

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Fakulta chemickej a potravinárskej technológie**

Evidenčné číslo: FCHPT-5415-61813

# **Prediktívne riadenie flexibilného dopravného pásu**

**Bakalárska práca**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Fakulta chemickej a potravinárskej technológie**

Evidenčné číslo: FCHPT-5415-61813

# **Prediktívne riadenie flexibilného dopravného pásu**

**Bakalárska práca**

Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve

Číslo študijného odboru: 2621

Názov študijného odboru: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo

Školiace pracovisko: Ústav informatizácie, automatizácie a matematiky

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

**Bratislava 2014**

**Monika Bučková**



## ZADANIE BAKALÁRSKEJ PRÁCE

Študentka: **Monika Bučková**  
ID študenta: 61813  
Študijný program: automatizácia, informatizácia a manažment v chémii a potravinárstve  
Kombinácia študijných odborov: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo  
Vedúci práce: doc. Ing. Michal Kvasnica, PhD.

Názov práce: **Prediktívne riadenie flexibilného dopravného pásu**

Špecifikácia zadania:

Cieľom práce je navrhnuť prediktívny regulátor pre flexibilný dopravný pás. Zariadenie je tvorené 6 piestami, pomocou ktorých na páse tvarujeme vlnu, ktorá pohybuje objektom. Úlohou je navrhnuť optimálne riadenie jednotlivých piestov tak, aby bol objekt prepravený do želanej polohy za čo najkratší čas.

Úlohy:

- \* Vytvoriť matematický model dopravníka.
- \* Navrhnuť prediktívny regulátor, ktorý vytvára požadovaný sklon pásu pomocou krajných piestov.
- \* Preskúmať možnosť vytvorenia prediktívneho regulátora, ktorý ovláda všetky piesty naraz.
- \* Regulátor implementovať v prostredí MATLAB/Simulink a pripojiť ho na reálne zariadenie.
- \* Experimentálne verifikovať navrhnutý regulátor.

Rozsah práce: 30

Zoznam odbornej literatúry:

1. Kvasnica, M. *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools : Theory and Tools*. Saarbrücken: VDM Verlag Dr. Müller, 2009. 231 s. ISBN 978-3-639-20644-9.

Riešenie zadania práce od: 17. 02. 2014

Dátum odovzdania práce: 24. 05. 2014

L. S.

**Monika Bučková**  
študentka

**prof. Ing. Miroslav Fikar, DrSc.**  
vedúci pracoviska

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

## **Podakovanie**

Týmto chcem poďakovať svojmu školiteľovi, doc. Ing. Michalovi Kvasnicovi, PhD., za odbornú pomoc, usmernenie pri vypracovaní mojej práce, cenné rady a ochotu. Osobitné poďakovanie patrí aj mojej rodine za podporu a pochopenie.

# Abstrakt

Cielom práce bolo navrhnúť prediktívny regulátor pre flexibilný dopravný pás tvorený šiestimi piestami. Pomocou vlny, tvarovanej piestami, sme sa snažili objekt, v našom prípade loptičku, dopraviť na žiadanú pozíciu za čo najkratší čas. Prvou úlohou práce bolo vytvoriť matematický model zariadenia. Následne bolo potrebné navrhnúť prediktívny regulátor, ktorý vytváral požadovaný sklon na základe pohybu krajných piestov. Tento regulátor sme implementovali v prostredí MATLAB/Simulink a porovnali ho s ďalším typom regulátora, konkrétne s PID regulátorom. Po odsimulovaní procesu sme prediktívny regulátor pripojili aj na reálne zariadenie a experimentálne sme overili, či bol navrhnutý správne.

## Kľúčové slová:

mpc, prediktívne riadenie, loptička na páse

# Abstract

The goal of this project was to design a predictive controller for a flexible conveyor belt, which consists of six pistons. Using wave shaped by pistons, we tried to relocate the object, in this case the ball, to the desired position in the shortest time. In the first part mathematical model was created. Then, it was necessary to design a predictive controller that generates the desired angle achieved by the movement of the pistons. This controller was implemented in MATLAB/ Simulink and it was compared to PID controller. After simulated runs, predictive controller was connected to the real device, and the design was experimentally verified.

## **Keywords:**

mpc, model predictive control, ball and beam

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Prediktívne riadenie</b>	<b>3</b>
2.1	Zložky prediktívneho riadenia . . . . .	3
2.1.1	Matematický model procesu . . . . .	4
2.1.2	Horizont predikcie . . . . .	4
2.1.3	Účelová funkcia . . . . .	5
2.1.4	Ohraničenia . . . . .	5
2.2	Princíp riadenia . . . . .	7
<b>3</b>	<b>Flexibilný dopravný pás</b>	<b>8</b>
3.1	Fyzikálny opis . . . . .	8
3.2	Matematický model . . . . .	11
<b>4</b>	<b>Simulácie</b>	<b>13</b>
4.1	MPC vs. PID . . . . .	13
4.1.1	Riešenie pomocou PID . . . . .	13
4.1.2	Riešenie pomocou MPC . . . . .	14
4.1.3	Porovnanie . . . . .	17
4.2	Riadenie pomocou MPC . . . . .	18

<i>Obsah</i>	viii
<b>5 Riadenie reálneho procesu</b>	<b>20</b>
5.1 Postup riadenia . . . . .	20
5.1.1 Riadenie cez m-file . . . . .	22
5.1.2 Riadenie vytvorením schémy v Simulinku . . . . .	22
5.2 Výsledky . . . . .	23
<b>6 Záver</b>	<b>28</b>
<b>Literatúra</b>	<b>30</b>



# Kapitola 1

## Úvod

Mnohé z riadených systémov, s ktorými sa stretávame v živote, vieme uradiť jednoducho. Patria sem procesy bez ohraničení. Tieto rozdeľujeme na systémy s jedným akčným zásahom alebo s väčším počtom vstupov. Prvý spomenutý typ vieme riadiť pomocou PI alebo PID regulátorov, pre druhý typ sa dá použiť LQR riadenie. Akonáhle však pridáme ohraničenia, pri klasických regulátoroch strácame garanciu spätnoväzbovej stability, čím sa procesy stávajú neovládateľnými, a teda nedokážu fungovať bezpečne.

Drvivá väčšina moderných priemyselných procesov a technologických systémov má určené ohraničenia, či už na akčné zásahy, stavy a výstupy, alebo dokonca na všetky spomenuté veličiny. Praktickým príkladom takéhoto systému je napríklad lietadlo. Keďže máme ohraničenia na náklon klapiek, je možné realizovať iba určitý druh manévrov. To môže spôsobiť, že nie je vždy možné zabrániť zrážke lietadiel, keďže kvôli ohraničeniam nie je možné okamžite zmeniť smer, prípadne výšku. Riadenie takto náročných procesov je teda potrebné študovať v laboratóriách.

V našej práci sme sa rozhodli riadiť systém loptičky na páse pre jeho jednoduchý a bezpečný mechanizmus, ale zároveň zadané ohraničenia. Naším cieľom bolo pomocou piestov tvarovať pás a tak posúvať loptičku k požadovanej polohe. Plocha, po ktorej sa loptička pohybovala bola ohraničená a navyše sme mali ohraničenia aj

na maximálnu výchylku piestov, t.j. na akčné zásahy. V dôsledku toho sme sa rozhodli riadiť tento systém pomocou moderného spôsobu riadenia, a to prediktívneho riadenia.

Prediktívne riadenie, ako aj názov napovedá, dokáže pozeráť do budúcnosti a tak zohľadňovať okrem aktuálneho stavu aj budúci, priamo pracuje s ohraňčeniami a ľahko sa vysporiada aj s viacerými vstupmi a výstupmi. Jeho nevýhodou je potreba veľmi dobre opísať proces matematickým modelom, pretože čím viac sa bude odlišovať, o to výraznejšie odchýlky v predpovedi stavov a optimalizácii akčných zásahov dostaneme.

Prvou úlohou našej práce preto bolo vytvoriť matematický model opisujúci pohyb loptičky na páse vzhľadom na zdvih piestov. Následne sme vytvorili simulácie, kde sme proces riadili okrem prediktívneho aj PID regulátorom a tieto spôsoby sme navzájom porovnali. Nakoniec sme riadenie aplikovali aj na reálne zariadenie.

## Kapitola 2

# Prediktívne riadenie

Prediktívne riadenie alebo MPC (Model Predictive Control) sa začalo rozvíjať koncom 70. rokov minulého storočia a za tento čas prešlo z teoretických návrhov k rozšírenému praktickému využitiu v priemysle. Patrí teda medzi moderné spôsoby riadenia systémov.[1]

Hlavnou výhodou MPC v porovnaní s klasickými metódami riadenia je schopnosť predvídať budúce stavy systému a priamo pracovať s ohraničeniami na akčné zásahy, výstupy a stavy zariadenia. Tieto vlastnosti spôsobujú, že výpočty optimálnych akčných zásahov sú zložitejšie a náročnejšie ako napríklad pri PID regulátoroch, čo bol v minulosti dôvod, prečo sa MPC používalo len pri pomalších procesoch.[2] V súčasnosti, vďaka výraznému napredovaniu a rozvoju techniky sa postupne aplikuje aj pri rýchlejších procesoch. Ďalšou dôležitou vlastnosťou je jeho jednoduchá aplikácia pri riadení systémov s viacerými vstupmi a výstupmi (systémy MIMO - multiple inputs and multiple outputs).

### 2.1 Zložky prediktívneho riadenia

Hlavnými prvkami, ktoré spájajú metódy prediktívneho riadenia, sú: model procesu, účelová funkcia, horizont predikcie a ohraničenia.

### 2.1.1 Matematický model procesu

Matematický model sa používa na predikciu nasledujúceho riadeného výstupu systému využitím aktuálnych informácií a známej trajektórie žiadanej veličiny v budúcnosti. [3] Riadený systém môžeme popísať nasledovným modelom:

$$\dot{x} = Ax + Bu, \quad (2.1)$$

ktorý môžeme zapísať v určitom čase  $t$  v tvare:

$$x(t + \Delta t) = Ax(t) + Bu(t), \quad (2.2)$$

čo vo všeobecnosti v diskretnom opise označujeme ako:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k, \\ x_0 &= x(t), \end{aligned} \quad (2.3)$$

kde  $x$  je stav,  $u$  je akčný zásah,  $A$  je matica stavov,  $B$  je matica vstupu,  $C$  je matica výstupu a  $k$  označuje násobok periódy vzorkovania  $\Delta t$ .

### 2.1.2 Horizont predikcie

MPC je implementované v časovom intervale, tzv. predikčnom horizonte. Je to počet periód, v ktorých prebieha vždy nová optimalizácia, ale iba prvý vypočítaný zásah sa použije v riadení systému. To znamená, že na základe aktuálnej situácie určíme budúce správanie v rámci predikčného horizontu, ale do úvahy zoberieme iba prvý krok a ostatné zahodíme. Podľa tohto kroku riešime novú optimalizáciu a celý postup opakujeme v každej perióde vzorkovania.

Veľkosť predikčného horizontu ovplyvňuje kvalitu riadenia ale aj presnosť predikcie. Ak zvolíme príliš veľkú hodnotu tohto parametra, predĺži to trvanie výpočtu akčného zásahu a zároveň sa nám môže odhad budúceho stavu výraznejšie odlišovať od skutočného. [4]

### 2.1.3 Účelová funkcia

Akčný zásah je určený optimalizáciou, minimalizáciou účelovej funkcie vzhľadom na obmedzenia. Účelovú funkciu si môžeme odvodiť nasledovne:

$$J = \int f(u)dt + \int g(w - y)dt, \quad (2.4)$$

jej hodnota je vyjadrená súčtom integrálov funkcií opisujúcich akčné zásahy a odchýlky výstupov od referencie  $w$ . Vyjadrením pomocou noriem získame:

$$J = \int \|u\|_p dt + \int \|w - y\|_p dt, \quad (2.5)$$

dosadením za  $p = 2$ , t.j. použitím Euklidovskej 2-normy a presunutím členov pod jeden integrál dostaneme vzťah:

$$J = \int (u^T u + (w - y)^T (w - y)) dt, \quad (2.6)$$

ktorý upravíme pridaním váhových matíc  $Q_u$  a  $Q_y$  s funkciou penalizačného faktora:

$$J = \int (u^T Q_u u + (w - y)^T Q_y (w - y)) dt. \quad (2.7)$$

Ako bolo vyššie spomenuté, pri MPC uvažujeme v diskretnom opise, preto spojitý integrál nahradíme sumou vo všetkých krokoch:

$$J = \sum_{k=0}^{N=\infty} (u(t+k)^T Q_u u(t+k) + (w(t+k) - y(t+k))^T Q_y (w(t+k) - y(t+k))), \quad (2.8)$$

a predikčný horizont  $N$  uvažujeme s konečnou hodnotou, preto môžeme účelovú funkciu vyjadriť vzťahom: [5]

$$J = \sum_{k=0}^{N-1} (u_k^T Q_u u_k + (w_k - y_k)^T Q_y (w_k - y_k)), \quad (2.9)$$

### 2.1.4 Ohraničenia

Ako bolo už vyššie spomenuté, veľkou výhodou MPC je schopnosť priamo pracovať s ohraničeniami vstupov, výstupov aj stavov. Je to jedným z dôvodov rozšírenia MPC

v priemysle, nakoľko sa v praxi vždy stretávame so zadanými obmedzeniami. Pre jednotlivé ohraničenia stavovej, riadiacej a riadenej veličiny môžeme použiť vzťahy:

$$\begin{aligned}\Delta \underline{x} &\leq \Delta x_k \leq \Delta \bar{x}, \\ \Delta \underline{u} &\leq \Delta u_k \leq \Delta \bar{u}, \\ \Delta \underline{y} &\leq \Delta y_k \leq \Delta \bar{y},\end{aligned}\tag{2.10}$$

kde vo všeobecnosti  $\Delta \underline{z}$  je dolná a  $\Delta \bar{z}$  horná hranica obmedzení prírastkov veličín. [3] Tieto ohraničenia nazývame tvrdými ohraničeniami, vychádzajú napríklad z fyzikálnych zákonov, avšak môže sa stať, že úloha sa stane tvrdými stavovými alebo výstupnými ohraničeniami neriešiteľnou. Z tohto dôvodu pracuje MPC najmä s mäkkými ohraničeniami. [6] Uvažujme napríklad ohraničenie na maximálne reregulovanie v tvare:

$$y_k - w_k \leq \varepsilon,\tag{2.11}$$

kde  $\varepsilon$  je maximálne dovolené prereregulovanie. Takéto ohraničenie však môže vplyvom dynamiky procesu viesť k neriešiteľnosti optimalizačného problému. Preto tento typ ohraničenia zmäksíme pridaním nezápornej konštanty, vo všeobecnosti  $s_k$ :

$$-s_k + \varepsilon \leq y_k - w_k \leq \varepsilon + s_k.\tag{2.12}$$

Pre váhovú maticu priradenú týmto ohraničeniam, označme  $Q_s$ , platí:

$$Q_s \gg Q_u, Q_y,\tag{2.13}$$

čo znamená, že porušenie ohraničení výrazne zvýši hodnotu účelovej funkcie. Tá po zohľadnení zmäksenia získava nasledovný tvar:

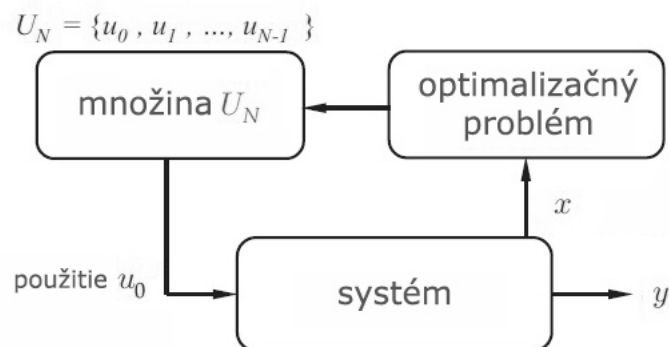
$$J = \sum_{k=0}^{N-1} (u_k^T Q_u u_k + (w_k - y_k)^T Q_y (w_k - y_k) + s_{x,k}^T Q_s s_{x,k} + s_{y,k}^T Q_s s_{y,k}),\tag{2.14}$$

kde matice  $Q_s$  môžu, aj nemusia byť rovnaké.

## 2.2 Princíp riadenia

Model riadeného systému formuluje v prediktívnom riadení ohraničený optimalizačný problém, ktorého úlohou je nájsť postupnosť optimálnych riadiacich vstupov tak, aby tieto minimalizovali účelovú funkciu a zároveň brali do úvahy stavové a vstupné ohraničenia.

MPC je realizované v uzavretej slučke horizontu predikcie, kde je optimalizačný problém riešený na základe aktuálnych stavových meraní, t.j. začiatočných podmienok. Pomocou tohto riešenia získame postupnosť takých akčných zásahov  $U_N = \{u_0, u_1, \dots, u_{N-1}\}$ , aby sme sa vedeli udržať čo najbližšie k referenčnej trajektórii. Z akčných zásahov následne implementujeme ako riadiaci signál iba prvý prvok  $u_0$  a ostatné vyradíme, pretože v nasledujúcom kroku opäť vykonáme aktuálne stavové meranie a s touto novou hodnotou formulujeme a riešime nový optimalizačný problém. Zo získaných akčných zásahov do riadenia implementujeme prvý prvok a takto opakujeme celý proces. Schéma opísaného postupu je znázornená na obrázku (Obr. 2.1). [1]



Obr. 2.1: Postup riadenia - riešenie optimalizačného problému na základe aktuálneho stavu, výpočet množiny akčných zásahov a následné použitie prvého prvku pri riadení systému

## Kapitola 3

# Flexibilný dopravný pás

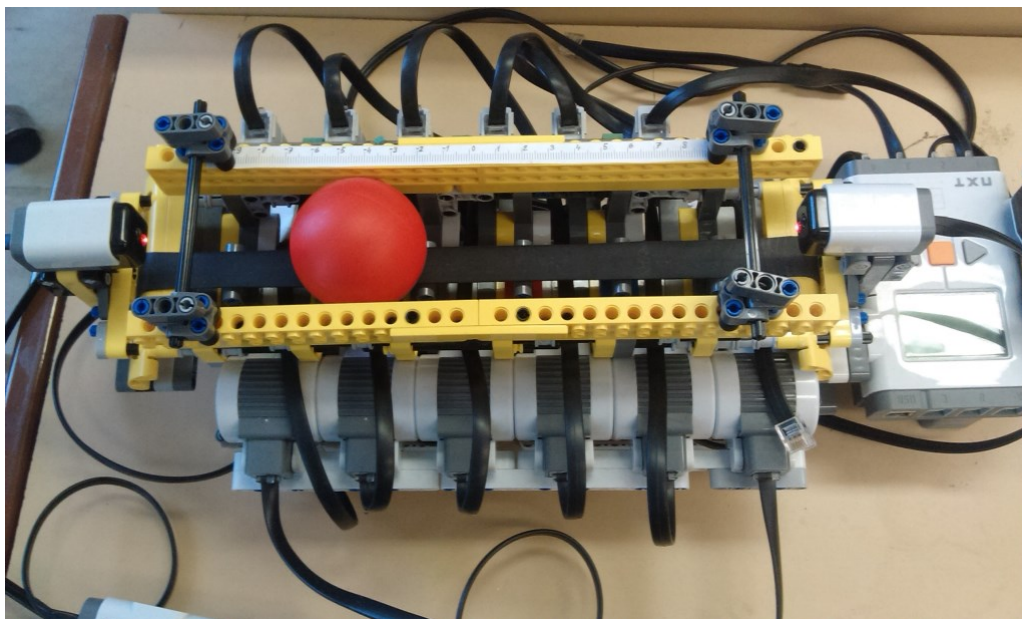
Konštrukcia zariadenia (Obr. 3.1) je vytvorená z lega a horizontálneho gumeného pásu, pod ktorým sa nachádza šesť piestov. Nimi na páse tvarujeme vlnu a tá presúva objekt, v našom prípade loptičku, na určenú polohu. Úlohou je navrhnúť také optimálne riadenie piestov, aby sa loptička dostala na žiadanú polohu za čo najkratší čas. Je to náročná úloha, pretože loptička nezostáva na jednom mieste, ale presúva sa so zrýchlením priamo úmerným sklonu pásu.

Na oboch stranách pásu sú umiestnené senzory, ktorých výstupy vieme prepočítať na aktuálnu pozíciu loptičky. Zdvih piestov je ovládaný motormi, do ktorých posielame informáciu o veľkosti akčného zásahu vo forme napätia. Motory sú pripojené na LEGO NXT kocku, ktorá komunikuje s Matlabom pomocou USB kábla. Na podporu tejto komunikácie sme si museli pred začiatkom práce so zariadením nainštalovať NXT Fantom Driver a RWTH - Mindstorms NXT Toolbox pre Matlab, ktorý slúži na riadenie LEGO robotov Matlabom cez USB kábel.

### 3.1 Fyzikálny opis

Ako bolo spomenuté v kapitole 2.1, pri návrhu riadenia je potrebné poznať model procesu, ktorý môžeme odvodiť pomocou fyzikálnych zákonitostí. [7] Najskôr si





Obr. 3.1: Zariadenie - loptička na páse: na krajoch dva senzory, pod gumeným pásom 6 piestov, pozdĺž pásu pravítko.

potrebujeme zdefinovať základné vzťahy, z ktorých budeme pri vytváraní modelu vychádzať. Majme loptičku s hmotnosťou  $m$  a polomerom  $r$ , jej pozíciu označme  $x$  a uhol prislúchajúci zdvihnutiu piestov  $\alpha$ . Sila pôsobiaca na loptičku je vyjadrená kombináciou trecej sily  $F_R$  a gravitačnej sily  $F_H$ , ktorú vieme rozpísať v tvare:

$$F_H = mgsin\alpha, \quad (3.1)$$

sila pôsobiaca na loptičku je potom:

$$F = F_H - F_R. \quad (3.2)$$

Ďalší spôsob vyjadrenia sily v smere naklonenia roviny:

$$F = m\ddot{x}, \quad (3.3)$$

kde  $\ddot{x}$  je zrýchlenie. Vzťah pre rýchlosť je vyjadrený:

$$\dot{x} = \omega r, \quad (3.4)$$

kde  $\omega$  je uhlová rýchlosť. Z rovnice (3.4) vieme odvodiť vzťah pre zrýchlenie:

$$\ddot{x} = \dot{\omega}r. \quad (3.5)$$

Moment zotrvačnosti  $J_K$ :

$$J_K = \frac{2}{5}mr^2, \quad (3.6)$$

moment sily je vyjadrený:

$$M = F_R r, \quad (3.7)$$

a pomocou predchádzajúcich vzťahov môžeme odvodiť rovnicu pre moment sily aj ako:

$$M = J_K \dot{\omega}. \quad (3.8)$$

Použitím vyššie odvodených vzťahov medzi jednotlivými silami dostávame:

$$\begin{aligned} F &= F_H - F_R = F_H - M \frac{1}{r}, \\ m\ddot{x} &= mg \sin \alpha - J_K \dot{\omega} \frac{1}{r}, \\ m\ddot{x} &= mg \sin \alpha - \frac{2}{5}mr^2 \ddot{x} \frac{1}{r} \frac{1}{r}, \end{aligned} \quad (3.9)$$

z čoho po jednoduchých matematických úpravách získame výsledný model systému

$$\ddot{x} = \frac{5}{7}g \sin \alpha. \quad (3.10)$$

Systém loptičky na páse považujeme za nelineárny, avšak pri malých hodnotách uhlov ho môžeme ľahko zlinearizovať. V kapitole 5.1, kde sú uvedené konkrétne rozmery nášho zariadenia, vidíme, že uhly, ktoré môžeme zdvihnutím piestov vytvoriť, sú pomerne malé, a preto pre ne platí vzťah:

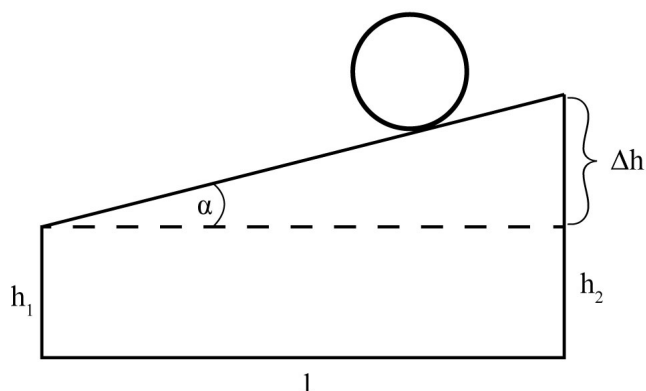
$$\sin \alpha = \alpha, \quad (3.11)$$

a po tejto linearizácii môžeme rovnicu (3.10) zapísať v tvare:

$$\ddot{x} = \frac{5}{7}g\alpha. \quad (3.12)$$

Nakoľko je naším akčným zásahom vysunutie piestov, model sme namiesto závislosti od uhla, ako je uvedené v rovnici (3.12), preformulovali ako závislosť od výšky piestov. Ako vidno na obrázku (Obr. 3.2), uhol  $\alpha$  vieme vyjadriť pomocou rozdielu vysunutia piestov  $\Delta h$  a vzdialenosti  $l$ :

$$\operatorname{tg} \alpha = \frac{\Delta h}{l}. \quad (3.13)$$



Obr. 3.2: Zdvihnutie krajných piestov

Pri zohľadnení toho, že pre malé uhly platí:

$$\operatorname{tg} \alpha = \alpha, \quad (3.14)$$

môžeme rovnicu (3.13) zapísať:

$$\alpha = \frac{\Delta h}{l}. \quad (3.15)$$

## 3.2 Matematický model

Stavový model, ktorý ideme odvodiť, predstavujú dva stavy (poloha loptičky  $p$  a jej rýchlosť  $v$ ) a dva akčné zásahy (výšky dvoch krajných piestov, označené ako  $h_1$  a  $h_2$ ). Maticový tvar modelu je zadaný vzhľadom na to, že v prvom kroku riadenia

vysunieme krajné piesty na 50% a  $\Delta h$  dosiahneme vysunutím, resp. zasunutím piestu o polovičnú hodnotu. To znamená, že akčné zásahy sú obrátené čísla v absolútnej hodnote rovnajúce sa  $\frac{\Delta h}{2}$ . Na výstupe môžeme sledovať oba stavy alebo len pozíciu loptičky. Podľa toho vytvoríme maticový tvar modelu:

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{10g}{7l} & -\frac{10g}{7l} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \quad (3.16)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p \\ v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

Do všeobecného tvaru stavového opisu:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du, \end{aligned} \quad (3.17)$$

potom dosadíme matice spojeného modelu v tvare:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ \frac{10g}{7l} & -\frac{10g}{7l} \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (3.18)$$

Tieto pri riadení musíme preformulovať do diskrétného stavu (naznačené v kapitole 4.1.2).

## Kapitola 4

# Simulácie

Prvým krokom pri riešení úlohy zadefinovanej v kapitole 3.2 bolo vytvoriť skripty a simulácie v prostredí MATLAB/Simulink. Pri práci sme použili toolbox YALMIP, ktorý užívateľovi umožňuje zjednodušený zápis optimalizačného problému, t.j. účelovej funkcie spolu s ohraňčeniami.

### 4.1 MPC vs. PID

V teoretickej časti práce sme vysvetlili výhody použitia MPC a tie sme si v MATLABE overili tak, že sme problém riešili okrem tohto spôsobu aj pomocou PID regulátora a výsledky sme navzájom porovnali. Všetky vytvorené a použité súbory a skripty sú kvôli prehľadnosti práce priložené na CD.

#### 4.1.1 Riešenie pomocou PID

Pri použití PID regulátora sme uvažovali situáciu, kedy akčným zásahom nie sú vysunutia krajných piestov ale náklon plochy, nakoľko je PID príklad SISO riadenia (Single Input Single Output - jeden vstup a jeden výstup) a PID a MPC sme chceli porovnať objektívne. Pri úlohe, ako sme ju odvodili v rovniciach (3.16) by sme museli

použiť dva regulátory na riadenie oboch krajných piestov, s ktorými sme pracovali. Použili sme preto matematický model vychádzajúci z rovnice (3.12).

Vytvorili sme si simulinkovú schému *ball\_model* (Obr. 4.1) so skriptom *ball\_script* a Level-2 S-funkciu *ball\_sfun*, ktorou sme získali na základe dvoch vstupov, a to veľkosti uhla (výstup z PID bloku) a meraného času (blok Clock), výstupy: pozíciu a rýchlosť loptičky, čo bolo cieľom nášho riadenia a ukladali sme aj hodnoty vysunutia piestov (*hout*) na základe zmeneného uhla. Cieľom bolo presunúť loptičku z pôvodnej pozície 0,5 cm s nulovou rýchlosťou (loptička bola v pokoji) na pozíciu 0 cm a v čase 12,5 s zmeniť referenciu na 1 cm. Veľkosti P, I a D zložky sme určili experimentálne. Pridanie D zložky spôsobovalo nestabilitu systému, preto sme jej hodnotu zvolili nulovú a ďalej pracovali s PI regulátorom. Konkrétne hodnoty zložiek boli:

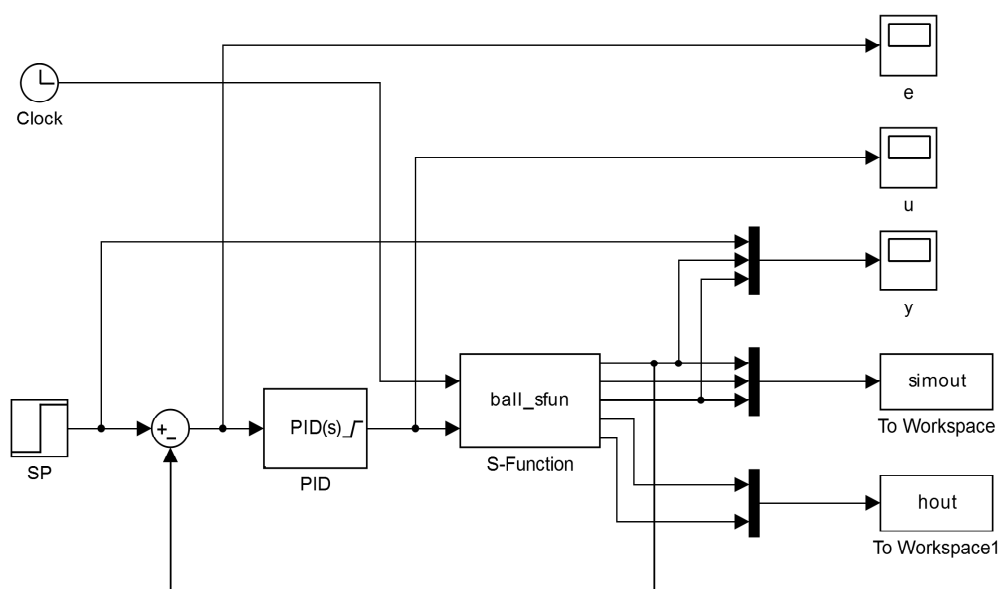
- $P = 0,2$
- $I = 0,1$
- $D = 0$ .

V schéme sme použili aj blok saturácie, potrebný pre ohraničenie akčných zásahov. Okrem vykreslenia vstupnej  $u$  a výstupných veličín  $y$ , sme vzhľadom na určenú referenciu (blok *SP – Setpoint*) vykreslovali aj regulačnú odchýlku  $e$ .

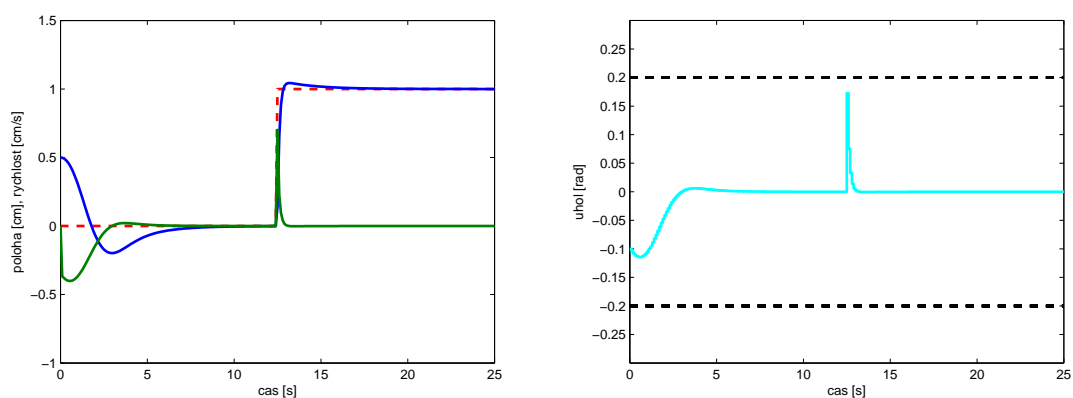
Získané výsledky, pozície, rýchlosť loptičky a akčné zásahy, sú vykreslené na obrázku 4.2. PI regulátor vykazuje pomerne pomalé sledovanie žiadanej hodnoty, čo je spôsobené jeho konzervatívnym naladením tak, aby spĺňal ohraničenia. Nemáme ale žiadnu záruku, že tieto ohraničenia budú dodržané vždy. Navyše chceme zrýchliť regulačný pochod. To nás motivuje použiť MPC.

### 4.1.2 Riešenie pomocou MPC

Ako sme uviedli na začiatku kapitoly, pri práci sme použili toolbox YALMIP, zapísali sme účelovú funkciu a ohraničenia všeobecne do súboru *opt\_mpc\_soft\_delta\_sim*



Obr. 4.1: Simulinková schéma znázorňujúca riadenie pomocou PI regulátora.



Obr. 4.2: Simulačné riadenie procesu pomocou PID regulátora: vľavo - žiadaná veľičina (červená), pozícia (modrá), rýchlosť (zelená); vpravo - zmena akčného zásahu (tyrkysová) a obmedzenie veľkosti akčného zásahu (čierna prerušovaná).

(ukážka 4.3). Nasledujúci postup je opisom skriptu s názvom *ball\_mpc\_uhol*. Pracujeme v diskretnom stave, takže matice stavového opisu sme najskôr zdiskretizovali a používali ich len v takom tvare. Na diskretizáciu sme použili v MATLABE príkaz

*c2d*, ktorý spojitý systém prevedie na diskretný. Pre výpočet optimálnych akčných zásahov sme skonštruovali optimizer - volali sme funkciu *opt\_mpc\_soft\_delta\_sim* s parametrami: zdiskretizované matice A, B, C, D, predikčný horizont  $N$ , váhové matice ( $Q_u$  a  $Q_y$ ) a ohraničenia vstupov ( $u_{\min}$ ,  $u_{\max}$ ), stavov ( $x_{\min}$ ,  $x_{\max}$ ) a výstupov ( $y_{\min}$ ,  $y_{\max}$ ). Určili sme veľkosť kroku  $\Delta t$ , v ktorom sme opakovali výpočet optimálneho akčného zásahu a prislúchajúcich výpočtov stavov. Úloha bola riešená s rovnakou referenciou, začiatočnými podmienkami a časom simulácie ako pri riadení s PI regulátorom.

Konkrétne hodnoty parametrov:

$$\begin{array}{ll}
 N = 5 & x_{\min} = [-2; -2] \\
 Q_u = 1 & x_{\max} = [2; 2] \\
 Q_y = 1 & y_{\min} = -2 \\
 u_{\min} = -5 & y_{\max} = 2 \\
 u_{\max} = 5 & \Delta t = 0, 1.
 \end{array}$$

```

obj = obj + u(:, k)'*Qu*u(:, k) + ...
      (w - y(:, k))'*Qy*(w - y(:, k)) + ...
      sx(:, k)'*Qs*sx(:, k) + ...
      sy(:, k)'*Qs*sy(:, k);

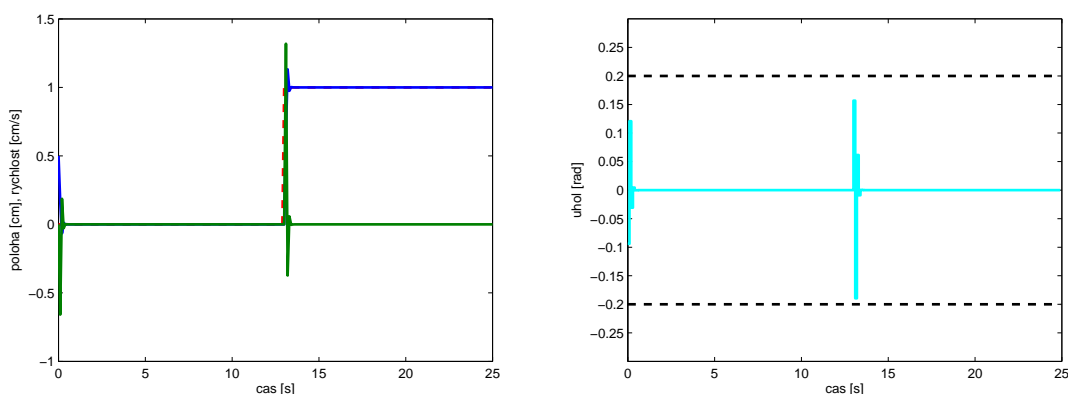
con = con + [ x(:, k+1) == A*x(:, k) + B*u(:, k) ];
con = [con; ...
      y(:, k) == C*x(:, k) + D*u(:, k); ...
      -sx(:, k) + xmin <= x(:, k) <= xmax + sx(:, k); ...
      umin <= u(:, k) <= umax; ...
      -sy(:, k) + ymin <= y(:, k) <= ymax + sy(:, k)];
con = con + [ dumin <= deltau <= dumax ];

```

Obr. 4.3: *Opt\_mpc\_soft\_delta\_sim*: Zápis účelovej funkcie a ohraničení.



Dosiahnuté výsledky, graf výstupných veličín - pozície a rýchlosti a graf zmeny akčných zásahov sú zobrazené na obrázku 4.4.



Obr. 4.4: Simulačné riadenie procesu pomocou MPC: vľavo - žiadaná veličina (červená), pozícia (modrá), rýchlosť (zelená); vpravo - zmena akčného zásahu (tyrkysová) a obmedzenie veľkosti akčného zásahu (čierna prerušovaná).

### 4.1.3 Porovnanie

Z výsledkov simulácie, ktoré vidíme na grafoch 4.2 a 4.4, t.j. z výstupov systému, môžeme povedať, že riadenie tohto systému pomocou MPC je lepšie. Poloha loptičky sa na žiadanú pozíciu dostane týmto spôsobom rýchlejšie a odchýlka od žiadanej veličiny je na celom časovom intervale menšia.

Metódy vieme porovnať aj presnejším a jasnejším spôsobom ako len grafickými výsledkami, a to výpočtom hodnoty účelovej funkcie (2.9), ktorú sa snažíme minimalizovať. Pre jednoduchý a ľahko použiteľný výpočet sme vytvorili funkciu *object\_fun* so vstupnými parametrami: maticou akčných zásahov, žiadanej pozície, reálnej pozície loptičky a váhových matíc. Funkcia je vytvorená pre ľubovoľný počet akčných zásahov, aby sme túto mohli použiť aj neskôr, keď budeme riadiť priamo

pomocou vysunutia piestov a nie pomocou uhla medzi nimi. To znamená, keď sa počet vstupných veličín zmení z jednej, na dve.

Hodnoty účelových funkcií opisujúcich správanie systému pri riadení s PI regulátorom (kap. 4.1.1) a pri prediktívnom riadení (kap. 4.1.2) boli nasledovné:

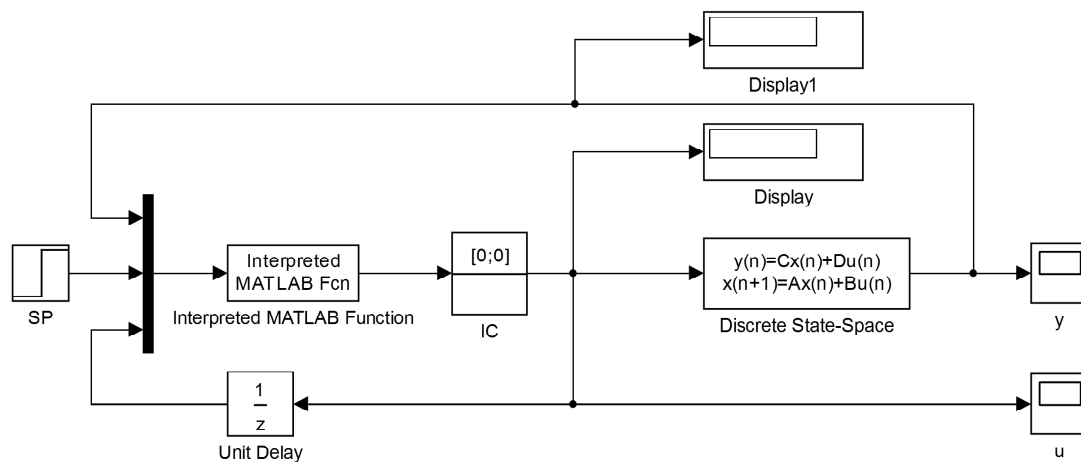
- hodnota účelovej funkcie (PI) = 4,1205
- hodnota účelovej funkcie (MPC) = 1,5397,

čím sme si potvrdili správnosť grafických výsledkov a predpoklad, že výhodnejšie je použiť MPC.

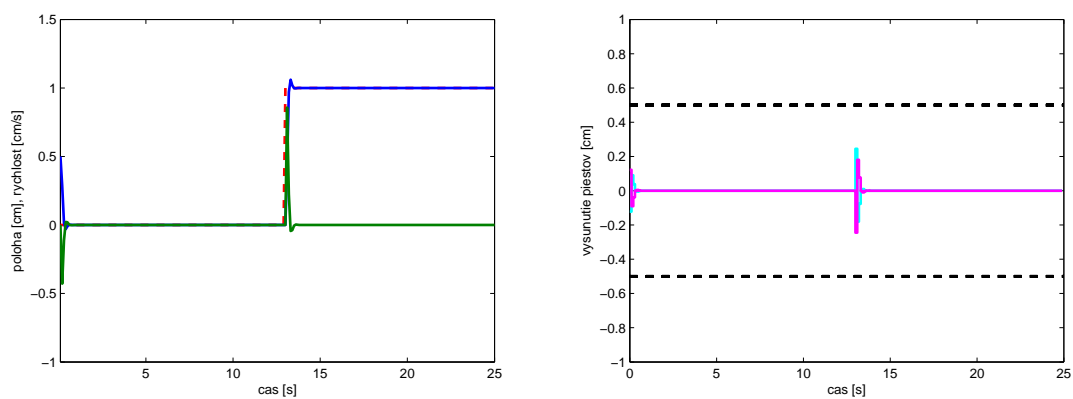
## 4.2 Riadenie pomocou MPC

Na základe predchádzajúcich výsledkov sme vytvorili skript aj schému v Simulinku a pomocou MPC sme riadili systém so vstupnými veličinami, ktorými boli vysunutia dvoch krajných piestov. V tomto prípade sme použili matice (3.18). Názvy súborov sú *ball\_mpc\_vysky* a *schema\_mpc\_vysky*. Prvý súbor sme vytvorili analogicky ako súbor *ball\_mpc\_uhol* opísaný v kap. 4.1.2 s rozdielom ohraničenia pre vstupné veličiny, ktorými sú minimálne a maximálne vysunuté piesty. Simulinková schéma je zobrazená na Obr. 4.5. V bloku *Interpreted MATLAB Function* prebieha optimalizácia, vstupmi do tohto bloku sú stavy (poloha a rýchlosť), referencia a akčné zásahy (výšky piestov). V bloku *Discrete State-Space* sú zadané odvodené matice po diskretizácii.

Výsledky získané oboma spôsobmi (Obr. 4.6) sú rovnaké, čím sme si potvrdili správnosť návrhu simulinkovej schémy. Hodnota účelovej funkcie je rovná 2,1562, ale nakoľko priebeh zmeny stavov je plynulejší, to znamená pri zmene referencie dochádza k menšiemu preregulovaniu ako pri riadení pomocou uhlu, rozhodli sme sa pri riadení reálneho zariadenia za vstupné veličiny považovať výšky krajných piestov.



Obr. 4.5: Simulinková schéma znázorňujúca riadenie pomocou MPC.



Obr. 4.6: Simulačné riadenie procesu pomocou MPC - výšky piestov: vľavo - referencia (červená), pozícia (modrá), rýchlosť (zelená); vpravo - výšky piestov (tyrkysová a fialová), obmedzenie akčného zásahu (čierna prerušovaná).

## Kapitola 5

# Riadenie reálneho procesu

Predtým, ako sme mohli MPC aplikovať na reálny systém loptičky na flexibilnom páse, museli sme zariadenie prepojiť s MATLAB-om. Na komunikáciu medzi počítačom a zariadením slúžila LEGO NXT kocka.

Pre prácu bolo potrebné vedieť prepočítať údaje, získavané zo senzorov pripnutých na okrajoch pása, na aktuálnu polohu loptičky na páse v centimetroch a údaje, posielané do motorčeka, ktorý podľa vypočítaných akčných zásahov pohyboval s piestami.

### 5.1 Postup riadenia

Postup a spôsob riadenia bol založený na rovnakom princípe ako pri simuláciách opísaných v predchádzajúcej kapitole (4.1.2).

Prvým krokom bolo zistenie a zadefinovanie konkrétnych parametrov systému, a to:

- gravitačné zrýchlenie  $g = 9,81 \text{ m.s}^{-1}$
- dĺžka pása  $l = 19,7 \text{ cm}$
- maximálne vysunutie piestov  $h = 0,96 \text{ cm}$ .

Všetky veličiny sme používali v jednotkách SI sústavy, to znamená, že všetky dĺžky

sme prepočítali na metre.

Následne sme si, tak ako aj v simuláciách, zadefinovali matice spojitého modelu (3.18) a zdiskretizovali ich, určili sme žiadanú veličinu (rýchlosť bola vždy nulová, nakoľko sme chceli, aby sa loptička ustálila a polohu sme určovali ľubovoľnú), periódu vzorkovania, predikčný horizont a váhové matice vstupov aj oboch stavov. Tieto sme následne experimentálne zmenili. Keďže naším cieľom bolo dostať loptičku na žiadanú pozíciu, váhová matica prislúchajúca odchýlke od žiadanej polohy bola rádovo väčšia ako váha pri rýchlosti a najmenšiu váhu sme priradili akčným zásahom. Dôležitou časťou bolo zadefinovať si ohraničenia.

- Ohraničenia pre vstupy vychádzajú z možnej dĺžky vysunutia piestov. V našom modeli uvažujeme začiatkové vysunutie piestov na polovičnú hodnotu, takže interval obmedzení je  $\left[-\frac{0.96}{2}, \frac{0.96}{2}\right]$ .
- Ohraničenia pre stavy závisia od dĺžky pásu, ktorého stred sme priradili hodnotu nula. Ohraničenia sme určili menšie, ako je celková dĺžka pásu, pretože pri krajoch sú tyčky, ktoré zabraňujú loptičke opustiť pás a my chceme systém riadiť tak, že bude schopný zastaviť loptičku sám, bez nárazu. Preto je interval ohraničenia polohy  $[-7, 7]$ . Rýchlosť sme neobmedzili, mohla nadobudnúť akúkoľvek hodnotu, ktorú jej fyzika dovolila s tým, že jej hodnoty sme uvažovali kladné jedným smerom a záporné opačným.
- Ohraničenia výstupov sú rovnaké ako stavov.
- V neposlednom rade sme určili aj mäkké ohraničenia, ktorých dôležitú úlohu sme vysvetlili v teoretickej časti (2.1.4).

Pomocou všetkých vyššie uvedených parametrov sme skonštruovali optimizer.

V tomto bode sme sa rozhodli riešiť úlohu dvoma spôsobmi. Prvým bolo pokračovanie riešenia v skripte (kap. 5.1.1) a druhým vytvorenie simulácie v Simulinku (kap. 5.1.2).

### 5.1.1 Riadenie cez m-file

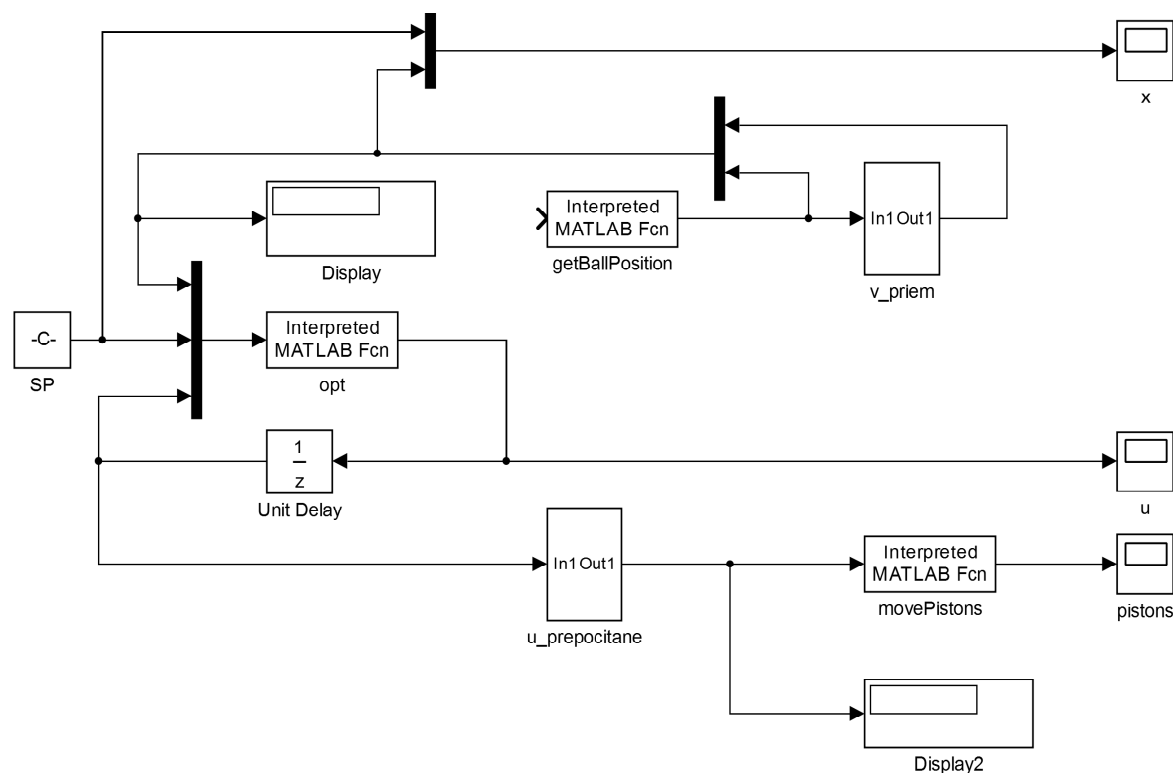
Poslednou časťou skriptu bolo vytvorenie cyklu, v ktorom sa v každom kroku počítal optimálny akčný zásah. Pre jeho výpočet bolo potrebné určovať aktuálnu polohu, čo nám umožnili senzory. My sme okrem sledovania aktuálnej polohy, ukladali aj predchádzajúcu a sledovali čas, za ktorý sa loptička presunula medzi týmito miestami. Z týchto údajov sme mohli vypočítať rýchlosť a tým sme získali celkom presné informácie a stavoch systému, ktoré sme použili pri výpočte nasledujúceho optimálneho akčného zásahu. Tento sme vypočítali v metroch, ale museli sme ho prepočítať na percentuálnu hodnotu vysunutia piestov, ktorú sme poslali do motorčeka a ten už preposlal vstupný signál a piesty sa vysunuli do optimálnej výšky.

Pre neskoršie potreby sme si všetky vypočítané akčné zásahy, stavy a čas ukladali počas behu programu do štruktúr.

Všetky vyššie uvedené výpočty a postupy sme vykonávali v jednom skripte (*mpc\_real\_vysky*).

### 5.1.2 Riadenie vytvorením schémy v Simulinku

Vytvorili sme schému s názvom *real\_schema\_sim* (zobrazená na Obr. 5.1). Tak, ako pri simuláciách, žiadaná veličina (blok *SP*) spolu so stavmi  $x$  - polohou a rýchlosťou a akčnými zásahmi  $u$  vstupuje do bloku *opt*, v ktorom sa v každom kroku optimalizuje nasledujúci akčný zásah. Rozdiel v porovnaní so simuláciou (Obr. 4.5) je v pridaní blokov *getBallPosition* a *movePistons*, ktoré komunikujú priamo so zariadením. Prvý blok zisťuje aktuálnu polohu loptičky a táto pokračuje do zloženého bloku, kde je uložená aj predchádzajúca pozícia a spolu s časom z nich rátame rýchlosť loptičky. Druhý spomenutý blok posiela prepočítaný optimálny akčný zásah do motora.



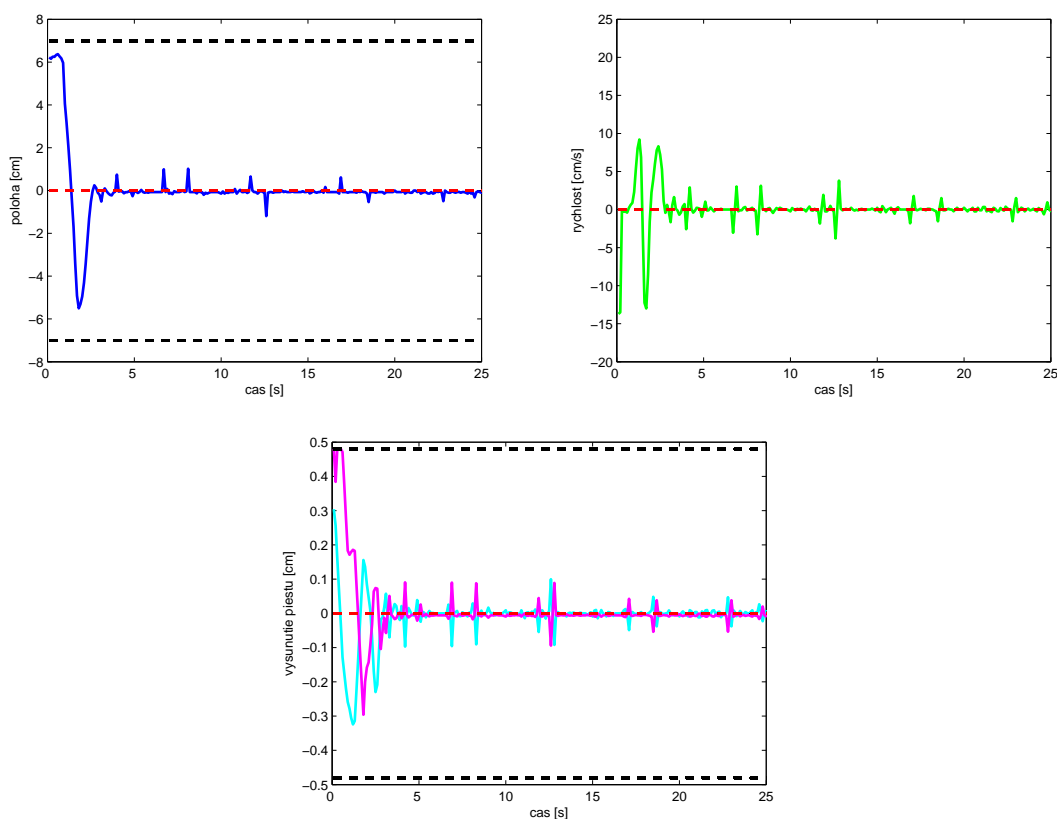
Obr. 5.1: Simulinková schéma znázorňujúca riadenie reálneho zariadenia pomocou MPC.

## 5.2 Výsledky

Oboma vyššie spomenutými spôsobmi sme pri riadení systému loptička na páse získali podobné výsledky. Na nasledujúcich grafoch sú uvedené pozície a zmeny rýchlosti loptičky pri rôznych referenciách a ohraničeníach polohy a k nim prislúchajúce akčné zásahy.

Keďže polohu a teda aj rýchlosť sme určovali pomocou senzorov, každá nepresnosť v meraní sa odzrkadlila na kmitaní výstupných signálov. Senzor meral loptičku iba na malej ploche, a preto aj jej mierny vertikálny pohyb hore dolu na mieste,

zapríčinený inak nevýrazným zdvíhaním piestov, spôsobil pre senzor mylnú informáciu, že loptička sa hýbe. Takýto výstup vidíme na Obr. 5.2.

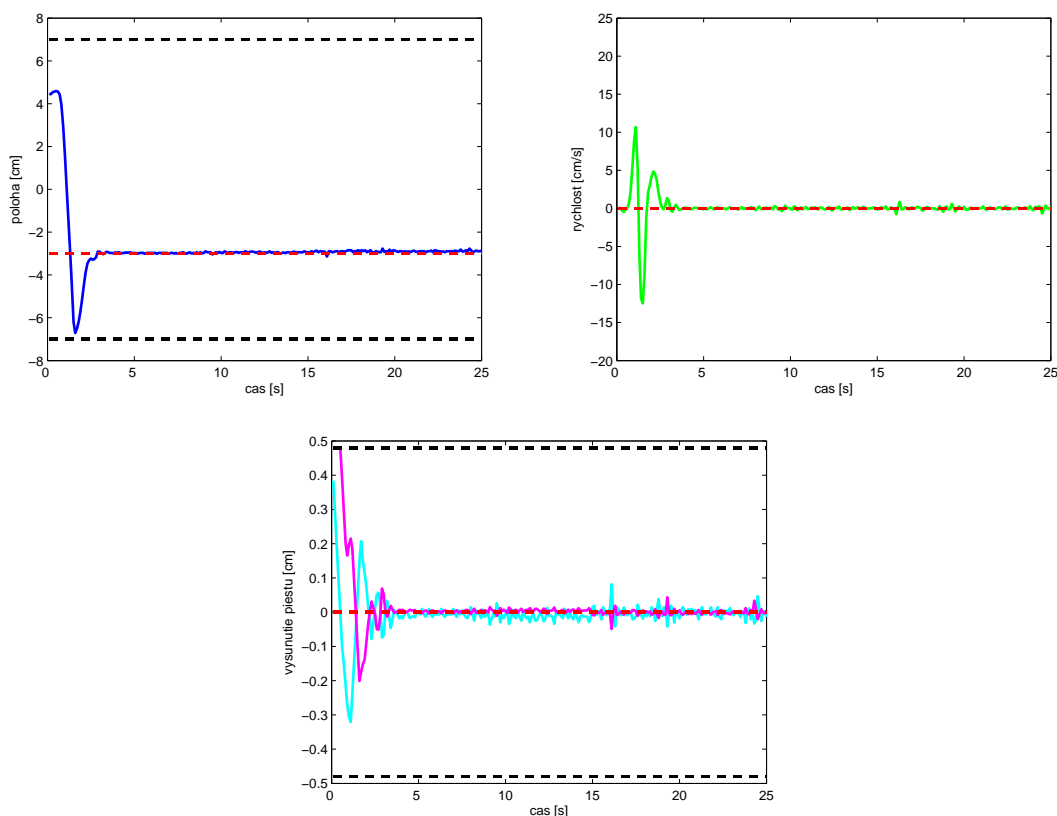


Obr. 5.2: Obr. hore: zmena polohy (modrá) a rýchlosti (zelená) loptičky pri všeobecných ohraničeniach polohy  $[-7, 7]$  (čierna prerušovaná) bez použitia filtra, referencia polohy a rýchlosti (červená prerušovaná). Obr. dole: vysunutie krajných piestov (tyrkysová, fialová). Hodnota účelovej funkcie je 112,7392.

Z tohto dôvodu sme použili pri riadení filter 1. rádu, čo znamená, že sme dve po sebe nasledujúce hodnoty rýchlosti spriemerovali. Príklad riadenia s použitím filtra je vidieť napr. na Obr. 5.3 a 5.4. Rozdiel medzi týmito úlohami je v ohraničeniach rýchlosti. Na Obr. 5.3 je graf úlohy s veľkými ohraničeniami rýchlosti, čo znamená, že rýchlosť prakticky neobmedzujú, preto sa smerom k referencii loptička rýchlo

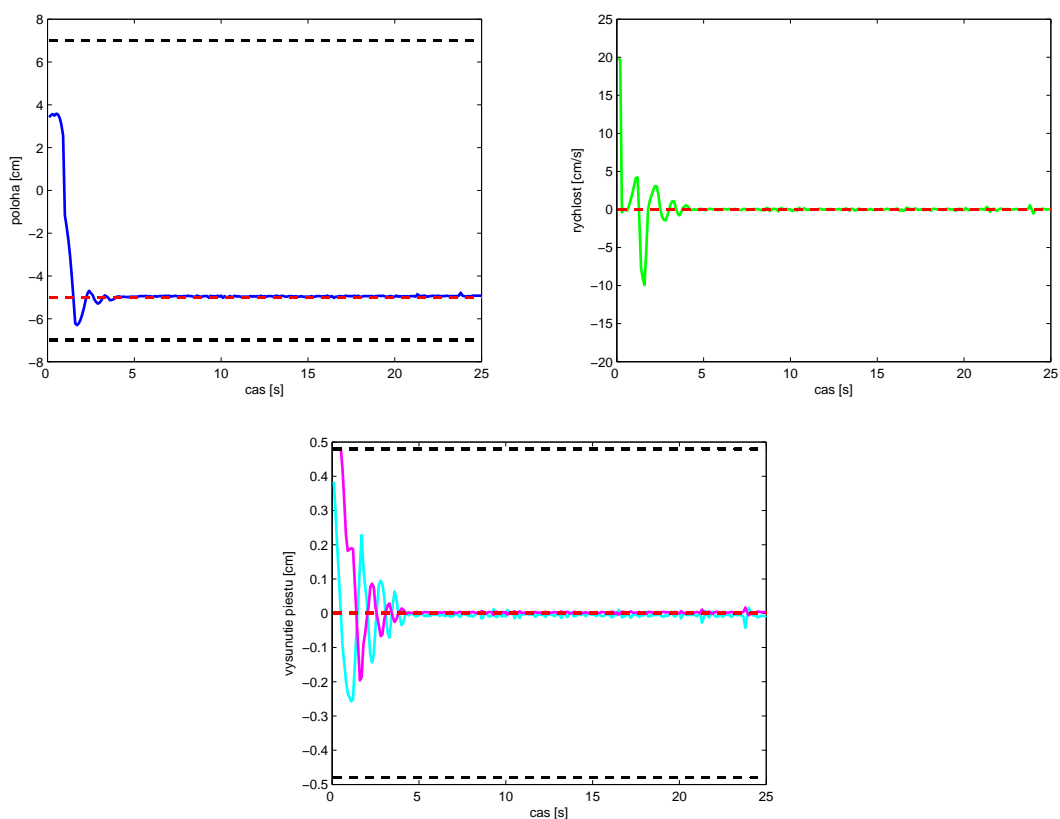


rozbehne a zastaví sa iba krátku vzdialenosť od ohraničenia polohy. Na Obr. 5.4 vidíme, že rýchlosť nadobúda menšie hodnoty, pretože jej ohraničenia boli zadané konkrétnejšie a teda mali vplyv na pohyb loptičky, ktorá sa im musela prispôbovať.



Obr. 5.3: Obr. hore: zmena polohy (modrá) a rýchlosti (zelená) loptičky pri všeobecných ohraničeniach polohy  $[-7, 7]$  (čierna prerušovaná) s použitím filtra 1. rádu, referencia polohy a rýchlosti (červená prerušovaná). Obr. dole: vysunutie krajných piestov (tyrkysová, fialová). Hodnota účelovej funkcie je 96,6195.

Na Obr. 5.5 môžeme vidieť správanie systému pri sprísnení ohraničení pre polohu loptičky. Systém sa snažil udržať loptičku v povolenom intervale vzdialenosti na páse, čo malo sa následok jej kmitanie okolo žiadanej veličiny a neskôršie ustálenie. Ale aj napriek malému priestoru, v ktorom sa mohla loptička pohybovať, sa

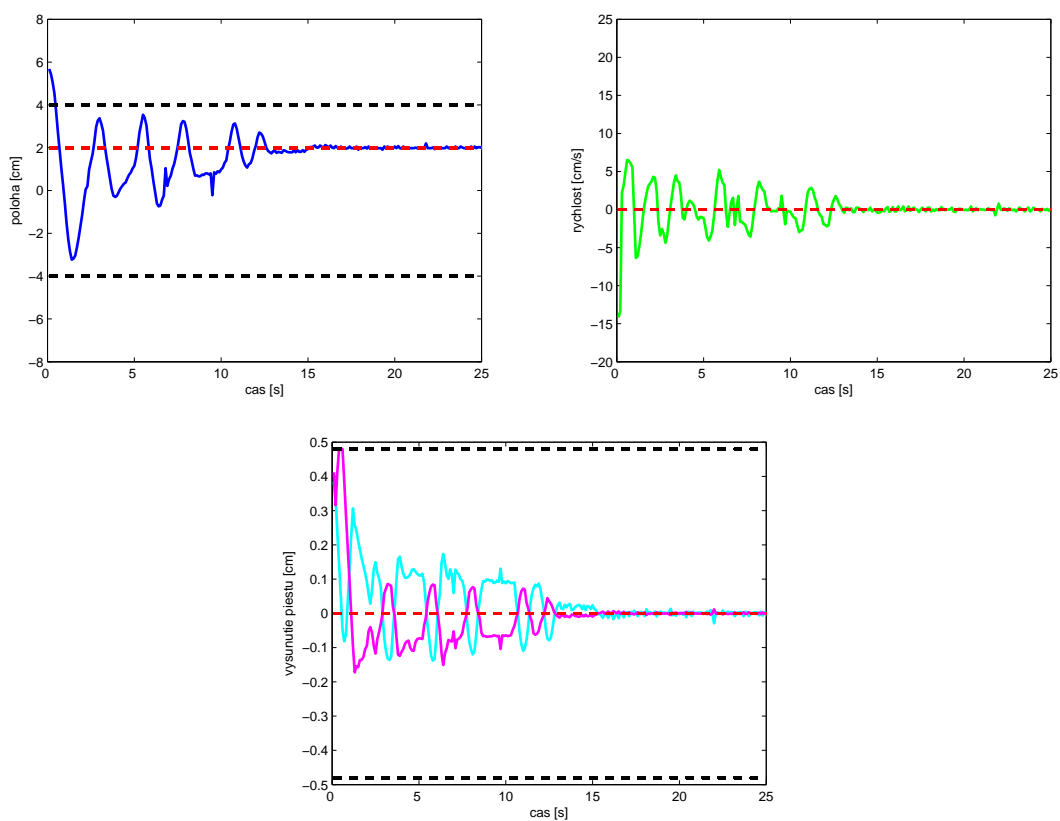


Obr. 5.4: Obr. hore: zmena polohy (modrá) a rýchlosti (zelená) loptičky pri všeobecných ohraničeniach polohy  $[-7, 7]$  (čierna prerušovaná) s použitím filtra 1. rádu, referencia polohy a rýchlosti (červená prerušovaná). Obr. dole: vysunutie krajných piestov (tyrkysová, fialová). Hodnota účelovej funkcie je 84,8251.

dokázala ustáliť na žiadanej hodnote.

Pri všetkých obrázkoch sú uvedené aj hodnoty účelových funkcií pre každý prípad a tieto potvrdzujú náš opis grafov, čiže opis správania sa loptičky na páse. Hodnota účelovej funkcie systému riadeného bez použitia filtra je vyššia ako s jeho použitím. Podľa konkrétneho systému môže byť použitie ohraničení prínosom, ako v prípade obmedzenia rýchlosti, ale ak použijeme príliš striktné ohraničenia, systém sa s nimi môže vysporiadať až v dlhšom časovom intervale, poprípade začne okolo

referencie kmitať, ale má problém sa hneď ustáliť, čo je v našom prípade príklad s ohraničením polohy.



Obr. 5.5: Obr. hore: zmena polohy (modrá) a rýchlosti (zelená) loptičky pri ohraňeniach polohy  $[-4, 4]$  (čierna prerušovaná), referencia polohy a rýchlosti (červená prerušovaná). Obr. dole: vysunutie krajných piestov (tyrkysová, fialová). Hodnota účelovej funkcie je 131,5578.

## Kapitola 6

### Záver

Pomocou základných fyzikálnych vzťahov a rovníc sme odvodili matematický model dopravníka pre vstupnú veličinu predstavujúcu v prvom prípade uhol a v druhom dva krajné piesty, ktoré vysunutím vytvárali požadovaný sklon pásu, po ktorom sa pohybovala loptička. Odvodený model sme potrebovali pri riadení tohto systému.

Systém pohybujúcej sa loptičky na dopravnom páse má ohraničenia stavov aj vstupov, čo bol hlavný dôvod, prečo sme sa rozhodli na riadenie použiť MPC. Navrhli sme prediktívny regulátor, ktorý riešil úlohu optimalizácie vysunutia krajných piestov s cieľom dopraviť loptičku na žiadanú pozíciu pri zohľadnení ohraničení vstupov, stavov a výstupov. Túto úlohu sme riešili simulačne v prostredí MATLAB/Simulink, kde sme MPC porovnali s klasickým PID regulátorom. Ukázalo sa, že pre tento proces sa jednoznačne oplatí použiť prediktívny regulátor. S týmto výsledkom sme regulátor pripojili aj na reálne zariadenie a experimentálne sme ukázali, že je vhodný na riadenie flexibilného dopravného pásu s loptičkou, pretože sa loptička vždy v krátkom časovom intervale premiestnila na referenciu a zároveň pri pohybe dodržala aj zadané ohraničenia.

Možnosť vytvorenia prediktívneho regulátora, ktorý by ovládal všetky piesty naraz, sme po zoznámení sa so zariadením nateraz vylúčili, pretože sa vyskytlo niekoľko problémov s touto myšlienkou. Prvým je, že použitá loptička nemá dokonalý

tvar gule, a preto sa nespráva tak, ako opisuje model. Ďalším problémom je gumený pás, ktorý používaním strácal svoje vlastnosti, pretože sa na ňom medzi piestami začali vytvárať malé priehlbiny a vyšúchané časti, čo zvyšovalo trenie, a teda rovnako spôsobovalo rozdiely správania reálneho zariadenia od modelu.

Namiesto tejto úlohy sme sa viac zamerali na optimalizáciu akčného zásahu a výpočty s ňou spojené. Vytvorili sme odhad stavu, rýchlosti, na základe sledovanej pozície a pomocou tohto odhadu sme získali optimálnejšie akčné zásahy.

Ďalšou úlohou, ktorú sme pridali do práce na základe správania sa systému, bolo pridanie filtra 1. rádu. Dôvodom bolo, že niekedy trenie spôsobovalo zastavenie loptičky na minimálnu vzdialenosť od referencie, pretože sa kvôli nemu nepresúvala plynulo. V tomto prípade sa ale piesty neprestali pohybovať, pretože im bol stále posielaný nový vypočítaný optimálny akčný zásah a ich pohyb spôsobil, že sa loptička mierne hýbala na páse vo vertikálnom smere. Tento minimálny pohyb zachytávali senzory a posielali informácie o zmene polohy, preto výstupy výrazne kmitali okolo referencie. Spriemerovaním posledných hodnôt výsledkov, t.j. pridaním filtra 1. rádu, sme tento problém vyriešili.

# Literatúra

- [1] KVASNICA, Michal: *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools*. Saarbrücken: VDM Verlag Dr. Müller, 2009.
- [2] MORRISON, Don: *Addressing the Myths of Model Predictive Control (MPC)* [online]. Posledná úprava 2010-10-02. [cit. 2014-05-19]. Dostupné na internete: <http://www.automation.com/automation-news/article/addressing-the-myths-of-model-predictive-control-mpc>
- [3] KARAS, Adrián, ROHAL-ILKIV, Boris, BELAVÝ, Cyril: *Praktické aspekty prediktívneho riadenia*. Bratislava: Vyd. STU, 2007.
- [4] VESELÝ, Vojtech, ROSINOVÁ Danica: *Robust Model Predictive Control Design* [online], Model Predictive Control, Tao Zheng (Ed.). Posledná úprava 2010-08-18. [cit. 2014-05-19]. Dostupné na internete: <http://www.intechopen.com/books/model-predictive-control/robust-model-predictive-control-design>
- [5] BEMPORAD, Alberto, MORARI, Manfred: *Robust Model Predictive Control: A Survey* [online]. [cit. 2014-05-19]. Dostupné na internete: <http://www8.cs.umu.se/research/ifor/dl/survey-robust-mpc.pdf>
- [6] KERRIGAN, Eric C., MACIEJOWSKI, Jan M.: *Soft Constraints and Exact Penalty Functions in Model Predictive Control* [online]. [cit. 2014-05-19]. Dostupné

na internete: [http://www-control.eng.cam.ac.uk/Homepage/papers/cued\\_control\\_53.pdf](http://www-control.eng.cam.ac.uk/Homepage/papers/cued_control_53.pdf)

- [7] FEYNMAN, Richard P.: *The Feynman Lectures on Physics, Vol. I: The New Millennium Edition: Mainly Mechanics, Radiation, and Heat (Volume 1)*. Basic Books; First Trade Paper Edition edition, 2011.