dot{\theta}_2)$$

$$\frac{\partial E_k}{\partial \dot{\theta}_2} = (m_2 t_2^2 + I_{2T2} + M l_2^2) (\dot{\theta}_1 + \dot{\theta}_2) + (m_2 l_1 t_2 + M l_1 l_2) \cos(\theta_2) \dot{\theta}_1$$

$$\frac{\partial E_k}{\partial \theta_1} = 0 \quad (4.19)$$

$$\frac{\partial E_k}{\partial \theta_2} = -(m_2 l_1 t_2 + M l_1 l_2) \sin(\theta_2) \dot{\theta}_1 (\dot{\theta}_1 + \dot{\theta}_2) \quad (4.20)$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{\theta}_1} \right) &= (m_1 t_1^2 + I_{1T1} + m_2 l_1^2 + M l_1^2) \ddot{\theta}_1 + (m_2 t_2^2 + I_{2T2} + M l_2^2) (\ddot{\theta}_1 + \ddot{\theta}_2) \\ &\quad + (m_2 l_1 t_2 + M l_1 l_2) \left(\cos(\theta_2) (2\ddot{\theta}_1 + \ddot{\theta}_2) - \sin(\theta_2) \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \right) \end{aligned} \quad (4.21)$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial E_k}{\partial \dot{\theta}_1} \right) &= (m_2 t_2^2 + I_{2T2} + Ml_2^2) (\ddot{\theta}_1 + \ddot{\theta}_2) \\ &+ (m_2 l_1 t_2 + Ml_1 l_2) \left(\cos(\theta_2) \ddot{\theta}_1 - \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 \right) \end{aligned} \quad (4.22)$$

Po dosazení členů (4.19), (4.20), (4.21) a (4.22) do Lagrangeových rovnic (4.13) dostáváme rovnice pro momenty v pohonech manipulátoru:

$$\begin{aligned} (m_1 t_1^2 + I_{1T1} + m_2 l_1^2 + Ml_1^2 + m_2 t_2^2 + Ml_2^2 + I_{2T2} + 2(m_2 l_1 t_2 + Ml_1 l_2) \cos(\theta_2)) \ddot{\theta}_1 \\ + (m_2 t_2^2 + Ml_2^2 + I_{2T2} + (m_2 l_1 t_2 + Ml_1 l_2) \cos(\theta_2)) \ddot{\theta}_2 \\ - (m_2 l_1 t_2 + Ml_1 l_2) \sin(\theta_2) \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) = n_1 \end{aligned} \quad (4.23)$$

$$\begin{aligned} (m_2 t_2^2 + Ml_2^2 + I_{2T2} + (m_2 l_1 t_2 + Ml_1 l_2) \cos(\theta_2)) \ddot{\theta}_1 \\ + (m_2 t_2^2 + Ml_2^2 + I_{2T2}) \ddot{\theta}_2 + (m_2 l_1 t_2 + Ml_1 l_2) \sin(\theta_2) \dot{\theta}_1^2 = n_2 \end{aligned} \quad (4.24)$$

Pro zachování značení z rovnice (4.12) vyjádříme matice $\mathbf{I}(\boldsymbol{\theta})$ a $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$:

$$\mathbf{I}(\boldsymbol{\theta}) = \begin{pmatrix} \mathbf{I}(1,1) & \mathbf{I}(1,2) \\ \mathbf{I}(2,1) & \mathbf{I}(2,2) \end{pmatrix} \quad (4.25)$$

$$\begin{aligned} \mathbf{I}(1,1) &= m_1 t_1^2 + I_{1T1} + m_2 l_1^2 + Ml_1^2 \\ &+ m_2 t_2^2 + Ml_2^2 + I_{2T2} + 2(m_2 l_1 t_2 + Ml_1 l_2) \cos(\theta_2) \end{aligned}$$

$$\mathbf{I}(1,2) = \mathbf{I}(2,1) = m_2 t_2^2 + Ml_2^2 + I_{2T2} + (m_2 l_1 t_2 + Ml_1 l_2) \cos(\theta_2)$$

$$\mathbf{I}(2,2) = m_2 t_2^2 + Ml_2^2 + I_{2T2}$$

$$\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{pmatrix} - (m_2 l_1 t_2 + Ml_1 l_2) \sin(\theta_2) \dot{\theta}_2 (2\dot{\theta}_1 + \dot{\theta}_2) \\ (m_2 l_1 t_2 + Ml_1 l_2) \sin(\theta_2) \dot{\theta}_1^2 \end{pmatrix} \quad (4.26)$$

Výpočty rychlostního profilu jsou provedeny v závislosti na parametru p , z toho důvodu do matice (4.26) dosadíme parciální derivace a zavádíme matici:

$$\mathbf{C}'(\boldsymbol{\theta}, \boldsymbol{\theta}') = \begin{pmatrix} - (m_2 l_1 t_2 + Ml_1 l_2) \sin(\theta_2) \theta_2' (2\theta_1' + \theta_2') \\ (m_2 l_1 t_2 + Ml_1 l_2) \sin(\theta_2) (\theta_1')^2 \end{pmatrix} \quad (4.27)$$

4.5 Limity pohonů

Základním předpokladem pro nalezení vhodného průběhu $p(t)$ na zvolené trajektorii manipulátoru je znalost omezení hodnot \dot{p} a \ddot{p} . Pro RR manipulátor uvažujeme tato omezení:

$$n_i^- - F_i \cdot \dot{\theta}_i \leq n_i \leq n_i^+ - F_i \cdot \dot{\theta}_i \quad (4.28)$$

$$\dot{\theta}_i^- \leq \dot{\theta}_i \leq \dot{\theta}_i^+ \quad (4.29)$$

$$\ddot{\theta}_i^- \leq \ddot{\theta}_i \leq \ddot{\theta}_i^+ \quad (4.30)$$

Volíme symetrické hodnoty:

$$\begin{aligned} -n_i^- &= n_i^+ \\ -\dot{\theta}_i^- &= \dot{\theta}_i^+ \\ -\ddot{\theta}_i^- &= \ddot{\theta}_i^+ \end{aligned} \quad (4.31)$$

Limity kroutících momentů uvažujeme závislé na úhlové rychlosti. Pro konkrétní manipulátor by omezení závisela na daných typech použitých pohonů, jako příklad lze uvést závislosti momentů na úhlových rychlostech pro stejnosměrný motor s permanentními magnety. Jsou uvažovány rovnice [16]:

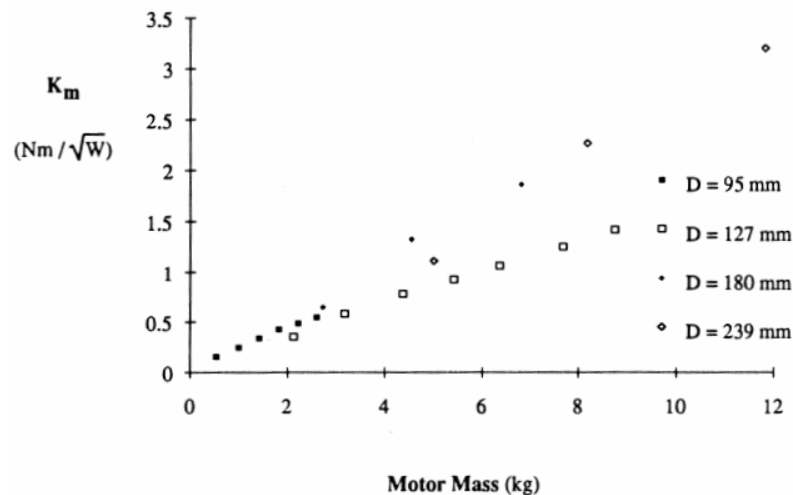
$$n = K_T \cdot i \quad (4.32)$$

$$V = K_T \cdot \dot{\theta} + R \cdot i \quad (4.33)$$

Kde K_T je momentová konstanta, i je proud, V je dodávané napětí, $\dot{\theta}$ je úhlová rychlost rotoru, $E = K_T \cdot \dot{\theta}$ je zpětná elektromagnetická síla a R je odpor vinutí. Pro fixní napětí V jsou závislosti momentu na úhlové rychlosti vyjádřeny z rovnic takto:

$$n = \frac{K_T (V - K_T \cdot \omega)}{R} = \frac{K_T V}{R} - K_m^2 \cdot \dot{\theta} \quad (4.34)$$

Kde výraz $K_T V / R$ je momentem motoru při nulové úhlové rychlosti a $K_m = K_T / R$ je konstanta motoru. Člen K_m^2 tedy představuje F z uvažovaných omezení kroutících momentů pohonů (4.28).



Obr. 4.6: Hodnoty K_m bezkartáčových stejnosměrných motorů [16]

V práci [16] je provedena rešerše aktuátorů užívaných v robotice a uvedeny typické hodnoty konstanty K_m . Konstanta závisí na parametrech jednotlivých motorů. Pomocí vhodného převodu v aktuátorech lze také zvýšit hodnotu momentu za nulové

rychlosti a snížit hodnotu konstanty K_m , tím tedy i snížit klesání kroutícího momentu s rostoucí úhlovou rychlostí. Závislosti momentů na úhlové rychlosti dále závisí na teplotě, ta by ve vesmírné aplikaci mohla hrát významnou roli. V této práci se však omezíme na použití typické hodnoty K_m z článku [16]. Na Obr. 4.6 jsou zobrazeny hodnoty pro některé bezkartáčové stejnosměrné motory. Je patrné, že hodnota K_m roste s hmotností motoru a také s vnějším průměrem statoru D .

Pro výpočet je zvolena hodnota $K_m = 2 Nm/\sqrt{W}$, hodnotu F_i tedy uvažujeme rovnou $4 N^2 m^2/W$.

Z momentů v pohonech je možné vyjádřit realizovatelný interval \ddot{p} podle rovnic (3.18), (3.19) a (3.20). Omezující hodnoty označíme $L_{i,dyn}(p, \dot{p})$ a $U_{i,dyn}(p, \dot{p})$. Pro zvolená omezení manipulátoru a vypočtené matice dynamiky (4.25) a (4.27) lze sestavit vztahy:

$$L_{i,dyn}(p, \dot{p}) \leq \ddot{p} \leq U_{i,dyn}(p, \dot{p}) \quad (4.35)$$

Kde:

$$L_{i,dyn}(p, \dot{p}) = \frac{n_i^- \cdot \delta_{i,dyn}(p) + n_i^+ (1 - \delta_{i,dyn}(p)) - b_i(p) \dot{p}^2 - c_i(p) \dot{p}}{a_i(p)} \quad (4.36)$$

$$U_{i,dyn}(p, \dot{p}) = \frac{n_i^+ \cdot \delta_{i,dyn}(p) + n_i^- (1 - \delta_{i,dyn}(p)) - b_i(p) \dot{p}^2 - c_i(p) \dot{p}}{a_i(p)} \quad (4.37)$$

$$if \ a_i(p) > 0 : \ \delta_{i,dyn}(p) = 1$$

$$if \ a_i(p) < 0 : \ \delta_{i,dyn}(p) = 0$$

Kde:

$$a_i(p) = \mathbf{I}_i(\boldsymbol{\theta}) \boldsymbol{\theta}'(p) \quad (4.38)$$

$$b_i(p) = \mathbf{I}_i(\boldsymbol{\theta}) \boldsymbol{\theta}''(p) + C'_i(\boldsymbol{\theta}, \boldsymbol{\theta}') \quad (4.39)$$

$$c_i(p) = F_i \cdot \theta'_i(p) \quad (4.40)$$

Výrazy $\mathbf{I}_i(\boldsymbol{\theta})$ a $C'_i(\boldsymbol{\theta}, \boldsymbol{\theta}')$ představují i -té řádky matic. Následně lze zapsat sjednocené dynamické limity:

$$L_{dyn}(p, \dot{p}) \leq \ddot{p} \leq U_{dyn}(p, \dot{p}) \quad (4.41)$$

Kde:

$$L_{dyn}(p, \dot{p}) = \max_i \{L_{i,dyn}(p, \dot{p})\} \quad (4.42)$$

$$U_{dyn}(p, \dot{p}) = \min_i \{U_{i,dyn}(p, \dot{p})\} \quad (4.43)$$

Pokud v některém bodě křivky platí $a_i(p) = 0$, je nutné pro danou hodnotu p zkontrolovat podmínku:

$$n_i^- \leq b_i(p) \cdot \dot{p}^2 + c_i(p) \cdot \dot{p} \leq n_i^+ \quad (4.44)$$

Nyní je třeba ke vztahům plynoucím z omezení kroutících momentů v pohonech přidat podmínky plynoucí z omezení úhlových rychlostí a zrychlení. Nejprve se zaměříme na úhlové rychlosti. Dosazením vztahu (3.7) do podmínky (4.29) dostáváme vztah:

$$\dot{\theta}_i^- \leq \theta'_i(p) \cdot \dot{p} \leq \dot{\theta}_i^+ \quad (4.45)$$

Z důvodu symetricky zvolených omezení (4.31) a faktu, že je parametrizace hledána pro jednosměrný pohyb, lze pro každý pohon pro hodnotu \dot{p} vyjádřit realizovatelný interval:

$$0 \leq \dot{p} \leq U_{i,vel}(p) \quad (4.46)$$

Kde:

$$U_{i,vel}(p) = \frac{\dot{\theta}_i^+}{|\theta'_i(p)|} \quad (4.47)$$

Intervaly jednotlivých pohonů opět sjednotíme:

$$0 \leq \dot{p} \leq U_{vel}(p) \quad (4.48)$$

Kde:

$$U_{vel}(p) = \min_i \{U_{i,vel}(p)\} \quad (4.49)$$

V případě, že v bodě křivky platí $\theta_i(p)$, realizovatelný interval je teoreticky pro daný pohon v daném bodě při zanedbání všech ostatních omezení $(-\infty, \infty)$.

Pro omezení úhlových zrychlení lze dosazením vztahů (3.7) a (3.8) do podmínky (4.30) získat vztah:

$$\ddot{\theta}_i^- \leq \theta'_i(p) \cdot \ddot{p} + \theta''_i(p) \cdot \dot{p}^2 \leq \ddot{\theta}_i^+ \quad (4.50)$$

Z této nerovnice je možné podobně jako v případě vztahů (4.35), (4.36) a (4.37) vyjádřit pro každý pohon realizovatelné intervaly hodnot \ddot{p} :

$$L_{i,akc}(p, \dot{p}) \leq \ddot{p} \leq U_{i,akc}(p, \dot{p}) \quad (4.51)$$

Kde:

$$L_{i,akc}(p, \dot{p}) = \frac{\ddot{\theta}_i^- \cdot \delta_{i,akc}(p) + \ddot{\theta}_i^+ (1 - \delta_{i,akc}(p)) - \theta''_i(p) \cdot \dot{p}^2}{\theta'_i(p)} \quad (4.52)$$

$$U_{i,akc}(p, \dot{p}) = \frac{\ddot{\theta}_i^+ \cdot \delta_{i,akc}(p) + \ddot{\theta}_i^- (1 - \delta_{i,akc}(p)) - \theta''_i(p) \cdot \dot{p}^2}{\theta'_i(p)} \quad (4.53)$$

$$if \theta'_i(p) > 0 : \delta_{i,akc}(p) = 1$$

$$if \theta'_i(p) < 0 : \delta_{i,akc}(p) = 0$$

Tyto intervaly lze opět sjednotit pro celý manipulátor:

$$L_{akc}(p, \dot{p}) \leq \ddot{p} \leq U_{akc}(p, \dot{p}) \quad (4.54)$$

Kde:

$$L_{akc}(p, \dot{p}) = \max_i \{L_{i,akc}\} \quad (4.55)$$

$$U_{akc}(p, \dot{p}) = \min_i \{U_{i,akc}\} \quad (4.56)$$

V případě, kdy v bodě křivky platí $\theta'_i(p) = 0$, řešíme pro danou hodnotu p podmínku:

$$\ddot{\theta}_i^- \leq \theta''_i(p) \cdot \dot{p}^2 \leq \ddot{\theta}_i^+ \quad (4.57)$$

Z omezení momentů v pohonech a úhlových zrychlení plynou podmínky pro realizovatelné hodnoty \ddot{p} , je tedy výhodné vytvořit i průnik těchto dvou intervalů:

$$L(p, \dot{p}) \leq \ddot{p} \leq U(p, \dot{p}) \quad (4.58)$$

Kde:

$$L(p, \dot{p}) = \max\{L_{akc}(p, \dot{p}), L_{dyn}(p, \dot{p})\} \quad (4.59)$$

$$U(p, \dot{p}) = \min\{U_{akc}(p, \dot{p}), U_{dyn}(p, \dot{p})\} \quad (4.60)$$

Výraz (4.58) představuje výsledný interval údajající realizovatelné hodnoty \ddot{p} . Hodnoty \dot{p} jsou omezeny vztahem (4.48) a limitní křivkou rychlosti, což je křivka v $p - \dot{p}$ diagramu, na které platí:

$$L(p, \dot{p}) = U(p, \dot{p}) \quad (4.61)$$

Pro bod křivky daný hodnotou parametru p , označíme jako $U_{eq}(p)$ hodnotu \dot{p} , pro kterou platí rovnice (4.61). Tato hodnota bude nalezena algoritmicky v další části práce. Protože se musí rychlostní profil v každém bodě nacházet pod limitní křivkou rychlosti, lze napsat podmínku:

$$0 \leq \dot{p} \leq U_{eq}(p) \quad (4.62)$$

Z podmínek (4.48) a (4.62) lze sestavit výsledný realizovatelný interval hodnot \dot{p} :

$$0 \leq \dot{p} \leq U_v(p) \quad (4.63)$$

Kde:

$$U_v(p) = \min\{U_{vel}(p), U_{eq}(p)\} \quad (4.64)$$

4.6 Shrnutí výpočtu

Celý výpočet je vázán na známé polohy v pracovním prostoru robotu. Je zvolena úsečka a body na ní jsou popsány pomocí hodnot parametru $p \in \langle 0, 1 \rangle$. Tato parametrizace je popsána v rovnicích (4.2):

$$\begin{aligned} x(p) &= a_x + p \cdot u_x & u_x &= b_x - a_x \\ y(p) &= a_y + p \cdot u_y & u_y &= b_y - a_y \end{aligned} \quad p \in \langle 0, 1 \rangle$$

V bodě křivky jsou pro danou hodnotu parametru p dosažením $x(p), y(p)$ do vztahů (4.5) a (4.7) vypočteny natočení v pohonech $\theta_1(p), \theta_2(p)$:

$$\boldsymbol{\theta}(p) = [\theta_1(p), \theta_2(p)]^T$$

Ze znalosti $\boldsymbol{\theta}(p)$ jsou v rovnicích (4.10) a (4.11) vyjádřeny členy matic parciálních derivací:

$$\begin{aligned} \boldsymbol{\theta}'(p) &= [\theta_1'(p), \theta_2'(p)]^T \\ \boldsymbol{\theta}''(p) &= [\theta_1''(p), \theta_2''(p)]^T \end{aligned}$$

Z Lagrangeových rovnic II. druhu (4.13) jsou pro parametry RR manipulátoru odvozeny rovnice dynamiky (4.23) a (4.24). Z nich jsou následně podle (4.25) a (4.27) vyjádřeny matice:

$$\mathbf{I}(\boldsymbol{\theta}), \mathbf{C}'(\boldsymbol{\theta}, \boldsymbol{\theta}')$$

Jsou zvolena uvažovaná omezení manipulátoru (4.28), (4.29), (4.30):

$$\begin{aligned} n_i^- - F_i \cdot \dot{\theta}_i &\leq n_i \leq n_i^+ - F_i \cdot \dot{\theta}_i \\ \dot{\theta}_i^- &\leq \dot{\theta}_i \leq \dot{\theta}_i^+ \\ \ddot{\theta}_i^- &\leq \ddot{\theta}_i \leq \ddot{\theta}_i^+ \end{aligned}$$

K původně zadanému řešení dynamických schopností manipulátoru je přidána možnost definovat omezení pro úhlové rychlosti a úhlová zrychlení v pohonech.

4.6.1 Omezení zrychlení

Realizovatelné zrychlení na trajektorii je vázáno na omezení hodnot \ddot{p} pro bod trajektorie (p) a danou rychlost end-effektoru (\dot{p}). Rychlostní profil lze najít algoritmicky ze znalosti realizovatelného kužele v daném bodě $p - \dot{p}$ diagramu, jak bylo zmíněno v Kapitole 3. Na realizovatelné hodnoty \ddot{p} mají vliv limity momentů v pohonech (4.28) a zvolená maximální úhlová zrychlení (4.30).

Z hodnot $\theta(p)$, $\theta'(p)$, $\theta''(p)$, $\mathbf{I}(\theta)$, $\mathbf{C}'(\theta, \theta')$, F_i jsou podle (4.35)-(4.43) vyjádřeny hodnoty omezení od momentů v pohonech:

$$L_{dyn}(p, \dot{p}), U_{dyn}(p, \dot{p})$$

Ze znalosti $\theta'(p)$, $\theta''(p)$, $\dot{\theta}_i^-, \dot{\theta}_i^+$ jsou podle rovnic (4.51)-(4.56) získány omezující hodnoty:

$$L_{akc}(p, \dot{p}), U_{akc}(p, \dot{p})$$

Výsledný interval pro realizovatelné hodnoty zrychlení je získán podle vztahů (4.58), (4.59) a (4.60):

$$L(p, \dot{p}) \leq \ddot{p} \leq U(p, \dot{p})$$

Kde:

$$L(p, \dot{p}) = \max\{L_{akc}(p, \dot{p}), L_{dyn}(p, \dot{p})\}$$

$$U(p, \dot{p}) = \min\{U_{akc}(p, \dot{p}), U_{dyn}(p, \dot{p})\}$$

Tyto hodnoty určují realizovatelný kužel, ve kterém se musí rychlostní profil nacházet, aby byly splněny zavedená omezení.

4.6.2 Omezení rychlostí

Realizovatelné hodnoty \dot{p} jsou vázány na omezení úhlových rychlostí pohonů a limitní křivku rychlosti. Ze znalosti $\dot{\theta}_i^-, \dot{\theta}_i^+, \theta'(p)$ je podle vztahů (4.46)-(4.49) vyjádřena hodnota:

$$U_{vel}(p)$$

Hodnota $U_{eq}(p)$ ze vztahu (4.62) bude získána algoritmicky v následující části práce ze znalosti $L(p, \dot{p})$ a $U(p, \dot{p})$. Výsledný realizovatelný interval hodnot \dot{p} je poté získán pomocí vztahů (4.63) a (4.64):

$$0 \leq \dot{p} \leq U_v(p)$$

Kde:

$$U_v(p) = \min\{U_{eq}(p), U_{vel}(p)\}$$

Posledními podmínkami, jejichž platnost je třeba zkontrolovat pro zajištění realizovatelnosti pohybu manipulátoru, jsou případy nulových hodnot $a_i(p)$, $\theta'_i(p)$. V těchto případech je třeba ověřit platnost vztahů (4.44) a (4.57):

$$n_i^- \leq b_i(p) \cdot \dot{p}^2 + c_i(p) \cdot \dot{p} \leq n_i^+$$

$$\ddot{\theta}_i^- \leq \theta_i''(p) \cdot \dot{p}^2 \leq \ddot{\theta}_i^+$$

Se znalostí realizovatelných hodnot \dot{p} a \ddot{p} je možné provést výpočet rychlostního profilu, který splňuje zvolená omezení manipulátoru a dává představu o realizovatelnosti zachycení a zpomalení letícího předmětu. Hledané řešení musí splňovat podmínky (4.44), (4.57), (4.58) a (4.63)

Algoritmus výpočtu profilu je popsán v následující kapitole.

Kapitola 5

Algoritmizace

Výpočty z předchozí kapitoly je možné využít k sestrojení algoritmu pro určení rychlostního profilu zachycení předmětu. Algoritmus by měl vypočítat průběh pro zadané parametry manipulátoru, zadané hodnoty počátečního a koncového bodu zachytové úsečky, hmotnost zachytávaného tělesa a čas potřebný pro vykonání zachycení mechanismem end-effectoru. Měl by zároveň být dostatečně jednoduchý a robustní, aby byla umožněna optimalizace trajektorie, problém tedy omezíme pouze na hledání řešení, které neprotíná limitní křivku rychlosti. Bude se jednat o průběh s jedním přepínacím bodem mezi akcelerací a decelerací, kdy místo přepínacího bodu bude v profilu umístěna část pohybu s konstantní rychlostí end-effectoru.

Vstupní parametry algoritmu lze pro přehlednost rozdělit na volené a odvozené. Pro ukázkový výpočet jsou zvoleny následující parametry:

$$\begin{array}{ll} a_x = 3 \text{ m} & a_y = 1,5 \text{ m} \\ b_x = -3 \text{ m} & b_y = 1,5 \text{ m} \\ l_1 = 2,2 \text{ m} & l_2 = 2,2 \text{ m} \\ n_1 = 100 \text{ Nm} & n_2 = 70 \text{ Nm} \\ \dot{\theta}_1^+ = \dot{\theta}_2^+ = 7 \text{ rad/s} & \ddot{\theta}_1^+ = \ddot{\theta}_2^+ = 10 \text{ rad/s}^2 \\ \rho = 7850 \text{ kg/m}^3 & M = 5 \text{ kg} \\ F_1 = 4 \text{ N}^2\text{m}^2/\text{W} & F_2 = 4 \text{ N}^2\text{m}^2/\text{W} \\ t_{ef} = 0.5 \text{ s} & \end{array}$$

Kde ρ je zvolená hustota materiálu manipulátoru a t_{ef} je čas potřebný pro vykonání zachycení. Ostatní parametry již byly v práci zmíněny. Odvozené parametry jsou následující:

$$\begin{aligned}
t_1 &= \frac{l_1}{2} = 1,1 \text{ m} & t_2 &= \frac{l_2}{2} = 1,1 \text{ m} \\
d_1 &= \frac{l_1}{50} = 0,044 \text{ m} & d_2 &= \frac{l_1}{50} = 0,044 \text{ m} \\
m_1 &= \frac{\pi \cdot d_1^2}{4} \cdot l_1 \cdot \rho = 26,2596 \text{ kg} & m_2 &= \frac{\pi \cdot d_2^2}{4} \cdot l_2 \cdot \rho = 26,2596 \text{ kg} \\
I_{1T1} &= \frac{1}{12} \cdot m_1 l_1^2 = 10,5914 \text{ kg} \cdot \text{m}^2 & I_{2T2} &= \frac{1}{12} \cdot m_2 l_2^2 = 10,5914 \text{ kg} \cdot \text{m}^2
\end{aligned}$$

Parametry jsou zvoleny pouze orientační a neodpovídají hodnotám reálného manipulátoru. Hodnoty slouží pro teoretický vývoj algoritmu, který by bylo následně možné aplikovat na parametry reálných robotů. Pro výpočet je uvažován zjednodušený model složený ze dvou válcových těles. Průměry ramen d_1, d_2 manipulátoru jsou zvoleny s fixním poměrem vůči jejich délce. V praxi by bylo nutné rozměry volit s ohledem na namáhání jednotlivých těles v průběhu zachycení. Jako křivka byla pro tento příklad zvolena horizontální úsečka symetrická podle osy y .

5.1 Princip algoritmu

Profil je rozdělen na zrychlování z počátečního bodu, zachycení předmětu a zpomalování do koncového bodu. Pomocí známých hodnot $L(p, \dot{p}), U(p, \dot{p})$ z rovnice (4.58) provedeme dopřednou integraci $U(p, \dot{p})$ z počátečního bodu a zpětnou integraci $L(p, \dot{p})$ z koncového bodu. Nalezneme průsečík akcelerace s decelerací a průběhy sjednotíme do realizovatelného profilu s jedním přepínacím bodem, který je v každém bodě (kromě přepínacího) tečný k realizovatelnému kuželi zrychlení. Mezi zrychlování a zpomalování následně umístíme část profilu s konstantní rychlostí, jejíž délka je určena potřebnou dobou pro zachycení t_{ef} . Celý profil zkontrolujeme na podmínky realizovatelných rychlostí (4.44), (4.57) a (4.63).

5.2 Numerická integrace

Pro výpočet rychlostního profilu je zvolena metoda numerické integrace. Jedná se o výpočet, kdy v každém bodě diagramu vycházíme z limitních hodnot \ddot{p} a ze známé hodnoty zrychlení určujeme průběh rychlosti.

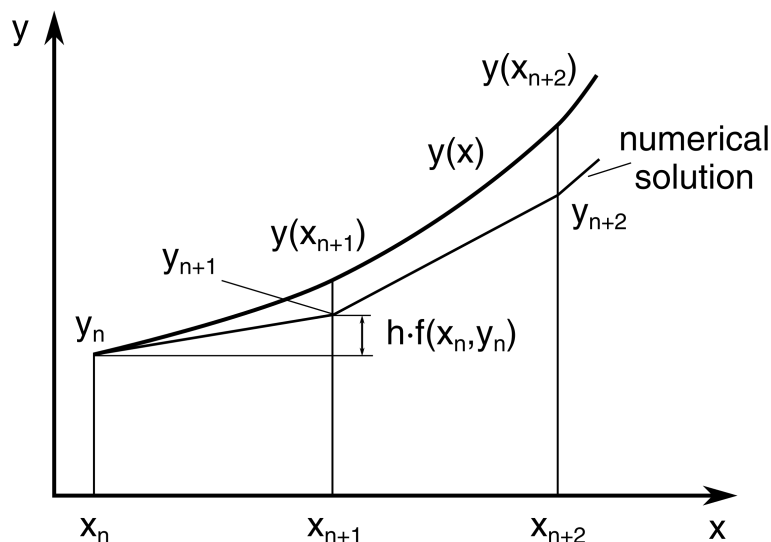
Je potřeba provést dva výpočty:

1. Dopředná integrace realizovatelného zrychlení z počátečního bodu
2. Zpětná integrace realizovatelného zpomalení z koncového bodu

Nejprve zvolíme způsob numerického výpočtu. Je možné použít metody známé z numerické matematiky, z nichž výpočetně nejjednodušší je Eulerova dopředná metoda

zobrazená na Obr. 5.1. Jednoduchost metody je výhodná pro následnou optimalizaci trajektorie. Body jsou v metodě hledány s fixním krokem ve směru osy x . Specifickým problémem naší úlohy je fakt, že v počátečním a koncovém bodě je tečna rychlostního profilu svislá, v tomto případě by tedy byl fixní krok ve směru osy x problematický a výpočet je potřeba upravit.

Eulerova dopředná metoda je definována takto:



Obr. 5.1: Eulerova metoda [17]

$$y(x+h) = y(x) + hf(x, y) \quad (5.1)$$

5.2.1 Dopředná integrace

Požadavkem je funkce, která vypočte průběh zrychlování manipulátoru od počátečního bodu a počáteční rychlosti do zvoleného koncového bodu. Fixní hodnota kroku ve směru osy x je příčinou velké nepřesnosti v případech, kdy se tečna rychlostního profilu blíží svislici. Proto bude fixní krok ve směru osy x nahrazen fixním krokem ve směru tečny. Platí vztahy:

$$f = y' = \tan(\alpha) = \frac{U}{\dot{p}} \quad (5.2)$$

$$y(x + h \cdot \cos(\alpha)) = y(x) + h \cdot \sin(\alpha)$$

Princip výpočtu je zjednodušeně popsán v Algoritmu 5.1. Nejedná se však o fungující kód, ty jsou uvedeny v příloze práce. Tímto způsobem budou v práci zobrazeny i další výpočty v části algoritmizace.

```

1  p = p_start;
2  dp_dt = dp_start;
3  h = 0.001;
4
5  while p < p_end
6      [~,U] = function(p,dp_dt);
7      alpha = atan(U/dp_dt);
8      delta_p = delta*cos(alpha);
9      delta_dp = delta*sin(alpha);
10     p = p + delta_p;
11     dp_dt = dp_dt + delta_dp;
12 end

```

Algoritmus 5.1: Dopředná integrace

Princip je takový, že jsou z počátečních hodnot $p_{start}, \dot{p}_{start}$ pomocí tečen počítány následující hodnoty, dokud není dosaženo hodnoty $p > p_{end}$. Tečny vycházejí z hodnot $U(p, \dot{p})$. Hodnoty p, \dot{p}, U jsou v tomto procesu ukládány do vektorů, aby je bylo možné využít v dalších výpočtech.

5.2.2 Zpětná integrace

Zpětná integrace je obdobná jako dopředná. Opět místo konstantního kroku ve směru osy x použijeme krok ve směru tečny. Pro algoritmus pak platí rovnice:

$$f = y' = \tan(\alpha) = \frac{L}{\dot{p}} \quad (5.3)$$

$$y(x - h \cdot \cos(\alpha)) = y(x) - h \cdot \sin(\alpha)$$

```

1  p = p_start;
2  dp_dt = dp_start;
3  h = 0.001;
4
5  while p > p_end
6      [L,~] = function(p,dp_dt);
7      alpha = atan(L/dp_dt);
8      delta_p = delta*cos(alpha);
9      delta_dp = delta*sin(alpha);
10     p = p - delta_p;
11     dp_dt = dp_dt - delta_dp;
12 end

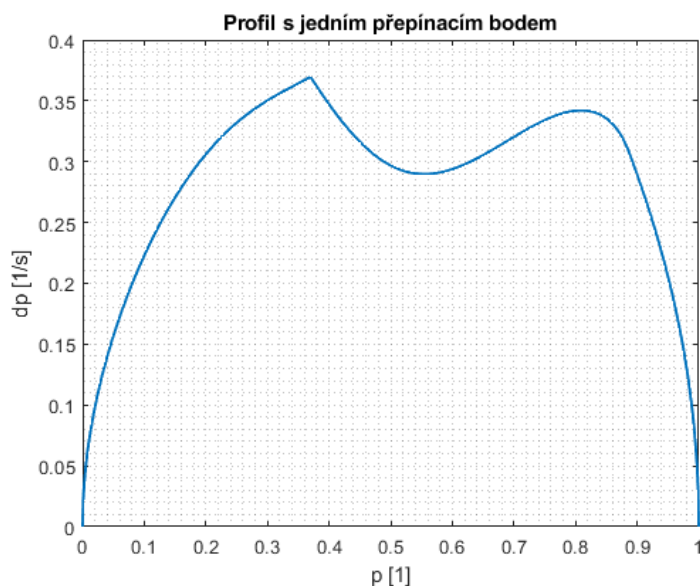
```

Algoritmus 5.2: Zpětná integrace

Algoritmus je téměř stejný jako v případě dopředné integrace, v tomto případě jsou použity limitní hodnoty zpomalení $L(p, \dot{p})$, jak je naznačeno v Algoritmu 5.2. Je třeba zdůraznit, že zpomalování provádí manipulátor až po zachycení samotného tělesa, proto je nutné zpětnou integraci počítat s nenulovou hodnotou M .

Hodnoty p, \dot{p}, L jsou opět ukládány do vektorů, aby byly k dispozici pro další výpočty. Veškeré výsledky z dopředné a zpětné integrace jsou vypočteny pro měnící se kroky parametru p . Po samotném výpočtu je tedy provedena diskretizace hodnot

p a v těchto bodech je provedena interpolace průběhů zrychlení a zpomalení. Průběhy jsou poté sjednoceny do profilu s jedním přepínacím bodem mezi akcelerací a decelerací. Profil je zobrazen na Obr. 5.2.



Obr. 5.2: Výsledky numerické integrace

5.3 Výpočet konstantní části profilu

Průběhy vypočtené v numerických integracích jsou základem pro tvorbu rychlostního profilu se zachycením. Do vzniklého průběhu s jedním přepínacím bodem mezi akcelerací a decelerací je nutné přidat úsek s konstantní hodnotou \dot{p} . Tato hodnota musí být udržena na úseku dlouhém p_{konst} , který je dán vztahem:

$$p_{konst} = \dot{p} \cdot t_{ef} \quad (5.4)$$

Hodnota t_{ef} je časový úsek, který je potřeba k vykonání zachycení mechanismem end-effectoru. Dalším krokem je tedy hledání hodnoty \dot{p} , ve které bude konstantní rychlost udržitelná po dobu t_{ef} . Je zjištěna maximální hodnota \dot{p} profilu s jedním přepínacím bodem. Poté je provedena diskretizace hodnot \dot{p} menších než vypočtené maximum profilu a pro každou hodnotu \dot{p} jsou nalezeny průsečíky profilu s konstantními funkcemi o hodnotě daného \dot{p} . Průsečíky jsou pro každou úroveň uspořádány chronologicky do vektoru X . V případě aktuálního profilu může vektor X mít 1 až 4 prvky. Mezi jednotlivými průsečíky je následně hledána potenciální konstantní část profilu. Konstantní úsek profilu je možné pro danou hodnotu \dot{p} hledat mezi jednotlivými průsečíky v závislosti na jejich počtu, jak je znázorněno v Algoritmu 5.3. Proměnnou p_{pot} je označen potenciální konstantní úsek profilu, tím je na každé úrovni s hodnotou \dot{p} největší vzdálenost průsečíků, mezi nimiž má profil vyšší hodnoty než dané \dot{p} . Na Obr. 5.3 je znázorněna diskretizace hodnot \dot{p} .

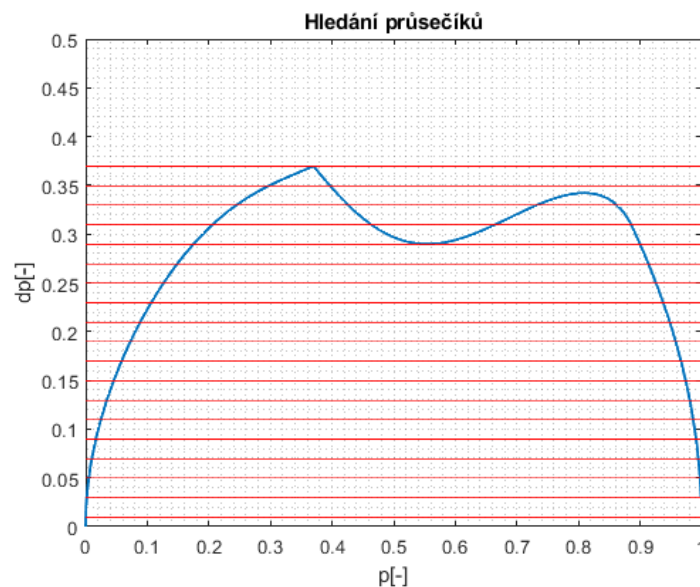
```

1  if length(X) == 2;
2      p_pot = X(1,2)-X(1,1);
3  elseif length(X) == 4
4      p_pot1 = X(1,2)-X(1,1);
5      p_pot2 = X(1,4)-X(1,3);
6      p_pot = max([p_pot1 p_pot2])
7  else
8      p_pot = 0;

```

Algoritmus 5.3: Průsečíky profilu

V algoritmu jsou zanedbány situace, kdy je konstantní funkce tečnou profilu. Případ, kdy existují 3 průsečíky, je tedy opomíjen. Po obdržení průsečíků pro každou hodnotu rychlosti je algoritmicky zjišťováno, pro jakou nejvyšší hodnotu \dot{p} platí $p_{pot} \geq p_{konst}$ a v tomto místě je do profilu umístěna konstantní část. V úseku je ještě provedena kontrola na splnění podmínky (4.58) $L(p, \dot{p}) \leq \ddot{p} \leq U(p, \dot{p})$. Celý profil je dále zkontrolován pro splnění podmínky hodnot \dot{p} (4.63) a splnění podmínek nulového dělení (4.44) a (4.57). O těchto podmínkách se zmíníme v dalších sekcích práce. Pokud není některá podmínka splněna, je pokračováno ve snižování hodnoty \dot{p} , dokud není nalezen vyhovující výsledek. Pokud by takový výsledek nebyl nalezen ani po dosažení hodnoty $\dot{p} = 0$, byl by rychlostní profil nastaven konstantní roven 0.



Obr. 5.3: Hledání průsečíků profilu

Výpočet je zjednodušeně znázorněn v Algoritmu 5.4. Výrazem *podmínka* je myšleno ověření, jestli profil s konstantním úsekem o dané hodnotě \dot{p} splňuje podmínky (4.44), (4.57), (4.58) a (4.63). Je nutno zdůraznit, že na hledané konstantní části profilu již je do výpočtu dynamiky dosazována nenulová hodnota hmotnosti zachyceného tělesa M .

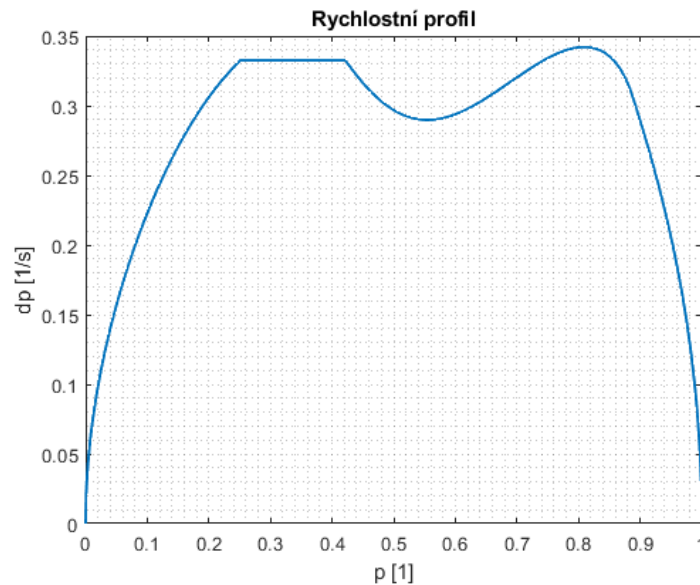
```

1  success = 0;
2  dp_dt = max(profil)
3  delta = 0.001;
4
5  while dp_dt > 0 & success == 0
6      dp_dt = dp_dt - delta;
7      X = function(dp_dt);
8      p_pot = function(X);
9      p_konst = function(dp_dt, t_ef);
10     podminka == function(dp_dt);
11     if podminka = 1 & p_pom > p_konst
12         success = 1;
13     else
14         success = 0;
15     end
16 end

```

Algoritmus 5.4: Hledání rychlostního profilu

Příklad výsledku algoritmu je zobrazen na Obr. 5.4. Z výsledku je patrné, že zachycovací část profilu nemusí být částí s nejvyšší rychlostí end-effektoru. To je v tomto případě pohybu způsobeno tím, že se manipulátor ve třetí části profilu dostává do konfigurace, kdy je kvůli limitům manipulátoru realizovatelné pouze zvyšování rychlosti a platí $L(p, \dot{p}) > 0$.



Obr. 5.4: Vypočtený rychlostní profil

Pokud bychom chtěli zobrazit skutečné hodnoty rychlostí v metrech za sekundu, stačí vypočtený profil vynásobit délkou trajektorie:

$$|AB| = \sqrt{(b_x - a_x)^2 + (b_y - a_y)^2} \quad (5.5)$$

$$v(p) = \dot{p}(p) \cdot |AB| \quad (5.6)$$

V tomto případě dosahuje záchytitelná rychlost tělesa hodnoty 1,996 m/s.

5.4 Výpočet limitní křivky rychlosti

V Algoritmu 5.4 je třeba provést kontrolu podmínky limitní křivky rychlosti (4.63). Nejprve je nutné vyjádřit hodnoty $U_{eq}(p)$, tedy hodnoty \dot{p} , ve kterých pro dané p platí vztah:

$$L(p, \dot{p}) = U(p, \dot{p})$$

S vyjádřenými vztahy pro výpočet $L(p, \dot{p})$ a $U(p, \dot{p})$ lze pro každou hodnotu p nalézt $U_{eq}(p)$ výpočtem znázorněným v Algoritmu 5.5.

```

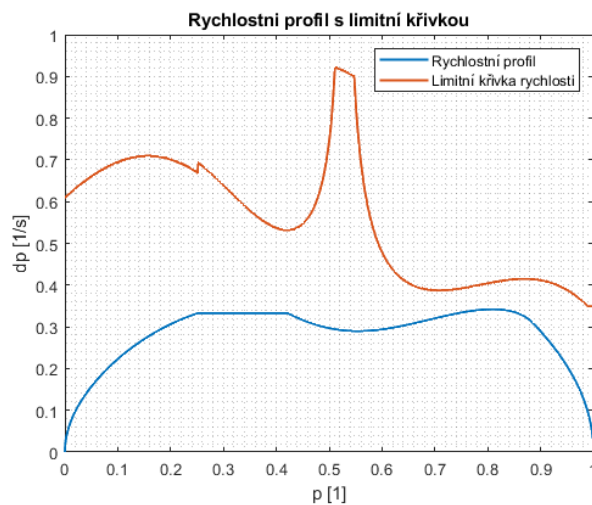
1  dp_dt = 0;
2  [L,U] = function(p, dp_dt);
3
4  if U > L
5      while U > L
6          dp_dt = dp_dt + 0.001;
7          [L,U] = function(p, dp_dt);
8      end
9  else
10 end
11 U_eq = dp_dt;
```

Algoritmus 5.5: Algoritmus limitní křivky rychlosti

Jak je vidět, je nastavena hodnota \dot{p} , pro ni jsou vypočteny limity L a U , poté je hodnota \dot{p} zvyšována o zvolený krok, dokud neplatí $L > U$. Takto získáme hodnotu $U_{eq}(p)$ z rovnice (4.62). Tu je třeba porovnat s omezením od úhlových rychlostí pohonů U_{vel} vyjádřeným v rovnici (4.49). Řešíme podmínky ze vztahů (4.63) a (4.64):

$$\dot{p} \leq U_v(p)$$

$$U_v(p) = \min\{U_{eq}(p), U_{vel}(p)\}$$



Obr. 5.5: Profil a limitní křivka rychlosti

Podmínky $U_{eq}(p)$ a $U_{vel}(p)$ sjednotíme a limitní křivku rychlosti sestavíme z hodnot $U_v(p)$. Vypočtený rychlostní profil i s limitní křivkou je zobrazen na Obr. 5.5. Kontrola limitní křivky rychlosti je prováděna v každém cyklu Algoritmu 5.4, pokud není splněna, je pokračováno ve snižování hodnoty \dot{p} .

5.5 Řešení nulového dělení

V každém kroku Algoritmu 5.4 je také nutné provést kontroly nulových dělení (4.44) a (4.57):

$$\begin{aligned} a_i(p) &= 0 \\ \theta'_i(p) &= 0 \end{aligned}$$

Při diskrétním výpočtu je velmi nepravděpodobné, že by algoritmus dospěl přesně do situace, kdy nastane jedna z těchto možností. V případě, kdy by se tak stalo, dochází k dělení nulou. Program Matlab je schopný tuto operaci provést a jako výsledek dává hodnoty $\pm\infty$. Poté vychází pro daný pohon realizovatelná hodnota \dot{p} z intervalu $(-\infty, \infty)$ a omezení přechází na hodnotu \dot{p} . V případě, kdy $a_i(p) = 0$, řešíme podmínku (4.44):

$$n_i^- \leq b_i(p) \cdot \dot{p}^2 + c_i(p) \cdot \dot{p} \leq n_i^+$$

Pro $\theta'_i(p) = 0$ řešíme podmínku (4.57):

$$-\ddot{\theta}_{max} \leq \theta''_i(p) \cdot \dot{p}^2 \leq \ddot{\theta}_{max}$$

Tyto podmínky je třeba ověřit i v případě výpočtu, kdy z důvodu diskretizace tyto situace nenastanou. Proto jsou v algoritmu průběhy $a_i(p)$ a $\theta'_i(p)$ uloženy do vektoru, následně je zjištěno, jestli tyto průběhy protínají nulovou hodnotu. Pokud ano, jsou pomocí interpolace zjištěny hodnoty p , ve kterých by probíhalo dělení nulou. Pro tyto hodnoty je následně provedena kontrola splnění podmínek pro hodnotu \dot{p} . Pokud není splněna některá z podmínek, je v Algoritmu 5.4 pokračováno ve snižování hodnoty \dot{p} .

5.6 Vykreslení požadovaných veličin

Z vypočteného rychlostního profilu je možné získat veličiny, které jsou pro nás z hlediska pohybu manipulátoru nejpodstatnější. Jedná se především o momenty v pohonech, úhlové rychlosti a úhlová zrychlení. V první řadě je ale nutné dopočítat závislost průběhů na čase pro zjištění předstihu, s jakým je třeba zahájit pohyb manipulátoru před příletem tělesa do místa zachycení.

Přepočet na časovou závislost lze analyticky vyjádřit tímto vztahem:

$$T_c = \int_0^1 \frac{1}{\dot{p}(p)} dp \quad (5.7)$$

Kde T_c vyjadřuje celkovou dobu trvání pohybu manipulátoru. Rychlostní profil je však diskretizovaný, jsou tedy známy hodnoty \dot{p} příslušící daným hodnotám parametru p . Pokud je n hodnot parametrů, lze profil rozdělit na $n - 1$ intervalů, kde pro i od 1 do $n - 1$ bude platit (volíme indexování od jedné stejně jako v případě programu Matlab):

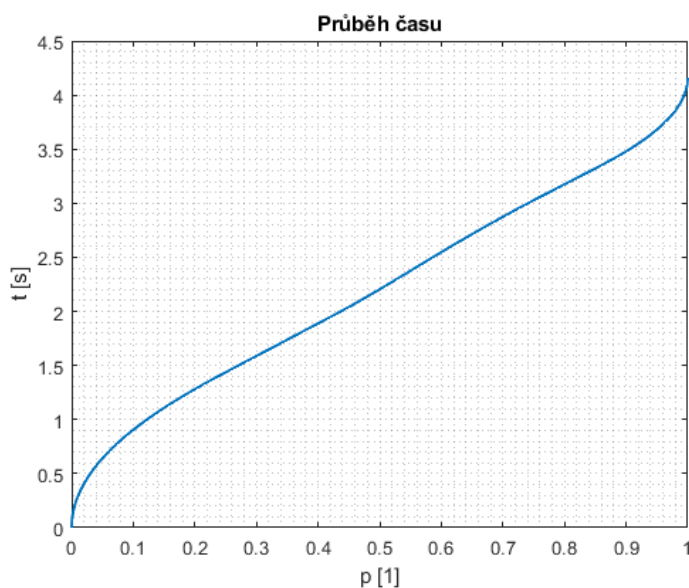
$$v_{i+1} = \frac{\dot{p}_i + \dot{p}_{i+1}}{2} \quad (5.8)$$

$$t_{i+1} = \frac{p_{i+1} - p_i}{v_{i+1}} \quad (5.9)$$

Kde:

t_{i+1} je trvání pohybu mezi bodem p_i a p_{i+1}

v_{i+1} je průměrná rychlost mezi body p_i a p_{i+1}



Obr. 5.6: Časový průběh podél křivky

Kumulativním sčítáním trvání jednotlivých úseků je pak pro bod p_{i+1} vyjádřena hodnota $t_{(i+1),abs}$, za kterou se bude manipulátor v daném bodě nacházet od počátku pohybu:

$$t_{(i+1),abs} = \sum_{j=1}^{i+1} t_j \quad (5.10)$$

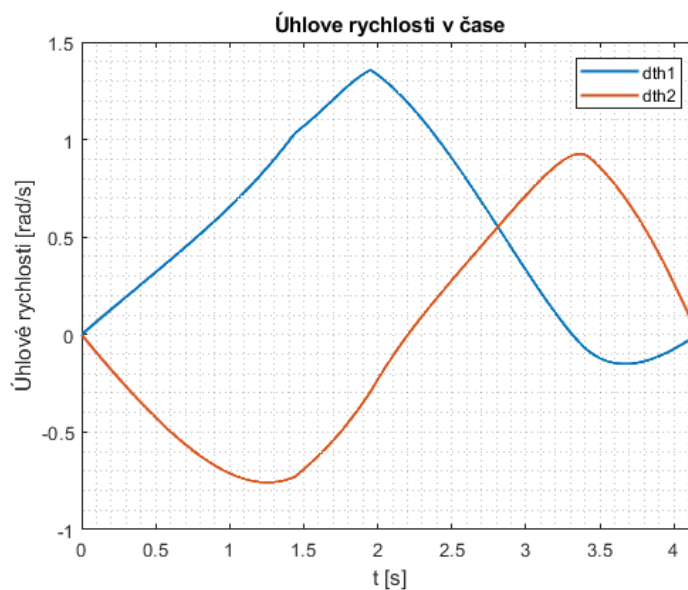
Pro počátek křivky platí $t_1 = 0$. Na Obr. 5.6 jsou vykresleny absolutní časové hodnoty pro celou trajektorii, z grafu je možné odečíst i celkovou dobu trvání pohybu, která se v tomto případě rovná 4,16 s.

Se známými hodnotami času pro jednotlivé hodnoty parametru je již snadné všechny následující závislosti zobrazovat v závislosti na čase. Nejprve zobrazíme pomocí vztahů (3.7) a (3.8) průběhy úhlových rychlostí a úhlových zrychlení v pohonech:

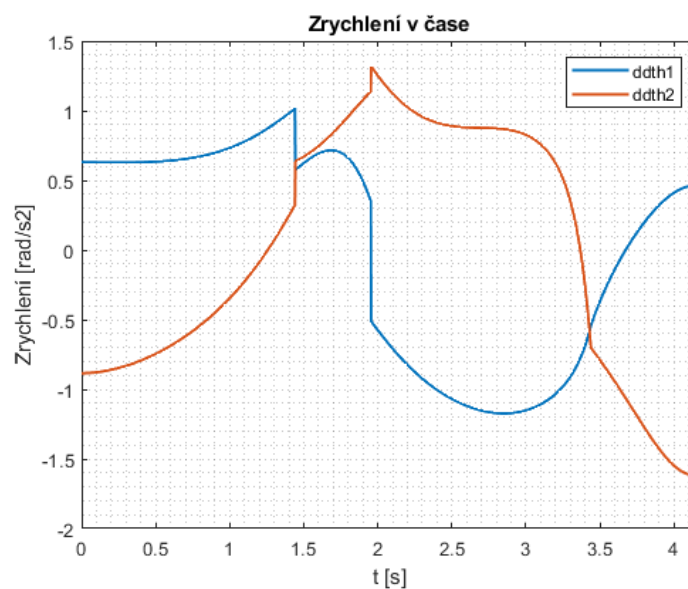
$$\dot{\theta} = \theta' \cdot \dot{p}$$

$$\ddot{\theta} = \theta'' \cdot \dot{p}^2 + \theta' \cdot \ddot{p}$$

Na Obr. 5.7 a 5.8 jsou zobrazeny závislosti úhlových rychlostí a zrychlení na čase.



Obr. 5.7: Časový průběh úhlových rychlostí

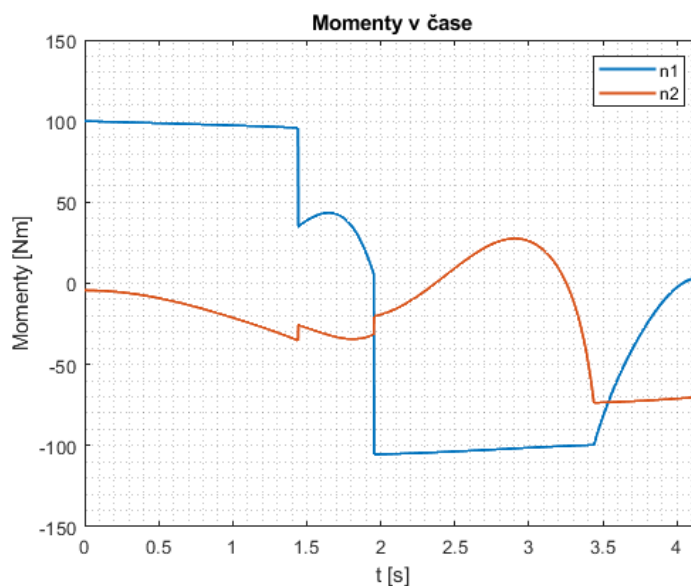


Obr. 5.8: Časový průběh úhlových zrychlení

Všechny potřebné členy rovnic dynamiky je již možné vyjádřit v závislosti na parametru i čase, momenty poté lze vyjádřit z rovnice (4.12):

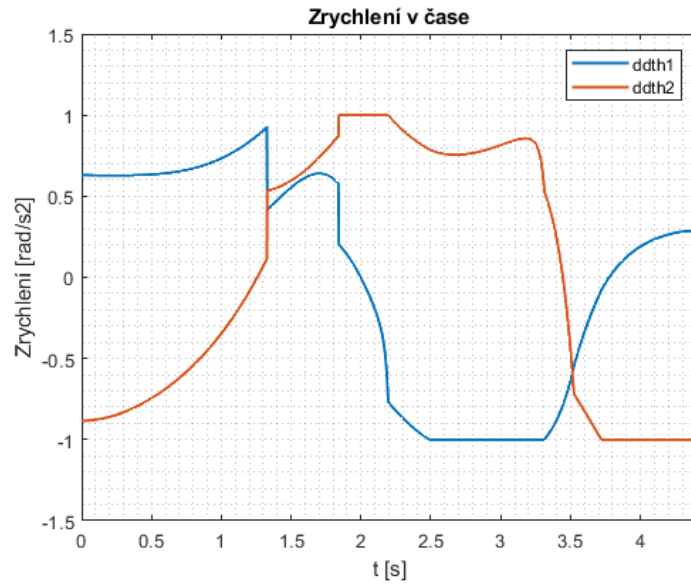
$$\mathbf{n} = \mathbf{I}(\boldsymbol{\theta}) \ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$$

Momenty v závislosti na čase jsou zobrazeny na Obr. 5.9. Z průběhů je možné vidět, že alespoň jeden z momentů pohonů je, s výjimkou části zachytávání předmětu, na svém limitu. Omezení úhlových rychlostí tedy v tomto případě do průběhů nevstupují.

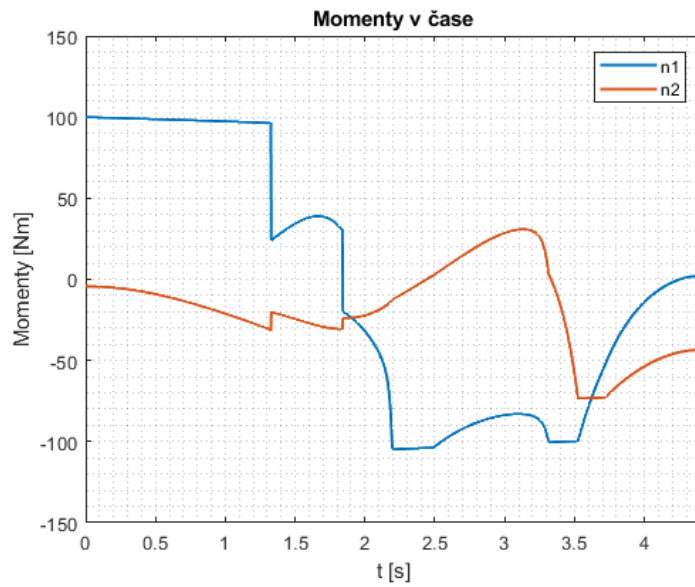


Obr. 5.9: Časový průběh momentů

Pro kontrolu správnosti algoritmu je tedy ještě nastavena snížená hodnota omezení úhlových zrychlení $\ddot{\theta}_1^+ = \ddot{\theta}_2^+ = 1 \text{ rad/s}^2$ a pro ni je znovu proveden výpočet rychlostního profilu. Průběh úhlových zrychlení profilu se sníženými hodnotami omezení pro $\ddot{\boldsymbol{\theta}}$ je zobrazen na Obr. 5.10 a průběh momentů na Obr. 5.11. Z průběhů je vidět, že se při pohybu střídají úseky na limitu kroutících momentů pohonů a na limitu zvolených hodnot úhlových zrychlení. Lze tedy usuzovat, že výpočetní algoritmus funguje korektně a respektuje zvolená omezení.



Obr. 5.10: Časový průběh úhlových zrychlení pro $\ddot{\theta}_i^+ = 1 \text{ rad/s}^2$



Obr. 5.11: Časový průběh momentů pro $\ddot{\theta}_i^+ = 1 \text{ rad/s}^2$

Kapitola 6

Optimalizace

V předchozí části práce byly parametry výpočtu náhodně zvoleny. Tímto způsobem není dosaženo maximálních možností manipulátoru. Z toho důvodu nyní na výpočetní algoritmus aplikujeme vybranou optimalizační metodu a porovnáme výsledky s původními pro náhodně zvolené parametry. Nejprve definujeme situace, ve kterých by mohla být optimalizace výhodná, budeme uvažovat tyto.

1. Jsou dány parametry manipulátoru a cílem je upravit polohu záchytové úsečky v pracovním prostoru za účelem dosažení co nejvyšší zachytitelné rychlosti letícího tělesa.
2. Jsou dány parametry manipulátoru, ale cílem je naplánovat zachycení tělesa letícího rychlostí menší, než je maximální zachytitelná rychlost manipulátoru.
3. Jsou dány pouze pohony manipulátoru a cílem je zjistit rozměry ramen a polohu úsečky pro umožnění maximálního zachycení.

Úprava polohy úsečky je uvažována pomocí umístění manipulátoru na říditelný mechanismus, nebo změnou polohy plavidla, ke kterému je manipulátor uchycen.

6.1 Funkce rychlostního profilu

Pro výše uvažované případy optimalizace je třeba definovat kritériální funkce. Limitní rychlostní profil, jehož výpočet byl proveden v předchozích částech práce, bude základem těchto optimalizačních úloh. Je převeden do podoby Matlab funkce, která při zadání všech potřebných parametrů spočítá průběh \dot{p} na limitu kinematických a dynamických možností manipulátoru. Zde jsou uvedeny všechny vstupní parametry:

- a_x x souřadnice počátečního bodu
- a_y y souřadnice počátečního bodu

b_x	x souřadnice koncového bodu
b_y	y souřadnice koncového bodu
l_1	délka ramene 1
l_2	délka ramene 2
t_1	poloha těžiště ramene 1
t_2	poloha těžiště ramene 2
m_1	hmotnost ramene 1
m_2	hmotnost ramene 2
I_{1T1}	moment setrvačnosti ramene 1
I_{2T2}	moment setrvačnosti ramene 2
n_1	omezení kroutícího momentu pohonu 1
n_2	omezení kroutícího momentu pohonu 2
$\dot{\theta}_1^+$	omezení úhlové rychlosti pohonů
$\ddot{\theta}_1^+$	omezení úhlového zrychlení pohonů
$\dot{\theta}_2^+$	omezení úhlové rychlosti pohonů
$\ddot{\theta}_2^+$	omezení úhlového zrychlení pohonů
M	hmotnost zachytávaného tělesa
F_1	konstanta závislosti omezení pohonu 1 na úhlové rychlosti
F_2	konstanta závislosti omezení pohonu 2 na úhlové rychlosti
t_{ef}	doba uzavření mechanismu end-effectoru

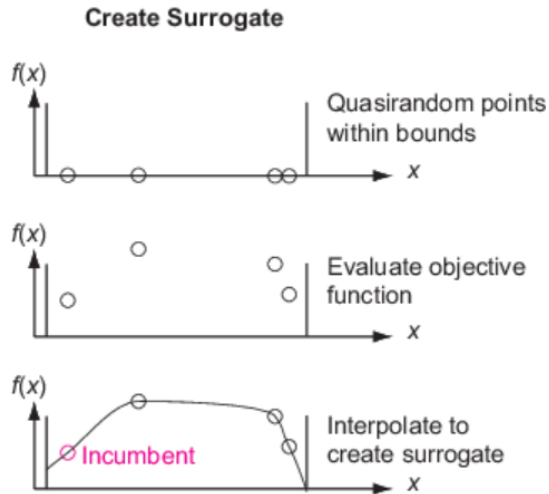
Funkce provádí veškeré výpočty a kontroly profilu uvedené v kapitole algoritmizace. V případě, že by byly zadány parametry křivky, při kterých by se její část nacházela mimo pracovní prostor robotu, bude jako výsledek funkce nastaven automaticky konstantní profil $\dot{p} = 0$. Funkce je uvedena v příloze práce. Z hlediska optimalizace je klíčovým výstupem hodnota zachytitelné rychlosti tělesa $v_{konst} = \dot{p}_{konst} \cdot |AB|$.

6.2 Surrogate optimalizace

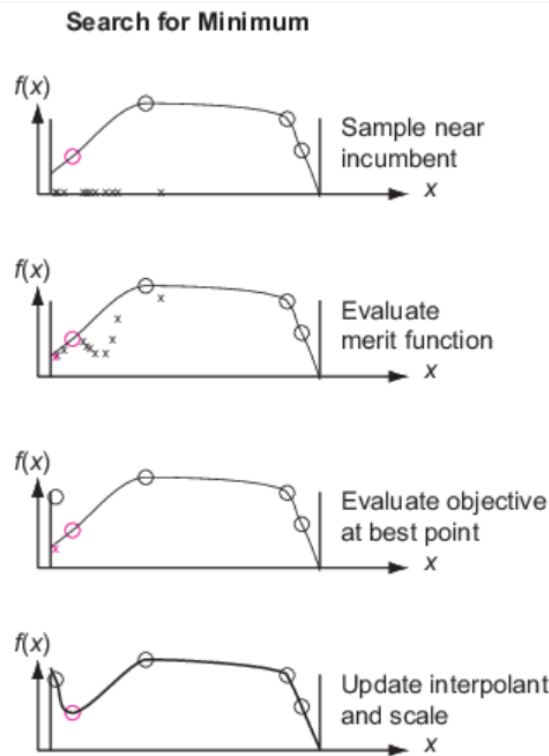
Pro optimalizace použijeme metodu surrogate optimization v programu Matlab. Jedná se o globální negradientní metodu, která je vhodná pro optimalizace s časově náročnými kriteriálními funkcemi, mezi které lze zařadit i výpočet rychlostního profilu. Algoritmus má dvě fáze [18]:

1. Vytvoření náhraní funkce (surrogate)
2. Hledání minima funkce

Během první fáze jsou vzaty náhodné body z definované oblasti vstupních parametrů, ve kterých je vyhodnocena časově náročná kriteriální funkce. Z nich je poté



Obr. 6.1: Fáze vytvoření náhradní funkce [18]



Obr. 6.2: Fáze hledání minima [18]

vytvořen surrogate (náhradní funkce), který vzniká interpolováním radiální bázové funkce těmito body [18]. Tato fáze je zobrazena na Obr. 6.1.

Během fáze hledání minima je náhodně zvoleno velké množství bodů uvnitř vybrané oblasti vstupních parametrů a je v nich vyhodnocena náhradní funkce. Mezi získanými hodnotami jsou poté hledány nejnižší výsledky, v těch je hodnota náhradní funkce porovnána s hodnotou původní časově náročné funkce. Po tomto porovnání je aktualizována náhradní funkce a jsou opět hledána minima [18]. Proces je znázor-

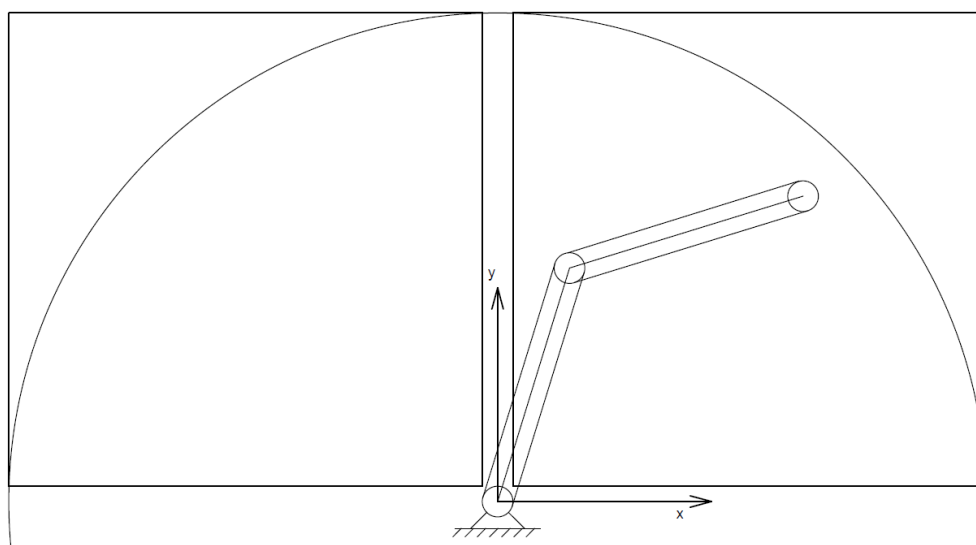
něn na Obr. 6.2, slovem Incumbent je označen bod s nejnižší hodnotou kriteriální funkce od posledního resetu surrogate funkce.

6.3 Optimalizace záchytové úsečky

První optimalizace je zaměřena na případ, kdy jsou známy parametry manipulátoru a cílem je pouze výhodné umístění záchytové úsečky v pracovním prostoru manipulátoru. Je zaveden vektor optimalizovaných parametrů:

$$P_{kin} = [a_x, a_y, b_x, b_y]^T \quad (6.1)$$

Zbylé parametry jsou použity shodně s ukázkovými parametry v předchozím výpočtu. Dále je třeba určit oblast optimalizovaných parametrů, na které bude hledáno minimum kriteriální funkce. Oblast je omezena obdélníkem opsaným pracovnímu prostoru robotu, počáteční bod je hledán I. a koncový bod ve II. kvadrantu, jak je zobrazeno na Obr. 6.3. Tato oblast by v praxi mohla být upravena v závislosti na uchycení manipulátoru, případně na uvažovaných omezeních pracovního prostoru způsobených dorazy pohonů.



Obr. 6.3: Oblasti hledaných bodů úsečky

Je zavedena oblast:

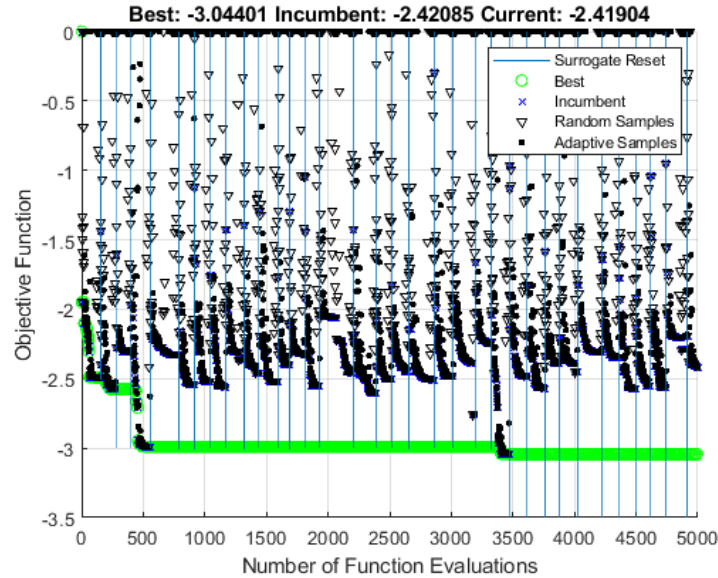
$$\begin{aligned} 0,01 &\leq a_x \leq l_1 + l_2 [m] \\ 0,01 &\leq a_y \leq l_1 + l_2 [m] \\ -(l_1 + l_2) &\leq b_x \leq -0,01 [m] \\ 0,01 &\leq b_y \leq l_1 + l_2 [m] \end{aligned} \quad (6.2)$$

Pokud je do funkce rychlostního profilu algoritmem dosazena úsečka s body mimo pracovní prostor robotu, je profil nastaven na konstantní roven nule. Cílem je nale-

zení profilu s co největší záchyťovou rychlostí, kritériální funkce je tedy nastavena takto:

$$CF_{kin} = -\dot{p}_{konst} \cdot |AB| = -v_{konst} \quad (6.3)$$

Hledání minima probíhá do přerušení algoritmu, které je zvoleno pro daný počet vyhodnocení funkce. Na Obr. 6.4 je znázorněn průběh optimalizace polohy záchyťové úsečky pro 5000 výpočtů. Svislé čáry v grafu představují resetování surrogate funkce. Zelenou barvou je vždy zobrazen doposud nejlepší výsledek optimalizace.

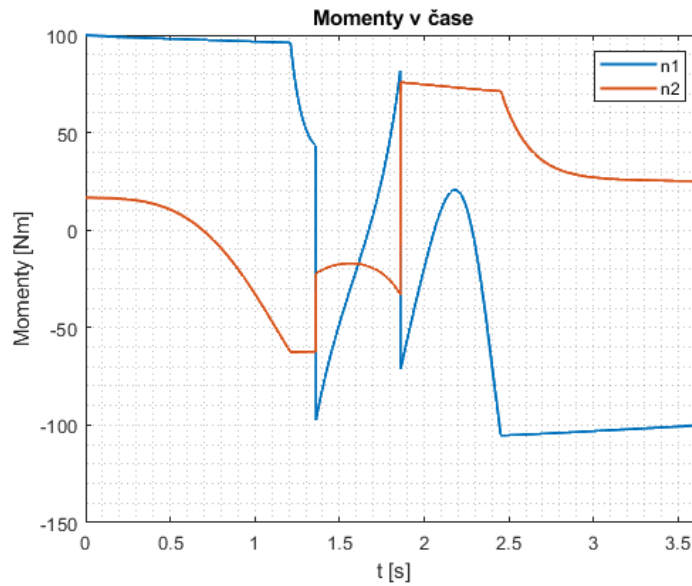


Obr. 6.4: Optimalizace kinematiky

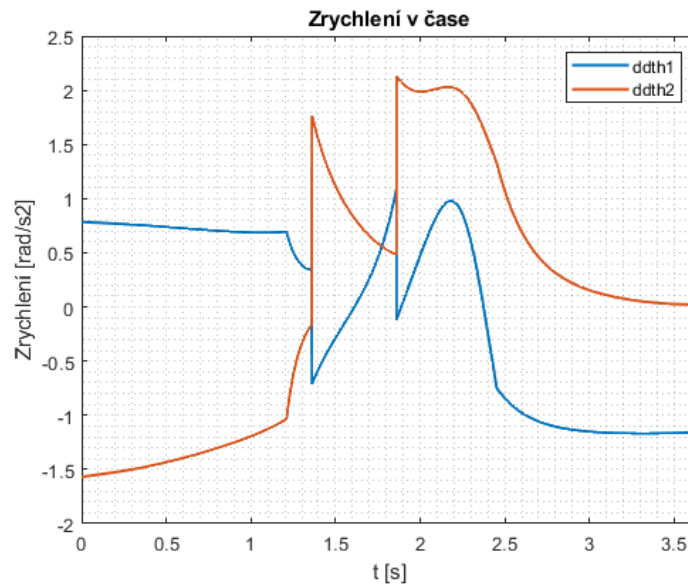
Z hodnoty kritériální funkce je možné vidět, že nejlepší nalezený výsledek po 5000 výpočtech je průběh se zachytitelnou rychlostí 3,04 m/s . Pouhou optimalizací polohy úsečky se tedy podařilo záchyťovou rychlost zvýšit na 152% původní hodnoty 1,996 m/s . Optimalizované parametry mají tyto hodnoty:

$$\begin{aligned} a_x &= 2,98 \text{ m} \\ a_y &= 3,23 \text{ m} \\ b_x &= -0,16 \text{ m} \\ b_y &= 0,22 \text{ m} \end{aligned} \quad (6.4)$$

Všechny rychlostní profily budou pro porovnání zobrazeny ve společném grafu na konci kapitoly optimalizace. Na Obr. 6.5 jsou zobrazeny průběhy momentů pro pohyb po optimalizované trajektorii. Je patrné, že průběh rychlostního profilu po optimalizované úsečce je s výjimkou konstantní části profilu na limitech momentů v pohonech. Úhlová zrychlení v tomto případě nenabývají limitních hodnot, jsou zobrazena na Obr. 6.6.



Obr. 6.5: Momenty po optimalizaci úsečky



Obr. 6.6: Zrychlení po optimalizaci úsečky

6.4 Optimalizace dynamiky

Druhým typem optimalizace jsou úlohy, kdy kromě parametrů křivky hledáme i vhodné parametry manipulátoru. V tomto případě uvažujeme dvě úlohy.

6.4.1 Optimalizace momentů pro dosažení zadané rychlosti

Prvním příkladem této úlohy je situace, kdy bychom daným manipulátorem chtěli nalézt průběh parametrizace pro zachycení tělesa s rychlostí menší, než je maxi-

mální rychlost, kterou je manipulátor schopný zachytit. Dá se tedy říci, že se jedná o hledání časově neoptimálního řešení. Při tomto zachycení je naším cílem dosáhnout pohybu za použití co nejmenších momentů v pohonech. Úlohu lze převést na hledání časově optimálního řešení pro limity pohonů menší, než jsou skutečné limity manipulátoru. Optimalizační parametry volíme takto:

$$P_{dyn,1} = [a_x, a_y, b_x, b_y, n_1, n_2]^T \quad (6.5)$$

Minimum kriteriální funkce bude hledáno na této oblasti:

$$\begin{aligned} 0,01 &\leq a_x \leq l_1 + l_2 [m] \\ 0,01 &\leq a_y \leq l_1 + l_2 [m] \\ -(l_1 + l_2) &\leq b_x \leq -0,01 [m] \\ 0,01 &\leq b_y \leq l_1 + l_2 [m] \\ 0 &\leq n_1 \leq 100 [Nm] \\ 0 &\leq n_2 \leq 70 [Nm] \end{aligned} \quad (6.6)$$

Jsou použity délky ramen a maximální momenty pohonů shodné s dříve zvolenými. Cílem je dosažení nastavené rychlosti při použití co nejmenších momentů v pohonech. Kriteriální funkci definujeme tímto způsobem:

$$CF_{dyn,1} = k_v |v_{set} - \dot{p}_{konst} \cdot |AB|| + k_n (n_1 + n_2) \quad (6.7)$$

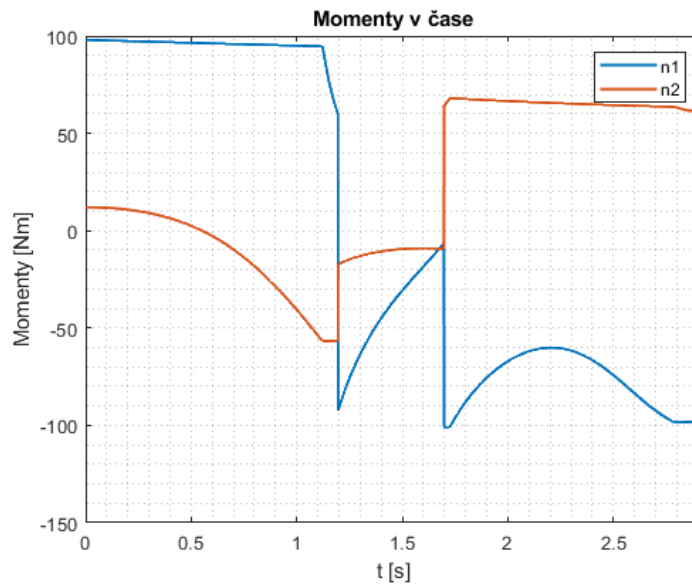
Funkce je váhovým součtem dvou členů. První člen udává přiblížení k nastavené rychlosti, volíme absolutní hodnotu vzdálenosti, protože cílem není překročit nastavenou rychlost a nezáleží nám na směru, ze kterého se k hodnotě přiblížíme. Druhým členem je součet limitů pohonů potřebných k realizaci pohybu. Pomocí váhových koeficientů k_v a k_n je možné regulovat důležitost jednotlivých kritérií. V tomto případě budou zvoleny hodnoty:

$$\begin{aligned} k_v &= 100 \\ k_n &= \frac{1}{n_{1,max} + n_{2,max}} = \frac{1}{170} \end{aligned} \quad (6.8)$$

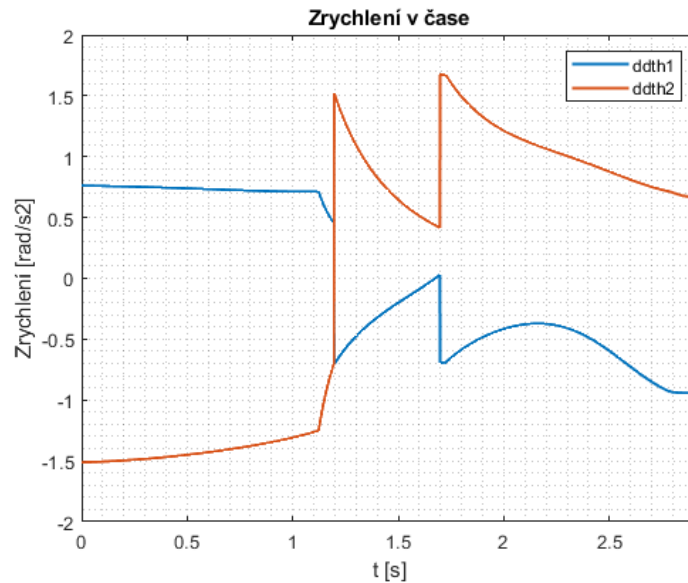
Hodnotami n_i, max jsou myšleny svrchní limity pohonů nastavené pro optimalizaci v definici oblasti (6.6). V našem případě $n_{1,max} = 100 Nm$ a $n_{2,max} = 70 Nm$.

Pro optimalizaci je nastavena pořadovaná rychlost $v_{set} = 2,5 m/s$. Pro ukončení výpočtu je zvoleno 5000 generací výpočtu. Optimalizované hodnoty parametrů vycházejí následující:

$$\begin{aligned}
 a_x &= 0,74 \text{ m} \\
 a_y &= 4,23 \text{ m} \\
 b_x &= -0,03 \text{ m} \\
 b_y &= 0,60 \text{ m} \\
 n_1 &= 98,01 \text{ Nm} \\
 n_2 &= 63,22 \text{ Nm}
 \end{aligned}
 \tag{6.9}$$



Obr. 6.7: Průběh momentů pro zadanou záchyťovou rychlost



Obr. 6.8: Průběh zrychlení pro zadanou záchyťovou rychlost

Pro tyto parametry bylo dosaženo hodnoty $v_{konst} = 2,5000 \text{ m/s}$. V daných desetinných místech bylo přesně dosaženo nastavené rychlosti, výsledek optimalizace se

tedy dá označit za uspokojivý. Pokud bychom se spokojili s větší odchylkou od nastavené rychlosti a chtěli bychom pohyb dosáhnout za použití nižších momentů v pohonech, bylo by možné upravit hodnoty váhových koeficientů k_v a k_n . Na Obr. 6.7 je zobrazen průběh momentů pro daný profil. Opět je patrné, že se jedná o pohyb na limitu kroutících momentů pohonů a ani v tomto případě nehrají roli zvolená omezení úhlových zrychlení. Na Obr. 6.8 je zobrazen průběh zrychlení.

6.4.2 Optimalizace rozměrů robotu

V posledním případě jsou optimalizovány rozměry manipulátoru pro dané pohony. Pohony budou použity shodné jako v předchozích případech, aby bylo možné zhodnotit rozdíl v původních a optimalizovaných parametrech. Ve vektoru optimalizovaných parametrů jsou v tomto případě nahrazeny momenty v pohonech délkami ramen manipulátoru:

$$P_{dyn,2} = [a_x, a_y, b_x, b_y, l_1, l_2]^T \quad (6.10)$$

Řešení je hledáno na této oblasti:

$$\begin{aligned} 0,01 &\leq a_x \leq l_1 + l_2 [m] \\ 0,01 &\leq a_y \leq l_1 + l_2 [m] \\ -(l_1 + l_2) &\leq b_x \leq -0,01 [m] \\ 0,01 &\leq b_y \leq l_1 + l_2 [m] \\ 0 &\leq l_1 \leq 5 [m] \\ 0 &\leq l_2 \leq 5 [m] \end{aligned} \quad (6.11)$$

Kriteriální funkce je shodná jako v případě optimalizace polohy zachytové úsečky:

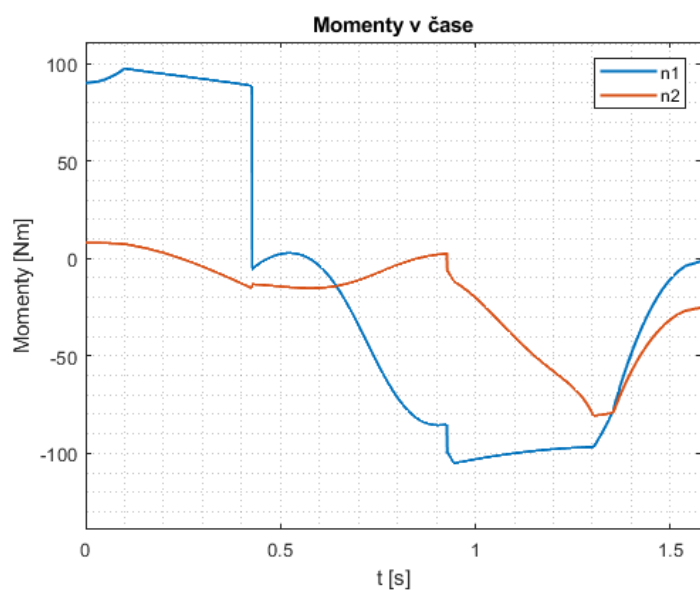
$$CF_{dyn,2} = -\dot{p}_{konst} \cdot |AB| = -v_{konst} \quad (6.12)$$

Pro optimalizaci bylo opět zvoleno 5000 výpočtů, toto jsou výsledné hodnoty parametrů:

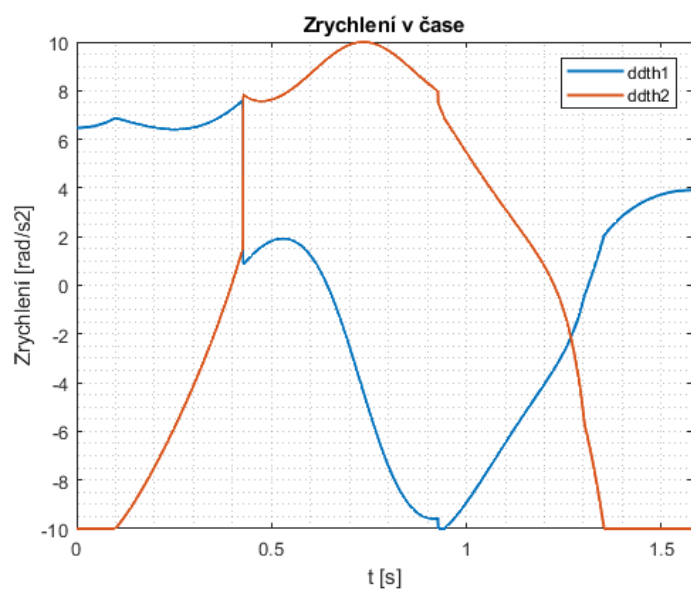
$$\begin{aligned} a_x &= 2,39 \text{ m} \\ a_y &= 0,11 \text{ m} \\ b_x &= -1,21 \text{ m} \\ b_y &= 2,25 \text{ m} \\ l_1 &= 1,37 \text{ m} \\ l_2 &= 1,30 \text{ m} \end{aligned} \quad (6.13)$$

Je dosaženo zachytitelné rychlosti předmětu $3,91 \text{ m/s}$, což je výrazné zlepšení oproti původním délkám ramen, pro které byla po optimalizaci trajektorie dosažena rychlost $3,04 \text{ m/s}$. Na Obr. 6.9 a 6.10 jsou vidět průběhy momentů a zrychlení pohonů.

Z průběhů je patrné, že již hrají značnou roli i omezení úhlových zrychlení manipulatorů. To je způsobeno zvolenými menšími rozměry ramen, což vede k menšímu namáhání pohonů, při daných hodnotách momentů jsme tak schopni dosahovat větších zrychlení.



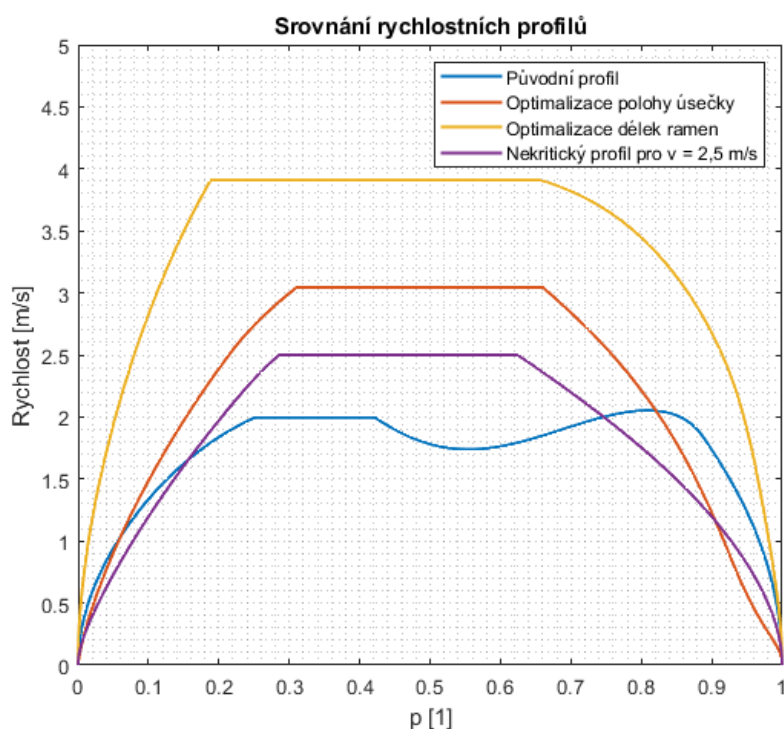
Obr. 6.9: Průběh momentů po optimalizaci rozměrů



Obr. 6.10: Průběh zrychlení po optimalizaci rozměrů

6.5 Zhodnocení

Všechny rychlostní profily jsou pro porovnání zobrazeny ve společném grafu na Obr. 6.11. Je možné vidět, že optimalizací polohy záchytové úsečky (oranžový profil) došlo k výraznému zvětšení zachytitelné rychlosti tělesa oproti původním parametrům (modrý profil), konkrétně zachytitelná rychlost vzrostla na 152% původní hodnoty. Pro dané parametry manipulátoru je tedy použita optimalizace vhodným nástrojem pro maximalizaci potenciálu stroje. Pokud je stroj v návrhové fázi, je vhodné využít optimalizaci i v případě volby rozměrů jeho ramen, jak je patrné z maximálního průběhu zobrazeného žlutou barvou. Fialová křivka ukazuje možnosti plánování parametrizace pro zachycení předmětu letícího rychlostí nižší, než je maximální zachytitelná, v tomto případě byla nastavena požadovaná rychlost zachyceného tělesa rovna $2,5 \text{ m/s}$, té se pomocí optimalizace podařilo dosáhnout.



Obr. 6.11: Porovnání rychlostních profilů

Optimalizace spočívá v hledání výhodné kombinace parametrů křivky a manipulátoru, větší rozměry manipulátoru umožňují delší záchytovou křivku a tedy i větší prostor pro zrychlení a dosažení maximální záchytové rychlosti, zároveň však s většími rozměry roste i hmotnost a momenty setrvačnosti ramen, což vyžaduje vyvinutí větších momentů v pohonech pro dosažení stejných úhlových zrychlení. Jak bylo vidět v případě optimalizace rozměrů manipulátoru, u menších délek ramen mohou z důvodu menších potřebných momentů v pohonech hrát větší roli zvolená omezení maximálního úhlového zrychlení v pohonech manipulátoru. Pro výpočet byla zvolena hustota materiálu odpovídající oceli a byl zvolen fixní poměr průměru ramen vůči jejich délkám. Pro skutečnou aplikaci výpočtu by bylo nutné vycházet z materiálů vhodných pro vesmírnou aplikaci a rozměry ramen volit v závislosti na

namáhání manipulátoru. Namáhání je závislé také na hmotnosti tělesa, které by měl zvolený manipulátor být schopný zachytit. Zvolená hmotnost tělesa značně ovlivňuje i zachytitelnou rychlost tělesa, v této práci byl výpočet proveden pro hodnotu 5 kg.

Byla provedena ukázková optimalizace pro sadu zvolených omezení manipulátoru. V optimalizaci jsou nastíněny možné příklady použití vytvořeného algoritmu a je ověřena jeho funkčnost a aplikovatelnost na tyto úlohy. Z porovnaných výsledků je zřejmý přínos optimalizace pro maximalizaci schopností vybraného manipulátoru při řešení úlohy zachytávání předmětů ve vesmíru. Zde jsou ještě pro přehlednost uvedeny dosažené hodnoty zachytitelných rychlostí:

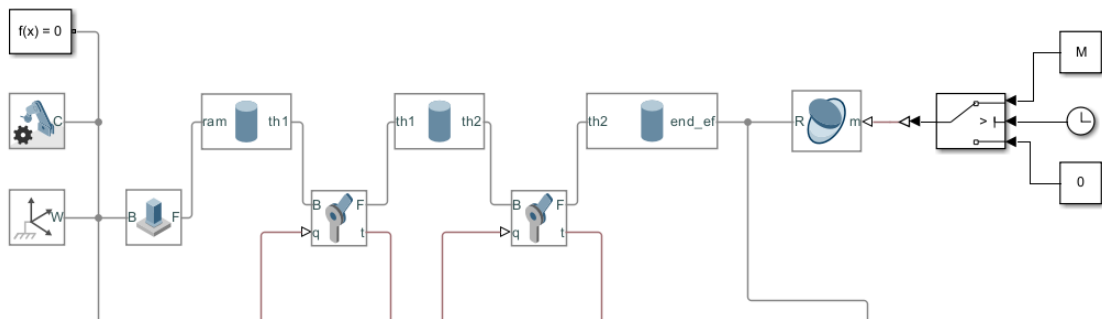
Původní parametry:	$v = 1,996 \text{ m/s}$
Optimalizace úsečky:	$v = 3,04 \text{ m/s}$
Optimalizace rozměrů robotu	$v = 3,91 \text{ m/s}$

Kapitola 7

Ověření výpočtu

Na závěr je vhodné provést kontrolu výsledků výpočetního algoritmu. Ta je provedena pomocí knihovny Simscape v programu Matlab-Simulink. Tato knihovna umožňuje modelování a simulace s fyzikálními systémy, je tedy možné provést simulaci pro zjednodušený RR manipulátor složený ze dvou válcových těles.

Model manipulátoru sestavený ze dvou rotačních vazeb a dvou válcových těles (třetí válcové těleso v modelu představuje rám) je zobrazen na Obr. 7.1.



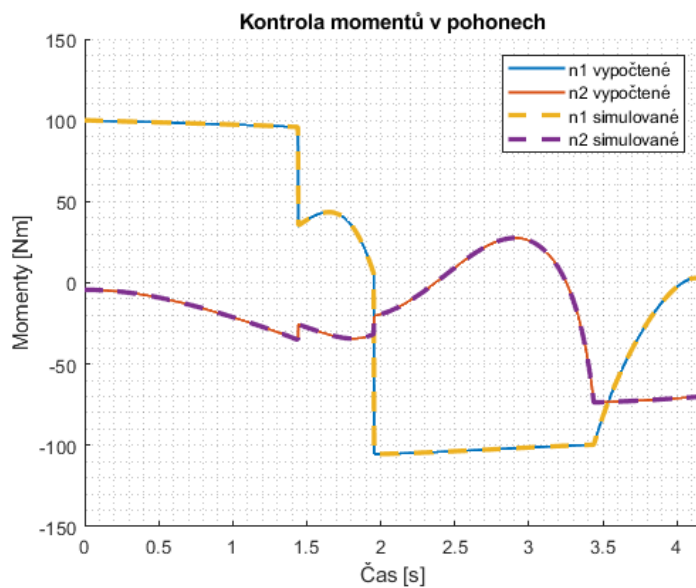
Obr. 7.1: Model manipulátoru

Rotační vazby umožňují předepisování a čtení vybraných veličin. Předepíšeme tedy vazbám vypočtené průběhy natočení pohonů a budeme odečítat momenty, které jsou potřeba pro vykonání daného pohybu. Při uvažování ideálních pohonů, nulových vnějších sil a při nastavení nulové gravitace v modelu by měly odečítané průběhy momentů odpovídat průběhům vypočteným.

Do portů q jsou přiváděny vypočtené údaje $\theta(t)$, $\dot{\theta}(t)$, $\ddot{\theta}(t)$. K end-effektoru je dále připojen hmotný bod, jehož hodnota se v čase zahájení konstantní části profilu změní z nulové hodnoty na hmotnost zachytávaného předmětu. Z portů t jsou následně odečítány simulované hodnoty momentů. Pro původní neoptimalizované parametry jsou zobrazeny na Obr. 7.2.

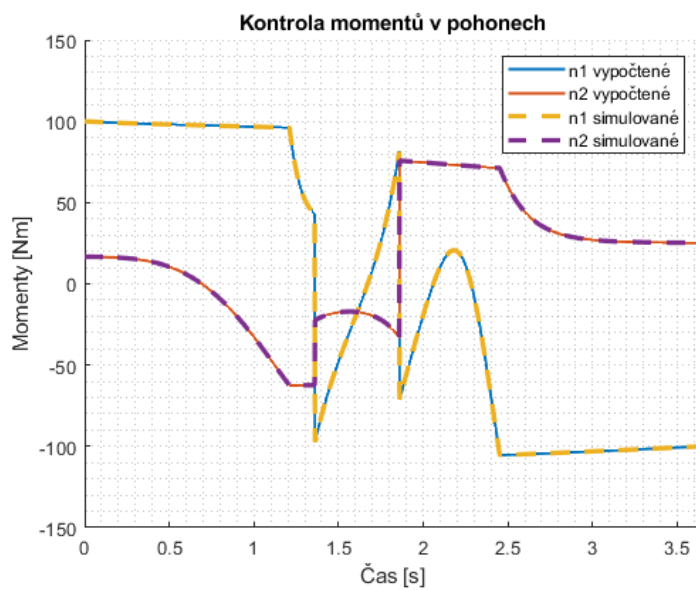
Je možné vidět, že se vypočtené a simulované průběhy shodují, lze tedy výpočet

považovat za správný, je ale třeba brát v úvahu zjednodušený model manipulátoru, předpoklad ideálních pohonů a zanedbání vnějších poruchových členů.



Obr. 7.2: Kontrola momentů

Stejně ověření je provedeno i pro optimalizovanou polohu záchytné úsečky, průběhy jsou zobrazeny na Obr. 7.3.



Obr. 7.3: Kontrola momentů po optimalizaci polohy úsečky

Kapitola 8

Závěr

Cílem práce bylo řešení dynamických možností robotu zachytit letící předmět ve vesmíru. V úvodu Kapitoly 3 shrnující stav problematiky byla formulována úloha zachycení předmětu jako pohyb end-effectoru po zadané křivce, který se skládá ze zrychlení na rychlost zachytávaného předmětu, provedení zachycení end-effectorem robotu a zpomalení do koncového bodu křivky. V kapitole byly následně popsány metody, které slouží k analýze dynamických schopností manipulátoru. Pro další řešení parametrizace pohybu byla zvolena metoda hledání rychlostního profilu v $p - \dot{p}$ diagramu.

Úloha řešení možného zpomalení robotem na trajektorii v pracovním prostoru byla formulována v Kapitole 4 jako hledání rychlostního profilu pro zachycení a zpomalení tělesa s nejvyšším realizovatelným průběhem rychlosti. To spočívá v určení profilu, který je složen z časově optimálního zrychlení na rychlost zachytávaného tělesa, provedení zachycení při konstantní rychlosti end-effectoru robotu rovné rychlosti zachytávaného tělesa a časově optimálního zpomalení do nulové rychlosti. Celý pohyb je vykonán na trajektorii v pracovním prostoru manipulátoru.

Byly stanoveny předpoklady pro řešení úlohy. Je uvažována možnost úpravy polohy robotu vůči trajektorii letícího předmětu. Toho by bylo možné dosáhnout pomocí změny polohy plavidla, na kterém je manipulátor upevněn, nebo umístěním manipulátoru na říditelný mechanismus. Dále je uvažováno, že hmotnost plavidla, ke kterému je robot upevněn, je dostatečně veliká a nedochází tak k ovlivnění jeho polohy pohybem robotu nebo setrvačností zachytávaného tělesa.

V Kapitole 4 byl pro řešení úlohy zvolen RR manipulátor především z důvodu výhodných analytických výpočtů. Jako záchyťová křivka end-effectoru robotu byla zvolena úsečka, která náleží uvažované přímce, po níž se pohybuje zachytávané těleso. Tato volba byla provedena především z důvodu časově méně náročných výpočtů a výhodné optimalizace trajektorie. Dále byly v Kapitole 4 pro RR manipulátor vyjádřeny veškeré vztahy potřebné pro zhodnocení jeho dynamických schopností na zvolené trajektorii. K dynamickým omezením byla přidána možnost stanovení kinematických omezení pohonů manipulátoru.

Tyto vztahy byly následně v Kapitole 5 aplikovány ve vývoji algoritmu v prostředí Matlab. Algoritmus pro vstupy v podobě parametrů manipulátoru, polohy záchyťové

úsečky, hmotnosti zachytávaného předmětu a času end-effectoru počítá rychlostní profil s maximální zachytitelnou rychlostí letícího tělesa. Jsou uvažovány pouze průběhy, kdy se jedná o bang-bang řízení s jedním přepínacím bodem, který je následně nahrazen částí profilu s konstantní rychlostí pro umožnění zachycení tělesa. Pro zvolené parametry byl proveden výpočet dosažitelného zachycení.

Algoritmus byl převeden do podoby funkce. Ta byla v Kapitole 6 využita pro 3 úlohy optimalizace. Jedná se o optimalizaci polohy křivky s cílem maximální rychlosti zachytávaného předmětu, hledání rychlostního profilu pro dosažení stanovené rychlosti tělesa, která je menší než maximální zachytitelná rychlost, a optimalizace rozměrů manipulátoru pro zadané pohony. Výsledky optimalizace byly porovnány s průběhem pro náhodně zvolené parametry a pro vypočtené profily bylo provedeno zhodnocení možnosti záchytu předmětu.

Pro ověření správnosti výpočtu byl v Kapitole 7 vytvořen model v prostředí Matlab-Simulink. Vypočtené hodnoty průběhů natočení pohonů byly použity jako vstupy pro rotační vazby v modelu a byly odečteny průběhy momentů potřebných pro realizaci tohoto pohybu. Hodnoty byly shodné s průběhy dříve vypočtenými při návrhu rychlostního profilu, výpočetní algoritmus se tedy dá považovat za správný.

Vytvořený algoritmus je nástrojem, který by bylo možné použít při předběžné analýze schopností robotu zachytit těleso ve vesmíru. Výpočet je navržen pro zjednodušený model RR manipulátoru. Pro RR manipulátor se jedná o nástroj, který poskytuje ucelenou představu o možnostech zachycení tělesa a umožňuje optimalizaci trajektorie i parametrů manipulátoru.

Na práci je možné navázat v několika oblastech. Do výpočtu by bylo možné zahrnout ovlivnění polohy plavidla v důsledku pohybu robotu a setrvačnosti zachytávaného předmětu. Dále je možné rozšíření výpočtu o složitější záchytové křivky a výpočty pro víceosé struktury manipulátorů. K nalezení příznivějších výsledků by také mohlo přispět zahrnutí průběhů s více přepínacími body v bang-bang řízení.

Seznam použitých zdrojů

- [1] About Canadarm. *Government of Canada* [online]. Ottawa, 2021 [cit. 2023-08-04]. Dostupné z: <https://www.asc-csa.gc.ca/eng/canadarm/about.asp>
- [2] VALÁŠEK, M. a Z ŠIKA. Evaluation of dynamic capabilities of machines and robots. *Multibody System Dynamics*. Springer, 2001, **6**, 183-202.
- [3] VALÁŠEK, M. Dynamic Time Parametrization of Manipulator Trajectories. *Kybernetika*. Institute of Information Theory and Automation AS CR, 1987, **23**(2), 154-174.
- [4] STEJSKAL, V. a M. VALÁŠEK. *Kinematics and dynamics of machinery*. New York: Marcel Dekker, 1996. ISBN 0-8247-9731-0.
- [5] LYNCH, K. M. a F. C. PARK. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge: Cambridge University Press, 2017. ISBN 9781107156302.
- [6] FERRENTINO, E; A. DELLA CIOPPA, A. MARCELLI a P. CHIACCHIO. An evolutionary approach to time-optimal control of robotic manipulators. *Journal of Intelligent & Robotic Systems*. Springer, 2020, **99**(2), 245-260. Dostupné z: <https://doi.org/10.1007/s10846-019-01116-9>
- [7] PHAM, Q-C. A General, Fast, and Robust Implementation of the Time-Optimal Path Parametrization Algorithm. *IEEE Transactions on Robotics*. 2014, **30**(6), 1533-1540. Dostupné z: <https://doi.org/10.1109/TRO.2014.2351113>
- [8] SHIN, K. a N. MCKAY. Selection of near-minimum time geometric paths for robotic manipulators. *IEEE Transactions on Automatic Control*. IEEE, 1986, **31**(6), 501-511. Dostupné z: <https://doi.org/10.1109/TAC.1986.1104316>
- [9] VERSCHEURE, D; B. DEMEULENAERE, J. SWEVER, J. DE SCHUTTER a M. DIEHL. Time-Optimal Path Tracking for Robots: A Convex Optimization Approach. *IEEE Transactions on Automatic Control*. 2009, **54**(10), 2318-2327. Dostupné z: <https://doi.org/10.1109/TAC.2009.2028959>
- [10] BOBROW, J. E; S. DUBOWSKY a J. S. GIBSON. Time-optimal control of robotic manipulators along specified paths. *The international journal of robotics research*. Thousand Oaks, CA: Sage Publications Sage CA, 1985, **4**(3), 3-17. Dostupné z: <https://doi.org/10.1177/027836498500400301>

- [11] SLOTINE, J-J; a HYUN S. Y. Improving the efficiency of time-optimal path-following algorithms. *1988 American Control Conference*. IEEE, 1988, 2129-2134. Dostupné z: <https://doi.org/10.23919/ACC.1988.4790076>
- [12] PHAM, H. a Q.-C. PHAM. A new approach to time-optimal path parameterization based on reachability analysis. *IEEE Transactions on Robotics*. IEEE, 2018, **34**(3), 645-659. Dostupné z: <https://doi.org/10.1109/TRO.2018.2819195>
- [13] HORNYCH, T. *Časově optimální řízení robotů s uvažováním dynamiky pohonů*. Praha, 2011. Disertační práce. Fakulta strojní ČVUT v Praze. Vedoucí práce Michael Valášek.
- [14] HOLLERBACH, J. M. Dynamic Scaling of Manipulator Trajectories. *1983 American Control Conference*. San Francisco, CA, 1983, 752-756. Dostupné z: <https://doi.org/10.23919/ACC.1983.4788214>
- [15] SCIAVICCO, L. a B. SICILIANO. *Modeling and control of robot manipulators*. McGraw-Hill, 1996. ISBN 0-07-057217-8.
- [16] HUNTER, I; J. M. HOLLERBACH a J. BALLANTYNE. A comparative analysis of actuator technologies for robotics. *Robotics Review*. MIT press, 1991, **2**, 299-342.
- [17] Euler's Method Explained with Examples. *FreeCodeCamp* [online]. 26.1. 2020 [cit. 2023-07-16]. Dostupné z: <https://www.freecodecamp.org/news/eulers-method-explained-with-examples/>
- [18] Surrogate Optimization Algorithm. *MathWorks* [online]. [cit. 2023-07-16]. Dostupné z: <https://www.mathworks.com/help/gads/surrogate-optimization-algorithm.html>
- [19] How to find intersection between a straight line and a curve? *MathWorks* [online]. [cit. 2023-08-04]. Dostupné z: <https://www.mathworks.com/matlabcentral/answers/1643405-how-to-find-intersection-between-a-straight-line-and-a-curve>

Seznam obrázků

3.1	Manipulátor na vesmírné stanici [1]	3
3.2	Realizovatelná oblast pohonu [3]	6
3.3	Realizovatelná oblast manipulátoru [3]	6
3.4	Průběh parametrizace křivky [5]	7
3.5	Realizovatelný kužel v $s - \dot{s}$ diagramu [5]	9
3.6	Časově optimální profil v $s - \dot{s}$ diagramu [5]	10
3.7	Hledání přepínacích bodů [6]	11
4.1	Pracovní prostor RR manipulátoru	13
4.2	Volba záchytové křivky	15
4.3	I. konfigurace RR robotu	16
4.4	II. konfigurace RR robotu	16
4.5	Rozměry potřebné pro rovnice dynamiky	21
4.6	Hodnoty K_m bezkartáčových stejnosměrných motorů [16]	24
5.1	Eulerova metoda [17]	33
5.2	Výsledky numerické integrace	35
5.3	Hledání průsečíků profilu	36
5.4	Vypočtený rychlostní profil	37
5.5	Profil a limitní křivka rychlosti	38
5.6	Časový průběh podél křivky	40
5.7	Časový průběh úhlových rychlostí	41
5.8	Časový průběh úhlových zrychlení	41
5.9	Časový průběh momentů	42
5.10	Časový průběh úhlových zrychlení pro $\ddot{\theta}_i^+ = 1 \text{ rad/s}^2$	43
5.11	Časový průběh momentů pro $\ddot{\theta}_i^+ = 1 \text{ rad/s}^2$	43

6.1	Fáze vytvoření náhradní funkce [18]	46
6.2	Fáze hledání minima [18]	46
6.3	Oblasti hledaných bodů úsečky	47
6.4	Optimalizace kinematiky	48
6.5	Momenty po optimalizaci úsečky	49
6.6	Zrychlení po optimalizaci úsečky	49
6.7	Průběh momentů pro zadanou záchytovou rychlost	51
6.8	Průběh zrychlení pro zadanou záchytovou rychlost	51
6.9	Průběh momentů po optimalizaci rozměrů	53
6.10	Průběh zrychlení po optimalizaci rozměrů	53
6.11	Porovnání rychlostních profilů	54
7.1	Model manipulátoru	56
7.2	Kontrola momentů	57
7.3	Kontrola momentů po optimalizaci polohy úsečky	57

Příloha

Výpočet inverzní kinematiky

```
1 function [x,y,th1,th2] = inverzni_kinematika(p,ax,ay,bx,by,l1,l2)
2
3     ux = (bx-ax);
4     uy = (by-ay);
5     x = ax + p*ux;
6     y = ay + p*uy;
7
8     h=(x^2+y^2-l1^2-l2^2)/(2*l1*l2);
9     xarg2=h;
10    yarg2=-sqrt(1-h^2);
11    th2=atan2(yarg2,xarg2);
12    xarg1=(l1+l2*cos(th2))*x+l2*sin(th2)*y;
13    yarg1=(l1+l2*cos(th2))*y-l2*sin(th2)*x;
14    th1=mod(atan2(yarg1,xarg1),2*pi);
15 end
```

Výpočet parciálních derivací

```
1 function [dth1_dp,dth2_dp,ddth1_ddp,ddth2_ddp]= parcialni_derivace(
2     p,l1,l2,ax,ay,bx,by)
3
4     [x,y,~,th2] = inverzni_kinematika(p,ax,ay,bx,by,l1,l2);
5
6     ux = bx-ax;
7     uy = by-ay;
8
9     h=(x^2+y^2-l1^2-l2^2)/(2*l1*l2);
10    xarg2=h;
11    yarg2=-sqrt(1-h^2);
12    xarg1=(l1+l2*cos(th2))*x+l2*sin(th2)*y;
13    yarg1=(l1+l2*cos(th2))*y-l2*sin(th2)*x;
14
15    dxarg2_dh=1;
16    dh_dx=x/(l1*l2);
17    dx_dp=ux;
18    dh_dy=y/(l1*l2);
19    dy_dp=uy;
20    dyarg2_dh=h/sqrt(1-h^2);
```



```

20     ddh_ddx=1/(l1*l2);
21     ddh_ddy=1/(l1*l2);
22     ddx_ddp=0;
23     ddy_ddp=0;
24     ddxarg2_ddh=0;
25     ddyarg2_ddh=1/(1-h^2)^(3/2);
26     dh_dp=dh_dx*dx_dp+dh_dy*dy_dp;
27     ddh_ddp=ddh_ddx*dx_dp^2+dh_dx*ddx_ddp+ddh_ddy*dy_dp^2+dh_dy*
    ddy_ddp;
28
29     dxarg1_dth2=l2*(-x*sin(th2)+y*cos(th2));
30     dxarg1_dx=l1+l2*cos(th2);
31     dxarg1_dy=l2*sin(th2);
32     dyarg1_dth2=-l2*(y*sin(th2)+x*cos(th2));
33     dyarg1_dx=-l2*sin(th2);
34     dyarg1_dy=l1+l2*cos(th2);
35     ddxarg1_ddth2=l2*(-x*cos(th2)-y*sin(th2));
36     ddxarg1_ddx=0;
37     ddxarg1_ddy=0;
38     ddyarg1_ddth2=-l2*(y*cos(th2)-x*sin(th2));
39     ddyarg1_ddx=0;
40     ddyarg1_ddy=0;
41     ddxarg1_dth2_dx=-l2*sin(th2);
42     ddxarg1_dth2_dy=l2*cos(th2);
43     ddyarg1_dth2_dx=-l2*cos(th2);
44     ddyarg1_dth2_dy=-l2*sin(th2);
45
46     %% Parcialni derivace th2
47
48     dxarg2_dp=dxarg2_dh*dh_dp;
49     dyarg2_dp=dyarg2_dh*dh_dp;
50     ddxarg2_ddp=ddxarg2_ddh*dh_dp^2+dxarg2_dh*ddh_ddp;
51     ddyarg2_ddp=ddyarg2_ddh*dh_dp^2+dyarg2_dh*ddh_ddp;
52
53     dth2_dp = -yarg2/(xarg2^2+yarg2^2)*dxarg2_dp+xarg2/(xarg2^2+
    yarg2^2)*dyarg2_dp;
54
55     d1=2*xarg2*yarg2/(xarg2^2+yarg2^2)^2;
56     d2=(yarg2^2-xarg2^2)/(xarg2^2+yarg2^2)^2;
57     d3=-(xarg2^2-yarg2^2)/(xarg2^2+yarg2^2)^2;
58     d4=-2*xarg2*yarg2/(xarg2^2+yarg2^2)^2;
59     ddth2_ddp=(d1*dxarg2_dp+d2*dyarg2_dp)*dxarg2_dp-yarg2/(xarg2^2+
    yarg2^2)*ddxarg2_ddp+(d3*dxarg2_dp+d4*dyarg2_dp)*dyarg2_dp+xarg2
    /(xarg2^2+yarg2^2)*ddyarg2_ddp;
60
61     %% Parcialni derivace th1
62
63     dxarg1_dp=dxarg1_dth2*dth2_dp+dxarg1_dx*dx_dp+dxarg1_dy*dy_dp;
64     dyarg1_dp=dyarg1_dth2*dth2_dp+dyarg1_dx*dx_dp+dyarg1_dy*dy_dp;
65     ddxarg1_ddp=(ddxarg1_ddth2*dth2_dp+ddxarg1_dth2_dx*dx_dp+
    ddxarg1_dth2_dy*dy_dp)*dth2_dp+dxarg1_dth2*ddth2_ddp+(
    ddxarg1_ddx*dx_dp+ddxarg1_dth2_dx*dth2_dp)*dx_dp+dxarg1_dx*
    ddx_ddp+(ddxarg1_ddy*dy_dp+ddxarg1_dth2_dy*dth2_dp)*dy_dp+
    dxarg1_dy*ddy_ddp;
66     ddyarg1_ddp=(ddyarg1_ddth2*dth2_dp+ddyarg1_dth2_dx*dx_dp+
    ddyarg1_dth2_dy*dy_dp)*dth2_dp+dyarg1_dth2*ddth2_ddp+(

```

```

ddyarg1_ddx*dx_dp+ddyarg1_dth2_dx*dth2_dp)*dx_dp+dyarg1_dx*
ddx_ddp+(ddyarg1_ddy*dy_dp+ddyarg1_dth2_dy*dth2_dp)*dy_dp+
dyarg1_dy*ddy_ddp;
67
68   dth1_dp = -yarg1/(xarg1^2+yarg1^2)*dxarg1_dp+xarg1/(xarg1^2+
yarg1^2)*dyarg1_dp;
69
70   D1=2*xarg1*yarg1/(xarg1^2+yarg1^2)^2;
71   D2=(yarg1^2-xarg1^2)/(xarg1^2+yarg1^2)^2;
72   D3=-(xarg1^2-yarg1^2)/(xarg1^2+yarg1^2)^2;
73   D4=-2*xarg1*yarg1/(xarg1^2+yarg1^2)^2;
74
75   ddth1_ddp=(D1*dxarg1_dp+D2*dyarg1_dp)*dxarg1_dp-yarg1/(xarg1^2+
yarg1^2)*ddxarg1_ddp+(D3*dxarg1_dp+D4*dyarg1_dp)*dyarg1_dp+xarg1
/(xarg1^2+yarg1^2)*ddyarg1_ddp;
76 end

```

Určení limitů pohonů

```

1 function [L,U] = limity(p,dp,ax,ay,bx,by,m1,m2,t1,t2,l1,l2,I1T1,
I2T2,M,n1_min,n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min
,ddth2_max,F1,F2)
2
3   [~,~,~,th2] = inverzni_kinematika(p,ax,ay,bx,by,l1,l2);
4   [dth1_dp,dth2_dp,ddth1_ddp,ddth2_ddp] = parcialni_derivace(p,l1
,l2,ax,ay,bx,by);
5
6   I = [m1*t1^2+I1T1+(m2+M)*l1^2+m2*t2^2+M*l2^2+2*(m2*l1*t2+M*l1*
l2)*cos(th2)+I2T2   m2*t2^2+M*l2^2+(m2*l1*t2+M*l1*l2)*cos(th2)+
I2T2;
7       m2*t2^2+M*l2^2+(m2*l1*t2+M*l1*l2)*cos(th2)+I2T2
       m2*t2^2+M*l2^2+I2T2];
8   C = [-(m2*l1*t2+M*l1*l2)*sin(th2);
9        (m2*l1*t2+M*l1*l2)*sin(th2)];
10
11   I1_1=I(1,1);
12   I2_1=I(1,2);
13   C_1=C(1,1);
14   I1_2=I(2,1);
15   I2_2=I(2,2);
16   C_2=C(2,1);
17
18   b_1 = I1_1*ddth1_ddp+I2_1*ddth2_ddp+C_1*(2*dth1_dp*dth2_dp+
dth2_dp^2);
19   a_1 = I1_1*dth1_dp+I2_1*dth2_dp;
20   c_1 = F1*dth1_dp;
21
22   b_2 = I1_2*ddth1_ddp+I2_2*ddth2_ddp+C_2*(dth1_dp^2);
23   a_2 = I1_2*dth1_dp+I2_2*dth2_dp;
24   c_2 = F2*dth2_dp;
25
26   if a_1 >= 0
27       delta1 = 1;
28   else

```

```

29         delta1 = 0;
30     end
31
32     if a_2 >= 0
33         delta2 = 1;
34     else
35         delta2 = 0;
36     end
37
38
39     L1 = (n1_min*delta1+n1_max*(1-delta1)-b_1*dp^2-c_1*dp)/a_1;
40     U1 = (n1_max*delta1+n1_min*(1-delta1)-b_1*dp^2-c_1*dp)/a_1;
41
42     L2 = (n2_min*delta2+n2_max*(1-delta2)-b_2*dp^2-c_2*dp)/a_2;
43     U2 = (n2_max*delta2+n2_min*(1-delta2)-b_2*dp^2-c_2*dp)/a_2;
44
45     if dth1_dp >= 0
46         delta_akc1 = 1;
47     else
48         delta_akc1 = 0;
49     end
50     if dth2_dp >= 0
51         delta_akc2 = 1;
52     else
53         delta_akc2 = 0;
54     end
55
56     L_akc1=(ddth1_min*delta_akc1+ddth1_max*(1-delta_akc1)-ddth1_ddp
57 *dp^2)/dth1_dp;
58     U_akc1=(ddth1_max*delta_akc1+ddth1_min*(1-delta_akc1)-ddth1_ddp
59 *dp^2)/dth1_dp;
60
61     L_akc2=(ddth2_min*delta_akc2+ddth2_max*(1-delta_akc2)-ddth2_ddp
62 *dp^2)/dth2_dp;
63     U_akc2=(ddth2_max*delta_akc2+ddth2_min*(1-delta_akc2)-ddth2_ddp
64 *dp^2)/dth2_dp;
65
66     L = max([L1;L2;L_akc1;L_akc2]);
67     U = min([U1;U2;U_akc1;U_akc2]);
68 end

```

Určení bodů limitní křivky rychlosti

```

1 function dp_mvc = dp_mvc(p,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,
2 M,n1_min,n1_max,n2_min,n2_max,dth1_max,dth2_max,ddth1_min,
3 ddth1_max,ddth2_min,ddth2_max,F1,F2)
4
5     %% Limit od momentu pohonu
6     dp = 0;
7     [L,U] = limity(p,dp,ax,ay,bx,by,m1,m2,t1,t2,l1,l2,I1T1,I2T2,M,
8 n1_min,n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min,
9 ddth2_max,F1,F2);
10
11     if U > L

```

```

8     while U > L
9         dp=dp+0.001; % volba presnosti
10        [L,U] = limity(p,dp,ax,ay,bx,by,m1,m2,t1,t2,l1,l2,I1T1,
I2T2,M,n1_min,n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min
,ddth2_max,F1,F2);
11        end
12        dp_max_mom = dp;
13    else
14        dp_max_mom = 0;
15    end
16
17    %% Limit od rychlosti pohonu
18    [dth1_dp,dth2_dp,~,~] = parcialni_derivace(p,l1,l2,ax,ay,bx,by)
;
19
20    dth1_dp = abs(dth1_dp);
21    dth2_dp = abs(dth2_dp);
22
23    dp1_dt = dth1_max/dth1_dp;
24    dp2_dt = dth2_max/dth2_dp;
25
26    dp_max_rychl = min([dp1_dt dp2_dt]);
27
28    %% Porovnani dvou limitu (vybereme mensi)
29    dp_mvc = min([dp_max_mom dp_max_rychl]);
30 end

```

Funkce dopředné integrace

```

1 function [p_vekt_do,dp_vekt_do,U_vekt] = dopredna_integrace_U(
p_start,dp_start,p_end,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,M
,n1_min,n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min,
ddth2_max,F1,F2)
2
3     p_vekt_do = [];
4     dp_vekt_do = [];
5     U_vekt = [];
6     p_vekt_do(1,1) = p_start;
7     dp_vekt_do(1,1) = dp_start;
8     count = 1;
9     p = p_start;
10    dp = dp_start;
11
12    while (p < p_end) && (dp >= 0) && (count < 10^9)
13        [~,U] = limity(p,dp,ax,ay,bx,by,m1,m2,t1,t2,l1,l2,I1T1,I2T2
,M,n1_min,n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min,
ddth2_max,F1,F2);
14        U_vekt(1,count) = U;
15        alfa = atan(U/dp);
16        delta = 0.001;
17        delta_p = delta*cos(alfa);
18        delta_dp = delta*sin(alfa);
19        p = p + delta_p;
20        dp = dp + delta_dp;

```

```

21     count = count + 1;
22     p_vekt_do(1,count) = p;
23     dp_vekt_do(1,count) = dp;
24     end
25     U_vekt(1,count) = U_vekt(1,count-1);
26 end

```

Funkce zpětné integrace

```

1 function [p_vekt_zp,dp_vekt_zp,L_vekt] = zpetna_integrace_L(p_start
,dp_start,p_end,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,M,n1_min
,n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min,ddth2_max,F1
,F2)
2     p_vekt_zp = [];
3     dp_vekt_zp = [];
4     L_vekt = [];
5     p_vekt_zp(1,1) = p_start;
6     dp_vekt_zp(1,1) = dp_start;
7     count = 1;
8     p = p_start;
9     dp = dp_start;
10
11     while (p > p_end) && (dp >= 0) && (count < 10^9)
12         [L,~] = limity(p,dp,ax,ay,bx,by,m1,m2,t1,t2,l1,l2,I1T1,I2T2
,M,n1_min,n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min,
ddth2_max,F1,F2);
13         L_vekt(1,count) = L;
14         alfa = atan(L/dp);
15         delta = 0.001;
16         delta_p = delta*cos(alfa);
17         delta_dp = delta*sin(alfa);
18         p = p - delta_p;
19         dp = dp - delta_dp;
20         count = count + 1;
21         p_vekt_zp(1,count) = p;
22         dp_vekt_zp(1,count) = dp;
23     end
24     L_vekt(1,count)=L_vekt(1,count-1);
25 end

```

Funkce pro hledání průsečíků

Funkce inspirována příspěvkem na Matlab foru [19].

```

1 function pruseciky = intersect(p_vekt,profil,y_value)
2     if (length(p_vekt)==length(profil))
3         if (size(y_value)==1)
4             indexy = find(diff(sign(profil-y_value)));
5             if isempty(indexy) == 0
6                 for k = 1:numel(indexy)
7                     if indexy(k) == 1

```

```

8         index_range = [1 2];
9         pruseciky(k) = interp1(profil(index_range),
p_vekt(index_range), y_value);
10        elseif indexy(k) == length(p_vekt)
11            index_range = [length(p_vekt)-1 length(
p_vekt)];
12            pruseciky(k) = interp1(profil(index_range),
p_vekt(index_range), y_value);
13        else
14            index_range = [-1 1]+indexy(k);
15            pruseciky(k) = interp1(profil(index_range),
p_vekt(index_range), y_value);
16        end
17    end
18    pruseciky = rmmissing(pruseciky);
19 else
20    pruseciky = 0;
21 end
22 else
23    pruseciky = 0;
24 end
25 else
26    pruseciky = 0;
27 end
28 end

```

Ověření nulových dělení

```

1 function success = kontrola_nuly(profil,p_chyt,ax,ay,bx,by,m1,m2,t1
,t2,l1,l2,I1T1,I2T2,M,n1_min,n1_max,n2_min,n2_max,ddth1_min,
ddth1_max,ddth2_min,ddth2_max,F1,F2)
2
3     p_vekt = linspace(0,1,length(profil));
4
5
6     dth1_dp_vekt = zeros(1,length(profil));
7     dth2_dp_vekt = zeros(1,length(profil));
8
9     a1_vekt = zeros(1,length(profil));
10    a2_vekt = zeros(1,length(profil));
11
12    for i = 1:length(profil)
13        p = p_vekt(1,i);
14        if p < p_chyt
15            M_vyp = 0;
16        else
17            M_vyp = M;
18        end
19
20        [~,~,~,th2] = inverzni_kinematika(p,ax,ay,bx,by,l1,l2);
21
22        [dth1_dp,dth2_dp,~,~] = parcialni_derivace(p,l1,l2,ax,
ay,bx,by);
23        dth1_dp_vekt(1,i)=dth1_dp;

```

```

24         dth2_dp_vekt(1,i)=dth2_dp;
25
26         I = [m1*t1^2+I1T1+(m2+M_vyp)*l1^2+m2*t2^2+M_vyp*l2
^2+2*(m2*l1*t2+M_vyp*l1*l2)*cos(th2)+I2T2   m2*t2^2+M_vyp*l2^2+(
m2*l1*t2+M_vyp*l1*l2)*cos(th2)+I2T2;
27         m2*t2^2+M_vyp*l2^2+(m2*l1*t2+M_vyp*l1*l2)*cos(th2)
+I2T2           m2*t2^2+M_vyp*l2^2+I2T2];
28
29         I1_1=I(1,1);
30         I2_1=I(1,2);
31         I1_2=I(2,1);
32         I2_2=I(2,2);
33
34         a1_vekt(1,i) = I1_1*dth1_dp+I2_1*dth2_dp;
35         a2_vekt(1,i) = I1_2*dth1_dp+I2_2*dth2_dp;
36     end
37
38     a1_zer = intersect(p_vekt,a1_vekt,0);
39     a2_zer = intersect(p_vekt,a2_vekt,0);
40     dth1_zer = intersect(p_vekt,dth1_dp_vekt,0);
41     dth2_zer = intersect(p_vekt,dth2_dp_vekt,0);
42
43     a1_check = zeros(1,length(a1_zer));
44     a2_check = zeros(1,length(a2_zer));
45     dth1_check = zeros(1,length(dth1_zer));
46     dth2_check = zeros(1,length(dth2_zer));
47
48     for i = 1:length(a1_zer)
49         p = a1_zer(1,i);
50         dp = interp1(p_vekt,profil,p);
51         if p < p_chyt
52             M_1 = 0;
53         else
54             M_1 = M;
55         end
56         [dth1_dp,dth2_dp,ddth1_ddp,ddth2_ddp] = parcialni_derivace(
p,l1,l2,ax,ay,bx,by);
57
58         I = [m1*t1^2+I1T1+(m2+M_1)*l1^2+m2*t2^2+M_1*l2^2+2*(m2*l1*
t2+M_1*l1*l2)*cos(th2)+I2T2   m2*t2^2+M_1*l2^2+(m2*l1*t2+M_1*l1*
l2)*cos(th2)+I2T2;
59         m2*t2^2+M_1*l2^2+(m2*l1*t2+M_1*l1*l2)*cos(th2)+I2T2
m2*t2^2+M_1*l2^2+I2T2];
60         C = [-(m2*l1*t2+M_1*l1*l2)*sin(th2);
61             (m2*l1*t2+M_1*l1*l2)*sin(th2)];
62
63         I1_1=I(1,1);
64         I2_1=I(1,2);
65         C_1=C(1,1);
66
67         b1 = I1_1*ddth1_ddp+I2_1*ddth2_ddp+C_1*(2*dth1_dp*dth2_dp+
dth2_dp^2);
68         c1 = F1*dth1_dp;
69
70         torque = b1*dp^2+c1*dp;
71

```

```

72     if (torque >= n1_min) && (torque <= n1_max)
73         a1_check(1,i) = 1;
74     else
75         a1_check(1,i) = 0;
76     end
77 end
78
79 for i = 1:length(a2_zer)
80     p = a2_zer(1,i);
81     dp = interp1(p_vekt,profil,p);
82     if p < p_chyt
83         M_2 = 0;
84     else
85         M_2 = M;
86     end
87     [dth1_dp,dth2_dp,ddth1_ddp,ddth2_ddp] = parcialni_derivace(
p,l1,l2,ax,ay,bx,by);
88
89     I = [m1*t1^2+I1T1+(m2+M_2)*l1^2+m2*t2^2+M_2*l2^2+2*(m2*l1*
t2+M_2*l1*l2)*cos(th2)+I2T2 m2*t2^2+M_2*l2^2+(m2*l1*t2+M_2*l1*
l2)*cos(th2)+I2T2;
90         m2*t2^2+M_2*l2^2+(m2*l1*t2+M_2*l1*l2)*cos(th2)+I2T2
m2*t2^2+M_2*l2^2+I2T2];
91     C = [-(m2*l1*t2+M_2*l1*l2)*sin(th2);
92         (m2*l1*t2+M_2*l1*l2)*sin(th2)];
93
94     I12_2=I(2,1);
95     I23_2=I(2,2);
96     C_2=C(2,1);
97
98     b2 = I12_2*ddth1_ddp+I23_2*ddth2_ddp+C_2*(dth1_dp^2);
99     c2 = F2*dth2_dp;
100
101     torque = b2*dp^2+c2*dp;
102
103     if (torque >= n2_min) && (torque <= n2_max)
104         a2_check(1,i) = 1;
105     else
106         a2_check(1,i) = 0;
107     end
108 end
109
110 for i = 1:length(dth1_zer)
111     p = dth1_zer(1,i);
112     dp = interp1(p_vekt,profil,p);
113
114     [~,~,ddth1_ddp,~] = parcialni_derivace(p,l1,l2,ax,ay,bx,by)
;
115
116     acc = ddth1_ddp*dp^2;
117
118     if (acc >= ddth1_min) && (acc <= ddth1_max)
119         dth1_check(1,i) = 1;
120     else
121         dth1_check(1,i) = 0;
122     end

```



```

123     end
124
125     for i = 1:length(dth2_zer)
126         p = dth2_zer(1,i);
127         dp = interp1(p_vekt,profil,p);
128
129         [~,~,~,ddth2_ddp] = parcialni_derivace(p,l1,l2,ax,ay,bx,by)
130     ;
131
132     acc = ddth2_ddp*dp^2;
133
134     if (acc >= ddth2_min) && (acc <= ddth2_max)
135         dth2_check(1,i) = 1;
136     else
137         dth2_check(1,i) = 0;
138     end
139 end
140
141 a1_res = min(a1_check);
142 a2_res = min(a1_check);
143 dth1_res = min(dth1_check);
144 dth2_res = min(dth2_check);
145
146 success = min([a1_res a2_res dth1_res dth2_res]);
147 end

```

Funkce rychlostního profilu

```

1 function [p_vekt, dp_dt_vekt, v_max, mvc, ddp_ddt_vekt, th1_vekt,
2         th2_vekt, dth1_dp_vekt, dth2_dp_vekt, ddth1_ddp_vekt, ddth2_ddp_vekt
3         , konst_start, konst_end] = rychl_profil(t_ef, p_start, p_end,
4         n_points, ax, ay, bx, by, l1, l2, t1, t2, m1, m2, I1T1, I2T2, M, n1_min, n1_max
5         , n2_min, n2_max, dth1_max, dth2_max, ddth1_min, ddth1_max, ddth2_min,
6         ddth2_max, F1, F2)
7
8     p_vekt = linspace(p_start, p_end, n_points);
9     args = zeros(2, n_points);
10
11     %% Kontrola realizovatelnosti drahy (kvuli optimalizaci)
12     for i = 1:length(p_vekt)
13         p = p_vekt(1,i);
14         ux = (bx-ax);
15         uy = (by-ay);
16         x = ax + p*ux;
17         y = ay + p*uy;
18
19         h=(x^2+y^2-l1^2-l2^2)/(2*l1*l2);
20         xarg2=h;
21         yarg2=-sqrt(1-h^2);
22         args(1,i) = xarg2;
23         args(2,i) = yarg2;
24     end

```

```

21 pocitat1 = isreal(args); % vypocet je proveden pouze pokud jsou
    argumenty uhlu realne, jinak je vysledek nastaven na nepriznivy
22
23 if pocitat1 == 1
24
25     p_start_do = 0;
26     dp_start_do = 0;
27     p_end_do = 1;
28     p_start_zp = 1;
29     dp_start_zp = 0;
30     p_end_zp = 0;
31
32     [p_vekt_do, dp_vekt_do] = dopredna_integrace_U(p_start_do,
    dp_start_do, p_end_do, ax, ay, bx, by, l1, l2, t1, t2, m1, m2, I1T1, I2T2, 0,
    n1_min, n1_max, n2_min, n2_max, ddth1_min, ddth1_max, ddth2_min,
    ddth2_max, F1, F2);
33     [p_vekt_zp, dp_vekt_zp] = zpetna_integrace_L(p_start_zp,
    dp_start_zp, p_end_zp, ax, ay, bx, by, l1, l2, t1, t2, m1, m2, I1T1, I2T2, M,
    n1_min, n1_max, n2_min, n2_max, ddth1_min, ddth1_max, ddth2_min,
    ddth2_max, F1, F2);
34
35     if (p_vekt_do(1, end)) > p_end_do && (p_vekt_zp(1, end) <
    p_end_zp)
36
37         [p_vekt_zp, in1, ~] = unique(p_vekt_zp, 'first');
38         dp_vekt_zp = dp_vekt_zp(in1);
39
40         [p_vekt_do, in1, ~] = unique(p_vekt_do, 'first');
41         dp_vekt_do = dp_vekt_do(in1);
42
43         fin1 = min(isfinite(p_vekt_do));
44         fin2 = min(isfinite(dp_vekt_do));
45         fin3 = min(isfinite(p_vekt_zp));
46         fin4 = min(isfinite(dp_vekt_zp));
47
48         if (fin1 == 1) && (fin2 == 1) && (fin3 == 1) && (fin4
    == 1)
49             pocitat2 = 1;
50         else
51             pocitat2 = 0;
52         end
53     else
54         pocitat2 = 0;
55     end
56 else
57     pocitat2 = 0;
58 end
59
60 if pocitat1 == 1 && pocitat2 == 1
61     pocitat = 1;
62 else
63     pocitat = 0;
64 end
65
66 %% Po provedeni kontroly je proveden skutecny vypocet
67

```

```

68     if pocitat == 1
69
70         p_vekt = linspace(p_start, p_end, n_points);
71         draha = sqrt((bx-ax)^2+(by-ay)^2);
72
73         %% Inverzni kinematika (vse je pocitano pro hodnotu
parametru p)
74
75         x_vekt = zeros(1, length(p_vekt));
76         y_vekt = zeros(1, length(p_vekt));
77         th1_vekt = zeros(1, length(p_vekt));
78         th2_vekt = zeros(1, length(p_vekt));
79
80         for i = 1:length(p_vekt)
81             p = p_vekt(1,i);
82             [x,y,th1,th2] = inverzni_kinematika(p, ax, ay, bx, by, l1, l2
);
83
84             x_vekt(1,i) = x;
85             y_vekt(1,i) = y;
86             th1_vekt(1,i) = th1;
87             th2_vekt(1,i) = th2;
88         end
89
90         %% Parcialni derivace
91
92         dth1_dp_vekt = zeros(1, length(p_vekt));
93         dth2_dp_vekt = zeros(1, length(p_vekt));
94         ddth1_ddp_vekt = zeros(1, length(p_vekt));
95         ddth2_ddp_vekt = zeros(1, length(p_vekt));
96
97         for i = 1:length(p_vekt)
98             p = p_vekt(1,i);
99             [dth1_dp, dth2_dp, ddth1_ddp, ddth2_ddp] =
parcialni_derivace(p, l1, l2, ax, ay, bx, by);
100             dth1_dp_vekt(1,i) = dth1_dp;
101             dth2_dp_vekt(1,i) = dth2_dp;
102             ddth1_ddp_vekt(1,i) = ddth1_ddp;
103             ddth2_ddp_vekt(1,i) = ddth2_ddp;
104         end
105
106         %% Numericka integrace zrychleni a zpomaleni
107
108         p_start_do = 0;
109         dp_start_do = 0;
110         p_end_do = 1;
111         p_start_zp = 1;
112         dp_start_zp = 0;
113         p_end_zp = 0;
114
115         [p_vekt_do, dp_vekt_do, U_vekt] = dopredna_integrace_U(
p_start_do, dp_start_do, p_end_do, ax, ay, bx, by, l1, l2, t1, t2, m1, m2,
I1T1, I2T2, 0, n1_min, n1_max, n2_min, n2_max, ddth1_min, ddth1_max,
ddth2_min, ddth2_max, F1, F2);
116
117         [p_vekt_zp, dp_vekt_zp, L_vekt] = zpetna_integrace_L(
p_start_zp, dp_start_zp, p_end_zp, ax, ay, bx, by, l1, l2, t1, t2, m1, m2,
I1T1, I2T2, M, n1_min, n1_max, n2_min, n2_max, ddth1_min, ddth1_max,

```

```

ddth2_min , ddth2_max , F1 , F2 );
116
117     [p_vekt_zp , in1 , ~] = unique(p_vekt_zp , 'first');
118     dp_vekt_zp = dp_vekt_zp(in1);
119     L_vekt = L_vekt(in1);
120
121     [p_vekt_do , in1 , ~] = unique(p_vekt_do , 'first');
122     dp_vekt_do = dp_vekt_do(in1);
123     U_vekt = U_vekt(in1);
124
125     dp_vekt_do = interp1(p_vekt_do , dp_vekt_do , p_vekt);
126     U_vekt = interp1(p_vekt_do , U_vekt , p_vekt);
127
128     dp_vekt_zp = interp1(p_vekt_zp , dp_vekt_zp , p_vekt);
129     L_vekt = interp1(p_vekt_zp , L_vekt , p_vekt);
130
131     dp_dt_vekt = zeros(1 , length(p_vekt));
132     ddp_ddt_vekt = zeros(1 , length(p_vekt));
133
134     for i = 1:length(p_vekt)
135         if dp_vekt_do(1,i) < dp_vekt_zp(1,i)
136             dp_dt_vekt(1,i) = dp_vekt_do(1,i);
137             ddp_ddt_vekt(1,i) = U_vekt(1,i);
138         else
139             dp_dt_vekt(1,i) = dp_vekt_zp(1,i);
140             ddp_ddt_vekt(1,i) = L_vekt(1,i);
141         end
142     end
143
144     %% Vypocteni konstantni casti profilu
145     maximum = max(dp_dt_vekt);
146     delta = 0.001;
147     dp = maximum;
148     success = 0;
149
150     if dp > 0
151         cont = 1;
152     else
153         cont = 0;
154     end
155
156     while cont == 1
157         p_konst = dp*t_ef;
158         X = intersect(p_vekt , dp_dt_vekt , dp);
159         pocet_prus = length(X);
160
161         if pocet_prus == 2
162             p_pot = X(1,2)-X(1,1);
163             konst_start = X(1,1);
164             konst_end = X(1,2);
165         elseif pocet_prus == 4
166             p_pot1 = X(1,2)-X(1,1);
167             p_pot2 = X(1,4)-X(1,3);
168             if p_pot1 > p_pot2
169                 p_pot = p_pot1;
170                 konst_start = X(1,1);

```

```

171         konst_end = X(1,2);
172     else
173         p_pot = p_pot2;
174         konst_start = X(1,3);
175         konst_end = X(1,4);
176     end
177 else
178     p_pot = 0;
179     konst_start = 0;
180     konst_end = 0;
181 end
182
183 if p_pot > p_konst
184     p_kontr1 = konst_start - (p_vekt(1,2)-p_vekt(1,1));
185     p_kontr2 = konst_end + (p_vekt(1,2)-p_vekt(1,1));
186     p_kontr_vekt = p_vekt(p_vekt>p_kontr1);
187     p_kontr_vekt = p_kontr_vekt(p_kontr_vekt<p_kontr2);
188     L_vekt = zeros(1,length(p_kontr_vekt));
189     U_vekt = zeros(1,length(p_kontr_vekt));
190     for i = 1:length(p_kontr_vekt)
191         [L,U] = limity(p_kontr_vekt(1,i),dp,ax,ay,bx,by
, m1,m2,t1,t2,l1,l2,I1T1,I2T2,M,n1_min,n1_max,n2_min,n2_max,
ddth1_min,ddth1_max,ddth2_min,ddth2_max,F1,F2);
192         L_vekt(1,i) = L;
193         U_vekt(1,i) = U;
194     end
195
196     if (max(L_vekt)<=0)&&(min(U_vekt)>=0)
197
198         for i=1:length(p_vekt)
199             if (p_vekt(1,i) > konst_start) && (p_vekt
(1,i) < konst_end)
200                 dp_dt_vekt(1,i)=dp;
201                 ddp_ddt_vekt(1,i)=0;
202             else
203                 end
204             end
205
206             mvc = zeros(1,length(p_vekt));
207
208             for i = 1:length(p_vekt)
209                 p = p_vekt(1,i);
210                 if p < konst_start
211                     mvc(1,i) = dp_mvc(p,ax,ay,bx,by,l1,l2,
t1,t2,m1,m2,I1T1,I2T2,0,n1_min,n1_max,n2_min,n2_max,dth1_max,
dth2_max,ddth1_min,ddth1_max,ddth2_min,ddth2_max,F1,F2);
212                 else
213                     mvc(1,i) = dp_mvc(p,ax,ay,bx,by,l1,l2,
t1,t2,m1,m2,I1T1,I2T2,M,n1_min,n1_max,n2_min,n2_max,dth1_max,
dth2_max,ddth1_min,ddth1_max,ddth2_min,ddth2_max,F1,F2);
214                 end
215             end
216
217             delta = mvc - dp_dt_vekt;
218
219             if min(delta)>0

```

```

220             success = kontrola_nuly(dp_dt_vekt ,
konst_start , ax , ay , bx , by , m1 , m2 , t1 , t2 , l1 , l2 , I1T1 , I2T2 , M , n1_min ,
n1_max , n2_min , n2_max , ddth1_min , ddth1_max , ddth2_min , ddth2_max , F1 ,
F2);
221
222             if success == 1
223                 v_max = dp*draha;
224             else
225                 end
226
227             else
228                 success = 0;
229             end
230         else
231             success = 0;
232         end
233     else
234         success = 0;
235     end
236
237     dp = dp - delta;
238     if dp > 0
239         if success == 0
240             cont = 1;
241         else
242             cont = 0;
243         end
244     else
245         cont = 0;
246     end
247 end
248
249 if success == 0
250     v_max = 0;
251     dp_dt_vekt = zeros(1,n_points);
252     mvc = zeros(1,n_points);
253     ddp_ddt_vekt = zeros(1,n_points);
254     th1_vekt = zeros(1,n_points);
255     th2_vekt = zeros(1,n_points);
256     dth1_dp_vekt = zeros(1,n_points);
257     dth2_dp_vekt = zeros(1,n_points);
258     ddth1_ddp_vekt = zeros(1,n_points);
259     ddth2_ddp_vekt = zeros(1,n_points);
260     konst_start = 0;
261     konst_end = 0;
262 else
263     end
264 else
265     v_max = 0;
266     dp_dt_vekt = zeros(1,n_points);
267     mvc = zeros(1,n_points);
268     ddp_ddt_vekt = zeros(1,n_points);
269     th1_vekt = zeros(1,n_points);
270     th2_vekt = zeros(1,n_points);
271     dth1_dp_vekt = zeros(1,n_points);
272     dth2_dp_vekt = zeros(1,n_points);

```

```

273     ddth1_ddp_vekt = zeros(1,n_points);
274     ddth2_ddp_vekt = zeros(1,n_points);
275     konst_start = 0;
276     konst_end = 0;
277     end
278 end

```

Optimalizační funkce úsečky

```

1 function max_rychlost = opt_fce_usecka(X)
2
3     %% Parametry manipulatoru
4     l1 = 2.2;
5     l2 = 2.2;
6     ro = 7850;
7     n1 = 100;
8     n2 = 70;
9     dth1 = 7;
10    dth2 = 7;
11    ddth1 = 10;
12    ddth2 = 10;
13    M = 5;
14    t_ef = 0.5;
15    F1 = 4;
16    F2 = 4;
17
18    %% Odvozene parametry
19    t1 = l1/2;
20    t2 = l2/2;
21    d1 = l1/50;
22    d2 = l2/50;
23    m1 = pi*d1^2/4*l1*ro;
24    m2 = pi*d2^2/4*l2*ro;
25    I1T1 = m1*l1^2/12;
26    I2T2 = m2*l2^2/12;
27    n1_max = n1;
28    n1_min = -n1;
29    n2_max = n2;
30    n2_min = -n2;
31    dth1_max = dth1;
32    dth2_max = dth2;
33    ddth1_min = -ddth1;
34    ddth1_max = ddth1;
35    ddth2_min = -ddth2;
36    ddth2_max = ddth2;
37
38    %% Parametry krivky
39    p_start = 0;
40    p_end = 1;
41    n_points = 1000;
42
43    ax = X(1);
44    ay = X(2);
45    bx = X(3);

```

```

46     by = X(4);
47
48     %% Vypocet rychlostniho profilu
49     [~,~,v_max,~,~,~,~,~,~,~,~,~] = rychl_profil(t_ef,p_start,
p_end,n_points,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,M,n1_min,
n1_max,n2_min,n2_max,dth1_max,dth2_max,ddth1_min,ddth1_max,
ddth2_min,ddth2_max,F1,F2);
50     max_rychlost = -v_max;
51 end

```

Optimalizační funkce momentů

```

1 function F = opt_fce_pohony(x)
2
3     setRychlost = 2.5;
4
5     %% Nezávislé parametry
6
7     ro = 7850;
8     dth1 = 7;
9     dth2 = 7;
10    ddth1 = 10;
11    ddth2 = 10;
12    M = 5;
13    t_ef = 0.5;
14    F1 = 4;
15    F2 = 4;
16    p_start = 0;
17    p_end = 1;
18    n_points = 1000;
19    l1 = 2.2;
20    l2 = 2.2;
21
22    %% Optimalizované parametry
23
24    ax = x(1);
25    ay = x(2);
26    bx = x(3);
27    by = x(4);
28    n1 = x(5);
29    n2 = x(6);
30
31    %% Závislé parametry
32
33    t1=l1/2;
34    t2=l2/2;
35    d1 = l1/50;
36    d2 = l2/50;
37    m1 = pi*d1^2/4*l1*ro;
38    m2 = pi*d2^2/4*l2*ro;
39    I1T1 = m1*l1^2/12;
40    I2T2 = m2*l2^2/12;
41    n1_min = -n1;
42    n1_max = n1;

```



```

43     n2_min = -n2;
44     n2_max = n2;
45     dth1_max = dth1;
46     dth2_max = dth2;
47     ddth1_min = -ddth1;
48     ddth1_max = ddth1;
49     ddth2_min = -ddth2;
50     ddth2_max = ddth2;
51
52     %% Optimalizacni funkce
53
54     [~,~,v_max,~,~,~,~,~,~,~,~,~] = rychl_profil(t_ef,p_start,
55     p_end,n_points,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,M,n1_min,
56     n1_max,n2_min,n2_max,dth1_max,dth2_max,ddth1_min,ddth1_max,
57     ddth2_min,ddth2_max,F1,F2);
58
59     F = 100*abs(setRychlost-v_max)+(n1+n2)/170;
60 end

```

Optimalizační funkce délek ramen

```

1 function F = opt_fce_delky(x)
2
3     %% Nezávislé parametry
4
5     ro = 7850;
6     dth1 = 7;
7     dth2 = 7;
8     ddth1 = 10;
9     ddth2 = 10;
10    M = 5;
11    t_ef = 0.5;
12    F1 = 4;
13    F2 = 4;
14    p_start = 0;
15    p_end = 1;
16    n_points = 1000;
17    n1 = 100;
18    n2 = 70;
19
20    %% Optimalizované parametry
21
22    ax = x(1);
23    ay = x(2);
24    bx = x(3);
25    by = x(4);
26    l1 = x(5);
27    l2 = x(6);
28
29    %% Závislé parametry
30
31    t1=l1/2;
32    t2=l2/2;
33    d1 = l1/50;

```

```

34     d2 = l2/50;
35     m1 = pi*d1^2/4*l1*ro;
36     m2 = pi*d2^2/4*l2*ro;
37     I1T1 = m1*l1^2/12;
38     I2T2 = m2*l2^2/12;
39     n1_min = -n1;
40     n1_max = n1;
41     n2_min = -n2;
42     n2_max = n2;
43     dth1_max = dth1;
44     dth2_max = dth2;
45     ddth1_min = -ddth1;
46     ddth1_max = ddth1;
47     ddth2_min = -ddth2;
48     ddth2_max = ddth2;
49
50     %% Optimalizacni funkce
51
52     [~,~,v_max,~,~,~,~,~,~,~,~,~] = rychl_profil(t_ef,p_start,
p_end,n_points,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,M,n1_min,
n1_max,n2_min,n2_max,dth1_max,dth2_max,ddth1_min,ddth1_max,
ddth2_min,ddth2_max,F1,F2);
53
54     F = -v_max;
55 end

```

Spuštění optimalizací

Optimalizace trajektorie

```

1 clc; clear; close all;
2
3 objconstr = @opt_fce_usecka;
4 lb = [0.01;0.01;-4.4;0.01];
5 ub = [4.4;4.4;-0.01;4.4];
6
7 options = optimoptions('surrogateopt','PlotFcn','surrogateoptplot',
'MaxFunctionEvaluations',5000);
8 [x,fval,exitflag,output,trials] = surrogateopt(objconstr,lb,ub,
options);

```

Optimalizace momentů

```

1 clc; clear; close all;
2
3 rng default
4 lb = [0.01;0.01;-4.4;0.01;0;0];
5 ub = [4.4;4.4;-0.01;4.4;100;70];
6
7 options = optimoptions('surrogateopt','PlotFcn','surrogateoptplot',
'MaxFunctionEvaluations',5000);

```

```

8 objconstr = @opt_fce_pohony;
9 [x,fval,exitflag,output,trials] = surrogateopt(objconstr,lb,ub,
    options);

```

Optimalizace délek ramen

```

1 clc; clear; close all;
2
3 rng default % For reproducibility
4 lb = [0.01;0.01;-10;0.01;0;0];
5 ub = [10;10;-0.01;10;5;5];
6 A = [1 0 0 0 -1 -1;
7       0 1 0 0 -1 -1;
8       0 0 -1 0 -1 -1;
9       0 0 0 1 -1 -1];
10 b = [0;0;0;0];
11 intcon = [];
12 Aeq = [];
13 beq = [];
14 options = optimoptions('surrogateopt','PlotFcn','surrogateoptplot',
    'MaxFunctionEvaluations',5000);
15 objconstr = @opt_fce_delky;
16 [x,fval,exitflag,output,trials] = surrogateopt(objconstr,lb,ub,
    intcon,A,b,Aeq,beq,options);

```

Vykreslení průběhů

```

1 clc; clear; close all;
2 format long;
3
4 %% Nezávisle parametry
5 l1 = 2.2;
6 l2 = 2.2;
7 ro = 7850;
8 n1 = 100;
9 n2 = 70;
10 dth1 = 7;
11 dth2 = 7;
12 ddth1 = 10;
13 ddth2 = 10;
14 t_ef = 0.5;
15 M = 5;
16 F1 = 4;
17 F2 = 4;
18
19 %% Zavisle parametry
20 t1=l1/2;
21 t2=l2/2;
22 d1 = l1/50;
23 d2 = l2/50;
24 m1 = pi*d1^2/4*l1*ro;
25 m2 = pi*d2^2/4*l2*ro;

```

```

26 I1T1 = m1*l1^2/12;
27 I2T2 = m2*l2^2/12;
28 n1_min = -n1;
29 n1_max = n1;
30 n2_min = -n2;
31 n2_max = n2;
32 dth1_max = dth1;
33 dth2_max = dth2;
34 ddth1_max = ddth1;
35 ddth1_min = -ddth1;
36 ddth2_max = ddth2;
37 ddth2_min = -ddth2;
38
39 %% Definice trajektorie
40
41 ax = 3;
42 ay = 1.5;
43 bx = -3;
44 by = 1.5;
45 p_start = 0;
46 p_end = 1;
47 n_points = 1000;
48
49 draha = sqrt((bx-ax)^2+(by-ay)^2);
50 [x_init,y_init,th1_init,th2_init] = inverzni_kinematika(0,ax,ay,bx,
    by,l1,l2);
51
52 [p_vekt,dp_dt_vekt,v_max,mvc,ddp_ddt_vekt,th1_vekt,th2_vekt,
    dth1_dp_vekt,dth2_dp_vekt,ddth1_ddp_vekt,ddth2_ddp_vekt,hranice1
    ,hranice2] = rychl_profil(t_ef,p_start,p_end,n_points,ax,ay,bx,
    by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,M,n1_min,n1_max,n2_min,n2_max,
    dth1_max,dth2_max,ddth1_min,ddth1_max,ddth2_min,ddth2_max,F1,F2)
    ;
53
54 %% Casova zavislost
55
56 t_vekt = zeros(1,length(p_vekt));
57 t_vekt(1,1) = 0;
58 profil_scale = dp_dt_vekt*draha;
59
60 for i = 1:(length(p_vekt)-1)
61     vel = (dp_dt_vekt(1,i)+dp_dt_vekt(1,i+1))/2;
62     dist = (p_vekt(1,i+1)-p_vekt(1,i));
63     t = dist/vel;
64     t_vekt(1,i+1) = t_vekt(1,i)+t;
65 end
66
67 t_celk = t_vekt(end);
68 t_hmotnost = interp1(p_vekt,t_vekt,hranice1);
69
70 %% Dopocet dynamiky
71 I1_1_vekt = zeros(1,length(p_vekt));
72 I2_1_vekt = zeros(1,length(p_vekt));
73 C_1_vekt = zeros(1,length(p_vekt));
74 I1_2_vekt = zeros(1,length(p_vekt));
75 I2_2_vekt = zeros(1,length(p_vekt));

```

```

76 C_2_vekt = zeros(1,length(p_vekt));
77
78 for i = 1:length(p_vekt)
79     if p_vekt(1,i) < hranice1
80         M_post=0;
81     else
82         M_post=M;
83     end
84     I1_1_vekt(1,i) = m1*t1^2+I1T1+(m2+M_post)*l1^2+m2*t2^2+M_post*
85     l2^2+2*(m2*l1*t2+M_post*l1*l2)*cos(th2_vekt(1,i))+I2T2;
86     I2_1_vekt(1,i) = m2*t2^2+M_post*l2^2+(m2*l1*t2+M_post*l1*l2)*
87     cos(th2_vekt(1,i))+I2T2;
88     C_1_vekt(1,i) = -(m2*l1*t2+M_post*l1*l2)*sin(th2_vekt(1,i));
89     I1_2_vekt(1,i) = m2*t2^2+M_post*l2^2+(m2*l1*t2+M_post*l1*l2)*
90     cos(th2_vekt(1,i))+I2T2;
91     I2_2_vekt(1,i) = m2*t2^2+M_post*l2^2+I2T2;
92     C_2_vekt(1,i) = (m2*l1*t2+M_post*l1*l2)*sin(th2_vekt(1,i));
93 end
94
95 %% Dopocet casove zavislosti
96 dth1_dt_vekt = dth1_dp_vekt.*dp_dt_vekt;
97 dth2_dt_vekt = dth2_dp_vekt.*dp_dt_vekt;
98 ddth1_ddt_vekt = ddth1_ddp_vekt.*dp_dt_vekt.^2+dth1_dp_vekt.*
99     ddp_ddt_vekt;
100 ddth2_ddt_vekt = ddth2_ddp_vekt.*dp_dt_vekt.^2+dth2_dp_vekt.*
101     ddp_ddt_vekt;
102
103 %% Dopocet momentu
104 n1_vekt = ddth1_ddt_vekt.*I1_1_vekt+ddth2_ddt_vekt.*I2_1_vekt+
105     C_1_vekt.*(dth2_dt_vekt.^2+2*dth1_dt_vekt.*dth2_dt_vekt);
106 n2_vekt = ddth1_ddt_vekt.*I1_2_vekt+ddth2_ddt_vekt.*I2_2_vekt+
107     C_2_vekt.*dth1_dt_vekt.^2;
108
109 %% Grafy
110
111 %% Rychlostni profil
112 figure(1)
113 plot(p_vekt,dp_dt_vekt,'LineWidth',1.5)
114 title('Rychlostni profil')
115 grid minor
116 xlabel('p [1]')
117 ylabel('dp [1/s]')
118
119 figure(100)
120 plot(p_vekt,profil_scale,'LineWidth',1.5)
121 title('Rychlostni profil v [m/s]')
122 xlabel('p [1]')
123 ylabel('Rychlost [m/s]')
124 grid minor
125
126 figure(20)
127 plot(p_vekt,dp_dt_vekt,p_vekt,mvc,'LineWidth',1.5)
128 title('Rychlostni profil s limitni krivkou')
129 grid minor
130 legend('Rychlostni profil','Limitni krivka rychlosti')
131 xlabel('p [1]')

```

```

125 ylabel('dp [1/s]')
126
127 %% Numericka integrace
128 p_start = 0;
129 p_end = 1;
130 dp_start = 0;
131
132 [p_vekt_do,dp_vekt_do,U_vekt_do] = dopredna_integrace_U(p_start,
    dp_start,p_end,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,0,n1_min,
    n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min,ddth2_max,F1,
    F2);
133
134 p_start = 1;
135 p_end = 0;
136 [p_vekt_zp,dp_vekt_zp,L_vekt_zp] = zpetna_integrace_L(p_start,
    dp_start,p_end,ax,ay,bx,by,l1,l2,t1,t2,m1,m2,I1T1,I2T2,M,n1_min,
    n1_max,n2_min,n2_max,ddth1_min,ddth1_max,ddth2_min,ddth2_max,F1,
    F2);
137
138 [p_vekt_zp,in1,~] = unique(p_vekt_zp,'first');
139 dp_vekt_zp = dp_vekt_zp(in1);
140
141 [p_vekt_do,in1,~] = unique(p_vekt_do,'first');
142 dp_vekt_do = dp_vekt_do(in1);
143
144 dp_vekt_do = interp1(p_vekt_do,dp_vekt_do,p_vekt);
145
146 dp_vekt_zp = interp1(p_vekt_zp,dp_vekt_zp,p_vekt);
147
148
149 profil = zeros(1,length(p_vekt));
150 for i = 1:length(p_vekt)
151     if dp_vekt_do(1,i) < dp_vekt_zp(1,i)
152         profil(1,i) = dp_vekt_do(1,i);
153     else
154         profil(1,i) = dp_vekt_zp(1,i);
155     end
156 end
157
158 figure(70)
159 plot(p_vekt,profil,'LineWidth',1.5)
160 title('Profil s jednim prepinacim bodem')
161 grid minor
162 xlabel('p [1]')
163 ylabel('dp [1/s]')
164
165 %% Casovy prubeh
166 figure(90)
167 plot(p_vekt,t_vekt,'LineWidth',1.5)
168 title('Prubeh casu')
169 grid minor
170 xlabel('p [1]')
171 ylabel('t [s]')
172
173 %% Casova zavislost velicin
174

```

```

175 figure(110)
176 plot(t_vekt, th1_vekt, t_vekt, th2_vekt, 'LineWidth', 1.5)
177 title('Uhly v case')
178 xlabel('t [s]')
179 ylabel('Uhly [rad]')
180 grid minor
181 xlim([0 t_celk])
182 legend('th1', 'th2')
183
184 figure(170)
185 plot(t_vekt, n1_vekt, t_vekt, n2_vekt, 'Linewidth', 1.5)
186 title('Momenty v case')
187 xlabel('t [s]')
188 ylabel('Momenty [Nm]')
189 legend('n1', 'n2')
190 grid minor
191 xlim([0 t_celk])
192
193 figure(50)
194 plot(t_vekt, ddth1_ddt_vekt, t_vekt, ddth2_ddt_vekt, 'LineWidth', 1.5)
195 title('Zrychleni v case')
196 legend('ddth1', 'ddth2')
197 grid minor
198 xlabel('t [s]')
199 ylabel('Zrychleni [rad/s2]')
200 xlim([0 t_celk])
201 %ylim([-11 11])
202
203 figure(150)
204 plot(t_vekt, dth1_dt_vekt, t_vekt, dth2_dt_vekt, 'LineWidth', 1.5)
205 title('Uhlove rychlosti v case')
206 legend('dth1', 'dth2')
207 xlabel('t [s]')
208 ylabel('Uhlove rychlosti [rad/s]')
209 grid minor
210 xlim([0 t_celk])

```

Elektronické přílohy

Elektronické přílohy jsou všechny obsaženy v souboru **salva_priloha.zip**

Ve složce **matlab_kody** jsou všechny výše uvedené kódy ve formátu **.m**

Ve složce **simulink_model** je soubor **kontrola_vypoctu.slx** sloužící k simulačnímu ověření algoritmu. Před spuštěním simulace je nutné provést výpočet souboru **postproc.m** se vstupními parametry, pro které chceme simulací ověřovat vypočtené hodnoty.

Použitý software

- Matlab R2022b
- Matlab-Simulink
- Autodesk Inventor Professional 2021