

Diplomová práce

# **Prediktivní řízení laterální dynamiky automobilu**

*Matouš Raisigl*



Květen 2015

vedoucí práce:  
Ondřej Šantin

České vysoké učení technické v Praze  
Fakulta elektrotechnická, Katedra řídicí techniky

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra řídicí techniky

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Matouš Raisigl**

Studijní program: Kybernetika a robotika  
Obor: Systémy a řízení

Název tématu: **Prediktivní řízení laterální dynamiky automobilu**

Pokyny pro vypracování:

1. Seznamte se s literaturou zabývající se modelováním dynamiky vozu a jeho stabilizací.
2. Vytvořte model vhodný pro prediktivní regulátor.
3. Navrhněte a implementujte základní prediktivní regulátor.
4. Proveďte analýzu citlivosti kvality výsledného řízení na redukci problému prediktivního regulátoru. Zaměřte se na kvalitu splnění měkkých omezení na výstupy.

Seznam odborné literatury:

- [1] Rossiter, J. A. (2003). Model-Based Predictive Control: A Practical Approach. CRC Press.
- [2] Besselmann, T., Rostalski, P., & Morari, M. (2007). Hybrid Parameter-Varying Model Predictive Control for Lateral Vehicle Stabilization. In European Control Conference.
- [3] Falcone, P., Tufo, M., Borrelli, F., Asgari, J., & Tseng, H. E. (2007). A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. 46th IEEE Conference on Decision and Control, 2980–2985.
- [4] Keviczky, T., Falcone, P., Borrelli, F., Asgari, J., & Hrovat, D. (2006). Predictive control approach to autonomous vehicle steering. In American Control Conference (p. 6 pp.).

Vedoucí: Ing. Ondřej Šantin

Platnost zadání: do konce letního semestru 2015/2016



prof. Ing. Michael Šebek, DrSc.  
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 29. 9. 2014

## **Prohlášení**

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, dne: .....

Podpis: .....

## **Poděkování**

Rád bych poděkoval vedoucímu Ondřeji Šantinovy za vedení a rady při vypracování práce. Také děkuji rodině a přátelům za trpělivost a podporu při studiu.

## Abstrakt

V této práci se zabýváme MPC regulátorem, což je moderní regulátor, kde je regulační úloha převedena na řešení optimalizačního problému, jehož řešení je použito jako vstup do systému. Optimalizační problém se musí vyřešit v každé vzorkovací periodě, proto je nejvýhodnější mít co nejmenší optimalizační problém. V této práci jsme se věnovali dvěma způsobům snížení velikosti optimalizačního problému, prvním je blokování vstupů, kdy redukuje počet kroků, kdy můžeme měnit vstup do systému. Druhým je snížit počet kroků v predikci, kde kontrolujeme překročení měkkých omezení na výstupu, toto se nazývá redukce měkkých omezení.

Snížování velikosti regulačního problému vede k zhoršování výkonu regulátoru. Zhoršuje se sledování reference a dochází k překročení měkkých omezení.

Cílem této práce je implementovat MPC regulátor a prozkoumat vliv blokování vstupů a redukce měkkých omezení na výkon MPC regulátoru a nalézt vhodnou kombinaci blokování a redukce omezení, které by zmenšilo problém ale zároveň přijatelně sledovalo referenci a plnilo měkké omezení.

Implementovali jsme MPC regulátor, který používá linearizaci podél trajektorie nelineárního systému automobilu, ale pro zjednodušení jsme citlivosti získané v prvním kroku aplikovali po celou dobu predikce. Tento regulátor pracuje s tvrdými omezeními na vstupu systému a měkkými omezeními na výstupu systému.

Poté jsme se věnovali vlivu změn blokování vstupů a redukci měkkých omezení na schopnost regulátoru splnit měkká omezení a sledovat referenci.

Podařilo se nám otestovat a ověřit chování regulátoru a nalézt vhodné nastavení regulátoru.

## Klíčová slova

MPC, blokování vstupů, redukce omezení, měkká omezení, tvrdá omezení, RMSE, actuator activity, model automobilu, model pneumatiky, regulace

## Abstrakt

The focus of this work is on a modern MPC regulator which finds solutions by transforming the regulation problem to an optimization problem. The solutions of these optimization problem are then used as inputs to the system. The optimization problem has to be computed every sampling period, hence the optimization problem order must be as small as possible. There are two ways to reduce the optimization problem size. The first is to reduce the number of steps in the input to the system (input blocking). The second is to reduce the number of steps in the prediction, using the reduction of the so called soft limitations.

Reducing the regulation problem size worsens the regulator performance in following the reference and in fulfilling the soft limitations.

The purpose of this work is to implement an MPC regulator, to investigate the influence of input blocking and soft limitations reduction on the performance of the regulator and to find an optimal combination of input blocking and soft limitations reduction which would reduce the problem size while still exhibiting a sufficient performance.

We implemented an MPC regulator which is based on linearization along a trajectory of a non-linear automobile system. For simplicity the sensitivities obtained in the first step are applied during the whole process of prediction. This regulator implements hard limits on the system input and soft limits on the system output.

After that we investigated the influence of input blocking and soft limitations reduction. We successfully tested the performance of the regulator after implementing proper problem reduction.

## Keywords

MPC, block inputs, reduced constraints, soft constraints, hard constraints, RMSE, actuator activity, car model, pneu model, control

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Model</b>	<b>2</b>
2.1	Nelineární model . . . . .	2
2.2	Lineární model . . . . .	2
2.2.1	Linearizace . . . . .	3
	Linearizace v ekvilibriu . . . . .	3
	Linearizace v okolí trajektorie . . . . .	4
<b>3</b>	<b>Prediktivní řízení</b>	<b>5</b>
3.1	Formulace problému . . . . .	5
3.1.1	Základní případ . . . . .	6
3.1.2	Tvrdá omezení . . . . .	6
3.1.3	Měkká omezení . . . . .	7
3.1.4	Případ s omezeními . . . . .	7
3.1.5	Sledování reference . . . . .	7
3.1.6	Integrální akce . . . . .	8
3.2	Kvadratické programování a lineární MPC . . . . .	8
3.2.1	Vektor optimalizace . . . . .	9
3.2.2	Maticový zápis kritériální funkce . . . . .	9
3.2.3	Predikce vývoje systému . . . . .	9
3.2.4	Vážení sledování reference . . . . .	10
3.2.5	Vážení výstupů . . . . .	10
3.2.6	Vážení splnění měkkých omezení . . . . .	11
3.2.7	Tvrdá a měkká omezení v QP řešiteli . . . . .	11
3.3	Rozšíření MPC na linearizaci podél trajektorie . . . . .	12
3.3.1	Aproximace pomocí LVT . . . . .	13
3.3.2	Kvadratické programování a LVT MPC . . . . .	14
3.4	Nástroje pro redukci velikosti problému . . . . .	14
3.4.1	Redukovaná měkká omezení . . . . .	15
3.4.2	Blokování vstupů . . . . .	17
3.4.3	Normalizace kritériální funkce . . . . .	18
<b>4</b>	<b>Model automobilu</b>	<b>20</b>
4.1	Nelineární model . . . . .	20
4.2	Model pneumatik . . . . .	23
<b>5</b>	<b>Analýza citlivosti kvality řízení na redukci problému</b>	<b>25</b>
5.1	Cíle řízení . . . . .	25
5.2	Parametry simulace . . . . .	26
5.2.1	Konstanty modelu automobilu . . . . .	26
5.2.2	Vstupy a výstupy modelu . . . . .	26
5.2.3	Nastavení regulátoru . . . . .	26
5.3	Vliv blokování vstupu . . . . .	28
5.4	Vliv redukce měkkých omezení . . . . .	31
5.5	Vliv kombinace blokování vstupu a redukce měkkých omezení . . . . .	35
5.5.1	Vliv změny blokování na splnění měkkých omezení . . . . .	35
5.5.2	Vliv umístění kontroly překročení měkkých omezení . . . . .	35

<b>6 Závěr</b>	<b>42</b>
<b>Literatura</b>	<b>43</b>
<b>Přílohy</b>	



# 1 Úvod

Mezi moderní regulátory patří diskrétní regulátory, které využívají k řízení výpočet vývoje systému na základě znalosti modelu systému. Tento typ regulátorů se nazývá prediktivní regulátor nebo MPC z anglického Model Prediction Control. Pokud známe model systému, který popisuje vývoj systému z počátečního stavu a vstupů systému, můžeme formulovat optimalizační úlohu, jež minimalizuje zadané kritérium změnou vstupů systému. Kritérium většinou bývá ve tvaru kvadratické formy. Ke správně funkci MPC regulátoru tedy potřebujeme znát správný model systému a současný stav systému, ten obvykle změříme, nebo odhadneme. Uzavření regulační smyčky provádíme technikou klouzavého horizontu, kdy vypočteme ideální trajektorii vstupů systému, ale aplikujeme pouze jeden krok. Poté znovu změříme stav systému, provedeme výpočet, aplikujeme první krok vstupů a tak stále znovu [18, 12].

Mezi hlavní výhody MPC patří schopnost regulovat systémy s více vstupy a výstupy (MIMO). Mezi další výhody patří schopnost do výpočtu zahrnout omezení jak na vstupech, tak výstupech systému. Tyto omezení mohou být tvrdá, kdy nesmí být vůbec překročena, nebo mohou být měkká, kdy jejich překročení je penalizováno. Mezi nevýhody MPC patří výpočetní náročnost optimalizačního problému, protože výpočet se provádí každý krok. Je schopen buď řídit rychlé systémy s malým optimalizačním problémem, nebo pomalé systémy s větším optimalizačním problémem. Pro snížení výpočetní náročnosti se používá takzvané blokování vstupů, kdy omezujeme možnost změny vstupu [4]. Další je redukce měkkých omezení, kdy překročení omezení kontrolujeme jen v některých krocích [1]. Obě možnosti redukování problému obvykle vedou ke zhoršení výkonu regulátoru.

Nejjednodušší řešení má MPC regulátor v případě, že neuvažujeme žádná omezení. Toto řešení vede na konstantní zpětnou vazbu ze stavu, pokud budeme prodlužovat dobu predikce do nekonečna, dostaneme řešení LQR regulátoru [12, kapitola 3]. Konstantní zpětná vazba má výhodu v tom, že se jedná o lineární regulátor a k analýze můžeme použít teorii lineárních systémů. Naproti tomu se vzdáváme možnosti uvažovat omezení, což je jedna z největších výhod MPC.

Jestliže uvažujeme omezení, musí MPC regulátor vyřešit problém kvadratického programování. Mnoho optimalizačních problémů vede na kvadratické programování, proto existuje mnoho různých řešitelů např. v optimalizačním toolboxu Matlabu [17], qpOASES [7], Maple, Mathematica a další. Řešitelé se liší algoritmem, který je použit k nalezení optimálního řešení. Algoritmů existuje celá řada, některé jsou popsány v [14, 16], v naší práci používáme active set method viz [8]. Kvadratický problém může být vypočten buď online (v tomto případě potřebujeme výkonný procesor), nebo offline (v tomto případě potřebujeme více paměti)[3].

V této práci se zabýváme zkoumáním splnění měkkých omezení v závislosti na blokování vstupů a redukcí měkkých omezení.

## 2 Model

Model je zjednodušená verze reality, která by měla zachycovat vztahy z reálného světa, které chceme studovat. Při tvorbě modelu musíme postupovat tak, aby byli zachovány vlastnosti, které chceme pomocí modelu studovat a ty které nechceme studovat tak zanedbáme. Pro popis fyzikálních systémů používáme matematické modely. U matematického modelu jsou vstupy a výstupy reprezentovány jako čísla. Model pomocí matematických funkcí přepočítá vstup na výstup.

### 2.1 Nelineární model

Nejobecnější model je nelineární časově proměnný model. Mějme vektor  $x \in \mathbb{R}^{n_x}$ ,  $n_x \in \{0 + \mathbb{N}\}$ . Vektor  $x$  obsahuje historii systému, jedná se o nejmenší vektor, který může obsahovat kompletní historii systému. Pokud je velikost  $n_x$  rovna 0, tak systém nemá žádnou dynamiku. Vstup je vektor  $u$  velikosti  $u \in \mathbb{R}^{n_u}$ ,  $n_u \in \mathbb{N}$ . Také mějme výstupní vektor  $y \in \mathbb{R}^{n_y}$ ,  $n_y \in \mathbb{N}$ . Poté můžeme nelineární model zapsat jako:

$$\dot{x}(t) = f(x(t), u(t), t), \quad (1a)$$

$$y(t) = h(x(t), u(t), t). \quad (1b)$$

Proměnná  $t$  vyjadřuje čas. Tento model je časově proměnný, tedy není časově invariantní, jeho výstup závisí na absolutní hodnotě času. Pokud je systém časově nezávislý, jeho rovnice se zjednoduší na:

$$\dot{x}(t) = f(x(t), u(t)), \quad (2a)$$

$$y(t) = h(x(t), u(t)). \quad (2b)$$

U nelineárního systému také může vektor  $x$  a  $u$  náležet jen do určité podmnožiny  $\mathbb{R}^{n_x}$  a  $\mathbb{R}^{n_u}$ .

### 2.2 Lineární model

Nelineární model není obecně jednoduše řešitelný, některé jsme schopni vyřešit analyticky ale často jsme ho schopni řešit jen numericky. Z těchto důvodů se provádí linearizace modelu, která nelineární model převede na lineární. Aby zobrazení  $\mathcal{L} : A \rightarrow B$ , kde  $A, B$  jsou lineární prostory, bylo lineární, musí splňovat:

$$\alpha \mathcal{L}(x) = \mathcal{L}(\alpha x), \quad \alpha \in \mathbb{R}, x \in A, \quad (3a)$$

$$\mathcal{L}(x) + \mathcal{L}(y) = \mathcal{L}(x + y), \quad x, y \in A. \quad (3b)$$

Pokud je model lineární zobrazení, tak se jedná o lineární model. Díky tomu, že lineární zobrazení je zobrazení z lineárního prostoru do lineárního prostoru, tedy  $x$  a  $u$  už nemohou u lineárního modelu být jen z určité podmnožiny  $\mathbb{R}^{n_x}$  a  $\mathbb{R}^{n_u}$ .

### 2.2.1 Linearizace

Lineární model z nelineárního modelu získáme pomocí procesu zvaného linearizace. Existují dva způsoby linearizace. Linearizace v ekvilibriu a linearizace podél trajektorie.

#### Linearizace v ekvilibriu

Pro linearizaci časově neproměnného systému v ekvilibriu platí:

$$0 = f(x_0, u_0), \quad (4a)$$

$$y_0 = h(x_0, u_0). \quad (4b)$$

Tedy ve stavu  $x_0$  a vstupu  $u_0$  je systém v klidu, jeho stav se nemění. Nalezení kombinace  $x_0$  a  $u_0$  nemusí být vždy jednoduché a ani nemusí existovat. Platnost linearizace v ekvilibriu platí také jen v nějakém okolí daného ekvilibria. Pokud tedy nelineární systém pracuje ve velkém rozsahu stavů a vstupů, nemusí být linearizace v ekvilibriu vhodná. Zavedeme difference  $\Delta u = u - u_0$  a  $\Delta x = x - x_0$ ,  $\Delta y = y - y_0$ . Linearizaci dostaneme pomocí taylorova rozvoje a zanedbáním vyšších členů než prvního řádu.

$$\begin{aligned} \Delta \dot{x} = \dot{x} - 0 &= f(\Delta x + x_0, \Delta u + u_0) - f(x_0, u_0) = f(\Delta x, \Delta u + u_0) = \\ &= f(x_0, u_0) + \frac{\partial f}{\partial x}(x_0, u_0)\Delta x + \frac{\partial f}{\partial u}(x_0, u_0)\Delta u + \Theta(x_0, u_0) - f(x_0, u_0) \approx \\ &= \frac{\partial f}{\partial x}(x_0, u_0)\Delta x + \frac{\partial f}{\partial u}(x_0, u_0)\Delta u = \\ &= \mathbf{A}\Delta x + \mathbf{B}\Delta u. \end{aligned}$$

Takže platí:

$$\mathbf{A} = \frac{\partial f}{\partial x}(x_0, u_0),$$

$$\mathbf{B} = \frac{\partial f}{\partial u}(x_0, u_0).$$

Obdobně dostaneme pro výstup:

$$\mathbf{C} = \frac{\partial h}{\partial x}(x_0, u_0),$$

$$\mathbf{D} = \frac{\partial h}{\partial u}(x_0, u_0).$$

Tedy kompletní lineární časově neproměnný systém (LTI - linear time-invariant) je v ekvilibriu  $x_0$ ,  $y_0$  a  $u_0$  popsán jako:

$$\Delta \dot{x} = \mathbf{A}\Delta x + \mathbf{B}\Delta u, \quad (5a)$$

$$\Delta y = \mathbf{C}\Delta x + \mathbf{D}\Delta u. \quad (5b)$$

Často se vynechává symbol  $\Delta$  v rovnicích, ale vždy si musíme pamatovat, že pracujeme s diferenciálním modelem.

### Linearizace v okolí trajektorie

Pokud systém nemá ekvilibrium, nebo se hodnoty stavů a vstupů mění ve velkém rozsahu, můžeme použít linearizaci podél trajektorie. V tomto případě máme zadanou nějakou trajektorii stavu, vstupu, nebo výstupu a zbytek neznámých trajektorií dopočítáme tak, aby platili rovnice (1a) a (1b). Např. při vypouštění rakety spočteme optimální trajektorii jejího letu a podél této trajektorie linearizujeme nelineární systém.

Mějme tedy zadanou nebo vypočtenou požadovanou trajektorii stavů ( $\bar{x}(t)$ ) a vstupu ( $\bar{u}(t)$ ). Nyní zavedme difference  $\Delta u = u - \bar{u}$  a  $\Delta x = x - \bar{x}$ ,  $\Delta y = y - \bar{y}$ .

$$\Delta \dot{x}(t) = \dot{x}(t) - \dot{\bar{x}}(t) = f(\Delta x(t) + \bar{x}(t), \Delta u(t) + \bar{u}(t), t) - f(\bar{x}(t), \bar{u}(t), t). \quad (6)$$

Výraz (6) můžeme rozepsat do Taylorova rozvoje a zkrátit rozvoj pouze na první stupeň:

$$\Delta \dot{x}(t) = f(\Delta x(t) + \bar{x}(t), \Delta u(t) + \bar{u}(t), t) - f(\bar{x}(t), \bar{u}(t), t) \quad (7)$$

$$= f(\bar{x}(t), \bar{u}(t), t) + \frac{\partial f}{\partial x}(\bar{x}(t), \bar{u}(t), t)\Delta x(t) + \frac{\partial f}{\partial u}(\bar{x}(t), \bar{u}(t), t)\Delta u(t) + \Theta(\bar{x}(t), \bar{u}(t), t) - f(\bar{x}(t), \bar{u}(t), t) \quad (8)$$

$$\approx \frac{\partial f}{\partial x}(\bar{x}(t), \bar{u}(t), t)\Delta x(t) + \frac{\partial f}{\partial u}(\bar{x}(t), \bar{u}(t), t)\Delta u(t) \quad (9)$$

$$= \mathbf{A}(t)\Delta x(t) + \mathbf{B}(t)\Delta u(t). \quad (10)$$

Podobně můžeme postupovat i u výstupu:

$$\begin{aligned} \Delta y(t) &\approx \frac{\partial h}{\partial x}(\bar{x}(t), \bar{u}(t), t)\Delta x(t) + \frac{\partial h}{\partial u}(\bar{x}(t), \bar{u}(t), t)\Delta u(t) \\ &= \mathbf{C}(t)\Delta x(t) + \mathbf{D}(t)\Delta u(t). \end{aligned} \quad (11)$$

Jednotlivé matice tedy jsou ve tvaru:

$$\mathbf{A} = \frac{\partial f}{\partial x}(\bar{x}(t), \bar{u}(t), t), \quad (12a)$$

$$\mathbf{B} = \frac{\partial f}{\partial u}(\bar{x}(t), \bar{u}(t), t), \quad (12b)$$

$$\mathbf{C} = \frac{\partial h}{\partial x}(\bar{x}(t), \bar{u}(t), t), \quad (12c)$$

$$\mathbf{D} = \frac{\partial h}{\partial u}(\bar{x}(t), \bar{u}(t), t). \quad (12d)$$

V tomto případě se jedná o lineární časově proměnný model (LTV - linear time varying), protože předpokládáme, že se mění matice systému podél vývoje trajektorie, která se mění v čase. Takže se dá tento model zapsat jako:

$$\Delta \dot{x}(t) = \mathbf{A}(t)\Delta x(t) + \mathbf{B}(t)\Delta u(t), \quad (13a)$$

$$\Delta y(t) = \mathbf{C}(t)\Delta x(t) + \mathbf{D}(t)\Delta u(t). \quad (13b)$$

## 3 Prediktivní řízení

Prediktivní regulátor (MPC - Model Prediction Control) využívá k řízení znalost modelu systému a hledá optimální posloupnost řízení vzhledem k definovanému kritériu. Mezi jeho hlavní výhody patří možnost přímo pracovat s MIMO (s více vstupy a výstupy) systémy. Také je schopen zahrnout do výpočtu omezení jak na straně vstupů, tak i na straně výstupů.

Na počátku šedesátých let byl vyvinut optimální lineární kvadratický regulátor (LQR), tento regulátor minimalizuje kvadratickou kritériální funkci stavů a vstupů bez omezení. Mezi jeho výhody patří dobrá stabilita které je dosaženo díky nekonečnému horizontu predikce [19]. Nepočítá však s žádnými omezeními a tak jeho aplikace je problematická kvůli řešení saturací. MPC regulátor byl původně vyvinut pro aplikace v chemickém průmyslu, kde řídil stovky vstupů a výstupů. Na konečném horizontu spočítá optimální posloupnost řízení systému pro otevřenou smyčku, kde počáteční podmínka odpovídá současnému stavu systému. Poté je aplikován první krok řízení a probíhá znovu výpočet na základě nového měření. Problémem optimalizace na konečném horizontu je, že nezaručuje stabilitu systému. Tu lze dosáhnout rozšířením na nekonečný horizont nebo přidáním konečných podmínek [13].

Rozvoj výpočetní techniky umožňuje zvětšovat velikost problému pro řešení. V roce 1979 je prezentován regulátor DMC, který používá lineární přechodový model a kvadratickou kritériální funkci. Regulátor počítá optimální výsledek jako úlohu nejmenších čtverců. V roce 1983 byl popsán algoritmus QDMC, který umožňoval systematicky zahrnout vstupní a výstupní omezení.

Na konci osmdesátých let byl uveden Shell Multivariable Optimizing Controller (SMOC), který používal stavový popis systému, takže může reprezentovat kompletní dynamiku (včetně nepozorovatelných módů) systému.

### 3.1 Formulace problému

Základem MPC regulátoru je diskretní model, který se používá k výpočtu vývoje systému (od toho název model prediction). Vstupem do toho modelu je počáteční stav  $x_0$  modelu, který odpovídá současnému stavu systému. Dalším vstupem je posloupnost řízení  $u(k)$ , kde jsou uloženy vstupy v kroku  $k$ . Vývoj systému se provádí v konečném horizontu o délce  $T_p$ , který se nazývá *doba predikce*. Na tomto horizontu vypočteme výstup systému. Dále mějme nějakou kritériální funkci, která ohodnotí výstup a vstup nějakým číslem. Dále mějme nějaký řešitel, který změnou vstupní posloupnosti minimalizuje kritériální funkci. V okamžiku nalezení minima je aplikován první krok řízení a systém se posouvá o krok dál. Opět je získán (změřen, nebo odhadnut) současný stav systému  $x_0$  a výpočet probíhá znovu. Tento postup se nazývá klouzavý horizont a dochází tím k uzavření regulační smyčky.

Regulátory MPC můžou být nelineární, nicméně v této práci se budeme zabývat pouze lineárním MPC, více o nelineárním MPC např v [9].

### 3.1.1 Základní případ

Mějme definován diskrétní lineární model systému ve tvaru:

$$x(k+1) = \mathbf{M}x(k) + \mathbf{N}u(k), \quad (14a)$$

$$y(k) = \mathbf{C}x(k) + \mathbf{D}u(k). \quad (14b)$$

Rovnice (14a) nám tvoří omezení vývoje systému. Sloupcový vektor  $x(k)$  popisuje stav systému v čase  $k$ , jeho velikost je  $n_x \in \mathbb{Z}$  a toto číslo odpovídá řádu systému. Sloupcový vektor  $u(k)$  je řízení aplikované na systém v čase  $k$  a jeho velikost je  $n_u \in \mathbb{Z}$ , což odpovídá počtu vstupů systému. Sloupcový vektor  $y(k)$  jsou výstupy systému v čase  $k$ , jeho velikost je  $n_y \in \mathbb{N}$ , což odpovídá počtu výstupů systému. Stavová matice  $\mathbf{M}$  je čtvercová matice rozměru  $n_x \times n_x$ . Matice  $\mathbf{N}$  je matice řízení a její rozměr je  $n_x \times n_u$ . Matice  $\mathbf{C}$  a  $\mathbf{D}$  jsou maticemi výstup a jejich rozměry jsou  $n_y \times n_x$  a  $n_y \times n_u$ .

Nyní mějme definovanou kriteriální funkci:

$$J = \sum_{k=0}^{T_p-1} \frac{1}{2} \left( y^T(k) \mathbf{Q} y(k) + u^T(k) \mathbf{R} u(k) \right). \quad (15)$$

Matice  $\mathbf{Q}$  je symetrická matice o rozměru  $n_y \times n_y$ , která musí být pozitivně semi/definitní. Matice  $\mathbf{R}$  je symetrická matice velikosti  $n_u \times n_u$ , která musí být pozitivně definitní. Kriteriální funkce je součet dvou kvadratických forem přes celou dobu predikce  $T_p$ . Kvadratické formy mají tu výhodu, že za jasně definovaných podmínek mají zajištěnou existenci minima. První člen kriteriální funkce váží výstup systému, tedy hodnoty  $\mathbf{Q}$  ovlivňují výstupy. Druhý člen váží vstupy systému, tedy váhy  $\mathbf{R}$  ovlivňují vstup systému.

Definujme posloupnost řízení  $U$  jako uspořádanou posloupnost  $(u(0), u(1), \dots, u(T_p - 1))$ . Nyní můžeme zapsat úlohu MPC regulátoru jako:

$$\min_U \sum_{k=0}^{T_p-1} \frac{1}{2} \left( y^T(k) \mathbf{Q} y(k) + u^T(k) \mathbf{R} u(k) \right), \quad (16a)$$

$$\text{s.t. } x(k+1) = \mathbf{M}x(k) + \mathbf{N}u(k), k = 0, 1, \dots, T_p - 1, \quad (16b)$$

$$y(k) = \mathbf{C}x(k) + \mathbf{D}u(k), k = 0, 1, \dots, T_p - 1. \quad (16c)$$

Řešení této úlohy vede na časově proměnnou zpětnou vazbu, pokud bychom prodlužovali dobu predikce až do nekonečna a vzali výsledné ustálené řešení, dostali bychom řešení LQR problému, tedy konstantní zpětnou vazbu ze stavu. MPC nám však umožňuje zavést omezení. Omezení můžou být aplikována jak na vstup, tak na výstup. Dají se rozdělit do dvou kategorií, na tvrdá omezení, která se nedají za žádných okolností překročit a na měkká omezení, která se dají překročit s určitou penalizací.

### 3.1.2 Tvrdá omezení

Tvrdá omezení se nedají překročit, jsou aplikována na vstup. Vyjadřují maximální akční zásahy vstupních členů, které se nedají překročit. Jejich matematický zápis je:

$$u_{min} \leq u(k) \leq u_{max}, k = 1, 2, \dots, T_p - 1. \quad (17)$$

Tedy řízení v každém kroku musí být mezi minimální a maximální hodnotou.

### 3.1.3 Měkká omezení

Měkká omezení umožňují překročit hodnotu omezení, ale s určitou penalizací. Jsou aplikovány na výstup. Vlivem chyb modelu a měření nejsme v reálném systému schopni zaručit splnění tvrdých omezení na výstup, mohla by nastat situace že by vlivem poruchy došlo k jejich překročení a tedy získali bychom nesplnitelnou trajektorii systému. Z tohoto důvodu se zavádí „soft omezení“, kde je překročení omezení penalizováno a výsledný akční zásah je vždy přípustný. Omezení by mělo splňovat:

$$y_{min} \leq y(k) \leq y_{max}, k = 1, 2, \dots, T_p - 1. \quad (18)$$

V této formulaci odpovídají tvrdým omezením, abychom dosáhli požadovaného chování, musíme přidat pomocnou proměnnou „slack variables“  $z(k)$ , kde velikost vektoru  $z(k)$  je  $n_y$ , vyjadřující povolené meze hodnoty výstupu  $y(k)$ . Poté přidáme vážení rozdílů výstupu a proměnné  $z(k)$ . Matematicky formulován je tento princip v článku [10]:

$$\min_{U, Z} \frac{1}{2} \sum_{k=0}^{T_p-1} (y(k) - z(k))^T \mathbf{\Omega} (y(k) - z(k)) \quad (19a)$$

$$\text{s.t. } x(k+1) = \mathbf{M}x(k) + \mathbf{N}u(k), k = 0, 1, \dots, T_p - 1, \quad (19b)$$

$$y(k) = \mathbf{C}x(k) + \mathbf{D}u(k), k = 0, 1, \dots, T_p - 1 \quad (19c)$$

$$y_{min} \leq z(k) \leq y_{max}, k = 0, 1, \dots, T_p - 1. \quad (19d)$$

Tedy, pokud je výstup  $y(k)$  v povolených mezích, lze pomocí  $z(k)$  vynulovat rozdíl  $y(k) - z(k)$ . Pokud nejde rozdíl vynulovat, tak je výstup  $y(k)$  mimo povolené meze a člen (19a) je nenulový, tedy ovlivňuje kritériální funkci. Rozdíl  $y(k) - z(k)$  je vážen maticí  $\mathbf{\Omega}$ , která má rozměry  $n_y \times n_y$ . Opět můžeme pro  $z(k)$  vytvořit posloupnost  $Z = (z(0), z(1), \dots, z(T_p - 1))$ , která nám umožní zjednodušit zápis kritéria.

### 3.1.4 Příklad s omezeními

Základní úloha MPC s omezeními je ve tvaru:

$$\min_{U, Z} \sum_{k=0}^{T_p-1} \frac{1}{2} \left( y^T(k) \mathbf{Q} y(k) + u^T(k) \mathbf{R} u(k) + (y(k) - z(k))^T \mathbf{E} (y(k) - z(k)) \right), \quad (20a)$$

$$\text{s.t. } x(k+1) = \mathbf{M}x(k) + \mathbf{N}u(k), k = 0, 1, \dots, T_p - 1, \quad (20b)$$

$$y(k) = \mathbf{C}x(k) + \mathbf{D}u(k), k = 0, 1, \dots, T_p - 1, \quad (20c)$$

$$u_{min} \leq u(k) \leq u_{max}, k = 0, 1, \dots, T_p - 1, \quad (20d)$$

$$y_{min} \leq z(k) \leq y_{max}, k = 0, 1, \dots, T_p - 1. \quad (20e)$$

### 3.1.5 Sledování reference

Minimalizační kritéria (16a) a (20a) dosáhnout minima v okamžiku, kdy  $x(k)$  a  $u(k)$  budou nulové pro všechny  $k = 0, 1, \dots, T_p - 1$ . To by nám vyhovovalo v případě, že bychom chtěli sledovat jen nulovou referenci. Obvykle však chceme sledovat nějakou nenulovou referenci. Toho dosáhneme tím, že požadovanou referenci  $r(k)$  odečteme od

výstupu. Tedy dostaneme kritérium:

$$\min_{U, Z} \sum_{k=0}^{T_p-1} \frac{1}{2} \left( (y(k) - r(k))^T \mathbf{Q} (y(k) - r(k)) + u^T(k) \mathbf{R} u(k) + (y(k) - z(k))^T \mathbf{\Omega} (y(k) - z(k)) \right), \quad (21a)$$

$$\text{s.t. } x(k+1) = \mathbf{M}x(k) + \mathbf{N}u(k), k = 0, 1, \dots, T_p - 1, \quad (21b)$$

$$y(k) = \mathbf{C}x(k) + \mathbf{D}u(k), k = 0, 1, \dots, T_p - 1, \quad (21c)$$

$$u_{\min} \leq u(k) \leq u_{\max}, k = 0, 1, \dots, T_p - 1, \quad (21d)$$

$$y_{\min} \leq z(k) \leq y_{\max}, k = 0, 1, \dots, T_p - 1. \quad (21e)$$

V tomto případě bude člen s  $\mathbf{Q}$  roven nule v okamžiku, kdy výstup, bude odpovídat zadané referenci.

### 3.1.6 Integrální akce

Kritérium v (21a) sleduje zadanou referenci, neumožňuje však pro všechny systémy dosáhnout nulové ustálené odchylky [18]. Toho dosáhneme tím, že budeme místo absolutní hodnoty řízení vážit v členu s  $\mathbf{R}$  změnu řízení. Tedy změníme člen pro řízení na:

$$J_R = \sum_{k=0}^{T_p-1} \frac{1}{2} (u(k) - u(k-1))^T \mathbf{R} (u(k) - u(k-1)). \quad (22)$$

S tím, že člen  $u(-1)$  je řízení, které bylo aplikováno v předchozím kroku. Celkově se tedy dá úloha zapsat jako:

$$\min_{U, Z} \sum_{k=0}^{T_p-1} \frac{1}{2} \left( y^T(k) \mathbf{Q} y(k) + (u(k) - u(k-1))^T \mathbf{R} (u(k) - u(k-1)) + (y(k) - z(k))^T \mathbf{\Omega} (y(k) - z(k)) \right), \quad (23a)$$

$$\text{s.t. } x(k+1) = \mathbf{M}x(k) + \mathbf{N}u(k), k = 0, 1, \dots, T_p - 1, \quad (23b)$$

$$y(k) = \mathbf{C}x(k) + \mathbf{D}u(k), k = 0, 1, \dots, T_p - 1, \quad (23c)$$

$$u_{\min} \leq u(k) \leq u_{\max}, k = 0, 1, \dots, T_p - 1, \quad (23d)$$

$$y_{\min} \leq z(k) \leq y_{\max}, k = 0, 1, \dots, T_p - 1. \quad (23e)$$

## 3.2 Kvadratické programování a lineární MPC

Všechny úlohy MPC, které jsou uvedeny výše se dají přepsat do úlohy kvadratického programování. Úloha kvadratického programování je následující [15]:

$$\min_U \frac{1}{2} \Theta^T \mathbf{H} \Theta + c^T(t) \mathbf{F} \Theta \quad (24a)$$

$$\text{s.t. } \mathbf{G} \Theta \leq W + \mathbf{E}c(t). \quad (24b)$$

Vektor  $\Theta$  je vektor řešení, které hledáme optimalizací. Jedná se o sloupcový vektor o velikosti  $n \in \mathbb{N}$ . Matice  $\mathbf{H}$  je symetrická pozitivně semi/definitní matice velikosti  $n \times n$ . Vektor  $c$  je sloupcový vektor o délce  $m \in \mathbb{N}$ . Matice  $\mathbf{F}$  je velikosti  $m \times n$ .

V podmínce je matice  $\mathbf{G}$  velikosti  $o \times n$ ;  $o \in \mathbb{N}$ . Vektor  $W$  má velikost  $o$  a matice  $\mathbf{E}$  je velikosti  $o \times m$ . Operace menší nebo rovno je v tomto případě myšlena tak, že platí pro odpovídající si prvky ve vektorech.



### 3.2.1 Vektor optimalizace

Vektor  $\Theta$  v rovnici (24a) je ve tvaru:

$$\Theta = \begin{pmatrix} U \\ Z \end{pmatrix}. \quad (25)$$

První prvek vektoru  $\Theta$  je posloupnost řízení  $u_k$ . Druhá složka je  $z$ , která představuje porušení měkkých omezení.

### 3.2.2 Maticový zápis kriteriální funkce

Pokud upravíme matice  $\mathbf{Q}$  a  $\mathbf{R}$  do tvaru:

$$\mathbf{Q} = \text{diag} \begin{pmatrix} \mathbf{Q} & \mathbf{Q} & \dots & \mathbf{Q} \end{pmatrix}, \quad (26)$$

$$\mathbf{R} = \text{diag} \begin{pmatrix} \mathbf{R} & \mathbf{R} & \dots & \mathbf{R} \end{pmatrix}, \quad (27)$$

$$\mathbf{\Omega} = \text{diag} \begin{pmatrix} \mathbf{\Omega} & \mathbf{\Omega} & \dots & \mathbf{\Omega} \end{pmatrix}. \quad (28)$$

Zavedeme pomocné matice pro zapsání  $\Delta u_k(k) = u(k) - u(k-1)$ , pro  $k = 0, 1, \dots, T_p - 1$ :

$$\Delta u_k = \mathbb{D}_i u_k - \tilde{u}_k, \quad (29)$$

$$\mathbb{D}_i = \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix}, \quad \tilde{u}_k = \begin{pmatrix} u(k-1) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (30)$$

Můžeme přepsat kritérium (23a) jako:

$$J = \frac{1}{2} \left( (y_k - r_k)^T \mathbf{Q} (y_k - r_k) + \Delta u_k^T \mathbf{R} \Delta u_k + (y_k - z_k)^T \mathbf{E} (y_k - z_k) \right), \quad (31)$$

kde  $r_k$  je posloupnost reference definovaná jako:

$$r_k = \begin{pmatrix} r(0)^T & r(1)^T & \dots & r(T_p - 1)^T \end{pmatrix}^T. \quad (32)$$

### 3.2.3 Predikce vývoje systému

Musíme vypočítat vývoj systému, základní vzorce jsme vzali z [11].

Vývoj systému definovaného rovnicemi (14a) a (35), můžeme rozdělit na odezvu danou stavem a vstupem systému. Odezva na stav je dána:

$$\begin{pmatrix} y_x(k) \\ y_x(k+1) \\ y_x(k+2) \\ \vdots \\ y_x(k+T_p-1) \end{pmatrix} = \begin{pmatrix} \mathbf{C} \\ \mathbf{CM} \\ \mathbf{CM}^2 \\ \vdots \\ \mathbf{CM}^{(T_p-1)} \end{pmatrix} x(k) = \mathbb{V}x(k) \quad (33)$$

Odezva na vstup je popsána:

$$\begin{pmatrix} y_u(k) \\ y_u(k+1) \\ y_u(k+2) \\ \vdots \\ y_u(k+T_p-1) \end{pmatrix} = \begin{pmatrix} \mathbf{D} & & & & \\ \mathbf{CN} & \mathbf{D} & & & \\ \mathbf{CMN} & \mathbf{CN} & \mathbf{D} & & \\ \vdots & & & \ddots & \\ \mathbf{CM}^{T_p-2}\mathbf{N} & & & \mathbf{CN} & \mathbf{D} \end{pmatrix} \begin{pmatrix} u(k) \\ u(k+1) \\ u(k+2) \\ \vdots \\ u(k+T_p-1) \end{pmatrix} = \mathbb{S}u_k. \quad (34)$$

Odezva je tedy dána

$$y_k = \begin{pmatrix} y(k) \\ y(k+1) \\ y(k+2) \\ \vdots \\ y(k+T_p-1) \end{pmatrix} = \begin{pmatrix} y_x(k) \\ y_x(k+1) \\ y_x(k+2) \\ \vdots \\ y_x(k+T_p-1) \end{pmatrix} + \begin{pmatrix} y_u(k) \\ y_u(k+1) \\ y_u(k+2) \\ \vdots \\ y_u(k+T_p-1) \end{pmatrix} = \mathbb{V}x(k) + \mathbb{S}u_k. \quad (35)$$

### 3.2.4 Vážení sledování reference

Dosažením rovnice (35) do prvního členu (31) dostaneme:

$$\frac{1}{2}(\mathbb{V}x(k) + \mathbb{S}u_k - r_k)^T \mathbb{Q}(\mathbb{V}x(k) + \mathbb{S}u_k - r_k). \quad (36)$$

Roznásobením dostaneme součet součinnů, kde v součinech budou zleva nebo zprava násobeny  $u_k$ . Dostaneme kvadratický člen pro  $u_k$  ve tvaru:

$$\frac{1}{2}u_k \mathbb{S}^T \mathbb{Q} \mathbb{S} u_k \quad (37)$$

a lineární ve tvaru:

$$\left( x^T(k) \mathbb{V}^T \mathbb{Q} \mathbb{S} - r_k^T \mathbb{Q} \mathbb{S} \right) u_k. \quad (38)$$

Kvadratický člen bude součástí matice  $\mathbf{H}$  v rovnici (24a). Matice bude vypadat:

$$\mathbf{H} = \begin{pmatrix} \mathbb{S}^T \mathbb{Q} \mathbb{S} & \times \\ \times & \times \end{pmatrix}, \quad (39)$$

kde členy  $s \times$  označují pozice, kde budou členy závislé i na  $z$ , které jsme zatím neurčili. Pokud si rozepíšeme člen s  $\mathbf{H}$  v rovnici (24a), pomocí násobení matic, tak výsledný člen bude souhlasit s (37). Pro lineární člen dostaneme podobný výsledek:

$$\mathbf{F} = \begin{pmatrix} \mathbb{V}^T \mathbb{Q} \mathbb{S} & \times \\ \times & \times \\ -\mathbb{Q} \mathbb{S} & \times \end{pmatrix}, \quad (40)$$

kde  $\times$  označuje členy závislé na  $z$  nebo neurčené.

Také musíme určit vektor  $c(t)$ :

$$c(t) = \begin{pmatrix} x(t) \\ \tilde{u}_k \\ r_k \end{pmatrix}. \quad (41)$$

Pořadí jednotlivých částí vektoru  $c$  určuje pořadí řádek v matici  $\mathbf{F}$ .

### 3.2.5 Vážení výstupů

V druhém členu kriteriální funkce (31) vážíme rozdíl vstupů, s použitím rovnic (29) a (30) dostáváme:

$$\frac{1}{2} \left( u_k^T \mathbb{D}_i^T - \tilde{u}_k^T \right) \mathbb{R} \left( \mathbb{D}_i u_k - \tilde{u}_k \right). \quad (42)$$

Roznásobením dostaneme kvadratický člen pro  $u_k$ :

$$\frac{1}{2} u_k \mathbb{D}_i^T \mathbb{R} \mathbb{D}_i u_k \quad (43)$$

a lineární člen pro  $u_k$ :

$$-\tilde{u}\mathbb{R}\mathbb{D}_i u_k. \quad (44)$$

Takže můžeme přidat další členy do matice  $\mathbf{H}$ :

$$\mathbf{H} = \begin{pmatrix} \mathbb{S}^T \mathbb{Q} \mathbb{S} + \mathbb{D}_i^T \mathbb{R} \mathbb{D}_i & \times \\ \times & \times \end{pmatrix} \quad (45)$$

a matici  $\mathbf{F}$ :

$$\mathbf{F} = \begin{pmatrix} \mathbb{V}^T \mathbb{Q} \mathbb{S} & \times \\ -\mathbb{R} \mathbb{D}_i & \times \\ -\mathbb{Q} \mathbb{S} & \times \end{pmatrix}. \quad (46)$$

### 3.2.6 Vážení splnění měkkých omezení

V posledním členu (31) vážíme měkká omezení, pokud tento člen nedokážeme vynulovat, tak jsou měkká omezení aktivní.

Z rovnice:

$$\frac{1}{2} (y_k^T - z_k^T) \Omega (y_k - z_k). \quad (47)$$

Dostaneme kvadratický člen pro  $u_k$ :

$$u_k^T \mathbb{S}^T \Omega \mathbb{S} u_k, \quad (48)$$

kvadratický člen pro  $z_k$ :

$$z_k^T \Omega z_k, \quad (49)$$

křížový člen (existuje i v transponované verzi):

$$-z_k^T \Omega \mathbb{S} u_k. \quad (50)$$

Lineární členy pro  $u_k$ :

$$x^T \mathbb{V}^T \Omega \mathbb{S} u_k, \quad (51)$$

a pro  $z_k$ :

$$-x^T \mathbb{V}^T \Omega z_k. \quad (52)$$

Výsledná matice  $\mathbf{H}$  je tedy ve tvaru:

$$\mathbf{H} = \begin{pmatrix} \mathbb{S}^T \mathbb{Q} \mathbb{S} + \mathbb{D}_i^T \mathbb{R} \mathbb{D}_i + \mathbb{S}^T \Omega^T & -\mathbb{S}^T \Omega \\ -\mathbb{S} \Omega & \Omega \end{pmatrix} \quad (53)$$

Pro matici  $\mathbf{F}$  platí:

$$\mathbf{F} = \begin{pmatrix} \mathbb{V}^T \mathbb{Q} \mathbb{S} + \mathbb{V}^T \Omega \mathbb{S} & \mathbb{V}^T \Omega \\ -\mathbb{R} \mathbb{D}_i & 0 \\ -\mathbb{Q} \mathbb{S} & 0 \end{pmatrix}. \quad (54)$$

### 3.2.7 Tvrdá a měkká omezení v QP řešiteli

V rovnici (24b) máme uvedeno v jakém tvaru se zadávají omezení do použitého řešitele. Omezení se do tohoto tvaru musejí přepočítat. Omezení v rovnicích (23d) a (23e) jsou čtyři jednoduché nerovnice, které musejí být splněny. Nerovnice

$$u(k) \leq u_{max}, \quad (55)$$

$$z(k) \leq y_{max}, \quad (56)$$

se dají do tvaru (24b) přepsat přímo. Rovnice

$$u_{min} \leq u(k), \quad (57)$$

$$y_{min} \leq z(k), \quad (58)$$

se musí vynásobit  $-1$ . Tím se rovnice změní na tvar:

$$-u(k) \leq -u_{min}, \quad (59)$$

$$-z(k) \leq -y_{min}. \quad (60)$$

Nyní musíme tyto rovnice dosadit do (24b) a dostaneme

$$\begin{pmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} u_k \\ z \end{pmatrix} \leq \begin{pmatrix} -u_{min} \\ u_{max} \\ -y_{min} \\ y_{max} \end{pmatrix} + \mathbf{0}c(t). \quad (61)$$

Z této rovnice vyplývá:

$$\mathbf{G} = \begin{pmatrix} -\mathbf{I} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}, \quad (62)$$

$$\mathbf{W} = \begin{pmatrix} -u_{min} \\ u_{max} \\ -y_{min} \\ y_{max} \end{pmatrix}, \quad (63)$$

$$\mathbf{E} = 0. \quad (64)$$

Tedy pro náš případ je matice  $\mathbf{E}$  nulová.

### 3.3 Rozšíření MPC na linearizaci podél trajektorie

Standardní MPC používá model získaný linearizací v ekvilibriu. Tato metoda linearizace předpokládá, že existuje stav  $x_0$  a vstup  $u_0$  takový:

$$0 = f(x_0, u_0). \quad (65)$$

Linearizaci v ekvilibriu nelze pro model našeho auta použít. Místo toho používáme linearizaci podél trajektorie. To vede na použití časově proměnného modelu.

Pokud máme nelineární systém, dobu predikce  $T_p$  kroků a známe trajektorii vstupu  $u$  po dobu predikce, můžeme vypočítat vývoj stavů po dobu predikce. V každém okamžiku vzorkování můžeme díky znalosti stavu a vstupu vypočítat linearizovaný model. Musíme tedy provést  $T_p$  linearizací. V tomto linearizovaném diferenciálním modelu hledáme minimum změnou difference vstupní trajektorie. Protože máme obecně nelineární systém, pro novou vstupní trajektorii. Provedeme linearizaci a výpočet. Optimální řešení jsme našli, pokud je změna trajektorie vstupu malá.

### 3.3.1 Aproximace pomocí LVT

Mějme diskrétní nelineární systém ve tvaru rovnice (2a):

$$x(k+1) = f(x(k), u(k)). \quad (66)$$

Dále předpokládejme, že do vstup do systému je konstantní, tedy platí:

$$u(k) = u_0. \quad (67)$$

Poté lze vývoj systému popsat rovnicemi:

$$\hat{x}(k+1) = f(\hat{x}(k), u(k)), \quad (68a)$$

$$\hat{x}(0) = x_0, \quad (68b)$$

$$u(k) = u_0. \quad (68c)$$

Poté můžeme vývoj systému (68) aproximovat LTV modelem ve tvaru:

$$\delta x(k+1) = \mathbf{A}_{k,0} \delta x(k) + \mathbf{B}_{k,0} \delta u(k), \quad (69a)$$

$$\text{s.t. } \mathbf{A}_{k,0} = \left. \frac{\partial f}{\partial x} \right|_{x_0(k), u_0}, \quad \mathbf{B}_{k,0} = \left. \frac{\partial f}{\partial u} \right|_{x_0(k), u_0}, \quad (69b)$$

$$\delta x(k) = x(k) - \hat{x}(k), \quad \delta u(k) = u(k) - u_0. \quad (69c)$$

V rovnicích (69) používáme diferenciální model od trajektorie, lze přejít na systém v absolutních souřadnicích [6], který je ve tvaru:

$$x(k+1) = \mathbf{A}_{k,0}x(k) + \mathbf{B}_{k,0}u(k) + d_{k,0}(k), \quad (70)$$

kde  $d_{k,0}(k) = \hat{x}(k+1) - \mathbf{A}_{k,0}\hat{x}(k) - \mathbf{B}_{k,0}u_0$  pro  $k \geq 0$ .

V této rozšířené úloze můžu definovat vývoj systému jako:

$$y_k = \mathbb{V}x(k) + \mathbb{S}u_k + \mathbb{D}d. \quad (71)$$

Matice  $\mathbb{V}$  a  $\mathbb{S}$  jsou matice z rovnic (33) a (34). Vektor stavu  $x(k)$  známe z měření stavu a  $u_k$  je trajektorie vstupu. Přidala se nám tam matice  $\mathbb{D}$  a vektor  $d$ . Člen  $\mathbb{D}d$  je konstantní posunutí trajektorie. Tedy v tomto případě se v podstatě o afinní regulátor.

Nejprve definujeme vektor  $d$ , tento vektor obsahuje trajektorii vývoje nelineárního systému a zároveň jeho vstup. Tedy:

$$d = \begin{pmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(T_p - 1) \\ u(0) \end{pmatrix}. \quad (72)$$

Pro vstup využíváme toho, že vstup systému je konstantní a tedy ji lze redukovat jen na případ  $u(0)$ . Z konstrukce vektoru  $d$  vyplývá, že jeho velikost je  $T_p n_x + n_u$ .

Matice  $\mathbb{D}$  je ve tvaru:

$$\mathbb{D} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ -\mathbf{CA} & \mathbf{C} & 0 & \dots & 0 & -\mathbf{CB} - \mathbf{D} \\ -\mathbf{CA}^2 & 0 & \mathbf{C} & \dots & 0 & -\mathbf{CAB} - \mathbf{CB} - \mathbf{D} \\ \vdots & & & \ddots & \vdots & \vdots \\ -\mathbf{CA}^{T_p-1} & 0 & 0 & \dots & \mathbf{C} & -\mathbf{CA}^{T_p-2}\mathbf{B} - \mathbf{CA}^{T_p-3}\mathbf{B} - \dots - \mathbf{B} - \mathbf{D} \end{pmatrix}. \quad (73)$$

Odvození matice  $\mathbb{D}$  provedeme vývoj systému pomocí rovnice (70) za předpokladu  $x(0) = \hat{x}(0)$ , protože počáteční stav je změřen a je pro lineární i nelineární systém stejný dostaneme:

$$x(1) = \mathbf{A}x(0) + \mathbf{B}u(0) + \hat{x}(1) - \mathbf{A}\hat{x}(0) - \mathbf{B}u_0, \quad (74a)$$

$$x(2) = \mathbf{A}x(1) + \mathbf{B}u(1) + \hat{x}(2) - \mathbf{A}\hat{x}(1) - \mathbf{B}u_0, \quad (74b)$$

$$x(2) = \mathbf{A}^2x(0) + \mathbf{A}\mathbf{B}u(0) + \mathbf{A}\hat{x}(1) - \mathbf{A}^2\hat{x}(0) - \mathbf{A}\mathbf{B}u_0 + \mathbf{B}u(1) + \hat{x}(2) - \mathbf{A}\hat{x}(1) - \mathbf{B}u_0, \quad (74c)$$

Zde sice uvažujeme stav  $a$  v rovnici (71) uvažujeme výstup, ale můžeme rovnici (74c) dosadit do rovnice (35) a tím dostat výstup. Pro náš případ je také matice  $\mathbf{C}$  jednotková a matice  $\mathbf{D}$  nulová, takže rovnici (74c) neovlivní.

Dále můžeme z rovnice (74c) vyextrahovat členy, které nedostaneme z matic  $\mathbb{V}$  a  $\mathbb{S}$ , tak nám zbude:

$$\mathbf{A}\hat{x}(1) - \mathbf{A}^2\hat{x}(0) - \mathbf{A}\mathbf{B}u_0 + \hat{x}(2) - \mathbf{A}\hat{x}(1) - \mathbf{B}u_0. \quad (75)$$

Člen s  $\hat{x}(1)$  se odečte, u členu s  $\hat{x}(0)$  je konstanta  $-\mathbf{A}^2$ , u členu  $\hat{x}(2)$  je jednotková matice a pro konstantní vstup  $u_0$  dostáváme člen  $-\mathbf{A}\mathbf{B} - \mathbf{B}$ . Z rovnice (70) vidíme, že vždy se vynásobí člen s  $\hat{x}(0)$  maticí  $\mathbf{A}$ , členy s  $\hat{x}(n)$ ;  $n = 1, 2, \dots, T_p - 1$  budou jen jednotkové, protože se v dalších krocích odečtou. U konstantního vstupu  $u_0$  se bude odečítat celý vývoj systému vynucený vstupem.

### 3.3.2 Kvadratické programování a LVT MPC

Jediná změna oproti lineárnímu MPC je přidání konstantního členu do vzorce pro vývoj stavu systému. Tedy vývoj systému je popsán:

$$y = \mathbb{V}x + \mathbb{S}u + \mathbb{D}d. \quad (76)$$

Odvození kvadratických a lineárních členů je stejné, jen se přidá člen s  $\mathbb{D}d$ . To změní vektor  $c$  na:

$$c = \begin{pmatrix} x \\ d \\ \tilde{u}_k \\ r_k \end{pmatrix}. \quad (77)$$

A matici  $\mathbf{F}$  změní na:

$$\mathbf{F} = \begin{pmatrix} \mathbb{V}^T\mathbb{Q}\mathbb{S} + \mathbb{V}^T\mathbb{N}\mathbb{S} & \mathbb{V}^T\mathbb{N} \\ \mathbb{D}^T\mathbb{Q}\mathbb{S} + \mathbb{D}^T\mathbb{N}\mathbb{S} & -\mathbb{D}^T\mathbb{N} \\ -\mathbb{R}\mathbb{D}_i & 0 \\ -\mathbb{Q}\mathbb{S} & 0 \end{pmatrix} \quad (78)$$

## 3.4 Nástroje pro redukci velikosti problému

Ve formulaci (23) předpokládáme, že měníme řízení v každém kroku, také aplikujeme měkká omezení na každý krok výstupu po celou dobu predikce. Tímto přístupem narůstá velikost problému, jestliže máme  $n_y$  výstupů a  $n_u$  vstupů a dobu predikce  $T_p$ , tak celkově dostáváme  $n_y T_p + n_u T_p$  proměnných, které optimalizujeme. Čím více máme proměnných, které optimalizujeme, tím déle trvá nalezení řešení. Proto je výhodné omezit jejich počet.

Toho dosáhneme dvěma způsoby, při prvním způsobu aplikujeme měkká omezení výstupu jen na určité výstupy v určitých predikčních krocích. Druhou možností je snížit počet kroků, ve kterém se může měnit vstup.

### 3.4.1 Redukovaná měkká omezení

Redukci měkkých omezení můžeme aplikovat dvěma způsoby. Při prvním sledujeme splnění omezení jen na určitých proměnných. Pokud máme  $n_y$  výstupních proměnných a  $n$ -tou proměnou označíme  $y_n(k)$  v čase  $k$ . Pokud budeme chtít množinu proměnných, tak jí označíme jako  $(a, b, c)$ . V tomto případě tedy vybíráme  $a$ -tou,  $b$ -tou a  $c$ -tou proměnnou. Takto můžeme přeformulovat podmínky (23e) jako:

$$y_{(a,b,c,\dots)min} \leq z_{(a,b,c,\dots)}(k) \leq y_{(a,b,c,\dots)max}, k = 0, 1, 2, \dots, T_p - 1. \quad (79)$$

Pokud vybereme  $n \leq n_y$  výstupů, dostaneme velikost problému  $nT_p \leq n_yT_p$ .

Druhou možností je sledovat překročení omezení jen v určitý čas, tedy  $k$  nebude od 0 do  $T_p - 1$  ale jen nějaká podmnožina kterou označíme  $\Omega$ , tedy zapsáno v kombinaci s (79)

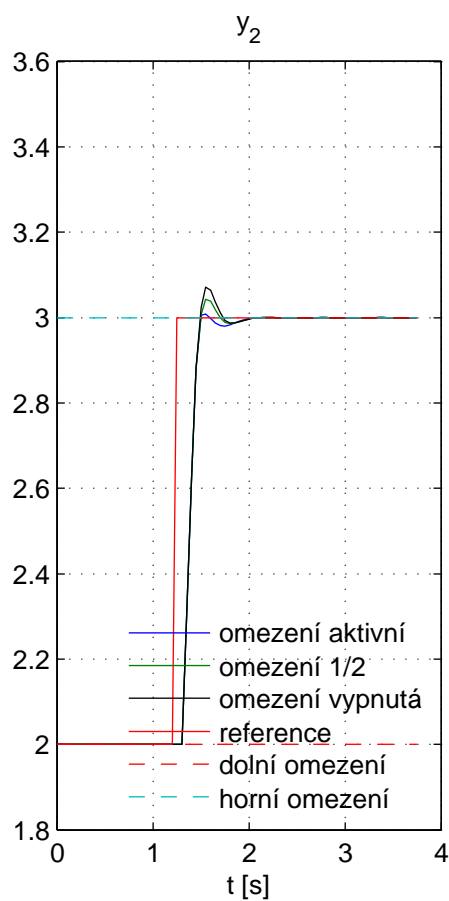
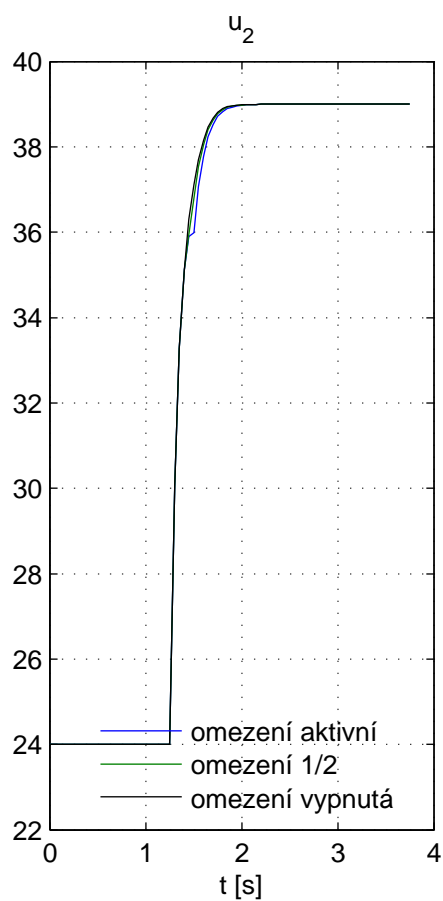
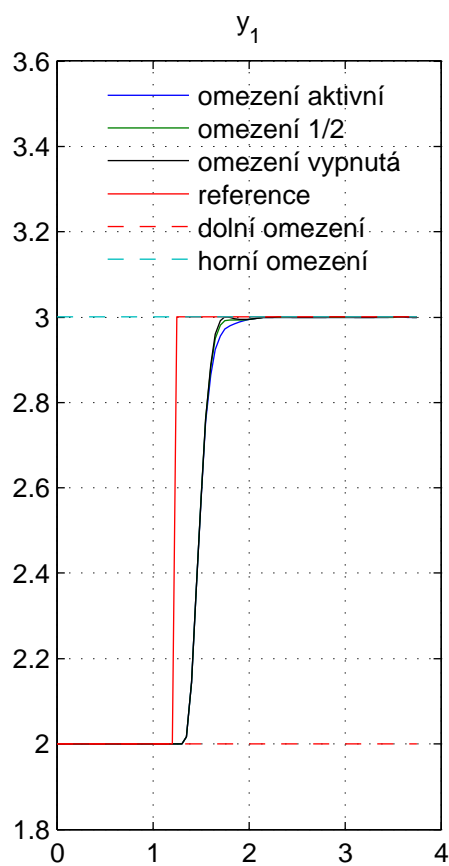
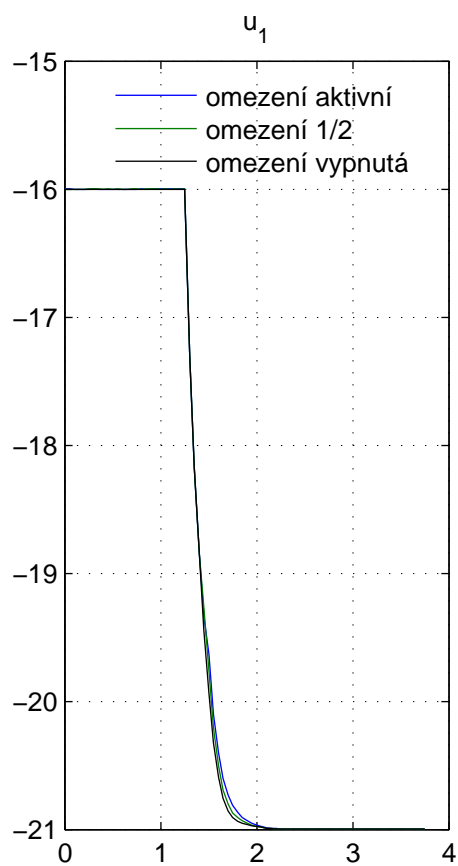
$$y_{(a,b,c,\dots)min} \leq z_{(a,b,c,\dots)}(k) \leq y_{(a,b,c,\dots)max}, k \in \Omega, \Omega = \{i : i \in \mathbb{N}\}. \quad (80)$$

Vliv měkkých omezení je vidět na následujícím Obrázku 1, simulujeme nelineární systém se dvěma vstupy a výstupy s rovnicemi:

$$\dot{x}_1 = -x_1^2 + 10x_2 + u_1, \quad (81a)$$

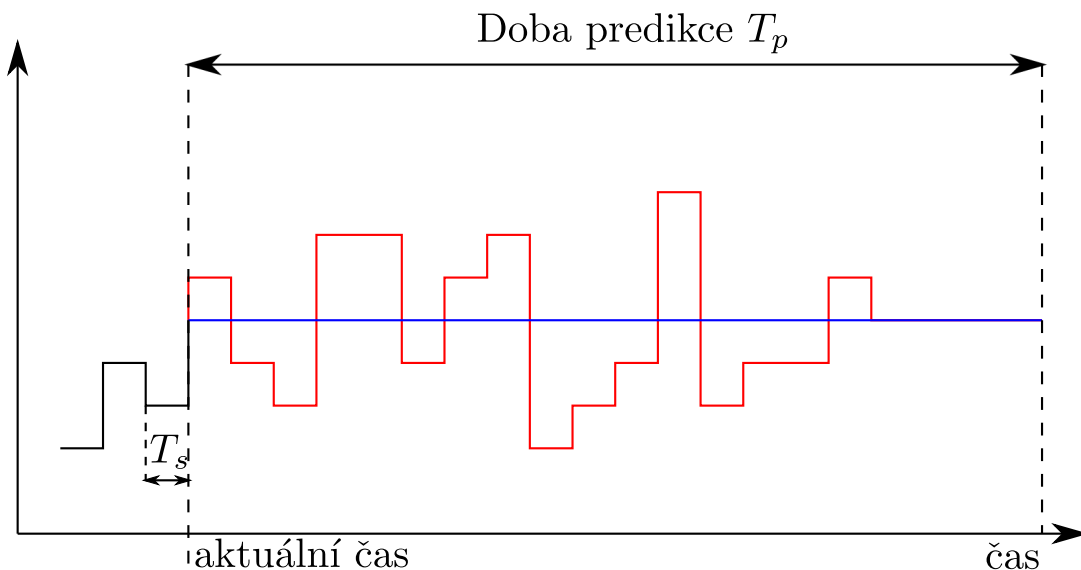
$$\dot{x}_2 = -x_2^2 - 10x_1 + u_2. \quad (81b)$$

Oba stavy jsou zároveň výstupy, takže matice  $\mathbf{C}$  je jednotková. Oba výstupy mají sledovat stejnou referenci. Na Obrázku 1 jsou v levém zobrazeny vstupy a v pravém výstupy systému. Simulaci provádíme ve třech nastaveních měkkých omezení. V prvním případě jsou omezení aktivní v každém bodě a dochází k jejich splnění, velikost problému je  $n_u$  za nalezení vhodného řízení (blokujeme vstupy tento postup je vysvětlen v následující sekci) a  $T_p n_y$  pro měkká omezení, pro nás konkrétně je  $T_p = 10$  tedy celková velikost problému je 22. V druhém případě aplikujeme omezení jen v každém druhém vzorku. Dochází k nesplnění omezení, ale velikost problému je jen 12 oproti uvažování všech omezení. V posledním případě jsou omezení vypnuta, tedy dochází k jejich překročení (vůbec je neuvažujeme), ale tento problém má velikost jen 2.



**Obrázek 1** Odezva systému se zapnutými měkkými omezeními





Obrázek 2 Různé možnosti blokování.

### 3.4.2 Blokování vstupů

Druhou možností je takzvané blokování vstupů, ve formulaci (23a) předpokládáme, že můžeme měnit vstup  $u(k)$  pro všechna  $k = 0, 1, 2, \dots, T_p - 1$ . Pro snížení velikosti problémů můžeme některé kroky sloučit, tedy že pro nějaká  $k_s = a, a + 1, \dots, b; 0 \leq a \leq b \leq T_p - 1$ , bude platit  $u(k_s) = \text{konstanta}$ .

Na Obrázku 2 je vidět případ dvou různých blokování, v červeném případě můžeme měnit vstup v každém kroku (tedy máme  $T_p n_u$  stupňů volnosti), v modrém případě máme všechny kroky sloučené do jednoho (takže máme jen  $n_u$  stupňů volnosti).

Blokování vstupů provedeme úpravou matice  $S$ , která popisuje odezvu systému na vstup:

$$\mathbf{S}' = \mathbf{S}\mathbf{T}. \quad (82)$$

Matice  $\mathbf{S}'$  je redukováná matice a matice  $\mathbf{T}$  je matice provádějící redukci. Její rozměry jsou  $T_p n_x \times T_k n_u$ . Mějme jednotkovou matici  $E$  o rozměru  $n_u \times n_u$ . Pomocí ní je bloková matice  $\mathbf{T}$  definována jako:

$$\mathbf{T} = \begin{pmatrix} E & & & & \\ & E & & & \\ & & E & & \\ & & & \ddots & \\ & & & & E \end{pmatrix}. \quad (83)$$

Počet sloupců blokové matice  $\mathbf{T}$  odpovídá době korekce  $T_c$ . Počet řádků odpovídá době predikce  $T_p$ . Pokud chceme sloučit dva po sobě následující kroky řízení do jednoho, umístíme na příslušné řádky do stejného sloupce jednotkovou matici  $E$ . Tedy jestliže chceme aby kroky 3 a 4 byli sloučené, tak na třetím a čtvrtém řádku bude matice  $E$  ve třetím sloupci. Za předpokladu, že kroky 1 a 2 nejsou sloučené. Obsazování sloupců musí postupně narůstat. Tedy jestliže v  $i$ -tém řádku je jednotková matice ve sloupci  $j$ , tak musí platit pro řádky  $n > i$  to, že matice  $E$  je ve sloupci  $m \geq j$ .

Na Obrázku 3 je zobrazen systém popsán rovnicemi (81). V prvním případě máme pro každý krok řízení jeden stupeň volnosti, což vede na velmi agresivní a dynamický průběh regulace. Ve druhém případě používáme pouze 5 bloků, tedy každé 2 kroky jsou

spojené dohromady. To vede na méně agresivní chování regulátoru, což vede k lepšímu sledování reference než v případě systému s 10 bloky. Je to způsobeno tím, že při výpočtu predikce se dostaneme mimo okolí bodu, kde platí linearizace systému v bodě daném vstupem  $u$  a stavem  $x$ . Pokud použijeme konzervativnější způsob řízení, tedy 5 bloků místo 10 snížíme tím velikost akčního zásahu a tím dojde ke zmenšení změny stavu systému a aproximace pomocí LVT je přesnější. V posledním případě sloučíme všech 10 kroků řízení do 1. To vede k malým regulačním zásahům a regulátor je celkově pomalejší.

### 3.4.3 Normalizace kriteriální funkce

Pokud používáme redukci QP problému pomocí redukce omezení nebo blokování vstupů, dochází k tomu, že se mění počet kroků v kriteriální funkci, přes které počítáme hodnotu kriteriální funkce pro člen se vstupem a měkkými omezeními. To mění hodnotu kriteriální funkce a tedy i hodnotu optima. Pokud je doba predikce 100 kroků a vstup je redukován jen na 10 bloků, bude sledování výstupů 10násobně důležitější, než vážení vstupů. Tomu můžeme předcházet normalizováním jednotlivých členů kriteriální funkce. Poté vypadá definice MPC problému:

$$\begin{aligned} \min_{U,Z} \sum_{k=0}^{T_p-1} \frac{1}{2T_p} \left( y^T(k) \mathbf{Q} y(k) \right) + \frac{1}{2T_k} \sum_{k=0}^{T_k-1} (u(k) - u(k-1))^T \mathbf{R} (u(k) - u(k-1)) + \\ + \frac{1}{2\omega T_p} \sum_{k=0}^{T_p-1} \left( y(k) - z(k) \right)^T \mathbf{\Omega} (y(k) - z(k)) \end{aligned} \quad (84a)$$

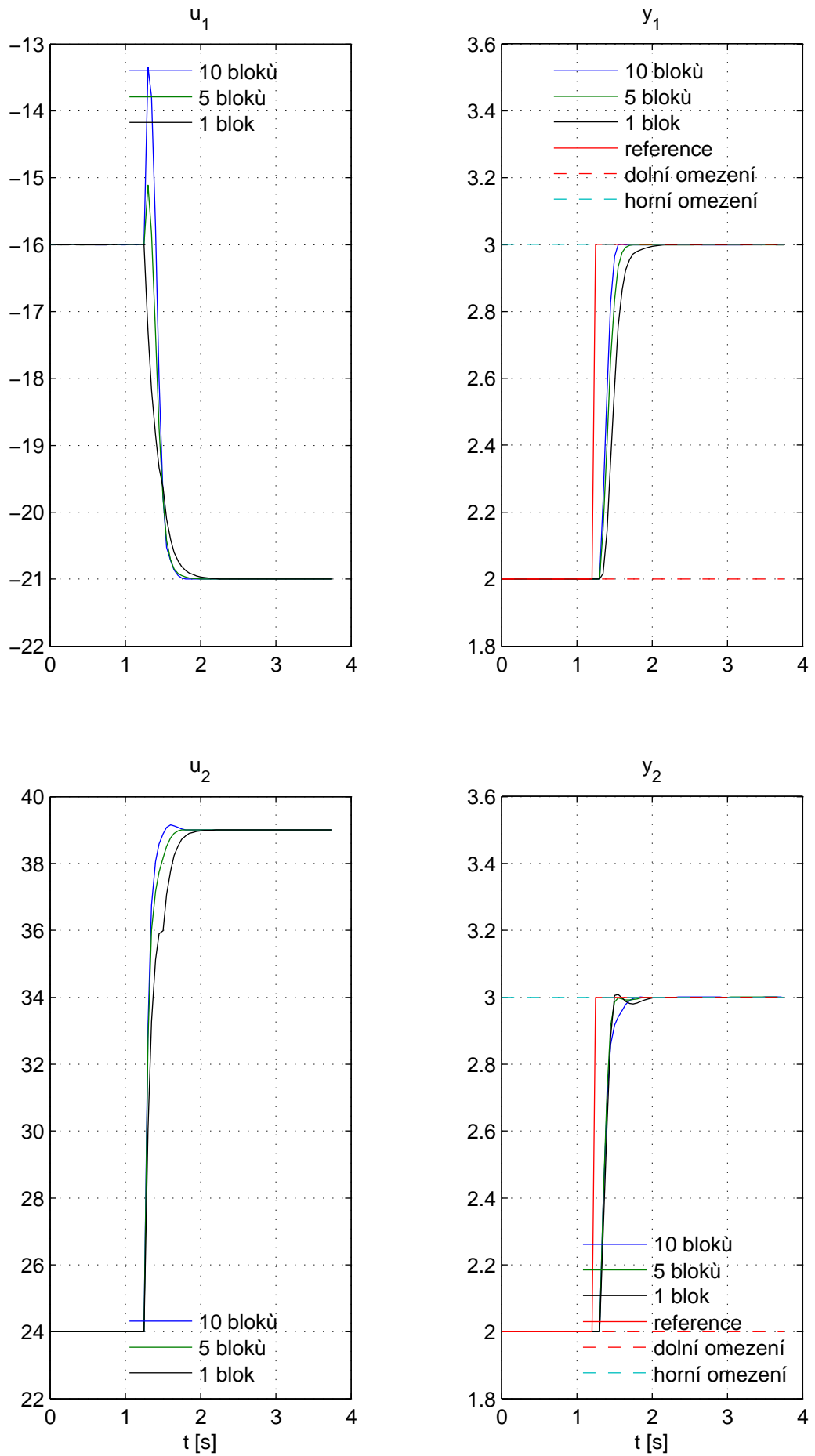
kde  $\omega$  je počet nenulových prvků na diagonále matice  $\mathbf{\Omega}$

$$\text{s.t. } x(k+1) = \mathbf{M}x(k) + \mathbf{N}u(k), k = 0, 1, \dots, T_p - 1, \quad (84b)$$

$$y(k) = \mathbf{C}x(k) + \mathbf{D}u(k), k = 0, 1, \dots, T_p - 1, \quad (84c)$$

$$u_{min} \leq u(k) \leq u_{max}, k = 0, 1, \dots, T_p - 1, \quad (84d)$$

$$y_{min} \leq z(k) \leq y_{max}, k = 0, 1, \dots, T_p - 1. \quad (84e)$$



Obrázek 3 Vliv blokování na chování systému.

## 4 Model automobilu

Základ modelu automobilu je z práce A Linear Time Varying Model Predictive Control Approach to the Integrated Vehicle Dynamics Control Problem in Autonomous System [6]. Základní model počítá při popisu dynamiky automobilu se čtyřmi pneumatikami a konstantním zatížením kol. Jako vstupy má prokluz pneumatik a natočení přední nápravy. Model také umožňuje měnit konstantu tření mezi jednotlivými koly a povrchem. Při implementaci byl náš model mírně upraven, přidali jsme možnost řízení zadní nápravy a neuvažovali jsme disipaci energie, takže jedoucí auto se nikdy nezastaví.

### 4.1 Nelineární model

Model automobilu obsahuje čtyři pneumatiky. Názvy jednotlivých veličin se opakují pro všechny čtyři pneumatiky i když veličiny pro jednotlivé pneumatiky mohou mít rozdílné hodnoty. Rozlišení dosáhneme pomocí indexů, stejně jako v [6]. Mějme například rychlost pravé přední pneumatiky v ose  $x$ , ta bude zapsána jako  $v_{x_{f,r}}$ .  $v$  je rychlost, dolní index  $x$  značí  $x$  osu,  $f$  značí přední nápravu (anglické front) a  $r$  značí pravou stranu (anglické right). Obecně je tedy zápis  $v_{x_{*,\bullet}}$ , kde  $*$  je z množiny  $\{f, r\}$  jako přední (front) a zadní (rear),  $\bullet$  je z množiny  $\{l, r\}$  tedy levá (left) a pravá (right).

Stav modelu je reprezentován jako:

$$\begin{pmatrix} \dot{y} & \dot{x} & \psi & \dot{\psi} & Y & X \end{pmatrix}^T. \quad (85)$$

Proměnné  $\dot{y}$  a  $\dot{x}$  jsou rychlost v  $x$ -ové a  $y$ -ové ose v automobilu. Veličiny  $\psi$  a  $\dot{\psi}$  je natočení a úhlová rychlost vzhledem ke globální  $X$  souřadnici. Souřadnice  $X$  a  $Y$  je globální poloha automobilu vůči počátku souřadnicové soustavy viz Obrázek 4.

Vstup systému je ve tvaru:

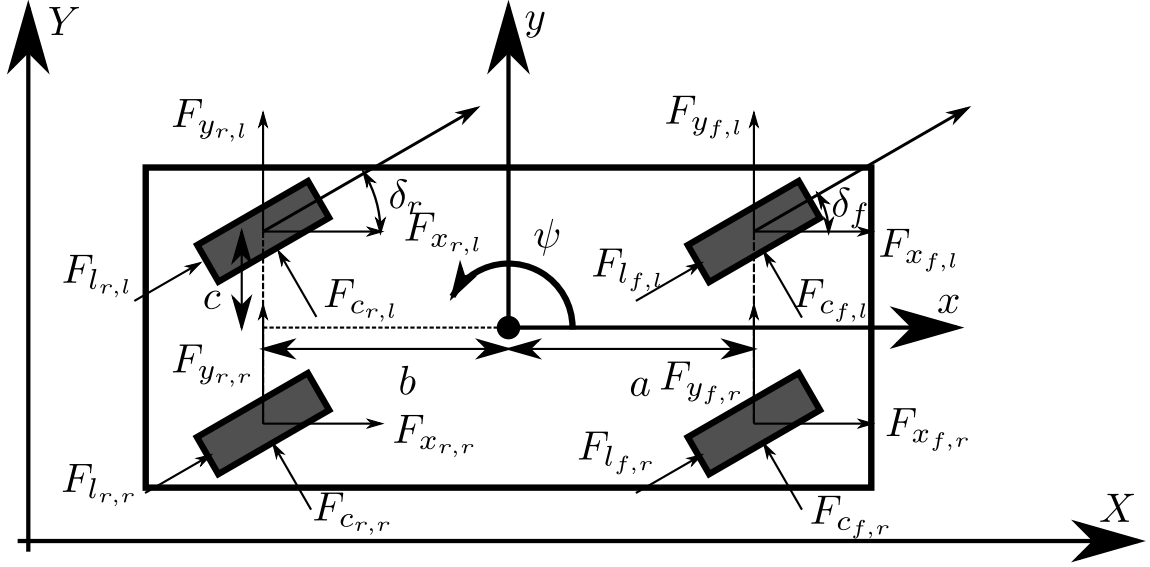
$$\begin{pmatrix} \delta_f & \delta_r & s_{f,l} & s_{f,r} & s_{r,l} & s_{r,r} \end{pmatrix}^T. \quad (86)$$

Proměnné  $\delta_f$  a  $\delta_r$  představují natočení přední a zadní nápravy. Zadní náprava má obvykle nulové natočení, ale z důvodu snadného rozšíření modelu o možnost natáčení i zadní nápravou jsme ji ponechali ve vstupu. Proměnné  $s_{f,l}$ ,  $s_{f,r}$ ,  $s_{r,l}$ ,  $s_{r,r}$  jsou skluzy pneumatik pro jednotlivá kola viz Obrázek 4.

Skluz  $s_{*,\bullet}$  je definován jako:

$$s_{*,\bullet} = \begin{cases} \frac{r_\omega \omega_{*,\bullet}}{v_{l_{*,\bullet}}} - 1 & \text{Pokud } v_{l_{*,\bullet}} > r_\omega \omega_{*,\bullet}, v_{l_{*,\bullet}} \neq 0 \\ & \text{brždění} \\ 1 - \frac{v_{l_{*,\bullet}}}{r_\omega \omega_{*,\bullet}} & \text{Pokud } v_{l_{*,\bullet}} < r_\omega \omega_{*,\bullet}, \omega_{*,\bullet} \neq 0 \\ & \text{rozjezd} \end{cases} \quad (87)$$

Veličina  $r_\omega$  je poloměr kola. Obecně tato veličina nemusí odpovídat poloměru kola, které je v klidu a nezatížené. Pokud dojde k zatížení kola, může se vlivem pružnosti pneumatiky poloměr změnit. My jsme však pro jednoduchost v našem modelu uvažovali



Obrázek 4 Dynamický model automobilu.

poloměr jako konstantní a neměnný. Symbol  $\omega_{*,\bullet}$  je úhlová rychlost otáčení pneumatiky. Proměnná  $v_{l*,\bullet}$  vyjadřuje rychlost pneumatiky v podélném směru.

Podíly v rovnici (87) vyjadřují poměr mezi obvodovou rychlostí pneumatiky a rychlostí pneumatiky v daném směru. Skluz je tedy v rozsahu  $\{-1, 1\}$ . Pokud má skluz hodnotu  $-1$  kolo plně brzdí a neotáčí se, pro  $(-1, 0)$  se kolo otáčí, ale brzdí. Pro nulový skluz se kolo volně otáčí a jeho obvodová rychlost odpovídá rychlosti automobilu. V intervalu  $(0, 1)$  kolo zrychluje a pro skluz  $1$  kolo se začíná otáčet z nulové rychlosti.

Samotná dynamika automobilu je popsána rovnicemi:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}}, \quad (88a)$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}}, \quad (88b)$$

$$\dot{\psi} = \dot{\psi}, \quad (88c)$$

$$I\ddot{\psi} = a(F_{y_{f,l}} + F_{y_{f,r}}) - b(F_{y_{r,l}} + F_{y_{r,r}}) + c(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}). \quad (88d)$$

Konstanty  $a, b, c$  popisují rozměry automobilu,  $a$  je vzdálenost přední nápravy od těžiště automobilu,  $b$  je vzdálenost zadní nápravy od těžiště automobilu a  $c$  je vzdálenost pneumatik od osy automobilu (předpokládáme osově symetrický automobil). Konstanta  $m$  je hmotnost automobilu a  $I$  je moment setrvačnosti automobilu.

Výsledná poloha těžiště automobilu vůči počátku souřadnicového systému je dána rovnicemi:

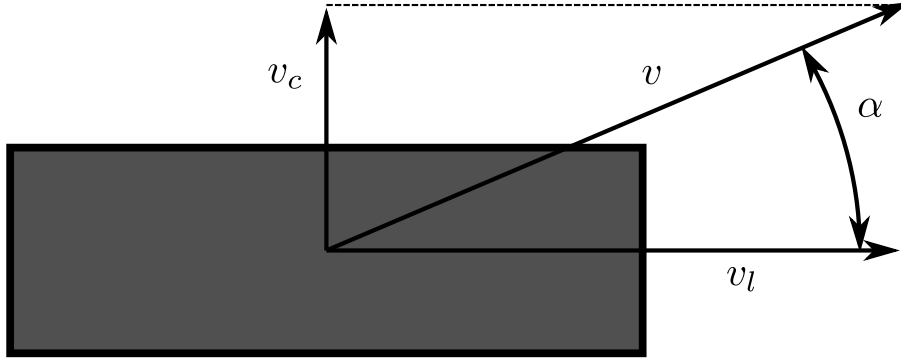
$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi, \quad (89a)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi. \quad (89b)$$

Jednotlivé síly, které působí na pneumatiky v osách automobilu  $F_{x_{*,\bullet}}$  a  $F_{y_{*,\bullet}}$  spočteme z příčné (cornering)  $F_{c_{*,\bullet}}$  a podélné (longitudinal)  $F_{l_{*,\bullet}}$  síly působící na pneumatiku podle vzorce:

$$F_{y_{*,\bullet}} = F_{l_{*,\bullet}} \sin \delta_* + F_{c_{*,\bullet}} \cos \delta_*, \quad (90a)$$

$$F_{x_{*,\bullet}} = F_{l_{*,\bullet}} \cos \delta_* - F_{c_{*,\bullet}} \sin \delta_*. \quad (90b)$$



**Obrázek 5** Úhel směrové úchytky pneumatiky.

Příčné a podélné síly pneumatiky jsou definovány v modelu pneumatiky jako:

$$F_{c_{*,\bullet}} = f_c (\alpha_{*,\bullet}, s_{*,\bullet}, F_{z_{*,\bullet}}), \quad (91a)$$

$$F_{l_{*,\bullet}} = f_l (\alpha_{*,\bullet}, s_{*,\bullet}, F_{z_{*,\bullet}}). \quad (91b)$$

Zatížení pneumatik je definováno jako:

$$F_{z_{f,\bullet}} = \frac{bmg}{2(a+b)}, \quad (92a)$$

$$F_{z_{r,\bullet}} = \frac{amg}{2(a+b)}. \quad (92b)$$

Úhel směrové úchytky, který vyjadřuje úhel mezi vektorem aktuální rychlosti kola a jeho podélné složky (viz Obrázek 5), je definován jako:

$$\alpha_{*,\bullet} = \tan^{-1} \frac{v_{c_{*,\bullet}}}{v_{l_{*,\bullet}}}. \quad (93)$$

Rychlost pneumatiky v podélném  $v_{l_{*,\bullet}}$  a příčném  $v_{c_{*,\bullet}}$  jsou definovány jako:

$$v_{c_{*,\bullet}} = v_{y_{*,\bullet}} \cos \delta_* - v_{x_{*,\bullet}} \sin \delta_*, \quad (94a)$$

$$v_{l_{*,\bullet}} = \sin \delta_* + v_{x_{*,\bullet}} \cos \delta_*. \quad (94b)$$

Rychlosti pneumatik vypočteme z rychlosti auta:

$$v_{y_{f,l}} = \dot{y} + a\dot{\psi} \quad v_{x_{f,l}} = \dot{x} - c\dot{\psi}, \quad (95a)$$

$$v_{y_{f,r}} = \dot{y} + a\dot{\psi} \quad v_{x_{f,r}} = \dot{x} + c\dot{\psi}, \quad (95b)$$

$$v_{y_{r,l}} = \dot{y} - b\dot{\psi} \quad v_{x_{r,l}} = \dot{x} - c\dot{\psi}, \quad (95c)$$

$$v_{y_{r,r}} = \dot{y} - b\dot{\psi} \quad v_{x_{r,r}} = \dot{x} + c\dot{\psi}. \quad (95d)$$

## 4.2 Model pneumatik

Problém modelování pneumatik spočívá v tom, že se skládá z různých materiálů (jedná se tedy o kompozit a nemá všude stejné vlastnosti). Také plocha kontaktu pneumatiky s povrchem vozovky se mění podle zatížení pneumatiky a celá tato plocha nemá stejné vlastnosti, na začátku kontaktu s vozovkou je pevně přimknutá k vozovce a jak se odvaluje, začíná se smýkat. Rozhraní, kde se začíná pneumatika smýkat, se mění podle rychlosti otáčení pneumatiky. Z těchto důvodů existují různé přístupy k problematice a z nich vycházejí různé modely pneumatik. Některé se snaží z části popsat jevy, ke kterým dochází v pneumatice při styku s vozovkou (modely typu grey box), jiné jen aproximují naměřené charakteristiky (black box modely).

V této práci byl zvolen jednoduchý model typu black box, který je popsán v článku [2]. Stejný model je využíván i v článku, který jsme použili pro dynamický model automobilu [6]. Black box modely využívají naměřená data, kterými proloží křivku. Tato křivka může být reprezentována nějakou řadou (Fourierova řada, polynom, ...), nebo speciálním vzorcem. Při reprezentaci řadou je většinou potřeba velké množství koeficientů k přesnému proložení funkce, funkce je zvlhňená, mimo naměřená data má často velký rozdíl mezi skutečností a modelem a změny koeficientů se neprojeví přímo a jednoduše na změnách tvaru křivky [2, p. 1]. Proto k popisu modelu [2] využívá speciální vzorec, jehož parametry jsou dopočítány optimalizačním softwarem z naměřených hodnot.

Příčná síla je popsána rovnicemi:

$$D = a_1 F_z^2 + a_2 F_z, \quad (96a)$$

$$C = 1, 3, \quad (96b)$$

$$B = \frac{a_3 \sin(a_4 \arctan(a_5 F_z))}{CD}, \quad (96c)$$

$$E = a_6 F_z^2 + a_7 F_z + a_8, \quad (96d)$$

$$\varphi = (1 - E) \alpha + \frac{E}{B} \arctan(B\alpha), \quad (96e)$$

$$F_c = D \sin(C \arctan(B\varphi)). \quad (96f)$$

Koeficienty  $a_{\{1, \dots, 8\}}$  jsou vypočteny optimalizačním programem,  $F_z$  je zatížení pneumatiky,  $\alpha$  je úhel směrové úchyly pneumatiky. inter Podélná síla je popsána vzorcí:

$$D = a_1 F_z^2 + a_2 F_z, \quad (97a)$$

$$C = 1, 65, \quad (97b)$$

$$B = \frac{a_3 F_z^2 + a_4 F_z}{C D e^{a_5 F_z}}, \quad (97c)$$

$$E = a_6 F_z^2 + a_7 F_z + a_8, \quad (97d)$$

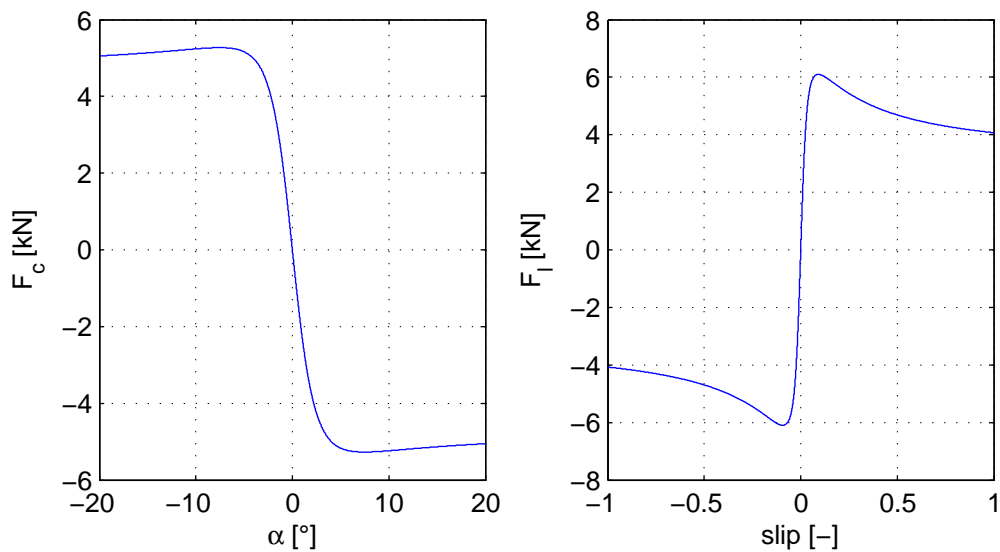
$$\varphi = (1 - E) s + \frac{E}{B} \arctan(Bs), \quad (97e)$$

$$F_l = D \sin(C \arctan(B\varphi)). \quad (97f)$$

Koeficienty  $a_{\{1, \dots, 8\}}$  byly dopočítány z naměřených dat a  $s$  je skluz pneumatiky definovaný v rovnici (87).

Ze vzorců je vidět, že podélná síla závisí jen na skluzu a příčná jen na úhlu natočení.

Používáme jednodušší model, než je v [2], jejich model počítá i s možností náklonu pneumatik, my předpokládáme svislou osu pneumatiky vždy rovnoběžnou s osou Z.



**Obrázek 6** Průběhy  $F_c$  a  $F_l$  při zatížení  $F_z = 6$  kN.

Původní model také umožňuje posunout graf kolem počátku a uvažovat nenulové síly při nulovém skluzu, my je neuvažujeme a tak zanedbáváme tření. V původním modelu jsou data pouze pro brzdění, v naší práci jsme model rozšířili i pro rozjezd tím způsobem, že jsme graf vytvořili souměrně podle počátku.



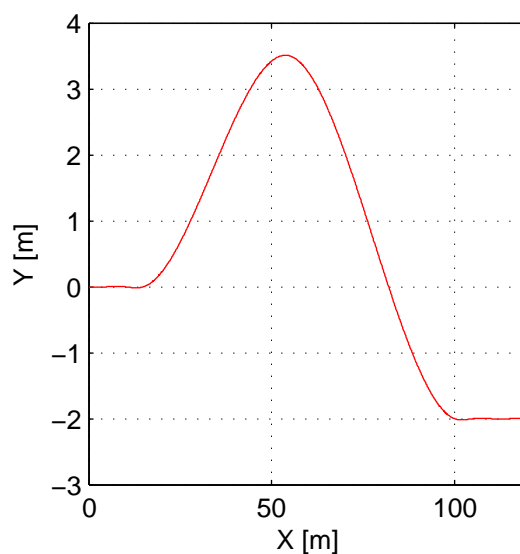
## 5 Analýza citlivosti kvality řízení na redukci problému

Při řízení pomocí MPC regulátoru je nejvýhodnější mít co nejmenší problém kvadratického programování. Menší problém lze vyřešit s méně výkonným procesorem, pokud kvadratický problém řešíme on-line, nebo s menší spotřebou paměti, pokud používáme off-line řešení. Obě metody redukce problému vedou ke změně chování regulátoru. Pokud příliš zredukujeme měkká omezení, hrozí nám, že regulátor nebude dostatečně reagovat a předcházet jejich překročení. Pokud budeme příliš blokovat vstupy regulátoru zhoršíme tím jeho chování při sledování požadovaných referencí.

V této kapitole postupně probereme naše cíle řízení pro model automobilu, nastavení regulátoru a změny chování, pokud měníme blokování vstupů, jejich omezení a různé vzájemné kombinace.

### 5.1 Cíle řízení

Simulaci provádíme na generované trajektorii ve 2D rovině  $XY$ , která je ukázána na Obrázku 7. Tato trajektorie představuje cestu těžiště auta po ploše, kde  $X$  a  $Y$  představují standardní souřadnice roviny. Z trajektorie v  $XY$  vypočteme trajektorie pro úhel natočení automobilu ( $\varphi$ ) od osy  $X$  a úhlovou rychlost natáčení ( $\dot{\varphi}$ ). Sledovanými veličinami je úhel natočení automobilu od osy  $X$   $\varphi$ , rychlost natáčení automobilu  $\dot{\varphi}$  a pozice středu automobilu v ose  $Y$ . Zároveň aplikujeme měkká omezení na rychlost otáčení  $\dot{\varphi}$ , tím simulujeme požadavek na malou rychlost otáčení automobilu pro lepší komfort pasažérů.



Obrázek 7 Referenční trajektorie.

## 5.2 Parametry simulace

### 5.2.1 Konstanty modelu automobilu

Konstanty do modelu automobilu popsaného v kapitole 4, jsme získali z nástroje pro jejich výpočet [5] a jsou uvedeny v Tabulce 1, za konstantu gravitačního zrychlení jsme dosadili standardně používanou hodnotu  $9.81 \text{ ms}^{-2}$ .

**Tabulka 1** Hodnoty konstant pro model automobilu.

konstanta	hodnota	jednotka
a	1,446	(m)
b	1,408	(m)
c	1,437	(m)
I	1549,034	(kgm <sup>2</sup> )
m	2220	(kg)
g	9,81	(ms <sup>-2</sup> )

### 5.2.2 Vstupy a výstupy modelu

Výstup modelu automobilu je ve tvaru:

$$y(k) = \begin{pmatrix} \dot{y} & \dot{x} & \varphi & \dot{\varphi} & Y & X \end{pmatrix}^T. \quad (\text{ms}^{-1}, \text{ms}^{-1}, \text{rad}, \text{rads}^{-1}, \text{m}, \text{m}) \quad (98)$$

První dvě proměnné  $\dot{y}$  a  $\dot{x}$  jsou rychlosti automobilu v osách automobilu, další představuje natočení automobilu  $\varphi$  od osy  $X$ , čtvrtá proměnná  $\dot{\varphi}$  představuje rychlost otáčení automobilu, poslední dvě proměnné  $Y$  a  $X$  představují polohu těžiště automobilu v globálních souřadnicích.

Vstup modelu je ve tvaru:

$$u(k) = \begin{pmatrix} \delta f & s_{fl} & s_{fr} \end{pmatrix}. \quad (^\circ, -, -) \quad (99)$$

První proměnná  $\delta f$  je natočení přední nápravy, zbylé proměnné  $s_{fl}$  a  $s_{fr}$  představují skluz levého a pravého kola přední nápravy.

### 5.2.3 Nastavení regulátoru

Regulátor minimalizuje kritérium které se skládá ze tří částí, první část minimalizuje rozdíl výstupu a reference a je ve tvaru:

$$J_{\mathbf{Q}}(U) = \frac{1}{2T_p} \sum_{k=0}^{T_p-1} \left( (y(k) - r(k))^T \mathbf{Q} (y(k) - r(k)) \right) \quad (100)$$

Druhá část minimalizuje diferenci vstupu a je ve tvaru:

$$J_{\mathbf{R}}(U) = \frac{1}{2T_{vz}} \sum_{k=0}^{T_k-1} \left( (u(k) - u(k-1))^T \mathbf{Q} (u(k) - u(k-1)) \right) \quad (101)$$

Poslední část minimalizuje měkká omezení:

$$J_{\Omega}(U, Z) = \frac{1}{2N} \sum_{k=0}^{T_p-1} \left( (y(k) - z(k))^T \Omega (y(k) - z(k)) \right) \quad (102)$$

Pro všechny tři rovnice (100), (101) a (102) platí omezení:

$$\begin{aligned} \text{s.t. } & x(k+1) = \mathbf{M}x(k) + \mathbf{M}u(k) + d(k), \\ & y(k) = \mathbf{C}x(k) + \mathbf{D}u(k), k = 0, 1, \dots, T_p - 1, \\ & d(k) = \xi(k+1) - \mathbf{A}\xi(k) - \mathbf{B}u_0, \\ & \xi(k) \text{ vývoj stavu nelineárního systému, } k = 0, 1, \dots, T_p, \\ & u_0 \text{ vstup aplikovaný v předchozím kroku,} \\ & u_{min} \leq u(k) \leq u_{max}, k = 0, 1, \dots, T_p - 1, \\ & y_{min} \leq z(k) \leq y_{max}, k = 0, 1, \dots, T_p - 1, \\ & N \text{ počet kontrol měkkých omezení.} \end{aligned}$$

Stav nelineárního systému  $\xi(k)$  je vypočítán ze známého stavu systému  $x(0) = \xi(0)$  a při aplikování konstantního vstupu  $u_0$ , což je vstup, který byl aplikován v předchozím kroku. Tím vypočteme trajektorii systému podél které aproximujeme lineární model.

Celkové kritérium tedy je:

$$J(U, Z) = J_{\mathbf{Q}}(U) + J_{\mathbf{R}}(U) + J_{\Omega}(U, Z). \quad (103)$$

Doba predikce  $T_p$  regulátoru je 30 kroků, vzorkovací perioda je 0,01 sekundy, takže regulátor vidí 0,3 sekundy ve vývoji systému. Sledujeme veličiny  $\varphi$  a  $\dot{\varphi}$  s váhou 10, polohu sledujeme s váhou 20, ostatní veličiny, což jsou rychlost  $\dot{x}$  a  $\dot{y}$  a polohu v ose  $X$ , nesledujeme a proto mají nastavenou hodnotu váhy 0. Matice  $\mathbf{Q}$  je tedy ve tvaru:

$$\mathbf{Q} = \text{diag}(0, 0, 10, 10, 20, 0). \quad (104)$$

Váhy pro velikost vstupů je pro natočení přední nápravy  $\delta f = 1$ , pro skluzu předních kol  $s_{fl}$  a  $s_{fr}$  jsou nastaveny na váhu 100 000. Tento velký rozdíl je způsoben tím, že veličiny mají rozdílné rozsahy. Tedy natočení přední nápravy má 100krát větší váhu než skluz. Přesto je skluz přes normalizaci nastaven 1000krát citlivěji, než natočení přední nápravy z důvodu aby regulátor používal více natočení nápravy, než skluz. Menší rozdíl vedl k tomu, že přední náprava se příliš nenatáčela, kdežto skluzu byly na limitech, čímž omezovali možnosti regulátoru ovlivňovat dynamiku systému.

Matice  $\mathbf{R}$  má tedy tvar:

$$\mathbf{R} = \text{diag}(1, 100\ 000, 100\ 000). \quad (105)$$

Tvrdá omezení vstupu jsou ve tvaru:

$$u_{max} = \begin{pmatrix} 5 & 0,05 & 0,05 \end{pmatrix}, \quad (106)$$

$$u_{min} = \begin{pmatrix} -5 & -0,05 & -0,05 \end{pmatrix}. \quad (107)$$

Matice vážení měkkých omezení:

$$\Omega = \text{diag}(0, 0, 0, 1000, 0, 0). \quad (108)$$

Vidíme tedy, že měkké omezení aplikujeme pouze na úhlovou rychlost  $\dot{\varphi}$ . Omezení jsou nastavena na:

$$y_{max} = \left( \times \quad \times \quad \times \quad 0,17 \quad \times \quad \times \right), \quad (109)$$

$$y_{min} = \left( \times \quad \times \quad \times \quad -0,17 \quad \times \quad \times \right). \quad (110)$$

Měkká omezení jsou aplikována případ od případu a jsou popsána v každém případě zvlášť.

### 5.3 Vliv blokování vstupu

Vliv blokování vstupu na chování regulátoru jsme testovali na čtyřech případech. Strategie blokování je uvedena v Tabulce 2. Čísla udávají kolik následujících bloků se spojí dohromady. Součet by měl odpovídat době predikce, tedy 30. Tedy například případ BL1 má možnost nastavit každý blok samostatně. Případ BL3 spojuje pět následujících kroků dohromady, tedy kroky 1 až 5 mají stejnou hodnotu, dále 6 až 10, 11 až 15 atd.

Neaplikujeme žádné měkké omezení na výstup. Tedy matice  $\Omega$  je:

$$\Omega = \text{diag}(0, 0, 0, 0, 0, 0). \quad (111)$$

Pro posouzení kvality regulace jsme použili kritérium efektivní hodnoty chyby (anglicky root mean square error (RMSE)). Kdy chyba je rozdíl mezi požadovanou trajektorií a skutečnou hodnotovou sledovaných veličiny ( $\varphi$ ,  $\dot{\varphi}$ ,  $Y$ ). Matematicky zapsáno jako:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} e(i)^2}. \quad (112)$$

Kde  $N$  je počet prvků ve vektoru chyb  $e$ , ze kterých chceme RMSE spočítat a  $e(i)$  představuje prvek na pozici  $i$ . Pro posouzení regulace jsme použili kritérium actuator activity (AA), jedná se také o RMSE, ale chybu zde představuje změna velikosti akčního zásahu. Čím menší je AA, tím méně se mění hodnota akčních veličin. Hodnoty RMSE pro sledované výstup a pro actuator activity jsou normalizovány na dobu za jednu sekundu.

$$AA = RMSE(\Delta u) = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-2} (u(i) - u(i+1))^2}. \quad (113)$$

Kde  $u$  představuje vektor vstupu,  $u(i)$  je hodnota vstupu na pozici  $i$ ,  $N$  představuje délku vektoru  $u$ ,  $\Delta u$  představuje vektor rozdílů  $\Delta u(i) = u(i) - u(i+1)$ , pro  $i = 1, 2, \dots, N-1$ .

V Tabulce 3 jsou vidět naměřené výsledky pro sledované vstupy a AA. Vidíme, že čím více použitých bloků které může regulátor ovlivňovat, tím klesá RMSE na sledovaných výstupech. Zároveň roste velikost AA. Kromě posledního případu, kdy vstup do systému  $\delta\phi$  (natočení přední nápravy) dosáhne omezení, což omezí dynamiku systému viz vstupy na Obrázku 8. Jinak výsledek odpovídá předpokladům, čím více má regulátor stupňů volnosti při optimalizaci, tím lépe sleduje požadovanou trajektorii. Cenou za to je větší problém kvadratického programování, který musíme vyřešit.

V posledním případě vidíme na Obrázcích 8 a 9, že jen jeden blok vede k velmi výraznému odchýlení od požadované reference.

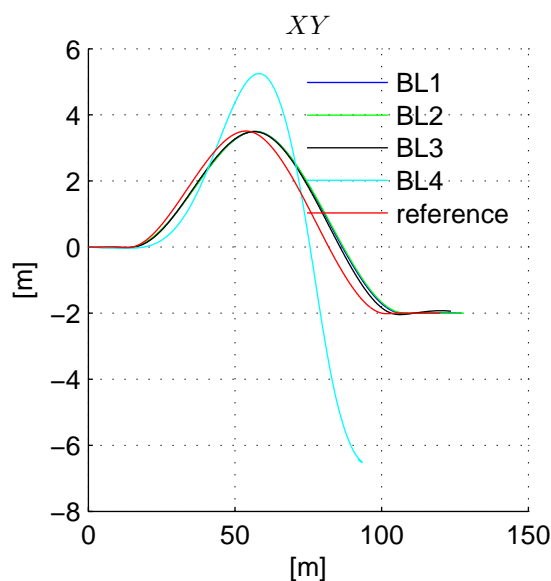
Tabulka 2 Použité blokování

Případ	Použité blokování vstupu	Počet bloků
BL1	(1, 1)	30
BL2	(2, 2)	15
BL3	(5, 5, 5, 5, 5, 5)	6
BL4	(30)	1

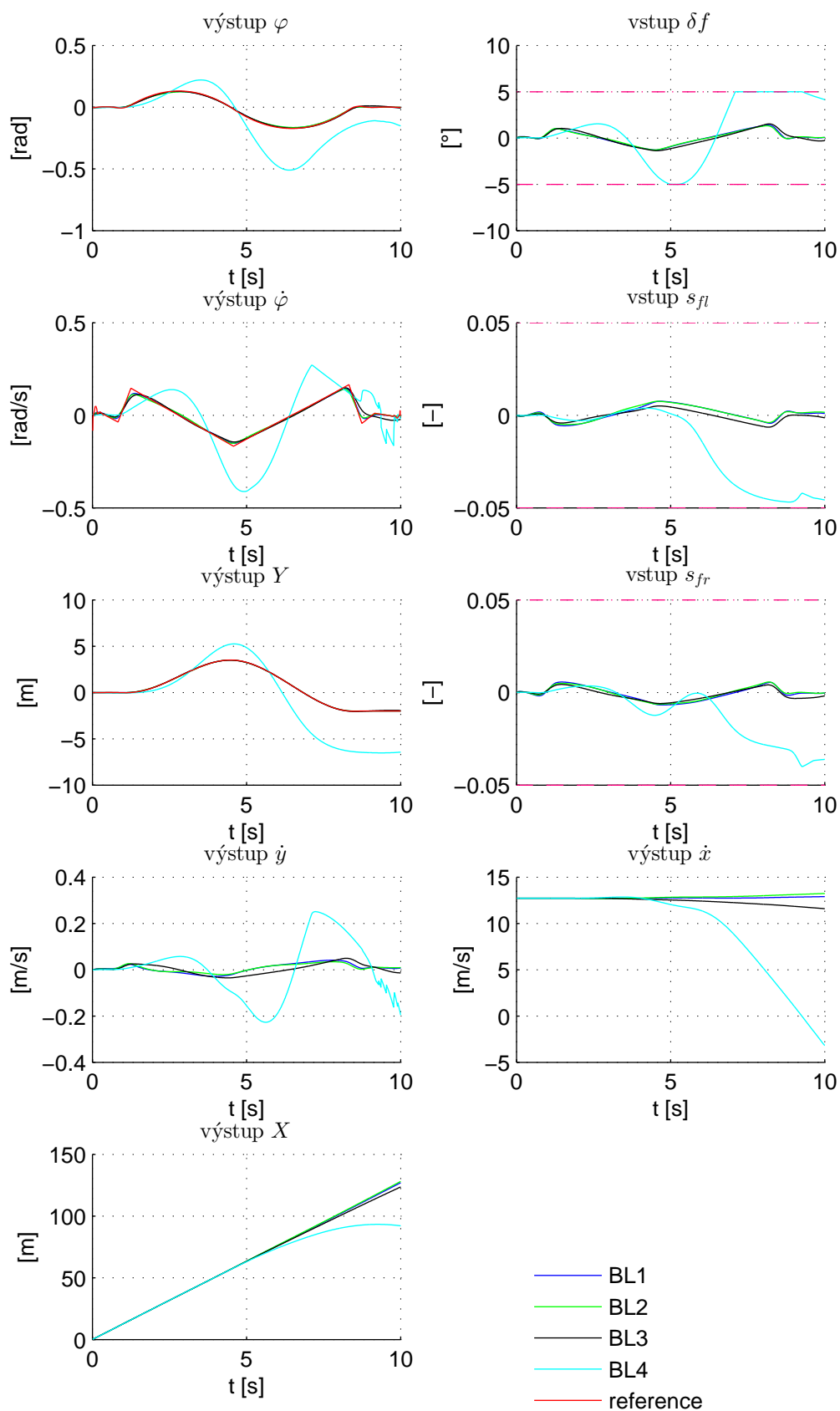
Tabulka 3 Výsledky blokování vstupu.

veličina	BL1	BL2	BL3	BL4
sledované výstupy (RMSE)				
$\varphi$ (rad)	$4.243e-04$	$5.424e-04$	$5.250e-04$	$1.548e-02$
$\dot{\varphi}$ (rad/s)	$8.865e-04$	$1.063e-03$	$1.590e-03$	$1.301e-02$
$Y$ (m)	$6.483e-04$	$1.228e-03$	$2.414e-03$	$2.601e-01$
actuator activity (RMSE)				
$\delta f$ (°)	$1.055e-03$	$1.018e-03$	$1.002e-03$	$3.015e-03$
$s_{fl}$ (-)	$5.624e-06$	$5.064e-06$	$4.379e-06$	$1.023e-05$
$s_{fr}$ (-)	$5.761e-06$	$5.126e-06$	$4.586e-06$	$1.005e-05$
Ostatní veličiny				
velikost QP problému	90	45	18	3

Z výpočtů vidíme, že schopnost regulátoru sledovat trajektorii roste s počtem bloků ve kterých může nastavovat vstupy systému. S počtem bloků ale roste přímo úměrně velikost QP problému. Tedy hledáme optimální počet bloků, kdy systém bude sledovat referenci a zároveň bude velikost QP problému taková, že ji budeme schopni řešit.



Obrázek 9 Trasa auta v rovině XY.



Obrázek 8 Průběhy výstupů sledující referenci a vstupů.

## 5.4 Vliv redukce měkkých omezení

Pro simulaci vlivu měkkých omezení jsme použili matice  $\mathbf{Q}$ ,  $\mathbf{R}$  a  $\mathbf{\Omega}$  v rovnicích (104), (105) a (108). V Tabulce 4 jsou uvedené kroky, kde se aplikovaly měkké omezení. Měkká omezení jsou vidět na výstupech a tvrdá jsou znázorněna na vstupech v Obrázku 11. Pro měření splnění měkkých omezení jsme zavedli chybu překročení omezení, jedná se o RMSE, kde chybu představuje hodnota o kolik došlo k překročení měkkých omezení, pokud v daném kroku nedošlo k překročení je tento krok přeskočen. Matematicky zapsáno, kde  $p(k)$  je výsledná hodnota na pozici  $k$ ,  $lb$  jsou dolní měkká omezení a  $ub$  jsou horní měkká omezení aplikovaná na  $p$ . Znak  $\times$  zančí přeskočení dané pozice a  $i$  představuje indexování výsledného vektoru. Vektor  $e(i)$  představuje nenulové výsledné chyby, které zadáme do výpočtu RMSE. Pokud nedošlo nikde k překročení měkkých omezení je výsledek 0.

$$e(i) = \begin{cases} p(k) < lb & : e(i) = p(k) - lb \\ lb \leq p(k) \leq ub & : e(k) = \times \\ ub < p(k) & : e(i) = p(k) - ub \end{cases} \quad (114)$$

Celé řízení bylo redukováno na jeden blok, takže blokování bylo nastaveno na (30). Je to z důvodu toho, abychom zamezili vlivu blokování na výstup při změně měkkých omezení.

V případě RO1 aplikujeme měkká omezení na celou dobu predikce, v případě RO2 aplikujeme na začátku, v půlce a na konci doby predikce. V případě RO3 aplikujeme měkká omezení jen na první krok. V případě RO4 neaplikujeme měkké omezení vůbec.

**Tabulka 4** Aplikované měkké omezení.

Případ	Kroky kontroly měkkých omezení
RO1	(0, 1, 2, ..., 27, 28, 29)
RO2	(0, 14, 29)
RO3	(0)
RO4	()

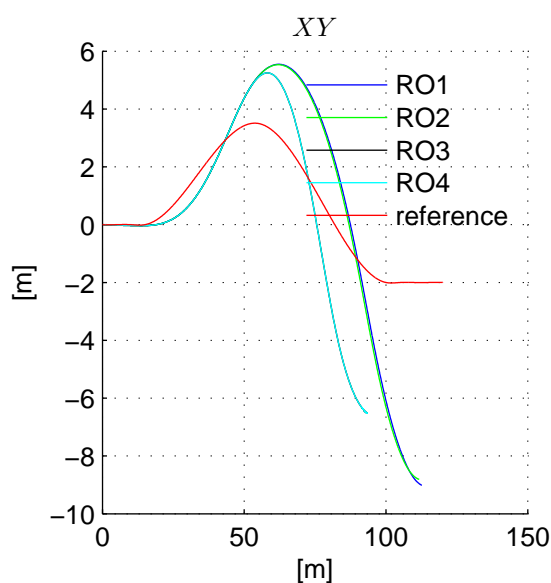
Na případech RO1 a RO2 je vidět, že ačkoli mají rozdílný počet míst aplikovaných měkkých omezení, jejich průběh je velmi podobný viz Obrázek 11.

V případě RO3 a RO4 je průběh stejný, případ RO3 je výjimečný tím, že aplikujeme omezení jen na první krok predikce. Pokud se podíváme na rovnici (35), tak z ní je vidět, že výstup v čase 0 lze změnit jen výstupem přes matici  $\mathbf{D}$  (stav v čase 0 je znám přesně ze systému). Protože pro náš případ je matice  $\mathbf{D}$  nulová, tak nemá vstup vliv na výstup (v nultém kroku predikce) a tak toto omezení nemá na predikci vliv. Proto dostaneme stejný výsledek, jako pro neaktivní měkké omezení.

Na Obrázku 12 je zobrazen úsek průběhu mezi 8 až 10 sekundou. Je zde vidět rozdíl mezi průběhy RO1 a RO2. V prvním případě aplikujeme měkká omezení na každý krok predikce, výsledkem je, že překročení měkkých omezení je menší než ve druhém případě, kdy aplikujeme jen na první, patnáctý a poslední vzorek. Je to způsobeno počtem vzorků, kdy dojde k překročení omezení. Čím pozdější vzorek sledujeme, tím později překračujeme měkká omezení a dříve se vracíme do povolené oblasti. Je to způsobeno tím, jak regulátor zjišťuje překročení omezení, pokud sledujeme omezení jen na počátku doby predikce, tak může dojít k situaci, kdy daný vzorek měkká omezení splňuje, ale na konci doby predikce dojde k jejímu překročení. Naopak pokud sledujeme

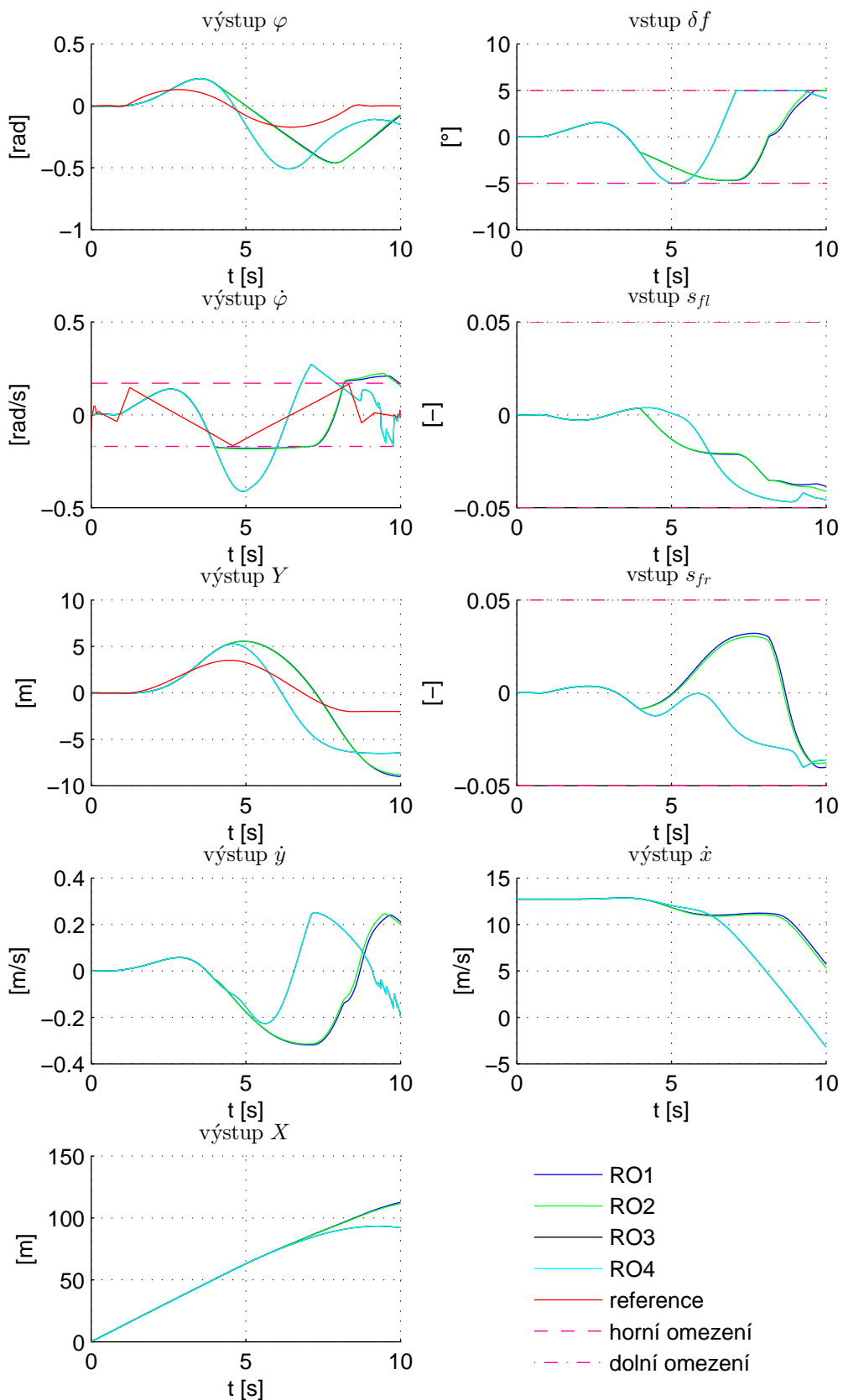
**Tabulka 5** Výsledky redukce měkkých omezení.

veličina	RO1	RO2	RO3	RO4
sledované výstupy (RMSE)				
$\varphi$ (rad)	$1.725e - 02$	$1.723e - 02$	$1.548e - 02$	$1.548e - 02$
$\dot{\varphi}$ (rad/s)	$1.292e - 02$	$1.302e - 02$	$1.301e - 02$	$1.301e - 02$
$Y$ (m)	$2.776e - 01$	$2.737e - 01$	$2.601e - 01$	$2.601e - 01$
actuator activity (RMSE)				
$\delta f$ (°)	$2.498e - 03$	$2.527e - 03$	$3.015e - 03$	$3.015e - 03$
$s_{fl}$ (-)	$7.761e - 06$	$7.855e - 06$	$1.023e - 05$	$1.023e - 05$
$s_{fr}$ (-)	$2.125e - 05$	$2.070e - 05$	$1.005e - 05$	$1.005e - 05$
Ostatní veličiny				
velikost QP problému	33	12	6	3
RMSE poruseni (rad/s)	$3.555e - 03$	$4.692e - 03$	$4.504e - 02$	$4.504e - 02$

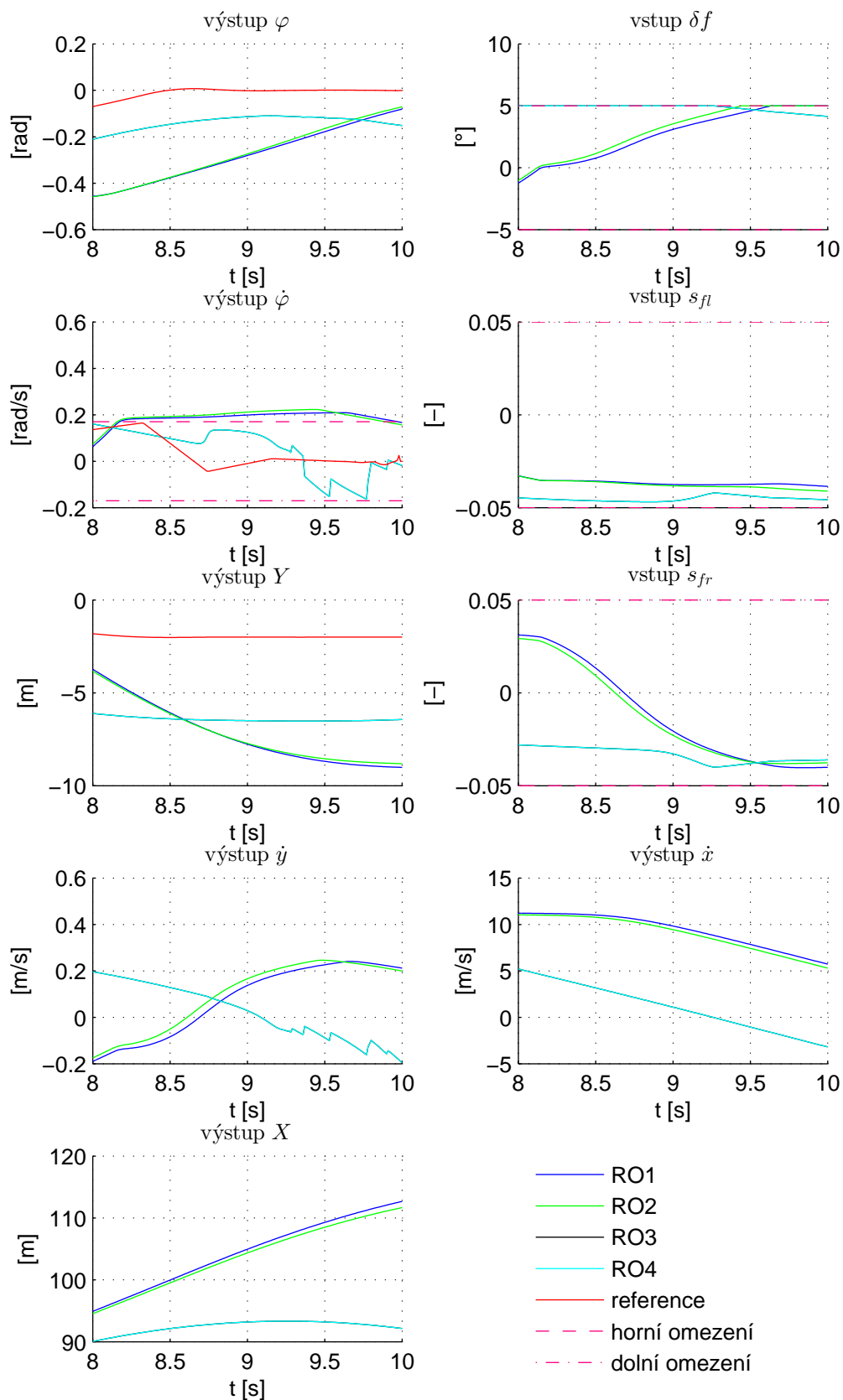
**Obrázek 10** Trasa auta v rovině XY.

překročení na konci doby predikce, regulátor bude volit takovou trajektorii, aby na konci doby predikce došlo k jejímu splnění.





Obrázek 11 Průběhy výstupů sledující referenci a vstupů.



Obrázek 12 Zvětšené průběhy výstupů sledující referenci a vstupů.

## 5.5 Vliv kombinace blokování vstupu a redukce měkkých omezení

Nejkomplexnější případ je kombinace blokování vstupu a redukce měkkých omezení. Oběma způsoby snižujeme velikost problému, ale také zhoršujeme schopnost regulátoru sledovat referenci a předcházet porušení měkkých omezení.

### 5.5.1 Vliv změny blokování na splnění měkkých omezení

V Tabulkách 6, 7 a 8 jsou uvedeny hodnoty pro měnící se velikost blokování bez omezení. Pro první dva případy CO1 a CO2 nedochází k překročení omezení. Pokud snížíme počet bloků na 3, dojde v případě CO3 k překročení omezení, které vidíme na Obrázku 16.

**Tabulka 6** Aplikované měkké omezení na výstup  $\varphi$ .

Případ	Kroky kontroly měkkých omezení
CO1	(-)
CO2	(-)
CO3	(-)

**Tabulka 8** Výsledky kombinace blokování vstupu a redukce měkkých omezení.

veličina	CO1	CO2	CO3
sledované výstupy (RMSE)			
$\varphi$	$5.250e - 04$	$7.545e - 04$	$4.412e - 03$
$\dot{\varphi}$	$1.590e - 03$	$2.139e - 03$	$7.100e - 03$
$Y$	$2.414e - 03$	$4.608e - 03$	$2.224e - 02$
actuator activity (RMSE)			
$\delta f$	$1.002e - 03$	$1.055e - 03$	$1.750e - 03$
$s_{fl}$	$4.379e - 06$	$4.290e - 06$	$5.164e - 06$
$s_{fr}$	$4.586e - 06$	$4.553e - 06$	$6.619e - 06$
Ostatní veličiny			
velikost QP problému	18	15	9
RMSE porušení ( $rad/s$ )	0	0	$3.141e - 02$

Tedy pokud má regulátor dostatečně velký počet bloků, je schopen splnit měkká omezení pouhým dobrým sledováním reference. Pokud předpokládáme, že reference je uvnitř měkkých omezení.

**Tabulka 7** Použité blokování

Případ	Použité blokování vstupu	Počet bloků
CO1	(5, 5, 5, 5, 5, 5)	6
CO2	(6, 6, 6, 6, 6)	5
CO3	(10, 10, 10)	3

### 5.5.2 Vliv umístění kontroly překročení měkkých omezení

V Tabulkách 9, 10 a 11 jsou zobrazeny výsledky pro blokování vstupu 3krát 10 spojených kroků. V prvním sloupci je uveden případ CO3 kdy je použité prosté blokování bez měkkých omezení. To vede k výraznému překročení měkkých omezení. V případě CO5

kontrolujeme překročení měkkých omezení v každém kroku predikce. To vede na relativně velký QP problém. Na základě metriky překročení omezení vidíme, že překračuje měkká omezení více než případ CO6. To je způsobeno normalizací kritéria překročení měkkých omezení.

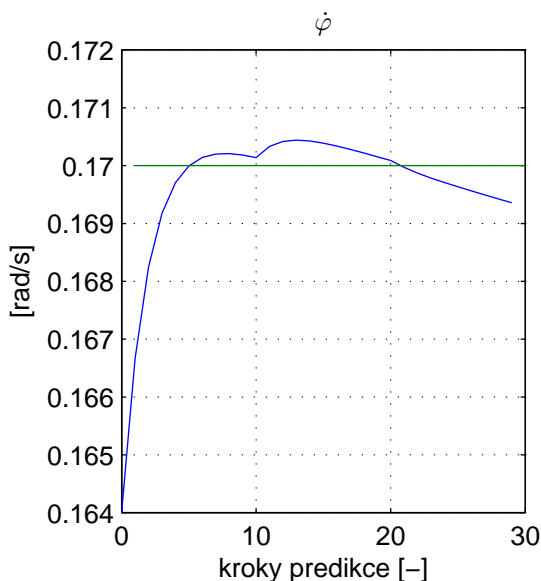
**Tabulka 9** Aplikované měkké omezení na výstup  $\dot{\varphi}$ .

Případ	Kroky kontroly měkkých omezení
CO3	(-)
CO4	(0, 1, 2, ..., 27, 28, 29)
CO5	(9, 15)
CO6	(1, 29)
CO7	(4)

Pokud bychom uvažovali jen možnosti chyby překročení omezení z množiny  $\{0, 1\}$ . Tedy buď nedochází překročení omezení, nebo má jednotkovou velikost. Pokud dojde k překročení po celé době predikce, bude to  $n$  vzorků násobeno  $\frac{1}{N}$ , tedy penalizace za překročení měkkých omezení bude do kritéria přičtena celá. Pokud však dojde k překročení jen v jednom vzorku, bude započtena jen penalizace o velikosti  $\frac{1}{N}$ .

To je vlastnost, kterou obecně požadujeme od normalizace, ale v tomto případě vede k horšímu výsledku. Na Obrázku 14 je zobrazena predikce regulátoru případu CO3 v čase 8.17 s, predikovaná trajektorie porušuje omezení jen ve středu doby predikce. Tedy v případě CO4 mají měkká omezení na krajích doby predikce porušení nulové a umožňují ve středu dojít k vyšším hodnotám porušení omezení, ale penalizace za člen měkkých omezení bude stejná. V případě CO5 detekujeme porušení omezení jen ve středu a tedy regulátor více omezuje překročení měkkých omezení. Naopak v případě CO6 detekujeme měkká omezení na okrajích doby predikce, takže regulátor bude překročení detekovat až později.

V případě CO7 vidíme nejmenší překročení měkkých omezení.



**Obrázek 14** Predikce vývoje systému.

Dopad normalizace kritéria si můžeme představit na jednoduchém případě. Mějme dobu predikce délky  $N$  a v každém kroku se kontroluje překročení měkkých omezení, takže celé kritérium bude násobeno  $\frac{1}{N}$ .

## 5.5 Vliv kombinace blokování vstupu a redukce měkkých omezení

**Tabulka 11** Výsledky kombinace blokování vstupu a redukce měkkých omezení.

veličina	CO3	CO4	CO5	CO6	CO7
sledované výstupy (RMSE)					
$\varphi$	$4.412e - 03$	$4.207e - 03$	$4.211e - 03$	$4.203e - 03$	$4.201e - 03$
$\dot{\varphi}$	$7.100e - 03$	$6.878e - 03$	$6.885e - 03$	$6.867e - 03$	$6.877e - 03$
$Y$	$2.224e - 02$	$2.341e - 02$	$2.344e - 02$	$2.335e - 02$	$2.350e - 02$
actuator activity (RMSE)					
$\delta f$	$1.750e - 03$	$1.568e - 03$	$1.568e - 03$	$1.570e - 03$	$1.564e - 03$
$s_{fl}$	$5.164e - 06$	$5.119e - 06$	$5.111e - 06$	$5.132e - 06$	$5.130e - 06$
$s_{fr}$	$6.619e - 06$	$6.539e - 06$	$6.548e - 06$	$6.534e - 06$	$6.530e - 06$
Ostatní veličiny					
velikost QP problému	9	39	11	11	10
RMSE porušení ( $rad/s$ )	$3.141e - 02$	$2.198e - 03$	$1.877e - 03$	$3.045e - 03$	$1.032e - 03$

V Tabulkách 12, 13 a 14 jsou zobrazeny výsledky pro případ CO8, kdy je umístěna kontrola měkkých omezení na počátek bloku (s výjimkou prvního bloku, protože nultý krok

nemá na měkké omezení vliv, takže omezení je kontrolováno v druhém kroku). V případě CO9 je kontrola umístěna na středy jednotlivých bloků a v případě CO10 na konci doby predikce. Opět je porovnávám s CO3, kdy nejsou žádná měkká omezení kontrolována. Vidíme, že aplikace měkkých omezení vede ke snížení jejich překročení, nicméně nejlepšího výsledku dosáhneme v případě CO7.

**Tabulka 10** Použité blokování

Případ	Použité blokování vstupu	Počet bloků
CO3	(10, 10, 10)	3
CO4	(10, 10, 10)	3
CO5	(10, 10, 10)	3
CO6	(10, 10, 10)	3
CO7	(10, 10, 10)	3

**Tabulka 12** Aplikované měkké omezení na výstup  $\dot{\varphi}$ .

Případ	Kroky kontroly měkkých omezení
CO3	(-)
CO8	(1, 10, 20)
CO9	(4, 14, 24)
CO10	(9, 19, 29)

**Tabulka 13** Použité blokování

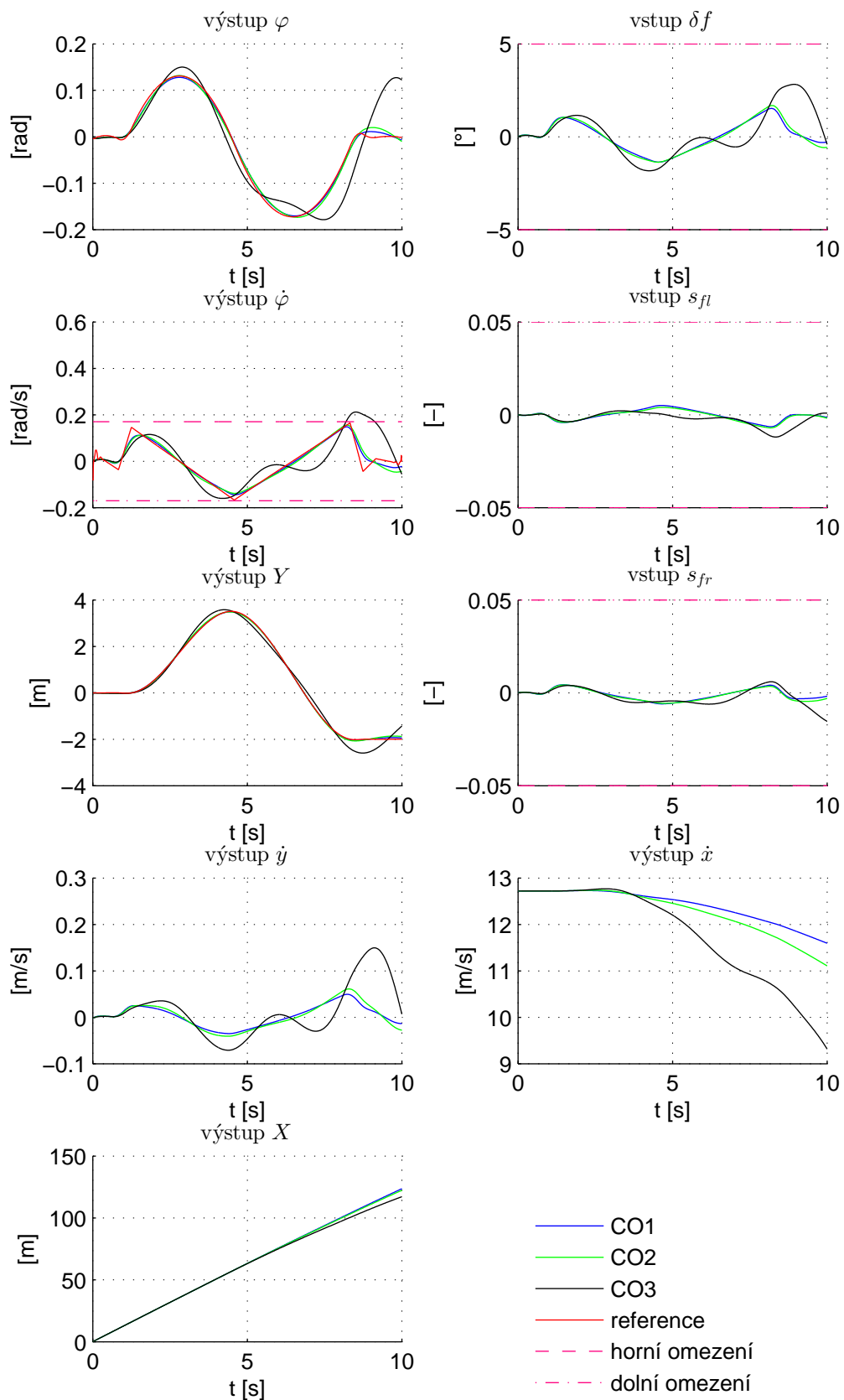
Případ	Použité blokování vstupu	Počet bloků
CO3	(10, 10, 10)	3
CO8	(10, 10, 10)	3
CO9	(10, 10, 10)	3
CO10	(10, 10, 10)	3

**Tabulka 14** Výsledky kombinace blokování vstupu a redukce měkkých omezení.

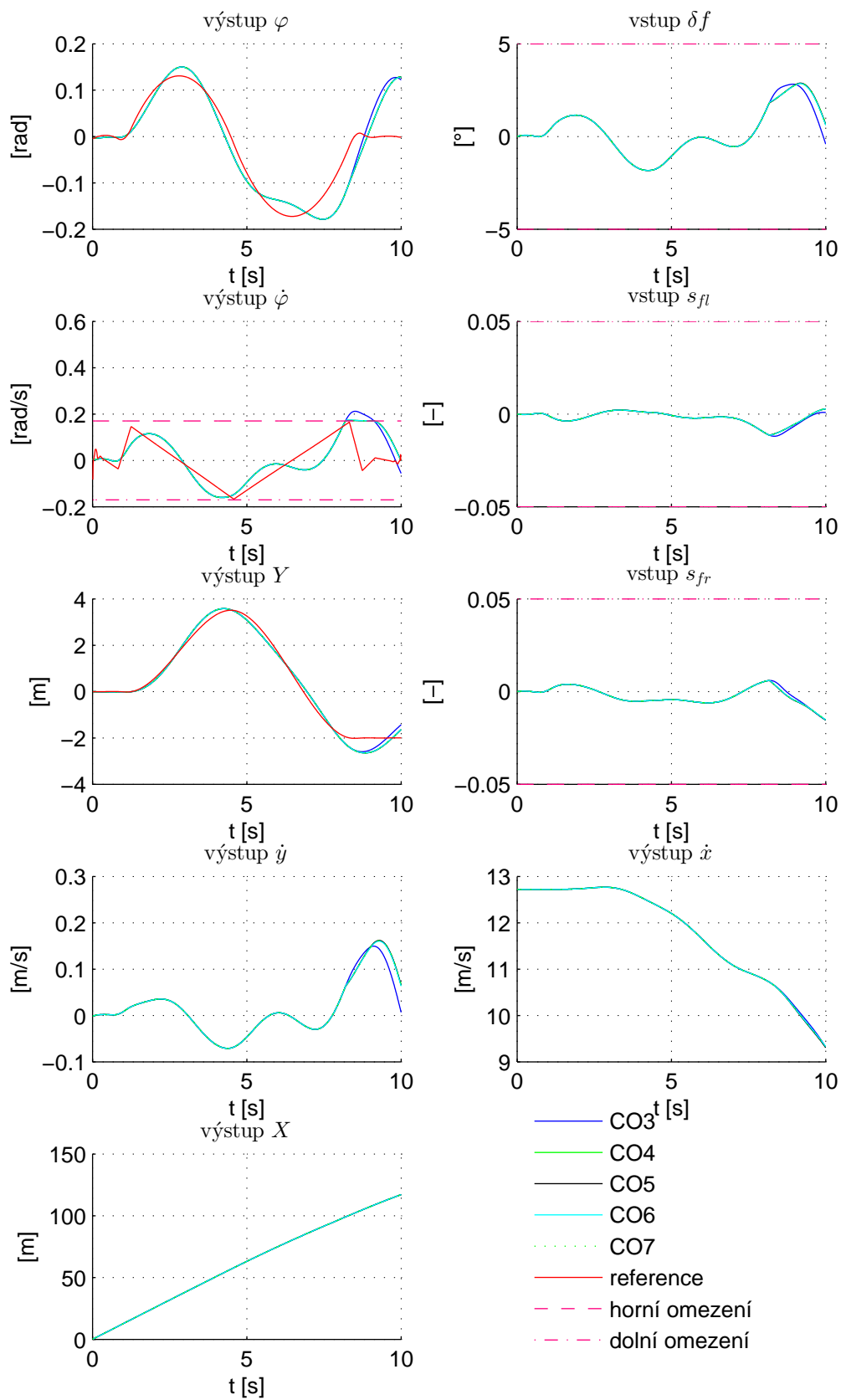
veličina	CO3	CO8	CO9	CO10
sledované výstupy (RMSE)				
$\varphi$	$4.412e - 03$	$4.205e - 03$	$4.205e - 03$	$4.212e - 03$
$\dot{\varphi}$	$7.100e - 03$	$6.877e - 03$	$6.876e - 03$	$6.883e - 03$
$Y$	$2.224e - 02$	$2.343e - 02$	$2.343e - 02$	$2.339e - 02$
actuator activity (RMSE)				
$\delta f$	$1.750e - 03$	$1.566e - 03$	$1.567e - 03$	$1.570e - 03$
$s_{fl}$	$5.164e - 06$	$5.121e - 06$	$5.123e - 06$	$5.110e - 06$
$s_{fr}$	$6.619e - 06$	$6.539e - 06$	$6.534e - 06$	$6.550e - 06$
Ostatní veličiny				
velikost QP problému	9	12	12	12
RMSE porušení ( $rad/s$ )	$3.141e - 02$	$1.953e - 03$	$2.002e - 03$	$2.346e - 03$

**Tabulka 15** Optimální nastavení pro danou trajektorii.

blokování vstupů	(10, 10, 10)
měkké omezení	(4)

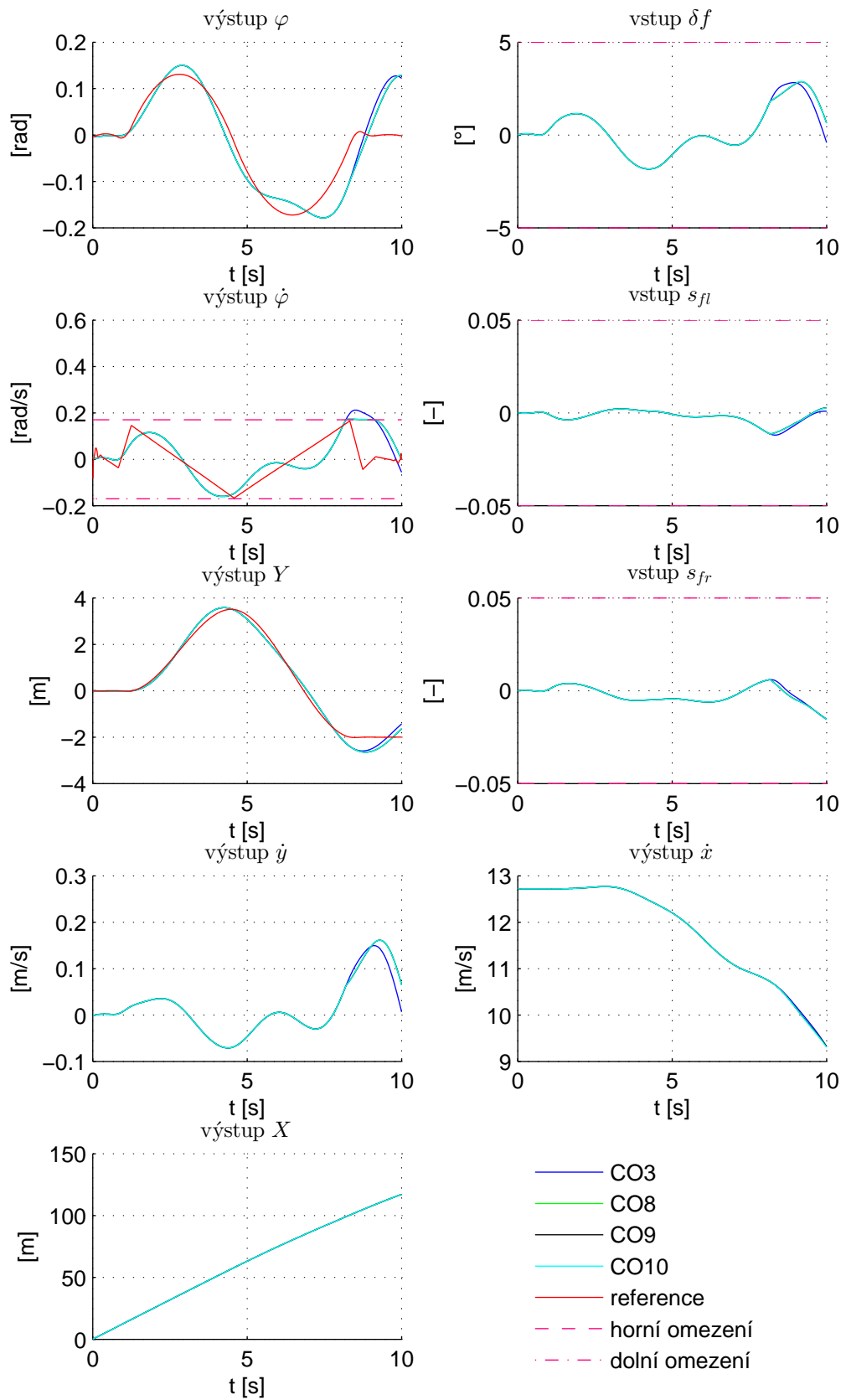


Obrázek 13 Překročení omezení vlivem blokování.



Obrázek 15 Překročení omezení vlivem redukce omezení.





Obrázek 16 Překročení omezení vlivem redukce omezení.

## 6 Závěr

V práci jsme ověřili, že ke splnění měkkých omezení musí MPC regulátor detekovat překročení měkkých omezení. Při redukci měkkých omezení musí být zachováno testování překročení měkkých omezení na takovém místě, kde dojde k jejich překročení. V ideálním případě by regulátor testoval překročení měkkých omezení, jen v těch krocích, kdy dojde k překročení omezení. Pokud regulátor bude kontrolovat omezení v krocích kde nedojde k překročení omezení, zvyšuje to možnost překročení měkkých omezení díky mechanismu popsaného v části 5.5.

Pro náš konkrétní systém a danou trajektorii vychází nejlépe případ, kdy systém kontroluje překročení ve čtvrtém kroku doby predikce. Důvodem toho je průběh překročení omezení v predikci, které nastává v první polovině doby predikce. Vlivem zjednodušení v predikci však na konci doby predikce už model v regulátoru nesouhlasí s nelineárním modelem automobilu, tedy prodloužení doby predikce by už nemělo přílišný vliv na funkci regulátoru.

Optimální nastavení regulátoru pro danou trajektorii jsme určili jako tři bloky po 10 krocích s kontrolou překročení měkkých omezení v čtvrtém kroku.

V dalším pokračování práce se zaměříme na rozšíření MPC regulátoru na plně nelineární systém, což by mohlo zlepšit predikci regulátoru. Dalším cílem by mohlo být vytvořit teorii, která by umožňovala určit velikost bloků a redukce měkkých omezení která by zaručovala splnění požadavků na kvalitu regulace (mohlo by se jednat o jakékoli kritérium), případně by určila ideální blokování a redukce pro dané kritérium regulace.

## Literatura

- [1] Ondřej Šantin. “Influence of Model Uncertainty on Constraints Handling in Predictive Control”. Dipl. the Czech Republic: Czech Technical University in Prague, 2009.
- [2] Nyborg L. Bakker E. a H. Pacejka. “Tyre Modelling for Use in Vehicle Dynamics Studies”. In: *SAE Technical Paper 870421*. 1987. DOI: 10.4271/870421.
- [3] Alberto Bemporad et al. “The explicit linear quadratic regulator for constrained systems”. English. In: *Automatica* 38.1 (2002), s. 3–20.
- [4] R. Cagienard et al. “Move Blocking Strategies in Receding Horizon Control”. In: *Journal of Process Control* 17.6 (čvc 2007), s. 563–570. URL: <http://control.ee.ethz.ch/index.cgi?action=details;id=2691;page=publications>.
- [5] Charlie Constant. *Vehicle inertia calculation tool*. <http://www.car-engineer.com/vehicle-inertia-calculation-tool/>. 2015.
- [6] P. Falcone et al. “A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems”. In: *Decision and Control, 2007 46th IEEE Conference on*. 2007, s. 2980–2985. DOI: 10.1109/CDC.2007.4434137.
- [7] H.J. Ferreau, A. Potschka a C. Kirches. *qpOASES webpage*. <http://www.qpOASES.org/>. 2007–2014.
- [8] H.J. Ferreau et al. “qpOASES: A parametric active-set algorithm for quadratic programming”. In: *Mathematical Programming Computation* 6.4 (2014), s. 327–363.
- [9] L. Grüne a J. Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms*. 1st ed. Communications and Control Engineering. Springer, 2011, s. xi+359. URL: <http://www.springer.com/978-0-85729-500-2>.
- [10] Vladimír Havlena a Jiří Findejs. “Application of model predictive control to advanced combustion control”. English. In: *Control Engineering Practice* 13.6 (2005), s. 671–680. URL: [www.summon.com](http://www.summon.com).
- [11] Daniel Pachner Vladimír Havlena Jiří Roubal Jaroslav Pekař. *Moderní Teorie Řízení*. Vydavatelství ČVUT, 2005. ISBN: 80-01-03208-6.
- [12] Jan M. Maciejowski. *Predictive control: with constraints*. English. Harlow: Prentice Hall, 2002. ISBN: 0201398230; 9780201398236.
- [13] D. Q. Mayne et al. “Survey Constrained Model Predictive Control: Stability and Optimality”. In: *Automatica* 36.6 (červ. 2000), s. 789–814. ISSN: 0005-1098. DOI: 10.1016/S0005-1098(99)00214-9. URL: [http://dx.doi.org/10.1016/S0005-1098\(99\)00214-9](http://dx.doi.org/10.1016/S0005-1098(99)00214-9).
- [14] JorgeJ. Moré a Gerardo Toraldo. “Algorithms for bound constrained quadratic programming problems”. English. In: *Numerische Mathematik* 55.4 (1989), s. 377–400. ISSN: 0029-599X. DOI: 10.1007/BF01396045. URL: <http://dx.doi.org/10.1007/BF01396045>.

- [15] J. Nocedal a S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. ISBN: 9780387303031. URL: <https://books.google.cz/books?id=eN1PAAAAAMAAJ>.
- [16] V. Havlena O. Šantin. “Combined Gradient and Newton Projection Quadratic PProgramming Solver for MPC”. In: 2011.
- [17] *quadprog*. <http://www.mathworks.com/help/optim/ug/quadprog.html>. 2015.
- [18] J.A Rossiter. *Model-based predictive control a practical approach*. CRC press, 2003. ISBN: 0-8493-1291-4.
- [19] Neha Raghu Ruchika. “Model Predictive Control: History and Development”. In: 6. 2013, s. 2600–2602.

Na přiloženém CD jsou kódy pro vygenerování obrázků a zdrojové kódy použitého řešitele qpOases.