

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



Energeticky optimální řízení robota napájeného z baterie

Diplomová práce

Petr Vejražka

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

2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vejražka** Jméno: **Petr** Osobní číslo: **483917**
Fakulta/ústav: **Fakulta strojní**
Zadávací katedra/ústav: **Ústav mechaniky, biomechaniky a mechatroniky**
Studijní program: **Robotika a výrobní technika**
Specializace: **Robotika**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Energeticky optimální řízení robota napájeného z baterie

Název diplomové práce anglicky:

Energetically - optimal control of robot motion supplied from battery

Pokyny pro vypracování:

Návrh energeticky optimálního řízení robota s ohledem na nelineární chování baterie bez rekuperace včetně sestavení modelu systému.

1. Ověřte předpokladu závislosti doby průběhu na spotřebované energii
2. Vytvořte model robota napájeného z baterie bez rekuperace
3. Formulujte energeticky optimální řízení pohybu robota
4. Řešte úlohu optimálního řízení
5. Proveďte srovnání optimalizace pomocí Pontryaginova principu minima s vhodnou přímou metodou optimalizace

Seznam doporučené literatury:

- Lewis, F.L. et al.: Optimal control, John Wiley 2012
- Valášek, M.: Synthesis of optimal trajectory of industrial robots, Kybernetika 22(1986), 5, 409-424
- Szumanowski, A., Chang, Y.: Battery management system based on battery nonlinear dynamics modeling, IEEE Trans. on Vehicle Technology, 57(2008), 3, 1425-1432

Jméno a pracoviště vedoucí(ho) diplomové práce:

prof. Ing. Michael Valášek, DrSc. centrum leteckého a kosmického výzkumu FS

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

Datum zadání diplomové práce: **28.04.2023**

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

.....

Petr Vejražka

Poděkování

Nejprve bych rád poděkoval vedoucímu práce, prof. Michaelu Valáškoví, za jeho neocenitelné rady, odborné vedení, předané znalosti a trpělivost během celého procesu psaní diplomové práce.

Dále děkuji své rodině a partnerce za nepřetržitou podporu a porozumění během mého studia.

Název práce:

Energeticky optimální řízení robota napájeného z baterie

Autor: Petr Vejražka

Druh práce: Diplomová práce

Akademický: 2022/2023

Univerzita: Fakulta strojní, ČVUT v Praze

Ústav: Mechaniky, biomechaniky a mechatroniky

Specializace: Robotika

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

Abstrakt:

Tato práce se zaměřuje na analýzu a optimalizaci energeticky efektivního pohybu robota napájeného z baterie. Cílem je ověřit předpoklad závislosti doby průběhu na spotřebované energii a vytvořit model robotického systému, který nezahrnuje rekuperaci energie zpět do baterie. V práci je formulováno energeticky optimální řízení pohybu robota, které minimalizuje spotřebovanou energii při dosažení požadovaného cíle. Hlavním úkolem je řešení úlohy optimálního řízení, kde jsou zohledněny fyzikální omezení robota a baterie. Aplikace Pontryaginova principu minima vede k nalezení bang-bang řízení, které je srovnáno s optimálním řízením nalezeným pomocí genetických algoritmů.

Klíčová slova:

Baterie, optimální řízení, nelineární systém, energeticky optimální

Title of thesis:

Energetically optimal control of robot motion supplied from battery

Author: Petr Vejražka

Thesis type: Master's thesis

Academic year: 2022/2023

University: Faculty of mechanical engineering, CTU in Prague

Department: Mechanics, Miomechanics and Mechatronics

Specialization: Robotics

Supervisor: prof. Ing. Michael Valášek, DrSc.

Abstract:

This work focuses on the analysis and optimization of energy-efficient motion of a battery-powered robot. The aim is to verify the assumption of the dependency of the duration of motion on the consumed energy and to create a model of a robotic system that does not include energy recuperation back into the battery. The work formulates energy-optimal control of robot motion, which minimizes the consumed energy while achieving the desired goal. The main task is to solve the optimal control problem, considering the physical limitations of the robot and the battery. The application of the Pontryagin's minimum principle leads to finding a bang-bang control, which is compared to the optimal control obtained using genetic algorithms.

Key words:

Battery, optimal control, nonlinear system, energetically optimal

Obsah

Úvod	1
1 Baterie	2
1.1 Typy baterií a článků	2
1.1.1 Základní rozdělení článků	3
1.1.2 Rozdělení dle elektrochemických vlastností	4
1.2 Fyzikální vlastnosti a parametry baterií	5
1.2.1 Kapacita, energie a výkon	5
1.2.2 Stav nabití baterie	5
1.2.3 Stav zdraví baterie	7
1.2.4 Napětí v baterii	7
1.3 Modelování baterií	8
1.3.1 Model s vnitřním odporem	8
1.3.2 RC model	9
1.3.3 Theveninův model	10
1.3.4 PNGV model	11
1.3.5 Vliv provozních podmínek na parametry baterie	12
2 Model systému	15
2.1 Komponenty systému	15
2.1.1 Model baterie	16
2.1.2 Řízení motoru	17
2.1.3 Model motoru	19
2.1.4 Model kola	20
2.1.5 Model robotického ramene	21

2.2	Model systému s robotickým ramenem	22
2.2.1	Stavový popis systému s robotickým ramenem	23
2.2.2	Přesná linearizace systému s robotickým ramenem	23
2.3	Model systému mobilního robota	25
2.3.1	Přesná linearizace modelu mobilního robota	26
3	Optimální řízení	28
3.1	Definice problému řízení	28
3.1.1	Kritérium optimality pro nelineární systém	29
3.1.2	Kritérium optimality pro lineární systém	30
3.2	Optimální řízení dle PMP	30
3.2.1	Podmínky optimálního řízení dle PMP	30
3.2.2	Aplikace PMP	31
3.2.3	Analytické řešení	34
3.3	Numerická řešení optimálního řízení	35
3.3.1	Transformace systému pro numerické řešení \mathbf{T}_{free}	35
3.3.2	Numerické řešení PMP	36
3.3.3	Numerické řešení analytického odhadu	37
3.3.4	Numerické řešení pomocí přímkové aproximace řízení	38
3.3.5	Numerické řešení pomocí přepínání limitních hodnot	39
3.3.6	Numerické řešení pomocí genetických algoritimů	40
4	Simulace a analýza	41
4.1	Numerický model pro simulaci systému	41
4.1.1	Parametry baterie	41
4.1.2	Parametry motoru	43
4.1.3	Sestavení modelů pro simulaci	43
4.2	Předpoklad závislosti spotřebované energie na době pohybu	45
4.3	Analýza chování lineárního systému	46
4.4	Simulace systému s optimálním řízením	48
4.4.1	Optimální analytické řešení	48
4.4.2	Optimální řízení pomocí přímkové aproximace	51
4.4.3	Optimální řízení přepínáním na limitní křivky	53

4.4.4	Optimálního řízení dle GA	54
4.5	Srovnání výsledků optimalizačních metod	56
	Závěr	58
	Seznam použitých zdrojů	60
	Seznam použitých obrázků	63
	Seznam použitých tabulek	65
	Přílohy	66

Seznam použitých veličin

Veličina	Jednotka	Popis
ϕ	rad	Úhel otočení motoru
ω	rad/s	Úhlová rychlost motoru
C_P	F	Kapacita kondenzátoru
C_n	Ah	Celková kapacita baterie
C_{ss}	Vs	Konstanta motoru
DoD	%	Stav vybití baterie
i_c	A	Elektrický proud
i_{cmax}	A	Maximální proud soustavy
i_{cmin}	A	Minimální proud soustavy
i_R	A	Elektrický proud v obvodu baterie
J_m	kgm^2	Moment setrvačnosti motoru
J_R	kgm^2	Moment setrvačnosti rob. ramene
L_m	H	Indukčnost kotvy motoru
p	%	Parametr sepnutí PWM - střída
P_E	W	Příkon elektroniky v systému
P_M	W	Příkon motoru
R_m	Ω	Odpor kotvy motoru
R_O	Ω	Interní odpor baterie
R_P	Ω	Odpor rezistoru v RC větvi
SoE	%	Stav kapacity energie v baterii
SoC	%	Stav nabití baterie

SoC_{max}	%	Maximální stav nabití baterie
SoC_{min}	%	Minimální stav nabití baterie
SoH	%	Stav zdraví baterie
u_m	V	Svorkové napětí na kotvě motoru
u_{OC}	V	Napětí baterie na prázdko
t	s	Čas
t_f	s	Celkový čas pohybu
T	K	Teplota

Seznam použitých zkratek

Zkratka	Popis
BMS	Battery Management System
DC	Direct Current - Stejnoseměrný proud
GA	Genetické algoritmy
OCV	Open Circuit Voltage
PMP	Pontryagin Minimum Principle
PNGV	Partnership for a New Generation of Vehicle
PWM	Pulse Width Modulation
SW	Software

Úvod

V dnešním světě se robotika stává čím dál důležitější oblastí vědeckého výzkumu a průmyslového využití. Stále se objevují nové možnosti robotických aplikací, které umožňují výrazně usnadnit a zefektivnit řadu činností ve společnosti, ať už se jedná o zemědělství, výrobu, automobilový průmysl, zdravotnictví nebo vesmírné aplikace.

Jednou z největších výzev robotiky jsou autonomní mobilní roboty, které budou schopny vykonávat složité a pro člověka fyzicky náročné práce. Pro dosažení co nejlepšího výkonu a prodloužení doby provozu je klíčové správné řízení energie, zejména u robotů napájených bateriemi. Energeticky optimální řízení těchto robotů je tedy nezbytným tématem pro dosažení vysokého výkonu, dlouhé výdrže a snížení celkových provozních nákladů. Tato diplomová práce se zabývá nalezením optimální strategie řízení robota napájeného baterií a současně zkoumá vztah mezi optimálním řízením a stavem baterie.

Pro úspěšný návrh a řízení robotů je nezbytné mít k dispozici spolehlivé simulační modely, které umožňují předpovědět jejich chování a optimalizovat jejich výkon. V diplomové práci jsou popsány modely a jejich vlastnosti pro součásti: baterie, DC motor, PWM a poháněný mechanismus (rameno průmyslového robota nebo kolo). Největší důraz je dán na baterii a její model, jelikož se jedná o velmi složitý systém s celou řadou nelineárních vlastností, které mohou při řízení robota představovat problém.

V praktické části je nejprve sestaven model pro simulaci robotického systému, který zohledňuje nelineární chování baterie. Důležité je poznamenat, že baterie projevuje odlišné chování během nabíjení a vybíjení. Model popsáný v této diplomové práci se zaměřuje na fázi vybíjení a není možné využít rekuperace energie z pohybu robota pro dobití baterie, což výrazně ovlivňuje nalezené optimální řešení. Pro systém s těmito vlastnostmi platí předpoklad, že při nižší rychlosti pohybu je pro ujetí stejné vzdálenosti potřeba menší množství energie. Tento předpoklad je v práci ověřen.

V rámci práce budou porovnány dvě metody pro dosažení energeticky optimálního řízení - Pontryaginův princip minima a vhodná přímá metoda optimalizace. Výsledky těchto metod budou dále srovnány a komentovány s doporučením, který postup je vhodnější pro vybrané aplikace.

Kapitola 1

Baterie

Baterie představuje jeden ze zásadních zdrojů energie ve 21. století, jehož význam bude v příštích letech společně s příchodem elektromobility stále růst. Na úvod je nutné vysvětlit rozdíl mezi pojmy "baterie" a "článek", které jsou často zaměňovány, ale v praxi představují odlišné komponenty.

Baterie a článek jsou součástmi elektrochemického systému, který slouží k uchování elektrické energie. Elektrický článek je základní stavební jednotkou baterie. Jedná se o elektrochemický systém, který se skládá z anody, katody a elektrolytu. Článek sám o sobě může generovat malé množství elektrické energie, ale jeho kapacita a výkon jsou omezené. Abychom byli schopni poskytovat vyšší výkon je nutné spojit články dohromady, čímž vzniká baterie s vyšší celkovou kapacitou a schopností poskytnout dostatečný výkon pro pohon zařízení.

V dalších částech této kapitoly jsou podrobněji popsány typy článků i baterií a charakteristiky jejich chování. Jedná se o rozsah dostatečný k pochopení vlastností modelu sestaveného v praktické části, problematika baterií je detailněji řešena například v publikaci [1].

1.1 Typy baterií a článků

Elektrochemické články lze dělit, dle celé řady vlastností od velikosti, a tvaru přes chemické vlastnosti, až po cenu. V této práci jsou popsána dvě tato rozdělení. Prvním je dělení dle principu fungování elektrochemického článku a je nazýváno základní, druhým je rozdělení dle chemického složení a elektrochemických vlastností.

Elektrochemické články lze následně spojovat do baterií paralelním, sériovým nebo kombinovaným zapojením. Typ zapojení rozhoduje o vlastnostech baterie, které podléhají Kirchhoffovým zákonům. Platí tedy, že v sériovém zapojení je napětí baterie rovno součtu napětí článků a v paralelním zapojení je výsledný proud procházející baterií roven součtu proudů procházejících články [1].

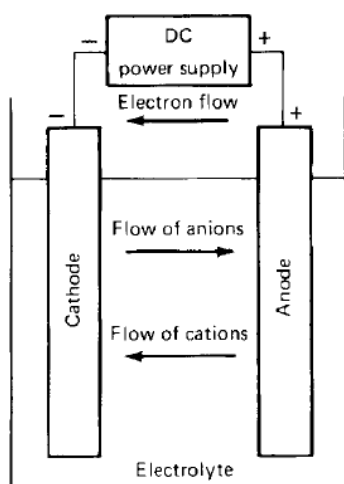
1.1.1 Základní rozdělení článků

1. Primární článek

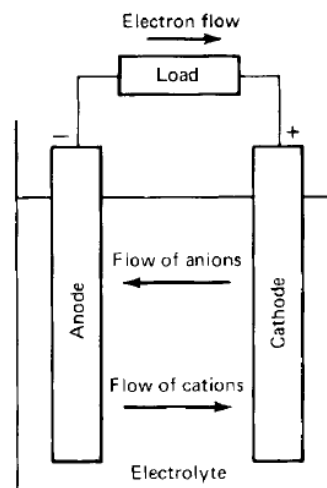
Jsou články, které již nejsou po vybití znovu použitelné. Výhodou primárních článků je především jejich nízká cena, vyšší hustota energie než mají akumulátory a bezúdržbové použití. Používají se tedy především u energeticky nenáročných aplikací jako jsou např. hodinky, dálkové ovladače a celá řada dalších zařízení[2].

2. Sekundární článek - akumulátor

Jsou články, které lze po vybití elektricky dobít do původního stavu (viz. Obr. 1.1) pomocí průchodu proudu v opačném směru, než je směr proudu při vybití (viz. Obr. 1.2). Akumulátory jsou v současné době nejpoužívanějším typem článků, který je instalován do celé řady výrobků přes elektromobily, elektrická úložiště, až po spotřebiče v domácnostech [1].



Obr. 1.1: Akumulátor při nabíjení [1]

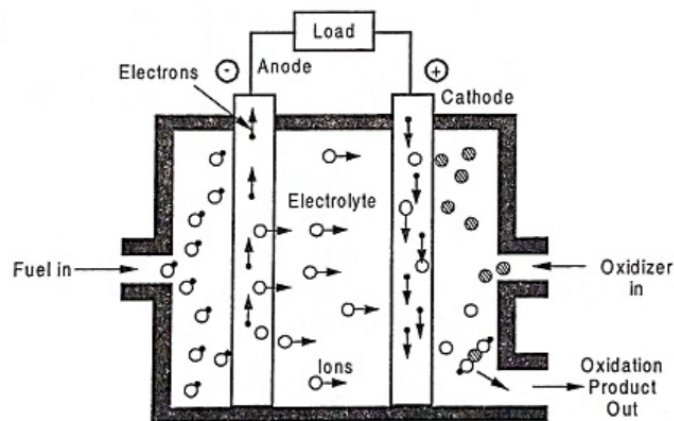


Obr. 1.2: Akumulátor při vybíjení [1]

3. Palivový článek

Palivové články představují konstrukčně podobný prvek jako baterie, neboť dokáží přímo přeměňovat chemickou energii elektroaktivních materiálů (paliva a oksylichovadla) na elektrickou energii. Hlavním rozdílem oproti primárním a sekundárním článkům je fakt, že elektrody palivového článku nejsou součástí chemické reakce, ale plní pouze funkci katalyzátoru (viz. Obr. 1.3). Tato vlastnost umožňuje palivovým článkům dosáhnout neomezené doby použití, která je závislá pouze na přísunu paliva a oksylichovadla a celkové životnosti elektrod.

Hlavní nevýhodou je jejich omezené schopnost rychle reagovat na změny velikosti zatížení. To znamená, že samostatný palivový článek je vhodný pro aplikace, kde se vyskytuje převážně konstantní zatížení [1].



Obr. 1.3: Palivový článek [3]

1.1.2 Rozdělení dle elektrochemických vlastností

Maximální energie, kterou lze dodat elektrochemickým systémem, je založena na typech a množství použitých aktivních materiálů. Lze tedy říci, že napětí a kapacita elektrochemického článku jsou funkcí anody, katody a jejich materiálů.

Na základě elektrochemických vlastností jsou články nejčastěji děleny dle materiálu anody. Mezi nejčastěji používané anody patří materiály uvedené v Tabulce 1.1, kde je také uveden jejich elektrický potenciál při teplotě 25° C.

Materiál	H_2	Li	Na	Mg	Pb	Ca	Fe
Potenciál	0	-3,01	-2,71	-2,38	-0,13	-2,84	-0,4

Tabulka 1.1: Elektrický potenciál materiálů používaných na anody [1]

Tyto materiály jsou kombinovány s kationty za účelem vytvořit stabilní elektrochemický článek s co největším rozdílem elektrického potenciálu. Nejčastěji používané typy článků a jejich charakteristické vlastnosti jsou popsány Tabulkou 1.2.

V praxi je však realizován pouze zlomek teoretické energie baterie. To je způsobeno potřebou elektrolytu a nereaktivních součástí (nádoby, separátory, elektrody), které zvyšují hmotnost a objem baterie.

Poslední dobou se také začínají objevovat nové typy článků, které jsou vhodné pro trakční aplikace a elektromobilitu obecně. Prvním takovým typem je Lithium-Sulfurová (Li-S), která se vyznačuje nízkou hmotností a velmi vysokou specifickou energií (450 Wh/kg) [5]. Druhým typem je Sodík-iontová (NIB), která je přelomová hlavně tím, že její výroba je výrazně levnější než je tomu u baterií Lithiových, kde je také používáno Lithium, které lze považovat za vzácnou surovinu vyskytující se pouze v několika zemích světa [6].

	Lead acid*	NiMH	Lithium-iont
Specifická energie [Wh/kg]	20-35	50-70	100-200
Energie na objem [Wh/l]	50-90	150-200	150-250
Specifický výkon [W/kg]	250	<1000	<2000
Jmenovité napětí [V]	2,1	1,2	3,6
Provozní teplota [C°]	-20 až 60	-20 až 50	-15 až 50
Počet cyklů	700	2000	2000

Tabulka 1.2: Přehled vlastností běžně užívaných baterií [1],[4]

Lead acid* - Olověný akumulátor

1.2 Fyzikální vlastnosti a parametry baterií

1.2.1 Kapacita, energie a výkon

Jmenovitá kapacita článku udává množství nabití v ampérhodinách (Ah) nebo miliampérhodinách (mAh), na které je článek dimenzován. Jednotkou SI pro nabíjení je coulomb (C), takže pro následné výpočty v modelu je důležité přepočítat běžně užívanou jednotku Ah ($1 \text{ Ah} = 3600 \text{ C}$).

Jmenovitá energie článku je množství energie ve watthodinách (Wh) nebo kilowatthodinách (kWh), na kterou je článek dimenzován. Je vypočítána jako jmenovité napětí článku vynásobené jeho nominální nabíjecí kapacitou.

Obě definice kapacity mají své opodstatnění a lze je vzájemně dopočítat. Protože se však v této práci zaměřujeme na vytváření modelů, které spojují vstupní proud baterie na jeho vnitřní stav a výstupní napětí, nabíjecí kapacita je vhodnější používat tento pojem.

Do třetice je nutné zavést pojem **výkon**, což je množství energie přenesené za jednotku času. Výkon se měří ve wattech (W) nebo kilowattech (kW). Maximální výkon, který může článek dodat, je omezen vnitřním odporem článku a nelze jej snadno kvantifikovat. Výkon je regulován pomocí minimálních a maximálních limitů napětí na svorkách článku [1].

1.2.2 Stav nabití baterie

SoC (State of Charge) článku nebo baterie je jedním z nejdůležitějších parametrů akumulátoru. Jeho definice a měření však přináší řadu různých problémů. Obecně se SOC určí jako podíl náboje, kterým článek aktuálně disponuje ku nominální kapacitě reprezentující maximální možný náboj. Matematicky vyjádřeno:

$$SoC = SoC_0 - \frac{1}{C_n} \int i_c(t) dt \quad (1.1)$$

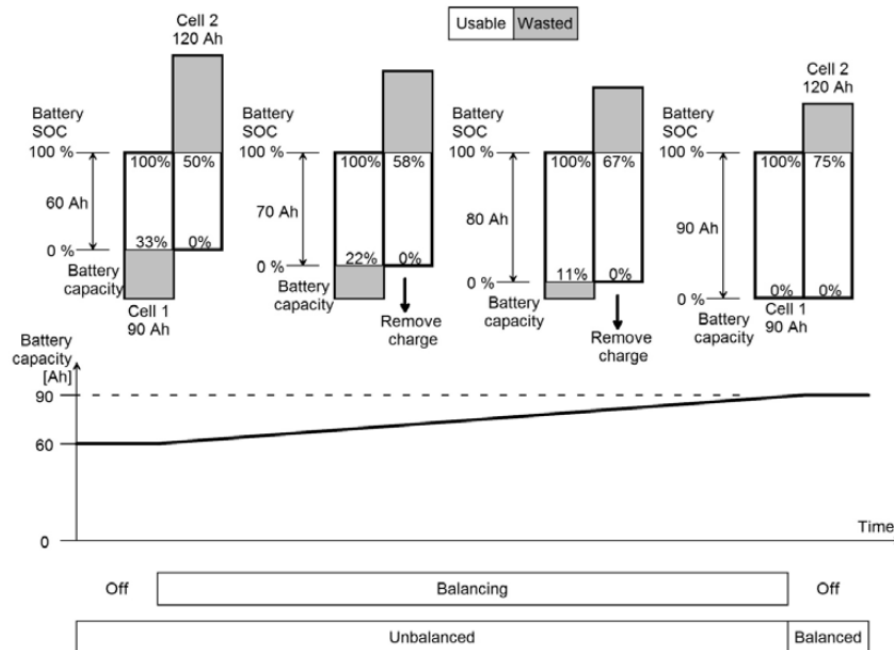
Kde SoC_0 představuje počáteční stav nabití a i_c proud procházející baterií, který při vybíjení nabývá kladných hodnot a při nabíjení záporných [1].

Obdobně jako SoC se používá i popis pomocí DoD (Depth of discharge), což lze do češtiny přeložit jako "hloubka vybití baterie". Jejich vztah je popsán následovně:

$$DoD = 1 - SoC \quad (1.2)$$

Pro správné fungování baterie je žádoucí, aby byl stav nabití baterie udržován v příslušných mezích, proto je nezbytné být schopen odhadnout stav nabití baterie. Odhad stavu nabití baterie (SoC) není snadný úkol, protože SoC závisí na mnoha faktorech, jako je teplota, kapacita baterie a vnitřní odpor. Existuje ale řada metod výpočtu, kde lze uvést několik základních způsobů mezi které patří: Odhad ze svorkového napětí, Integrace proudu (Coulomb counting), Nelineární mapování, aplikace Kalmanova filtru a různá kombinace těchto metod [7].

SoC je stav, který je potřeba sledovat jak pro celou baterii, tak i pro jednotlivé články, protože v reálných bateriích se SoC může mezi jednotlivými články lišit, což může způsobit snížení celkové kapacity baterie (viz. Obr. 1.4). Aby zatížení mezi články bylo rovnoměrné a životnost baterie byla co nejdélejší, tak jsou pokročilejší bateriové systémy vybaveny BMS (Battery management system), který slouží k odhadu celkové úrovně SoC, balancování článků a termoregulaci systému [8].



Obr. 1.4: Vliv balancování článků na kapacitu baterie [8]

1.2.3 Stav zdraví baterie

Před zavedením SOH je nutné definovat pojem stárnutí baterie. Jedná se o jev, který je možno pozorovat u akumulátoru, kdy se množství elektrického náboje, který je baterie schopna dodat, používáním snižuje [8].

Totéž platí pro výkon baterie. Čím více baterie stárne, tím méně energie je k dispozici. SOH baterie lze vyjádřit jako poměr maximální dostupné kapacity v současném stavu k nominální kapacitě baterie na začátku životnosti baterie. Matematicky vyjádřeno:

$$SoH = \frac{C_{navailabe}}{C_{nfresh}} \quad (1.3)$$

Obecně platí, že pokud kapacita baterie klesne pod 80% původní kapacity, je baterie považována za nepoužitelnou, což je způsobeno exponenciální degradací kapacity baterie. Existuje mnoho vnitřních a vnějších faktorů, které ovlivňují zdraví baterie a její výkonnost v průběhu času. Některé z vnitřních faktorů zahrnují materiál baterie, staří baterie a zvýšení vnitřního odporu. Vnějšími faktory jsou provozní teplota, nejisté jízdní podmínky, přebíjení/vybíjení, vysoká rychlost nabíjení/vybíjení nebo nesprávné cyklování nabíjení/vybíjení. Vzhledem k mnoha neznámým a nepředvídatelným faktorům, které ovlivňují stav baterie, je odhad SoH baterie značně náročný problém [9].

1.2.4 Napětí v baterii

Napětí je při popisu baterií velmi často užívaný termín, který ale téměř pokaždé má jiný význam, proto byla do diplomové práce zařazena tato kapitola, která má za cíl všechny termíny spjaté s napětím vysvětlit.

Teoretické napětí	funkce materiálů anody a katody, elektrolytu a teploty
Napětí naprázdno	stav bez zátěže - podobné teoretickému napětí
Jmenovité napětí	provozní napětí baterie
Pracovní napětí	odpovídá skutečnému provoznímu napětí baterie pod zátěží
Koncové napětí	konec výboje

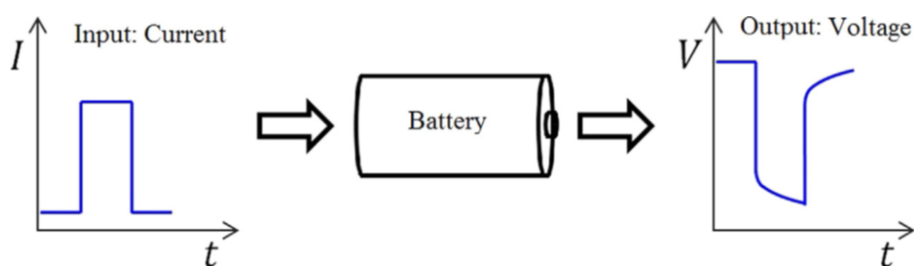
Při použití olověného akumulátoru jako příkladu je teoretické napětí a napětí naprázdno 2,1 V, jmenovité napětí je 2,0 V, pracovní napětí je mezi 1,8 - 2,0 V a koncové napětí je typicky 1,75 V a 1,5 V pro zatížení při roztáčení motoru. Při nabíjení se napětí může pohybovat od 2,3 do 2,8 V [1].

1.3 Modelování baterií

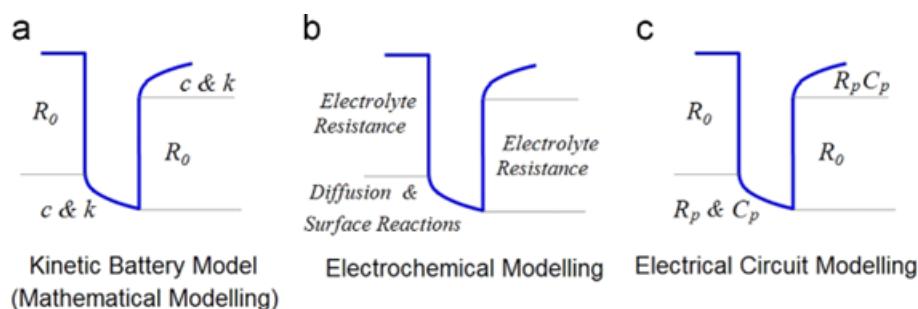
Jak již bylo zmíněno, tak baterie je silně nelineárním, dynamickým systémem s komplikovaným elektrochemickým procesem, který může být ovlivněn teplotou, proudem zatížení, stavem nabití (SoC), stavem zdraví baterie (SoH), prostředím a dalšími faktory.

Pro popis chování lze použít různé modely, včetně zjednodušených elektrochemických [10], matematických [11], tepelných [12], modelů stárnutí [13], ekvivalentních elektrických modelů [14] a jejich různých kombinací.

Na Obr. 1.6 jsou znázorněny rozdílné přístupy těchto metod k popisu odezvy svorkového napětí baterie na jednotkový skok proudu (viz. Obr. 1.5), což lze považovat za charakteristickou vlastnost baterie.



Obr. 1.5: Odezva napětí na baterii na skokovou změnu proudu [15]



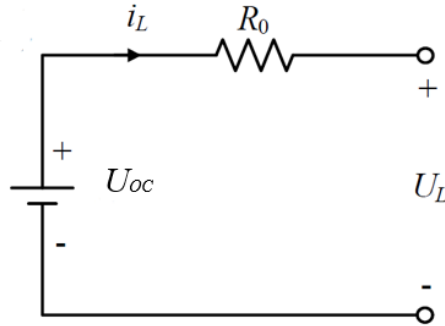
Obr. 1.6: Rozdílné přístupu k modelování baterie [15]

Elektrochemické modely jsou vhodné zejména pro fyzický návrh článků, matematické jsou založeny na empirických vzorcích a dokáží predikovat stavy baterie. V této práci jsou popsány detailněji ekvivalentní elektrické modely z důvodu jejich snadné aplikace do popisu mechatronických soustav a následné numerické simulaci.

1.3.1 Model s vnitřním odporem

Elektrický ekvivalentní model se skládá z ideálního zdroje napětí U_{oc} a vnitřního odporu R_{int} . Proud i_L nabývá kladné hodnoty, když se baterie vybíjí, a záporné hodnoty při nabíjení. Rovnice (1.4) znázorňuje vztah svorkového napětí U_L , napětí naprázdno, proudu a vnitřního odporu baterie. Základní Rint model popisuje lineární chování baterie bez znalosti jakýchkoliv informací o vnitřním složení a procesech

probíhajících v baterii. Rozšíření modelu lze dosáhnout pomocí definice zdroje napětí a vnitřního odporu jako funkce stavu nabití (SoC) a teploty, tyto závislosti lze aplikovat i u všech dalších zmíněných modelů a jsou blíže rozebrány v kapitole 1.3.5 [16], [17].

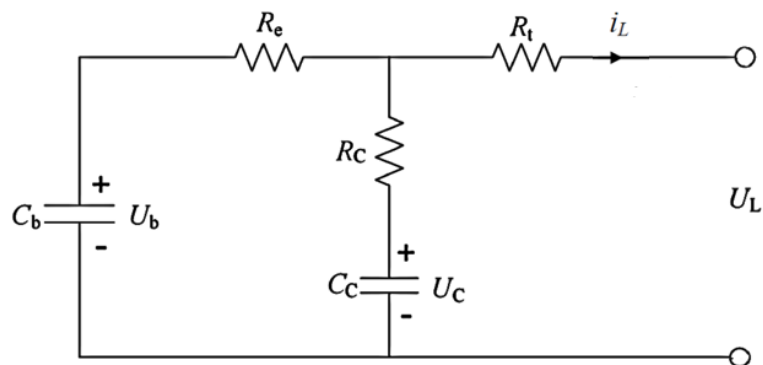


Obr. 1.7: Model s vnitřním odporem

$$U_L = U_{oc} - i_L R_0 \quad (1.4)$$

1.3.2 RC model

RC model byl navržen firmou SAFT Battery Company v roce 2000 a dosáhl aplikace v průmyslu prostřednictvím softwaru Advisor. Jak je znázorněno na Obr. 1.8, skládá se ze dvou kondenzátorů (C_b , C_c) a tří rezistorů (R_t , R_e , R_c). Kondenzátor C_c , který má malou kapacitu a většinou reprezentuje povrchové efekty baterie, se nazývá povrchový kondenzátor. Kondenzátor C_b , který má velmi velkou kapacitu a představuje schopnost baterie chemicky ukládat náboj, se nazývá velkokapacitní (bulk) kondenzátor. SoC může být určeno napětím na velkém kondenzátoru. U_b a U_c jsou napětí na kondenzátorech C_b a C_c . Elektrické chování obvodu lze vyjádřit rovnicemi (1.5) a (1.6) [16], [17].



Obr. 1.8: Schéma RC modelu

$$\begin{bmatrix} \dot{U}_b \\ \dot{U}_c \end{bmatrix} = \begin{bmatrix} -\frac{1}{C_b(R_e+R_c)} & \frac{1}{C_b(R_e+R_c)} \\ \frac{1}{C_c(R_e+R_c)} & -\frac{1}{C_c(R_e+R_c)} \end{bmatrix} \begin{bmatrix} U_b \\ U_c \end{bmatrix} + \begin{bmatrix} -\frac{R_c}{C_b(R_e+R_c)} \\ -\frac{R_e}{C_c(R_e+R_c)} \end{bmatrix} i_L \quad (1.5)$$

$$\begin{bmatrix} \dot{U}_L \end{bmatrix} = \begin{bmatrix} -\frac{R_c}{R_e+R_c} & \frac{R_c}{R_e+R_c} \end{bmatrix} \begin{bmatrix} U_b \\ U_c \end{bmatrix} + \left[-R_t - \frac{R_e R_c}{R_e+R_c} \right] i_L \quad (1.6)$$

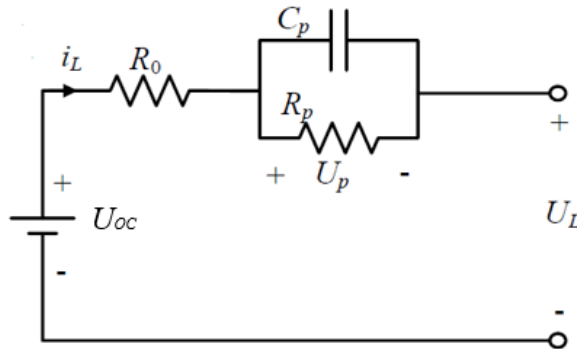
1.3.3 Theveninův model

Na základě modelu s vnitřním odporem mohou být k ekvivalentnímu modelu přidány paralelní RC větve pro simulaci polarizačního efektu v procesu nabíjení a vybíjení baterií. Zahrnutí více RC párů a zohlednění dalších závislostí poskytuje přesnější reprezentaci dynamiky baterie, nicméně to také zvýší výpočetní složitost. Použití více RC párů také rozšíří počet stavů potřebných k popisu v odpovídajícím stavovém modelu a zvýší řád přenosové funkce.

V této diplomové práci jsou popsány pouze modely prvního a druhého řádu z důvodu vhodnosti implementace do komplexních výpočtů pro celý systém, kde tyto modely dostatečně přesně popisují dynamické chování baterií.

Model prvního řádu

Celkový vnitřní odpor modelu se skládá z ohmického odporu R_0 , který je způsoben kontaktem elektrod s elektrolytem baterie a polarizačního odporu R_p , který je způsoben chemickými reakcemi v elektrodách během nabíjení a vybíjení. RC obvod modelu má na svých svorkách napětí U_p s výstupním proudem i_p a popisuje vliv chemických reakcí na povrchu elektrody a přenos hmoty iontů. Vztah těchto veličin je popsán rovnicemi (1.7) a (1.8), kde U_L představuje svorkové napětí baterie. Vzhledem ke složitosti fungování baterie je však dynamický výkon jedné RC větve omezený, což nemůže přesně odrazet polarizační efekt a další faktory ovlivňující baterie [16], [17]. Model je zobrazen na Obr. 1.9.



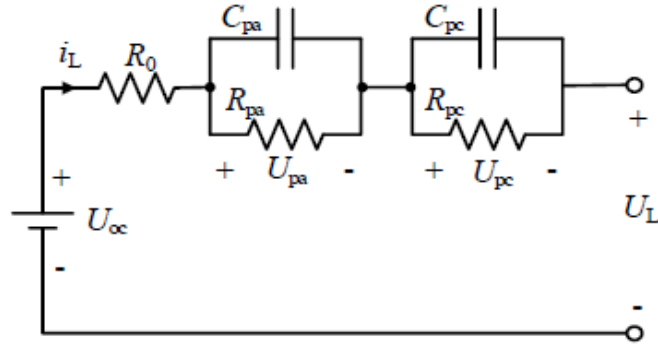
Obr. 1.9: Schéma Theveninova modelu prvního řádu

$$\dot{U}_P = -\frac{U_P}{R_P C_P} + \frac{i_L}{C_P} \quad (1.7)$$

$$U_L = U_{oc} - U_P - i_L R_0 \quad (1.8)$$

Model druhého řádu

Přidáním druhého RC páru do modelu je popsán efekt aktivační polarizace související s elektrochemickými reakcemi na povrchu elektrody. V průmyslu a akademické sféře je model druhého řádu také nazvaný model duální polarizace. Model je zobrazen na Obr. 1.10. V tomto modelu je celkový vnitřní odpor složen z vnitřního ohmického odporu R_0 , koncentračního polarizačního odporu R_{Pc} a aktivačního polarizačního odporu R_{Pa} . Kondenzátory C_{Pa} a C_{Pc} slouží k simulaci chování baterie při přechodných stavech vybíjení a nabíjení. Elektrické chování modelu je popsáno rovnicemi (1.9) a (1.9)[16], [17].



Obr. 1.10: Schéma Theveninova modelu druhého řádu

$$\begin{bmatrix} \dot{U}_{Pa} \\ \dot{U}_{Pc} \end{bmatrix} = \begin{bmatrix} -\frac{1}{C_{Pa}R_{Pa}} & 0 \\ 0 & -\frac{1}{C_{Pc}R_{Pc}} \end{bmatrix} \begin{bmatrix} U_{Pa} \\ U_{Pc} \end{bmatrix} + \begin{bmatrix} \frac{1}{C_{Pa}} \\ \frac{1}{C_{Pc}} \end{bmatrix} i_L \quad (1.9)$$

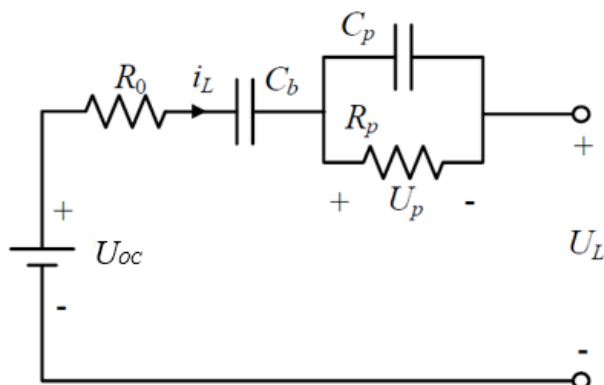
$$U_L = U_{oc} - U_{Pa} - U_{Pc} - i_L R_0 \quad (1.10)$$

1.3.4 PNGV model

Ministerstvo energetiky USA vydalo různé příručky související s testováním baterií pro (hybridní a plug-in) elektrická vozidla. Z příruček pro testování baterií FreedomCar [18] vznikl model baterie FreedomCar nebo jinak nazývaný model PNGV (Partnership for a New Generation of Vehicles).

Model PNGV, jak je znázorněno na Obr. 1.11, přidává do série deskovou kapacitu C_b k základnímu Theveninovu modelu. Kapacita má hodnotu $1/U_{oc}$ a používá se k

simulaci charakteristiky U_{oc} baterie měnící se s akumulací proudu. Tento model je však obtížné identifikovat parametry a má vysokou vyšší složitost výpočtu, takže není vhodný pro odhad SOC v reálném čase v praktické aplikaci [16], [17].



Obr. 1.11: Schéma modelu PNGV

1.3.5 Vliv provozních podmínek na parametry baterie

Všechny modely v předchozích kapitolách jsou popsány pomocí různých parametrů a jsou aplikovatelné pro popis chování baterií. Parametry modelů baterie ale nejsou konstanty s pevnou hodnotou a mění se za různých provozních podmínek jako je SoC, SoH, teplota nebo velikost procházejícího proudu. Aby bylo dosaženo reálného popisu chování fyzické baterie, musí být tyto vlastnosti také zaneseny do modelu. Obecně tedy lze všechny parametry uvažovat jako funkci provozních podmínek:

$$x_i = f(\text{SoC}, \text{SoH}, T, i) \quad (1.11)$$

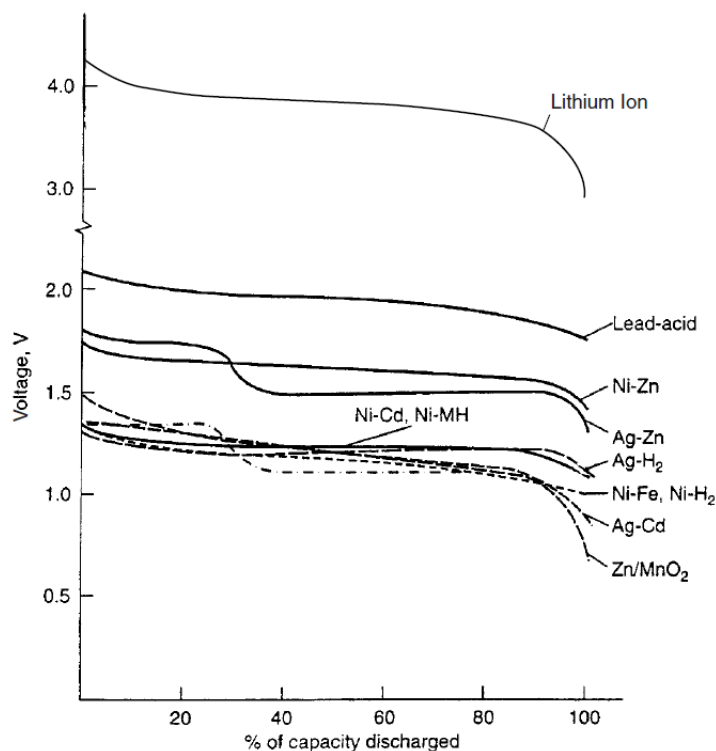
Existuje více možností jak tyto závislosti popsat, ale mezi základní metody se řadí look-up (vyhledávací) tabulky [19] a polynomiální funkce [20]

Look-up tabulky i polynomy jsou získány z experimentálních dat. Z experimentálních dat jsou vytvořeny tabulky závislostí, které je možné využít pro numerické simulace, nebo jsou jimi proloženy polynomy, tak aby se získalo množství neznámých koeficientů. Takto získané look-up tabulky i polynomiální funkce jsou neměnné, takže všechny možné provozní podmínky je třeba vzít v úvahu během procesu parametrizace modelu.

Tyto podmínky mohou zahrnovat různé hodnoty SoC, SoH, teploty i proudu. Tyto požadavky vyžadují obrovské množství testovacích dat, které není s ohledem na stárnutí baterie není snadné získat. Řešením tohoto problému je schopnost adaptace uvnitř modelu. Adaptivní model se může měnit získáváním parametrů online pomocí technik identifikace systému. Příklad adaptivního modelu lze nalézt například v publikaci [21].

V této diplomové práci se zaměříme na popis vlivu zejména SoC, který je dále aplikován v praktické části.

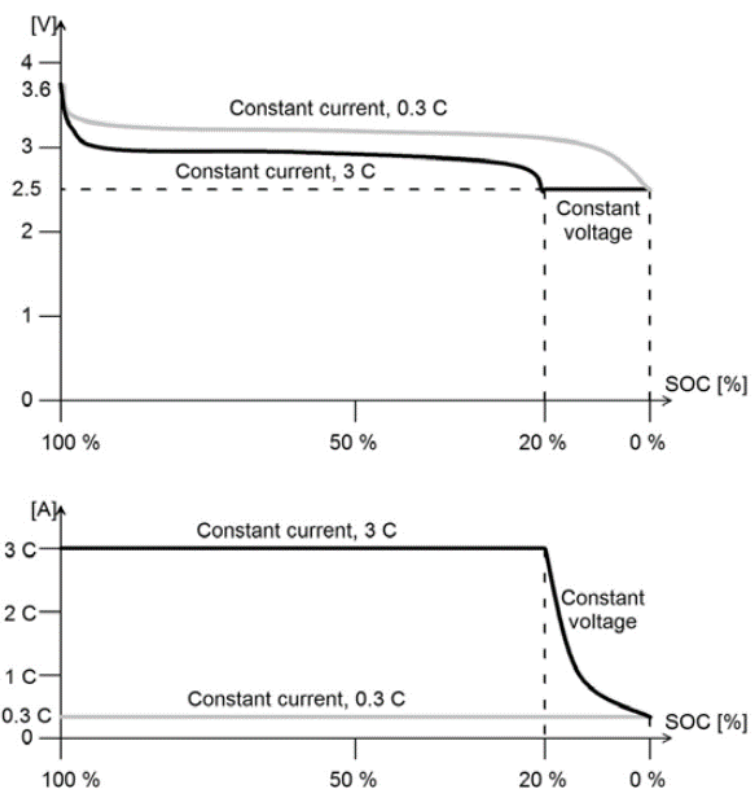
Typickým parametrem, který je ovlivněn hodnotou SoC je napětí naprázdno. Hodnoty napětí a jejich průběh se významně liší i dle typu a složení elektrochemického článku. Srovnání průběhu závislosti napětí na SoC pro různé články je zobrazeno na Obr. 1.12.



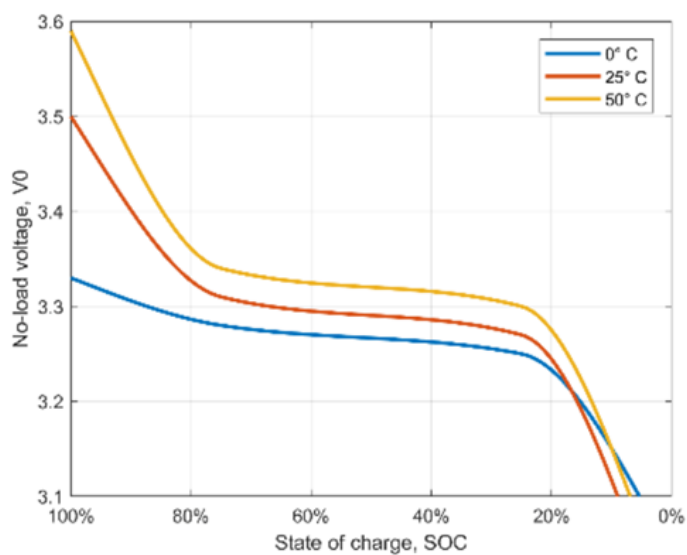
Obr. 1.12: Srovnání závislosti U_{oc} na SoC různých typů elchem. článků[1]

Na příkladu Li-ion baterie na Obr. 1.13 je zobrazena závislost tohoto průběhu na hodnotě procházejícího proudu baterií. Z obrázku také vyplývá, že při využívání vyšších hodnot proudu není možné využít celou kapacitu baterie. Obdobně lze popsat i vliv teploty, který je zobrazen na Obr. 1.14.

Stejně jako byl popsán vliv SoC a teploty na napětí na prázdko, lze z experimentálních dat analogicky popsat jakýkoliv parametr vyskytující se v náhradních elektrických modelech. Příkladem parametru který se často uvažuje jako funkce prostředí je i vnitřní odpor baterie, který do modelu zanáší rozdílné chování při nabíjení a vybíjení baterie.



Obr. 1.13: Závislost U_{oc} Lithium-iont baterie na SoC a procházecím proudem [8]



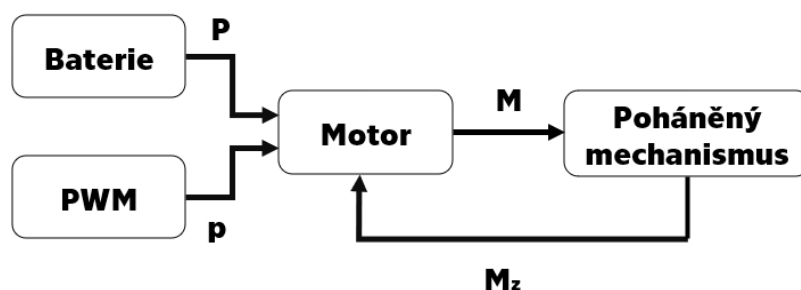
Obr. 1.14: Závislost U_{oc} baterie na teplotě[22]

Kapitola 2

Model systému

V této kapitole jsou na základě rešerše sestaveny teoretické modely, které lze použít pro popis mechatronických systémů využívajících baterie a DC motory. Největší výzvu při sestavení celkového modelu představoval popis řízení, který zároveň slouží jako propojení baterie s motorem. Po provedení důsledné rešerše nebyl nalezen žádný obecný teoretický model, který by umožňoval sledovat chování uvnitř baterie a zároveň chování systému jako celku, proto je v této práci navržen.

V první části jsou popsány modely komponent, které jsou následně syntetizovány do komplexního modelu pro popis chování celého systému dle logiky znázorněné na Obr. 2.1. Rozsah této práce je však omezen pouze na systémy, které nejsou schopny rekuperace energie, výsledný model tedy popisuje pouze chování baterie při vybíjení.



Obr. 2.1: Schéma obecného mechatronického systému s baterií

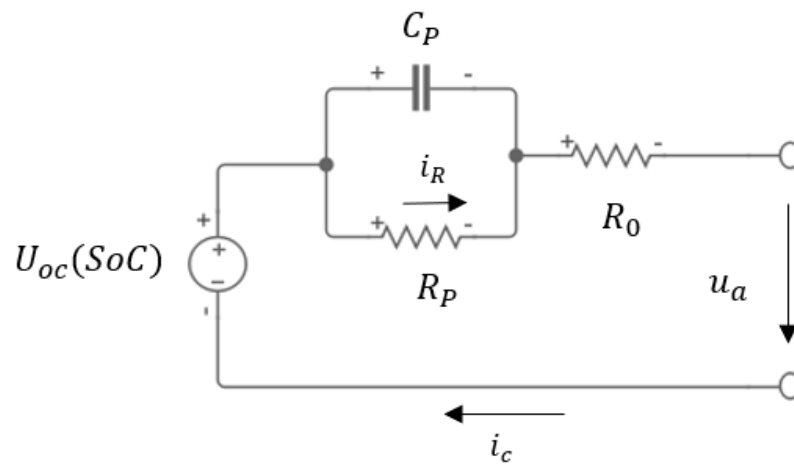
2.1 Komponenty systému

Pro popis mechatronického systému jako celku je nutné nejprve popsat jednotlivé modely komponent systému, kterými jsou v této práci jmenovitě baterie, DC motor, řízení systému a kolo, které je nezbytnou součástí aplikací pro mobilní robotiku.

2.1.1 Model baterie

Pro model baterie je vhodné zvolit některý z ekvivalentních elektrických modelů detailněji popsaných v kapitole 1.3 z důvodu kompatibility se zbytkem mechatronické soustavy a následných možností pro simulace v numerických řešičích.

Jako model baterie byl zvolen Theveninův model prvního řádu, který zanedbává závislost systému na teplotě a proměnném chování odporu baterie. Je však schopen dostatečně přesně popsat závislost napětí baterie na SoC a dynamické vlastnosti baterie. Jedná se o vhodný kompromis mezi přesným popisem chování baterie a komplexity výpočtů, které bude nutné integrovat do komplexního popisu celé soustavy. Schéma modelu je zobrazeno na Obr. 2.2



Obr. 2.2: Schéma modelu baterie

System lze popsat pomocí rovnic:

$$U_{oc} = \sum_{j=0}^k \alpha_j SoC^j \quad (2.1)$$

$$\frac{di_R}{dt} = \frac{1}{C_P R_P} i_c - \frac{1}{C_P R_P} i_R \quad (2.2)$$

$$\frac{dSoC}{dt} = - \left| \frac{i_c}{c_n} \right| \quad (2.3)$$

kde:

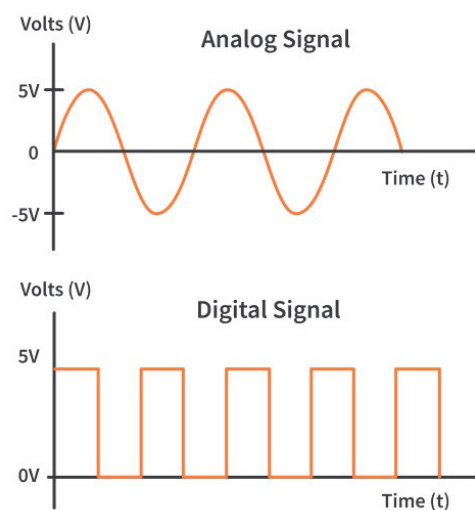
i_R	Proud v RC větvi obvodu procházející rezistorem
i_c	Celkový proud procházející obvodem
R_O	Interní odpor baterie
R_P	Odpor rezistoru v RC větvi
C_P	Kapacita kondenzátoru v RC větvi
C_n	Celková kapacita baterie
SoC	Stav nabití baterie
U_{OC}	Napětí baterie na prázdno - závislé na SoC

2.1.2 Řízení motoru

Jako řízení pro systém byla zvolena kombinace PWM a H-můstku, což umožňuje ovládat rychlost i směr otáčení motoru a zároveň umožňuje popsat vztah hodnot napětí na baterii a motoru.

PWM

PWM je běžným kontrolním prvkem v elektrických obvodech, obsahujících výkonovou elektroniku, který umožňuje převod mezi digitálním a analogickým signálem pomocí diskrétní modulační dvouhodnotovým signálem. Tento převod je znázorněn na příkladu napětí na Obr. 2.3.

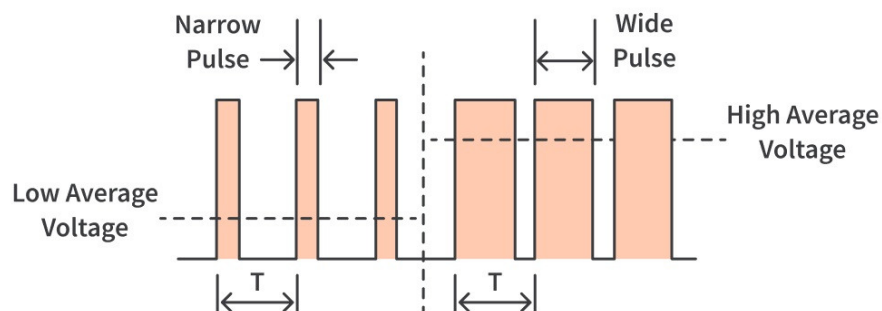


Obr. 2.3: Převod analogového signálu na digitální pomocí PWM [23]

Za použití logiky PWM z Obr. 2.4 lze definovat vztah:

$$p = \frac{t_{on}}{t_{on} + t_{off}} = \frac{t_{on}}{T} \quad (2.4)$$

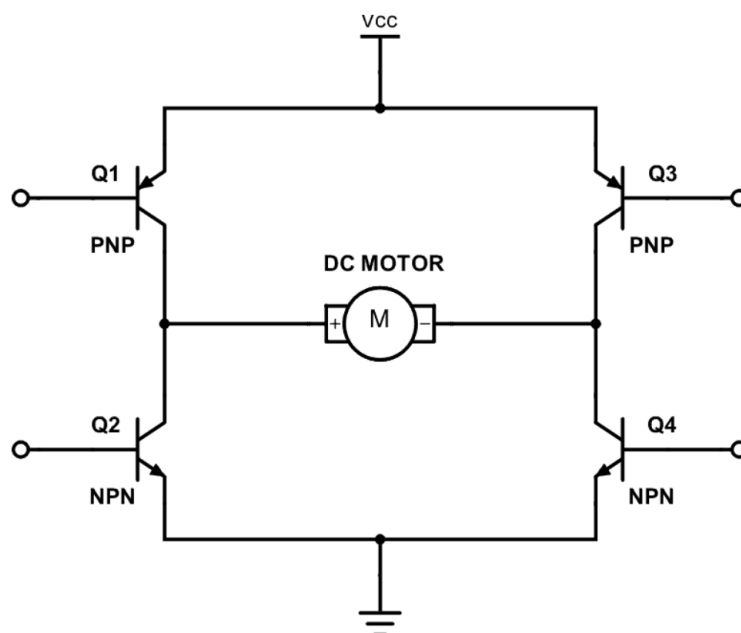
Tento vztah definuje parametr p , který vyjadřuje poměr sepnutí obvodu PWM, neboli "střídu", kde t_{on} vyjadřuje dobu sepnutí a t_{off} dobu vypnutí v periodě T PWM. Střída p tedy může nabývat hodnot pouze $\langle 0; 1 \rangle$ resp. 0-100%.



Obr. 2.4: Šířková modulace [23]

H-můstek

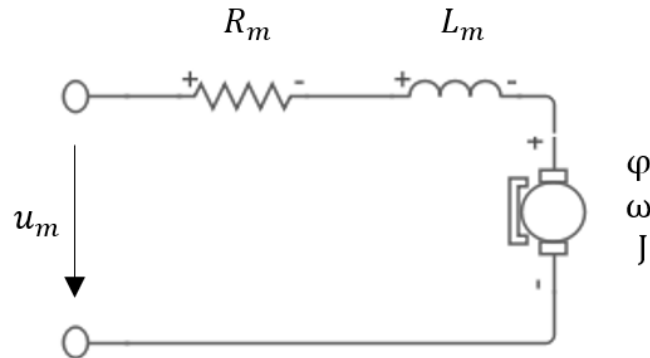
H-můstek je elektrický obvod sestavený ze 4 tranzistorů, které slouží pro řízení směru otáčení stejnosměrného motoru. Příklad zapojení tohoto obvodu je zobrazen na Obr. 2.5. Může dojít k sepnutí dvojic tranzistorů Q1 a Q4 nebo Q2 a Q3, dle sepnuté dvojice se motor otáčí v kladném nebo záporném směru. V kombinaci s H můstkem tedy může střída p nabývat hodnot $\langle -1; 1 \rangle$.



Obr. 2.5: H-můstek [24]

2.1.3 Model motoru

Jako model pro soustavu byl zvolen synchronní DC motor s permanentním magnetem. Parametrický model je obecně platný pro všechny motory tohoto typu, ale parametry se mohou násobně lišit dle zvoleného motoru a jeho užití. Obecné schéma je zobrazeno na Obr. 2.6.



Obr. 2.6: DC Motor

Systém lze popsat pomocí rovnic:

$$\frac{d\omega_m}{dt} = \frac{C_{ss}}{J_m} i_c - \frac{m_z(t)}{J_m} \quad (2.5)$$

$$\frac{di_c}{dt} = -\frac{C_{ss}}{L_m} \omega_m - \frac{R_m}{L_m} i_c + \frac{1}{L_m} u_m \quad (2.6)$$

kde:

ϕ	Úhel otočení motoru
ω	Úhlová rychlost motoru
J_m	Moment setrvačnosti motoru
C_{ss}	Konstanta motoru
R_m	Odpor kotvy motoru
L_m	Indukčnost kotvy motoru
u_m	Svorkové napětí na kotvě motoru

Rovnici (2.6) lze upravit pro užití v bateriovém systému pokud střihu p definujeme:

$$p = \frac{u_m}{u_a} \quad (2.7)$$

kde u_a má význam svorkového napětí baterie. Výsledný tvar pak lze zapsat:

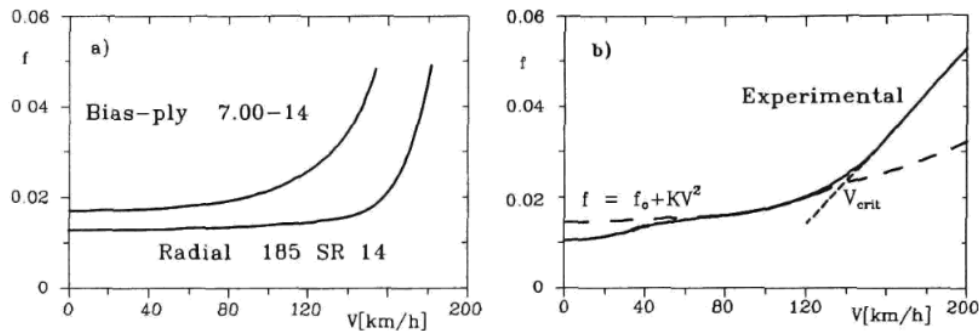
$$\frac{di_c}{dt} = -\frac{C_{ss}}{L_m} \omega_m - \frac{R_m}{L_m} i_c + \frac{1}{L_m} u_a p \quad (2.8)$$

2.1.4 Model kola

Model kola byl do popisu systému zařazen jako příklad proměnného zatížení motoru v čase. Jedná se o důležitý prvek modelu systému mobilních robotů, které jsou většinou poháněny bateriemi.

Při modelování kola se zohledňují různé aspekty, jako je geometrie kola, materiálové vlastnosti pneumatiky, valivý odpor a další faktory ovlivňující jízdní vlastnosti. Model pneumatiky je často založen na empirických rovnicích, které popisují vztahy mezi silou, valivým odporem a deformací pneumatiky. Mezi často používaný popis patří tzv. Pačejkova formule [25], který pneumatiku charakterizuje pomocí koeficientů pro každou důležitou sílu, která může vzniknout na kontaktní ploše. V této práci je kolo uvažováno pouze v 1D prostoru a je proto zaveden jednodušší popis, který má dvě hlavní vlastnosti.

První vlastností je zavedení závislosti koeficientu valivého odporu f na rychlosti pohybu v . Tato závislost také vychází z empirických vztahů pro každé kolo a konkrétní povrch, ukázkou závislosti lze vidět na Obr. 2.7. Obecně lze závislost uvažovat ve tvaru (2.11), pro systémy pohybující nízkou rychlostí lze koeficient považovat za konstantní.



Obr. 2.7: Závislost valivého odporu na rychlosti [26]

$$f = f_0 + Kv^2 \quad (2.9)$$

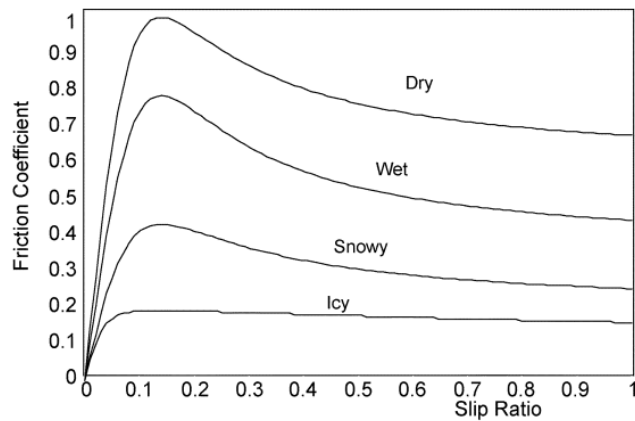
Druhou vlastností je prokluz kola, neboli "slip", který lze definovat:

$$s = \frac{r\omega - v}{\max(v, r\omega)} \quad (2.10)$$

V definici v představuje rychlost kola vůči podkladu, r poloměr kola a ω úhlovou rychlost kola. Z definice vyplývá, že pokud je $s < 0$ dochází k prokluzu při brzdění a pokud je $s > 0$ dochází k prokluzu při zrychlení.

Prokluz výrazně ovlivňuje přenos síly na povrch po kterém se robot pohybuje, což lze popsat pomocí závislosti zobrazené na grafu Obr. 2.8.

Pro většinu reálných aplikací není vhodné, aby kolo mělo shodné otáčky s motorem. Proto jsou robotické systémy vybaveny stálým převodem, který lze pomocí



Obr. 2.8: Závislost koeficientu tření na prokluzu [27]

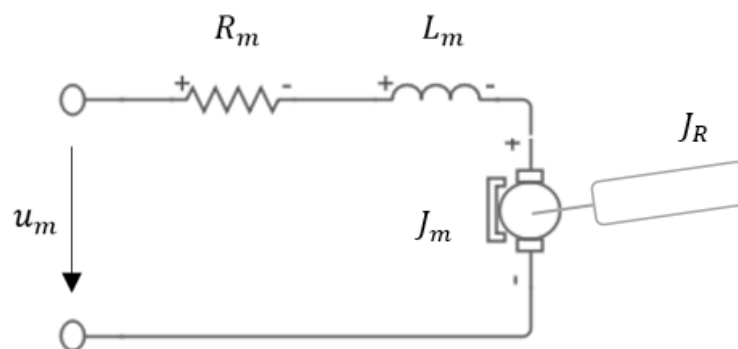
kinematiky definovat:

$$n_p = \frac{\omega_m}{\omega_k} \quad (2.11)$$

, kde ω_m představuje úhlovou rychlost motoru a ω_k úhlovou rychlost kola.

2.1.5 Model robotického ramene

Robotická ramena jsou základní součástí sériových robotů a v typické konstrukci je každé rameno vybaveno svojí převodovkou a poháněno jedním elektromotorem. Pro zjednodušení modelu simulovaného v této práci není zkoumáno chování celého robotu, je uvažována pouze jedna dvojice rameno - motor, tak jak je znázorněno na Obr. 2.9.



Obr. 2.9: Převod analogového signálu na digitální pomocí PWM [23]

Pro rameno jsou uvažovány vlastnosti dokonale tuhé tyče o hmotnosti m a délce l , platí tedy rovnice pro moment setrvačnosti $J_R = \frac{1}{12}ml^2$.

2.2 Model systému s robotickým ramenem

V předchozí kapitole byly popsány jednotlivé komponenty systému baterie - DC motor pomocí rovnic (2.2), (2.3), (2.4), (2.5) a (2.8). Propojením těchto rovnic lze vytvořit popis mechatronického systému, znázorněného na Obr. 2.1, popsaného rovnicí v maticovém tvaru (2.12), který lze řídit pomocí střídy p . Tento popis je obecně platný pro všechny systémy baterie - DC synchronní motor a dle volby aplikace je určen průběh momentu zatížení.

$$\begin{aligned}
 & \begin{bmatrix} 0 \\ \frac{m_z(t)}{J_m} \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
 & \quad \text{p -} \\
 & \begin{bmatrix} 0 \\ 0 \\ -\frac{R_0}{L_m}i_c - \frac{R_P}{L_m}i_R + \sum_{j=0}^k \frac{\alpha_j}{L_m} SoC^j \\ 0 \\ 0 \end{bmatrix} \\
 & \quad + \\
 & \begin{bmatrix} \phi \\ \omega \\ i_c \\ i_R \\ SoC \end{bmatrix} \\
 & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{C_P}{R_P} & 0 \\ 0 & \frac{C_{ss}}{J_m} & -\frac{R_m}{L_m} & \frac{C_P}{R_P} & -\frac{1}{C_n} \\ 1 & 0 & -\frac{C_{ss}}{L_m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
 & \quad = \\
 & \begin{bmatrix} \dot{\phi} \\ \dot{\omega} \\ \dot{i}_c \\ \dot{i}_R \\ \dot{SoC} \end{bmatrix}
 \end{aligned} \tag{2.12}$$

Tento obecný popis je v této kapitole dále upraven do tvaru stavového popisu pro dva konkrétní příklady. Rozdíl mezi těmito příklady spočívá v zatížení motoru, kde v prvním příkladu se jedná o motor zatížený robotickým ramenem a ve druhém vzniká moment zatížení připojením kola k soustavě. Vzniklé nelineární popisy jsou následně linearizovány do podoby vhodné pro výpočet optimálního řízení.

2.2.1 Stavový popis systému s robotickým ramenem

Případem takového systému je motor s robotickým ramenem poháněný baterií, pro který platí $m_z = J_R x_3$ a lze ho přepsat jako stavový popis ve tvaru:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}(\mathbf{x})u \\ \mathbf{y} &= \mathbf{C}\mathbf{x}\end{aligned}$$

kde:

$$\begin{aligned}\mathbf{x} &= [\phi \quad \omega \quad i_c \quad i_R \quad SoC]^T \\ \mathbf{u} &= [p]\end{aligned}$$

Pro přehlednost zápisu jsou také zavedeny koeficienty a_{ij} , kde i odpovídá \dot{x}_i a index j se váže k násobku x_j

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & 0 \\ 0 & -a_{32} & -a_{33} & 0 & 0 \\ 0 & 0 & a_{43} & -a_{44} & 0 \\ 0 & 0 & -a_{53} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -c_{1u}x_3 - c_{2u}x_4 + \sum_{j=0}^{k=7} \alpha_j x_5^{k-j} \\ 0 \\ 0 \end{bmatrix} u \quad (2.13)$$

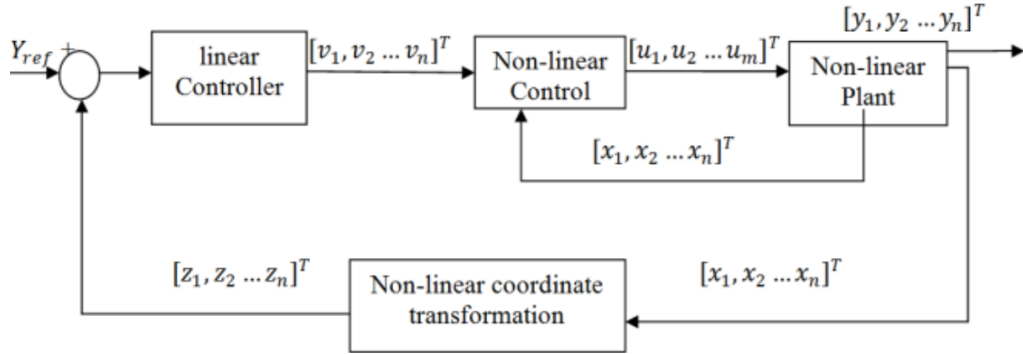
2.2.2 Přesná linearizace systému s robotickým ramenem

Pro zjednodušení dalších kroků při analýze systému je vhodné systém (2.13) linearizovat. Pro linearizaci byla zvolena metoda přesné vstupně-výstupní linearizace.

Pro metodu obecně platí:

$$u = \frac{1}{g(x)} [-f(x) + v] \quad (2.14)$$

, kde v představuje nový vstup systému, který slouží k nahrazení původního vstupu u . S tím souvisí transformace souřadnic \mathbf{x} na souřadnice \mathbf{z} . Vztah mezi novým a starým stavovým popisem je znázorněn Obr.2.10.



Obr. 2.10: Transformace souřadnic a řízení

Dle zvolené metody lze upravit stavový popis systému (2.13) do tvaru:

$$z_0 = x_1 \quad (2.15)$$

$$\dot{z}_0 = z_1 = x_2 \quad (2.16)$$

$$\dot{z}_1 = z_2 = a_{23}x_3 \quad (2.17)$$

$$\dot{z}_2 = a_{23}(-a_{32}x_2 - a_{33}x_3 + a_{3u}(-c_{1u}x_3 - c_{2u}x_4 + \sum_{j=0}^k \alpha_j x_5^j)u) = v \quad (2.18)$$

$$\dot{z}_3 = \dot{x}_4 = \frac{a_{43}}{a_{23}}z_2 - a_{44}z_3 \quad (2.19)$$

$$\dot{z}_4 = \dot{x}_5 = -\frac{a_{53}}{a_{23}}z_2 \quad (2.20)$$

Proměnné z_3 a z_4 lze považovat za zero-dynamics pro systém, protože nemají vliv na dynamiku soustavy. Jsou ovšem potřebné k dopočítání původního vstupu pro nelineární stavový popis u a určení omezení pro řízení v .

Linearizovaný stavový popis lze zapsat ve tvaru:

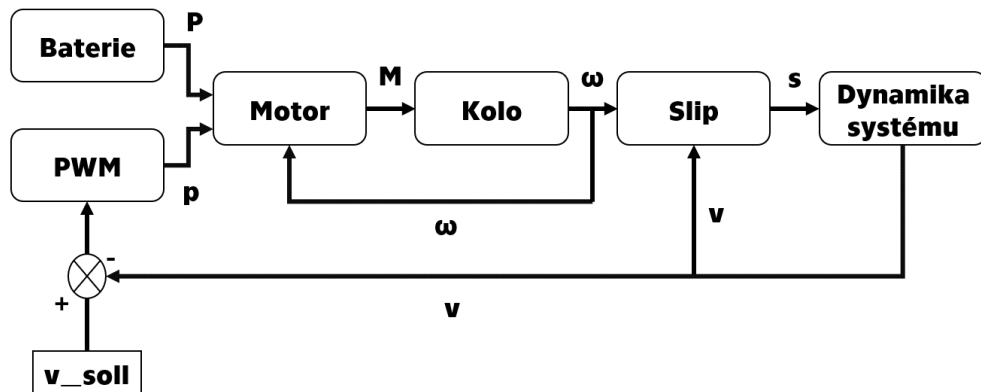
$$\begin{bmatrix} \dot{z}_0 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{a_{43}}{a_{23}} & -a_{44} & 0 \\ 0 & 0 & -\frac{a_{53}}{a_{23}} & 0 & 0 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} v \quad (2.21)$$

2.3 Model systému mobilního robota

Model v této kapitole vychází z obecného popisu (2.12), ale na rozdíl od předchozí kapitoly je zde uvažováno proměnné zatížení motoru. Zatížení vychází z popisu pohonu jednoho kola, kde je uvažován valivý odpor, ale jsou zanedbány vlastnosti jako např. odpor vzduchu a vliv gravitačního zrychlení při jízdě v terénu. Moment zatížení tedy lze popsat:

$$m_z = \frac{J_k}{n_p} \omega_m + KNr\omega_m^2 + f_0N \quad (2.22)$$

Pro mobilní roboty obecně platí schéma znázorněné na Obr. 2.11, kde jsou popsány vztahy mezi jednotlivými komponenty a jejich vlastnostmi.



Obr. 2.11: Schéma systému mobilního robota

Po dosazení rovnice (2.22) do matice (2.12) lze přepsat do obecného tvaru:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u \\ \mathbf{y} &= \mathbf{C}\mathbf{x}\end{aligned}$$

kde:

$$\begin{aligned}\mathbf{x} &= [\phi \quad \omega \quad i_c \quad i_R \quad SoC]^T \\ \mathbf{u} &= [p]\end{aligned}$$

Pro přehlednost zápisu jsou také zavedeny koeficienty a_{ij} , kde i odpovídá \dot{x}_i a index j se váže k násobku x_j .

System lze popsat pomocí následujících pěti diferenciálních rovnic:

$$\dot{x}_1 = x_2 \tag{2.23}$$

$$\dot{x}_2 = -a_{22}x_2^2 + a_{23}x_3 - a_{2c} \tag{2.24}$$

$$\dot{x}_3 = -a_{32}x_2 - a_{33}x_3 + a_{3u}(-c_{1u}x_3 - c_{2u}x_4 + \sum_{j=0}^{k=7} \alpha_j x_5^{k-j})u \tag{2.25}$$

$$\dot{x}_4 = a_{43}x_3 - a_{44}x_4 \tag{2.26}$$

$$\dot{x}_5 = -a_{53}|x_3| \tag{2.27}$$

2.3.1 Přesná linearizace modelu mobilního robota

Stejně jako v předchozí kapitole je systém linearizován za použití metody přesné vstupně-výstupní linearizace pomocí definice nového vstupu v dle (2.14). Rozdíl oproti soustavě s robotickým ramenem spočívá ve vztahu mezi novým vstupem v a původním vstupem u , který je získán upravením rovnic (2.23-2.27) do tvaru:

$$\dot{z}_0 = x_2 = z_1 \tag{2.28}$$

$$\dot{z}_1 = -a_{22}z_1^2 + a_{23}z_2 - a_{2c} = z_2 \tag{2.29}$$

$$\dot{z}_2 = -2a_{22}x_2(-a_{22}x_2^2 + a_{23}x_3 - a_{2c}) + a_{23}(g(x)u) = v \quad (2.30)$$

kde:

$$g(x) = -c_{1u}x_3 - c_{2u}x_4 + \sum_{j=0}^k \alpha_j x_5^j$$

$$\dot{z}_3 = \dot{x}_4 = \frac{a_{43}}{a_{23}}z_2 - a_{44}z_3 \quad (2.31)$$

$$\dot{z}_4 = \dot{x}_5 = -\frac{a_{53}}{a_{23}}z_2 \quad (2.32)$$

Linearizovaný popis lze zapsat stejně jako v předchozí kapitole ve tvaru:

$$\begin{bmatrix} \dot{z}_0 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{a_{43}}{a_{23}} & -a_{44} & 0 \\ 0 & 0 & -\frac{a_{53}}{a_{23}} & 0 & 0 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} v \quad (2.33)$$

Kapitola 3

Optimální řízení

Optimální řízení můžeme chápat jako proces, který má za cíl maximalizovat požadované výstupy nebo minimalizovat náklady, přičemž jsou respektovány omezení a podmínky daného systému. Tento přístup hraje zásadní roli v různých oblastech, od průmyslu a výroby přes energetiku, dopravu, až po řízení komplexních technologií, jako jsou právě baterie.

V kontextu diplomové práce zaměřené na návrh optimálního řízení baterií je hlavním cílem nalézt strategie a metody, které umožní co nejefektivnější využití energie uložené v bateriích. Důležité je zmínit, že tato diplomová práce se věnuje návrhu optimálního řízení pouze z krátkodobého hlediska a neuvažuje vliv na zdraví baterie a její celkovou životnost.

Cílem této kapitoly je nalézt optimální strategii řízení robota poháněného baterií, která umožní minimalizaci spotřeby energie robotického systému. Ke splnění tohoto cíle je nutné nejprve definovat kritérium optimality a problém řízení včetně všech jeho omezení a podmínek, této části se věnuje první část této kapitoly. Druhá část kapitoly je věnována popisu návrhu optimálního řízení aplikováním Pontryaginova principu minima (PMP). Třetí část je věnována sestavení numerických řešičů, které umožní porovnání výsledků s výsledky zjištěnými pomocí PMP.

3.1 Definice problému řízení

Optimální řízení pro vyřešení cíle této diplomové práce a nalezení energeticky optimálního řízení, lze úlohu formulovat jako omezený optimalizační problém, ve kterém je spotřebovaná energie, neboli příkon systému P , minimalizována v rámci daného pohybu za dodržení dynamických a statických omezení, matematicky vyjádřeno:

$$J = \int_0^{t_f} P(t, x, u) dt \quad (3.1)$$

$$i_{cmin} \leq x_3 \leq i_{cmax} \quad (3.2)$$

$$SoC_{min} \leq x_5 \leq SoC_{max} \quad (3.3)$$

$$-1 \leq u(t) \leq 1 \quad (3.4)$$

kde:

t	Čas
t_f	Celkový čas pohybu
SoC_{min}	Minimální stav nabití baterie
SoC_{max}	Maximální stav nabití baterie
i_{cmin}	Minimální proud procházející soustavou
i_{cmax}	Maximální proud procházející soustavou

3.1.1 Kritérium optimality pro nelineární systém

Pro nalezení optimálního řízení systému popsaného v kapitole 2.2.1 je nutné upravit kritérium optimality do tvaru obsahujícího stavu \mathbf{x} a řízení \mathbf{u} .

Fyzikálně se jedná o popsání výkonu nutného k pohonu systému, tedy:

$$P = P_M + P_E = u_m i_c + P_E = x_3(-c_{1u}x_3 - c_{2u}x_4 + \sum_{j=0}^{k=7} \alpha_j x_5^{k-j})u + P_E \quad (3.5)$$

kde:

P_M	Příkon motoru
P_E	Příkon elektroniky v systému - uvažován konstantní
u_m	Napětí na kotvě motoru
i_c	Proud procházející obvodem

Nové kritérium optimality, tedy lze formulovat ve tvaru:

$$J = \int_0^t \left(\left| x_3(-c_{1u}x_3 - c_{2u}x_4 + \sum_{j=0}^{k=7} \alpha_j x_5^{k-j})u \right| + P_E \right) dt \quad (3.6)$$

Absolutní hodnota příkonu je zavedena z důvodu neschopnosti rekuperace energie vzniklé při brzdění. Systém zároveň není vybaven mechanickou brzdou a k zastavení je nutné použít energii z baterie pro pohon motoru nebo indukované napětí na motoru.

3.1.2 Kritérium optimality pro lineární systém

Pro nalezení optimálního řízení systému popsaného v kapitole 2.2.2 je nutné upravit kritérium optimality do tvaru obsahujícího stav \mathbf{z} a řízení \mathbf{v} . Jelikož se jedná řízení pomocí vstupu v je tento popis závislý na zvoleném systému a momentu zatížení motoru. Tohoto tvaru je možné dosáhnout dosažením rovnic (2.17) a (2.18) do vztahu (3.6), kde po numerických úpravách vzniká:

$$J = \int_0^t \left(\left| \frac{v + a_{23}a_{32}z_1 + a_{33}z_2}{a_{23}^2 a_{3u}} z_2 \right| + P_E \right) dt \quad (3.7)$$

3.2 Optimální řízení dle PMP

Pontryaginův princip minima (PMP) [[28],[29]] je používaným nástrojem v teorii optimálního řízení, který poskytuje soubor nezbytných podmínek pro řešení omezeného problému optimálního řízení, tak aby řešení bylo globálně optimální.

V této kapitole jsou tyto podmínky definovány a následně aplikovány na systém popsaný v kapitole 2.2.2, tak aby bylo nalezeno optimální řešení dle kritéria optimality (3.7).

3.2.1 Podmínky optimálního řízení dle PMP

Optimální řízení je takové přípustné řízení u , které zajistí dosažení optimálního průběhu systému $\dot{x} = f(x(t), u(t), t)$ podle kritéria optimality:

$$J = \psi(x(t_f), t_f) + \int_0^{t_f} \mathcal{L}(x(t), u(t), t) dt \quad (3.8)$$

, tak aby optimální průběh stavů x byl přípustný. $\psi(x(t_f), t_f)$ představuje hodnotu nákladů v době ukončení.

Metoda PMP vyžaduje také definici funkce \mathcal{H} , nazývané Hamiltonián:

$$\mathcal{H}(x(t), u(t), p(t), t) = \mathcal{L}(x(t), u(t), t) + p^T(t)f(x(t), u(t), t) \quad (3.9)$$

, kde \mathbf{p} představuje vektor optimalizačních parametrů, tzv. co-stavů významově odpovídající Lagrangeový parametrům λ .

Při použití tohoto zápisu lze optimální řízení $u^*(t)$ vyjádřit:

$$u^*(t) = \operatorname{argmin}_{u \in \mathcal{U}} (\mathcal{H}(x(t), u(t), p(t), t)) \quad (3.10)$$

, kde \mathcal{U} představuje množinu přípustných řízení.

Nezbytné podmínky pro optimální řízení jsou:

$$\mathcal{H}(x^*(t), u^*(t), p^*(t), t) \leq \mathcal{H}(x(t), u(t), p(t), t) \quad (3.11)$$

$$\psi(x^*(t_f), t_f) = 0 \quad (3.12)$$

$$\dot{x}^* = \frac{\partial \mathcal{H}}{\partial p} \quad (3.13)$$

$$\dot{p}^* = -\frac{\partial \mathcal{H}}{\partial x} \quad (3.14)$$

Řešení diferenciálních rovnic (3.13) a (3.14) závisí na okrajových podmínkách stavových co-stavových vektorů. Tyto podmínky závisí na znalosti celého průběhu optimalizace a to není u reálných aplikací vždy možné, u těchto případů tedy nalezené optimální řešení nemusí být globální.

3.2.2 Aplikace PMP

V této kapitole je aplikován PMP na vyřešení úlohy optimálního řízení lineárního systému:

$$\begin{bmatrix} \dot{z}_0 \\ \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{a_{43}}{a_{23}} & -a_{44} & 0 \\ 0 & 0 & -\frac{a_{53}}{a_{23}} & 0 & 0 \end{bmatrix} \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} v \quad (3.15)$$

, kde jsou definovány počáteční a okrajové podmínky:

$$\begin{aligned} z_0(0) &= 0 & z_0(t_f) &= \phi_f \\ z_1(0) &= 0 & z_1(t_f) &= 0 \\ z_2(0) &= 0 & z_2(t_f) &= 0 \\ z_3(0) &= 0 & z_4(0) &= SoC_0 \end{aligned} \quad (3.16)$$

Dodržení těchto podmínek je zajištěno dosazením do rovnice (3.8) ve tvaru:

$$\psi(z(t_f), t_f) = |z_0(t_f) - \phi_f| + |z_1(t_f)| + |z_2(t_f)| = 0 \quad (3.17)$$

Obdobná podmínka platí, i jedná-li se o úlohu s pevně stanoveným časem průběhu:

$$\psi(z(t_f), t_f) = t_f - a = 0 \quad (3.18)$$

, kde a je definovaná konstanta rovná době průběhu.

Jedná se o úlohu s omezeným vstupem, který je závislý na stavech systému. Omezení vychází z definice vstupu v v rovnici (2.18), kde horní mez je rovna průběhu se vstupem $u = 1$ a dolní mez je rovna průběhu se vstupem $u = -1$.

$$v_{max} = a_{23}(-a_{32}z_1 - \frac{a_{33}}{a_{23}}z_2 + a_{3u}(-\frac{c_{1u}}{a_{23}}z_2 - c_{2u}z_3 + \sum_{j=0}^{k=7} \alpha_j z_4^{k-j})) \quad (3.19)$$

$$v_{min} = a_{23}(-a_{32}z_1 - \frac{a_{33}}{a_{23}}z_2 - a_{3u}(-\frac{c_{1u}}{a_{23}}z_2 - c_{2u}z_3 + \sum_{j=0}^{k=7} \alpha_j z_4^{k-j})) \quad (3.20)$$

Důležitou roli hraje také hodnota vstupu v_0 , která popisuje chování systému při odpojení motoru od napájení. Matematicky vyjádřeno:

$$v_0 = a_{23}(-a_{32}z_1 - \frac{a_{33}}{a_{23}}z_2) \quad (3.21)$$

Hamiltonián systému je popsán rovnicí:

$$\mathcal{H} = L + p_1 z_1 + p_2 z_2 + p_3 v + p_4 \dot{z}_3 + p_5 \dot{z}_4 \quad (3.22)$$

kde:

$$\mathcal{L} = \left| \frac{v + a_{23}a_{32}z_1 + a_{33}z_2}{a_{23}^2 a_{3u}} z_2 \right| \quad (3.23)$$

Co-stavové diferenciální rovnice jsou odvozeny z obecného vztahu (3.14) a vyjádřeny ve tvaru:

$$\dot{p}_1 = -\frac{\partial \mathcal{H}}{\partial z_0} = 0 \quad (3.24)$$

$$\dot{p}_2 = -\frac{\partial \mathcal{H}}{\partial z_1} = -\frac{a_{32}}{a_{3u}a_{23}}|z_2| - p_1 \quad (3.25)$$

$$\dot{p}_3 = -\frac{\partial \mathcal{H}}{\partial z_2} = -\left| \frac{v + a_{23}a_{32}z_1 + 2a_{33}z_2}{a_{23}^2 a_{3u}} \right| - p_2 - \frac{a_{43}}{a_{23}}p_4 + a_{53}p_5 \quad (3.26)$$

$$\dot{p}_4 = -\frac{\partial \mathcal{H}}{\partial z_3} = -a_{44}p_4 \quad (3.27)$$

$$\dot{p}_5 = -\frac{\partial \mathcal{H}}{\partial z_4} = 0 \quad (3.28)$$

Optimální řízení je odvozeno z rovnice:

$$\frac{\partial \mathcal{H}}{\partial v} = \frac{\partial L}{\partial v} + p_3 = 0 \quad (3.29)$$

Tato podmínka není pro tento systém dostatečná a výsledkem optimalizační úlohy je tzv. Bang-bang řízení, které je určeno podmínkou:

$$\left| \frac{z_2}{a_{23}^2 a_{3u}} \right| + p_3 \leq 0 \quad (3.30)$$

Tato podmínka vede na 3 možné stavy řízení, které významově představují situace:

1. Motor je plně poháněn v kladném směru - $v = v_{max}$

S podmínkou:

$$-\frac{z_2}{a_{23}^2 a_{3u}} \leq p_3 \quad (3.31)$$

2. Motor je odpojen od napájení - $v = v_0$

S podmínkou:

$$\frac{z_2}{a_{23}^2 a_{3u}} \geq p_3 \geq -\frac{z_2}{a_{23}^2 a_{3u}} \quad (3.32)$$

3. Motor je plně poháněn v záporném směru - $v = v_{min}$

S podmínkou:

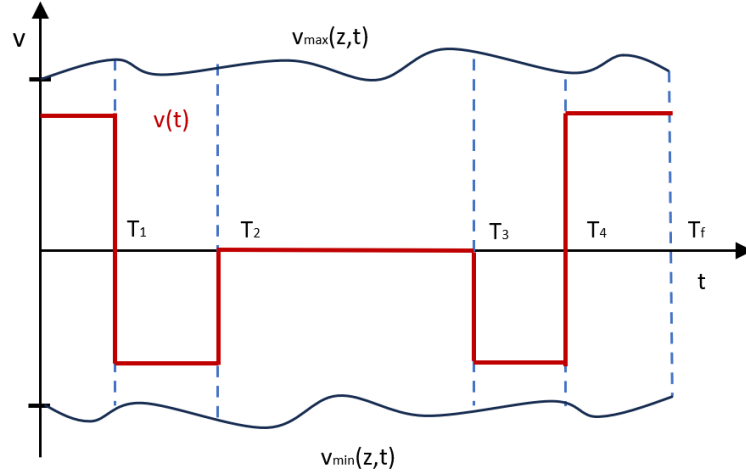
$$\frac{z_2}{a_{23}^2 a_{3u}} \geq p_3 \quad (3.33)$$

Optimální řízení je získáno vyřešením soustavy deseti diferenciálních rovnic (3.15), (3.24) - (3.28), jejich počátečních a okrajových podmínek (3.44) a podmínek pro Bang-bang řízení (3.31) - (3.33).

Z formulace vztahů pro nalezení optimálního řízení vyplývá, že optimalizační parametry \mathbf{p} tzv. co-stavy jsou z hlediska nalezení optimálního řízení systému zásadní a jejich počáteční hodnota \mathbf{p}_0 i koncová hodnota \mathbf{p}_{tf} ovlivní celý průběh optimalizace.

3.2.3 Analytické řešení

Ze znalosti charakteru řízení, tedy bang-bang, je možné sestavit analytický odhad řešení, který splní počáteční a okrajové podmínky (3.44). Vzhledem k charakteru vstupu v , jehož limity jsou závislé na stavech \mathbf{z} (viz. Obr. 3.1), se jedná pouze o přípustné řešení, které lze dále použít jako vstup pro numerickou iteraci pro nalezení globálního minima.



Obr. 3.1: Analytické řešení

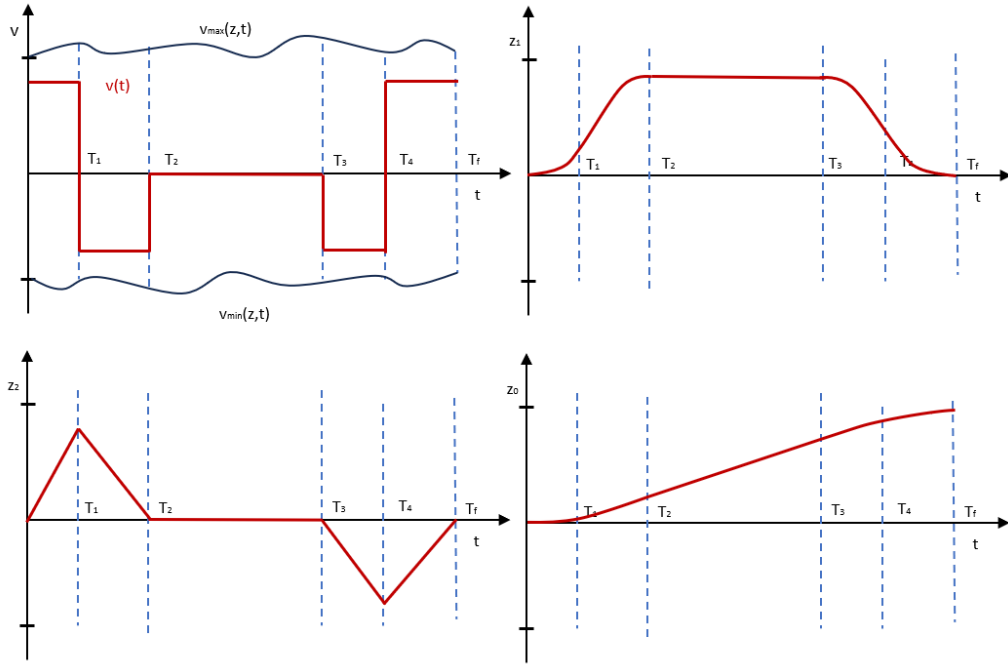
Za použití značení a průběhu vstupu v z Obr. 3.1 lze analyticky dopočítat průběhy z_0 , z_1 a z_2 rozdělením řešení do pěti intervalů, kde platí:

$$\begin{aligned}
 v(t) &= \nu_H & t \in \langle 0, T_1 \rangle \\
 v(t) &= \nu_D & t \in \langle T_1, T_2 \rangle \\
 v(t) &= 0 & t \in \langle T_2, T_3 \rangle \\
 v(t) &= \nu_D & t \in \langle T_3, T_4 \rangle \\
 v(t) &= \nu_H & t \in \langle T_4, T_f \rangle \\
 \nu_H &= a\nu_{max}(0) & \nu_D = b\nu_{min}(0) \\
 a &\leq 1 & 0 \leq b
 \end{aligned} \tag{3.34}$$

Výsledkem analytického výpočtu jsou průběhy \mathbf{z} , které jsou zobrazeny v grafu 3.2 a podmínky (3.35) - (3.37).

$$z_2(t_f) = \nu_{max} \cdot (T_1 - T_4 + t_f) + \nu_{min} \cdot (-T_1 + T_2 - T_3 + T_4) = 0 \tag{3.35}$$

$$z_1(t_f) = \int_0^{T_1} z_2 dt + \int_{T_1}^{T_2} z_2 dt + \int_{T_2}^{T_3} z_2 dt + \int_{T_3}^{T_4} z_2 dt + \int_{T_4}^{t_f} z_2 dt = 0 \tag{3.36}$$



Obr. 3.2: Analytické řešení

$$z_0(t_f) = \int_0^{T_1} z_1 dt + \int_{T_1}^{T_2} z_1 dt + \int_{T_2}^{T_3} z_1 dt + \int_{T_3}^{T_4} z_1 dt + \int_{T_4}^{t_f} z_1 dt = \phi_f \quad (3.37)$$

Na základě těchto podmínek je možné nalézt přípustné řízení pomocí kombinace parametrů $[\nu_{max}, \nu_{min}, T_1, T_2, T_3, T_4, t_f]$. Je ovšem nutné brát v potaz podmínku dosažitelnosti $t_f \geq T_{crit}$, kde T_{crit} je nejkratší možná doba pohybu, za kterou je úlohu možné vykonat, tak aby byly splněny okrajové podmínky.

3.3 Numerická řešení optimálního řízení

Jak vyplývá ze závěrů předchozí kapitoly, tak numerické řešení je pro nalezení optimálního řešení této úlohy nezbytné. Existují však rozdíly mezi numerickým a analytickým řešením, které je potřeba nejdříve vyjasnit. V první části této kapitoly je proto popsána metoda převodu úlohy s volným koncem do tvaru vhodného pro numerické řešení. Následně jsou navrženy algoritmy pro nalezení optimálního řešení pomocí numerických metod.

3.3.1 Transformace systému pro numerické řešení T_{free}

Pro numerické metody jsou běžnou formou problému úlohy s pevným časem ukončení a problém volného času T_{free} řešíme tak, že úlohu převedeme na problém s pevným časem. Zadat časový interval je nutné z důvodu vytvoření sítě diskrétních bodů v řešiči, tak aby bylo možné splnit okrajové podmínky diferenciálních rovnic.

Převod z volného času na pevný je řešen pomocí definice parametru τ , který je definován:

$$\tau = \frac{t}{t_f} \quad (3.38)$$

, kde $\tau \in \langle 0, 1 \rangle$, t představuje čas a t_f finální čas.

Tento parametr je použit k transformaci systému do tvaru:

$$\dot{\mathbf{z}} = f(z, v, t) \leftrightarrow \dot{\mathbf{z}} = t_f \cdot f(z, v, \tau) \quad (3.39)$$

Je také sestavena nová diferenciální rovnice jejíž průběh může být numerickou metodou optimalizován:

$$\dot{t}_f = 0 \quad (3.40)$$

3.3.2 Numerické řešení PMP

Numerické řešení vychází z diferenciálních rovnic sestavených v kapitole 3.2.2 a implementuje je do numerického řešiče, který pomocí iterací hledá optimální řízení systému na základě optimalizační parametrů \mathbf{p} .

Pro iterační proces byl sestaven algoritmus:

1. Vytvoření odhadu řízení $v^{(0)}$ pomocí rozdělení intervalu $[t_0, t_f]$ na intervaly a definováním konstantní hodnoty $v^{(0)}$ pro každý interval, tak aby byly splněny okrajové podmínky řešení.
2. Aplikace odhadovaného řízení $v^{(i)}$ na systém $\dot{\mathbf{z}} = t_f \cdot f(z, v, \tau)$ s počátečními podmínkami $z(t_0) = z_0$.
3. Uložení vypočítaných průběhů $z^{(i)}$
4. Ze znalosti $v^{(i)}$ a $z^{(i)}$ je dopočítán vektor $p^{(i)}$, který je dle PMP potřebných nalezení optimálního řešení
5. Vyhodnotit, jestli $\left| \frac{\partial H}{\partial v} \right| \leq \gamma$, kde γ je definované okolí nuly, pro Bang-bang však nemusí nastat.
6. Pokud není podmínka splněna je nutné upravit řízení: $v^{(i+1)} = v^{(i)} - \alpha \frac{\partial H}{\partial v}$, u Bang-bang řízení je řízení upraveno dle podmínek (3.31)-(3.33) a vrátit se zpět ke kroku 2.

Toto řešení je ovšem velmi náchylné na numerické chyby a citlivé na zvolené počáteční podmínky co-stavových parametrů p . Sestavený algoritmus je přiložen v Příloze 1.

3.3.3 Numerické řešení analytického odhadu

Pro výpočet přípustného suboptimálního řízení na základě analytického odhadu jsou v této práci použity genetické algoritmy, které umožňují nalézt nekonečné množství řešení úlohy pomocí vhodných parametrů $[\nu_H, \nu_D, T_1, T_2, T_3, T_4, t_f]$, tak aby byly splněny počáteční a okrajové podmínky a vztahy z (3.44). Genetické algoritmy jsou aplikovány pomocí funkce $ga()$ z knihovny "Global Optimization Toolbox" SW Matlab, která umožňuje dle informací z dokumentace [30] nalézt lokální minimum.

Tyto řešení lze následně lineárně kombinovat k nalezení optimálního řízení a za použití rovnic (3.41), (3.42) z [31] a připuštění i těch řízení, která nesplňují omezení řízení, ale snižují kritérium optimality J , lze nalézt optimální řízení jejich kombinací.

$$v^{(i+1)}(t) = (1 - \alpha)v^{(i)}(t) + \alpha v^{(i^*)}(t) \quad (3.41)$$

$$z^{(i+1)}(t) = (1 - \alpha)z^{(i)}(t) + \alpha z^{(i^*)}(t) \quad (3.42)$$

Kde i značí počet vykonaných iterací, $v^{(i+1)}(t)$ je hodnota řízení při iteraci i , $v^{(i^*)}(t)$ je řízení, které nepodléhá omezení a α je koeficient v rozmezí $\langle 0; 1 \rangle$, který podléhá jednokriteriální optimalizaci. Pro iterační proces byl sestaven algoritmus:

1. Definování počtu a omezení optimalizovaných parametrů $[\nu_H, \nu_D, T_i, t_f]$ a lineárních i nelineárních vztahů mezi nimi.
2. Spuštění $ga()$ a odhad parametrů
3. Minimalizace hodnoty fitness funkce, která je definována rovnicí:

$$f = a(z_0(t_f) - \phi_F)^2 + b(z_1(t_f))^2 + c(z_2(t_f))^2 + d \cdot J \quad (3.43)$$

, kde J je hodnota optimalizačního kritéria dle (3.7), P přidává penalizaci za nedodržení podmínky $v_{min}(t) \leq v(t) \leq v_{max}(t)$. Tyto hodnoty vyžadují výpočet průběhu stavů systému pro každý návrh optimalizovaných parametrů. Parametry a, b, c, d jsou volitelné váhy jednotlivých kritérií.

4. Uložení vypočítaných suboptimálních průběhů $z^{(i)}$ a hodnoty kritéria optimality J^i .
5. Je spuštěna jednokriteriální optimalizace α , tak aby platilo $J(i+1) < J(i)$, pro $v^{(i+1)}$, definované (3.41).

Sestavený algoritmus pro nalezení přípustných suboptimálních řešení úlohy s definovaným t_f , která jsou dále zkoumána v kapitole 4, je přiložen v Příloze 2.

3.3.4 Numerické řešení pomocí přímkové aproximace řízení

V této podkapitole je popsáno numerické řešení optimálního řízení pomocí přímkové aproximace řízení, kde jsou použity genetické algoritmy v SW Matlab pro nalezení parametrů $[\nu_{H_1}, \nu_{D_1}, \nu_{H_2}, \nu_{D_2}, \nu_0, s_{H_1}, s_{D_1}, s_{H_2}, s_{D_2}, \nu_0, T_1, T_2, T_3, T_4, t_f]$, tak aby byly splněny počáteční a okrajové podmínky a vztahy:

$$\begin{aligned}
 v(t) &= \nu_{H_1} + s_{H_1} \cdot t & t \in \langle 0, T_1 \rangle \\
 v(t) &= \nu_{D_1} + s_{D_1} \cdot t & t \in \langle T_1, T_2 \rangle \\
 v(t) &= \nu_0 + s_0 \cdot t & t \in \langle T_2, T_3 \rangle \\
 v(t) &= \nu_{D_2} + s_{D_2} \cdot t & t \in \langle T_3, T_4 \rangle \\
 v(t) &= \nu_{H_2} + s_{H_2} \cdot t & t \in \langle T_4, T_f \rangle
 \end{aligned} \tag{3.44}$$

Tato metoda optimalizace zavedena z důvodu nelinearit vyskytujících se v omezeních přípustných řízení v , které byly zjištěny pomocí analytického řešení. Přímková aproximace by měla umožnit se více přiblížit charakteru bang-bang řízení a přímky sloužit jako nahrazení limitních křivek. Optimální parametry řízení jsou nalezeny pomocí aplikace algoritmu:

1. Definování počtu a omezení optimalizovaných parametrů dle uvažovaných závislostí $[\nu_{H_1}, \nu_{D_1}, \nu_{H_2}, \nu_{D_2}, \nu_0, s_{H_1}, s_{D_1}, s_{H_2}, s_{D_2}, \nu_0, T_1, T_2, T_3, T_4, t_f]$ a lineárních i nelineárních vztahů mezi nimi.
2. Nastavení volných parametrů GA, určením počtu jedinců v generaci, zachování TOP jedinců pro další křížení.
3. Spuštění $ga()$ a odhad parametrů
4. Minimalizace hodnoty fitness funkce, která je definována rovnicí:

$$f = a(z_0(t_f) - \phi_F)^2 + b(z_1(t_f))^2 + c(z_2(t_f))^2 + dJ + P \tag{3.45}$$

, kde J je hodnota optimalizačního kritéria dle (3.7), P přidává penalizaci za nedodržení podmínky $v_{min}(t) \leq v(t) \leq v_{max}(t)$. Tyto hodnoty vyžadují výpočet průběhu stavů systému pro každý návrh optimalizovaných parametrů. Parametry a, b, c, d jsou volitelné váhy jednotlivých kritérií.

Sestavený algoritmus pro nalezení optimálního řešení pomocí přímkové aproximace a aplikace GA, které je pro úlohu s definovaným t_f dále zkoumáno v kapitole 4, je přiložen v Příloze 3.

3.3.5 Numerické řešení pomocí přepínání limitních hodnot

V této podkapitole je popsáno numerické řešení optimálního řízení pomocí přepínání hodnoty řízení v na hodnoty v_{max} nebo v_0 , kde jsou použity genetické algoritmy v SW Matlab pro nalezení parametrů pro přepínací časy $[T_1, T_2, T_3, T_4, t_f]$. Tento postup vychází ze znalosti bang-bang charakteru řízení a pokouší se rozvinout řešení navržené pomocí analytické metody. Parametry přepínacích časů jsou nalezeny pomocí aplikace algoritmu:

1. Definování počtu a omezení optimalizovaných parametrů $[T_i, t_f]$ a lineárních vztahů mezi nimi. Lineární nerovnosti popisují vztahy $T_i \leq T_{i+1}$.
2. Nastavení volných parametrů GA, určením počtu jedinců v generaci, zachování TOP jedinců pro další křížení
3. Spuštění $ga()$ a odhad parametrů
4. Vytvoření odhadu limitních křivek pomocí analytického řešení, které umožní splnění úlohy.
5. Minimalizace hodnoty cílové (fitness) funkce, která je definována rovnicí:

$$f = a(z_0(t_f) - \phi_F)^2 + b(z_1(t_f))^2 + c(z_2(t_f))^2 + dJ + P \quad (3.46)$$

, kde J je hodnota optimalizačního kritéria dle (3.7), P přidává penalizaci za nedodržení podmínky $v_{min}(t) \leq v(t) \leq v_{max}(t)$. Tyto hodnoty vyžadují výpočet průběhu stavů systému pro každý návrh optimalizovaných parametrů. Parametry a, b, c, d jsou volitelné váhy jednotlivých kritérií.

Z popsaného algoritmu vyplývá, že řídicí veličina nebude naprosto přesně na svém limitu, protože ten je ovlivněn stavy daného systému, které jsou ovlivněny historií řízení. K nalezení této hranice by tedy navíc k hledání přepínacích parametrů bylo nutné několikrát iterovat odhadované řízení, tak aby rozdíl byl menší než stanové okolí nuly. Nicméně, jak je popsáno v kapitole 4, tak už první odhad limitních křivek popisuje dostatečně přesnou reprezentaci kterou je možno využít. Této hodnoty by bylo možné dosáhnout iterací v původním nelineárním systému s řízením u , ale při pokusu o tuto iterační metodu bylo zjištěno, že je obtížné navrhnout řízení, tak aby byly splněny okrajové podmínky.

Sestavený algoritmus pro nalezení optimálního řešení pomocí přepínání na limitní hodnoty pro úlohu s pevným t_f je přiložen v Příloze 4.

3.3.6 Numerické řešení pomocí genetických algoritmů

V této podkapitole je popsáno numerické řešení optimálního řízení pomocí aplikace genetických algoritmů pro nalezení vektoru vstupu systému. Znamená to, že pro každý diskretní bod řízení v je zaveden jako parametr pro optimalizaci. Tato metoda optimalizace byla zvolena kvůli výskytu bang-bang řízení, které značně komplikuje aplikaci konvenčních optimalizačních metod, protože nesplňuje požadavek diferencovatelnosti optimalizační funkce a zároveň umožňuje zahrnutí okrajových podmínek.

Tento vektor vstupů je nalezen pomocí aplikace algoritmu:

1. Definování délky vektoru v a tím i počtu optimalizovaných parametrů.
2. Nastavení volných parametrů GA, určením počtu jedinců v generaci, zachování TOP jedinců pro další křížení
3. Spuštění $ga()$ a odhad parametrů
4. Minimalizace hodnoty cílové (fitness) funkce, která je definována rovnicí:

$$f = a(z_0(t_f) - \phi_F)^2 + b(z_1(t_f))^2 + c(z_2(t_f))^2 + dJ + P \quad (3.47)$$

, kde J je hodnota optimalizačního kritéria dle (3.7), P přidává penalizaci za nedodržení podmínky $v_{min}(t) \leq v(t) \leq v_{max}(t)$. Tyto hodnoty vyžadují výpočet průběhu stavů systému pro každý návrh optimalizovaných parametrů. Parametry a, b, c, d jsou volitelné váhy jednotlivých kritérií.

Tento algoritmus byl z důvodu náročnosti numerických výpočtů zaveden pouze pro úlohu s pevným koncem a je pro ni definován čas t_f . Z důvodu výpočetní náročnosti je nutné i zvolit realistický počet diskretních bodů pro popis chování systému. Ze zkušeností získaných při simulacích je doporučený počet mezi 50 a 100 body/optimalizačními parametry. Sestavený algoritmus pro nalezení optimálního řešení pomocí GA je přiložen v Příloze 5.

Kapitola 4

Simulace a analýza

Simulace představují užitečný nástroj pro zkoumání složitých systémů, jako jsou právě baterie, a jejich chování za různých podmínek, což umožňuje provádět analýzy a testování bez nutnosti fyzické implementace.

V rámci této kapitoly jsou prezentovány simulační modely systémů, které jsou sestaveny na základě teoretických modelů odvozených v kapitole 2. Pomocí těchto modelů je následně ověřena základní úvaha, zda je spotřebovaná energie závislá na době pohybu.

V neposlední řadě jsou implementovány postupy popsané v kapitole 3 pro nalezení optimálního řešení a průběh těchto řízení je simulován na sestavených modelech. Výsledky těchto simulací jsou detailně analyzovány, popsány a porovnány.

4.1 Numerický model pro simulaci systému

V předchozích kapitolách se pracovalo zatím pouze s teoretickými modely, neznámými parametry a obecnými závislostmi a omezeními. Tato kapitola slouží k definici použitých komponent a parametrizaci obecných modelů popsanych v předchozích částech této práce. Následně jsou navrženy simulační modely v různých rozšířeních Matlabu pro provedení simulací popsaného systému.

4.1.1 Parametry baterie

Pro simulační model byla zvolena baterie složená z článků SAMSUNG 30Q1NR18650. Je uvažována ideální baterie, to znamená že každý článek je zatížen rovnoměrně a články mají totožnou hodnotu SoC. Složená baterie má jmenovité napětí 25,2 V, celkovou kapacitu 23,1 Ah a maximální trvalý výbojový proud je 15 A.

Pro popis napětí baterie ve tvaru (4.1) jsou definovány parametry Tabulkou 4.2. Výsledná závislost je zobrazena v grafu na Obr. 4.1.

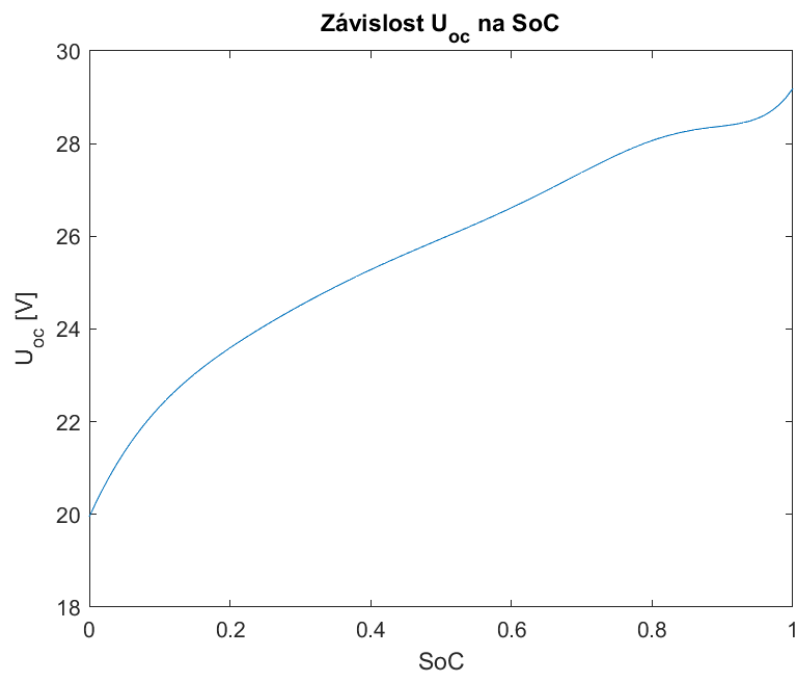
Parametr	R_0	R_P	C_P	C_n
Jednotka	Ω	Ω	F	Ah
Hodnota	0,0259	0,0133	163,38	23,1

Tabulka 4.1: Parametry pro model baterie [32]

$$U_{oc} = \sum_{j=0}^{k=8} \alpha_j SoC^j \quad (4.1)$$

Parametr	k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8
Hodnota	2.85	4.8	-17.8	38.59	4.91	-210.52	433.98	-368.45	115.81

Tabulka 4.2: Konstanty polynomu závislosti $U_{OC} = f(SoC)$ [33]



Obr. 4.1: Graf závislosti U_{oc} na Soc pro simulační model

4.1.2 Parametry motoru

Pro simulaci byl zvolen dvoupólový synchronní stejnosměrný motor s permanentními magnety a jmenovitým napětím 24V. Parametry pro popsání modelu jsou popsány v Tabulce 4.3.

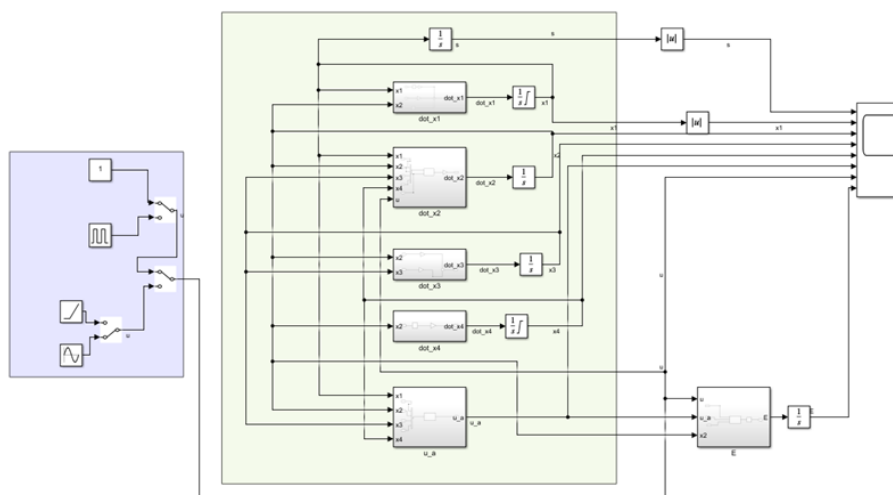
Parametr	R_m	L_m	c_{ss}	J_m
Jednotka	Ω	mH	Vs	gcm^2
Hodnota	2,19	6,38	0,066	180

Tabulka 4.3: Parametry pro model motoru

4.1.3 Sestavení modelů pro simulaci

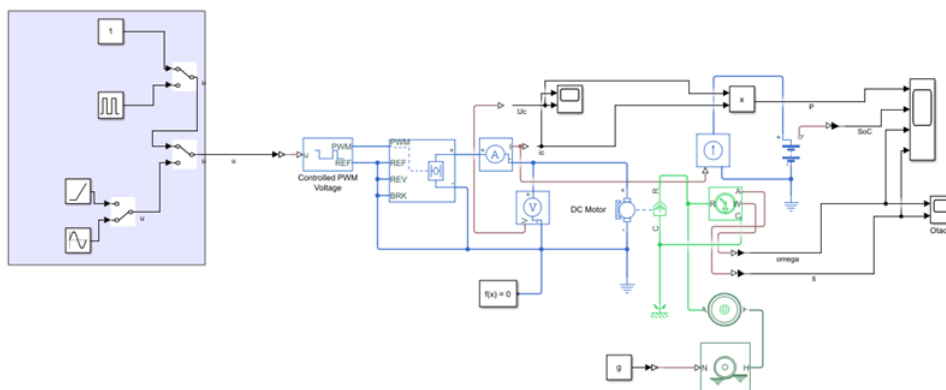
Vzhledem k rozmachu elektromobility se na trhu objevuje spousta nových SW, které nabízejí možnosti navrhování BMS (Battery Management System), identifikace vlastností a odhady parametrů pro modely baterií. Příkladem mohou být výrobci jako Ansys, Siemens a celá řada dalších včetně firmy MathWorks, jejíž SW Matlab-Simulink s aplikací Simscape je široce rozšířeným řešením, které je v průmyslu hojně využíváno. SW Matlab byl zvolen i k řešení simulační části této diplomové práce z důvodu rozsáhlých knihoven souvisejících s tématem a snadnému přístupu k SW na akademické půdě.

Pro SW Simulink byl na základě diferenciálních rovnic z kapitoly 2 sestaven model na Obr. 4.2, který popisuje chování původního nelineární model systému a je vhodný pro simulaci navržených vstupů a zkoumání chování systému. V modelu je modrý blok určen pro definici vstupu a zelený, který je složen z pěti subsystémů představujících stavy, simuluje chování systému.



Obr. 4.2: Sestavený model v Simulinku

Pro porovnání byl sestaven i model v nadstavbě Simscap (viz. Obr. 4.3), která umožňuje tzv. "Black box" modeling a je možné bez detailnějších znalostí v problematice simulovat složité soustavy. Pro problematiku baterií je zde široká nabídka předdefinovaných článků, které pracují na principu look-up tabulek a umožňují interpolaci velmi složitých nelineárních závislostí. Tento model byl použit k verifikaci sestaveného modelu. Vzhled k cílům práce, mezi které patří hlavně návrh optimálního řízení systému pomocí PMP, nebylo dále vhodné dále pracovat se Simscapem, protože pro návrh řízení byl použit popis nelinearit v bateriích pomocí polynomů.



Obr. 4.3: Sestavený model v Simscapu

Pro popis systému v lineárním systému při hledání optimálního řízení je využit Matlab, kde je použita funkce *lsim* a pro nelineární systém je autorem definována funkce ve tvaru:

```

1 $[Tx,X]$ = ode45(@ (t,x) stateEq(t,x,u,Tu), [t0 tf], [0 0 0 0 0.9],
2     options);
3 %State equations
4 function dx = stateEq(t,x,u,Tu)
5 dx = zeros(5,1);
6 u = interp1(Tu,u,t); %Interploate the control at time t
7 uoc=7*(k_0*x(5).^8+k_1*x(5).^7+k_2*x(5).^6+k_3*x(5).^5+k_4*x(5).^4+
8     k_5*x(5).^3+k_6*x(5).^2+k_7*x(5)+k_8);
9 dx(1) = x(2);
10 dx(2) = a23*x(3);
11 dx(3) = -a32*x(2) - a33*x(x) + a3u*(-c1u*x(3) - c2u*x(4) + uoc)*u;
12 dx(4) = (a43/a23)*x(3) - a44*x(4);
13 dx(5) = -abs(a53*x(3));
14 end

```

Algoritmus 4.1: Definice stavového popisu v Matlabu

4.2 Předpoklad závislosti spotřebované energie na době pohybu

Obecný předpoklad, že pomalejší pohyby jsou energeticky výhodnější, který byl součástí zadání diplomové práce, je inspirován vesmírnými aplikacemi. Při provádění úkolů ve vesmírných aplikacích je energie a přesnost obecně považována za důležitější než doba trvání výkonu.

Pro ověření toho předpokladu je ale nejprve nutné specifikovat podmínky, za kterých je ověření možné. Byly proto formátovány dvě úlohy, u kterých bude porovnávána spotřebovaná energie pro vykonání určeného pohybu na nelineárních modelech vytvořených v Simulinku.

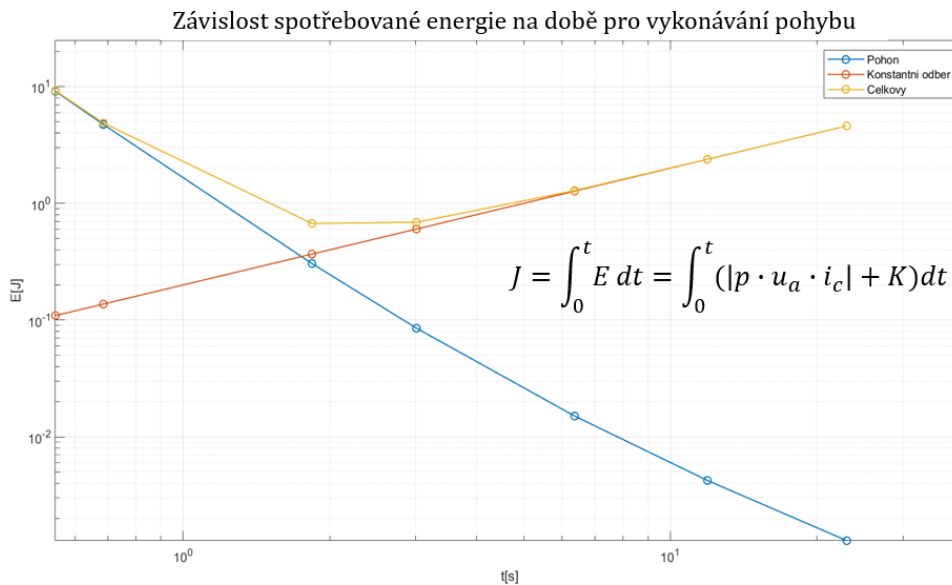
1. Úloha s konstantní rychlostí

Tato úloha má počáteční podmínky $\mathbf{x}(0) = \mathbf{0}$ a je sepnuta řízení PWM u , které je rovno konstantní hodnotě až do splnění podmínky $x_1 = \phi_F$.

2. Úloha s totožnými profily řízení

Tato úloha má počáteční podmínky $\mathbf{x}(0) = \mathbf{0}$ a řízení PWM u , je takové, aby byly splněny podmínky $x_1(t_f) = \phi_F$, $x_2(t_f) = 0$, $x_3(t_f) = 0$. Mezi porovnávanými řízeními platí: $u_i = k \cdot u_j$, kde k je konstanta.

Obě tyto úlohy byly simulovány s obdobným výsledkem (stejný profil, jiné numerické hodnoty), který je zobrazen na Obr. 4.4. Z výsledku vyplývá, že pouze pro pohyb motoru předpoklad závislosti koncového času obecně platí, ale pokud je uvažován celý systém, který má konstantní odběr energie, lze optimalizací najít ideální čas pro dobu pohybu.

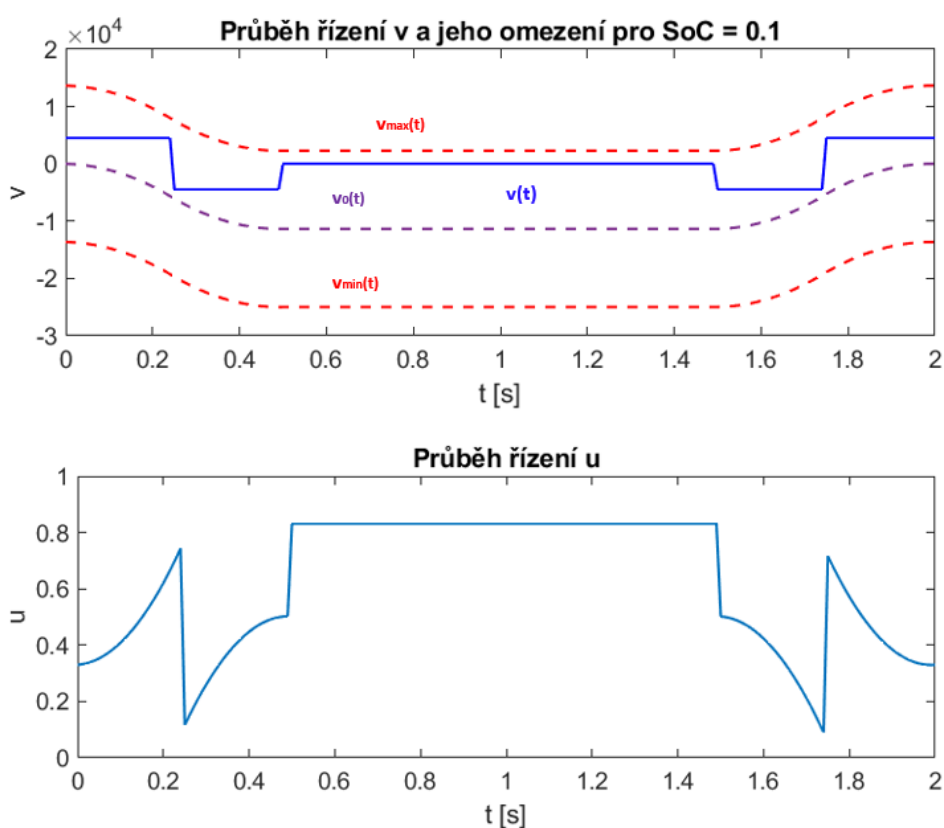


Obr. 4.4: Závislost spotřebované energie na čase

4.3 Analýza chování lineárního systému

Tato kapitola analyzuje vlastnosti sestaveného lineárního modelu systému a vysvětluje výsledky simulací získané při hledání optimálního řízení.

Pro vysvětlení těchto vlastností je vhodné využít analytický odhad řešení, který byl sestaven v kapitole 3.2.3, protože představuje nejjednodušší možný průběh řízení. Na základě simulace tohoto řízení byly získány průběhy stavů \mathbf{z} , které jsou zobrazeny na Obr. 4.6 a chování limitních křivek omezení vstupu v , které jsou nelineární a závislé na stavech systému. Výsledné křivky omezení v jsou zobrazeny na Obr. 4.5, ze kterých lze vyčíst silnou závislost řízení v na rychlosti soustavy. Na tomto obrázku je také zobrazen výsledek přepočtu řízení u pro nelineární systém.

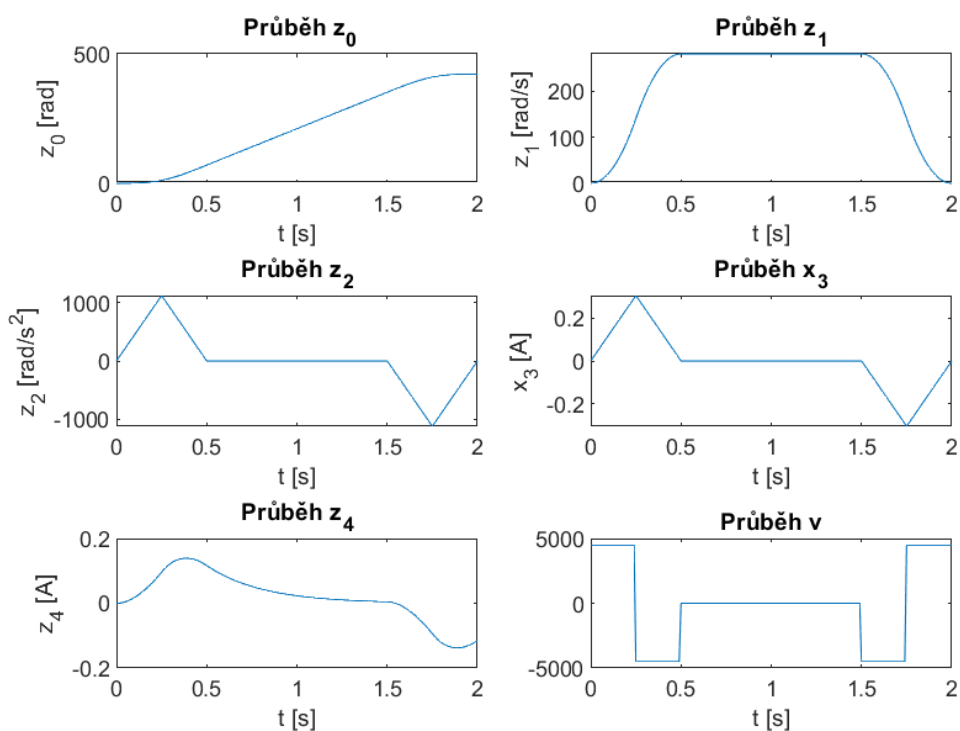


Obr. 4.5: Průběh přípustného analytického řešení v reálných mezích

Vliv rychlosti a zrychlení na chování vychází z fyzikální reprezentace vstupu v , což je "jerk" neboli ryv soustavy. Logicky z toho vyplývá, že pokud soustava dosáhne maximální ustálené rychlosti, tak maximální hodnota ryvu je rovna nule, jelikož soustava již nemůže zrychlit.

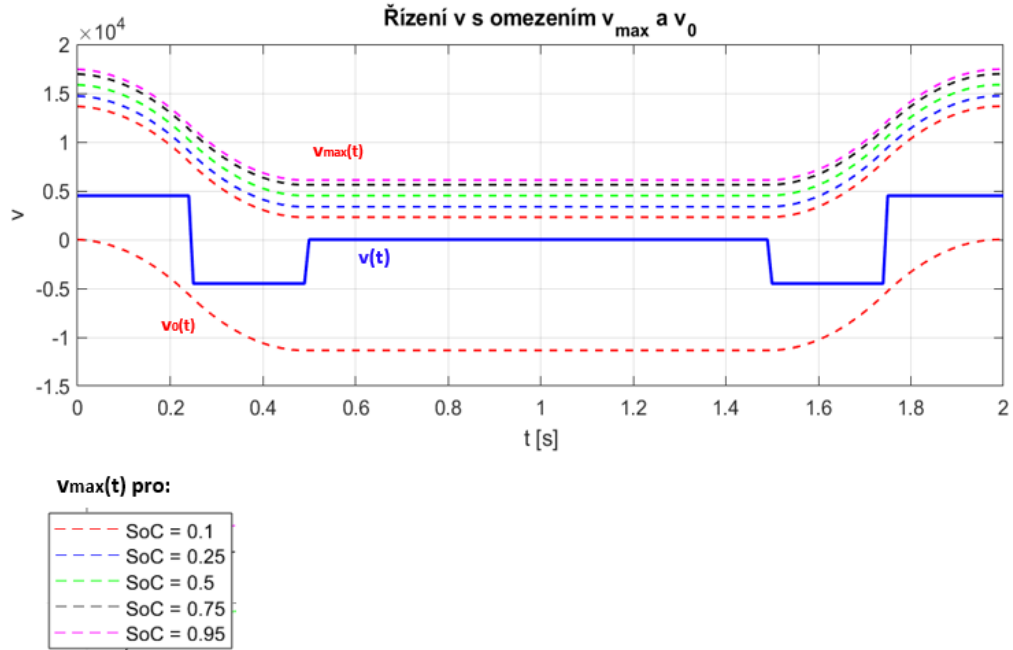
Obdobně to platí pro křivku v_0 , která reprezentuje chování při odpojení baterie a umožňuje brzdění pohybu motorem, který se dostává do režimu generátoru. Tento režim nastává pokud je indukované napětí na motoru vyšší než svorkové napětí

motoru, v modelu lze tento okamžik poznat dle záporné hodnoty proudu v modelu (obecně záporná hodnota oproti úhlové rychlosti). Tuto energii ale není možné dle zadání diplomové práce využít, protože baterie neumožňuje rekuperaci a energie je tedy "spálena". Toto chování z formulovaného kritéria přímo nevychází, jelikož kritérium obsahuje absolutní hodnotu a minimalizuje tak přenos energie obecně. Kritérium ovšem není sestaveno v rozporu s tímto chováním, jelikož se snažíme minimalizovat i množství "spálené" energie.



Obr. 4.6: Průběh stavů \mathbf{z} pro analytické řešení

Jedním ze stavů na kterých je průběh v_{max} závislý je stav nabití baterie, který může výrazně ovlivnit maximální možný ryv soustavy, jak je zobrazeno na Obr. 4.7. Tato závislost způsobuje výrazné zhoršení chování a snížení maximální možné dynamiky soustavy poháněné baterií při nízkých stavech nabití. Zároveň ze zobrazeného grafu a závislosti na Obr. 4.1 vyplývá, že bude docházet k rychlé změně dodávaného napětí a v reálných aplikacích může nastat problém s regulací při řízení soustavy. Na základě teoretické části se podobný vliv dá předpokládat i při výkyvech teplot baterie.



Obr. 4.7: Vliv SoC na maximální hodnotu jerku

4.4 Simulace systému s optimálním řízením

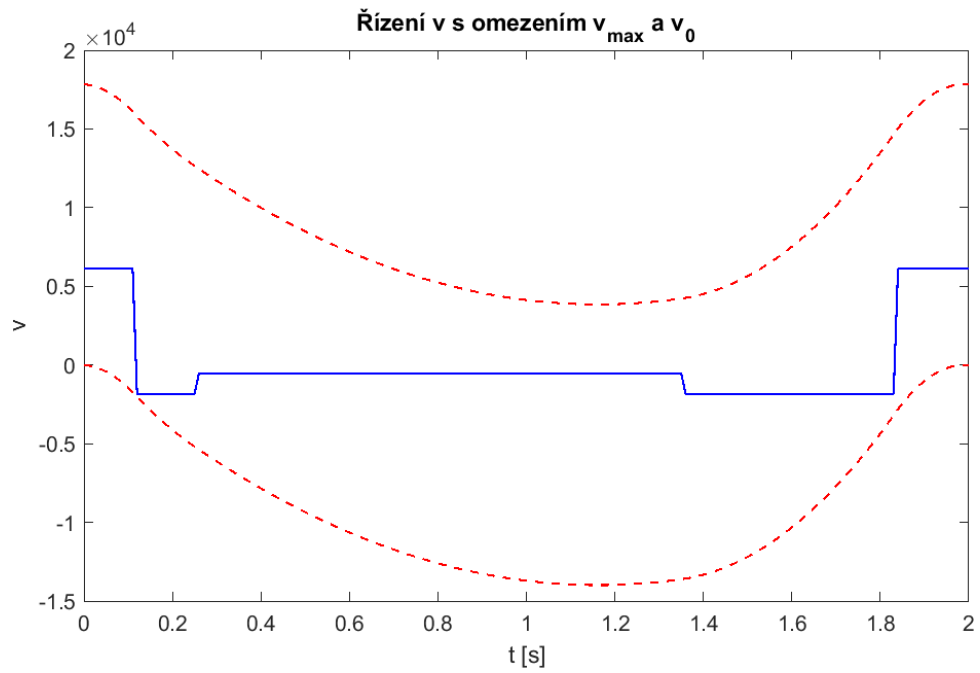
V této kapitole jsou prezentovány výsledky simulací optimalizovaných řízení zjištěných pomocí optimalizací založených na principu bang-bang a optimalizace zjištěné pomocí GA. Pro jednotlivé metody z kapitoly 3.3 jsou vykresleny průběhy stavů systému, řízení v a jeho omezení, dopočítaná hodnota u a také je porovnána výsledná spotřebovaná energie pro pohyb. Pro srovnatelnost metod a snížení numerických chyb jsou simulace provedeny pro stejnou úlohu s pevným koncem $t_f = 2s$.

4.4.1 Optimální analytické řešení

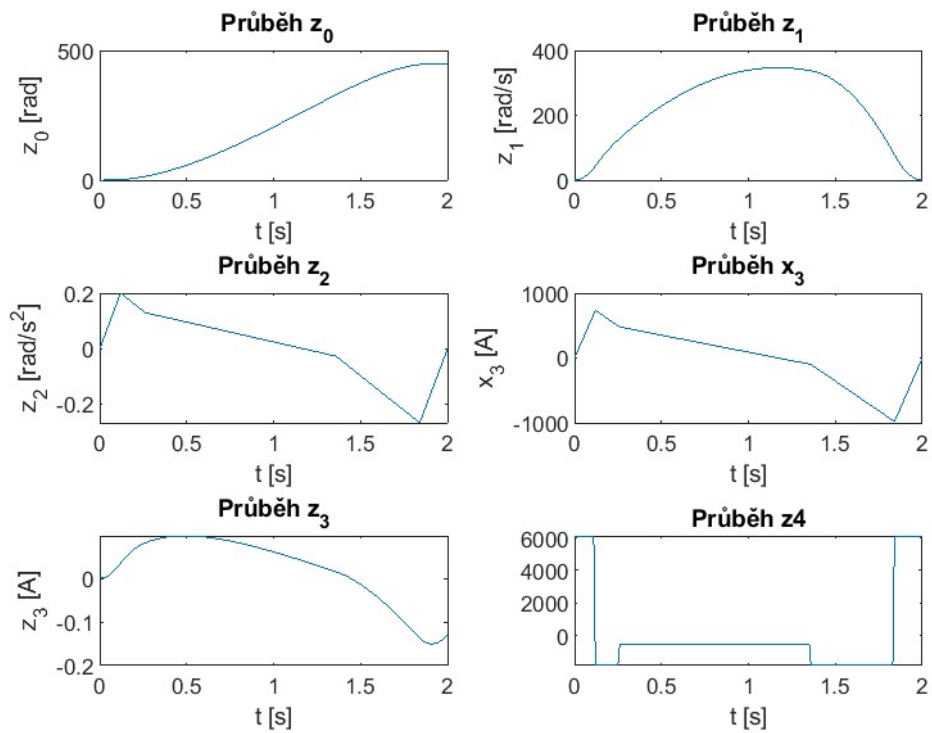
V této kapitole jsou komentovány výsledky získané simulací dle algoritmu navrženého v kapitole 3.3.3.

Příklad nalezeného suboptimálního řešení je zobrazen na Obr. 4.8, kde lze také dobře pozorovat závislost omezení vstupů v v závislosti na jeho průběhu a průběhu stavů zobrazených v grafech na Obr. 4.9.

Hodnoty spotřebované energie a výsledné optimalizované parametry pro výsledky získané pomocí GA jsou zobrazeny v Tabulce 4.4. Výsledky se liší, protože pro jejich nalezení byla použita funkce `ga()` v SW Matlab, která umožňuje pouze nalezení lokálního minima a záleží tedy na nastavení funkce a počátečním odhadu hodnot parametrů.



Obr. 4.8: Nalezené suboptimální analytické řešení nalezené pomocí GA



Obr. 4.9: Průběhy stavů pro suboptimální řešení

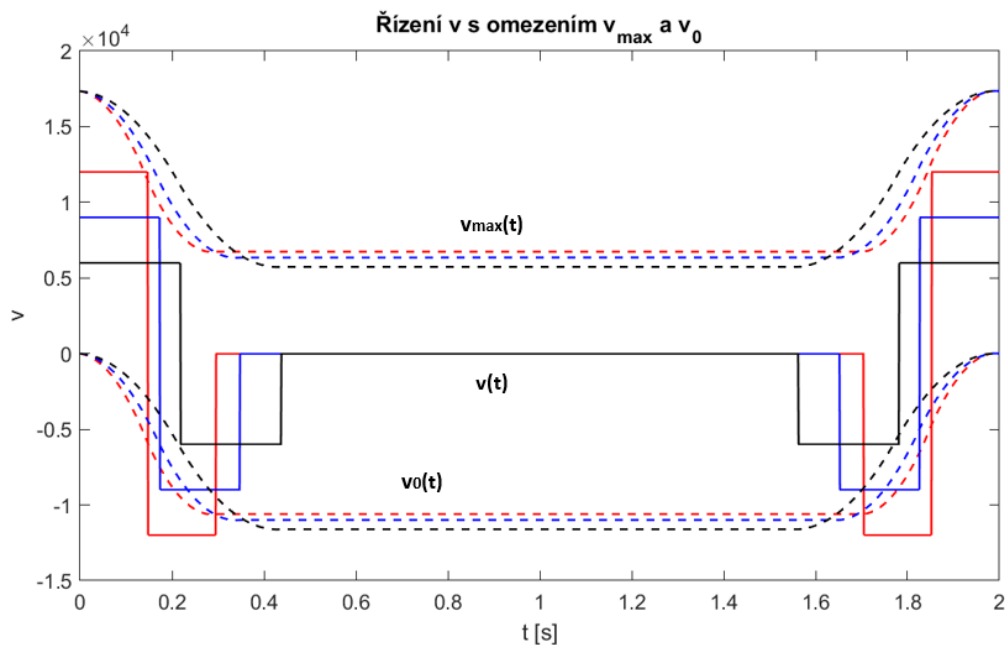
$E[J]$	$T_1[s]$	$T_2[s]$	$T_3[s]$	$T_4[s]$	$v_H \cdot 10^4$	$v_D \cdot 10^4$	v_0
4,17	0,228	0,247	1,77	1,79	0,561	-2,28	-1287
2,9	0,082	0,098	1,79	1,85	0,85	-1,16	-69
2,74	0,179	0,55	1,79	1,8	0,42	-0,12	-65
3,2	0,217	0,288	1,75	1,8	0,64	-1,5	-559
3,45	0,364	0,694	0,96	1,6	0,22	-0,16	-501
3,39	0,193	0,274	1,71	1,8	0,89	-2,19	-519

Tabulka 4.4: Výsledky pro optimalizaci analytického řešení

Obecně se ve výsledcích těchto simulací dá vypožorovat vlastnost, že pokud je hodnota $v(t)$ pro $t \in \langle T_2, T_3 \rangle$ blízká nule, tak je spotřebovaná energie nejnižší. Pro bližší analýzu a srovnání těchto simulací byla do algoritmu zavedena podmínka:

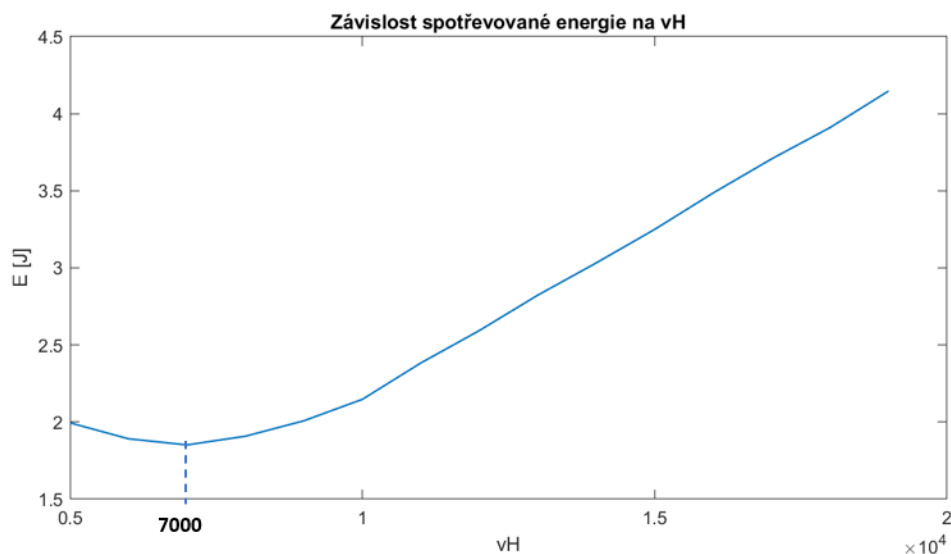
$$v_H T_1 = v_D (T_1 - T_2) \quad (4.2)$$

, což umožňuje výrazně zjednodušit výpočetní náročnost nalezení optimálního řešení. Zároveň to znamená, že vlivem řízení jsme schopni udržet rychlost konstantní. Příklady těchto řízení jsou zobrazeny na Obr. 4.10, kde lze snadno porovnat chování limitních křivek v závislosti na zvoleném řízení.



Obr. 4.10: Chování křivek v_{max} na rozdílné hodnoty v_H

Pro vykreslení závislosti spotřebované energie na hodnotě v , zobrazené na Obr. 4.11 bylo pro analytický model uvažováno $v_H = -v_D$. Minimum této závislosti bylo nalezeno pro hodnoty $v_H = 7000$ a $T_1 = 0.25$, což odpovídá řízení, kde ke zpomalení není využívána energie z baterie, ale pouze generované indukované napětí na motoru.



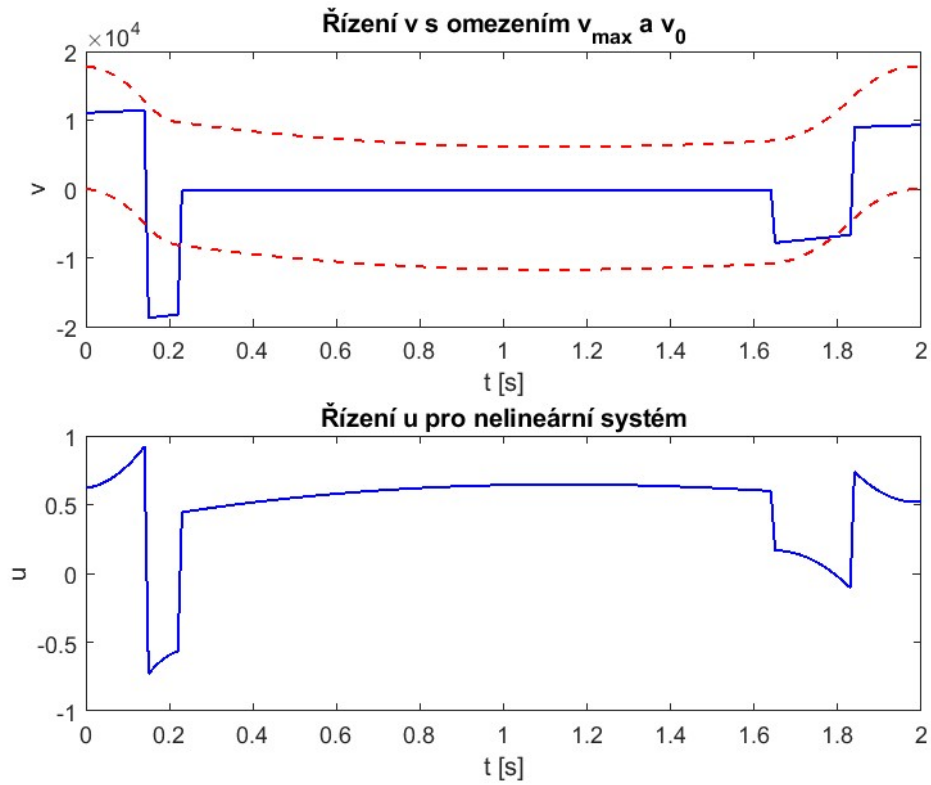
Obr. 4.11: Závislost spotřebované energie na v_H

4.4.2 Optimální řízení pomocí přímkové aproximace

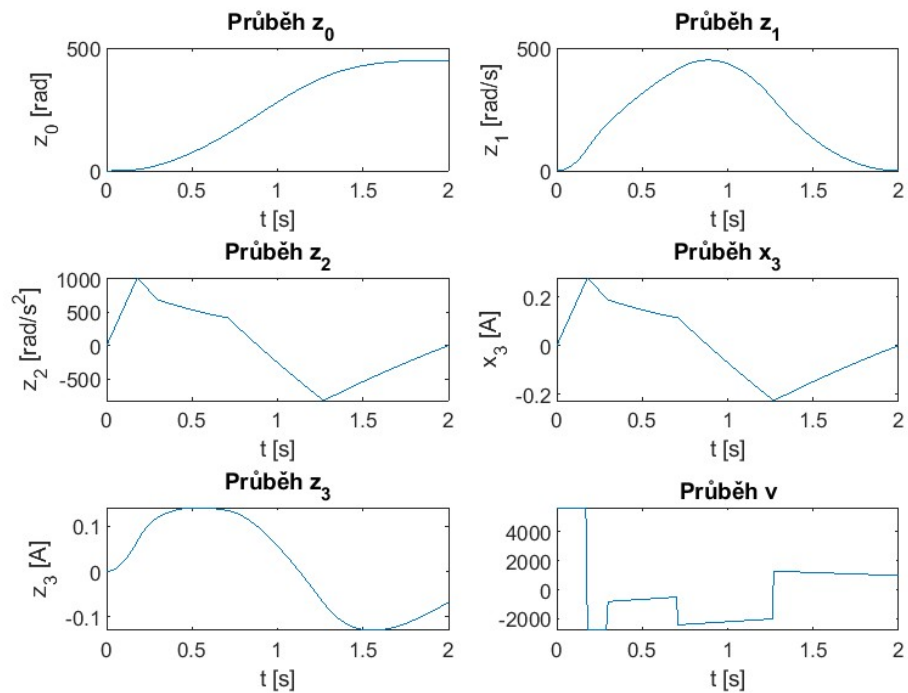
V této kapitole jsou komentovány výsledky získané simulací dle algoritmu navrženého v kapitole 3.3.4.

Příklad nalezeného řešení je zobrazen na Obr. 4.12 a průběh odpovídajících stavů je zobrazen v grafech na Obr. 4.13. Řešení vzniklo bez autorem zadaných počátečních odhadů a neodpovídá původnímu záměru této metody, kdy přímka měla představovat zjednodušenou substituci omezení v_{max} . To je pravděpodobně způsobeno nevhodným počátečním odhadem, nevhodnými volnými parametry při nastavení funkce $ga()$ nebo nedostatečným výpočetním výkonem.

Bylo provedeno několik optimalizací, hodnoty spotřebované energie a výsledné stavy, které jsou důležité pro okrajové podmínky úlohy jsou pro každou optimalizaci zobrazeny v Tabulce 4.5. Výsledky se značně liší, což je pravděpodobně způsobeno velkým počtem optimalizovaných parametrů a jejich počátečními odhady, což mělo za následek, že optimalizace našly pouze lokální minima složité funkce. Nejlepším výsledkem je optimalizace, jejíž průběh je zobrazen na Obr. 4.12 s výslednou spotřebovanou energií 2,8J.



Obr. 4.12: Nalezené řešení nalezené pomocí přímkové aproximace



Obr. 4.13: Průběhy stavů pro řešení pomocí přímkové aproximace

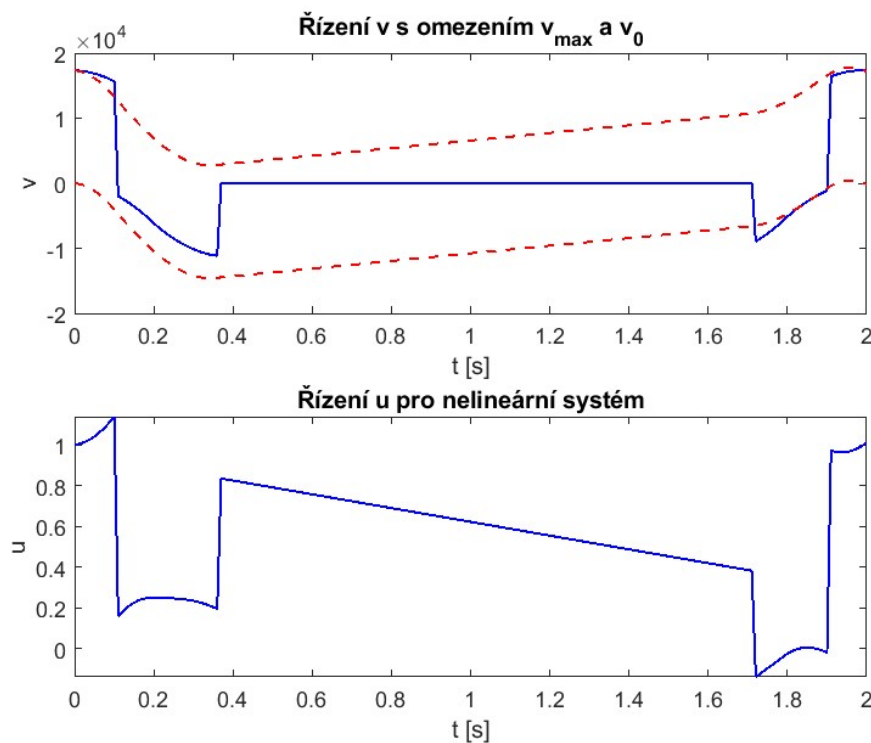
n	E	$z_0(t_f)$	$z_1(t_f)$	$z_2(t_f)$
1	2,8	499,99	0,215	-0,015
2	4,05	499,90	0,617	-0,128
3	7,86	499,95	2,717	-0,758

Tabulka 4.5: Výsledky pro optimalizaci pomocí přímkové aproximace

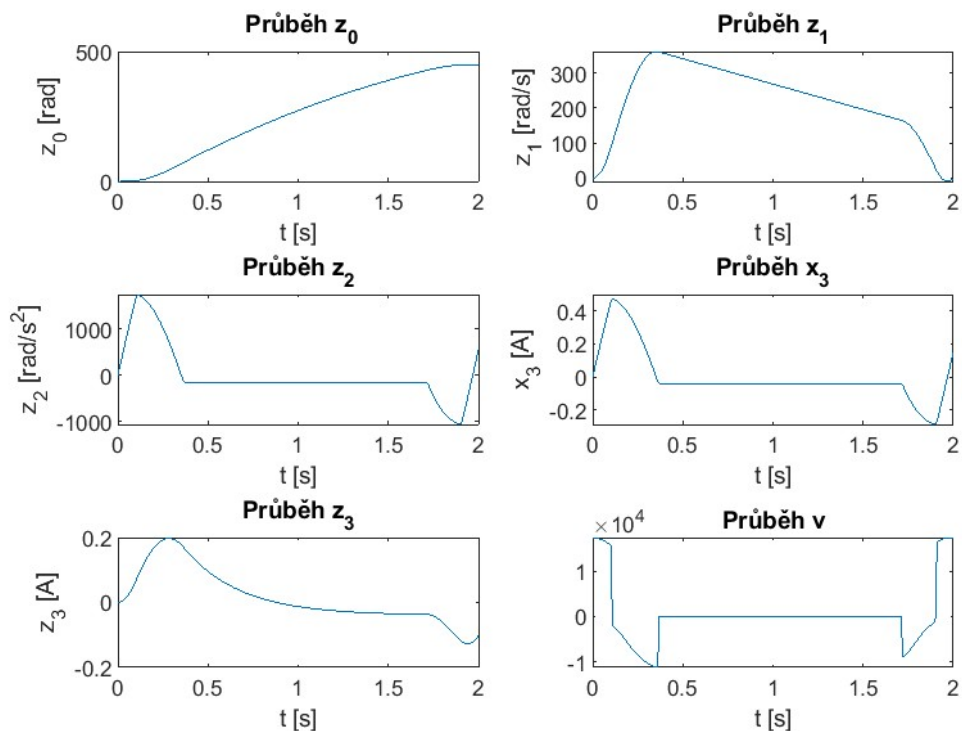
4.4.3 Optimální řízení přepínáním na limitní křivky

V této kapitole jsou komentovány výsledky získané simulací dle algoritmu navrženého v kapitole 3.3.5.

Příklad nalezeného řešení je zobrazen na Obr. 4.14 a průběh odpovídajících stavů je zobrazen v grafech na Obr. 4.15. Ze simulací lze vyčíst, že navržené řešení není dokonalé, jelikož limitní křivky nekopíruje přesně. Tato vlastnost by se dala minimalizovat zavedením iterace, která by v příštím kroku počítala s novým odhadem limitních křivek. Tato vlastnost vznikla přechodem na lineární model a pro dosažení limitů je jednodušší pracovat s původním nelineárním modelem a řízením u , ale v tom to případě se při pokusech o jeho sestavení ukázalo, že je velmi těžké dosáhnout splnění okrajových podmínek. Proto je zde navrženo řešení, které ačkoliv je zatíženo nepřesnostmi, tak zaručuje splnění podmínek. Nejlepší dosažené řízení pomocí této metody navrhuje průběh se spotřebou energie 2,2 J.



Obr. 4.14: Nalezené řešení nalezené pomocí bang-bang simulace



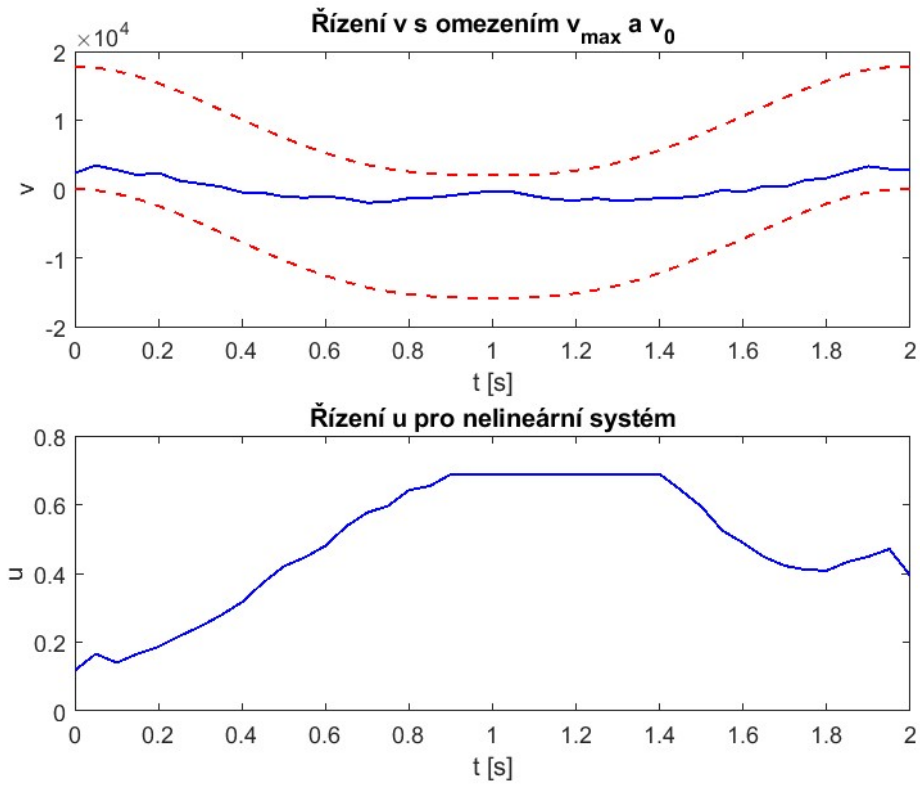
Obr. 4.15: Průběhy stavů pro řešení bang-bang simulace

4.4.4 Optimálního řízení dle GA

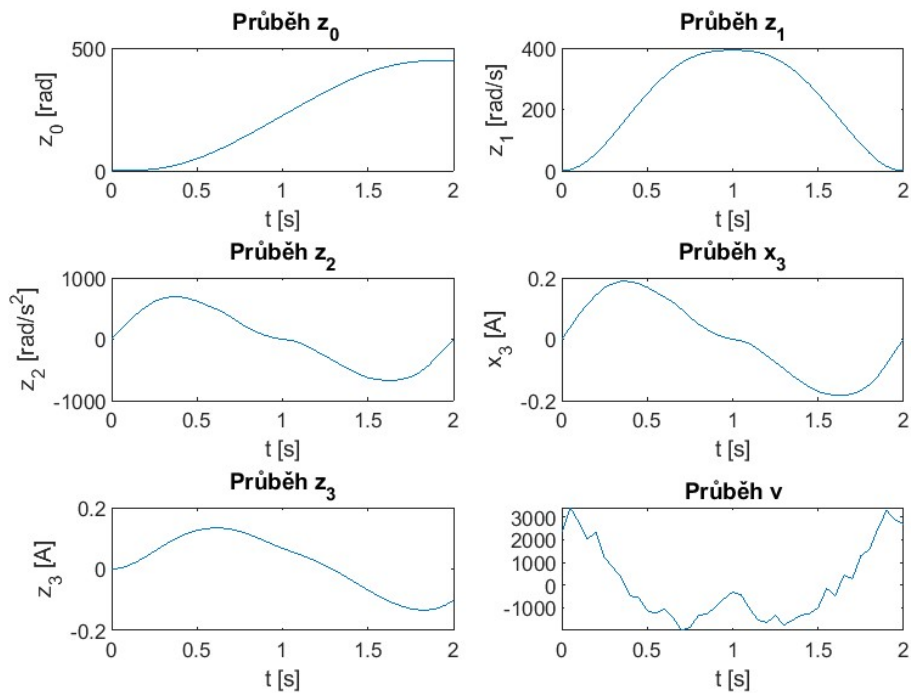
V této kapitole jsou komentovány výsledky získané simulací dle algoritmu navrženého v kapitole 3.3.6.

Příklad nalezeného řešení je zobrazen na Obr. 4.16 a průběh odpovídajících stavů je zobrazen v grafech na Obr. 4.17. Řešení vzniklo bez autorem zadaných počátečních odhadů.

Bylo provedeno několik optimalizací, hodnoty spotřebované energie a výsledné stavy, které jsou důležité pro okrajové podmínky jsou pro každou optimalizaci zobrazeny v Tabulce 4.6. Tato optimalizace se chová velmi konzistentně a každé řešení je velmi podobné s přijatelnou numerickou chybou a pohybuje se kolem výsledné hodnoty spotřebované energie 3,2 J. Charakter nalezeného řešení však neodpovídá předpokladu bang-bang, které bylo PMP zjištěno jako optimální. Tento výpočet tedy našel pouze lokální minimum. Aby se tak nestalo je potřeba nalézt správné volné hodnoty GA, zvýšit výpočetní výkon a nebo poskytnout odhad parametrů.



Obr. 4.16: Nalezené řešení nalezené pomocí aplikace GA



Obr. 4.17: Průběhy stavů pro řešení pomocí GA

n	E	$z_0(t_f)$	$z_1(t_f)$	$z_2(t_f)$
1	3,22	449,97	0,3	-0,04
2	3,17	449,96	0,32	-0,01
3	3,27	449,97	0,285	-0,04

Tabulka 4.6: Výsledky optimalizace pomocí GA

4.5 Srovnání výsledků optimalizačních metod

Výsledky simulací pomocí zkoumaných metod ukázaly, že úspěch aplikace genetického algoritmu v Matlabu závisí na mnoha faktorech, jako je volba parametrů, mutace, velikost populace a další. Problémy s mnoha lokálními minimy nebo silně nerovnoměrným rozložením optimálních bodů mohou být pro aplikovaný GA obtížné. V těchto případech existuje riziko uvíznutí v lokálním minimu, proto je jejich využití u nelineárních systémů výhodné spojit s další optimalizační metodou, která poskytne informaci o charakteru ideálního řízení, kterou v tomto případě bylo PMP.

Hodnoty získané pomocí GA se mohou násobně lišit simulace od simulace, pokud je jim ponechána velká volnost možných hodnot, není dostupný dostatečný výpočetní výkon a hledá se větší množství parametrů. Výsledky se zásadně zlepšují, pokud je funkci dán vhodný počáteční odhad hledaných parametrů.

Shrnutí nejlepších dosažených výsledků pomocí každé metody je uvedeno v Tabulce 4.7, kde je také uvedena časová náročnost metod.

Metoda	E[J]	Výpočetní čas
Analytika	1,9	Vteřiny
Přímková aproximace	2,8	Minuty
Limitní křivky	2,2	Minuty
Genetické algoritmy	3,2	Hodiny

Tabulka 4.7: Srovnání výsledků optimalizace

Řízení vycházející z analytického řešení je jednoduché na výpočet a dosahuje hodnot spotřebované energie 2,9 - 4,2 J. Pokud se do modelu zavede podmínka, že je udržována konstantní rychlost, tak bylo optimalizací dosaženo hodnoty 1,9 J.

Na základě výsledků a časové náročnosti lze říci, že nejhorší chování, bez návržení počáteční generace jedinců, vykazuje přímková aproximace, kde nejlepší nalezené řešení má hodnotu spotřebované energie 2,8 J. Nalezené výsledky jsou velmi různorodé a svým charakterem se zásadně neliší od analytické metody, což nebylo záměrem při návržení této metody. Tato metoda nesplnila navrhovaný záměr, jelikož nedokázala pomocí GA najít vhodnou přímkovou aproximaci limitních křivek omezení řízení v .

Metoda přepínání pomocí hraničních křivek zvládla dosáhnout nejnižší spotřeby energie 2,2 J. Metoda je oproti teoretickému minimu bang-bang zatížena relativně vysokou numerickou chybou plynoucí z nepřesného odhadu. Tato metoda má prostor pro výrazné zlepšení zavedením iterace a nalezením kvalitního odhadu hraniční křivky nebo aplikací prediktivního modelu chování.

Kompletní řešení pomocí GA, kde se hledá vektor řešení dosahuje hodnoty spotřebované energie 3,2 J, což je vyšší než hodnota která byla dosažena pomocí aplikace optimalizovaného analytického odhadu, ale chování této metody je velmi konzistentní a lze ho na problém aplikovat bez bližší analýzy systému s výsledkem, který je přijatelný.

Řešení pomocí genetických algoritmů se ve zkoumaných příkladech projevilo jako dobrý postup, který ovšem vyžaduje výpočetní čas a ladění volných parametrů, které mohou značně ovlivnit nalezené řešení. Z těchto důvodů je vhodné toto řešení používat k offline analýze problému a nejedná se o řešení vhodné pro real-time aplikaci. Tato problematika však vyžaduje další zkoumání a nalezení vhodných volných parametrů pro simulace.

Ve srovnání s tím analytické řešení založené na optimalizaci pomocí PMP a bang-bang chování umožňuje návrh suboptimálních řešení, které jsou vhodné pro real-time aplikace a jsou inženýrsky přijatelné.

Aplikace GA a její kombinace s PMP se projevila jako dobrý postup, který umožňuje numericky vyřešit i problémy řízení nelineárních systémů.

Závěr

Kapitola 1 diplomové práce byla věnována teoretické části, kde byly popsány základní typy elektrochemických článků, princip jejich fungování a možnosti zapojení do baterií, včetně důvodů pro využití Battery management systému. Byly také představeny různé možnosti pro modelování baterií, zejména elektrické ekvivalentní modely, které jsou vhodné pro mechatronické aplikace a lze jejich rozšířením popsat i vliv stavu nabití baterie nebo teploty na dynamické chování baterií.

V kapitole 2 byl splněn cíl vytvoření modelu robota napájeného z baterie. Pro splnění cíle byly nejdříve popsány komponenty nezbytné pro sestavení robotického systému, kde jako příklady pro zátěž motoru byly uvedeny aplikace s robotickým ramenem a aplikace s kolem vhodným pro mobilní robotiku. Následně byla provedena syntéza modelu, který je možno na základě spínání střídavy PWM řídit. Tento vzniklý model je nelineární, vlivem zavedení závislosti napětí baterie na jejím stavu nabití, a neumožňuje dobíjení baterie rekuperací při pohybu. Pro další postup bylo nutné systém linearizovat pomocí přesné vstupně-výstupní linearizace.

V kapitole 3 byla nejdříve definována optimalizační úloha a její kritérium optimality, kterým byla zvolena práce dodaná z baterie do systému. Následně bylo pro linearizovaný model navrženo optimální řízení pomocí Pontryaginova principu minima, které má charakter bang-bang a je určeno pomocí podmínek obsahujících co-stavové optimalizační parametry \mathbf{p} . Byly tak splněny cíle práce "řešení úlohy optimálního řízení" a "formulace energeticky optimálního řízení".

Pro srovnání s metodou PMP byla zvolena optimalizace pomocí genetických algoritmů, které na rozdíl od řady jiných běžně používaných optimalizačních metod mohou být jednoduše aplikovány i na problémy, které nejsou diferencovatelné. Kapitola 3 obsahuje i popis hledání numerických řešení optimalizační úlohy pomocí PMP a různých variant využití genetických algoritmů, které se liší numerickou náročností výpočtu.

Kapitola 4 je věnována představení simulačních modelů, numerických hodnot pro simulace a srovnání výsledků jednotlivých optimalizačních metod. V podkapitole 4.2 byl na základě simulací potvrzen předpoklad, že shodné pohyby jsou v nižších rychlostech energeticky úspornější. Toto tvrzení je platné pro pohon ideálního motoru, ale pokud je baterie zatížena i konstantním odběrem elektrické energie pro jiné komponenty systému nebo vlivem ztrát, tak je možné nalézt optimální dobu průběhu pohybu. S tímto dodatkem lze považovat cíl práce, ověření doby průběhu pohybu na spotřebované energii, považovat za splněný.

Pontryaginův princip minima posloužil k určení typu řízení, ale při simulacích byla zjištěna velmi vysoká citlivost okrajových podmínek co-stavových optimalizačních parametrů \mathbf{p} na průběh řízení a pro vyřešení bang-bang úlohy bylo zavedeno řešení vycházející z analytického odhadu.

Pro tento analytický odhad byly zavedeny přepínací časy a konstantní hodnoty vstupu, které byly optimalizovány. Pomocí této metody byly pro definovanou úlohu natočení o 450 radiánů s pevným časem 2 vteřiny nalezeny parametry, které umožňují pohyb při spotřebování 1,89 J. Toto suboptimální řešení se po srovnání s výsledky získanými pomocí genetických algoritmů, které dosahují hodnot 3,2 J dá považovat za přípustné a je vhodné pro real-time aplikace. Využití genetických algoritmů pro tuto problematiku vyžaduje další zkoumání a nalezení vhodných volných parametrů simulací, kterými jsou například počet jedinců v generaci nebo počáteční odhad.

Tato práce poskytuje úvod do problematiky optimálního řízení robotů využívajících baterie, kdy při návrhu optimálního řízení byla zanedbána řada dějů probíhajících v bateriích. Navržené optimální řízení je tedy optimální pouze s ohledem na sestavený model, který vyžaduje další zkoumání a detailnější zpracování nelineárních vlastností baterie. Z hlediska energetické efektivity zůstal nevyužitý potenciál dobíjení baterie pomocí rekuperace energie z prováděných pohybů.

Seznam použitých zdrojů

- [1] David Linden and Thomas B Reddy. *Linden's handbook of batteries*. McGraw-Hill Education, 2011.
- [2] RJ Brodd et al. What are batteries, fuel cells, and supercapacitors, (chem. rev. 2003, 104, 4245- 4269. published on the web 09/28/2004.). *Chemical Reviews*, 105(3):1021, 2005.
- [3] Fuel cell working principle. EEeguide [online]. [cit.2023-08-14].
Dostupné z: <https://www.eeeguide.com/fuel-cell-working-principle-and-schematic-diagram/>.
- [4] Giuseppe Guidi. Energy management systems on board of electric vehicles, based on power electronics. 2009.
- [5] Sheng S Zhang. Liquid electrolyte lithium/sulfur battery: Fundamental chemistry, problems, and solutions. *Journal of Power Sources*, 231:153–162, 2013.
- [6] KM Abraham. How comparable are sodium-ion batteries to lithium-ion counterparts? *ACS Energy Letters*, 5(11):3544–3547, 2020.
- [7] Federico Baronti, Gabriele Fantechi, Roberto Roncella, and Roberto Saletti. High-efficiency digitally controlled charge equalizer for series-connected cells based on switching converter and super-capacitor. *IEEE Transactions on Industrial Informatics*, 9(2):1139–1147, 2012.
- [8] Davide Andrea. *Battery management systems for large lithium-ion battery packs*. Artech house, 2010.
- [9] Prakash Venugopal. State-of-health estimation of li-ion batteries in electric vehicle using indrmn under variable load condition. *Energies*, 12(22):4338, 2019.
- [10] Thomas F Fuller, Marc Doyle, and John Newman. Simulation and optimization of the dual lithium ion insertion cell. *Journal of the electrochemical society*, 141(1):1, 1994.
- [11] Peng Rong and Massoud Pedram. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *IEEE transactions on very large scale integration (VLSI) systems*, 14(5):441–451, 2006.
- [12] WB Gu and CY Wang. Thermal-electrochemical modeling of battery systems. *Journal of The Electrochemical Society*, 147(8):2910, 2000.

- [13] Lorenzo Serrao, Zakaria Chehab, Y Guezennee, and Giorgio Rizzoni. An aging model of ni-mh batteries for hybrid electric vehicles. In *2005 IEEE Vehicle Power and Propulsion Conference*, pages 8–pp. IEEE, 2005.
- [14] Markus Einhorn, Fiorentino Valerio Conte, Christian Kral, and Jürgen Fleig. Comparison, selection, and parameterization of electrical battery models for automotive applications. *IEEE Transactions on Power Electronics*, 28(3):1429–1437, 2012.
- [15] Abbas Fotouhi, Daniel J Auger, Karsten Propp, Stefano Longo, and Mark Wild. A review on electric vehicle battery modelling: From lithium-ion toward lithium–sulphur. *Renewable and Sustainable Energy Reviews*, 56:1008–1021, 2016.
- [16] Hongwen He, Rui Xiong, and Jinxin Fan. Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach. *energies*, 4(4):582–598, 2011.
- [17] Wang Kai, Feng Xiao, Pang Jinbo, Ren Jun, Duan Chongxiong, and Li Liwei. State of charge (soc) estimation of lithium-ion battery based on adaptive square root unscented kalman filter. *International Journal of Electrochemical Science*, 15(9):9499–9516, 2020.
- [18] Jeffrey R Belt. Battery test manual for plug-in hybrid electric vehicles. Technical report, Idaho National Lab.(INL), Idaho Falls, ID (United States), 2010.
- [19] Yinjiao Xing, Wei He, Michael Pecht, and Kwok Leung Tsui. State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures. *Applied Energy*, 113:106–115, 2014.
- [20] Antoni Szumanowski and Yuhua Chang. Battery management system based on battery nonlinear dynamics modeling. *IEEE transactions on vehicular technology*, 57(3):1425–1432, 2008.
- [21] Habiballah Rahimi-Eichi and Mo-Yuen Chow. Adaptive parameter identification and state-of-charge estimation of lithium-ion batteries. In *IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society*, pages 4012–4017. IEEE, 2012.
- [22] Simscape tablebased battery. Mathworks [online]. [cit.2023-08-14].
Dostupné z: <https://www.mathworks.com/help/simscape-battery/ref/battery-tablebased.html>.
- [23] What is a PWM signal. circuitbread [online]. [cit.2023-08-14].
Dostupné z: <https://www.circuitbread.com/ee-faq/what-is-a-pwm-signal>.
- [24] H-bridge. Build electronic circuits [online]. [cit.2023-08-14].
Dostupné z: <https://www.build-electronic-circuits.com/h-bridge>.

- [25] Hans B Pacejka and Egbert Bakker. The magic formula tyre model. *Vehicle system dynamics*, 21(S1):1–18, 1992.
- [26] David E Hall and J Cal Moreland. Fundamentals of rolling resistance. *Rubber chemistry and technology*, 74(3):525–539, 2001.
- [27] Chankyu Lee, K. Hedrick, and Kyongsu Yi. Real-time slip-based estimation of maximum tire-road friction coefficient. *IEEE/ASME Transactions on Mechatronics*, 9(2):454–458, 2004.
- [28] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.
- [29] Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. CRC press, 1987.
- [30] Genetic algorithm. Mathworks [online]. [cit.2023-08-14].
Dostupné z: <https://www.mathworks.com/help/gads/genetic-algorithm-options.html#f17837>.
- [31] Michael Valášek. Dynamic time parametrization of manipulator trajectories. *Kybernetika*, 23(2):154–174, 1987.
- [32] Jiechao Lv, Baochen Jiang, Xiaoli Wang, Yirong Liu, and Yucheng Fu. Estimation of the state of charge of lithium batteries based on adaptive unscented kalman filter algorithm. *Electronics*, 9(9):1425, 2020.
- [33] Fadlaoui Elmahdi, Lagrat Ismail, and Masaif Noureddine. Fitting the ocv-soc relationship of a battery lithium-ion using genetic algorithm method. In *E3S Web of Conferences*, volume 234, page 00097. EDP Sciences, 2021.

Seznam použitých obrázků

1.1	Akumulátor při nabíjení [1]	3
1.2	Akumulátor při vybíjení [1]	3
1.3	Palivový článek [3]	4
1.4	Vliv balancování článků na kapacitu baterie [8]	6
1.5	Odezva napětí na baterii na skokovou změnu proudu [15]	8
1.6	Rozdílné přístupu k modelování baterie [15]	8
1.7	Model s vnitřním odporem	9
1.8	Schéma RC modelu	9
1.9	Schéma Theveninova modelu prvního řádu	10
1.10	Schéma Theveninova modelu druhého řádu	11
1.11	Schéma modelu PNGV	12
1.12	Srovnání závislosti U_{oc} na SoC různých typů elchem. článků[1]	13
1.13	Závislost U_{oc} Lithium-iont baterie na SoC a procházejícím proudu [8]	14
1.14	Závislost U_{oc} baterie na teplotě[22]	14
2.1	Schéma obecného mechatronického systému s baterií	15
2.2	Schéma modelu baterie	16
2.3	Převod analogového signálu na digitální pomocí PWM [23]	17
2.4	Šířková modulace [23]	18
2.5	H-můstek [24]	18
2.6	DC Motor	19
2.7	Závislost valivého odporu na rychlosti [26]	20
2.8	Závislost koeficientu tření na prokluzu [27]	21
2.9	Převod analogového signálu na digitální pomocí PWM [23]	21
2.10	Transformace souřadnic a řízení	24

2.11	Schéma systému mobilního robota	25
3.1	Analytické řešení	34
3.2	Analytické řešení	35
4.1	Graf závislosti U_{oc} na Soc pro simulační model	42
4.2	Sestavený model v Simulinku	43
4.3	Sestavený model v Simscapu	44
4.4	Závislost spotřebované energie na čase	45
4.5	Průběh přípustného analytického řešení v reálných mezích	46
4.6	Průběh stavů \mathbf{z} pro analytické řešení	47
4.7	Vliv SoC na maximální hodnotu jerku	48
4.8	Nalezené suboptimální analytické řešení nalezené pomocí GA	49
4.9	Průběhy stavů pro suboptimální řešení	49
4.10	Chování křivek v_{max} na rozdílné hodnoty v_H	50
4.11	Závislost spotřebované energie na v_H	51
4.12	Nalezené řešení nalezené pomocí přímkové aproximace	52
4.13	Průběhy stavů pro řešení pomocí přímkové aproximace	52
4.14	Nalezené řešení nalezené pomocí bang-bang simulace	53
4.15	Průběhy stavů pro řešení bang-bang simulace	54
4.16	Nalezené řešení nalezené pomocí aplikace GA	55
4.17	Průběhy stavů pro řešení pomocí GA	55

Seznam použitých tabulek

1.1	Elektrický potenciál materiálů používaných na anody [1]	4
1.2	Přehled vlastností běžně užívaných baterií [1],[4]	5
4.1	Parametry pro model baterie [32]	42
4.2	Konstanty polynomu závislosti $U_{OC} = f(SoC)$ [33]	42
4.3	Parametry pro model motoru	43
4.4	Výsledky pro optimalizaci analytického řešení	50
4.5	Výsledky pro optimalizaci pomocí přímkové aproximace	53
4.6	Výsledky optimalizace pomocí GA	56
4.7	Srovnání výsledků optimalizace	56

Přílohy

Příloha 1

```
1   close all
2   max_iteration=100;
3   t0=0;
4   Tf=1;
5   eps_tol=10^(-7)
6
7   %Konstanty
8   global a23 a32 a33 a3u a43 a44 a53 c1 c2 uoc
9   z5 = 0.9
10
11  %Vypocet odhadu
12  %Linearni system
13  A = [0 1 0 0 0;
14       0 0 1 0 0;
15       0 0 0 0 0;
16       0 0 a43/a23 -a44 0;
17       0 0 -a53 0 0];
18  B = [0; 0; 1; 0; 0];
19  C = [1 0 0 0 0;
20       0 1 0 0 0;
21       0 0 1 0 0;
22       0 0 0 0 0;
23       0 0 0 1 0;
24       0 0 0 0 1];
25  D = [0 ;0; 0;1;0;0];
26
27  sys1 = ss(A,B,C,D);
28
29  x0 = [0 0 0 0 1];
30  y=x0
31  t = 0:0.001:2;
32  v = zeros(length(t),1);
33  v_max=4500;
34  v_min=-4500;
35  v(t<.25) = v_max;
36  v(t>=.25) = v_min;
37  v(t>=.5) = 0;
38  v(t>=1.5) = v_min;
39  v(t>=1.75) = v_max;
40
41  grid on
42
43  [y,t,x]=lsim(sys1,v,t,x0);
44
45  dx5=-a53*abs(x3);
46  uoc=7*(k_0*z5.^8+k_1*z5.^7+k_2*z5.^6+k_3*z5.^5+k_4*z5.^4+k_5*z5.^3+
47       k_6*z5.^2+k_7*z5+k_8);
48  v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
49         )*a23;
50  v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
51         )*a23;
52  v_0=(-a32*y(:,2)-a33*y(:,3)/a23)*a23;
```

```

52 u=(y(:,4)/a23+a32*y(:,2)+a33*y(:,3)/a23)/(a3u*(uoc-c1*y(:,3)/a23-c2
    *y(:,5)));
53
54
55 x2=y(:,2);
56 x3=y(:,3);
57 u=y(:,4);
58 Tu=t;
59 Tx=t;
60 tf=t(end)
61
62 %Iterace reseni
63 for i = 1:max_iteration
64 % 1) Vyreseni optimalniho rizeni u
65 [Tx,X] = ode45(@(t,x) stateEq(t,x,u,Tu), [t0 tf], [0 0 0 0 0.9],
    options);
66 x2 = X(:,2); x3 = X(:,3); x4 = X(:,4);
67
68 v_max=(-a32*x2-a33*x3/a23+a3u*(uoc-c1*x3/a23-c2*x4))*a23;
69 v_0=(-a32*x2-a33*x3/a23))*a23;
70
71 % 2) Zpetne dopocitani co-stavu z predpokladu p(T)=0
72 %x1 = X(:,1); x2 = X(:,2); x3 = X(:,3);
73 [Tp,P] = ode45(@(t,p) costateEq(t,p,u,Tu,x2,x3,Tx), [tf t0], [0 0 0
    0 0], options);
74 p1 = P(:,1); p2 = P(:,2); p3 = P(:,3);
75
76
77 % Interpolace p dle x
78 p1 = interp1(Tp,p1,Tx);
79 p2 = interp1(Tp,p2,Tx);
80 p3 = interp1(Tp,p3,Tx);
81
82
83 % Vypocet hodnoty H
84 dH = pH(x3,p3,Tx,u,Tu);
85 H_Norm = (dH')*dH;
86
87 %Interpolace
88 u=interp1(Tu,u,t);
89 x2=interp1(Tx,x2,t);
90 x3=interp1(Tx,x3,t);
91 x4=interp1(Tx,x4,t);
92 p1 = interp1(Tx,p1,t);
93 p2 = interp1(Tx,p2,t);
94 p3 = interp1(Tx,p3,t);
95
96 %Vypocet hodnoty kriteria
97 J(i,:) = abs(((u+a23*a32*x2+a32*x3).*x3)/((a23^2)*a3u));
98 W1=trapz(t,J(i,:))
99
100
101 % if dH/du < epslon, exit
102 if H_Norm < eps_tol
103
104 break;

```



```

105 else
106 % Definice noveho rizeni
107 u_old = u;
108 u = AdjControl(u_old,x2,x3,x4,p3);
109
110 end
111 end
112
113 % State equations
114 function dx = stateEq(t,x,u,Tu)
115 global a23 a43 a44 a53
116 dx = zeros(5,1);
117 u = interp1(Tu,u,t); % Interploate the control at time t
118 dx(1) = x(2);
119 dx(2) = x(3);
120 dx(3) = u;
121 dx(4) = (a43/a23)*x(3)-a44*x(4);
122 dx(5)=-abs(a53*x(3));
123 end
124
125 % Costate equations
126 function dp = costateEq(t,p,u,Tu,x2,x3,xt)
127 global a23 a32 a33 a3u a43 a44 a53 c1 c2
128 dp = zeros(3,1);
129 x2 = interp1(xt,x2,t); % Interploate the state varialbes
130 x3 = interp1(xt,x3,t);
131 u = interp1(Tu,u,t); % Interploate the control
132 dp(1) = 0;
133 dp(2) = -a32/(a3u*a23)*x3+p(1);
134 dp(3) = (u+a23*a32*x2+2*a32*x3)/((a23^2)*a3u)+p(2)+(a43/a23)*p(4) -
          a53*p(5);
135 dp(4) = -p(4)*a44;
136 dp(5) = 0;
137 end
138
139 % Parcialni derivace H
140 function dH = pH(x3,p3,tx,u,Tu)
141 global a23 a32 a33 a3u a43 a44 a53 c1 c2
142 % interpolace rizeni
143 u = interp1(Tu,u,tx);
144 dH = (x3)/((a23^2)*a3u)+p3;
145 end
146
147
148 % Funkce pro nove rizeni
149 function u_new = AdjControl(u_old,x2,x3,x4, p3)
150 global a23 a32 a33 a3u a43 a44 a53 c1 c2 uoc
151 % step = 0.1
152 % u_new = u_old - step*pH;
153 % end
154 for m=1:length(u_old)
155 if (u_old(m)>0)
156     u_new(m) = (-a32*x2(m)-a33*x3(m)/a23+a3u*(uoc-c1*x3(m)/a23-c2*x4(
          m)))*a23;
157
158 elseif (u_old(m)==0)

```

```
159 u_new(m) = 0;
160
161 else
162 u_new(m) = ((-a32*x2(m)-a33*x3(m)/a23))*a23;
163
164 end
165 end
166 figure()
167 plot(u_new)
168 end
```

Algoritmus 2: Algoritmus pro numerický výpočet PMP

Příloha 2

```
1 %opts = optimoptions(@ga,'PopulationSize',200,'EliteCount',5,'
  FunctionTolerance',10e-5,'PlotFcn',@gaplotbestf,'
  InitialPopulationMatrix',[0.25,0.5,1.5,1.75,4500,-4500]);
2 opts = optimoptions(@ga,'PopulationSize',300,'EliteCount',10,'
  FunctionTolerance',10e-7,'PlotFcn',@gaplotbestf);
3 %options = bvpset("Stats","on","RelTol",1e-4);
4 T=2;
5 fi_F=450;
6
7 Ap = [1 -1 0 0 0 0 0;
8       0 1 -1 0 0 0 0;
9       0 0 1 -1 0 0 0];
10 bp = [0;0;0];
11 lb = [0; 0; 0.25*T; 0.25*T; 0; -30000;-10000];
12 ub = [0.25*T; 0.5*T; T; T; 30000; 0;0];
13 z5=1;
14 global A B C D x0 t sys1 T A B C D a23 a32 a33 a44 a53 y a33 uoc
  a3u c1 c2 fi_F
15
16 %Linearni system
17 A = [0 1 0 0 0;
18       0 0 1 0 0;
19       0 0 0 0 0;
20       0 0 a43/a23 -a44 0;
21       0 0 -a53 0 0];
22 B = [0; 0; 1; 0; 0];
23 C = [1 0 0 0 0;
24       0 1 0 0 0;
25       0 0 1 0 0;
26       0 0 0 0 0;
27       0 0 0 1 0;
28       0 0 0 0 1];
29 D = [0 ;0; 0;1;0;0];
30
31 sys1 = ss(A,B,C,D);
32
33 x0 = [0 0 0 0 1];
34 y=x0;
35 t = 0:0.01:2;
36 v = zeros(length(t),1);
37
38 %Iteracni cyklus hledani suboptimalnich reseni
39 n=5
40 for i=1:n
41 [x,Fval,exitFlag] = ga(@fun,7,Ap,bp,[],[],lb,ub,[],[],opts)
42 %[x,Fval,exitFlag] = ga(@fun,7,Ap,bp,[],[],lb,ub,@mycon,[],opts);
43
44 %Ulozeni subopti reseni
45 T1(i)=x(1);
46 T2(i)=x(2);
47 T3(i)=x(3);
48 T4(i)=x(4);
49 vH(i)=x(5);
50 vD(i)=x(6);
```

```

51 v0(i)=x(7);
52
53
54 %Vypocet a ulozeni prubehu pro vykresleni
55 v(t<T1(i)) = vH(i);
56 v(t>=T1(i)) = vD(i);
57 v(t>=T2(i)) = v0(i);
58 v(t>=T3(i)) = vD(i);
59 v(t>=T4(i)) = vH(i);
60
61 [y,t,x]=lsim(sys1,v,t,x0);
62
63 uoc=7*(k_0*z5.^8+k_1*z5.^7+k_2*z5.^6+k_3*z5.^5+k_4*z5.^4+k_5*z5.^3+
    k_6*z5.^2+k_7*z5+k_8);
64
65 v_max(:,i)=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y
    (:,5)))*a23;
66 v_min(:,i)=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y
    (:,5)))*a23;
67 v_0(:,i)=((-a32*y(:,2)-a33*y(:,3)/a23))*a23;
68
69
70 z0(:,i)=y(:,1);
71 z1(:,i)=y(:,2);
72 z2(:,i)=y(:,3);
73 z3(:,i)=y(:,5);
74
75 v_i(:,i)=y(:,4);
76 x3(:,i)=y(:,3)/a23;
77 z4_v(:,i)=1-a53*abs(y(:,3)/a23);
78
79 u(:,i)=(y(:,4)/a23+a32*y(:,2)+a33*y(:,3)/a23)./(a3u*(uoc-c1*y(:,3)/
    a23-c2*y(:,5)));
80
81
82 J(:,i)=abs((v_i(:,i)+a23*a32*z1(:,i)+a32*z2(:,i)).*z2(:,i))/((a23
    ^2)*a3u)+0.2;
83
84 W(i)=trapz(t,J(:,i))
85
86
87 end
88
89
90 function f=fun(x)
91 global A B C D x0 t sys1 T fi_F a23 a32 a3u T a23 a32 a33 a44 a53 y
    a33 uoc a3u c1 c2
92 T1=x(1);
93 T2=x(2);
94 T3=x(3);
95 T4=x(4);
96 vH=x(5);
97 vD=x(6);
98 v0=x(7);
99 Pen=0;
100

```

```

101 v(t<T1) = vH;
102 v(t>=T1) = vD;
103 v(t>=T2) = v0;
104 v(t>=T3) = vD;
105 v(t>=T4) = vH;
106
107 [y,t,x]=lsim(sys1,v,t,x0);
108
109 z0(:)=y(:,1);
110 z1(:)=y(:,2);
111 z2(:)=y(:,3);
112 z3(:)=y(:,5);
113 z4(:)=y(:,6);
114 v_i(:)=y(:,4);
115 v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
        )*a23;
116 v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
        )*a23;
117
118 J(:)=abs((v_i(:)+a23*a32*z1(:)+a32*z2(:)).*z2(:))/((a23^2)*a3u
        +0.2;
119
120 W=trapz(t,J(:))
121
122 for in=1:length(v_i)
123     if any(v_i(in)>v_max(in)|v_i(in)<v_min(in))
124
125         Pen=200000;
126     end
127 end
128
129 f= 20*(fi_F-z0(end))^2+(z1(end)^2)+(z2(end))^2+100*W+Pen
130 end
131
132 function [c,ceq] = mycon(x)
133 T1=x(1); T2=x(2); T3=x(3); T4=x(4); vH=x(5); vD=x(6); v0=x(7);
134
135
136 c = [] % Compute nonlinear inequalities at x.
137 ceq(1) = T1*vH - v0*(T2 - T3) - vD*(T1 - T2) - vD*(T3 - T4) + vH*(T
        - T4); % Compute nonlinear equalities at x.
138 ceq(2) = T2^2*v0 + T1^2*vD + T3^2*vD + T^2*vH + T4^2*vH - T*T2*v0 +
        T*T3*v0 - T2*T3*v0 - T*T1*vD + T*T2*vD - T*T3*vD - T1*T2*vD + T
        *T4*vD - T3*T4*vD + T*T1*vH - 2*T*T4*vH;
139
140 end

```

Algoritmus 3: Algoritmus výpočtu parametrů pro parametrů suboptimálního řízení pomocí GA

Příloha 3

```
1
2 %opts = optimoptions(@ga,'PopulationSize',200,'EliteCount',5,'
   FunctionTolerance',10e-5,'PlotFcn',@gaplotbestf,'
   InitialPopulationMatrix',[0.1,0.25,1.5,1.75,18500,-5500,0,
   15000,-9000,-6000,3000,0,3000,-6000]);
3 opts = optimoptions(@ga,'PopulationSize',500,'EliteCount',10,'
   FunctionTolerance',10e-7,'PlotFcn',@gaplotbestf);
4 %options = bvpset("Stats","on","RelTol",1e-4);
5 T=2;
6 fi_F=450;
7
8 Ap = [1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0;
9       0 1 -1 0 0 0 0 0 0 0 0 0 0 0 0;
10      0 0 1 -1 0 0 0 0 0 0 0 0 0 0 0];
11 bp = [0;0;0];
12 lb = [0; 0; 0.25*T; 0.25*T; 0; -30000;-15000; 0; -30000; -inf; -inf
   ; -inf; -inf; -inf];
13 ub = [0.25*T; 0.5*T; T; T; 30000; 0;0; 30000 ; 0; inf; inf; inf;
   inf; inf];
14 z5=1;
15 global A B C D x0 t sys1 T A B C D a23 a32 a33 a44 a53 y a33 uoc
   a3u c1 c2 fi_F
16
17 %Linearni system
18 A = [0 1 0 0 0;
19       0 0 1 0 0;
20       0 0 0 0 0;
21       0 0 a43/a23 -a44 0;
22       0 0 -a53 0 0];
23 B = [0; 0; 1; 0; 0];
24 C = [1 0 0 0 0;
25       0 1 0 0 0;
26       0 0 1 0 0;
27       0 0 0 0 0;
28       0 0 0 1 0;
29       0 0 0 0 1];
30 D = [0 ;0; 0;1;0;0];
31
32 sys1 = ss(A,B,C,D);
33
34 x0 = [0 0 0 0 1];
35 y=x0;
36 t = 0:0.01:2;
37 v = zeros(length(t),1);
38
39
40 n=3
41 for i=1:n
42 [x,Fval,exitFlag] = ga(@fun,14,Ap,bp,[],[],lb,ub,[],[],opts)
43 %[x,Fval,exitFlag] = ga(@fun,7,Ap,bp,[],[],lb,ub,@mycon,[],opts);
44
45
46 T1(i)=x(1);
47 T2(i)=x(2);
```

```

48 T3(i)=x(3);
49 T4(i)=x(4);
50 vH1(i)=x(5);
51 vD1(i)=x(6);
52 v0(i)=x(7);
53 vH2(i)=x(8);
54 vD2(i)=x(9);
55 svH1(i)=x(10);
56 svD1(i)=x(11);
57 sv0(i)=x(12);
58 svH2(i)=x(13);
59 svD2(i)=x(14);
60
61
62 %Vykresleni vyledku
63 for in=1:length(t)
64     if t(in)<T1
65 v(in) = vH1+svH1*t(in);
66     elseif t(in)>=T1 & t(in)<T2
67 v(in) = vD1+svD1*t(in);
68     elseif t(in)>=T2 & t(in)<T3
69 v(in) = v0+sv0*t(in);
70     elseif t(in)>=T3 & t(in)<T4
71 v(in) = vD2+svD2*t(in);
72     else
73 v(in) = vH2+svH2*t(in);
74     end
75 end
76
77
78 [y,t,x]=lsim(sys1,v,t,x0);
79
80 uoc=7*(k_0*z5.^8+k_1*z5.^7+k_2*z5.^6+k_3*z5.^5+k_4*z5.^4+k_5*z5.^3+
    k_6*z5.^2+k_7*z5+k_8);
81
82 v_max(:,i)=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y
    (:,5)))*a23;
83 v_min(:,i)=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y
    (:,5)))*a23;
84 v_0(:,i)=((-a32*y(:,2)-a33*y(:,3)/a23))*a23;
85
86
87 z0(:,i)=y(:,1);
88 z1(:,i)=y(:,2);
89 z2(:,i)=y(:,3);
90 z3(:,i)=y(:,5);
91
92 v_i(:,i)=y(:,4);
93 x3(:,i)=y(:,3)/a23;
94 z4_v(:,i)=1-a53*abs(y(:,3)/a23);
95
96 u(:,i)=(y(:,4)/a23+a32*y(:,2)+a33*y(:,3)/a23)./(a3u*(uoc-c1*y(:,3)/
    a23-c2*y(:,5)));
97
98
99 J(:,i)=abs((v_i(:,i)+a23*a32*z1(:,i)+a32*z2(:,i)).*z2(:,i))/(a23

```

```

    ^2)*a3u)+0.2;
100
101 W(i)=trapz(t,J(:,i))
102
103
104 end
105
106
107
108 function f=fun(p)
109 global A B C D x0 t sys1 T fi_F a23 a32 a3u T a23 a32 a33 a44 a53 y
    a33 uoc a3u c1 c2
110 T1=p(1);
111 T2=p(2);
112 T3=p(3);
113 T4=p(4);
114 vH1=p(5);
115 vD1=p(6);
116 v0=p(7);
117 vH2=p(8);
118 vD2=p(9);
119 svH1=p(10);
120 svD1=p(11);
121 sv0=p(12);
122 svH2=p(13);
123 svD2=p(14);
124
125 Pen=0;
126
127 for in=1:length(t)
128     if t(in)<T1
129 v(in) = vH1+svH1*t(in);
130     elseif t(in)>=T1 & t(in)<T2
131 v(in) = vD1+svD1*t(in);
132     elseif t(in)>=T2 & t(in)<T3
133 v(in) = v0+sv0*t(in);
134     elseif t(in)>=T3 & t(in)<T4
135 v(in) = vD2+svD2*t(in);
136     else
137 v(in) = vH2+svH2*t(in);
138     end
139 end
140
141
142 [y,t,x]=lsim(sys1,v,t,x0);
143
144 z0(:)=y(:,1);
145 z1(:)=y(:,2);
146 z2(:)=y(:,3);
147 z3(:)=y(:,5);
148 z4(:)=y(:,6);
149 v_i(:)=y(:,4);
150 v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;
151 v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;

```



```

152
153 J(:)=abs((v_i(:)+a23*a32*z1(:)+a32*z2(:)).*z2(:))/((a23^2)*a3u
    +0.2);
154
155 W=trapz(t,J(:))
156
157 for in=1:length(v_i)
158     if any(v_i(in)>(v_max(in)+50)|v_i(in)<(v_min(in)-50))
159
160         Pen=200000;
161     end
162 end
163
164 f= 50*(fi_F-z0(end))^2+(z1(end)^2)+(z2(end))^2+100*W+Pen
165 end

```

Algoritmus 4: Genetický algoritmus pro nalezení optimálního řízení pomocí přímkové aproximace

Příloha 4

```
1
2 %opts = optimoptions(@ga,'PopulationSize',200,'EliteCount',5,'
   FunctionTolerance',10e-5,'PlotFcn',@gaplotbestf,'
   InitialPopulationMatrix',[0.25,0.5,1.5,1.75,4500,-4500]);
3 opts = optimoptions(@ga,'PopulationSize',500,'EliteCount',10,'
   FunctionTolerance',10e-7,'PlotFcn',@gaplotbestf,'
   InitialPopulationMatrix',[0.1 0.4 1.4 1.9]);
4 %options = bvpset("Stats","on","RelTol",1e-4);
5 T=2;
6 fi_F=450;
7
8 Ap = [1 -1 0 0 ;
9       0 1 -1 0 ;
10      0 0 1 -1];
11 bp = [0;0;0];
12 lb = [0; 0; 0.25*T; 0.25*T];
13 ub = [0.25*T; 0.5*T; T; T];
14 z5=1;
15 global A B C D x0 t sys1 T A B C D a23 a32 a33 a44 a53 y a33 uoc
   a3u c1 c2 fi_F
16
17 %Linearni system
18 A = [0 1 0 0 0;
19       0 0 1 0 0;
20       0 0 0 0 0;
21       0 0 a43/a23 -a44 0;
22       0 0 -a53 0 0];
23 B = [0; 0; 1; 0; 0];
24 C = [1 0 0 0 0;
25       0 1 0 0 0;
26       0 0 1 0 0;
27       0 0 0 0 0;
28       0 0 0 1 0;
29       0 0 0 0 1];
30 D = [0 ;0; 0;1;0;0];
31
32 sys1 = ss(A,B,C,D);
33
34 x0 = [0 0 0 0 1];
35 y=x0;
36 t = 0:0.01:2;
37 v = zeros(length(t),1);
38
39 n=1
40 for i=1:n
41 [x,Fval,exitFlag] = ga(@fun,4,Ap,bp,[],[],lb,ub,[],[],opts)
42 %[x,Fval,exitFlag] = ga(@fun,7,Ap,bp,[],[],lb,ub,@mycon,[],opts);
43
44 T1(i)=x(1);
45 T2(i)=x(2);
46 T3(i)=x(3);
47 T4(i)=x(4);
48
49 vH=7000;
```

```

50 vD=-7000;
51
52 v(t<0.2) = vH;
53 v(t>=0.2) = vD;
54 v(t>=2*0.2) = 0;
55 v(t>=2-2*0.2) = vD;
56 v(t>=2-0.2) = vH;
57
58 [y,t,x]=lsim(sys1,v,t,x0);
59
60 z0(:)=y(:,1);
61 z1(:)=y(:,2);
62 z2(:)=y(:,3);
63 z3(:)=y(:,5);
64 z4(:)=y(:,6);
65 v_i(:)=y(:,4);
66 v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
        )*a23;
67 v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
        )*a23;
68 v_0=(-a32*y(:,2)-a33*y(:,3)/a23)*a23;
69
70
71 for in=1:length(t)
72     if t(in)<T1
73         v(in) = v_max(in);
74     elseif t(in)>=T1 & t(in)<T2
75         v(in) = v_0(in);
76     elseif t(in)>=T2 & t(in)<T3
77         v(in) = 0;
78     elseif t(in)>=T3 & t(in)<T4
79         v(in) = v_0(in);
80     else
81         v(in) = v_max(in);
82     end
83 end
84
85 [y,t,x]=lsim(sys1,v,t,x0);
86
87 z0(:)=y(:,1);
88 z1(:)=y(:,2);
89 z2(:)=y(:,3);
90 z3(:)=y(:,5);
91 z4_v(:)=y(:,6);
92 v_i(:)=y(:,4);
93 v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
        )*a23;
94 v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
        )*a23;
95 v_0=(-a32*y(:,2)-a33*y(:,3)/a23)*a23;
96
97 u(:)=(y(:,4)/a23+a32*y(:,2)+a33*y(:,3)/a23)./(a3u*(uoc-c1*y(:,3)/
        a23-c2*y(:,5)));
98 x3(:,i)=y(:,3)/a23;
99 J(:)=abs((v_i(:)+a23*a32*z1(:)+a32*z2(:)).*z2(:))/((a23^2)*a3u
        +0.2);

```

```

100
101 W=trapz(t,J(:))
102
103 end
104
105
106 function f=fun(x)
107 global A B C D x0 t sys1 T fi_F a23 a32 a3u T a23 a32 a33 a44 a53 y
    a33 uoc a3u c1 c2
108 T1=x(1);
109 T2=x(2);
110 T3=x(3);
111 T4=x(4);
112
113 Pen=0;
114
115 vH=7000;
116 vD=-7000;
117
118 v(t<0.2) = vH;
119 v(t>=0.2) = vD;
120 v(t>=2*0.2) = 0;
121 v(t>=2-2*0.2) = vD;
122 v(t>=2-0.2) = vH;
123 [y,t,x]=lsim(sys1,v,t,x0);
124
125 z0(:)=y(:,1);
126 z1(:)=y(:,2);
127 z2(:)=y(:,3);
128 z3(:)=y(:,5);
129 z4(:)=y(:,6);
130 v_i(:)=y(:,4);
131 v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;
132 v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;
133 v_0=(-a32*y(:,2)-a33*y(:,3)/a23)*a23;
134
135 for in=1:length(v_i)
136     if t(in)<T1
137 v(in) = v_max(in);
138     elseif t(in)>=T1 & t(in)<T2
139 v(in) = v_0(in);
140     elseif t(in)>=T2 & t(in)<T3
141 v(in) = 0;
142     elseif t(in)>=T3 & t(in)<T4
143 v(in) = v_0(in);
144     else
145 v(in) = v_max(in);
146     end
147 end
148
149 [y,t,x]=lsim(sys1,v,t,x0);
150
151 z0(:)=y(:,1);
152 z1(:)=y(:,2);

```

```

153 z2(:)=y(:,3);
154 z3(:)=y(:,5);
155 z4(:)=y(:,6);
156 v_i(:)=y(:,4);
157 v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;
158 v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;
159 v_0=(-a32*y(:,2)-a33*y(:,3)/a23)*a23;
160
161 J(:)=abs((v_i(:)+a23*a32*z1(:)+a32*z2(:)).*z2(:))/((a23^2)*a3u
    +0.2);
162
163 W=trapz(t,J(:))
164
165 for in=1:length(v_i)
166     if any(v_i(in)>(v_max(in)+100)|v_i(in)<(v_min(in)-100))
167
168         Pen=200000;
169     end
170 end
171
172 f=20*(fi_F-z0(end))^2+(z1(end)^2)+(z2(end))^2+100*W+Pen
173 end

```

Algoritmus 5: Genetický algoritmus pro nalezení přepínacích časů na hodnoty v_{max} a v_0

Příloha 5

```

1
2 %opts = optimoptions(@ga,'PopulationSize',200,'EliteCount',5,'
   FunctionTolerance',10e-5,'PlotFcn',@gaplotbestf,'
   InitialPopulationMatrix',[0.25,0.5,1.5,1.75,4500,-4500]);
3 opts = optimoptions(@ga,'PopulationSize',1000,'EliteCount',20,'
   FunctionTolerance',10e-7,'PlotFcn',@gaplotbestf,'
   InitialPopulationMatrix',[v_start1]);
4
5 T=2;
6 fi_F=450;
7
8 z5=1;
9 global A B C D x0 t sys1 T A B C D a23 a32 a33 a44 a53 y a33 uoc
   a3u c1 c2 fi_F
10
11 %Linearni system
12 A = [0 1 0 0 0;
13       0 0 1 0 0;
14       0 0 0 0 0;
15       0 0 a43/a23 -a44 0;
16       0 0 -a53 0 0];
17 B = [0; 0; 1; 0; 0];
18 C = [1 0 0 0 0;
19       0 1 0 0 0;
20       0 0 1 0 0;
21       0 0 0 0 0;
22       0 0 0 1 0;
23       0 0 0 0 1];
24 D = [0 ;0; 0;1;0;0];
25
26 sys1 = ss(A,B,C,D);
27
28 x0 = [0 0 0 0 1];
29 y=x0;
30 t = 0:0.01:2;
31 v = zeros(length(t),1);
32
33 n=1
34 for i=1:n
35 [x,Fval,exitFlag] = ga(@fun,length(t),[],[],[],[],[],[],[],[],opts)
36 %[x,Fval,exitFlag] = ga(@fun,7,Ap,bp,[],[],lb,ub,@mycon,[],opts);
37
38 for i=1:length(t)
39 %Vykresleni vyledku
40 v = x(i);
41 end
42 [y,t,x]=lsim(sys1,v,t,x0);
43
44 uoc=7*(k_0*z5.^8+k_1*z5.^7+k_2*z5.^6+k_3*z5.^5+k_4*z5.^4+k_5*z5.^3+
   k_6*z5.^2+k_7*z5+k_8);
45
46 v_max(:,i)=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y
   (:,5))*a23);
47 v_min(:,i)=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y

```

```

        (:,5))*a23;
48 v_0(:,i)=((-a32*y(:,2)-a33*y(:,3)/a23))*a23;
49
50 z0(:,i)=y(:,1);
51 z1(:,i)=y(:,2);
52 z2(:,i)=y(:,3);
53 z3(:,i)=y(:,5);
54
55 v_i(:,i)=y(:,4);
56 x3(:,i)=y(:,3)/a23;
57 z4_v(:,i)=1-a53*abs(y(:,3)/a23);
58
59 u(:,i)=(y(:,4)/a23+a32*y(:,2)+a33*y(:,3)/a23)./(a3u*(uoc-c1*y(:,3)/
    a23-c2*y(:,5)));
60
61 J(:,i)=abs((v_i(:,i)+a23*a32*z1(:,i)+a32*z2(:,i)).*z2(:,i))/((a23
    ^2)*a3u)+0.2;
62
63 W(i)=trapz(t,J(:,i))
64
65 end
66
67
68 function f=fun(x)
69 global A B C D x0 t sys1 T fi_F a23 a32 a3u T a23 a32 a33 a44 a53 y
    a33 uoc a3u c1 c2
70
71 for i=1:length(t)
72 %Vykresleni vysledku
73 v(i) = x(i);
74 end
75 Pen=0;
76
77
78 [y,t,x]=lsim(sys1,v,t,x0);
79
80 z0(:)=y(:,1);
81 z1(:)=y(:,2);
82 z2(:)=y(:,3);
83 z3(:)=y(:,5);
84 z4(:)=y(:,6);
85 v_i(:)=y(:,4);
86 v_max=(-a32*y(:,2)-a33*y(:,3)/a23+a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;
87 v_min=(-a32*y(:,2)-a33*y(:,3)/a23-a3u*(uoc-c1*y(:,3)/a23-c2*y(:,5))
    )*a23;
88
89 J(:)=abs((v_i(:)+a23*a32*z1(:)+a32*z2(:)).*z2(:))/((a23^2)*a3u
    +0.2;
90
91 W=trapz(t,J(:))
92
93 for in=1:length(v_i)
94 if any(v_i(in)>(v_max(in)+100)|v_i(in)<(v_min(in)-100))
95
96     Pen=200000;

```

```
97 end
98 end
99
100 f= 20*(fi_F-z0(end))^2+(z1(end)^2)+(z2(end))^2+100*W+Pen
101 end
```

Algoritmus 6: Genetický algoritmus pro nalezení optimálního řízení