SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta chemickej a potravinárskej technológie

Evidenčné číslo: FCHPT-5414-61829

MPC pre autonómne vozidlá s obchádzaním prekážok

Diplomová práca

2016 **Bc. Filip Janeček**

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta chemickej a potravinárskej technológie

Evidenčné číslo: FCHPT-5414-61829

MPC pre autonómne vozidlá s obchádzaním prekážok

Diplomová práca

Študijný program: automatizácia a informatizácia v chémii a potravinárstve

Študijný odbor: 5.2.14 automatizácia

Školiace pracovisko: Ústav informatizácie, automatizácie a matematiky

Vedúci záverečnej práce: doc. Ing. Michal Kvasnica, PhD.

Bratislava, 2016 **Bc. Filip Janeček**

Slovenská technická univerzita v Bratislave
Ústav informatizácie, automatizácie a matematiky

Fakulta chemickej a potravinárskej technológie
Akademický rok: 2015/2016
Evidenčné číslo: FCHPT-5414-61829

ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Filip Janeček**
ID študenta: 61829
Študijný program: automatizácia a informatizácia v chémii a potravinárstve
Študijný odbor: 5.2.14. automatizácia
Vedúci práce: doc. Ing. Michal Kvasnica, PhD.

Názov práce: **MPC pre autonómne vozidlá s obchádzaním prekážok**

Špecifikácia zadania:

Cieľom práce je navrhnúť a simulačne overiť prediktívny regulátor na riadenie autonómnych vozidiel, ktoré musia obchádzať prekážky. Pri návrhu MPC regulátora študent preskúma výpočtovú zložitosť rôznych formulácií problému obchádzania prekážok. Matematická formulácia bude založená na opise povoleného okolia prekážok pomocou binárnych premenných. Výpočtová zložitosť sa zníži pridaním doplnkových ohraničení, ktore obmedzia veľkosť prehľadávacieho priestoru.

Čiastkové úlohy:
* matematické modelovanie autonómneho vozidla
* matematická formulácia MPC problému s obchádzaním polytopických prekážok
* analýza vypočtovej zložitosti
* simulačné overenie funkčnosti regulátora

Rozsah práce: 50

Riešenie zadania práce od: 15. 02. 2016
Dátum odovzdania práce: 22. 05. 2016

L. S.

Bc. Filip Janeček
študent

prof. Ing. Miroslav Fikar, DrSc.
vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.
garant študijného programu

Poďakovanie

Týmto by som sa chcel poďakovať vedúcemu diplomovej práce doc. Ing. Michalovi Kvasnicovi, PhD. za odbornú pomoc, dobre mienené rady a pripomienky k mojej práci a Ing. Martinovi Klaučovi za pomoc pri dokončovaní mojej práce.

Abstrakt

V diplomovej práci sme riešili problematiku prediktívneho riadenia pre autonómne vozidlá, ktoré budú schopné detekovať a následne sa vyhýbať prekážkam. Vypočítaná trasa bude zároveň optimálna pre výpočet s použitím binárnych premenných vzhľadom na dĺžku trasy a minutú energiu. Pri použití časovo premenných ohraničení očakávame zníženie výpočtovej náročnosti za cenu straty garantovanej optimality riešenia.

Prvá kapitola je zameraná na teoretické priblíženie prediktívneho riadenia, jeho výhod a nevýhod, podmienok pre použitie a sú tu priblížené niektoré pokročilé formulácie.

V druhej kapitole sa venujeme matematickému modelovaniu vozidla ako sústavy dvojitých integrátorov, definíciou prekážky ako konvexnej množiny (polytopu) a formuláciou optimalizačných problémov pre riešenie s použitím binárnych premenných a rovnako pre použitie časovo premenlivých ohraničení.

Výsledky simulácii sa nachádzajú v poslednej kapitole, kde sme sa pozreli na dva scenáre - vyhýbanie sa statickým alebo pohyblivým prekážkam. Na základe hodnôt účelovej funkcie a výpočtového času sme porovnali dva prístupy k riadeniu spomínané vyššie - binárne premenné a časovo premenlivé ohraničenia.

Kľúčové slová: MPC; obchádzanie prekážok; zmiešané celočíselné kvadratické programovanie

Abstract

In diploma thesis we were concerned in Model Predictive Control for autonomous vehicles with obstacle avoidance. Computed trajectory will be optimal for calculation which uses binary variables considering length of trajectory and consumed energy. With usage of time-varying constraints, we expect reduction of computing time, but also loss of guaranty of optimality of the solution.

The first chapter is dedicated to theoretical aspects of Model Predictive Control, which are its advantages and disadvantages, conditions and some advanced techniques are also described.

In the second chapter, we are looking at mathematical modeling of vehicle as system of two integrators, definition of obstacle as convex set (polytope) and formulation of optimization problems with use of binary variables and also with use of time-varying constraints.

The results of simulations are contained in the last chapter, where we were studied two scenarios - static or moving obstacle avoidance. Considering values of objective function and computational time we compared two control approaches mentioned above - binary variables and time-varying constraints.

Key words: MPC; obstacle avoidance; mixed-integer quadratic programming

Obsah

Úvod

V dnešnej dobe sa čoraz častejšie stretávame so snahou automatizovať čo možno najviac úkonov, či ide o domácnosť, priemysel alebo dopravu. V našej práci sa budeme zaoberať treťou spomínanou oblasťou, teda využitím automatizácie pri riadení vozidiel.

Dôvody pre riešenie tejto problematiky sú napríklad zefektívnenie dopravy, teda skrátenie času a dĺžky trasy, úspora paliva, zvýšenie bezpečnosti a v neposlednom rade zvýšenie pohodlia pre cestujúcich.

Cieľom tejto práce je navrhnúť prediktívny regulátor na riadenie autonómnych vozidiel, ktoré budú detekovať a obchádzať prekážky. Pre matematickú formuláciu tohto problému bude najprv využitý opis povoleného okolia prekážok pomocou binárnych premenných. Výpočtovú zložitosť tohto problému porovnáme s prístupom, kedy na obchádzanie prekážok bude využitá zmena ohraničení v reálnom čase. Pri tomto druhom prístupe predpokladáme kratší výpočtový čas, no na úkor straty garantovanej optimality.

Dôvodov pre výber *prediktívneho riadenia* (skrátene *MPC*, z anglického *Model Predictive Control*) na riešenie nášho problému je viacero, tie hlavné sú, že MPC garantuje optimálne riešenie a dovoľuje implementáciu ohraničení priamo do zadania problému.

V prvej časti si najprv priblížime MPC z teoretického hľadiska, aké podmienky musíme dodržať, ak chceme toto riadenie použiť a aké sú jeho výhody aj nevýhody. Vysvetlíme si prepojenie MPC s *kvadratickým programovaním* (skrátene *QP*, z anglického *Quadratic Programming*), ukážeme si prevod MPC formulácie problému na QP formuláciu a opíšeme si niektoré pokročilé formulácie MPC.

Ďalej sa budeme venovať matematickému modelovaniu autonómneho vozidla, na čo bude využitá sústava dvojitých intgrátorov. Všeobecne definujeme prekážku ako konvexnú množinu, konkrétne polytop. Tiež si vysvetlíme spôsob riešenia nášho problému pomocou *zmiešaného celočíselného kvadratického programovania* (skrátene *MIQP*, z anglického *Mixed-Integer Quadratic Programming*), priblížime si MIQP matematickú formuláciu MPC problému s obchádzaním prekážok a porovnáme ju s QP matematickou formuláciou MPC problému pre časovo premenlivé ohraničenia.

V tretej časti simulačne overíme funkčnosť regulátora na dvoch scenároch - so statickými prekážkami a s pohyblivými prekážkami. Jednu konkrétnu situáciu si opíšeme detailne. Tiež v tejto časti porovnáme časy potrebné na výpočet pre oba spomenuté prístupy - s využitím binárnych premenných, teda MIQP a zmenou ohraničení v reálnom čase, teda QP.

Súčasťou tejto práce je využitie a rozšírenie Matlabovskej triedy *optiplan*, voľne stiahnuteľnej na webovej adrese: `https://bitbucket.org/kvasnica/optiplan/wiki/Home`.

Kapitola 1

Základná teória k MPC

MPC je pokročilý prístup k procesnému riadeniu. Do popredia sa začal dostávať niekedy v osemdesiatych rokoch minulého storočia hlavne v chemických továrňach a ropných rafinériách. Neskôr sa začal využívať aj v oblasti obnoviteľných zdrojov energie. Tu bolo potrebné nájsť takú formu riadenia, ktorá bude schopná pracovať v reálnom čase, bude rátať s dynamikou systému a predikciou neurčitostí, (Arnold a Andersson, 2011). MPC regulátory využívajú dynamické modely procesu, najčastejšie lineárne empirické modely získané z identifikácie systému. Hlavná výhoda MPC je fakt, že umožňuje optimalizovať aktuálny časový úsek so zreteľom na budúce časové úseky. Toto je docielené optimalizáciou na konečnom časovom horizonte, implementujúc iba prvý časový úsek. MPC má schopnosť predvídať budúce udalosti a tak môže adekvátne reagovať. Veľkou výhodou MPC je taktiež jeho schopnosť priamo pracovať s obmedzeniami vstupov, stavov a výstupov, ktoré sa v reálnom živote vyskytujú veľmi často, prakticky vždy. PID a LQR regulátory nemajú túto predikčnú schopnosť a nedokážu efektívne pracovať s obmedzeniami. MPC je takmer univerzálne implementovateľé ako digitálne riadenie, avšak pracuje sa na výskume ako docieliť rýchlejšie odozvy so špeciálne dizajnovaným analógovým okruhom, (Vichik a Borrelli, 2014).

1.1 Matematické modely

Modely používané v MPC by mali reprezentovať správanie komplexných dynamických systémov. Táto komplexnosť vyžadovaná v MPC algoritmoch nie je všeobecne vyžadovaná na uspokojivé riadenie jednoduchých systémov, ktoré sú často dobre riadené klasickými

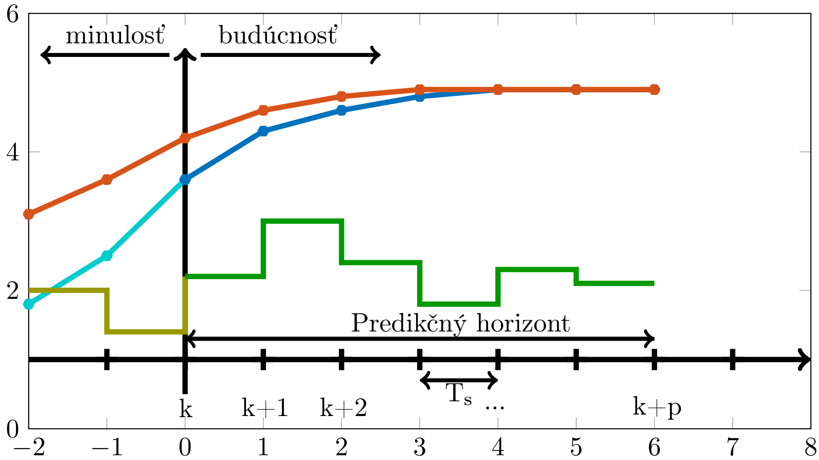
PID regulátormi.

MPC modely predikujú zmeny v závislých premenných modelovaného systému, ktoré sú spôsobené zmenami nezávislých premenných. Napríklad pri riadení vozidiel, nezávislé premenné, ktoré môžu byť upravované regulátorom sú často žiadané hodnoty regulátorov (vzdialenosť, rýchlosť, ...) alebo riadiace (akčné) členy (zrýchlenie, natočenie kolies, ...). Nezávislé premenné, ktoré nemôžu byť upravované regulátorom sú označované ako poruchy. Závislé premenné v týchto procesoch sú ostatné merania predstavujúce buď ciele riadenia alebo procesné ohraničenia.

MPC používa aktuálne merania stavov systému, aktuálnu dynamiku procesu, MPC modely a žiadané hodnoty a ohraničenia procesných premenných na výpočet budúcich zmien závislých premenných. Tieto zmeny sú vypočítavané kvôli udržaniu závislých premenných blízko referencie pri dodržaní ohraničení ako pre závislé tak pre nezávislé premenné. MPC posiela iba prvú vypočítanú zmenu v nezávislých premenných na implementáciu, a v ďalšom kroku zopakuje celý výpočet.

Napriek tomu, že mnohé reálne procesy nie sú lineárne, môžu byť považované za takmer lineárne v malom okolí pracovného bodu. Lineárne MPC je vo vačšine aplikácii používané spolu so spätnoväzbovým mechanizmom MPC, ktorý kompenzuje predikčnú chybu spôsobenú rozdielom medzi modelom a reálnym procesom. V prediktívnych regulátoroch, ktoré obsahujú len lineárne modely, princíp superpozície lineárnej algebry umožňuje spojiť zmeny viacerých nezávislých premenných dokopy a predikovať odozvu závislých premenných. Toto zjednodušuje problém riadenia na sekvenciu priamych maticových algebraických výpočtov, ktoré sú rýchle a robustné.

V prípade, že lineárne modely nie sú dostatočne presné na reprezentáciu nelinearít reálneho procesu, používajú sa viaceré prístupy. Proces môže byť riadený nelineárnym MPC, ktoré používa nelineárny model priamo v riadiacej aplikácii. Nelineárny model môže byť vo forme empirických dát (napr. umelé neurónové siete) alebo vysokopresné dynamické modely založené na základných materiálových a energetických bilanciách. Nelineárne modely môžu byť linearizované na odvodenie *Kalmanovho filtra* alebo na určenie modelu pre lineárne MPC.



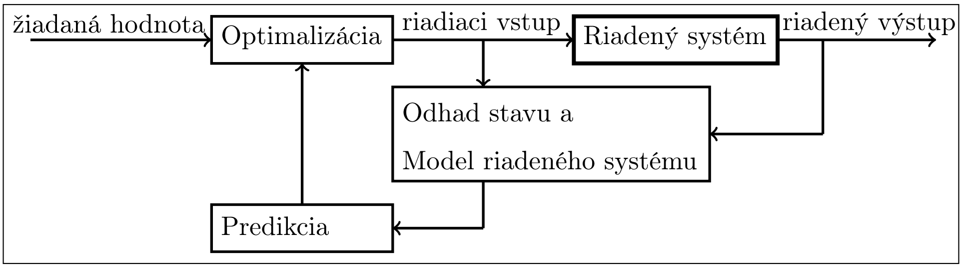
Obr. 1.1: Fungovanie MPC.

referenčná trajektória, predikovaný výstup, meraný výstup, predikovaný akčný zásah, minulý akčný zásah

1.2 MPC z teoretického hľadiska

MPC je založené na iteratívnej optimalizácii modelu systému s konečným horizontom. V čase k sú zaznamenané aktuálne stavy systému a je vypočítané minimalizačné kritérium (*účelová funkcia*) minimalizačným algoritmom pre relatívne krátky časový horizont (*predikčný horizont*) do budúcnosti: $[k, k+p]$. Presnejšie, na preskúmanie stavových trajektórií vychádzajúcich z aktuálneho stavu a nájdenie (výpočtom *Eulerových-Lagrangeových rovníc*) sekvencie riadenia minimalizujúceho účelovú funkciu do času $k+p$ je použitý výpočet v reálnom čase. Implementovaný je iba prvý krok tejto sekvencie riadenia, znova sa vykonajú merania a výpočet sa opakuje od nového aktuálneho stavu, získavajúc novú sekvenciu riadenia a predikovanú stavovú trajektóriu. Predikčný horizont sa posúva dopredu, a preto sa tento prístup nazýva aj *stratégia pohyblivého horizontu*. Na obr. 1.1 vidíme ako je počítaná sekvencia riadiacich vstupov na základe známych meraní dostupných v čase k a na základe známej referenčnej trajektórie pre výstupnú veličinu až do času $k+p$, kde p je predikčný horizont.

Bolo realizovaných viacero akademických štúdií na nájdenie rýchlych metód pre riešenie Eulerových-Lagrangeových rovníc a všeobecne na zlepšenie MPC riadenia. Teoretici sa do istej miery snažia dobehnúť riadiacich inžinierov, pokiaľ ide o MPC, (Nikolaou, 2001).



Obr. 1.2: Štruktúra stavovej reprezentácie MPC.

1.3 Základná štruktúra MPC regulátora

MPC je vo všeobecnosti stavové riadenie, ktorého reprezentáciu môžme vidieť na obr. 1.2.

Jednotlivé bloky a ich funkcie:

- *Optimalizácia* - obsahuje obmedzenia a jej hlavným znakom je, že pracuje nepretržite. Podľa referenčnej hodnoty pre výstup vypočíta sekvenciu riadiacich vstupov tak, aby riadený výstup čo najlepšie sledoval referenciu.
- *Odhad stavu a Model riadeného systému* - obsahuje pozorovač stavov, ktorý na základe vstupov a meraných výstupov odhaduje stavy nášho systému. Ďalej je tu obsiahnutý matematický model nášho riadeného systému.
- *Predikcia* - obsahuje algoritmy, ktoré predikujú budúce hodnoty vstupných a stavových veličín

Tieto bloky tvoria základnú štruktúru MPC regulátora. Pre rôzne riešené problémy môžu byť tieto bloky upravené, prípadne môžu byť pridané ďalšie, napr. odhad poruchy, identifikácia systému, atď, (Karas a kol., 2007).

1.4 Ohraničenia

Keďže ohraničenia (alebo obmedzenia) sú v reálnom živote, a teda aj v reálnych procesoch prítomné takmer všade, je dôležité s nimi rátať aj pri riadení týchto procesov. Pri ignorovaní ohraničení môže totiž dôjsť v lepšom prípade k nefunkčnému riadeniu, v horšom prípade to môže byť zdraviu alebo dokonca životu nebezpečné. Ak by sme napr. chceli naplniť zásobník kvapaliny do určitej hladiny, a neimplementovali by sme nejakým spôsobom ohraničenia na priemer potrubia a kapacitu ventilu, tak na simulácii nám môže všetko pekne fungovať, no pri zapojení na reálne zariadenie zistíme, že regulátor posiela do akčných členov nerealizovateľné akčné zásahy.

Horšia situácia by mohla nastať pri riadení reaktora, kde ak by sme nebrali do úvahy obmedzenia na hraničné bezpečné teploty, tak sa ľahko môže stať, že sa reaktor prehreje a hrozí výbuch.

Preto je MPC tak výhodné, lebo priamo dokáže implementovať a pracovať s ohraničeniami bez straty kvality riadenia a s garanciou dodržania obmedzení. Pri iných prístupoch k riadeniu by bolo možné orezať akčné zásahy, ale tým sa stráca kvalita riadenia a v niektorých prípadoch by sa mohlo stať, že referenčná hodnota výstupu nebude dosiahnuteľná.

Obmedzenia môžu byť tiež využívané ako ladiace parametre regulátora na dosiahnutie lepšej kvality riadenia, (Karas a kol., 2007).

Z hľadiska charakteru môžme ohraničenia rozdeliť do týchto skupín, (Kuznetsov, 1996):

- *Fyzikálne ohraničenia* - hodnoty veličín sa môžu pohybovať len vo fyzikálne realizovateľných medziach a meniť sa môžu len fyzikálne realizovateľnou rýchlosťou (napr. nie je možné, aby sme obyčajným laboratórnym horákom zohriali $1\,l$ $20\,°\mathrm{C}$ vody na $80\,°\mathrm{C}$ vodu za $2\,s$)

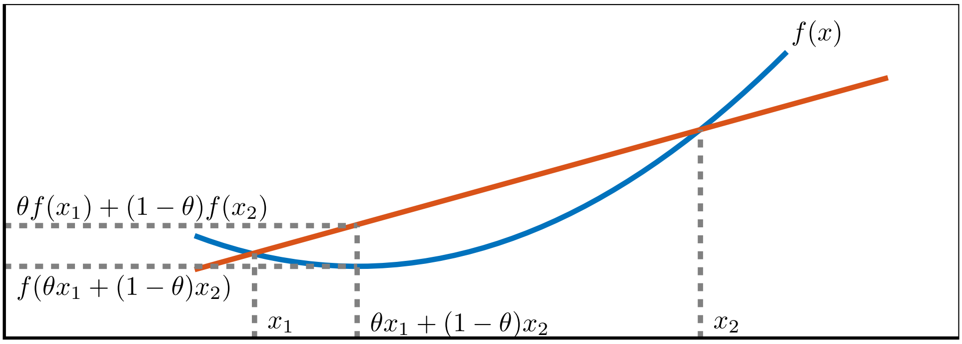
- *Technologické ohraničenia* - na dodržanie požadovanej kvality produkcie nesmú hodnoty stavov a výstupov prekračovať vopred určené hranice, musia sa nachádzať v prípustnom rozmedzí

- *Bezpečnostné ohraničenia* - aby nedošlo k havárii zariadenia, pomocné veličiny nesmú prekročiť svoje bezpečné limity

- *Stabilizujúce ohraničenia* - aby bola dodržaná stabilita systému pri riadení, musia sa stavy nachádzať v stabilných hraniciach

1.5 Konvexná optimalizácia v MPC

Ak chceme navrhnúť a zaviezť MPC riadenie, musíme brať do úvahy jeho pomerne vysokú výpočtovú náročnosť v porovnaní s inými spôsobmi riadenia. Náš hardvér musí zvládnuť vypočítať hodnotu riadiaceho vstupu za čas kratší alebo nanajvýš rovný perióde vzorkovania, teda v každej perióde vzorkovania musí byť vyriešený optimalizačný problém so všetkými implementovanými obmedzeniami.



Obr. 1.3: Graf konvexnej funkcie.

Preto sa snažíme náš problém vždy formulovať ako konvexný. To znamená, že naša účelová funkcia aj obmedzenia spĺňajú nasledujúcu podmienku konvexnosti funkcie:

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2) \tag{1.1}$$

a táto nerovnosť musí byť platná pre všetky $x_1, x_2 \in \mathbb{R}^n$ a $\theta \in [0,\ 1]$, (Boyd a Vandenberghe, 2004).

Na obr. 1.3 sa nachádza graf kvadratickej funkcie, ktorá spĺňa nerovnosť (1.1), preto môžme povedať, že je konvexná.

Na to, aby sme mohli naformulovať problém ako konvexnú optimalizáciu, tak nie len samotná účelová funkcia musí byť konvexná, ale aj jej množina prípustných hodnôt, teda definičný obor. Zistiť, či je nejaký definičný obor konvexný, nie je vo všeobecnosti jednoduché. Existujú ale množiny, ktorých konvexnosť vieme určiť pomerne jednoducho.

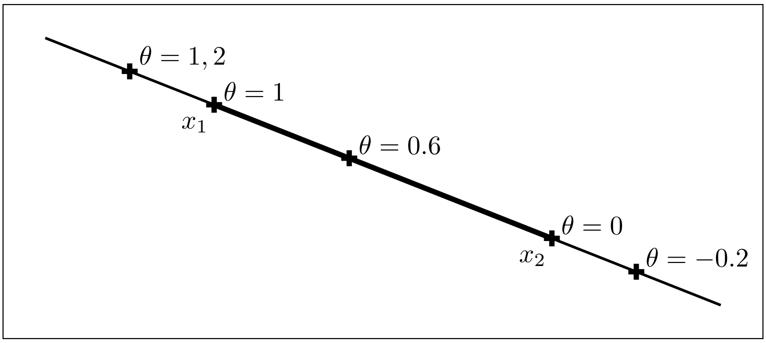
O množine C môžme povedať, že je *konvexná*, ak pre dva ľubovoľné body patriace do C platí, že úsečka medzi týmito dvoma bodmi celá leží v množine C, teda pre ľubovoľné $\theta \in [0,\ 1]$ a ľubovoľné $\mathbf{x}_1,\ \mathbf{x}_2 \in C$ platí:

$$\theta \mathbf{x}_1 + (1-\theta)\mathbf{x}_2 \in C \tag{1.2}$$

1.5.1 Afinné množiny

Osobitným typom konvexných množín sú afinné množiny. *Afinná* množina je taká, ak existuje priamka spájajúca dva ľubovoľné body patriace do množiny C taká, že leží v C (Obr. 1.4). Všeobecne pre ľubovoľne veľa bodov platí, že ak $\mathbf{x}_1,\ \mathbf{x}_2 \in C$ a $\theta_1 + \cdots + \theta_k = 1$, tak

$$\theta_1 \mathbf{x}_1 + \cdots + \theta_k \mathbf{x}_k \in C \tag{1.3}$$



Obr. 1.4: Znázornenie afinnej množiny.

Priamka p prechádza bodmi $\mathbf{x}_1$, $\mathbf{x}_2$, množina C je konvexná, ak $\mathbf{x}_1$, $\mathbf{x}_2 \in C$ a p leží v C.

1.5.2 Nadroviny

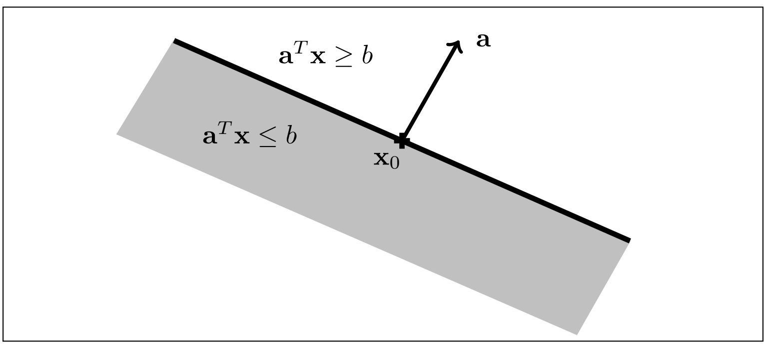
Nadroviny sú ďalšou významnou skupinou množín, pre ktoré platí:

$$\{\mathbf{x}|\mathbf{a}^T\mathbf{x} = b\} \tag{1.4}$$

kde

$$\mathbf{a} \in \mathbb{R}^n, \quad \mathbf{a} \neq \mathbf{0}, \quad b \in \mathbb{R}$$

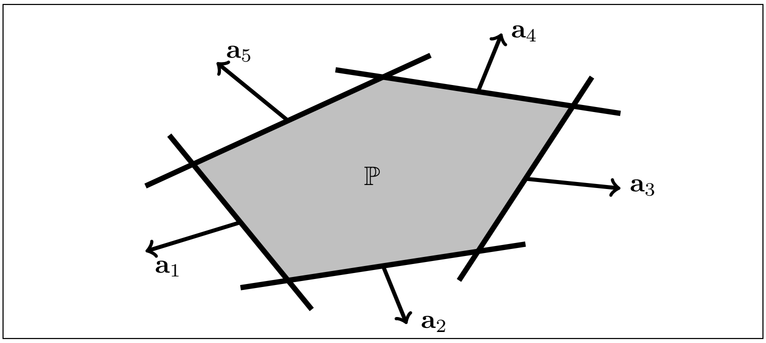
Priestor je nadrovinou rozdelený na dva *polpriestory* $\{\mathbf{x}|\mathbf{a}^T\mathbf{x} \leq b\}$ a $\{\mathbf{x}|\mathbf{a}^T\mathbf{x} \geq b\}$, ako to vidíme na obr. 1.5.



Obr. 1.5: Polpriestory v $\mathbb{R}^2$ tvorené nadrovinou $\mathbf{a}^T\mathbf{x} = b$.

1.5.3 Polytopy

Prienikom konečného počtu polpriestorov vzniká *polytop* (inak *mnohosten*), ktorý je najčastejšie sa vyskytujúcou množinou v MPC. Príklad polytopu tvoreného prienikom piatich polpriestorov môžme vidieť na obr. 1.6.



Obr. 1.6: Polytop.

Je tvorený prienikom polpriestorov s normálovými vektormi $\mathbf{a}_1, \ldots \mathbf{a}_5$.

A teda pre ľubovoľný polytop platí:

$$\mathbb{P} = \{\mathbf{x} \in \mathbb{R}^n | A\mathbf{x} \leq B\} \tag{1.5}$$

kde

$$A = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_m^T \end{bmatrix}, \qquad B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Boxové ohraničenia sú tiež polytopmi a budeme ich používať najčastejšie:

$$\mathbb{P} = \{\mathbf{x} \in \mathbb{R}^2 | x_{1,min} \leq x_{k,1} \leq x_{1,max},\ x_{2,min} \leq x_{k,2} \leq x_{2,max}\} \tag{1.6}$$

kde

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad B = \begin{bmatrix} x_{1,max} \\ x_{2,max} \\ -x_{1,min} \\ -x_{2,min} \end{bmatrix}$$

1.6 Všeobecný opis QP

Pomocou MPC vieme dobre riešiť lineárne alebo kvadratické problémy. V reálnych situáciach často chceme minimalizovať spotrebu energie na dosiahnutie zadaného cieľa. Táto spotreba energie je priamo úmerná *2-norme* daného signálu. Ak chceme napr. dopraviť tovar z miesta A do miesta B a spotrebovať pritom čo najmenej paliva, tak táto spotreba je priamo úmerná 2-norme, (Kvasnica, 2012a). A teda chceli by sme minimalizovať:

$$\min \|\mathbf{z}\|_2 \tag{1.7a}$$

$$\text{s.t. } V\mathbf{z} \leq W \tag{1.7b}$$

$$V_{eq}\mathbf{z} = W_{eq} \tag{1.7c}$$

kde nerovnosti (1.7b) predstavujú polytopické ohraničenia a rovnosti (1.7c) ohraničenia nadrovinami.
Z rovnice (1.7a) je zrejmé, že takáto funkcia nie je lineárna ani kvadratická, keďže

$$\|\mathbf{z}\|_2 = \sqrt{\mathbf{z}^T\mathbf{z}} \tag{1.8}$$

Úloha QP musí obsahovať kvadratickú účelovú funkciu a lineárne ohraničenia. Na to, aby bola funkcia kvadratická, môže obsahovať kvadratický člen, lineárny člen a konštantu. Všeobecný zápis QP vyzerá nasledovne, (Kvasnica, 2012a):

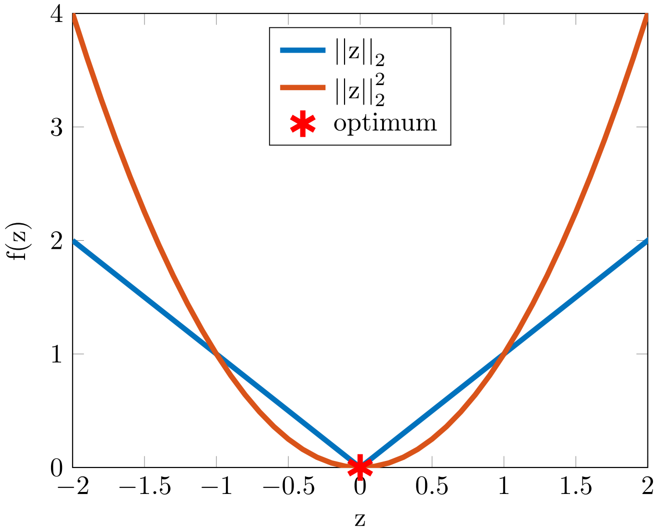
$$\min \left[\mathbf{z}^T P\mathbf{z} + 2Q^T\mathbf{z} + R\right] \tag{1.9a}$$

$$\text{s.t. } V\mathbf{z} \leq W \tag{1.9b}$$

$$V_{eq}\mathbf{z} = W_{eq} \tag{1.9c}$$

kde $P \succeq 0$ (musí byť pozitívne definitná), minimalizujeme kritérium (1.9a) vzhľadom na ohraničenia v tvare nerovností (1.9b) a ohraničenia v tvare rovností(1.9c).

Teraz si ukážeme, prečo minimalizácia 2-normy je z hľadiska nájdenia optima ekvivalentná s kvadratickou funkciou. Na obr. 1.7 vidíme grafy 2-normy a kvadrátu 2-normy pre skalárne z (čo je vlastne obyčajný kvadrát, keďže $\|z\|_2^2 = z^2$), no rovnako to platí aj pre $\mathbf{z}$ vektor. Optimum (minimum) oboch funkcií sa nachádza v tom istom bode [0,0]. Z grafov je tiež zrejmé že funkcia $\|\mathbf{z}\|_2^2$ bude hodnoty blízko nuly (v intervale (-1,1)) podhodnocovať, a hodnoty ďalej od nuly nadhodnocovať, čo však z hľadiska optima nevadí, (Kvasnica, 2012a).



Obr. 1.7: Porovnanie 2-normy a kvadrátu 2-normy.

1.7 Matematická formulácia MPC

Základ MPC je tvorený účelovou funkciou a obmedzeniami. Tie pozostávajú z predikčnej rovnice a z obmedzení vstupov, stavov a výstupov. Takto môžme zapísať všeobecnú matematickú formuláciu:

$$\min \sum_{k=0}^{N-1} \left(\|Q_x \mathbf{x}_{t+k}\|_p + \|Q_u \mathbf{u}_{t+k}\|_p \right) \tag{1.10a}$$

$$\text{s.t. } \mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} \tag{1.10b}$$

$$\mathbf{x}_{t+k} = \mathbf{x}(t+k) \tag{1.10c}$$

$$\mathbf{x}_{t+k} \in \mathbb{X} \tag{1.10d}$$

$$\mathbf{u}_{t+k} \in \mathbb{U} \tag{1.10e}$$

kde kritérium (1.10a) je účelovou funkciou, ktorá minimalizuje súčet N členov p-normy (napr. 1-norma, 2-norma, $\infty -$ norma) súčinu vektora vstupov $\mathbf{u}_{t+k}$ s váhovou maticou Q_u a p-normy súčinu vektora stavov $\mathbf{x}_{t+k}$ s váhovou maticou Q_x, N predstavuje predikčný horizont, rovnica (1.10c) vyjadruje počiatočnú podmienku, $\mathbb{X}$ a $\mathbb{U}$ sú množiny prípustných hodnôt pre stavy a riadiace vstupy. Stavový opis modelu vyjadruje rovnica (1.10b), (Rossiter, 2005).

Ak je hodnota N konečná, tak sa táto úloha nazýva *problém riadenia na konečnom horizonte*, ak $N \to \infty$, hovoríme že ide o *problém riadenia na nekonečnom horizonte*.

Počiatok stavového priestoru $\mathbf{x} = 0$ a $\mathbf{u} = 0$ musí byť obsiahnutý v zlúčiteľnej oblasti vyhovujúcej rovniciam (1.10d) a (1.10e), (Karas a kol., 2007).

My budeme používať ako účelovú funkciu kvadrát 2-normy, a teda formuláciu (1.10a) môžme prepísať:

$$\min \sum_{k=0}^{N-1} \left(\mathbf{x}_{t+k}^T Q_x \mathbf{x}_{t+k} + \mathbf{u}_{t+k}^T Q_u \mathbf{u}_{t+k} \right) \tag{1.11a}$$

$$\text{s.t. } \mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} \tag{1.11b}$$

$$\mathbf{x}_{t+k} = \mathbf{x}(t+k) \tag{1.11c}$$

$$\mathbf{x}_{t+k} \in \mathbb{X} \tag{1.11d}$$

$$\mathbf{u}_{t+k} \in \mathbb{U} \tag{1.11e}$$

Účelová funkcia (1.11a) je kvadratická a ohraničenia (1.11b) aj (1.11c) sú lineárne. Aby bol náš problém konvexný, množiny $\mathbb{X}$ a $\mathbb{U}$ musia byť konvexné. Zároveň, ak tieto ohraničenia budú tvorené sústavou lineárnych ohraničení, bude náš problém kvadratické programovanie. Teda $\mathbb{X}$ a $\mathbb{U}$ musia byť v tvare polytopov:

$$\mathbb{X} = \{x | H\mathbf{x} \leq K\} \tag{1.12a}$$

$$\mathbb{U} = \{u | L\mathbf{u} \leq M\} \tag{1.12b}$$

keďže najčastejšie budeme používať boxové ohraničenia:

$$\mathbf{x}_{min} \leq \mathbf{x}_{t+k} \leq \mathbf{x}_{max} \tag{1.13a}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{t+k} \leq \mathbf{u}_{max} \tag{1.13b}$$

budú mať matice H, K, L, M tvar:

$$H = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad K = \begin{bmatrix} \mathbf{x}_{max} \\ -\mathbf{x}_{min} \end{bmatrix}, \quad L = \begin{bmatrix} I \\ -I \end{bmatrix}, \quad M = \begin{bmatrix} \mathbf{u}_{max} \\ -\mathbf{u}_{min} \end{bmatrix} \tag{1.14}$$

1.8 Prevod MPC problému na problém QP

Keďže veľké množstvo solverov vyžaduje, aby zadaný problém bol v QP tvare (1.9a), ukážeme si, ako pretransformovať MPC problém (1.11a) na takýto problém. Budeme uvažovať ohraničenia stavov a vstupov ako polytopické:

$$\min_{\mathbf{x},\mathbf{u}} \sum_{k=0}^{N-1} \left(\mathbf{x}_{t+k}^T Q_x \mathbf{x}_{t+k} + \mathbf{u}_{t+k}^T Q_u \mathbf{u}_{t+k} \right) \tag{1.15a}$$

$$\text{s.t. } \mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} \tag{1.15b}$$

$$\mathbf{x}_{t+k} = \mathbf{x}(t+k) \tag{1.15c}$$

$$H\mathbf{x}_{t+k} \leq K \tag{1.15d}$$

$$L\mathbf{u}_{t+k} \leq M \tag{1.15e}$$

Prvý krok, ktorý musíme spraviť je to, že všetky optimalizované premenné spojíme do jedného vektora. Naše optimalizované premenné na celom predikčnom horizonte N:

$$X = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \\ \vdots \\ \mathbf{x}_{t+N} \end{bmatrix}, \qquad U = \begin{bmatrix} \mathbf{u}_t \\ \mathbf{u}_{t+1} \\ \vdots \\ \mathbf{u}_{t+N-1} \end{bmatrix} \tag{1.16}$$

Pre zjednodušenie si tento prevod ukážeme pre prípad, kedy predikčný horizont $N = 3$.

$$X = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \\ \mathbf{x}_{t+2} \\ \mathbf{x}_{t+3} \end{bmatrix}, \qquad U = \begin{bmatrix} \mathbf{u}_t \\ \mathbf{u}_{t+1} \\ \mathbf{u}_{t+2} \end{bmatrix} \tag{1.17}$$

Ďalej prepíšeme ohraničenia (1.15b) a (1.15c):

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \\ \mathbf{x}_{t+2} \\ \mathbf{x}_{t+3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & A & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \\ \mathbf{x}_{t+2} \\ \mathbf{x}_{t+3} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u}_t \\ \mathbf{u}_{t+1} \\ \mathbf{u}_{t+2} \end{bmatrix} + \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{x}(t) \tag{1.18}$$

Podobne by sme mohli prepísať aj účelovú funkciu (1.15a) a polytopické ohraničenia (1.15d) a (1.15e), uvedieme však už len kompaktnú podobu celého optimalizáčného problému, ktorú nazývame *hustá fotrmulácia*, (Kvasnica, 2012a):

$$\min \left[X^T \tilde{Q}_x X + U^T \tilde{Q}_u U \right] \tag{1.19a}$$

$$\text{s.t. } X = \tilde{A}X + \tilde{B}U + \tilde{E}\mathbf{x}(t) \tag{1.19b}$$

$$\tilde{H}X \leq \tilde{K} \tag{1.19c}$$

$$\tilde{L}U \leq \tilde{M} \tag{1.19d}$$

Jednotlivé matice vyzerajú nasledovne:

$$\tilde{Q}_x = \begin{bmatrix} Q_x & 0 & 0 & 0 \\ 0 & Q_x & 0 & 0 \\ 0 & 0 & Q_x & 0 \\ 0 & 0 & 0 & Q_x \end{bmatrix} \quad \tilde{Q}_u = \begin{bmatrix} Q_u & 0 & 0 \\ 0 & Q_u & 0 \\ 0 & 0 & Q_u \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 \\ 0 & A & 0 & 0 \\ 0 & 0 & A & 0 \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} 0 & 0 & 0 \\ B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{bmatrix} \quad \tilde{E} = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\tilde{H} = \begin{bmatrix} H & 0 & 0 & 0 \\ 0 & H & 0 & 0 \\ 0 & 0 & H & 0 \end{bmatrix} \quad \tilde{K} = \begin{bmatrix} K \\ K \\ K \end{bmatrix} \quad \tilde{L} = \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{bmatrix} \quad \tilde{M} = \begin{bmatrix} M \\ M \\ M \end{bmatrix}$$

Maticový tvar (1.19a) optimalizačného problému je už jednoducho tranformovateľný na problém QP. Najprv spojíme vektory optimalizovaných stavov a vstupov do jedného vektora $\mathbf{z}$:

$$\mathbf{z} = \begin{bmatrix} X \\ U \end{bmatrix} \tag{1.20}$$

Prepíšeme tvar (1.19a)-(1.19d) s použitím vektora $\mathbf{z}$:

$$\min \mathbf{z}^T \begin{bmatrix} \tilde{Q}_x & 0 \\ 0 & \tilde{Q}_u \end{bmatrix} \mathbf{z} \tag{1.21a}$$

$$\text{s.t.} \begin{bmatrix} \tilde{H} & 0 \\ 0 & \tilde{L} \end{bmatrix} \mathbf{z} \leq \begin{bmatrix} \tilde{K} \\ \tilde{M} \end{bmatrix} \tag{1.21b}$$

$$\begin{bmatrix} (I - \tilde{A}) & -\tilde{B} \end{bmatrix} \mathbf{z} = \tilde{E}\mathbf{x}(t) \tag{1.21c}$$

kde

$$P = \begin{bmatrix} \tilde{Q}_x & 0 \\ 0 & \tilde{Q}_u \end{bmatrix} \quad V = \begin{bmatrix} \tilde{H} & 0 \\ 0 & \tilde{L} \end{bmatrix} \quad W = \begin{bmatrix} \tilde{K} \\ \tilde{M} \end{bmatrix}$$

$$V_{eq} = \begin{bmatrix} (I - \tilde{A}) & -\tilde{B} \end{bmatrix} \quad W_{eq} = \tilde{E}\mathbf{x}(t)$$

$$Q = [0] \quad R = 0$$

a teda sme pretransformovali optimalizačný problém v MPC tvare do tvaru QP, nazývaného *riedka formulácia*, (Kvasnica, 2012a):

$$\min \left[\mathbf{z}^T P\mathbf{z} + 2Q^T\mathbf{z} + R\right] \tag{1.22a}$$

$$\text{s.t. } V\mathbf{z} \leq W \tag{1.22b}$$

$$V_{eq}\mathbf{z} = W_{eq} \tag{1.22c}$$

1.9 Pokročilé formulácie MPC

V doterajších častiach práce sme sa zaoberali základnou formuláciou MPC. To znamená, že našimi optimalizovanými premennými boli stavy a riadiace vstupy, ktoré sme sa snažili minimalizovať. V tejto časti si priblížime niektoré pokročilé formulácie, kedy budeme chcieť riadiť výstupy, či už do nuly alebo na nenulovú referenciu, ktorá bude buď konštantná alebo časovo premenná, ďalej si ukážeme špeciálny typ mäkkých ohraničení a nakoniec si povieme ako odstrániť poruchu v našom systéme. Pre jednoduchšiu ilustráciu budeme uvažovať boxové ohraničenia stavov, vstupov aj výstupov.

1.9.1 Regulácia výstupu

Ako môžme vidieť vo formulácii (1.10a), je to stavová regulácia. Teraz potrebujeme určiť ako budú vyzerať naše výstupy, čo docielime pridaním výstupnej rovnice do ohraničení, a pridáme výstupy do účelovej funkcie, takže nebudeme minimalizovať hodnotu stavov, ale výstupov, (Kvasnica, 2012b).

Matematická formulácia

$$\min \sum_{k=0}^{N-1} \left(\|Q_y\mathbf{y}_{t+k}\|_p + \|Q_u\mathbf{u}_{t+k}\|_p\right) \tag{1.23a}$$

$$\text{s.t. } \mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} \tag{1.23b}$$

$$\mathbf{y}_{t+k} = C\mathbf{x}_{t+k} + D\mathbf{u}_{t+k} \tag{1.23c}$$

$$\mathbf{x}_{t+k} = \mathbf{x}(t+k) \tag{1.23d}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{t+k} \leq \mathbf{x}_{max} \tag{1.23e}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{t+k} \leq \mathbf{u}_{max} \tag{1.23f}$$

$$\mathbf{y}_{min} \leq \mathbf{y}_{t+k} \leq \mathbf{y}_{max} \tag{1.23g}$$

Vidíme, že v účelovej funkcii (1.23a) penalizujeme odchýlku výstupu od nuly. Máme tu rovnicu výstupov (1.23c) aj výstupné ohraničenia (1.23g). Rovnica (1.23d) hovorí o tom, že stále používame stavovú spätnú väzbu.

1.9.2 Sledovanie konštantnej referencie výstupom

Táto formulácia hovorí o tom, že máme pevne danú referenčnú hodnotu, ktorá je rovnaká počas celého riadenia, a na tejto hodnote chceme udržať náš výstup (*pozn.:* takisto by sme mohli sledovať referenciu stavmi, ale výstupné sledovanie je častejšie používané, stavy sa väčšinou snažíme udržať v nejakom rozmedzí pomocou ohraničení), (Kvasnica, 2012b).

Matematická formulácia

$$\min \sum_{k=0}^{N-1} \left[\|Q_y(\mathbf{y}_{t+k} - \mathbf{y}_{ref})\|_p + \|Q_u \mathbf{u}_{t+k}\|_p \right] \tag{1.24a}$$

$$\text{s.t. } \mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} \tag{1.24b}$$

$$\mathbf{y}_{t+k} = C\mathbf{x}_{t+k} + D\mathbf{u}_{t+k} \tag{1.24c}$$

$$\mathbf{x}_{t+k} = \mathbf{x}(t+k) \tag{1.24d}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{t+k} \leq \mathbf{x}_{max} \tag{1.24e}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{t+k} \leq \mathbf{u}_{max} \tag{1.24f}$$

$$\mathbf{y}_{min} \leq \mathbf{y}_{t+k} \leq \mathbf{y}_{max} \tag{1.24g}$$

Rozdiel v účelovej funkcii (1.24a) oproti (1.23a) je, že nepenalizujeme hodnotu výstupov, ale hodnotu rozdielu medzi meraným výstupom a referenciou. Všimnime si tiež, že referencia y_{ref} nie je závislá od $[t+k]$, teda je nemenná v čase - konštantná.

1.9.3 Sledovanie časovo premennej referencie výstupom

Túto formuláciu použijeme vtedy, ak sa referencia na výstup mení v čase. Vtedy sa z koštantnej referencie stáva premenná, ktorá môže byť v každom kroku optimalizácie iná. Matematická formulácia je však veľmi podobná predchádzajúcemu prípadu, (Kvasnica, 2012b).

Matematická formulácia

$$\min \sum_{k=0}^{N-1} \left[\|Q_y(\mathbf{y}_{t+k} - \mathbf{y}_{ref,t+k})\|_p + \|Q_u \mathbf{u}_{t+k}\|_p \right] \tag{1.25a}$$

$$\text{s.t. } \mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} \tag{1.25b}$$

$$\mathbf{y}_{t+k} = C\mathbf{x}_{t+k} + D\mathbf{u}_{t+k} \tag{1.25c}$$

$$\mathbf{x}_{t+k} = \mathbf{x}(t+k) \tag{1.25d}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{t+k} \leq \mathbf{x}_{max} \tag{1.25e}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{t+k} \leq \mathbf{u}_{max} \tag{1.25f}$$

$$\mathbf{y}_{min} \leq \mathbf{y}_{t+k} \leq \mathbf{y}_{max} \tag{1.25g}$$

Jediná zmena, ktorá nastala vo formulácii problému je v účelovej funkcii (1.25a), kedy penalizujeme rozdiel meraného výstupu od referencie, ktorá závisí od času $[t + k]$, teda je časovo premenná.

1.9.4 Delta-u formulácia

Pri reálnom riadení sa často stretávame so situáciami, kedy máme problém dosiahnuť žiadanú hodnotu výstupu z dôvodu, že v účelovej funkcii penalizujeme absolútnu hodnotu akčného zásahu, a teda optimalizácia vyhodnotí ako lepšie riadenie také, ktoré nedosiahne referenciu, ale použije slabší akčný zásah, teda menej energie. Toto sa dá čiastočne vyriešiť prestavením jednotlivých váhových matíc, no toto riešenie väčšinou problém neodstráni, iba ho zmenší.

Lepším riešením je penalizovať nie samotný akčný zásah, ale $\Delta\mathbf{u}$ rozdiel medzi aktuálnym a predošlým akčným zásahom. Teda chceme aby boli naše vstupy ustálené, no nemusia byť nulové. Toto riešenie však vyžaduje okrem úpravy účelovej funkcie aj úpravu nášho stavového vektora, a teda úpravu systémových matíc, (Kvasnica, 2012b).

Postup

- najprv si vyjadríme: $\Delta \mathbf{u} = \mathbf{u}_{t+k} - \mathbf{u}_{t+k-1}$
- potom upravíme stavový vektor:

$$\tilde{\mathbf{x}}_{t+k} = \begin{bmatrix} \mathbf{x}_{t+k} \\ \mathbf{u}_{t+k-1} \end{bmatrix}$$

- ďalej upravíme predikčnú rovnicu a rovnicu výstupov:

$$\begin{aligned} \tilde{\mathbf{x}}_{t+k+1} &= \tilde{A}\tilde{\mathbf{x}}_{t+k} + \tilde{B}\Delta \mathbf{u}_{t+k} \\ \mathbf{y}_{t+k} &= \tilde{C}\tilde{\mathbf{x}}_{t+k} + \tilde{D}\Delta \mathbf{u}_{t+k} \end{aligned}$$

kde

$$\tilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix} \quad \tilde{C} = \begin{bmatrix} C & D \end{bmatrix} \quad \tilde{D} = D$$

Matematická formulácia

$$\min \sum_{k=0}^{N-1} \left[\|Q_y(\mathbf{y}_{t+k} - \mathbf{y}_{ref,t+k})\|_p + \|Q_{\Delta u}\Delta \mathbf{u}_{t+k}\|_p \right] \tag{1.26a}$$

$$\text{s.t. } \tilde{\mathbf{x}}_{t+k+1} = \tilde{A}\tilde{\mathbf{x}}_{t+k} + \tilde{B}\Delta \mathbf{u}_{t+k} \tag{1.26b}$$

$$\mathbf{y}_{t+k} = \tilde{C}\tilde{\mathbf{x}}_{t+k} + \tilde{D}\Delta \mathbf{u}_{t+k} \tag{1.26c}$$

$$\tilde{\mathbf{x}}_{t+k} = \begin{bmatrix} \mathbf{x}(t+k) \\ \mathbf{u}(t+k-1) \end{bmatrix} \tag{1.26d}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{t+k} \leq \mathbf{x}_{max} \tag{1.26e}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_{t+k} \leq \mathbf{u}_{max} \tag{1.26f}$$

$$\mathbf{y}_{min} \leq \mathbf{y}_{t+k} \leq \mathbf{y}_{max} \tag{1.26g}$$

$$\Delta \mathbf{u}_{t+k} = \mathbf{u}_{t+k} - \mathbf{u}_{t+k-1} \tag{1.26h}$$

$$\Delta \mathbf{u}_{min} \leq \Delta \mathbf{u}_{t+k} \leq \Delta \mathbf{u}_{max} \tag{1.26i}$$

Vidíme zmenenú účelovú funkciu (1.26a), zmenenú predikčnú (1.26b) a výstupnú (1.26c) rovnicu, upravený stavový vektor (1.26d), pridanú novú optimalizačnú premennú $\Delta \mathbf{u}$ (1.26h) a ohraničenia na veľkosť zmeny akčného zásahu (1.26i).

1.9.5 Mäkké ohraničenia

Z hľadiska dodržania ohraničení poznáme dva typy - *tvrdé* a *mäkké* ohraničenia.

Tvrdé ohraničenia

- musia byť dodržané v každom čase
- porušenie nie je dovolené
- príklady:
  - vstupné ohraničenia (ventil nemôže byť otvorený na viac ako 100%)
  - bezpečnostné limity (teplota, tlak nemôžu byť prekročené za žiadnych okolností)

Mäkké ohraničenia

- môžu byť niekedy porušené
- porušenie je vysoko penalizované
- príklady:
  - kvalitatívne premenné (koncentrácia by nemala klesnúť pod určitú hodnotu, v nevyhnutných prípadoch môže)
  - rýchlosť riadenia (maximálne preregulovanie - nemalo by byť prekročené, kým to nie je potrebné)

Mäkké ohraničenia môžme použiť na každú optimalizovanú premennú. Aby sme to však mohli urobiť, musíme si zadefinovať nové optimalizačné premenné, ktoré pridáme do pôvodných ohraničení a aj do účelovej funkcie, (Kvasnica, 2012b).

Matematická formulácia

$$\min \sum_{k=0}^{N-1} \Big[\|Q_y(\mathbf{y}_{t+k} - \mathbf{y}_{ref,t+k})\|_p + \|Q_{\Delta u}\Delta\mathbf{u}_{t+k}\|_p + \|Q_{s,y}\mathbf{s}_{y,t+k}\|_p \tag{1.27a}$$

$$+ \|Q_{s,u}\mathbf{s}_{u,t+k}\|_p \Big]$$

$$\text{s.t. } \tilde{\mathbf{x}}_{t+k+1} = \tilde{A}\tilde{\mathbf{x}}_{t+k} + \tilde{B}\Delta\mathbf{u}_{t+k} \tag{1.27b}$$

$$\mathbf{y}_{t+k} = \tilde{C}\tilde{\mathbf{x}}_{t+k} + \tilde{D}\Delta\mathbf{u}_{t+k} \tag{1.27c}$$

$$\tilde{\mathbf{x}}_{t+k} = \begin{bmatrix} \mathbf{x}(t+k) \\ \mathbf{u}(t+k-1) \end{bmatrix} \tag{1.27d}$$

$$\mathbf{x}_{min} \leq \mathbf{x}_{t+k} \leq \mathbf{x}_{max} \tag{1.27e}$$

$$\mathbf{u}_{min} - \mathbf{s}_{u,t+k} \leq \mathbf{u}_{t+k} \leq \mathbf{u}_{max} + \mathbf{s}_{u,t+k} \tag{1.27f}$$

$$\mathbf{y}_{min} - \mathbf{s}_{y,t+k} \leq \mathbf{y}_{t+k} \leq \mathbf{y}_{max} + \mathbf{s}_{y,t+k} \tag{1.27g}$$

$$\Delta\mathbf{u}_{t+k} = \mathbf{u}_{t+k} - \mathbf{u}_{t+k-1} \tag{1.27h}$$

$$\Delta\mathbf{u}_{min} \leq \Delta\mathbf{u}_{t+k} \leq \Delta\mathbf{u}_{max} \tag{1.27i}$$

$$\mathbf{s}_{y,t+k} \geq 0, \quad \mathbf{s}_{u,t+k} \geq 0 \tag{1.27j}$$

Použili sme mäkké ohraničenia pre výstupy a vstupy. Zadefinovali sme si nové optimalizačné premenné $s_{y,t+k}$ a $s_{u,t+k}$, ktoré sme pridali do príslušných ohraničení (1.27f) a (1.27g) a aj do účelovej funkcie (1.27a) vynásobené príslušnými váhovými maticami. Tieto sú rádovo väčšie ako váhové matice pre výstupy a rozdiel vstupov, čím sa zabezpečí porušovanie ohraničení iba v nevyhnutných situáciach.

1.9.6 Odstraňovanie nemerateľných porúch

V reálnych procesoch sa poruchy vyskytujú vždy. Môžu byť spôsobené nepresnosťou modelu v porovnaní s reálnym systémom alebo konštantnou nemerateľnou poruchou. V takýchto prípadoch sa môže stať, že riadenie nie je schopné dostať riadené výstupy na referenciu. Ak chceme tento problém odstrániť, musíme poruchu nejakým spôsobom zakomponovať do nášho modelu. Budeme uvažovať, že porucha sa môže prejaviť na hodnotách stavov aj výstupov. Do našej formulácie pribudne signál poruchy $\mathbf{d}_{t+k}$, a jeho vplyv na stavy a výstupy budú definovať matice E a F, (Muske a Badgwell, 2002). Tentokrát nebudeme používať *delta-u* formuláciu.

Postup

- pribudne predikcia pre poruchu:

$$\mathbf{d}_{t+k+1} = \mathbf{d}_{t+k} \tag{1.28}$$

- upravíme predikčnú rovnicu a rovnicu výstupov:

$$\mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} + E\mathbf{d}_{t+k} \tag{1.29a}$$

$$\mathbf{y}_{t+k} = C\mathbf{x}_{t+k} + D\mathbf{u}_{t+k} + F\mathbf{d}_{t+k} \tag{1.29b}$$

Matematická formulácia

$$\min \sum_{k=0}^{N-1} \left[\|Q_y(\mathbf{y}_{t+k} - \mathbf{y}_{ref,t+k})\|_p + \|Q_u\mathbf{u}_{t+k}\|_p \right] \tag{1.30a}$$

$$\text{s.t. } \mathbf{x}_{t+k+1} = A\mathbf{x}_{t+k} + B\mathbf{u}_{t+k} + E\mathbf{d}_{t+k} \tag{1.30b}$$

$$\mathbf{d}_{t+k+1} = \mathbf{d}_{t+k} \tag{1.30c}$$

$$\mathbf{y}_{t+k} = C\mathbf{x}_{t+k} + D\mathbf{u}_{t+k} + F\mathbf{d}_{t+k} \tag{1.30d}$$

$$\mathbf{x}_{t+k} = \mathbf{x}(t+k) \tag{1.30e}$$

$$\mathbf{d}_{t+k} = \mathbf{d}(t+k) \tag{1.30f}$$

$$\mathbf{x}_{min} \le \mathbf{x}_{t+k} \le \mathbf{x}_{max} \tag{1.30g}$$

$$\mathbf{u}_{min} \le \mathbf{u}_{t+k} \le \mathbf{u}_{max} \tag{1.30h}$$

$$\mathbf{y}_{min} \le \mathbf{y}_{t+k} \le \mathbf{y}_{max} \tag{1.30i}$$

Vidíme, že účelová funkcia sa nezmenila, zmenila sa predikčná (1.30b) a výstupná (1.30d) rovnica, do ktorej nám pribudol signál poruchy, tiež pribudla predikcia pre poruchu (1.30c) a počiatočná podmienka poruchy (1.30f). Matice E a F hovoria o tom, ako porucha ovplyvní stavy, resp. výstupy.

Kapitola 2

Riadenie autonómneho vozidla

V tejto časti si ukážeme, ako sme odvodili matematický model vozidla, definujeme prekážku, povieme si, prečo je vyhýbanie sa prekážkam náročný problém a ukážeme, ako sme tento problém riešili.

2.1 Matematický model vozidla

Na začiatok si musíme definovať naše vstupy, stavy a výstupy. Našimi stavmi budú rýchlosť a poloha. Chceme riadiť polohu vozidla vzhľadom na referenčnú trajektóriu a prekážky, preto naším riadeným výstupom bude poloha, a riadiacim vstupom bude zrýchlenie. Keďže MPC pracuje s diskrétnymi hodnotami, potrebujeme diskrétny model, ktorý odvodíme zo spojitého modelu.
Máme tri veličiny, s ktorými budeme pracovať:

- poloha d
- rýchlosť v
- zrýchlenie a

Vieme, že derivácia rýchlosti je zrýchlenie a derivácia polohy je rýchlosť. Dostávame spojitý matematický model vozidla:

$$\dot{v} = a \tag{2.1a}$$

$$\dot{d} = v + at \tag{2.1b}$$

Maticový tvar spojitého modelu má tvar:

$$\begin{bmatrix} \dot{v} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ d \end{bmatrix} + \begin{bmatrix} 1 \\ t \end{bmatrix} a \tag{2.2}$$

Keď si označíme stavy $\mathbf{z}$, riadiace vstupy $\mathbf{u}$ a riadené výstupy $\mathbf{p}$, tak spojitý model môžme zapísať nasledovne:

$$\dot{\mathbf{z}}(t) = A_s\mathbf{z}(t) + B_s\mathbf{u}(t) \tag{2.3a}$$

$$\mathbf{p}(t) = C_s\mathbf{z}(t) + D_s\mathbf{u}(t) \tag{2.3b}$$

kde matice spojitého modelu pred diskretizáciou sú:

$$A_s = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad B_s = \begin{bmatrix} 1 \\ t \end{bmatrix} \quad C_s = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad D_s = \begin{bmatrix} 0 \end{bmatrix} \tag{2.4}$$

Tento spojitý model musíme diskretizovať:

$$\frac{v(t+\Delta t) - v(t)}{\Delta t} \approx \dot{v} = a \tag{2.5a}$$

$$\frac{d(t+\Delta t) - d(t)}{\Delta t} \approx \dot{d} = v + a\Delta t \tag{2.5b}$$

Rovnice (2.5a) a (2.5b) upravíme:

$$v(t+\Delta t) = v(t) + a\Delta t \tag{2.6a}$$

$$d(t+\Delta t) = d(t) + v\Delta t + a\Delta t^2 \tag{2.6b}$$

Prepísaním rovníc (2.6a) a (2.6b) získavame maticový tvar diskretizovaného matematického modelu vozidla:

$$\begin{bmatrix} v \\ d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} \begin{bmatrix} v \\ d \end{bmatrix} + \begin{bmatrix} \Delta t \\ \Delta t^2 \end{bmatrix} a \tag{2.7}$$

kde ľavá strana rovnice je v čase $t + \Delta t$ a pravá v čase t. Náš stavový opis má potom tvar:

$$\mathbf{z}(t+\Delta t) = A_d\mathbf{z}(t) + B_d\mathbf{u}(t) \tag{2.8a}$$

$$\mathbf{p}(t) = C_d\mathbf{z}(t) + D_d\mathbf{u}(t) \tag{2.8b}$$

Ohraničenia na vstupy sú $\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}$, ohraničenia na stavy sú $\mathbf{z}_{min} \leq \mathbf{z}(t) \leq \mathbf{z}_{max}$ a ohraničenia na výstupy sú $\mathbf{p}_{min} \leq \mathbf{p}(t) \leq \mathbf{p}_{max}$. Ako sme už povedali, vstup $\mathbf{u}$ je zrýchlenie a, stavy $\mathbf{z}$ sú rýchlosť v a poloha d a výstup je poloha d. Matice pre náš stavový opis modelu sú:

$$A_d = \begin{bmatrix} 1 & 0 \\ \Delta t & 1 \end{bmatrix} \quad B_d = \begin{bmatrix} \Delta t \\ \Delta t^2 \end{bmatrix} \quad C_d = \begin{bmatrix} 0 & 1 \end{bmatrix} \quad D_d = \begin{bmatrix} 0 \end{bmatrix} \tag{2.9}$$

Keďže vozidlo sa pohybuje v dvojrozmernom priestore, každá veličina má svoju x-ovú a y-ovú súradnicu. Preto vektory $\mathbf{u}$, $\mathbf{z}$ a $\mathbf{p}$ budú vyzerať nasledovne:

$$\mathbf{u} = (\mathbf{u}_x,\ \mathbf{u}_y)^T = (a_x,\ a_y)^T \tag{2.10a}$$

$$\mathbf{z} = (\mathbf{z}_x,\ \mathbf{z}_y)^T = (v_x,\ d_x,\ v_y,\ d_y)^T \tag{2.10b}$$

$$\mathbf{p} = (\mathbf{p}_x,\ \mathbf{p}_y)^T = (d_x,\ d_y)^T \tag{2.10c}$$

a matice stavového opisu musíme upraviť:

$$A = \begin{bmatrix} A_d & \mathbf{0} \\ \mathbf{0} & A_d \end{bmatrix} \quad B = \begin{bmatrix} B_d & \mathbf{0} \\ \mathbf{0} & B_d \end{bmatrix} \quad C = \begin{bmatrix} C_d & \mathbf{0} \\ \mathbf{0} & C_d \end{bmatrix} \quad D = \begin{bmatrix} D_d & \mathbf{0} \\ \mathbf{0} & D_d \end{bmatrix} \tag{2.11}$$

kde maticové nuly majú rovnaký rozmer ako príslušné matice $A_d,\ B_d\ , C_d,\ D_d$.

Ďalej budeme namiesto argumentu (t) pre aktuálny krok používať argument (k) a namiesto $(t + \Delta t)$ pre najbližší nasledujúci krok budeme používať $(k + 1)$. Náš konečný stavový opis matematického modelu vozidla:

$$\mathbf{z}(k + 1) = A\mathbf{z}(k) + B\mathbf{u}(k) \tag{2.12a}$$

$$\mathbf{p}(k) = C\mathbf{z}(k) + D\mathbf{u}(k) \tag{2.12b}$$

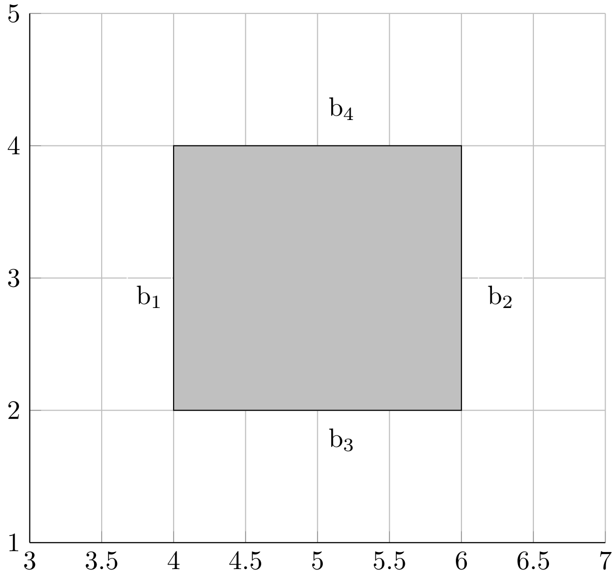
2.2 Definícia prekážky

Vo všeobecnosti uvažujeme mnohostenné prekážky, pričom každá prekážka je definovaná nerovnosťami:

$$O_j = \{\mathbf{p} | L_j^T \mathbf{p}_k \leq \mathbf{b}_j\} \qquad j = 1, \ldots, m \quad k = 1, \ldots, n \tag{2.13}$$

V našej práci sme uvažovali štvorhranné pravouhlé prekážky, príklad takejto prekážky vidíme na obr. 2.1. Z tohto dôvodu je pre každú prekážku O_j matica L_j rovnaká. Vektor ${\mathbf{p}_k}^T = (p_{k,x}, p_{k,y})$ predstavuje ľubovoľný bod na trajektórii, kde $p_{k,x}$ je horizontálna súradnica a $p_{k,y}$ je vertikálna súradnica, m je počet prekážok, n je počet bodov trajektórie. Vektor $\mathbf{b}^T = (b_1, -b_2, b_3, -b_4)$ predstavuje jednotlivé hrany prekážky, matica $L = (L_1, L_2, L_3, L_4)$ a teda pre každú prekážku platí:

$$\begin{aligned} L_1^T \mathbf{p}_k &\leq b_1 & L_1^T &= (1, 0) \\ L_2^T \mathbf{p}_k &\leq -b_2 & L_2^T &= (-1, 0) \\ L_3^T \mathbf{p}_k &\leq b_3 & L_3^T &= (0, 1) \\ L_4^T \mathbf{p}_k &\leq -b_4 & L_4^T &= (0, -1) \end{aligned} \tag{2.14}$$



Obr. 2.1: Príklad prekážky.

Rovnicu (2.13) môžme v tomto prípade zapísať:

$$O_j = \{\mathbf{p} | b_1 \leq p_{k,x} \leq b_2,\ b_3 \leq p_{k,y} \leq b_4\} \tag{2.15}$$

a nerovnosti (2.14) zjednodušiť do tvaru:

$$\begin{aligned} p_{k,x} &\geq b_1 \\ -p_{k,x} &\geq -b_2 \\ p_{k,y} &\geq b_3 \\ -p_{k,y} &\geq -b_4 \end{aligned} \tag{2.16}$$

Z takejto definície prekážky vyplýva, že vozidlo sa musí nachádzať v okolí prekážky. Toto je problém, pretože okolie uzavretej, ohraničenej prekážky je nekonvexná množina, a teda by sme museli do našej optimalizácie pridať nekonvexné ohraničenia na zaistenie podmienky, že trajektória bude ležať mimo prekážky. Takéto ohraničenia vo všeobecnosti rušia garanciu o nájdení globálneho optima v našom programe, (Boyd a Vandenberghe, 2004).

2.3 Zmiešaný celočíselný kvadratický problém

Vyššie spomínaný problém sa dá odstrániť tým, že do našej optimalizácie pridáme diskrétny komponent. Ak máme namodelovanú nekonvexnú sadu oblastí mimo prekážky ako spojenie konečného počtu konvexných oblastí, môžme vykonať *zmiešanú celočíselnú konvexnú optimalizáciu*, v ktorej celočíselné premenné zodpovedajú priradeniu častí trajektórie do konvexných oblastí. Toto však neostáva bez následkov, keďže pridanie binárnych premenných (celočíselné premenné ohraničené na hodnoty 0 a 1) menia kvadratické programovanie na zmiešané {0,1}-kvadratické programovanie, ktoré je známe ako *NP-úplný* problém (riešiteľný v nedeterministickom polynomiálnom čase), (Deits a Tedrake, 2015). To znamená, že čas potrebný na riešenie takéhoto problému rastie asymptoticky rýchlejšie ako polynomiálne (obvykle exponenciálne), berúc do úvahy veľkosť zadania problému.

Avšak v dnešnej dobe už existujú mnohé nástroje na riešenie zmiešaných celočíselných konvexných problémov, ktoré často veľmi efektívne dokážu nájsť globálne optimum pre zmiešané celočíselné kvadratické problémy.

Ak teda vezmeme do úvahy našu prekážku znázornenú na obr. 2.1, je zrejmé, že vyhnutie sa takejto prekážke znemená, že musí platiť $\mathbf{p}_k \notin O_j$, teda minimálne jedna z nerovníc (2.16) musí byť porušená. Ak všetky nerovnosti zmeníme na opačné, tak dostávame problém, kedy minimálne jedna z týchto zmenených nerovností (ohraničení) musí byť splnená.

Tu prichádzajú na rad binárne premenné, ktoré budeme označovať δ. Ak $\delta = 1$, príslušné ohraničenie je dodržané, inak je porušené:

$$\begin{aligned}
&(\delta_{k,1} = 1) \Rightarrow p_{k,x} \leq b_1 - \varepsilon \\
&(\delta_{k,2} = 1) \Rightarrow -p_{k,x} \leq -b_2 - \varepsilon \\
&(\delta_{k,3} = 1) \Rightarrow p_{k,y} \leq b_3 - \varepsilon \\
&(\delta_{k,4} = 1) \Rightarrow -p_{k,y} \leq -b_4 - \varepsilon
\end{aligned} \tag{2.17}$$

Implikáčné pravidlá (2.17) môžu byť zjednodušené na sadu konvexných ohraničení použitím jednoduchých pravidiel rozhodovacej logiky, (Williams, 2013).

Majme binárnu premennú $\delta \in \{0,\ 1\}$, premennú $x \in \mathbb{R}^m$ a ľubovoľnú funkciu $g : \mathbb{R}^m \to \mathbb{R}$. Potom

$$(\delta = 1) \Rightarrow g(x) \leq 0 \tag{2.18}$$

platí vtedy a len vtedy, ak

$$g(x) \leq M(1-\delta) \tag{2.19}$$

je splnené pre nejaké konštantné M také, že $M > \max(g(x))$.

Podľa tejto vety teraz prepíšeme pravidlá (2.17):

$$\begin{aligned} p_{k,x} - b_1 + \varepsilon &\leq M_1(1-\delta_{k,1}), \qquad & M_1 &> \max(p_{k,x} - b_1 + \varepsilon) \\ -p_{k,x} + b_2 + \varepsilon &\leq M_2(1-\delta_{k,2}), \qquad & M_2 &> \max(-p_{k,x} + b_2 + \varepsilon) \\ p_{k,y} - b_3 + \varepsilon &\leq M_3(1-\delta_{k,3}), \qquad & M_3 &> \max(p_{k,y} - b_3 + \varepsilon) \\ -p_{k,y} + b_4 + \varepsilon &\leq M_4(1-\delta_{k,4}), \qquad & M_4 &> \max(-p_{k,y} + b_4 + \varepsilon) \end{aligned} \tag{2.20}$$

pričom musí byť dodržaná podmienka:

$$\sum_{i=1}^{4} \delta_{k,i} \geq 1, \qquad i = 1, \ldots, 4 \tag{2.21}$$

Symbol ε v nerovnostiach (2.17) a (2.20) predstavuje priestor, o ktorý musí byť zväčšená medzera medzi prekážkou a vozidlom, aby nedošlo ku kolízii ani medzi jednotlivými bodmi trajektórie a dovoľuje nám použiť neostré nerovnosti.

2.3.1 Účelová funkcia

Chceme minimalizovať vzdialenosť reálnej a referenčnej trajektórie, a zároveň chceme minimalizovať aj zmenu rýchlosti vozidla, teda zrýchlenie (akčné zásahy). Budeme minimalizovať súčet kvadrátu rozdielu súradníc reálnej a referenčnej trajektórie a kvadrátu akčných zásahov. Naša účelová funkcia bude mať tvar:

$$\min \sum_{k=0}^{N-1} \left[(\mathbf{p}_k - \mathbf{p}_{k,ref})^T Q_p (\mathbf{p}_k - \mathbf{p}_{k,ref}) + \mathbf{u}_k^T Q_u \mathbf{u}_k \right] \tag{2.22}$$

2.3.2 MIQP optimalizačný problém

Náš kompletný MIQP optimalizačný problém bude obsahovať okrem účelovej funkcie ešte matematický model vozidla, ohraničenia zabezpečujúce vyhýbanie sa prekážkam a obmedzenia na vstupy, stavy a výstupy. Riešime ho v každej perióde vzorkovania, vždy s novými meraniami stavov systému. Ohraničenie (2.23j) je ohraničenie na stav rýchlosti, keďže maximálne a minimálne obmedzenie na druhý stav (polohu) sme nemali.

$$\min \sum_{k=0}^{N-1} \left[(\mathbf{p}_k - \mathbf{p}_{k,ref})^T Q_p (\mathbf{p}_k - \mathbf{p}_{k,ref}) + \mathbf{u}_k^T Q_u \mathbf{u}_k\right] \tag{2.23a}$$

$$\text{s.t. } \mathbf{z}(k+1) = A\mathbf{z}(k) + B\mathbf{u}(k) \tag{2.23b}$$

$$\mathbf{p}(k) = C\mathbf{z}(k) + D\mathbf{u}(k) \tag{2.23c}$$

$$p_{k,x} - b_1 + \varepsilon \leq M_1(1 - \delta_{k,1}), \qquad M_1 > \max\left(p_{k,x} - b_1 + \varepsilon\right) \tag{2.23d}$$

$$-p_{k,x} + b_2 + \varepsilon \leq M_2(1 - \delta_{k,2}), \qquad M_2 > \max\left(-p_{k,x} + b_2 + \varepsilon\right) \tag{2.23e}$$

$$p_{k,y} - b_3 + \varepsilon \leq M_3(1 - \delta_{k,3}), \qquad M_3 > \max\left(p_{k,y} - b_3 + \varepsilon\right) \tag{2.23f}$$

$$-p_{k,y} + b_4 + \varepsilon \leq M_4(1 - \delta_{k,4}), \qquad M_4 > \max\left(-p_{k,y} + b_4 + \varepsilon\right) \tag{2.23g}$$

$$\sum_{i=1}^{4} \delta_{k,i} \geq 1 \tag{2.23h}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \tag{2.23i}$$

$$\mathbf{v}_{min} \leq \mathbf{v}_k \leq \mathbf{v}_{max} \tag{2.23j}$$

2.3.3 QP optimalizačný problém

Náš kompletný QP optimalizačný problém na rozdiel od MIQP nebude obsahovať ohraničenia s binárnymi premennými, a ohraničenie na výstupy bude časovo premenné. Ohraničenie (2.24e) hovorí tak ako pri MIQP probléme o maximálnom a minimálnom obmedzení stavu rýchlosti.

$$\min \sum_{k=0}^{N-1} \left[(\mathbf{p}_k - \mathbf{p}_{k,ref})^T Q_p (\mathbf{p}_k - \mathbf{p}_{k,ref}) + \mathbf{u}_k^T Q_u \mathbf{u}_k\right] \tag{2.24a}$$

$$\text{s.t. } \mathbf{z}(k+1) = A\mathbf{z}(k) + B\mathbf{u}(k) \tag{2.24b}$$

$$\mathbf{p}(k) = C\mathbf{z}(k) + D\mathbf{u}(k) \tag{2.24c}$$

$$\mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \tag{2.24d}$$

$$\mathbf{v}_{min} \leq \mathbf{v}_k \leq \mathbf{v}_{max} \tag{2.24e}$$

$$\mathbf{p}_{min,k} \leq \mathbf{p}_k \leq \mathbf{p}_{max,k} \tag{2.24f}$$

2.4 Porovnanie MIQP a QP optimalizačných problémov

V predchádzajúcej sekcii sme si definovali dva optimalizačné problémy. Naše vozidlo sme riadili v dvojrozmernom priestore v zmysle osí x a y. Preto aj každá premenná (zrýchlenie, rýchlosť, poloha) bude definovaná dvoma zložkami, pre každú os jedna. Pre oba problémy máme rovnakú účelovú funkciu, ktorá minimalizuje súčet kvadrátu rozdielu medzi vypočítanou a referenčnou trajektóriou vynásobeným váhovou maticou Q_p a kvadrátu zrýchlenia vynásobeným váhovou maticou Q_u.

MIQP problém

Predikčná rovnica stavov (2.23b) pre MIQP problém obsahuje štyri ohraničenia v tvare rovností, lebo máme dva stavy (rýchlosť a polohu) a každý z nich má dve súradnice. Výstupná rovnica (2.23c) obsahuje dve ohraničenia v tvare rovností, lebo meriame len jeden výstup (polohu). Štyri ohraničenia v tvare nerovností obsahuje nerovnica (2.23i) pre riadiace vstupy, štyri ohraničenia v tvare nerovností obsahuje nerovnica (2.23j) pre stav rýchlosti. Podľa toho, koľko prekážok bolo detekovaných, podľa toho sa nám mení počet ohraničení v tvare nerovností (2.23d)-(2.23g) obsahujúcich binárne premenné. Každá detekovaná prekážka pridáva do nášho optimalizačného problému štyri ohraničenia na polohu a jedno ohraničenie na súčet binárnych premenných (2.23h) v tvare nerovností a štyri binárne premenné.
Náš MIQP optimalizačný problém obsahuje:

- kvadratickú účelovú funkciu
- 6 ohraničení v tvare rovností
- 8 ohraničení v tvare nerovností
- 5 ohraničení v tvare nerovností na každú detekovanú prekážku
- 4 binárne premenné na každú detekovanú prekážku

QP problém

Rozdiel QP problému oproti MIQP je v tom, že prekážky sú síce detekované radarom, no nie sú zarátavané do optimalizačného problému. Na základe detekcie prekážky sa zmenia ohraničenia, ktoré zabezpečia vyhnutie sa tejto prekážke. Takže pre QP problém predikčná rovnica stavov (2.24b) obsahuje štyri ohraničenia v tvare rovností a výstupná rovnica (2.24c) obsahuje dve ohraničenia v tvare rovností. Nerovnosti (2.24d) a (2.24e) obsahujú každá po štyri ohraničenia v tvare nerovností. V prípade QP problému používame časovo premenné ohraničenia na výstup, teda nerovnosť (2.24f) obsahuje štyri ohraničenia v tvare nerovností, ktoré sa menia v čase.

Dokopy náš QP optimalizačný problém obsahuje:

- kvadratickú účelovú funkciu
- 6 ohraničení v tvare rovností
- 8 ohraničení v tvare nerovností
- 4 časovo premenné ohraničenia v tvare nerovností

Kapitola 3

Simulácie

Budeme predpokladať dva scenáre a v každom si ukážeme riešenie pre tri rôzne typy referenčnej trajektórie, najprv s použitím MIQP, ktorého výpočtovú zložitosť porovnáme s riešením, na ktoré sme použili QP formuláciu.

V prvom scenári si priblížime vyhýbanie sa stacionárnym prekážkam, v druhom scenári budú prekážky pohyblivé. V každom scenári budeme riešiť tri rôzne referenčné trajektórie - *kruhovú*, *ostrohrannú* a tzv. *AB trajektóriu*, v ktorej sa budeme snažiť dostať z jedného bodu do druhého.

Na detekovanie prekážok sme používali tzv. *radar*, ktorý bol vlastne kruhovým priestorom okolo vozidla s voliteľným polomerom. Ak sa prekážka dostala do tohto priestoru, bola detekovaná riadiacim systémom, ktorý vypočítal trasu tak, aby nedošlo ku kolízii.

Ako sme spomínali, MPC je stavové riadenie, a teda museli sme zabezpečiť, aby trajektória nezasahovala do prekážky ani medzi jednotlivými krokmi vypočítanej trajektórie. Toto sme dosiahli tak, že sme umelo zväčšili prekážku v každej osi o rozmer rovnaký, ako bol rozmer vozidla a tiež obmedzeniami na zrýchlenie a rýchlosť.

V tabuľke 3.1 vidíme konštanty, ktoré boli pri všetkých scenároch rovnaké. N je predikčný horizont, T_s je perióda vzorkovania, v je prvý stav, teda rýchlosť vozidla, a je riadiaci vstup, teda zrýchlenie vozidla a Q_y a Q_u sú váhové matice zodpovedajúce výstupom $\mathbf{y}$ a vsupom $\mathbf{u}$. Vozidlo malo v každej situácii rozmery 4×4 a prekážka rozmery 6×6. Minimálna medzera medzi prekážkou a vozidlom v každej osi bola 4.

Vidíme, že sa tu nenachádzajú ohraničenia na stav polohy ani na polohu ako výstup. To je z toho dôvodu, že tieto ohraničenia sa pre každý scenár a situáciu menili. V prípade

Tabuľka 3.1: Hodnoty konštánt spoločné pre všetky scenáre

konštanta	hodnota	konštanta	hodnota
N	30	T_s	$0,25\,s$
$v_{min,x}$	-3	$v_{min,y}$	-3
$v_{max,x}$	3	$v_{max,y}$	3
$a_{min,x}$	-3	$a_{min,y}$	-3
$a_{max,x}$	3	$a_{max,y}$	3
Q_y, Q_u	$I^{2\times 2}$	r_{radar}	20

MIQP problému sme tieto ohraničenia nemali vôbec, v prípade QP problému boli tieto ohraničenia premenné, ktoré sa menili v každej perióde vzorkovania.

Vysvetlivky k obrázkom

Keďže v našej práci sme simulovali pohyb vozidla, výstupom so simulácii boli videá. Aby sme aspoň čiastočne priblížili výsledky aj vizuálne, ku každej situácii sme pripojili webovú adresu videa a z každého sme spravili štyri obrázky zachytené v rôznom čase simulácie.

V obrázkoch nie je prítomná legenda kvôli lepšej prehľadnosti, preto si jednotlivé symboly vysvetlíme teraz. Premenlivé ohraničenia, ako budeme vidieť, boli prítomné iba v situáciach vyhýbania sa prekážkam pomocou zmeny ohraničení. V nasledujúcej časti

Tabuľka 3.2: Symboly v obrázkoch simulácii

symbol		význam
■	*zelený plný štvorec*	vozidlo
□	*červený prázdny štvorec*	nedetekovaná prekážka
■	*červený plný štvorec*	detekovaná prekážka
○	*červený kruh okolo vozidla*	radar na detekciu prekážok
…	*tenká zelená bodkovaná čiara*	referenčná trajektória
—	*fialová čiara*	ralizovaná trajektória
□	*tenké čierne štvorce*	predikované pozície vozidla
▭	*hrubý čierny obdĺžnik*	premenlivé ohraničenia

si pre každý scenár popíšeme a vysvetlíme na obrázkoch a grafoch situáciu pre jednu trajektóriu - ostrohrannú. Obrázky a grafy ku všetkým ostatným situáciám pripájame v časti Dodatky, kvôli rozsiahlosti a podobnosti s ostatnými situáciami. Na začiatku každého scenáru uvádzame tabuľku porovnania hodnoty účelovej funkcie J a dĺžky času t_{sim} potrebného na výpočet celej simulácie.

3.1 Statické prekážky, ostrohranná trajektória

3.1.1 Riešenie pomocou MIQP

V tejto situácii, ako názov sekcie napovedá, sme riešili sledovanie ostrohrannej trajektórie, na ktorej sa vyskytovali statické prekážky. Na riešenie tohto problému sme použili binárne premenné.

Treba spomenúť, že pri riešení pomocou MIQP sme nemali ohraničenia na polohu (výstupy a stavy polohy), keďže toto nebolo v našom prípade potrebné. Poloha sledovala referenčnú trajektóriu a vyhýbanie sa prekážkam zabezpečovali binárne premenné.

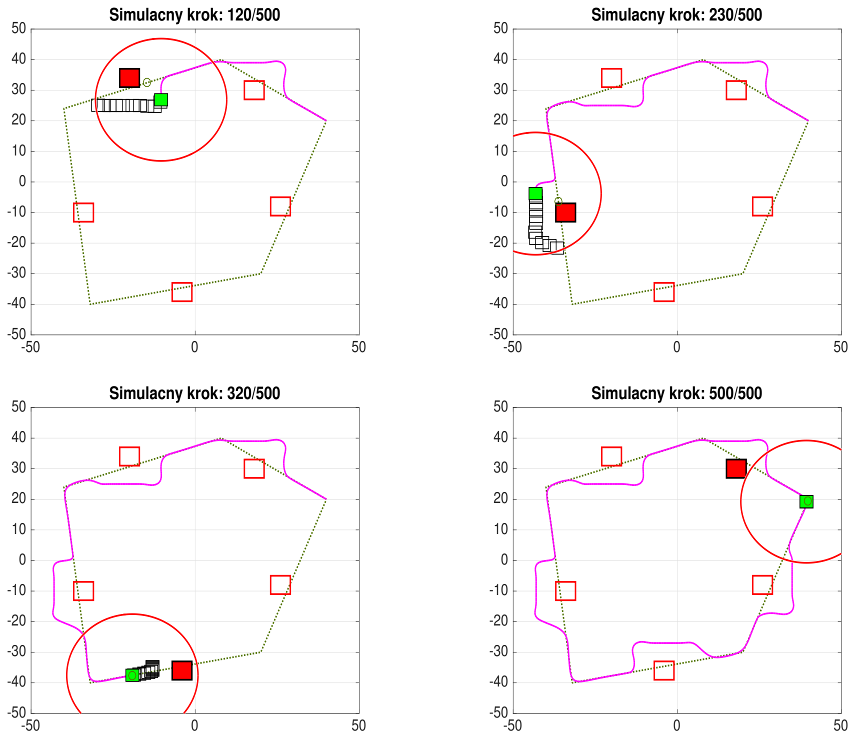
Na obr. 3.1 vidíme zachytený priebeh simulácie v štyroch rôznych krokoch. Vidíme, že prekážky sú vyplnené, pokiaľ sa nachádzajú v rádiuse radaru a teda sú zarátavané do optimalizácie, ktorá prispôsobuje vypočítanú trajektóriu tak, aby sa im vozidlo vyhlo. V opačnom prípade sú prekážky prázdne a optimalizácia si ich nevšíma.

Grafy akčných zásahov (zrýchlenia) v smere každej osi zvlášť vidíme na obr. 3.2, spolu s ohraničeniami. Miesta, kde zrýchlenie dosahovalo najväčšie absolútne hodnoty, zodpovedajú miestam s prekážkami. Ohraničenia sme sa snažili nastaviť tak, aby aspoň približne zodpovedali reálnej situácii v porovnaní s veľkosťou vozidla.

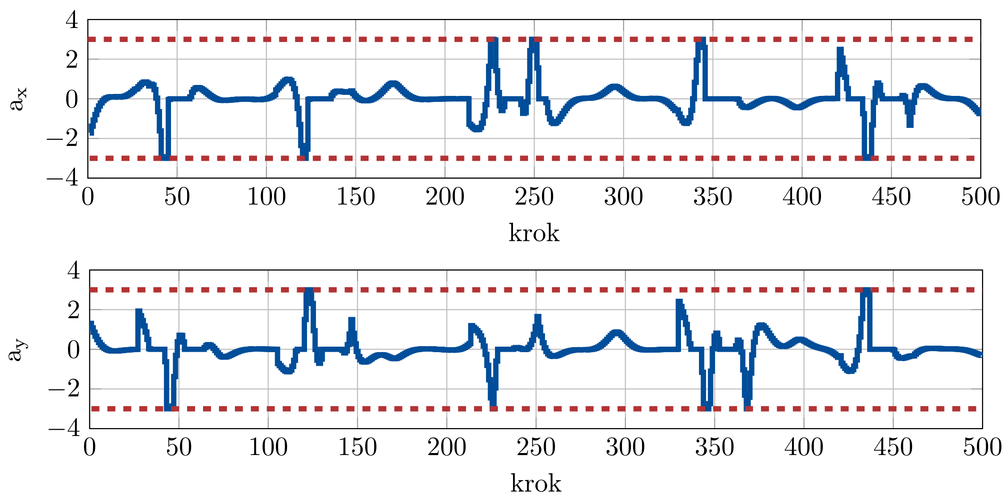
Ďalej môžme na obr. 3.3 vidieť grafy zmien rýchlosti v každej osi, ktorá sa menila na základe zrýchlenia, a tiež musela dodržovať ohraničenia na minimálne a maximálne hodnoty.

Grafy pre riadené výstupy pre každú os sú zobrazené na obr. 3.4. Ako sme spomínali, nie sú tu prítomné ohraničenia na výstup, vidíme odchýlky vypočítanej trajektórie od referenčnej v miestach, kde sa vyskytovali prekážky.

Webová adresa videa: `https://youtu.be/3pj7rLgABSU`

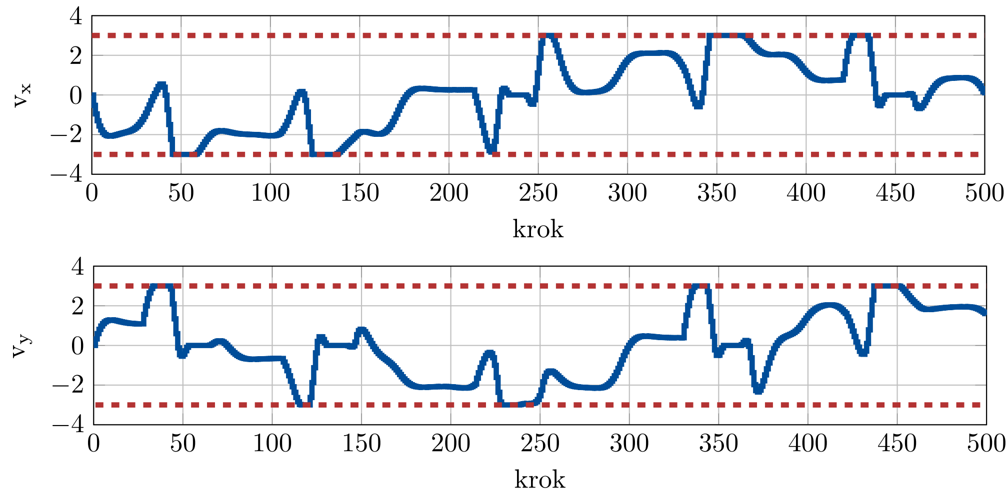


Obr. 3.1: Statické prekážky - ostrohranná trajektória (MIQP)



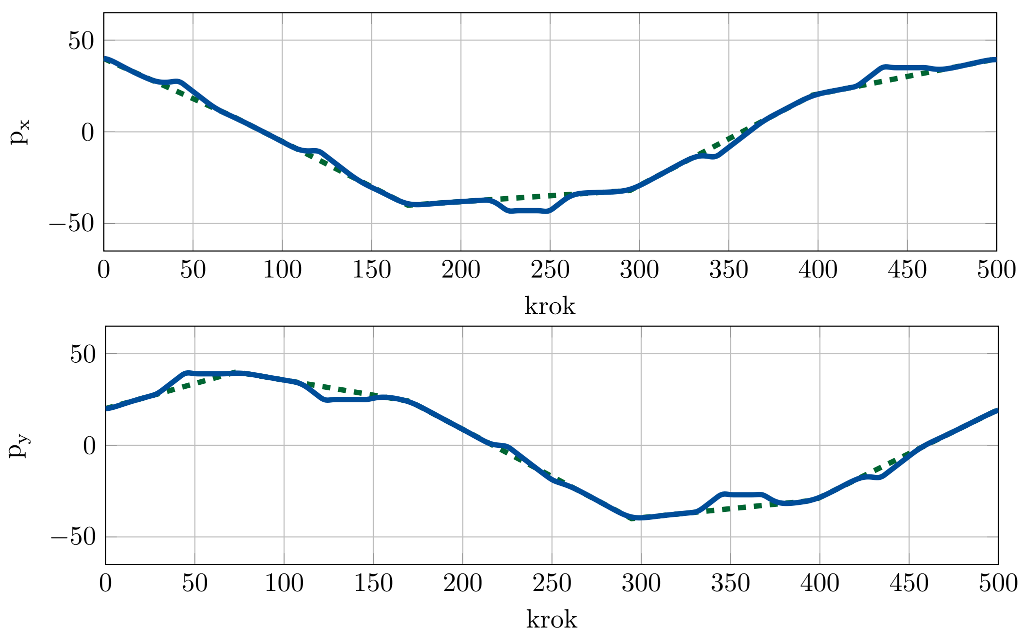
Obr. 3.2: Riadiace vstupy - zrýchlenie

a_x, a_y ▬, $a_{x,max/min}, a_{y,max/min}$ ▪▪▪



Obr. 3.3: Prvý stav - rýchlosť

v_x, v_y —, $v_{x,max/min}, v_{y,max/min}$ - - -



Obr. 3.4: Riadené výstupy - poloha

p_x, p_y —, $p_{x,ref}, p_{y,ref}$ - - -

3.1.2 Riešenie pomocou QP a zmeny ohraničení

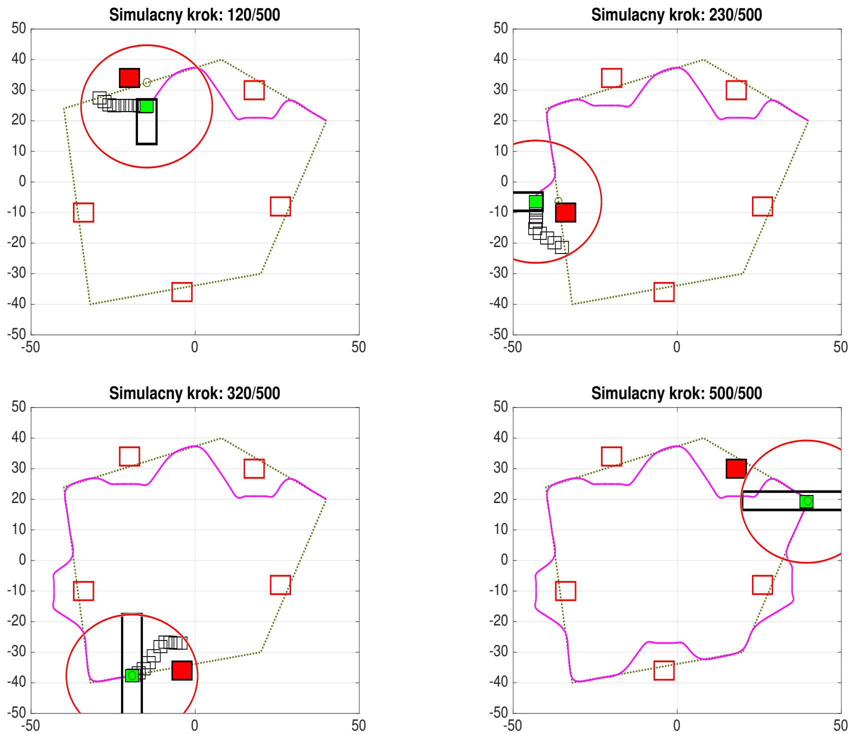
Pri tejto situácii sme riešili problém sledovania rovnakej trajektórie s rovnakým umiestnením prekážok ako v predchádzajúcej sekcii, no nevyužívali sme binárne premenné, ale zmenu ohraničení v čase.

Na to aby sme mohli takéto riešenie použiť, museli sme definovať, ako majú tieto ohraničenia vyzerať a tiež spôsob, akým sa budú meniť. Keďže sme náš problém riešili v zmysle x-ovej a y-ovej súradnice, určili sme, že zmena ohraničení nastane na tej osi, na ktorej je zmena hodnoty referenčnej trajektórie menšia. Táto zmena nastáva vtedy, ak by ohraničenie zasahovalo do vopred definovaného okolia prekážky. Na druhej súradnici zmena ohraničení nenastane.

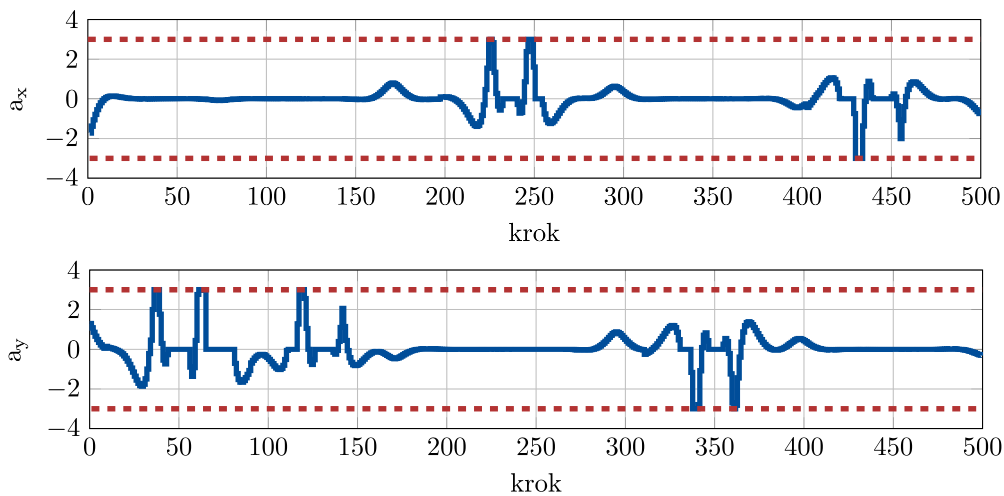
Treba spomenúť, že detekovanie prekážok prebiehalo podobne ako pri riešení s binárnymi premennými, avšak s jedným podstatným rozdielom. Ak je prekážka zachytená radarom, ten informáciu o nej nepošle do optimalizácie, ale na základe tejto informácie sú zmenené príslušné ohraničenia, ktoré sú obnovované v každej perióde vzorkovania. To znamená, že optimalizácia nerieši zložitý MIQP problém, ale QP problém, čím sa podstatne znižuje výpočtová náročnosť.

Na obr. 3.5 je zachytená simulácia v štyroch rôznych krokoch, na ktorých vidíme ako riadiaci systém reaguje na zmenu ohraničení. Keď sa pozrieme na grafy na obr. 3.6 a obr. 3.7, všimneme si rozdiel oproti grafom z riešenia MIQP (obr. 3.2 a obr. 3.3). Je to spôsobené ohraničeniami, ktorých grafy vidíme na obr. 3.8. Ako sme už povedali, ohraničenia sa menili na základe tvaru trajektórie, teda sa mohli meniť vždy iba v smere jednej osi. V smere druhej osi boli v tom prípade nastavené na tesné ohraničenie referencie zhora aj zdola. Dôvod tohto nastavenia bol ten, aby sa ohraničenia nemenili príliš skoro pred prekážkou. Pre toto nastavenie vyzerajú zmeny zrýchlenia aj rýchlosti tak, ako to vidíme na spomínaných grafoch.

Webová adresa videa: `https://youtu.be/Ifus2cFrdQ0`

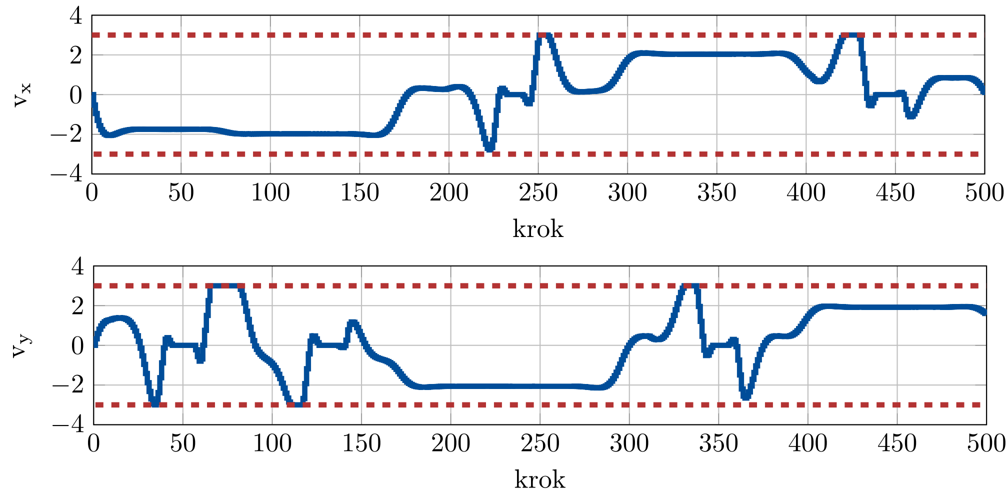


Obr. 3.5: Statické prekážky - ostrohranná trajektória (QP)



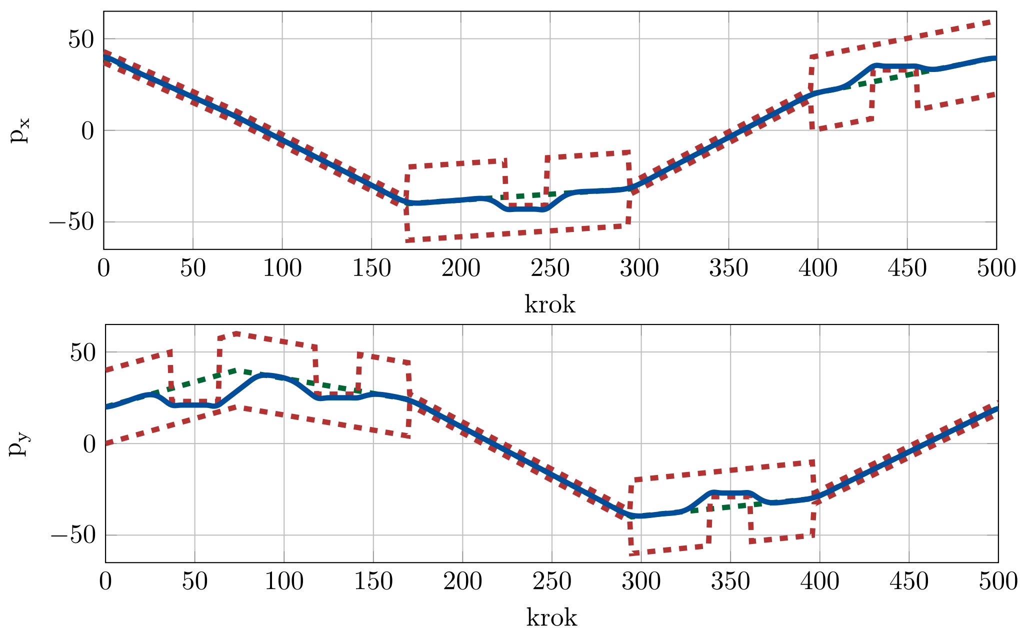
Obr. 3.6: Riadiace vstupy - zrýchlenie

a_x, a_y ▬▬, $a_{x,max/min}, a_{y,max/min}$ ▪▪▪



Obr. 3.7: Prvý stav - rýchlosť

v_x, v_y ▬, $v_{x,max/min}, v_{y,max/min}$ ▪▪▪



Obr. 3.8: Riadené výstupy - poloha

p_x, p_y ▬, $p_{x,ref}, p_{y,ref}$ ▪▪▪, $p_{x,max/min}, p_{y,max/min}$ ▪▪▪

3.1.3 Porovnanie výsledkov

Tab. 3.3 ukazuje hodnoty účelovej funkcie a dĺžku výpočtového času pre spôsoby riadenia MIQP a QP. Pri MIQP je hodnota J menšia ako pre druhý spôsob, čo bolo očakávané, keďže toto riešenie by malo byť optimálne, ak sme použili rovnaké nastavenia. Čo sa týka výpočtového času, tu je situácia opačná. Pre QP problém je výpočtový čas takmer 5-násobne kratší ako pre MIQP, čo nie je zanedbateľný rozdiel.

Tabuľka 3.3: Porovnanie výsledkov pre statické prekážky

	MIQP	QP
J	1	1,69
t_{sim}	$43,5\,s$	$8,9\,s$

3.2 Pohyblivé prekážky, ostrohranná trajektória

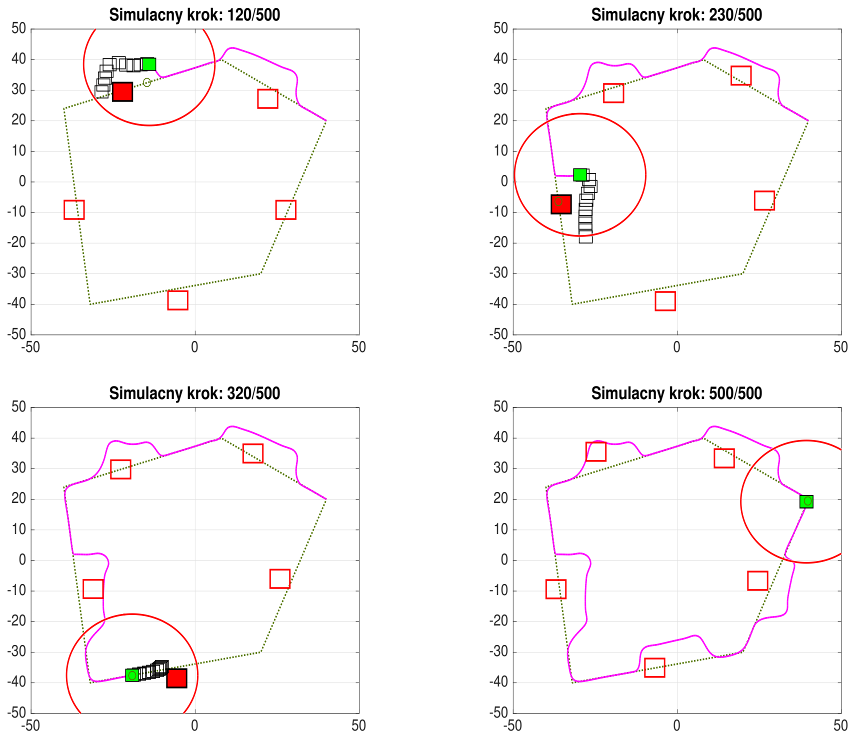
3.2.1 Riešenie pomocou MIQP

Na začiatku tejto sekcie si vysvetlíme, aký je rozdiel medzi statickou a pohyblivou prekážkou. Statická prekážka počas celej simulácie nemení svoju polohu. Teda vieme o nej s určitosťou povedať, kde sa bude nachádzať v každom kroku simulácie.

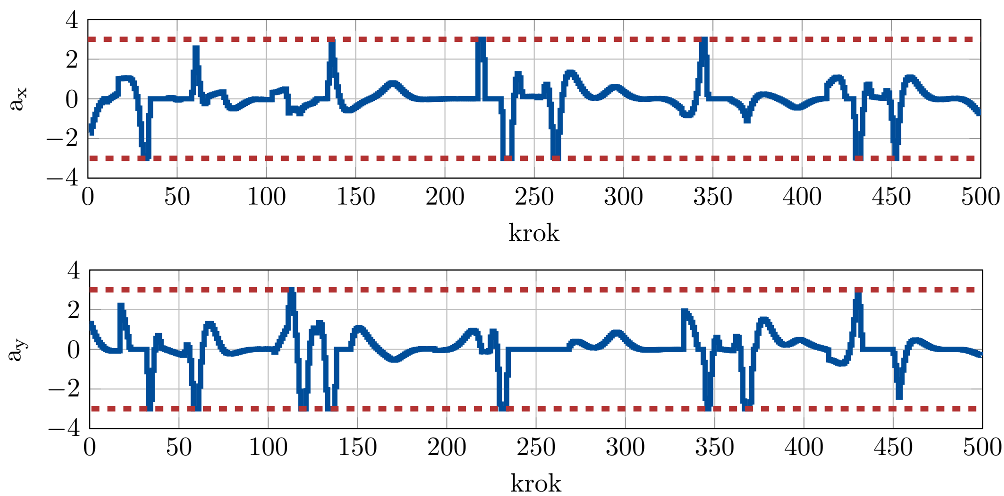
Pohyblivá prekážka je taká, ktorá sa v každom kroku simulácie môže nachádzať na inom mieste, teda jej poloha sa mení. Na to, aby sme sa jej mohli efektívne vyhnúť, musíme poznať jej predikovanú (budúcu) polohu na predikčnom horizonte. Toto sa môže zdať ako nezlúčiteľné s reálnou situáciou, no nie je to úplne pravda. Ak je radarom zachytená pohybujúca sa prekážka, jej budúci pohyb sa dá na krátkom časovom horizonte predikovať celkom presne. A teda pri simuláciách sme predpokladali, že poznáme budúci pohyb prekážky na predikčnom horizonte.

Pohyb prekážok môžme vidieť na obr. 3.9, kde vidíme, že v každom zo zobrazených krokov sa prekážky nachádzajú na inom mieste. Na konci simulácie v kroku 500 výsledná trajektória zasahuje do druhej prekážky, čo však nie je problém, keďže v kroku, kedy sa tam nachádzalo vozidlo, prekážka bola na inom mieste. Grafy zrýchlenia, rýchlosti a polohy, ktoré vidíme na obr. 3.10, obr. 3.11 a obr. 3.12 zobrazujú priebeh týchto veličín v smere každej osi zvlášť.

Webová adresa videa: `https://youtu.be/YVjHysLTJnE`

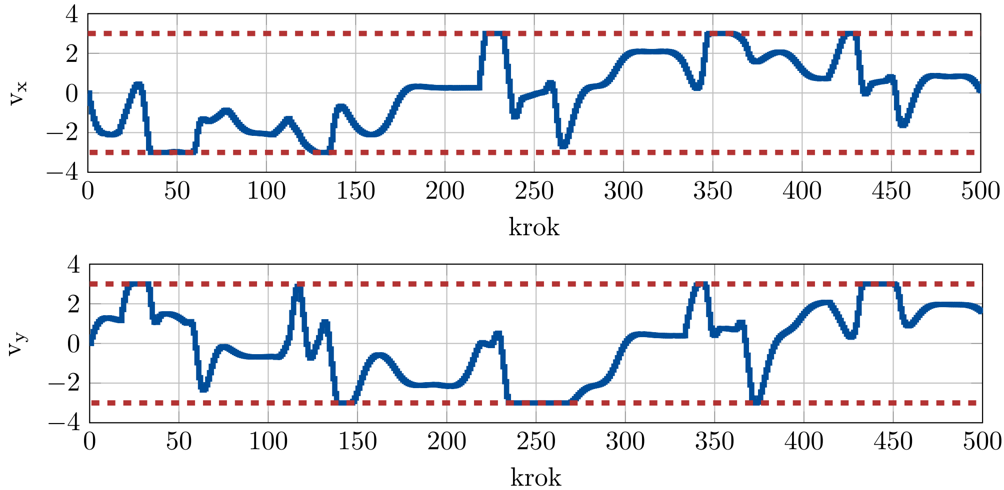


Obr. 3.9: Pohyblivé prekážky - ostrohranná trajektória (MIQP)



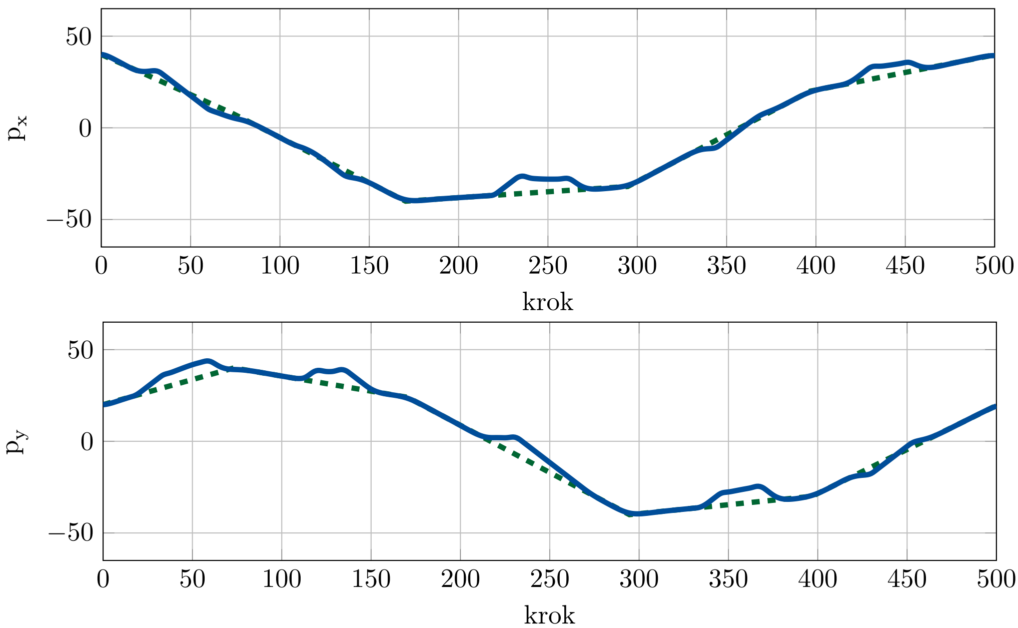
Obr. 3.10: Riadiace vstupy - zrýchlenie

a_x, a_y ━━, $a_{x,max/min}, a_{y,max/min}$ ┅



Obr. 3.11: Prvý stav - rýchlosť

v_x, v_y ———, $v_{x,max/min}, v_{y,max/min}$ - - -



Obr. 3.12: Riadené výstupy - poloha

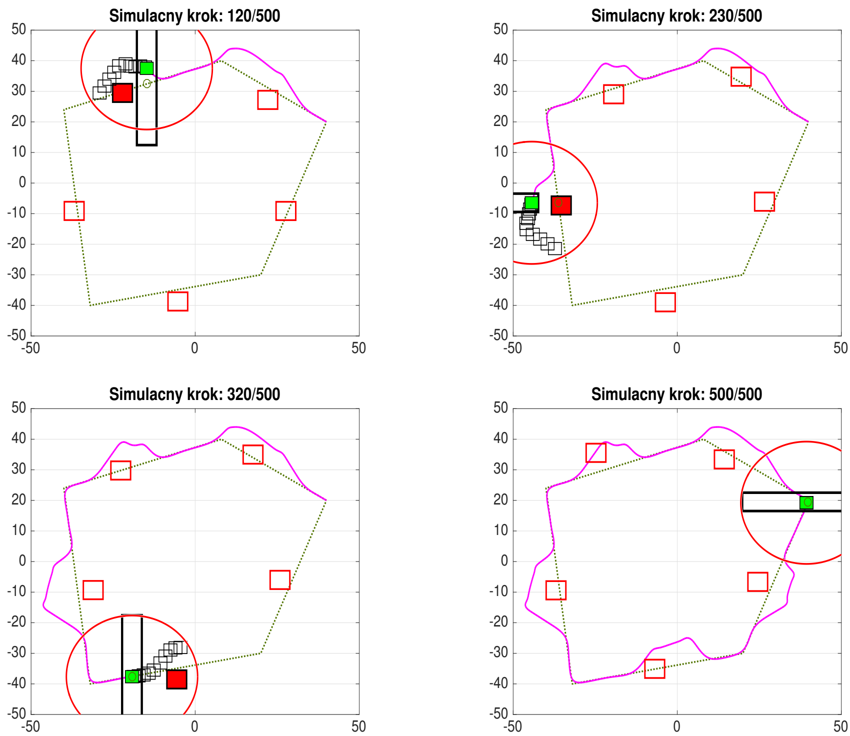
p_x, p_y ———, $p_{x,ref}, p_{y,ref}$ - - -

3.2.2 Riešenie pomocou zmeny ohraničení

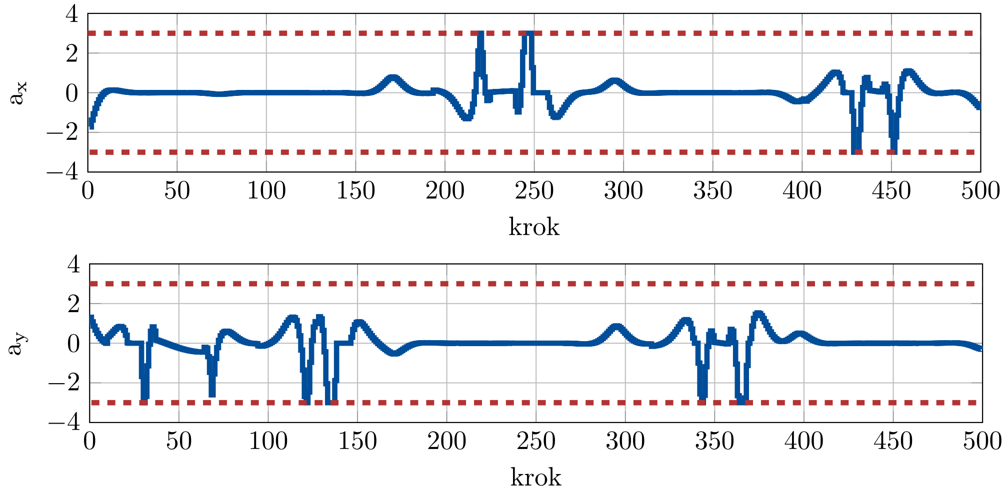
Keďže pri riešení obchádzania prekážok týmto spôsobom sú veľmi dôležité ohraničenia, musíme vhodne nastaviť minimálne a maximálne hodnoty, a podmienky pri ktorých sa budú meniť. Toto je samozrejme dôležité aj pri statických prekážkach, no pri pohyblivých je to ešte dôležitejšie, keďže sa ohraničenia budú meniť zložitejšie.

Priebeh simulácie vidíme na obr. 3.13, na obr. 3.14, 3.15 a 3.16 vidíme grafy priebehov zrýchlenia, rýchlosti a polohy vzhľadom ku každej osi zvlášť.

Webová adresa videa: `https://youtu.be/D1gJww0wlQ0`

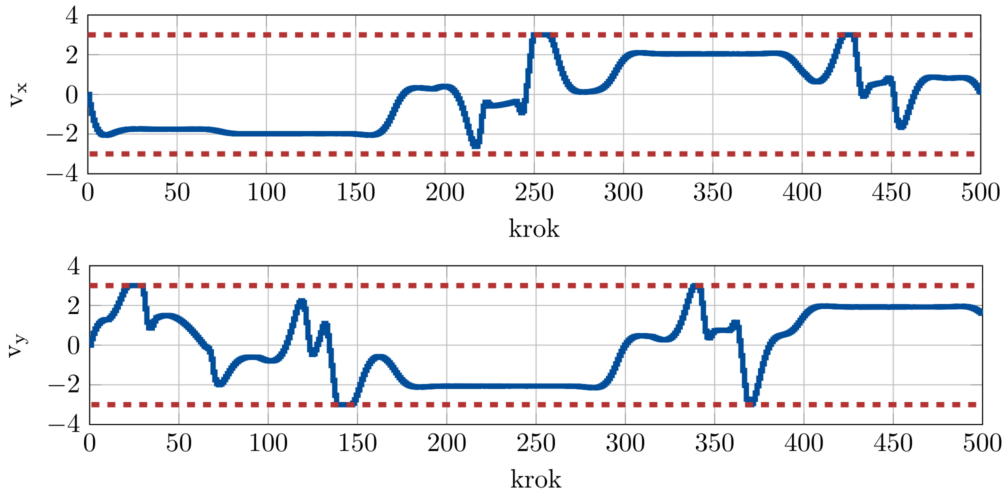


Obr. 3.13: Pohyblivé prekážky - ostrohranná trajektória (QP)



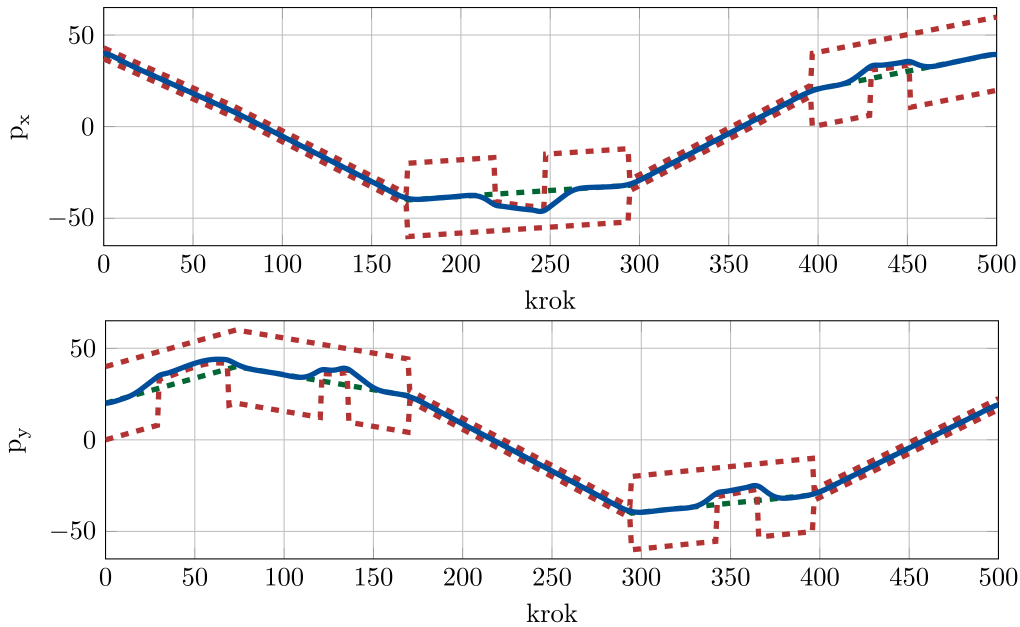
Obr. 3.14: Riadiace vstupy - zrýchlenie

a_x, a_y ▬▬, $a_{x,max/min}, a_{y,max/min}$ ▪▪▪



Obr. 3.15: Prvý stav - rýchlosť

v_x, v_y ▬▬, $v_{x,max/min}, v_{y,max/min}$ ▪▪▪



Obr. 3.16: Riadené výstupy - poloha

p_x, p_y ▬, $p_{x,ref}, p_{y,ref}$ ▪▪▪, $p_{x,max/min}, p_{y,max/min}$ ▪▪▪

3.2.3 Porovnanie výsledkov

V tab. 3.4 môžme vidieť hodnoty účelovej funkcie a dĺžku výpočtového času pre spôsoby riadenia MIQP a QP pre pohyblivé prekážky. V tomto prípade je hodnota J menšia pre zmenu ohraničení. Síce by sme očakávali, že to bude naopak, no vysvetľujeme to tak, že pri zmene ohraničení je vždy poloha ohraničená v smere jednej osi veľmi blízko referencie, čo v konečnom dôsledku pridáva akúsi nepriamu *váhu* na polohu aj zrýchlenie, no neprejaví sa to na hodnote účelovej funkcie, keďže váhy Q_y a Q_u sa nezmenili. Táto situácia sa prejavila pri obchádzaní pohyblivých prekážok. Výpočtový čas je podobne ako pri obchádzaní statických prekážok takmer 5-násobne kratší pre QP ako pre MIQP.

Tabuľka 3.4: Porovnanie výsledkov pre pohyblivé prekážky

	MIQP	QP
J	1	0,83
t_{sim}	$45,3\,s$	$9,1\,s$

Záver

Vytvorili sme matematický model vozidla pomocou dvojitých integrátorov, ktorý sme zo spojitého previedli na diskrétny. Definovali sme prekážku ako polytop. Túto malo vozidlo zdetekovať a následne sa jej vyhnúť. Na toto sme vytvorili riadiaci systém na báze MPC, s využitím zmiešaného celočíselného kvadratického programovania.

Tiež sme vytvorili riadiaci systém, ktorý sa vyhýba prekážkam pomocou zmeny ohraničení v čase. Pri tejto formulácii sme predpokladali zníženie výpočtovej náročnosti.

Oba tieto systémy sme simulačne overili na dvoch rôznych scenároch (pre statické prekážky a pre pohyblivé prekážky) a pre tri rôzne trajektórie (ostrohrannú, kruhovú a AB trajektóriu).

Potvrdili sme predpoklad, že vyhýbanie sa prekážkam pomocou QP a zmeny ohraničení v čase bude menej výpočtovo náročné, ako riešenie pomocou MIQP. Na druhej strane, pri MIQP máme garanciu optimality napriek tomu, že v niektorých prípadoch bola hodnota účelovej funkcie nižšia pre zmenu ohraničení (tento stav sme vysvetlili v sekcii 3.2.3). Výber riadiaceho systému teda závisí od toho, či je pre nás dôležitejšia garantovaná optimálna trajektória a minimálne náklady na riadenie, alebo rýchlosť výpočtu riadenia.

Na základe simulácii môžme povedať, že takýto riadiaci systém by mohol dobre fungovať napr. pri riadení dronov, ktoré sa vyskytujú okolo nás čoraz častejšie. Tieto by mohli napríklad pomáhať pri doručovaní malých zásielok na ťažko prístupné miesta. S postupom vývoja hardvéru (veľkosť a výdrž dronov) by sa tieto zásielky mohli postupne zväčšovať a trasy na doručovanie predlžovať. Pri použití takéhoto riadenia by odpadla nutnosť manuálne ovládať drona na diaľku.

V budúcnosti by sme chceli vyriešiť tretí scenár, kedy by riadiaci systém doviedol vozidlo do referenčného bodu, ktorý by bol ohraničený prekážkami z troch strán, teda akýsi simulátor autonómneho parkovania.

Takisto by sme chceli vyriešiť problém s prekryvom trajektórie a prekážky medzi jednotlivými bodmi trajektórie tak, aby bolo garantované zabráneniu konfliktu bez ohľadu na veľkosť a tvar vozidla aj prekážky.

Dodatky

Dodatok A

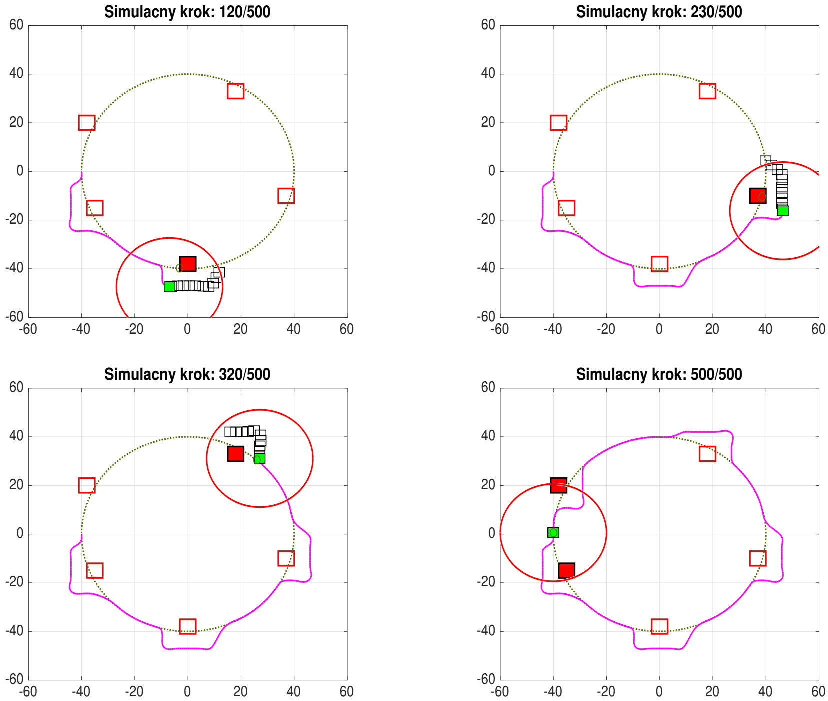
Výsledky a grafy simulácii

Táto kapitola ako už názov napovedá, obsahuje obrázky a grafy vytvorené zo simulácii oboch scenárov pre každý typ trajektórie a pre oba spôsoby riešenia optimalizačného problému. Dôvod, prečo nie sú tieto obrázky súčasťou kapitoly 3 je ten, že vysvetlenia a popis k týmto obrázkom by boli veľmi podobné tým v spomínanej kapitole, a mohli by pôsobiť neprehľadne. Pri každej situácii sa nachádza odkaz na webovú adresu, na ktorej sa dá spustiť video s konkrétnou simuláciou.

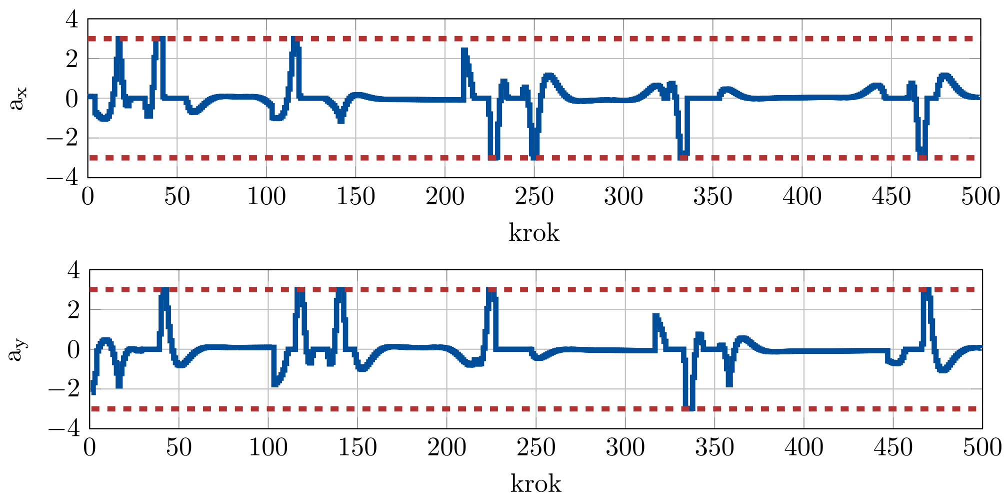
Tabuľka A.1: Porovnanie výsledkov simulácii

	Statické prekážky				Pohyblivé prekážky			
	Kruhová traj.		AB traj.		Kruhová traj.		AB traj.	
	MIQP	QP	MIQP	QP	MIQP	QP	MIQP	QP
J	1	1,56	1	0,99	1	1,28	1	1,67
t_{sim}	$43,1\,s$	$8,8\,s$	$25,5\,s$	$5,5\,s$	$44,9\,s$	$9,1\,s$	$26,9\,s$	$5,3\,s$

Kruhová trajektória, MIQP, statické prekážky

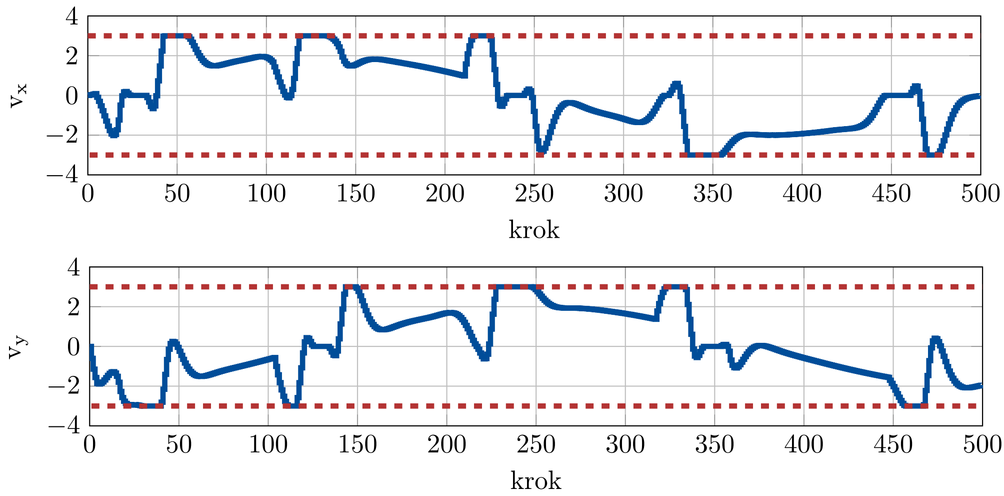


Obr. A.1: Statické prekážky - kruhová trajektória (MIQP)



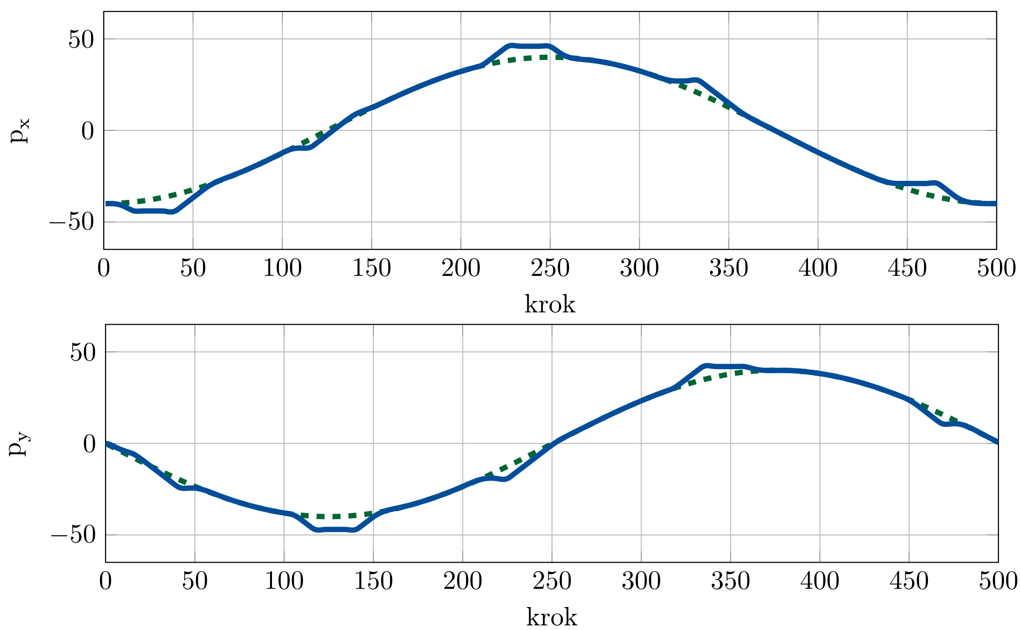
Obr. A.2: Riadiace vstupy - zrýchlenie

a_x, a_y ▬, $a_{x,max/min}, a_{y,max/min}$ ▪▪▪



Obr. A.3: Prvý stav - rýchlosť

v_x, v_y ——, $v_{x,max/min}, v_{y,max/min}$ - - -

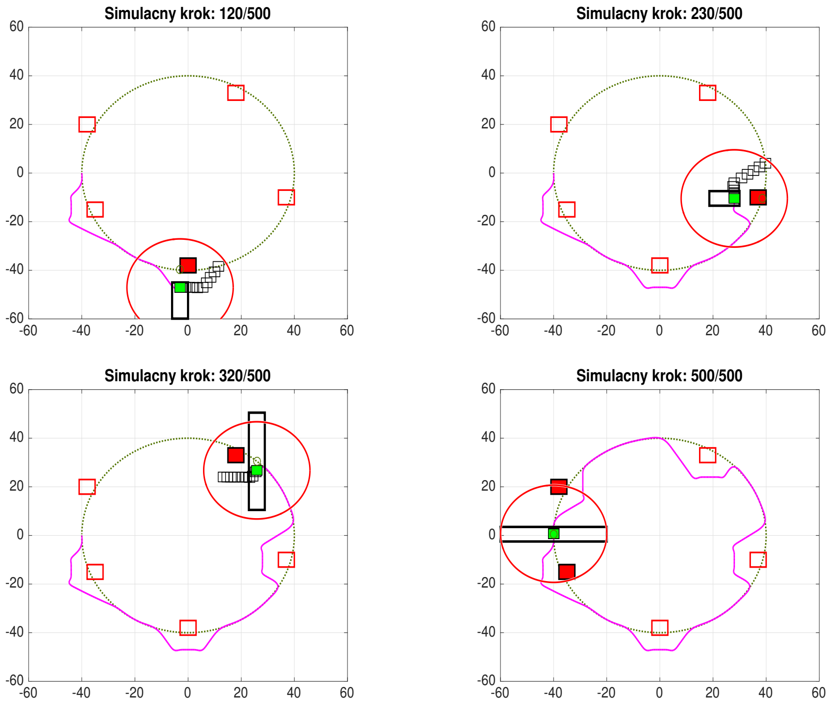


Obr. A.4: Riadené výstupy - poloha

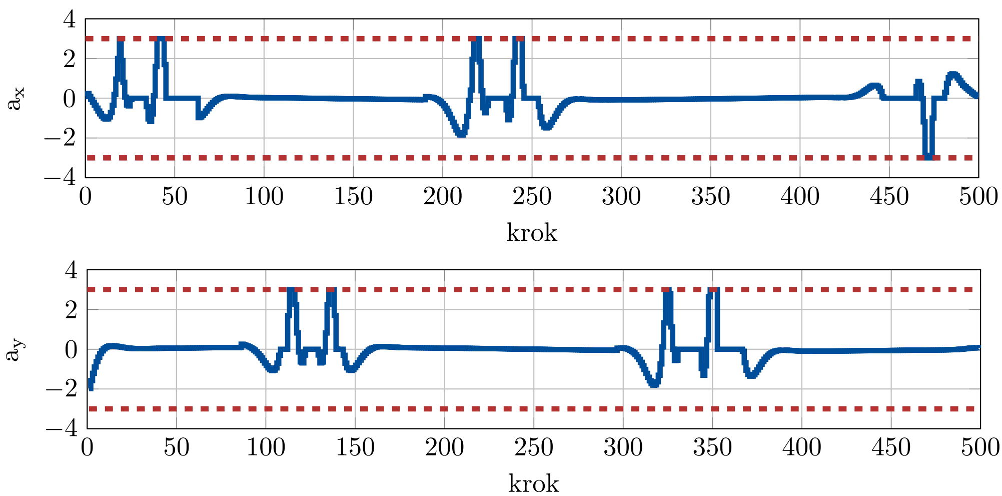
p_x, p_y ——, $p_{x,ref}, p_{y,ref}$ - - -

Webová adresa videa: `https://youtu.be/87HToYz5nAY`

Kruhová trajektória, QP, statické prekážky

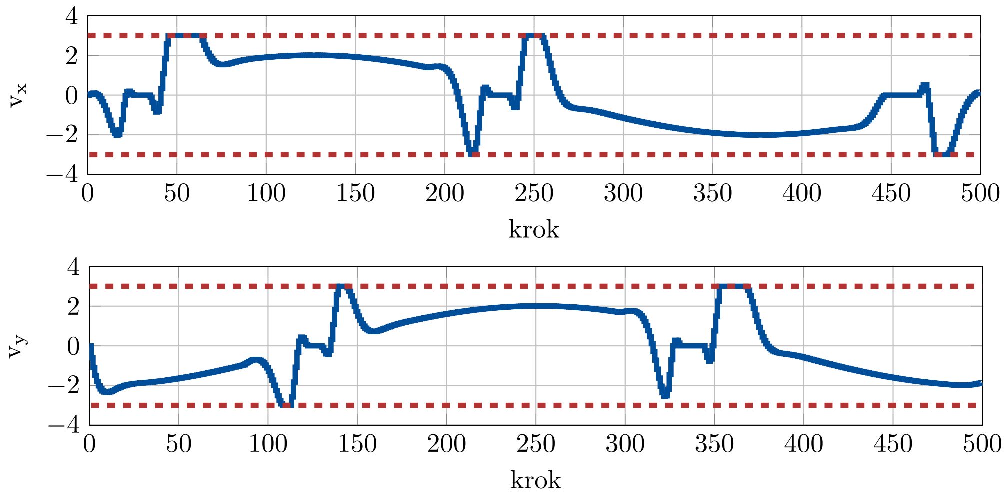


Obr. A.5: Statické prekážky - kruhová trajektória (QP)



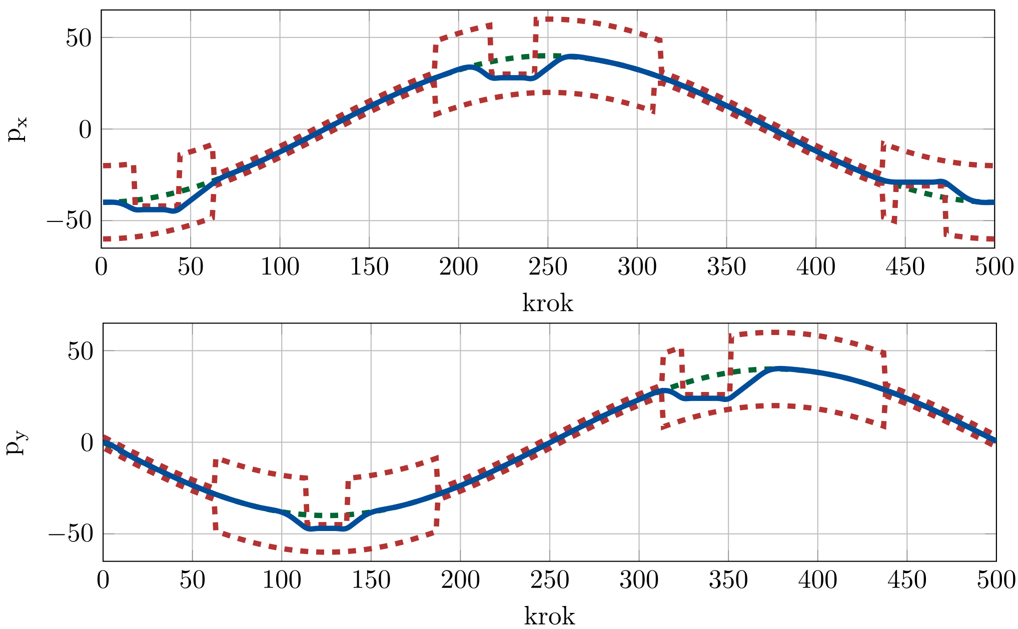
Obr. A.6: Riadiace vstupy - zrýchlenie

a_x, a_y , $a_{x,max/min}, a_{y,max/min}$



Obr. A.7: Prvý stav - rýchlosť

v_x, v_y ━━, $v_{x,max/min}, v_{y,max/min}$ ┅┅

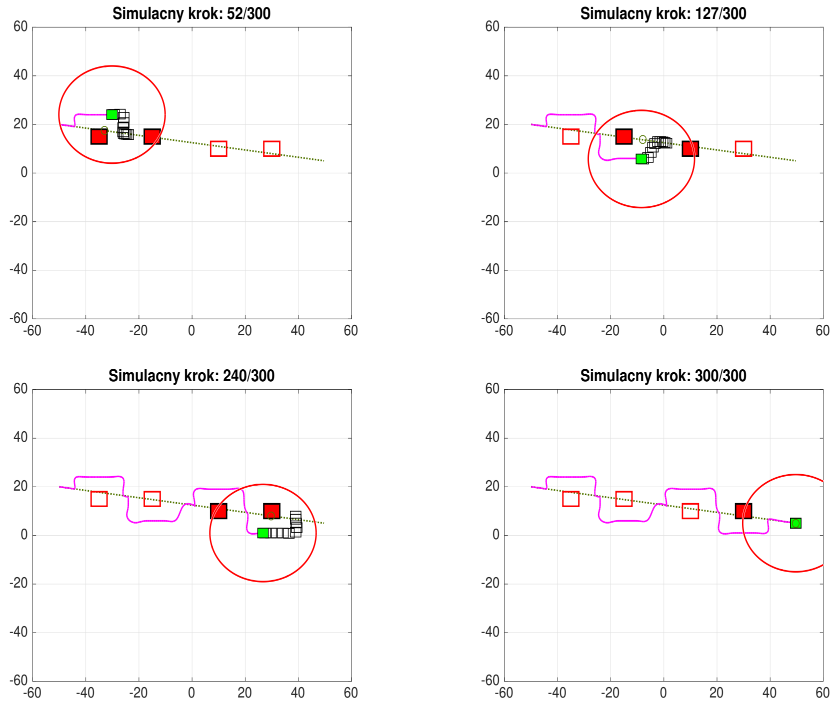


Obr. A.8: Riadené výstupy - poloha

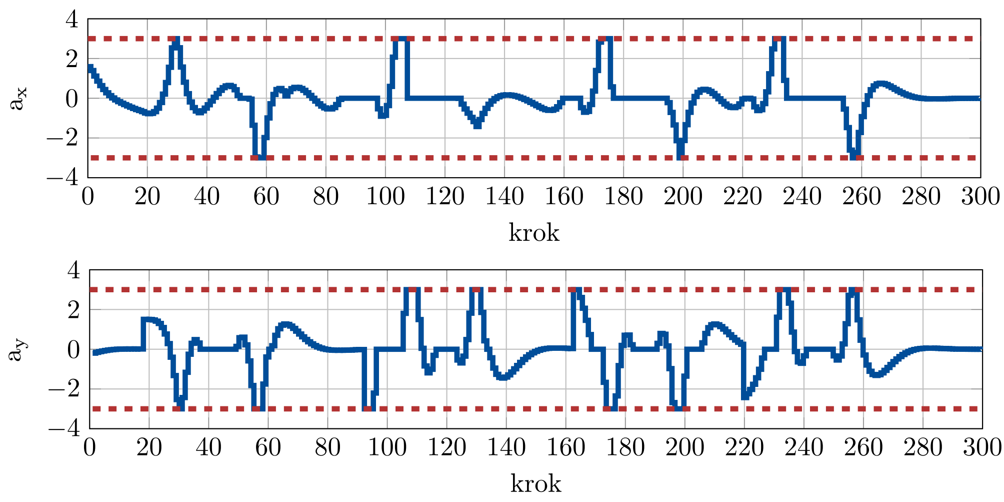
p_x, p_y ━━, $p_{x,ref}, p_{y,ref}$ ┅┅, $p_{x,max/min}, p_{y,max/min}$ ┅┅

Webová adresa videa: `https://youtu.be/u3eUoykCMNQ`

AB trajektória, MIQP, statické prekážky

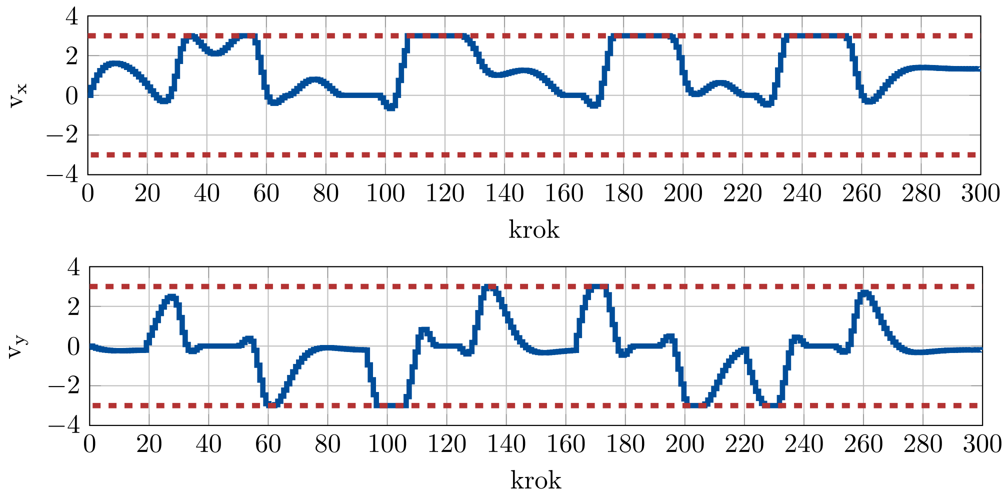


Obr. A.9: Statické prekážky - AB trajektória (MIQP)



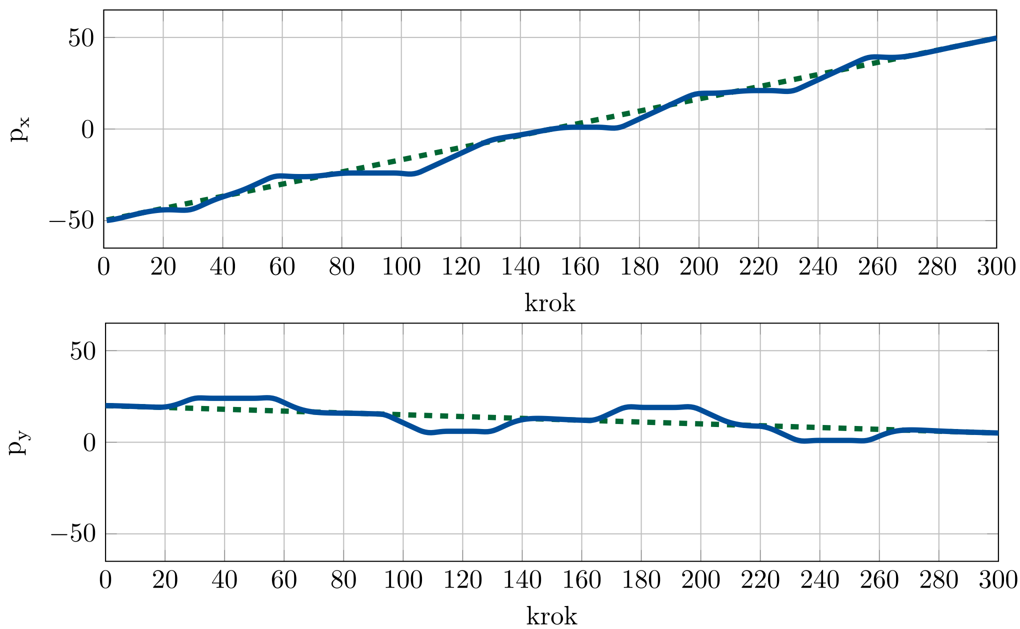
Obr. A.10: Riadiace vstupy - zrýchlenie

a_x, a_y ▬, $a_{x,max/min}, a_{y,max/min}$ ▪▪▪



Obr. A.11: Prvý stav - rýchlosť

v_x, v_y ——, $v_{x,max/min}, v_{y,max/min}$ - - -

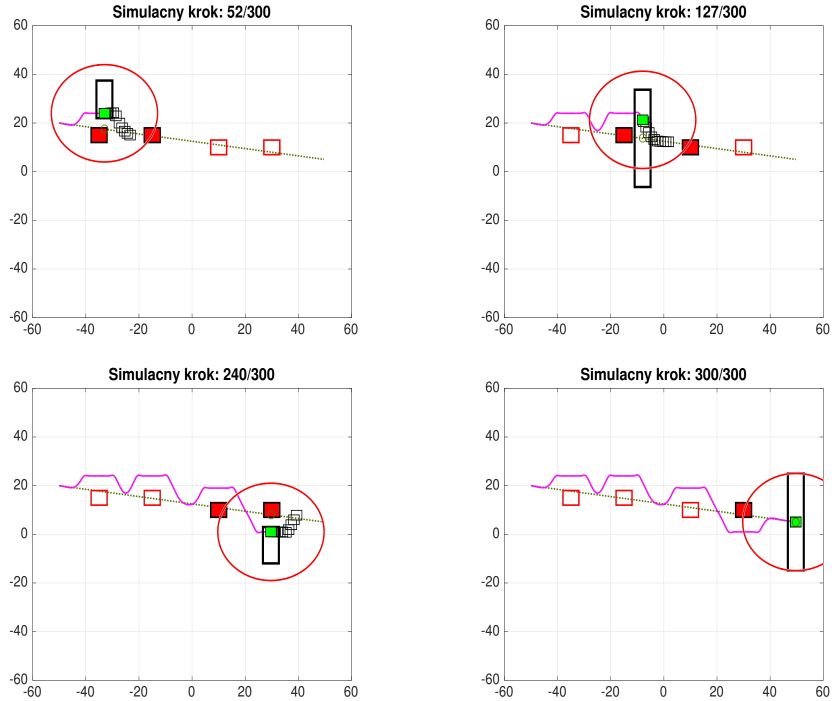


Obr. A.12: Riadené výstupy - poloha

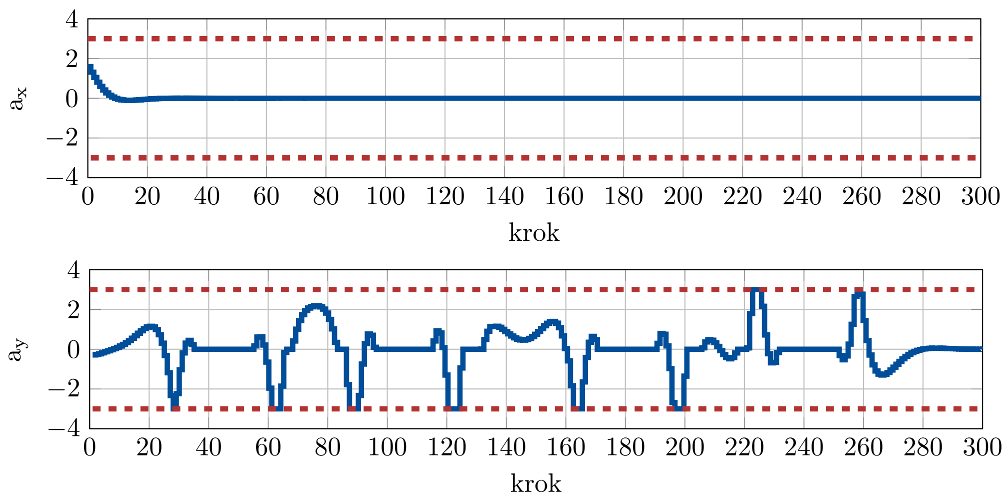
p_x, p_y ——, $p_{x,ref}, p_{y,ref}$ - - -

Webová adresa videa: `https://youtu.be/QPKQOPiKOtE`

AB trajektória, QP, statické prekážky

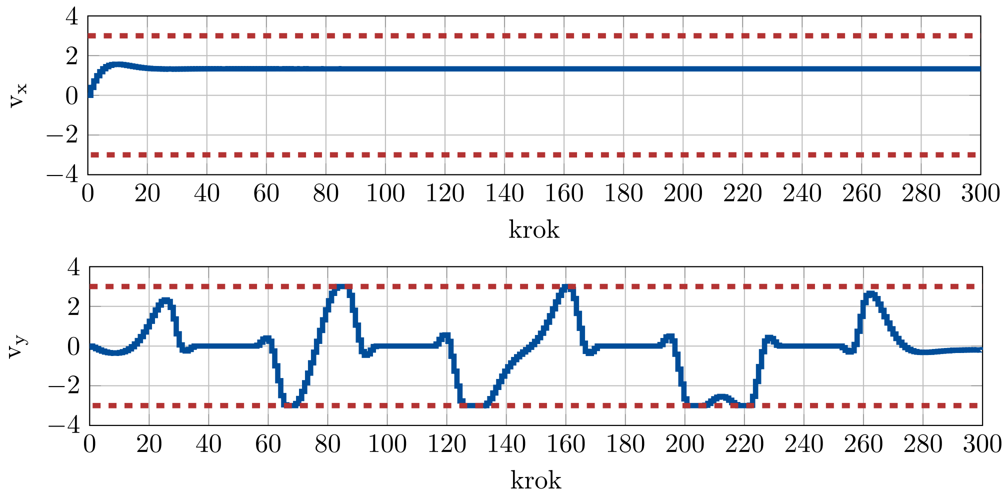


Obr. A.13: Statické prekážky - AB trajektória (QP)



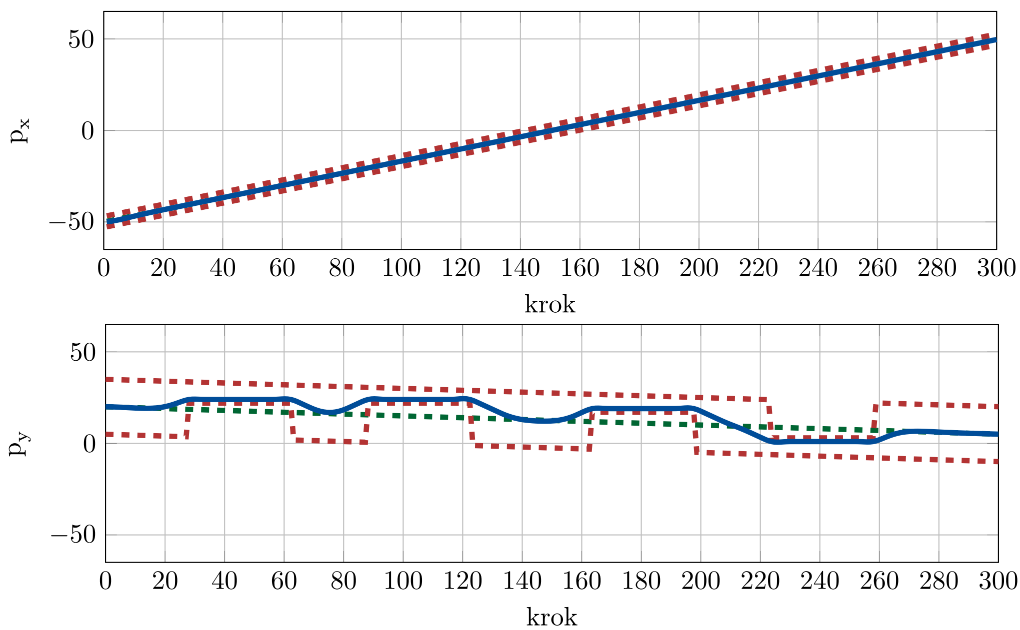
Obr. A.14: Riadiace vstupy - zrýchlenie

a_x, a_y ▬, $a_{x,max/min}, a_{y,max/min}$ ▪▪▪



Obr. A.15: Prvý stav - rýchlosť

v_x, v_y ——, $v_{x,max/min}, v_{y,max/min}$ - - -

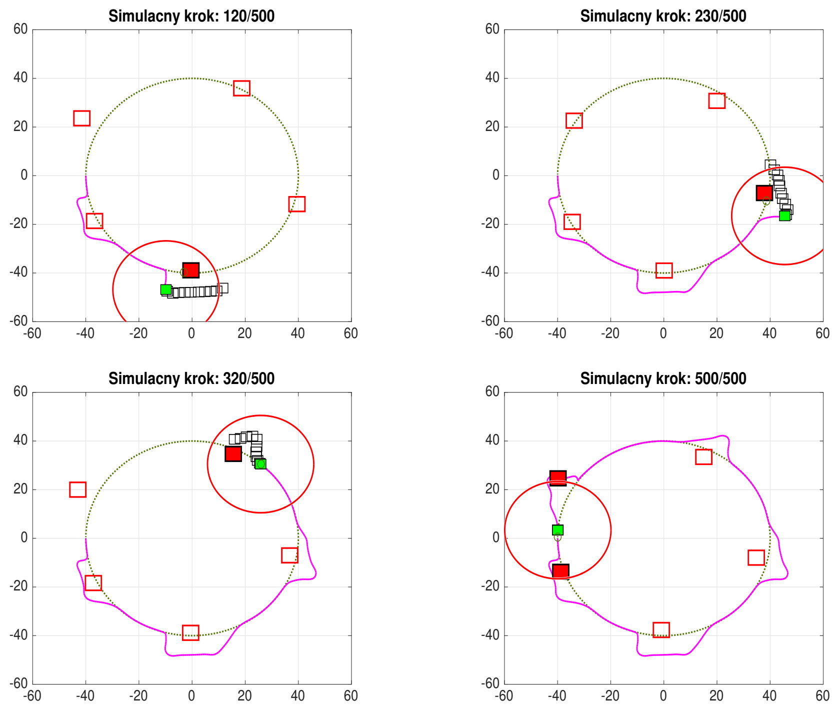


Obr. A.16: Riadené výstupy - poloha

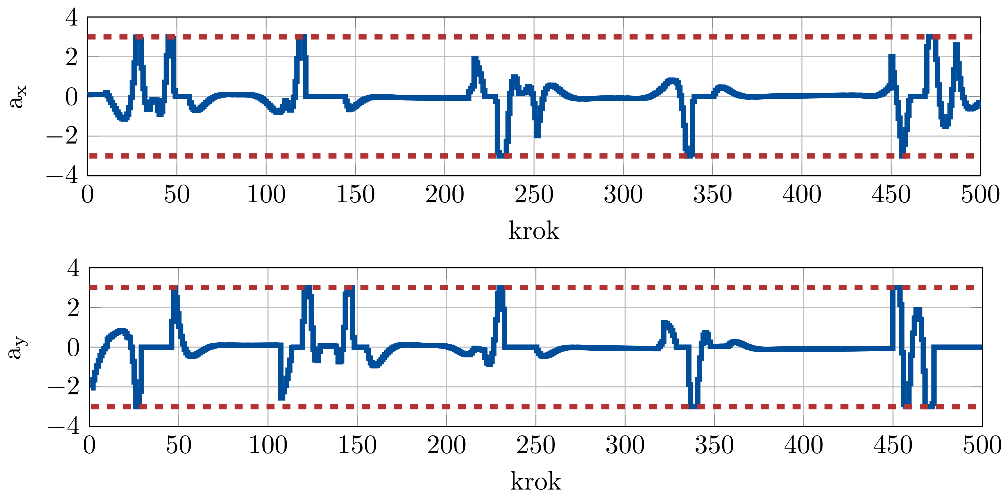
p_x, p_y ——, $p_{x,ref}, p_{y,ref}$ - - -, $p_{x,max/min}, p_{y,max/min}$ - - -

Webová adresa videa: `https://youtu.be/8Nt143Lu6hk`

Kruhová trajektória, MIQP, pohyblivé prekážky

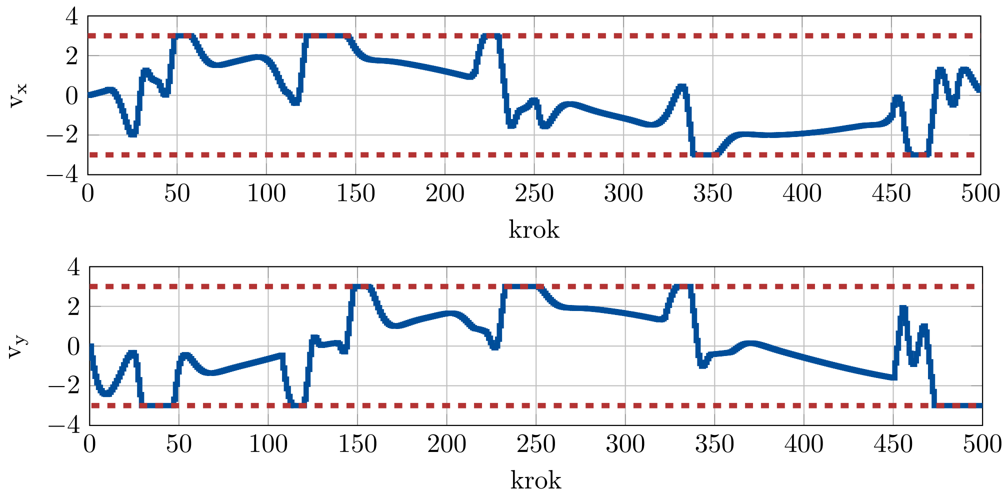


Obr. A.17: Pohyblivé prekážky - kruhová trajektória (MIQP)



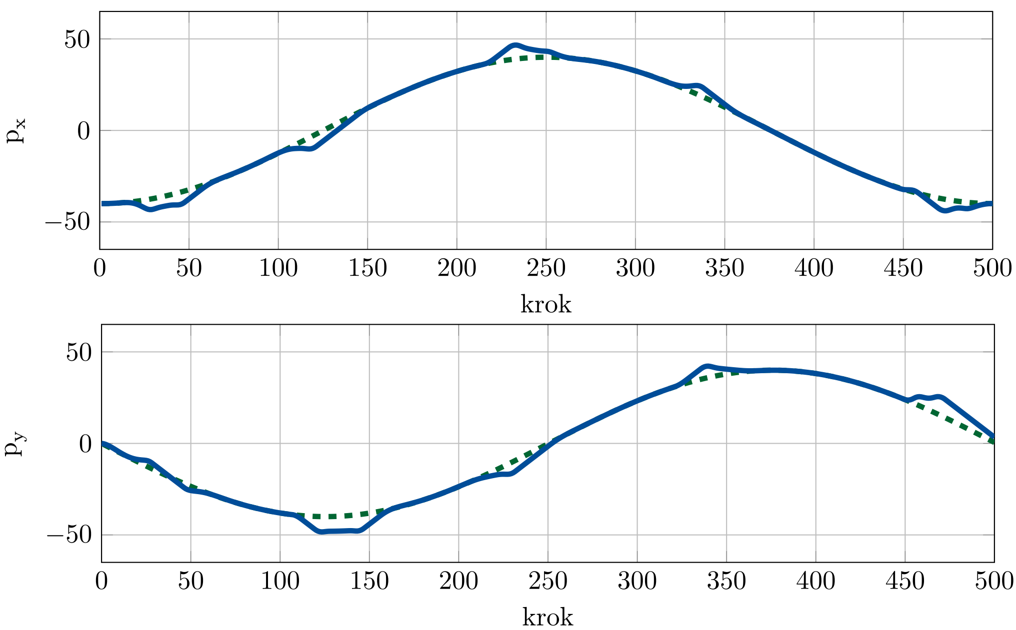
Obr. A.18: Riadiace vstupy - zrýchlenie

a_x, a_y ——, $a_{x,max/min}, a_{y,max/min}$ - - -



Obr. A.19: Prvý stav - rýchlosť

v_x, v_y ——, $v_{x,max/min}, v_{y,max/min}$ - - -

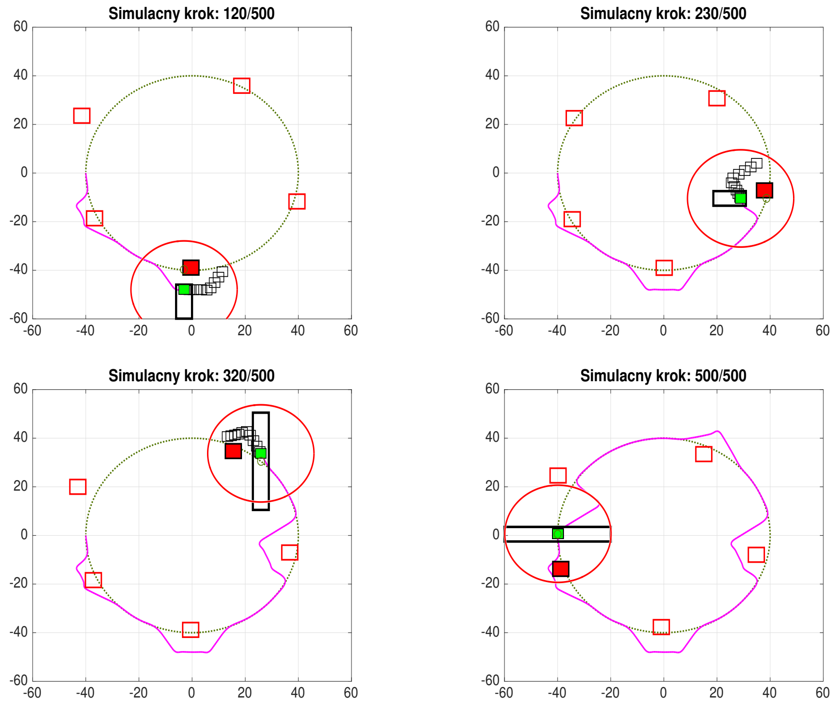


Obr. A.20: Riadené výstupy - poloha

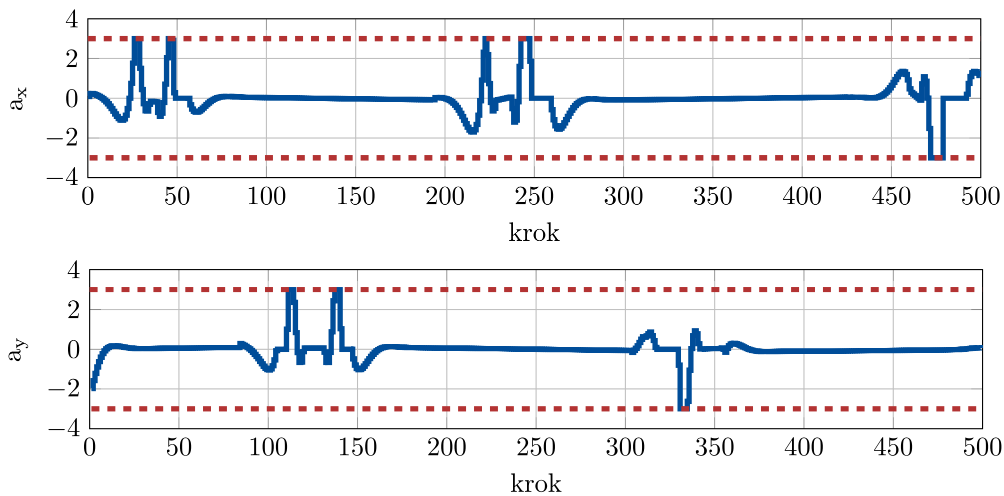
p_x, p_y ——, $p_{x,ref}, p_{y,ref}$ - - -

Webová adresa videa: `https://youtu.be/wrczyr0Y3QY`

Kruhová trajektória, QP, pohyblivé prekážky

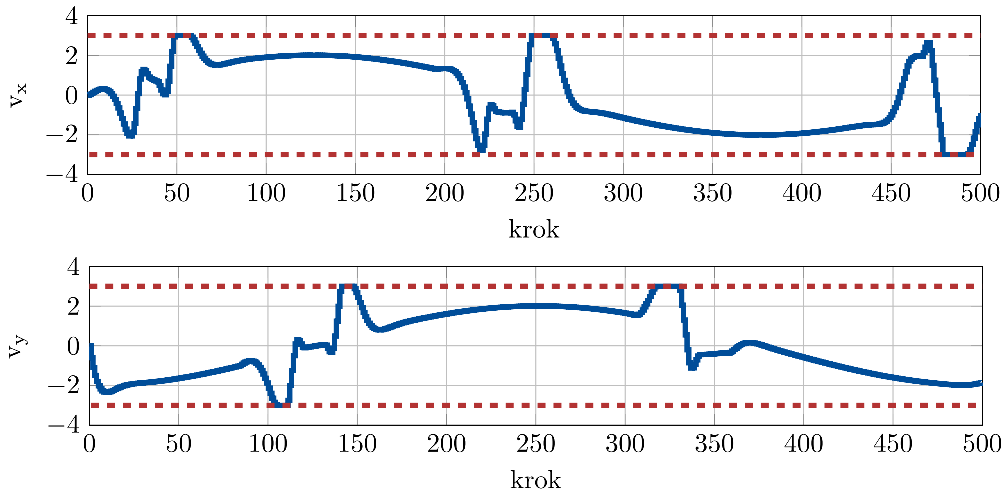


Obr. A.21: Pohyblivé prekážky - kruhová trajektória (QP)



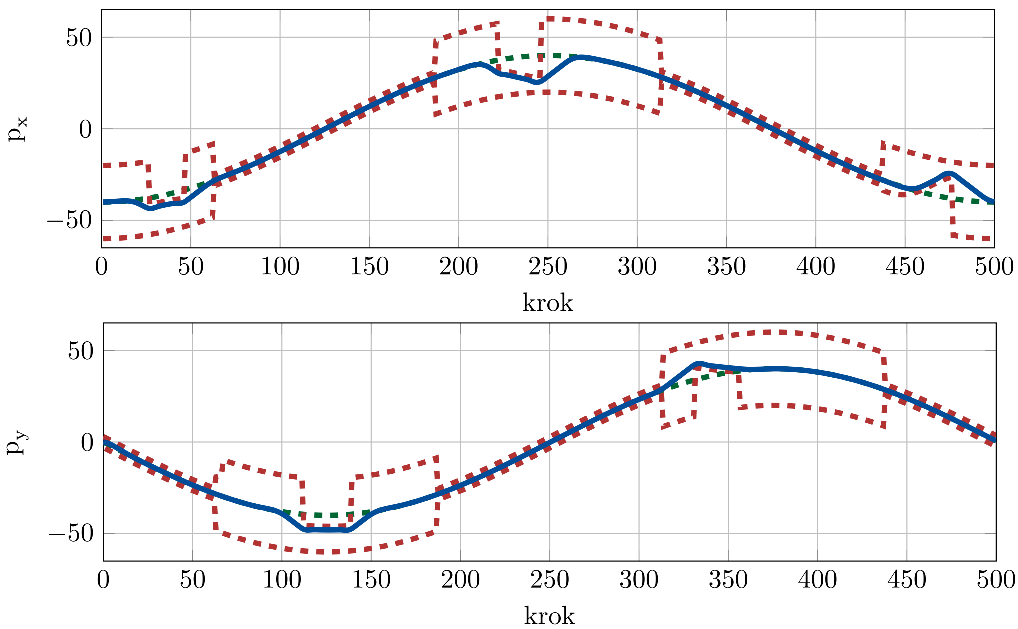
Obr. A.22: Riadiace vstupy - zrýchlenie

a_x, a_y ——, $a_{x,max/min}, a_{y,max/min}$ - - -



Obr. A.23: Prvý stav - rýchlosť

v_x, v_y ▬, $v_{x,max/min}, v_{y,max/min}$ ▪▪▪

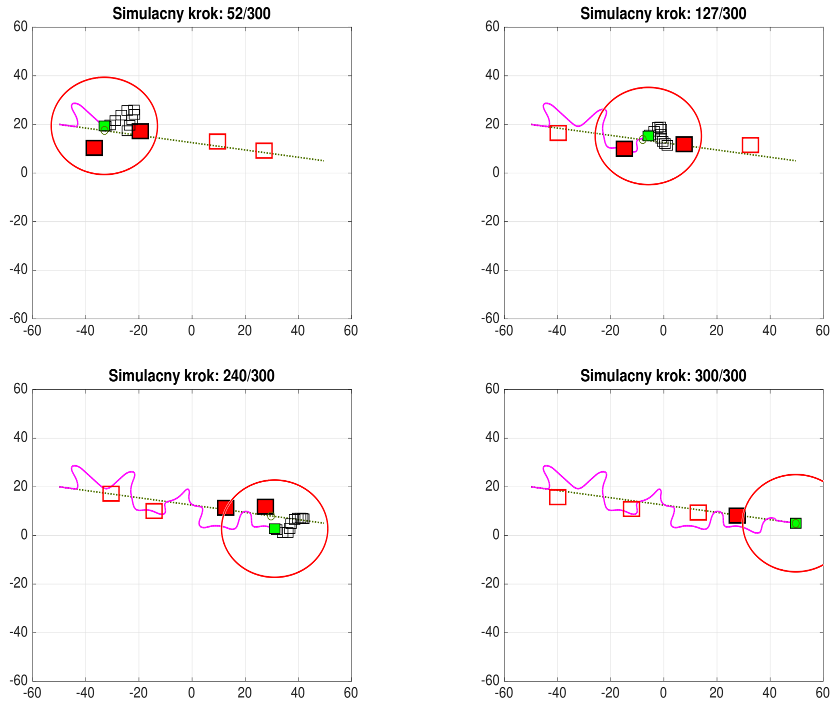


Obr. A.24: Riadené výstupy - poloha

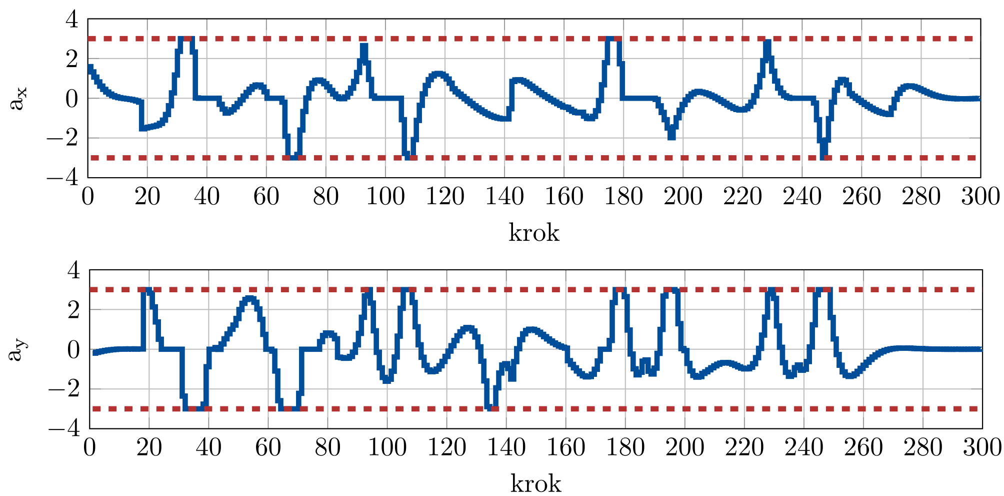
p_x, p_y ▬, $p_{x,ref}, p_{y,ref}$ ▪▪▪, $p_{x,max/min}, p_{y,max/min}$ ▪▪▪

Webová adresa videa: `https://youtu.be/5stuUx9REg8`

AB trajektória, MIQP, pohyblivé prekážky

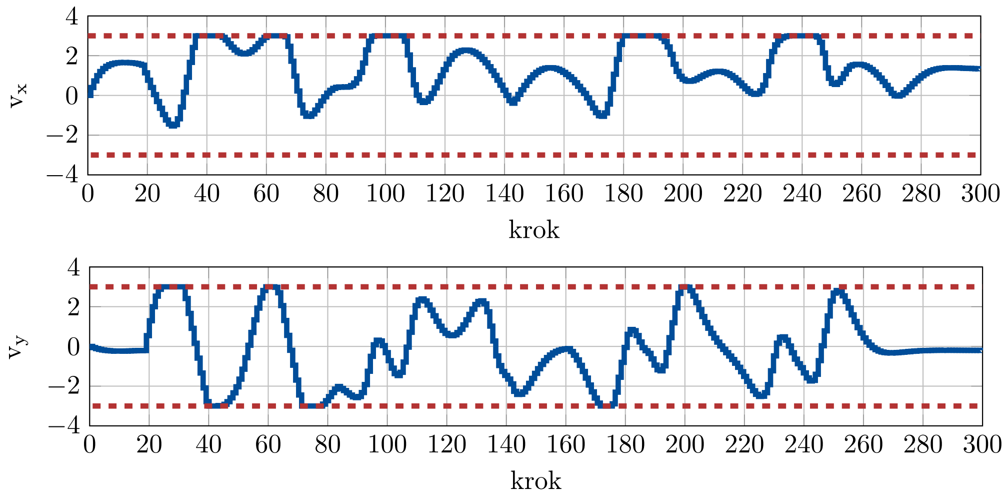


Obr. A.25: Pohyblivé prekážky - AB trajektória (MIQP)



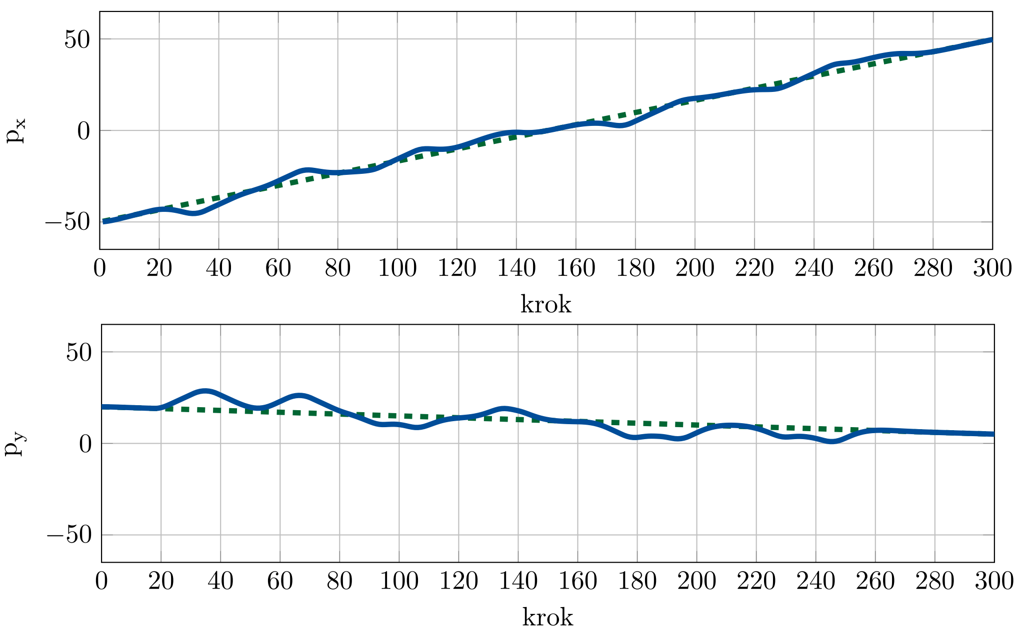
Obr. A.26: Riadiace vstupy - zrýchlenie

a_x, a_y ——, $a_{x,max/min}, a_{y,max/min}$ - - -



Obr. A.27: Prvý stav - rýchlosť

v_x, v_y , $v_{x,max/min}, v_{y,max/min}$

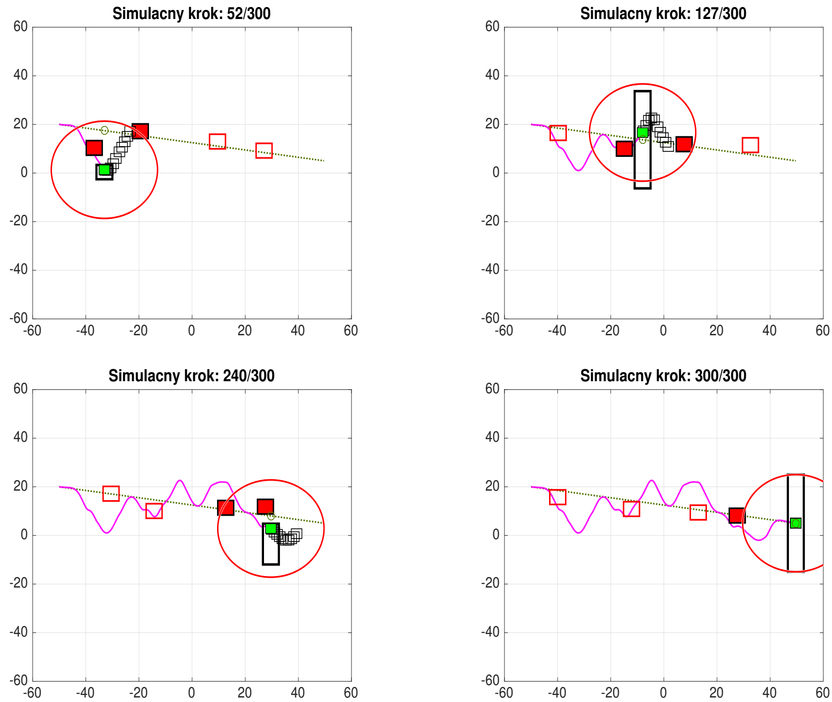


Obr. A.28: Riadené výstupy - poloha

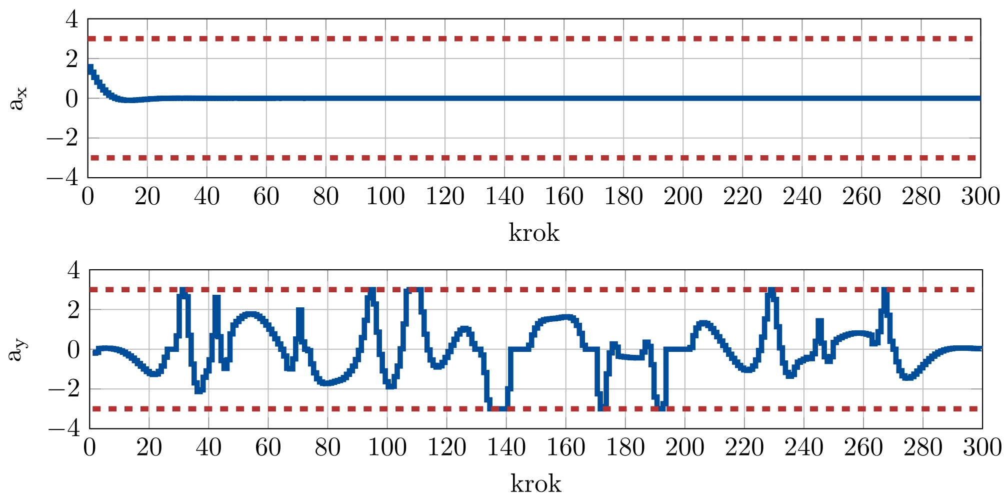
p_x, p_y , $p_{x,ref}, p_{y,ref}$

Webová adresa videa: `https://youtu.be/eecHgGxKztM`

AB trajektória, QP, pohyblivé prekážky

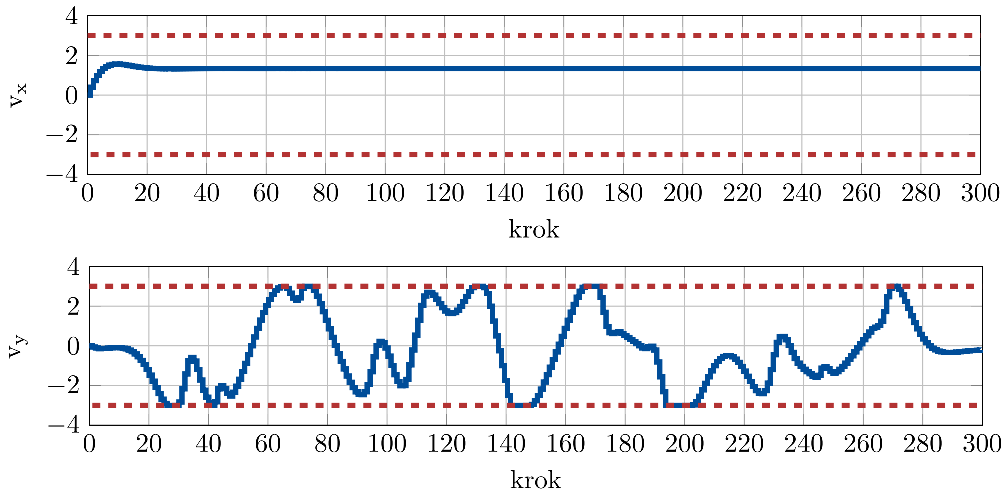


Obr. A.29: Pohyblivé prekážky - AB trajektória (QP)



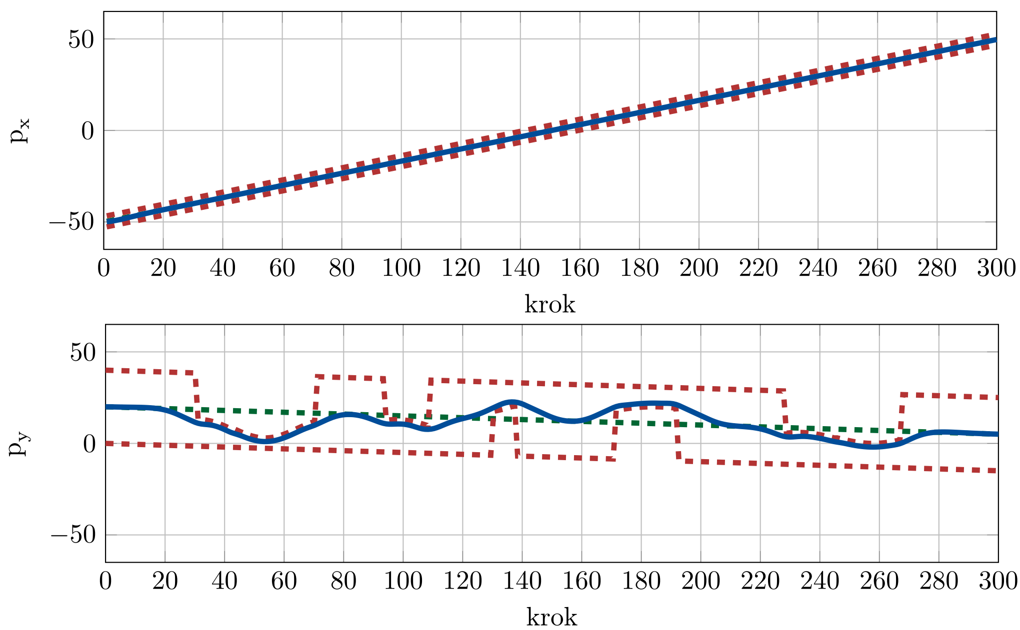
Obr. A.30: Riadiace vstupy - zrýchlenie

a_x, a_y ——, $a_{x,max/min}, a_{y,max/min}$ - - -



Obr. A.31: Prvý stav - rýchlosť

v_x, v_y ——, $v_{x,max/min}, v_{y,max/min}$ - - -



Obr. A.32: Riadené výstupy - poloha

p_x, p_y ——, $p_{x,ref}, p_{y,ref}$ - - -, $p_{x,max/min}, p_{y,max/min}$ - - -

Webová adresa videa: `https://youtu.be/1OaemYh8ZQE`

Literatúra

M. Arnold a G. Andersson (2011). Model Predictive Control of Energy Storage including Uncertain Forecasts. *The 17th Power Systems Computation Conference.*

S. P. Boyd a L. Vandenberghe (2004). *Convex optimization.* Cambridge university press, Cambridge, UK. ISBN 0 521 83378 7.

R. Deits a R. Tedrake (2015). Efficient Mixed-Integer Planning for UAVs in Cluttered Environments. URL `http://groups.csail.mit.edu/robotics-center/public_papers/Deits15.pdf`.

A. Karas, B. Rohaľ-Ilkiv a C. Belavý (2007). *Praktické aspekty prediktívneho riadenia.* STU, Bratislava. ISBN 978-80-89316-06-9.

A. G. Kuznetsov (1996). Constrained predictive control: a brief survey. *Journal A*, 37(2):3–8.

M. Kvasnica (2012a). Model Predictive Control (MPC), Part 1: Introduction.

M. Kvasnica (2012b). Model Predictive Control (MPC), Part 2: Advanced Techniques.

K. R. Muske a T. A. Badgwell (2002). Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12(5):617–632.

M. Nikolaou (2001). Model predictive controllers: A critical synthesis of theory and industrial needs. *Advances in Chemical Engineering*, 26:131–204.

J. A. Rossiter (2005). *Model-based predictive control: a practical approach.* Taylor & Francis e-Library. ISBN 0-8493-1291-4.

S. Vichik a F. Borrelli (2014). Solving linear and quadratic programs with an analog circuit. *Computers and Chemical Engineering*, 70:160–171.

H. P. Williams (2013). *Model Building in Mathematical Programming.* John Wiley & Sons Ltd, 5. vyd. ISBN 978-1-118-44333-0.