

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA APLIKOVANÝCH VĚD

KATEDRA KYBERNETIKY

BAKALÁŘSKÁ PRÁCE

Prediktivní regulátor pro řízení pohybu
elektromechanických soustav

Plzeň, 2021

David Wébr

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2020/2021

ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **David WÉBR**
Osobní číslo: **A18B0558P**
Studijní program: **B3918 Aplikované vědy a informatika**
Studijní obor: **Kybernetika a řídicí technika**
Téma práce: **Prediktivní regulátor pro řízení pohybu elektromechanických soustav**
Zadávací katedra: **Katedra kybernetiky**

Zásady pro vypracování

1. Seznamte se s technikami řízení elektrických pohonů a metodou model-based predictive control.
2. Navrhněte vhodný algoritmus prediktivního řízení pro pohon s pružnou zátěží.
3. Proveďte porovnání navrženého algoritmu s konvenční PID kaskádní strukturou řízení a LQR/LQG regulátorem.
4. Ověřte navržené metody na reálné mechanické soustavě.

Rozsah bakalářské práce: **30-40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

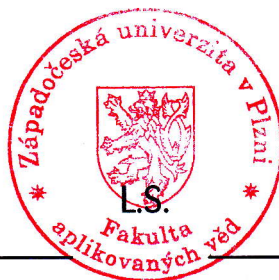
Rossiter, J.A., Model-based predictive control
Mikuláš, O., Quadratic Programming Algorithms for Fast Model-based Predictive Control

Vedoucí bakalářské práce: **Ing. Martin Goubej, Ph.D.**
Katedra kybernetiky

Datum zadání bakalářské práce: **15. října 2020**
Termín odevzdání bakalářské práce: **24. května 2021**

Radová

Doc. Dr. Ing. Vlasta Radová
děkanka



J. Psutka

Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 24. května 2021

A handwritten signature in blue ink that reads "David Wébr". The signature is written in a cursive style and is positioned above a horizontal dotted line.

David Wébr

Poděkování

Tímto bych chtěl poděkovat panu Ing. Martinu Gubejovi Ph.D., za skvělé vedení mé práce, cenné profesionální rady a připomínky. Bez jeho výborných znalostí a zkušeností v dané oblasti by tahle práce nebyla taková, jaká je. Dále dovoluji vyjádřit díky mé rodině a partnerce za to, že mi byli oporou během celého mého studia. Zejména pak v posledních měsících kdy jsem trávil více času nad rovnicemi než společně s nimi.

Abstrakt

Práce se zabývá studiem prediktivního řízení a to zejména jeho algoritmem a jeho praktickým použitím k regulaci systémů. V počátku práce je podrobně popsán princip činnosti prediktivní regulace. Je zde rozebráno, jak se predikuje chování systému, jak probíhá hledání optimálního akčního zásahu, jaká lze klást omezení na regulační smyčku prostřednictvím MPC regulátoru nebo jak lze do algoritmu zahrnout kompenzaci poruch. Další část práce je věnována algoritmům prediktivního řízení. Zde budou ukázány 3 možné způsoby implementace algoritmu pro MPC kompenzátory. V praktické části bakalářské práce se bude využívat algoritmus prediktivního řízení, který byl vytvořen jako praktická část. Prostřednictvím něho se pro 3 elektromechanické systémy s pružnou zátěží navrhne prediktivní regulátor polohy. Současně bude provedena diskuze nad dosaženou robustností ve stabilitě a nad časovou náročností kvadratické optimalizace. Následně proběhne srovnání výsledků při regulaci navrženým MPC kompenzátorem, kaskádním P-PI regulátorem a LQR. Jako poslední bude v práci odzkoušen navržený prediktivní algoritmus na řízení modelu reálného systému.

Klíčová slova

prediktivní řízení, řízení na základě modelu, MPC, regulace, regulátor, model, systém, stavový model systému, optimalizace, omezení, kaskádní P-PI regulace, LQR

Abstract

The main aim of this thesis is to study the predictive control. It focuses on its algorithm and its comparison with other commonly used control strategies. At the beginning of this thesis it is shown, how the prediction of future system behavior is made, how to find the optimal control action, how to handle constraints or how to properly handle disturbance. In the next section 3 possible options are presented, how to implement predictive control algorithm. In the next chapter will be used the algorithm of MPC for controlling three electromechanical systems. This algorithm was developed as a practical part of this bachelor thesis. Then, it will be discussed robustness in stability and duration of quadratic optimization. In the next section created predictive controller will be compared with other commonly used regulators as cascade compensator and LQR. In the end of this thesis developed algorithm for predictive control will be tested on the model of the real system.

Key words

predictive control, model-based predictive control, MPC, regulation, regulator, model, system, state-space model, optimization, constraints, cascade P-PI control, LQR

Obsah

1 Úvod	10
1.1 Dějinný kontext	11
1.2 Důvody použití MPC	11
1.3 Přehled použitých symbolů	11
2 Prediktivní řízení	13
2.1 Algoritmus MPC	13
2.1.1 Predikční matice	14
2.1.2 Horizont predikcí n_p a horizont řízení n_c	15
2.1.3 Technika move blocking	16
2.1.4 Účelová funkce a optimalizace	17
2.2 Omezení	19
2.2.1 Tvrdá omezení	19
2.2.2 Měkká omezení	20
2.3 Regulace poruch	21
2.4 Výhody a nevýhody prediktivního řízení	22
3 Algoritmy prediktivního řízení	24
3.1 Problém regulace	24
3.2 Sledování konstantní referenční trajektorie	25
3.2.1 Matice optimalizace \mathbf{G} a \mathbf{f}^T	26
3.3 Nulová odchylka v ustáleném stavu	27
3.3.1 Matice optimalizace \mathbf{G} a \mathbf{f}^T	29
3.4 MPC regulátor bez omezení	30
4 Návrh prediktivního řízení	33
4.1 Popis řízených systémů	33
4.2 Schéma v Simulinku použité pro otestování	39
4.3 Návrh MPC regulátoru pro systém s poměrem rezonancí r_1	39
4.3.1 Průběh regulace	40
4.3.2 Robustnost ve stabilitě	42
4.3.3 Časová náročnost solveru <i>quadprog</i> při řízení pružného a pevného systému	44
4.3.4 Časová náročnost optimalizační úlohy při použití solveru <i>qpOASES</i>	45
4.3.5 Ověření robustnosti regulátoru	46
4.4 Návrh MPC regulátoru pro systém s poměrem rezonancí r_2	50

4.4.1	Průběh regulace	50
4.4.2	Robustnost ve stabilitě	52
4.4.3	Časová náročnost solveru <i>quadprog</i> při řízení pružného a pevného systému	54
4.4.4	Časová náročnost optimalizační úlohy při použití solveru <i>qpOASES</i>	55
4.4.5	Ověření robustnosti regulátoru	57
4.5	Návrh MPC regulátoru pro systém s poměrem rezonancí r_3	60
4.5.1	Průběh regulace	61
4.5.2	Robustnost ve stabilitě	62
4.5.3	Časová náročnost solveru <i>quadprog</i> při řízení pružného a pevného systému	65
4.5.4	Časová náročnost optimalizační úlohy při použití solveru <i>qpOASES</i>	66
4.5.5	Ověření robustnosti regulátoru	67
5	Srovnání MPC s konvenčními regulátory	71
5.1	Srovnání MPC s kaskádní P-PI regulací	71
5.1.1	Schéma v Simulinku použité pro otestování	71
5.1.2	MPC vs. kaskádní regulace systému s poměrem rezonancí r_1	72
5.1.3	MPC vs. kaskádní regulace systému s poměrem rezonancí r_2	74
5.1.4	MPC vs. kaskádní regulace systému s poměrem rezonancí r_3	76
5.2	Srovnání MPC s regulací pomocí LQR	78
5.2.1	Schéma v Simulinku použité pro otestování	78
5.2.2	MPC vs. LQR regulace systému s poměrem rezonancí r_1	79
5.2.3	MPC vs. LQR regulace systému s poměrem rezonancí r_2	81
5.2.4	MPC vs. LQR regulace systému s poměrem rezonancí r_3	83
5.3	Srovnání MPC s kaskádní P-PI regulací - saturace aktuátoru	85
5.3.1	Návrh regulátorů	85
5.3.2	Kaskádní regulace bez ošetření unášení integrátoru	86
5.3.3	Kaskádní regulace s ošetřením unášení integrátoru	86
5.3.4	Regulace pomocí MPC	87
5.3.5	Srovnání výsledků	88
6	Prediktivní regulace reálného systému	90
6.1	Popis řízené soustavy	90
6.2	Návrh MPC regulátoru	91
6.3	Otestování průběhu regulace	92
6.3.1	Časová náročnost optimalizace při regulaci	93

6.4	Návrh MPC s delší periodou vzorkování	94
6.4.1	Návrh parametrů MPC pro systém s periodou vzorkování 1 <i>ms</i>	94
6.4.2	Otestování průběhu regulace	95
6.4.3	Diskuze nad časovou náročností optimalizace	97
7	Závěr	98
	Seznam literatury a informačních zdrojů	99
	Seznam obrázků	100

1 Úvod

Prediktivní řízení je moderní přístup k regulaci v oblasti automatizace. V angličtině se tato strategie řízení označuje jako *Model-based predictive control* (MPC), z čehož již lze lépe vyvodit, na čem je tato metoda založena. Termín by se dal přeložit jako *Řízení na základě modelu systému*. Je tedy možné odvodit, že tato strategie řízení k regulaci využívá model řízené soustavy.

Na základě modelu řízené soustavy se totiž v každém kroku MPC algoritmu vypočítává (predikuje), jak se systém bude v budoucnu chovat. Tato predikce se následně využije k řešení optimalizační úlohy, kdy se hledají ideální hodnoty akčních zásahů generovaných regulátorem tak, aby bylo dosaženo požadovaného výsledku. První hodnota nalezeného optimálního akčního zásahu se posléze aplikuje do řízeného systému, který se dostane do nového stavu. Tento postup se opakuje v každém kroku algoritmu.

Dalo by se říci, že prediktivní řízení v některých případech reflektuje lidské chování. Jsou to situace, kdy je hledán takový způsob provedení daného úkolu, který povede na co nejlepší kýžený výsledek. Nyní zde bude uvedený příklad, podobný tomu ze zdroje [1], který bude ilustrovat pilotování přistávajícího letadla za použití myšlenky prediktivního řízení. Zkušený pilot ví, jak se letadlo chová a ovládá při různých podmínkách. To by se dalo interpretovat jako znalost modelu. Při přistávání si pak pilot dělá predikce budoucího manévru tak, aby letadlo dosedlo na přistávací plochu v optimální rychlosti a co nejplynuleji. Poté aplikuje první krok z jeho predikovaného řízení a letadlo se tak dostane do nového stavu. Nicméně nový stav, například vlivem větru, nemusí odpovídat tomu předpokládanému. Pilot si tak udělá novou predikci chování letadla na základě nového stavu, ve kterém se aktuálně letadlo nachází. Snaží se opět najít co nejlepší řízení tak, aby letadlo dostal v pořádku na zem. Tento postup se dále opakuje dokud letadlo bezpečně nepřistane.

Tato bakalářská práce bude obsahově organizována následovně. V této kapitole bude nejdříve v krátkosti uvedeno historické pozadí MPC a jeho vznik. Dále budou rozebrány ty nejpřínosnější vlastnosti prediktivního řízení, díky kterým se v současnosti algoritmus používá. Následovat bude kapitola, kde bude prediktivní řízení a jeho princip detailně analyzován. Bude tam popsáno, jak se predikuje budoucí chování řízené soustavy, jak probíhá optimalizace, jaká je možné klást na prediktivní regulátor omezení a další. Třetí kapitola bude věnována algoritmům prediktivního řízení. Zde budou prezentovány 3 způsoby, jakými je možné MPC implementovat. V souvislosti s tím zde bude rozebrán i speciální případ, kde na MPC regulátor nejsou kladena žádná omezení. Další kapitola se bude zabývat již samotným návrhem prediktivního řízení polohy pro 3 elektromechanické systémy s pružnou zátěží. Zde se využije algoritmus prediktivní regulace, který byl vypracován jako praktická část této bakalářské práce. Současně zde bude i diskutováno, jaké bylo dosaženo robustnosti ve stabilitě. Dále pak bude srovnána časová náročnost kvadratické optimalizace realizované v Matlabu příkazem *quadprog* při řízení tuhého a pružného systému. V souvislosti s tím zde bude odzkoušen alternativní solver optimalizačních úloh zvaný *qpOASES*. U navržených regulátorů následně bude ověřena jejich robustnost v situaci, kdy dojde ke změnám parametrů v řízené soustavě. Jako další v této práci proběhne srovnání navrženého prediktivního regulátoru s více konvenčnějšími typy kompenzátorů, jako je kaskádní P-PI regulátor či LQR. Jako poslední v této práci proběhne odzkoušení vyvinutého MPC algoritmu na modelu reálného systému.

1.1 Dějinný kontext

Počátky této strategie řízení založené na optimalizaci sahají až do 60. let minulého století, kdy byl prezentován LQR (Linear–quadratic regulator). Tento kompenzátor přistupoval k regulaci trochu odlišným způsobem, než-li jiné konvenčnější regulátory používané v té době. Byl navržen tak, aby zaručil v jistém smyslu optimální regulaci. Nicméně v praxi se neuchytil a měl malý dopad na regulační technologie v průmyslu. Bylo to kvůli tomu, že nedokázal klást žádná omezení na uzavřenou smyčku, což by vedlo na kvalitnější řízení.

Téměř po 20 letech byly oficiálně prezentovány regulátory, které k řízení využívaly model řízeného systému, na základě kterého byla řešena minimalizační úloha pro nalezení optimálního akčního zásahu. Rovněž byly vyvinuty tak, aby dokázaly zohlednit omezení, které je chtěné klást na regulační smyčku. Zejména pak, aby dokázaly vhodným způsobem saturovat velikost akčního zásahu. Tyto kompenzátory dostaly název prediktivní regulátory a této strategii řízení se začalo říkat *Model-based predictive control* (MPC) [2].

V době jejich vzniku byly MPC regulátory používány především k regulaci složitých, pomalých chemických procesů, kde dokázaly bez problémů řídit systémy s mnoha vstupy i výstupy. Nicméně je nebylo možné použít k řízení procesů s rychlou dynamikou. Bylo to kvůli tomu, že v každém kroku algoritmu dochází k řešení optimalizační úlohy. Proces optimalizace je výpočetně náročná operace a výpočetní technika v době vzniku MPC jednoduše nedokázala pracovat tak rychle, aby se pomocí ní daly algoritmem MPC řídit systémy s rychlou dynamikou. V posledních dvou dekádách s rostoucí rychlostí výpočetní techniky roste i popularita MPC. V dnešní době je prediktivní řízení bráno jako kvalitní a všeobecně známý přístup k regulaci.

1.2 Důvody použití MPC

Prediktivní řízení se v současné době používá hlavně díky několika důležitým vlastnostem, kterými disponuje na rozdíl od jiných konvenčnějších přístupů k regulaci. Mezi ty nejdůležitější výhody patří, že dokáže již z principu algoritmu klást omezení na stavy řízeného systému, regulovanou veličinu, ale hlavně na generovaný akční zásah. Na rozdíl od klasičtějších regulátorů, jako je PID kompenzátor nebo LQR, tak není potřeba do uzavřené smyčky přidávat samostatný blok pro saturaci akční veličiny. Další předností MPC je, že díky řešení optimalizační úlohy dokáže generovat plynulé akční zásahy na řízený systém. Díky tomu nedochází k velkému namáhání aktuátoru a především v některých aplikacích je to mnohem přívětivější i pro řízený systém. Mezi další výhodou prediktivní regulace patří, že dokáže bez problémů řídit i procesy s mnoha vstupy i výstupy (MIMO systémy). Oproti tomu při použití PID kompenzátorů či LQR není tak jednoduché kvalitní řízení MIMO procesů realizovat.

1.3 Přehled použitých symbolů

- \mathbb{R} - obor reálných čísel
- tučná velká písmena od **A** do **Z** - matice odpovídající velikosti
- tučná malá písmena od **a** do **z** - vektory odpovídající velikosti
- \mathbf{A}^T (\mathbf{a})^T - transpozice matice (vektoru)

- $\mathbf{1}$ ($\mathbf{0}$) - jednotková (nulová) matice odpovídající velikosti
- \mathbf{I} - matice odpovídající velikosti s jedničkami na diagonále

2 Prediktivní řízení

V této kapitole bude detailně rozebrán princip fungování MPC regulace pro řízení lineárního diskrétního systému. Bude zde rovněž prezentována další modifikace algoritmu prediktivního řízení, prostřednictvím které lze klást omezení na uzavřenou smyčku. Jako další zde bude ukázáno, jak upravit algoritmus, aby se zaručila regulace konstantní poruchy. Jako zdroje informací pro tuto kapitolu sloužily především akademické práce [3] a [4].

Začne se s úplným popisem celého algoritmu prediktivního řízení. Zde budou rovněž podrobně vysvětleny dílčí úlohy, které je nutné při prediktivní regulaci řešit. Konkrétně zde bude uvedeno, jak vytvářet predikční matice, pomocí kterých bude odhadováno budoucí chování systému. Dále zde bude vysvětleno, co je v prediktivní regulaci *horizont predikcí* a *horizont řízení*. Rovněž zde bude ukázáno, jak modifikovat algoritmus technikou *move blocking* a jak definovat účelovou funkci, pro kterou bude probíhat optimalizace. Následně zde bude uvedeno, jak rozšířit algoritmus řízení, aby se zohlednila omezení na regulační smyčku a aby se dokázaly regulovat konstantní poruchy. Jako poslední zde budou vypsány výhody a nevýhody prediktivního řízení.

2.1 Algoritmus MPC

MPC (*Model-based predictive control*) algoritmus, jak název napovídá, vychází z modelu řízeného systému. Je tedy nutné disponovat znalostí modelu řízené soustavy a to v jeho diskrétní stavové reprezentaci. Jinými slovy je nutné znát matice stavového modelu řízené soustavy **A**, **B**, **C**, **D** a periodu vzorkování T_s . Algoritmus MPC je totiž již z principu diskrétní, proto je nutné mít i samotný řízený systém v diskrétním podobě. Výsledný MPC regulátor pak rovněž vyjde v diskrétním tvaru a to s totožnou periodou vzorkování jako má řízení systém.

Prvním krokem algoritmu je predikce budoucího chování systému a to pouze v diskrétních okamžicích odpovídající periodě vzorkování. K tomu se využije právě matic ze stavového modelu řízeného systému. Samotná predikce se pak získá prostřednictvím tzv. *predikčních matic*. Jejich vytvářením se bude zabývat následující podkapitola. Predikční matice budou dávat informaci o budoucích stavech a výstupech řízeného systému.

Do jaké doby od současnosti bude predikce budoucího chování systému vytvářena, je udáváno hodnotou predikčního horizontu n_p (angl. *prediction horizon*). Horizont predikcí je jedním z volitelných parametrů úlohy a jeho hodnotu volí návrhář regulační smyčky v závislosti na povaze řízené soustavy. Dalším volitelným parametrem je horizont řízení n_c (angl. *control horizon*), který udává, kolik kroků do budoucna od současnosti se bude vypočítávat optimální hodnota řízení. Tím se zabývá technika *move blocking*. Této problematice se podrobněji bude věnovat kapitola 2.1.3.

Optimální akční zásah, který povede k dosažení chtěného výsledku, je počítán pomocí tzv. *účelové funkce*, kterou je potřeba minimalizovat. Při řešení této optimalizační úlohy pak lze klást omezení na stavy systému, jeho výstupy a i na samotný generovaný akční zásah. Účelová funkce je vytvářena pomocí predikčních matic. Její minimalizací jsou nalezeny ideální hodnoty akčního zásahu a to až do kroku n_c v budoucnu. Z predikovaných akčních zásahů je pak do systému aplikovaný pouze první z nich. Systém se tak dostane do nového stavu. Nicméně tento stav nemusí odpovídat tomu předpokládanému, například vlivem poruchy. Vypočte se tedy nová predikce chování systému až do kroku n_p od současnosti, prostřednictvím které se opět hledá ideální sekvence akčních zásahů dle účelové funkce.

Popsaný proces se neustále opakuje a to v každém kroku algoritmu.

2.1.1 Predikční matice

Predikování budoucího chování systému, a s tím spojená znalost modelu, patří k základním stavebním kamenům celé prediktivní regulace. Stavový popis řízené soustavy pak musí být co nejpřesnější. I malá neshoda mezi modelem a řízenou soustavou vede na špatnou predikci chování systému. Díky tomu se při řešení optimalizační úlohy nenajde ideální akční zásah a celková kvalita regulace bude mnohem horší.

Diskrétní řízená soustava s n stavy, m vstupy a p výstupy bude reprezentována jejím modelem ve stavové reprezentaci jako

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

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \quad (2)$$

kde $\mathbf{x} \in \mathbb{R}^{n \times 1}$ označuje stavy systému, $\mathbf{u} \in \mathbb{R}^{m \times 1}$ vstupy a $\mathbf{y} \in \mathbb{R}^{p \times 1}$ výstupy řízené soustavy, dále pak matice dynamiky $\mathbf{A} \in \mathbb{R}^{n \times n}$, matice vstupů $\mathbf{B} \in \mathbb{R}^{n \times m}$, výstupní matice $\mathbf{C} \in \mathbb{R}^{p \times n}$ a vstupně-výstupní matice $\mathbf{D} \in \mathbb{R}^{p \times m}$.

Využitím vztahů (1) a (2) lze nyní vyjádřit vývoj stavů v následujícím kroku \mathbf{x}_{k+2} a výstupů v následujícím kroku \mathbf{y}_{k+1} . Dosazením se pak získá

$$\mathbf{x}_{k+2} = \mathbf{A}\mathbf{x}_{k+1} + \mathbf{B}\mathbf{u}_{k+1} = \mathbf{A}^2\mathbf{x}_k + \mathbf{A}\mathbf{B}\mathbf{u}_k + \mathbf{B}\mathbf{u}_{k+1} \quad (3)$$

$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1} + \mathbf{D}\mathbf{u}_{k+1} = \mathbf{C}\mathbf{A}\mathbf{x}_k + \mathbf{C}\mathbf{B}\mathbf{u}_k + \mathbf{D}\mathbf{u}_{k+1}. \quad (4)$$

Obdobně lze postupovat i pro predikci vývoje stavů a výstupů v následujících krocích. Do jakého časového okamžiku v budoucnu se bude predikce počítat určuje hodnota horizontu predikcí n_p . Aplikováním popsané myšlenky vývoje stavů a výstupů lze predikovat chování systému až do kroku n_p od současnosti. Maticově to lze zapsat jako

$$\underbrace{\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_{k+2} \\ \vdots \\ \mathbf{x}_{k+n_p} \end{bmatrix}}_{\mathbf{x}_{k+1, n_p}} = \underbrace{\begin{bmatrix} \mathbf{A} \\ \mathbf{A}^2 \\ \vdots \\ \mathbf{A}^{n_p} \end{bmatrix}}_{\mathbf{P}_x} \mathbf{x}_k + \underbrace{\begin{bmatrix} \mathbf{B} & \mathbf{0} & \cdots \\ \mathbf{A}\mathbf{B} & \mathbf{B} & \cdots \\ \vdots & \vdots & \ddots \\ \mathbf{A}^{n_p-1}\mathbf{B} & \mathbf{A}^{n_p-2}\mathbf{B} & \cdots \end{bmatrix}}_{\mathbf{H}_x} \underbrace{\begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+n_p-1} \end{bmatrix}}_{\mathbf{u}_{k, n_p-1}}. \quad (5)$$

Pro zjednodušení lze využít naznačené substituce. Vývoj stavů řízené soustavy se pak až do kroku n_p vypočte jako

$$\mathbf{x}_{k+1, n_p} = \mathbf{P}_x \mathbf{x}_k + \mathbf{H}_x \mathbf{u}_{k, n_p-1}. \quad (6)$$

Obdobně lze postupovat i pro predikci vývoje výstupů řízené soustavy

$$\begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n_p-1} \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n_p-1} \end{bmatrix}}_{\mathbf{P}} \mathbf{x}_k + \underbrace{\begin{bmatrix} D & 0 & 0 & \cdots \\ CB & D & 0 & \cdots \\ CAB & CB & D & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ CA^{n_p-2}B & CA^{n_p-3}B & CA^{n_p-4}B & \cdots \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_{k+n_p-1} \end{bmatrix}}_{\mathbf{u}_{k,n_p-1}}. \quad (7)$$

I zde se pro zjednodušení zápisu využije uvedené substituce. Predikovaný vývoj výstupů systémů lze pak vyjádřit tímto způsobem

$$\mathbf{y}_{k,n_p-1} = \mathbf{P}\mathbf{x}_k + \mathbf{H}\mathbf{u}_{k,n_p-1}. \quad (8)$$

Matice \mathbf{P}_x a \mathbf{H}_x jsou nazývány *stavové predikční matice*. Matice pro výstup, \mathbf{P} a \mathbf{H} , jsou označovány jako *výstupní predikční matice*. Pro získání predikovaného vývoje stavů a výstupů je tedy potřeba znát stavový model řízené soustavy, hodnoty stavů v aktuálním kroku \mathbf{x}_k a vstup do systému v následujících n_p krocích \mathbf{u}_{k,n_p-1} .

2.1.2 Horizont predikcí n_p a horizont řízení n_c

Jak již bylo uvedeno dříve, horizont predikcí se označuje n_p a je to jeden ze vstupních parametrů úlohy prediktivní regulace. Určuje, až do jakého kroku od toho aktuálního bude predikováno chování systému. Hodnota n_p vlastně říká, jak daleko do budoucnosti má MPC regulátor odhadovat hodnoty stavů a výstupů řízené soustavy, na základě kterých se budou vypočítávat optimální akční zásahy. Jestliže se horizont predikcí n_p vynásobí s periodou vzorkování T_s , zjistí se tím, do jaké doby v budoucnosti v jednotkách času bude od aktuálního okamžiku tvořena predikce. Například bude-li perioda vzorkování $T_s = 0.1$ a hodnota horizontu predikcí $n_p = 50$, tak se bude odhadovat chování řízené soustavy až do času 5 sekund od současnosti.

Dalším vstupním parametrem prediktivní regulace je horizont řízení, který se označuje n_c . Jeho hodnota říká, kolik kroků do budoucnosti od současnosti se mají počítat optimální akční zásahy. Hodnota tohoto parametru se musí volit menší nebo rovna hodnotě horizontu predikcí. Je logické, že v opačném případě, tj. $n_c > n_p$, by nebylo možné optimální akční zásah získat, jelikož by nebyly známé predikované hodnoty stavů a výstupů řízené soustavy. Horizont řízení se nicméně z pravidla volí o něco menší než je hodnota horizontu predikcí. Je to z toho důvodu, že na průběh regulace v následujících krocích má největší vliv pouze prvních pár hodnot vypočtených akčních zásahů. Vypočtené hodnoty řízení ve vzdálenější budoucnosti totiž s největší pravděpodobností budou v následujících krocích vypočteny poměrně odlišné. Obecně je tedy doporučováno hodnotu horizontu řízení n_c volit v rozmezí od 10 % do 20 % hodnoty horizontu predikcí n_p .

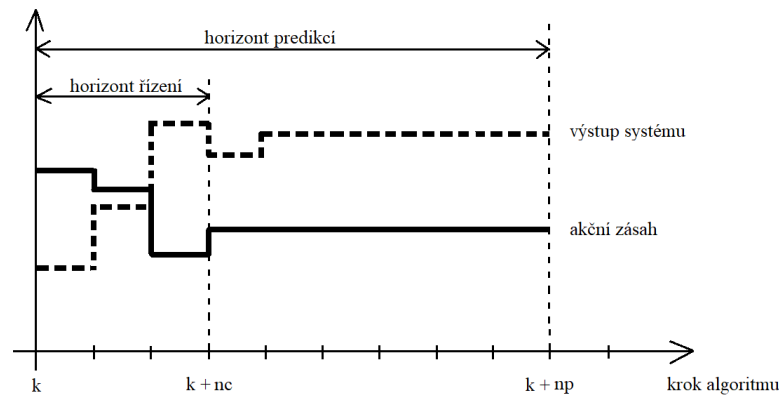
Pro získání predikce chování řízené soustavy je ale zapotřebí dle rovnic (6) a (8) znát hodnoty optimálního akčního zásahu pro celý horizont predikcí. To se obvykle řeší tak, že od kroku n_c až do kroku n_p zůstává hodnota optimálního akčního zásahu konstantní. Hodnota akčních zásahů v těchto krocích odpovídá tomu poslednímu vypočtenému, tj. tomu z kroku n_c . Touto problematikou se bude zabývat následující kapitola s názvem *move blocking*.

Po vyřešení optimalizační úlohy (tj. minimalizací účelové funkce), a tím i získání op-

timálních akčních zásahů v následujících n_c krocích, se do systému aplikuje pouze první z nich. Hodnoty predikovaných stavů, výstupů a i optimálních akčních zásahů od kroku $k + 1$ již nejsou zapotřebí a z algoritmu regulace jsou odstraněny. Řízená soustava se aplikováním prvního optimálního akčního zásahu dostane do nového stavu a je predikováno opět chování systému až do kroku n_p od současnosti. Interval predikcí chování se tak současně se systémem dostal o jeden krok dopředu. Tento princip je označován jako *receding horizon* [3].

Se zvyšující se hodnotou horizontu predikcí n_p a horizontu řízení n_c lze všeobecně dosáhnout kvalitnější regulace. Nicméně s každou predikcí navíc, která se musí v každém kroku počítat, roste výpočetní náročnost celé úlohy. Hodnoty těchto vstupních parametrů je tedy nutné vždy volit v závislosti na typu řízeného systému a na výkonnosti hardwaru, na kterém by byl MPC regulátor implementován.

Problematiku horizontu predikcí a horizontu řízení ilustruje následující obrázek.



Obrázek 1: Ilustrace horizontu predikcí n_p a horizontu řízení n_c

2.1.3 Technika move blocking

Technika move blocking se obecně zabývá tím, že při řešení optimalizační úlohy se počítají optimální akční zásahy pouze do kroku n_c od současnosti. Od kroku n_c do kroku n_p pak zůstávají hodnoty predikovaných akčních zásahů do systému na konstantní hodnotě a to právě na té, která byla jako optimální vypočtena pro krok n_c . Důvody, proč se n_c volí menší než n_p , byly uvedeny v předchozí podkapitole. Jedním z těch hlavních důvodů bylo, že se tím snižuje výpočetní náročnost celé úlohy prediktivní regulace.

Princip move blocking se do MPC algoritmu zahrne na základě následující myšlenky. Je chtěné, aby akční zásahy od kroku n_c byly konstantní. To lze vyjádřit jako $\mathbf{u}_i = \mathbf{u}_{k+n_c-1}$ pro všechna $i > k + n_c - 1$ [1]. Maticově to lze zapsat

$$\mathbf{u}_{k,n_p-1} = \begin{bmatrix} \mathbf{u}_k \\ \vdots \\ \mathbf{u}_{k+n_c-1} \\ \vdots \\ \mathbf{u}_{k+n_p-1} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & & & & \\ & \ddots & & & \\ & & \mathbf{I} & & \\ & & & \ddots & \\ & & & & \mathbf{I} \end{bmatrix}}_{\mathbf{M}_b} \underbrace{\begin{bmatrix} \mathbf{u}_k \\ \vdots \\ \mathbf{u}_{k+n_c-1} \end{bmatrix}}_{\mathbf{u}_{k,n_c-1}}, \quad (9)$$

kde $\mathbf{M}_b \in \mathbb{R}^{n_p m \times n_c m}$, přičemž m označuje počet vstupů řízeného systému.

Vektor \mathbf{u}_{k,n_c-1} vyjde jako řešení optimalizační úlohy (tj. minimalizací účelové funkce). Nicméně pro predikci budoucího chování systému jsou potřeba akční zásahy až do kroku n_p od současnosti (\mathbf{u}_{k,n_p-1}). Ty lze jednoduše získat pomocí vztahu

$$\mathbf{u}_{k,n_p-1} = \mathbf{M}_b \mathbf{u}_{k,n_c-1}. \quad (10)$$

Na základě toho pak lze stavovou predikční matici \mathbf{H}_x a výstupní predikční matici \mathbf{H} transformovat do nového tvaru, kdy za vstup bude brána pouze sekvence řízení \mathbf{u}_{k,n_c-1} . Konkrétně to bude $\mathbf{H}_{b,x} = \mathbf{H}_x \mathbf{M}_b$ pro stavy a $\mathbf{H}_b = \mathbf{H} \mathbf{M}_b$ pro výstupy. Upravené vztahy pro výpočet predikovaného chování řízené soustavy pak vyjdou jako

$$\mathbf{x}_{k+1,n_p} = \mathbf{P}_x \mathbf{x}_k + \mathbf{H}_{b,x} \mathbf{u}_{k,n_c-1} \quad (11)$$

$$\mathbf{y}_{k,n_p-1} = \mathbf{P} \mathbf{x}_k + \mathbf{H}_b \mathbf{u}_{k,n_c-1}. \quad (12)$$

2.1.4 Účelová funkce a optimalizace

Účelová funkce (angl. *cost function*) má při prediktivní regulaci zásadní roli. Pomocí číselného ohodnocení dává algoritmu v každém kroku informaci o tom, jak je daná predikce chování řízeného systému kvalitní.

Pro lepší představu, co účelová funkce znamená, zde bude uvedený příklad. Ten bude ilustrovat cestování osobním automobilem, kterým je chtěné se dostat z bodu A do bodu B. Pro samotnou cestu jsou dána dvě kritéria. Prvním z nich je to, aby cesta trvala co nejkratší dobu. Druhé kritérium se zaměřuje na to, aby cestování bylo co nejlevnější, tudíž aby se spotřebovalo co nejméně paliva. Logicky lze odvodit, že když auto pojede rychle, tak se spotřebuje hodně paliva. Naopak pojede-li auto pomalu, tak bude cesta trvat dlouho. Samozřejmě obě dvě tato kritéria mají odlišnou váhu. Někdy je potřeba se dostat na dané místo rychle a to bez ohledu na cenu. V jiném případě zase stačí na dané místo dojet a na čase zase tolik nezáleží. Mezi oběma kritérii je tedy vhodné nalézt určitý kompromis.

Účelová funkce je tvořena predikčními maticemi. Její hodnota se počítá tak, že je hledáno její minimum (tj. minimalizace). Tento proces se nazývá optimalizační úloha a na něm je celá prediktivní regulace založena. Účelová funkce s nejmenší hodnotou by měla zaručovat, že se při regulaci v následujících krocích bude výstup systému blížit k požadované hodnotě (např. hodnota regulované veličiny k referenční hodnotě). Zároveň by mělo hledání minima účelové funkce v každém kroku zajistit, že rozdíl akčních zásahů ve dvou následujících krocích po

sobě bude co nejmenší.

Když se vyřeší optimalizační úloha a nalezne se nejmenší hodnota účelové funkce (resp. optimální akční zásahy až do kroku n_c), tak se do řízeného systému aplikuje první hodnota z vypočtených hodnot řízení. Řízená soustava se tak dostane do nového stavu. Proces optimalizace se následně opakuje a v každém kroku algoritmu je opět hledáno minimum účelové funkce.

Jak již bylo řečeno, tento proces minimalizace je při prediktivní regulaci nejvíce výpočetně náročná úloha. Z toho důvodu je pak vhodné volit hodnoty predikčního horizontu a horizontu řízení v rozumných mezích, jelikož velikost těchto parametrů má zásadní vliv na výpočetní náročnost celé úlohy.

Nyní zde bude uveden tvar účelové funkce podobný tomu, který je prezentován ve zdroji [1]. Účelová funkce má obecně podobu

$$J = \sum_{i=k}^{k+n_p-1} \mathbf{e}_i^T \mathbf{Q} \mathbf{e}_i + \sum_{i=k}^{k+n_c-1} \mathbf{u}_i^T \mathbf{R} \mathbf{u}_i, \quad (13)$$

kde $J \in \mathbb{R}$ označuje hodnotu účelové funkce. Právě tuhle proměnnou je chtěné získat co nejmenší. Dále \mathbf{e}_i označuje rozdíl (angl. *error*) mezi požadovaným výstupem a tím aktuálně získaným z dané predikce. K penalizaci těchto rozdílů v každém kroku slouží váhová matice $\mathbf{Q} \in \mathbb{R}^{p \times p}$. Pomocí ní lze ovlivňovat aktuální hodnoty výstupů. Je-li váha pro daný výstup velká, tak se zvyšujícím se rozdílem regulované veličiny od požadované hodnoty se bude zvyšovat i hodnota účelové funkce. V procesu minimalizace pak bude snaha o to, aby se tato proměnná dostala na svojí požadovanou hodnotu co nejrychleji. Proměnná \mathbf{u}_i reprezentuje akční zásahy. V případě, že je nalezeno minimum účelové funkce, tak hodnoty \mathbf{u}_i jsou považovány za ty optimální, které povedou k dosažení kýženého výsledku. Váhu každého výstupu v každém kroku reprezentuje matice $\mathbf{R} \in \mathbb{R}^{m \times m}$. Její vliv na hodnotu účelové funkce je analogický tomu pro váhovou matici \mathbf{Q} . Rozdíl zde je v tom, že jsou váženy vstupy do systému v následujících n_c krocích. Je-li hodnota váhy aktuálního vstupu do řízené soustavy velká, tak se to projeví vysokou hodnotou účelové funkce a naopak.

Vhodnou úpravou účelové funkce lze pak dostat výraz, který je chtěné minimalizovat. Jeho výsledkem je optimální vektor akčních zásahů \mathbf{u}_{k,n_c-1} (v této kapitole označen pouze jako \mathbf{u}). Výraz pro optimalizační úlohu, kterou MPC regulátor musí řešit v každém kroku algoritmu, pak má tvar

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{x}_k^T \mathbf{f}^T \mathbf{u}, \quad (14)$$

kde vektor \mathbf{u} (resp. \mathbf{u}_{k,n_c-1}) označuje nalezenou optimální sekvenci akčních zásahů, které povedou na dosažení chtěného výsledku. Dále \mathbf{x}_k označuje vektor stavu řízeného systému v aktuálním kroku. Matice \mathbf{G} a \mathbf{f} se nazývají matice optimalizace. Jsou tvořeny predikčními maticemi. V jakém konkrétním tvaru jsou v těchto maticích optimalizace naskládané predikční matice závisí na typu prediktivní regulace a co je jejím cílem. Podrobně se algoritmy prediktivního řízení zabývá kapitola 3.

K samotnému řešení této optimalizační úlohy existují již dobře odladěné algoritmy. Z toho důvodu se jimi tato práce již nebude dopodrobna zabývat. Praktická část této ba-

kalářské práce byla vytvořena v softwaru Matlab. V tomto prostředí existuje pro řešení této optimalizační úlohy příkaz $\gg \text{quadprog}(\mathbf{G}, \mathbf{f}, \mathbf{A}, \mathbf{b}, \mathbf{A}_{eq}, \mathbf{b}_{eq}, \mathbf{lb}, \mathbf{ub})$. Jeho parametry jsou matice optimalizace \mathbf{G} a \mathbf{f} . Dále matice pro specifikace omezení na optimalizovanou veličinu \mathbf{A} a \mathbf{b} , kdy $\mathbf{A}\mathbf{u} \leq \mathbf{b}$. Dále rovněž matice pro zadání mezí kladených na minimalizovanou veličinu \mathbf{A}_{eq} a \mathbf{b}_{eq} . Zde ale v tomto tvaru $\mathbf{A}_{eq}\mathbf{u} = \mathbf{b}_{eq}$. Posledními vypsanými parametry uvedeného příkazu pro optimalizaci jsou vektory \mathbf{lb} a \mathbf{ub} . Pomocí těch se definují spodní a horní meze, kterých může nabývat optimalizovaná veličina, tedy $\mathbf{lb} \leq \mathbf{u} \leq \mathbf{ub}$.

Nicméně solver *quadprog* z Matlabu nepatří obecně mezi ty nejrychlejší. Z toho důvodu zde bude odzkoušen ještě jiný solver, tzv. *qpOASES*, který je volně stažitelný ze stránky <https://github.com/coin-or/qpOASES>. Tento solver byl následně implementován do Matlabu pomocí zdroje [5]. Jedná se o open-source implementaci určité varianty numerické metody vhodné pro výpočty s krátkými periodami vzorkování. Umožňuje totiž v rámci jednotlivých kroků volit tzv. hot-start, který výrazně zkracuje dobu výpočtu, protože se využívají mezivýsledky z minulého kroku.

2.2 Omezení

Jeden z hlavních důvodů používání prediktivní regulace je, že lze klást omezení na důležité veličiny v uzavřené smyčce. Poměrně jednoduše je tak možné vhodnou implementací algoritmu zaručit, že se stavy řízeného systému, regulovaná veličina a hlavně generovaný akční zásah budou pohybovat pouze v zadaných mezích. V této kapitole jako zdroj informací sloužila především akademická práce [4].

Při prediktivní regulaci lze samotná omezení rozdělit do dvou tříd. První z nich jsou tzv. **tvrdá omezení** a druhá se nazývají **měkká omezení**. Těmto limitacím se budou věnovat zvlášť následující 2 podkapitoly.

2.2.1 Tvrdá omezení

Tvrdá omezení jsou taková omezení, jejichž meze nelze za žádných okolností překročit. Z toho důvodu se tvrdá omezení používají hlavně u generovaných akčních zásahů. Z pravidla totiž platí, že akční veličina má nějaký fyzikální význam. Jako příklad může být uveden výkon kotle v % pro řízení teploty v domě. Je totiž nesmysl, aby výkon kotle pro tento případ nabyl například hodnoty -50 %.

Při aplikaci tvrdých omezení je chtěné, aby se generovaný akční zásah \mathbf{u} pohyboval mezi určitou horní mezí $\bar{\mathbf{u}}$ a danou dolní mezí $\underline{\mathbf{u}}$. To by mělo platit pro všechny predikované akční zásahy až do kroku $k + n_p$. Maticově to lze vyjádřit jako $\mathbf{S}\mathbf{u} \leq \mathbf{t}$ [3]. Po rozepsání této nerovnice pak

$$\underbrace{\begin{bmatrix} I & & & & \\ & \ddots & & & \\ & & I & & \\ -I & & & & \\ & & & \ddots & \\ & & & & -I \end{bmatrix}}_{\mathbf{S} \in \mathbb{R}^{2 \cdot m \cdot n_p \times m \cdot n_p}} \mathbf{u}_{k, n_p-1} \leq \underbrace{\begin{bmatrix} \bar{\mathbf{u}} \\ \vdots \\ \bar{\mathbf{u}} \\ -\underline{\mathbf{u}} \\ \vdots \\ -\underline{\mathbf{u}} \end{bmatrix}}_{\mathbf{t} \in \mathbb{R}^{2 \cdot m \cdot n_p \times 1}}. \quad (15)$$

Tato omezení jsou následně brána v potaz při řešení optimalizační úlohy, kde na ně musí být brán ohled.

Jednou z modifikací algoritmu zde může být to, že omezení budou kladena pouze na určitý počáteční interval predikcí a ne pro celý horizont n_p . Touto úpravou se sníží výpočetní náročnost úlohy. Nicméně zároveň se s tím zhorší celková kvalita regulace.

2.2.2 Měkká omezení

Jako měkká omezení jsou označována taková omezení, která lze pod učenou penalizací překročit. Používají se především k limitacím regulované veličiny a stavů řízeného systému. Je to z toho důvodu, že když by na tyto veličiny byla kladena tvrdá omezení, tak to může vést až k tomu, že pro optimalizační úlohu nebude existovat řešení [3]. Jako příklad zde může být uvedeno, když se stavy vlivem poruchy dostanou mimo interval tvrdých omezení. Řešení optimalizační úlohy vzhledem k omezení pak nemá řešení.

Aby bylo možné měkká omezení zahrnout do algoritmu prediktivního řízení, tak je nutné vhodným způsobem upravit optimalizační úlohu. Řešením je zavést tzv. slack proměnné (angl. *slack variables*). Postup jejich začlenění do algoritmu byl převzat ze zdroje [4].

Slack proměnné se vhodnou úpravou doplní do účelové funkce. Jejich zavedení do algoritmu regulace zde bude ilustrováno pro regulovanou veličinu \mathbf{y} . Pro případ stavových proměnných by byl postup zavedení slack proměnných analogický.

Dolní a horní omezení pro regulovanou veličinu jsou nyní uvažována ve tvaru

$$\underline{\mathbf{y}}_{k,n_p-1} - \underline{\mathbf{s}} \leq \mathbf{y}_{k,n_p-1} \leq \overline{\mathbf{y}}_{k,n_p-1} + \overline{\mathbf{s}}, \quad (16)$$

kde $\underline{\mathbf{y}}_{k,n_p-1} \in \mathbb{R}^{p \cdot n_p \times 1}$, $\overline{\mathbf{y}}_{k,n_p-1} \in \mathbb{R}^{p \cdot n_p \times 1}$ (dále v této kapitole pouze jako $\underline{\mathbf{y}}$, $\overline{\mathbf{y}}$) označují měkké meze pro regulovanou veličinu a $\underline{\mathbf{s}} \in \mathbb{R}^{p \cdot n_p \times 1}$, $\overline{\mathbf{s}} \in \mathbb{R}^{p \cdot n_p \times 1}$ jsou zmiňované slack proměnné. Jejich hodnoty vyjadřují o kolik byla dolní nebo horní mez regulovanou veličinou překročena. Substitucí se pak získá

$$\mathbf{s} = \begin{bmatrix} \overline{\mathbf{s}} \\ \underline{\mathbf{s}} \end{bmatrix}, \underline{\mathbf{y}} = \begin{bmatrix} \overline{\mathbf{y}} \\ \underline{\mathbf{y}} \end{bmatrix}, \mathbf{Q}_s = \begin{bmatrix} \overline{\mathbf{Q}}_s & \mathbf{0} \\ \mathbf{0} & \underline{\mathbf{Q}}_s \end{bmatrix}, \quad (17)$$

kde $\overline{\mathbf{Q}}_s \in \mathbb{R}^{p \cdot n_p \times p \cdot n_p}$ a $\underline{\mathbf{Q}}_s \in \mathbb{R}^{p \cdot n_p \times p \cdot n_p}$ jsou váhové matice pro penalizace horních a dolních mezí.

Prostřednictvím predikčních matic ze vztahu (12) lze pak nerovnici (16) vyjádřit jako

$$\underline{\mathbf{y}}_{k,n_p-1} - \underline{\mathbf{s}} \leq \mathbf{P}\mathbf{x}_k + \mathbf{H}_b \mathbf{u}_{k,n_c-1} \leq \overline{\mathbf{y}}_{k,n_p-1} + \overline{\mathbf{s}}. \quad (18)$$

Účelová funkce se nyní musí upravit do tvaru pro zahrnutí slack proměnných. Její tvar je pak

$$J' = J + \sum_{i=k}^{k+n_p-1} \mathbf{s}_i^T \mathbf{Q}_s \mathbf{s}_i. \quad (19)$$

Cílem upraveného algoritmu nyní bude minimalizovat mimo akčních zásahů i slack proměnné. Vektor, který bude řešením optimalizační úlohy, má tvar

$$\mathbf{h} = [\mathbf{u}_{k,n_c-1} \quad \mathbf{s}]^T. \quad (20)$$

Na základě upravené účelové funkce se získal výraz, který bude chtěné v každém kroku algoritmu minimalizovat. Má tvar

$$\min_{\mathbf{u}_{k,n_c-1}, \mathbf{s}} \frac{1}{2} \mathbf{h}^T \mathbf{G}' \mathbf{h} + \begin{bmatrix} \mathbf{x}_k \\ \underline{\mathbf{y}} \end{bmatrix}^T \mathbf{f}'^T \mathbf{h}, \quad (21)$$

kde

$$\mathbf{G}' = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_s \end{bmatrix}, \mathbf{f}' = \begin{bmatrix} \mathbf{f} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (22)$$

Soft omezení nyní lze přepsat do tvaru lineárních nerovností, které budou kladeny na minimalizační proměnné $(\mathbf{u}_{k,n_c-1}, \mathbf{s})$. Tyto nerovnosti musejí být brány při řešení optimalizační úlohy v potaz. Tvar těchto nerovností je

$$\begin{bmatrix} \mathbf{H}_b & -\mathbf{1} & \mathbf{0} \\ -\mathbf{H}_b & \mathbf{0} & -\mathbf{1} \end{bmatrix} \mathbf{h} \leq \begin{bmatrix} -\mathbf{P} & \mathbf{1} & \mathbf{0} \\ \mathbf{P} & \mathbf{0} & -\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \underline{\mathbf{y}} \end{bmatrix}. \quad (23)$$

2.3 Regulace poruch

V principu existují 3 základní zdroje chyb v ustáleném stavu, se kterými je nutné se vypořádat. První z nich je rozdíl mezi modelem systému a jeho skutečnou dynamikou. Jak již bylo psáno dříve, tak je potřeba, aby model systému byl co nejvěrnější realitě. Druhým zdrojem chyb jsou vnější poruchy působící na systém a to měřitelné nebo neměřitelné. Dále častějším případem jsou ovšem poruchy neměřitelné. Dalším zdrojem chyb je nenulová reference a požadavek na nulovou odchylku v ustáleném stavu - přesnost regulace při sledování.

V kapitole 3 o algoritmech prediktivního řízení budou uvedeny 3 možné způsoby implementace prediktivní regulace. Každý je vytvořen tak, aby plnil trochu jiný účel. Nicméně ani jeden z nich bez další modifikace neřeší problém nepřesného modelu a vnějších poruch. Ty je možné odstranit v zásadě dvěma způsoby.

Prvním z nich je zavedení měřitelných poruch do predikčního modelu. Měřitelné poruchy se jednoduše zavedou do modelu na místo, kde ve skutečnosti působí. Pořád ale není vyřešen

problém s nepřesným modelem.

Druhý, častější případ, je odhad neměřitelných poruch a jejich zavedení do predikčního modelu. Neměřitelná porucha se většinou přidává na vstup nebo výstup systému a její hodnota se odhaduje v reálném čase rekonstruktorem stavu. Poruchy se obvykle modelují jako konstantní (model v podobě integrátoru). Pro složitější známé tvary je ale možné zařadit libovolný generátor signálu. Výhodou tohoto postupu je to, že dokáže zároveň opravit chybu predikce modelu v ustáleném stavu, protože rozdíl mezi predikovaným a měřeným výstupem je vždycky možné srovnat konstantní poruchou působící na vstup nebo výstup systému. To znamená, že tento postup odregulování poruchy prostřednictvím jejího odhadu rekonstruktorem řeší současně první dva zdroje chyb uvedené v prvním odstavci této kapitoly.

Pro odregulování poruchy je nutné algoritmus prediktivního řízení upravit následujícím způsobem. Je potřeba rozšířit stavový model řízené soustavy o další konstantní stav, jehož hodnota působí na stejném místě jako vstup na vnitřní stavy řízeného systému. Hodnota této poruchy se pak stává přímo součástí vektoru stavu. Dosazuje se přímo její známá hodnota, pokud je měřitelná. V opačném případě, je-li neměřitelná, tak se vloží do vektoru stavu její odhad z rekonstruktora. Tím je zajištěná konzistence predikcí s reálným řízeným systémem.

2.4 Výhody a nevýhody prediktivního řízení

MPC regulátory se využívají zejména díky několika jejich vlastnostem, kterými disponují na rozdíl od klasičtějších přístupů k regulaci. Jeden z nejvýznamnějších rozdílů oproti PID kompenzátorům či LQR je, že prediktivní regulátor využívá pro svoji činnost model řízeného systému. U ostatních strategií řízení se predikce chování řízené soustavy vůbec nevyužívá. MPC regulátory naopak již z principu pracují tak, že predikují chování řízeného systému na základě jeho modelu. Díky tomu lze pak dosáhnout plynulejších akčních zásahů. To může být přívětivější pro aktuátor, který realizuje akční zásahy a díky tomu se tolik neopotřebovává. Zároveň je to přínosné i pro samotný řízený systém, kdy prudké změny stavů systému nemusejí být v některých aplikacích vůbec vhodné.

Mezi další výhody prediktivní regulace patří, že MPC kompenzátory dokáží již z principu klást omezení na důležité veličiny uzavřené smyčky. Jmenovitě lze bez problémů zavést omezení na stavy systému, regulovanou veličinu, ale hlavně na generovaný akční zásah. Omezení lze při tom rozdělit na tvrdá omezení a měkká omezení, což pro případ klasičtějších strategií řízení není dost dobře realizovatelné. Obecně je tedy na tom MPC regulátor z hlediska omezení oproti ostatním přístupům k regulaci mnohem lépe. Do regulační smyčky totiž není zapotřebí přidávat žádné další dodatečné bloky pro saturaci. Jak správně zavést omezení do algoritmu prediktivní regulace bylo ukázáno v kapitole 2.2.1, kdy byla tato problematika podrobněji rozebrána.

V neposlední řadě je u prediktivní regulace výhodou, že se prostřednictvím ní dají bez problému řídit i procesy s mnoha vstupy i výstupy. Algoritmus prediktivního řízení byl již tak každopádně vytvářen, aby jím bylo možné MIMO systémy jednoduše regulovat [2]. Při použití PID regulátorů či LQR je totiž velice složité regulaci systémů s více vstupy a výstupy správně realizovat.

Nicméně i u MPC regulace lze najít určitá negativa. Mezi ty nejzásadnější patří, že je potřeba velice přesného modelu systému, aby bylo dosaženo dobré kvality regulace. V některých případech totiž může být identifikace řízené soustavy pro získání jejího modelu velice zdlouhavá a náročná. Každopádně, když se dostatečně věrný model systému podaří získat, tak lze dosáhnout prostřednictvím MPC regulátoru vysoké kvality regulace.

Oproti klasičtějším způsobům řízení je pak prediktivní regulace poměrně výpočetně náročný algoritmus. Je to hlavně díky tomu, že v každém kroku je potřeba počítat (predikovat) budoucí chování systému, na základě kterého pak dochází k řešení optimalizační úlohy. Z tohoto důvodu při vzniku MPC regulátorů jimi nebylo možné řídit systémy s rychlou dynamikou. Je to proto, že výpočetní technika v době vzniku prediktivní regulace nebyla ještě dostatečně výkonná na to, aby se pomocí ní dal algoritmus MPC implementovat i u systémů s rychlou dynamikou.

S tím souvisí, že celkově je algoritmus prediktivní regulace při porovnání s PID regulátorem či LQR implementačně složitější správně realizovat. Jako příklad zde může být uvedena kompenzace poruch u MPC regulátoru a u PID regulátoru. Zjednodušeně řečeno, pro případ PID regulace stačí do otevřené smyčky přidat model poruchy (princip vnitřního modelu) a uzavřená smyčka již pak dokáže vliv poruchy vyrušit. Pro případ MPC kompenzátoru to zcela tak jednoduché a přímočaré není, jak bylo ukázáno v kapitole 2.3.

3 Algoritmy prediktivního řízení

V předchozí kapitole byl detailně popsán princip fungování prediktivní regulace. Nicméně byla tam vynechána část, jak přesně definovat optimalizační úlohu. Konkrétně jak vytvořit matice \mathbf{G} a \mathbf{f} , přes které je chtěné řešit minimalizaci, a tím pádem nalézt optimální akční zásahy. Právě tím, jak vytvořit tyto optimalizační matice prostřednictvím matic predikčních, se bude zabývat tato kapitola. Jako zdroj informací zde sloužila především akademická práce [3].

Z jakých predikčních matic optimalizační matice \mathbf{G} a \mathbf{f} budou tvořeny a v jakém budou tvaru, závisí na typu řešené úlohy. V této práci budou uvedeny 3 možné způsoby implementace prediktivního algoritmu. V každém z nich budou optimalizační matice vytvářeny trochu jinak a to v závislosti na tom, jaký je požadovaný účel MPC regulátoru. První z nich je implementace prediktivního řízení pouze pro řešení úlohy regulace. Tento algoritmus je podobný klasické stavové zpětné vazbě. Nepoužívá se referenční signál a regulátor funguje jen pro řízení do rovnovážného stavu $\mathbf{x}=\mathbf{0}$. Další způsob implementace zaručuje sledování referenčního signálu. Ten zavádí referenci pro výstup jako další prvky vektoru stavu. Penalizují se odchylky stavů/výstupů a vstupů od ustálených hodnot. Jako poslední zde bude uveden algoritmus pro zaručení nulové odchylky v ustáleném stavu. Na konci této kapitoly bude uveden ještě speciální případ MPC algoritmu. Je to situace, kdy na MPC regulátor nejsou kladena žádná omezení. V tom případě lze úlohu prediktivní regulace přeformulovat na klasickou stavovou zpětnou vazbu.

Ještě než se začne s popisem samotných algoritmů, tak by bylo vhodné uvést, v jakém tvaru bude uvažována minimalizační úloha. Jinými slovy, jak bude vypadat rovnice, kterou je chtěné přes vektor řízení \mathbf{u}_{k,n_c-1} (dále v této kapitole pouze \mathbf{u}) minimalizovat. Optimalizační úloha bude uvažována ve tvaru

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{f}^T \mathbf{u} + c \quad (24)$$

s ohledem na možná omezení na \mathbf{u} .

3.1 Problém regulace

V této části práce bude ukázáno, jaký zvolit tvar optimalizačních matic \mathbf{G} a \mathbf{f} pro minimalizaci (24), aby algoritmus prediktivní regulace držel řízený systém ve svém pracovním bodě. Je tedy chtěné, aby vektor stavu byl v ustáleném stavu nulový, tj. $\mathbf{x}_k = \mathbf{0}$ (resp. $\mathbf{y}_k = \mathbf{0}$). V tomto případě se nepoužívá žádný referenční signál.

Na základě těchto požadavků bude sestavena účelová funkce, kterou bude chtěné minimalizovat. Je požadováno, aby vektor stavu byl v ustáleném stavu nulový. Stejně tak je chtěné, aby bylo řízení v ustáleném stavu nulové. Účelová funkce pak má tvar

$$J = \frac{1}{2} \left(\sum_{i=k+1}^{k+n_p} \mathbf{x}_i^T \mathbf{Q} \mathbf{x}_i + \sum_{i=k}^{k+n_c-1} \mathbf{u}_i^T \mathbf{R} \mathbf{u}_i \right), \quad (25)$$

kde \mathbf{x}_i značí vektor stavu pro aktuální krok. Matice $\mathbf{Q} \in \mathbb{R}^{n \times n}$, $\mathbf{Q} \succeq 0$ je váhovou maticí

penalizující stavu. Bude-li váha zvoleného stavu v této matici vysoká oproti ostatním, tak to povede k tomu, že prostřednictvím minimalizace bude snaha dostat tento stav do počátku stavového prostoru co nejrychleji. Dále vstupní vektor \mathbf{u}_i pro daný krok. A váhová matice pro vstupy $\mathbf{R} \in \mathbb{R}^{m \times m}$, $\mathbf{R} \succeq 0$.

Použitím vztahu (11) lze účelovou funkci J rozepsat do tvaru

$$J = \frac{1}{2} \left((\mathbf{P}_x \mathbf{x}_k + \mathbf{H}_{b,x} \mathbf{u})^T \mathbf{Q}' (\mathbf{P}_x \mathbf{x}_k + \mathbf{H}_{b,x} \mathbf{u}) + \mathbf{u}^T \mathbf{R}' \mathbf{u} \right), \quad (26)$$

kde $\mathbf{Q}' \in \mathbb{R}^{n_p \cdot n \times n_p \cdot n}$ je váhová matice pro stavy po celý horizont predikcí. Matice $\mathbf{R}' \in \mathbb{R}^{n_c \cdot m \times n_c \cdot m}$ je váhová matice pro vstupy po celý horizont řízení.

Po úpravě vztahu (26) lze dojít do tvaru účelové funkce vhodné pro minimalizaci přes vektor \mathbf{u} (resp. \mathbf{u}_{k,n_c-1}). Ta má tvar

$$J = \frac{1}{2} \mathbf{u}^T \underbrace{(\mathbf{H}_{b,x}^T \mathbf{Q}' \mathbf{H}_{b,x} + \mathbf{R}')}_{\mathbf{G}} \mathbf{u} + \underbrace{(\mathbf{x}_k^T \mathbf{P}_x^T \mathbf{Q}' \mathbf{H}_{b,x})}_{\mathbf{f}^T} \mathbf{u} + \underbrace{\mathbf{x}_k^T (\mathbf{P}_x^T \mathbf{Q}' \mathbf{P}_x)}_c \mathbf{x}_k, \quad (27)$$

kde $\mathbf{G} \in \mathbb{R}^{n_c \cdot m \times n_c \cdot m}$ a $\mathbf{f}^T \in \mathbb{R}^{n_c \cdot m \times 1}$ jsou matice optimalizace. Je vidět, že na konstantě $c \in \mathbb{R}$ není minimalizační proměnná \mathbf{u} závislá, tudíž při samotné optimalizaci nebude uvažována.

Optimalizační úloha pro algoritmus prediktivní regulace zaručující udržení stavů řízeného stavu v počátku stavového prostoru má tvar

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{f}^T \mathbf{u} \quad (28)$$

s ohledem na možná omezení na \mathbf{u} .

Za poznámku zde ještě stojí, že tento algoritmus takto bez dalších úprav nekompensuje žádnou z uvedených chyb uvedených v kapitole 2.3. Nedokáže si tedy poradit s nepřesností modelu ani s působícími poruchami.

3.2 Sledování konstantní referenční trajektorie

V této kapitole bude uvedeno, jak vytvořit matice optimalizace \mathbf{G} a \mathbf{f} pro prediktivní regulaci, aby bylo při řízení zaručeno, že výstup systému bude sledovat referenční, po částech konstantní, trajektorii. Zjednodušeně řečeno, pro tento účel je nutné zavést reference výstupu jako další prvky vektoru stavu řízeného systému. Následně se zde penalizují odchylky výstupů od referenčních hodnot.

Problém sledování referenčního signálu se v prediktivní regulaci řeší tak, že se výstupní reference zahrnou do modelu řízené soustavy. Toho lze dosáhnout tak, že se vektor stavu pro samotný řízený systém rozšíří o tolik stavů, kolik má řízená soustava výstupů. Hodnoty těchto nově vzniklých stavů ve vektoru stavu pak odpovídají zadaným referencím pro každý výstup. Samotný stavový model se pak pozmění takto

$$\underbrace{\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_{r,k+1} \end{bmatrix}}_{\hat{\mathbf{x}}_{k+1}} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\hat{\mathbf{A}}} \underbrace{\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_{r,k} \end{bmatrix}}_{\hat{\mathbf{x}}_k} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}}_{\hat{\mathbf{B}}} \mathbf{u}_k, \quad (29)$$

kde $\mathbf{y}_{r,k} \in \mathbb{R}^{p \times 1}$ značí vektor referencí pro každý výstup.

Vzhledem k rozšíření vektoru stavu je nutné upravit i výstupní rovnici stavového modelu jako

$$\mathbf{y}_k = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix}}_{\hat{\mathbf{C}}} \hat{\mathbf{x}}_k + \mathbf{D}\mathbf{u}_k. \quad (30)$$

Pro zaručení sledování referenčního signálu je nyní nutné si vyjádřit rozdíl mezi referencemi a aktuálním výstupem systému. Tento *error* je pak dán vztahem

$$\mathbf{e}_k = \underbrace{\begin{bmatrix} \mathbf{C} & -\mathbf{I} \end{bmatrix}}_{\hat{\mathbf{C}}_e} \hat{\mathbf{x}}_k + \mathbf{D}\mathbf{u}_k. \quad (31)$$

Aby bylo zaručeno sledování referenční trajektorie algoritmem prediktivního řízení, tak predikční matice musejí být vytvořené pomocí modifikovaného modelu řízené soustavy. Pro výpočet predikcí budoucího chování systému se tedy použijí matice $\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}_e, \mathbf{D}$. Na základě toho pak vzniknou modifikované predikční matice $\hat{\mathbf{P}}_x, \hat{\mathbf{H}}_{b,x}, \hat{\mathbf{P}}, \hat{\mathbf{H}}_b$. Predikovaný vývoj rozdílů mezi regulovanou veličinou a aktuálním výstupem systému je pro celý horizont predikcí dán vztahem

$$\mathbf{e}_{k,n_p-1} = \hat{\mathbf{P}}\hat{\mathbf{x}}_k + \hat{\mathbf{H}}_b\mathbf{u}. \quad (32)$$

3.2.1 Matice optimalizace \mathbf{G} a \mathbf{f}^T

Nyní již lze vytvořit účelovou funkci, přes kterou bude probíhat optimalizace. To pro tento případ sledování referenční trajektorie bude funkce obsahovat vážený vektor řízení a vážené rozdíly mezi referenční trajektorií a skutečným výstupem. Matematicky zapsáno jako

$$J = \frac{1}{2} \left(\sum_{i=k}^{k+n_p-1} \mathbf{e}_i^T \mathbf{Q} \mathbf{e}_i + \sum_{i=k}^{k+n_c-1} \mathbf{u}_i^T \mathbf{R} \mathbf{u}_i \right), \quad (33)$$

kde \mathbf{e}_i označuje rozdíl mezi vektorem referencí a vektorem výstupů v aktuálním kroku. Matice $\mathbf{Q} \in \mathbb{R}^{p \times p}$, $\mathbf{Q} \succeq 0$ je váhová matice pro \mathbf{e}_i a matice $\mathbf{R} \in \mathbb{R}^{m \times m}$, $\mathbf{R} \succeq 0$ je váhová matice pro vstupy.

Za použití vztahu (32) lze účelovou funkci (33) rozepsat jako

$$J = \frac{1}{2} \left(\left(\hat{\mathbf{P}}\hat{\mathbf{x}}_k + \hat{\mathbf{H}}_b\mathbf{u} \right)^T \mathbf{Q}' \left(\hat{\mathbf{P}}\hat{\mathbf{x}}_k + \hat{\mathbf{H}}_b\mathbf{u} \right) + \mathbf{u}^T \mathbf{R}' \mathbf{u} \right), \quad (34)$$

kde $\mathbf{Q}' \in \mathbb{R}^{n_p \cdot p \times n_p \cdot p}$ je váhová matice pro rozdíly mezi referencí a skutečným výstupem po celý horizont predikcí. Matice $\mathbf{R}' \in \mathbb{R}^{n_c \cdot m \times n_c \cdot m}$ je váhová matice pro vstupy po celý horizont řízení.

V tomto tvaru již lze účelovou funkci algebraickými úpravami přepsat do podoby vhodné pro optimalizaci. Účelová funkce pak má tvar

$$J = \frac{1}{2} \mathbf{u}^T \underbrace{\left(\hat{\mathbf{H}}_b^T \mathbf{Q}' \hat{\mathbf{H}}_b + \mathbf{R}' \right)}_{\mathbf{G}} \mathbf{u} + \underbrace{\left(\hat{\mathbf{x}}_k^T \hat{\mathbf{P}}^T \mathbf{Q}' \hat{\mathbf{H}}_b \right)}_{\mathbf{f}^T} \mathbf{u} + \underbrace{\hat{\mathbf{x}}_k^T \left(\hat{\mathbf{P}}^T \mathbf{Q}' \hat{\mathbf{P}} \right)}_c \hat{\mathbf{x}}_k, \quad (35)$$

kde $\mathbf{G} \in \mathbb{R}^{n_c \cdot m \times n_c \cdot m}$ a $\mathbf{f}^T \in \mathbb{R}^{n_c \cdot m \times 1}$ jsou matice optimalizace. Je vidět, že na konstantě $c \in \mathbb{R}$ není minimalizační proměnná \mathbf{u} závislá, tudíž při samotné optimalizaci nebude uvažována.

Optimalizační úloha pro algoritmus prediktivní regulace zaručující sledování referenčního signálu při použití matic \mathbf{G} a \mathbf{f}^T pak má tvar

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{f}^T \mathbf{u} \quad (36)$$

s ohledem na možná omezení na \mathbf{u} .

Co se týče regulací poruch, tak ani tento algoritmus pro sledování po částech konstantního signálu bez dalších úprav nekompensuje žádnou z chyb uvedených v kapitole 2.3. Výjimkou je situace, kdy je požadavek na sledování referenční trajektorie výstupem řízeného systému s integrační složkou - v tomto případě může být v ustáleném stavu na vstupu nulové řízení i pro nenulový výstup odpovídající referenci. Minimum účelové funkce je 0 právě pro správný ustálený stav. Pro systémy bez astatismu, při výskytu poruch nebo při rozdílu mezi modelem a realitou, ale opět vznikne trvalá odchylka.

3.3 Nulová odchylka v ustáleném stavu

Pro zajištění nulové odchylky od konstantního referenčního signálu je při regulaci zapotřebí, aby akční zásahy v ustáleném stavu setrvaly na stejné hodnotě. Jako v předchozím případě, tak i zde to bude řešeno úpravou účelové funkce.

Je nutné, aby hodnota účelové funkce v ustáleném stavu byla nula. Nicméně v ustáleném stavu obecně nemusí platit, že hodnoty akčních zásahů budou nulové (systémy bez astatismu). Z toho důvodu dojde k následující úpravě účelové funkce. Namísto samotných hodnot řízení \mathbf{u}_i se v každém kroku budou penalizovat hodnoty rozdílů akčních zásahů ve dvou po sobě následujících krocích $\Delta \mathbf{u}_i = \mathbf{u}_i - \mathbf{u}_{i-1}$. Samotná optimalizace bude prováděna ale přes vektor řízení \mathbf{u} . Díky tomu lze stále jednoduše zavádět tvrdá omezení na generovaný akční zásah. Pro případ přírůstků $\Delta \mathbf{u}$ by bylo začlenění omezení přímo na hodnoty řízení o mnoho složitější.

Nyní zde bude představen modifikovaný model řízení soustavy, na základě kterého bude prováděna prediktivní regulace pro zaručení nulové odchylky v ustáleném stavu. Stejně jako v minulém případě, tak i zde bude stavový model doplněn o reference pro každý výstup systému. Nicméně ale zde navíc přibudou ještě stavy, do kterých se budou zavádět aplikované hodnoty řízení do systému z posledního kroku algoritmu, tedy \mathbf{u}_{-1} . Samotný vektor stavu se pak rozšíří nejen o stavy pro reference $\mathbf{y}_{r,k} \in \mathbb{R}^{p \times 1}$, ale i o aplikované hodnoty řízení z minulého kroku $\mathbf{u}_{-1} \in \mathbb{R}^{m \times 1}$. V algoritmu to bude probíhat tak, že se aplikované hodnoty řízení z minulého kroku v aktuálním kroku přímo dosadí do vektoru stavu. Modifikovaný model řízení soustavy, na základě kterého se bude predikovat chování systému, pak vypadá takto

$$\underbrace{\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{y}_{r,k+1} \\ \mathbf{u}_{-1} \end{bmatrix}}_{\tilde{\mathbf{x}}_{k+1}} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_{r,k} \\ \mathbf{u}_{-1} \end{bmatrix}}_{\tilde{\mathbf{x}}_k} + \underbrace{\begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{B}}} \mathbf{u}_k, \quad (37)$$

Výstupní rovnici ze stavového modelu je nyní potřeba upravit do tvaru

$$\mathbf{y}_k = \underbrace{\begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{C}}} \tilde{\mathbf{x}}_k + \mathbf{D}\mathbf{u}_k. \quad (38)$$

Stejným způsobem bude vyjádřen rozdíl mezi referencemi a aktuálním výstupem systému jako

$$\mathbf{e}_k = \underbrace{\begin{bmatrix} \mathbf{C} & -\mathbf{I} & \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{C}}_e} \tilde{\mathbf{x}}_k + \mathbf{D}\mathbf{u}_k. \quad (39)$$

Pro zaručení nulové odchylky v ustáleném stavu je nutné, aby byly predikční matice vytvářeny pomocí matic modifikovaného stavového modelu, tj $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}_e$, \mathbf{D} . Výsledkem toho budou následující predikční matice $\tilde{\mathbf{P}}_x$, $\tilde{\mathbf{H}}_{b,x}$, $\tilde{\mathbf{P}}$, $\tilde{\mathbf{H}}_b$. Predikci vývoje rozdílů požadovaných výstupů od těch aktuálních je pro celý horizont predikcí dána vztahem

$$\mathbf{e}_{k,n_p-1} = \tilde{\mathbf{P}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{H}}_b\mathbf{u}. \quad (40)$$

Nyní je potřeba vyjádřit rozdíl hodnot řízení ve dvou po sobě jdoucích krocích $\Delta\mathbf{u}_i = \mathbf{u}_i - \mathbf{u}_{i-1}$ pro celý horizont řízení. Maticově jako

$$\underbrace{\begin{bmatrix} \Delta \mathbf{u}_k \\ \Delta \mathbf{u}_{k+1} \\ \Delta \mathbf{u}_{k+2} \\ \vdots \end{bmatrix}}_{\Delta \mathbf{u}_{k, n_c-1}} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots \\ -\mathbf{I} & \mathbf{I} & \mathbf{0} & \cdots \\ \mathbf{0} & -\mathbf{I} & \mathbf{I} & \cdots \\ \vdots & & \ddots & \ddots \end{bmatrix}}_{\mathbf{K} \in \mathbb{R}^{n_c \cdot m \times n_c \cdot m}} \underbrace{\begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \mathbf{u}_{k+2} \\ \vdots \end{bmatrix}}_{\mathbf{u}_{k, n_c-1}} + \underbrace{\begin{bmatrix} -\mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{bmatrix}}_{\mathbf{M}} \mathbf{u}_{k-1}, \quad (41)$$

Poslední hodnotu aplikovaného řízení lze jednoduše získat pomocí vektoru stavu

$$\mathbf{u}_{k-1} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix}}_{\mathbf{L}} \tilde{\mathbf{x}}_k. \quad (42)$$

3.3.1 Matice optimalizace \mathbf{G} a \mathbf{f}^T

Nyní je již možné se pustit do sestavení účelové funkce. Ta, jak již bylo řečeno, bude namísto hodnot akčních zásahů penalizovat odchylky akčních zásahů ve dvou po sobě jdoucích krocích. Pro zajištění sledování reference bude rovněž penalizovat rozdíly mezi referencí a aktuálním výstupem systému. Matematicky takto

$$J = \frac{1}{2} \left(\sum_{i=k}^{k+n_p-1} \mathbf{e}_i^T \mathbf{Q} \mathbf{e}_i + \sum_{i=k}^{k+n_c-1} \Delta \mathbf{u}_i^T \mathbf{R} \Delta \mathbf{u}_i \right), \quad (43)$$

kde \mathbf{e}_i reprezentuje rozdíl mezi vektorem požadovaných výstupů a vektorem výstupů v aktuálním kroku. Matice $\mathbf{Q} \in \mathbb{R}^{p \times p}$, $\mathbf{Q} \succeq 0$ je váhová matice pro \mathbf{e}_i a matice $\mathbf{R} \in \mathbb{R}^{m \times m}$, $\mathbf{R} \succeq 0$ je váhová matice pro rozdíly vstupů v následujících dvou krocích.

Využitím vztahu (40) lze nyní účelovou funkci (43) rozepsat do tvaru

$$J = \frac{1}{2} \left(\left(\tilde{\mathbf{P}} \tilde{\mathbf{x}}_k + \tilde{\mathbf{H}}_b \mathbf{u} \right)^T \mathbf{Q}' \left(\tilde{\mathbf{P}} \tilde{\mathbf{x}}_k + \tilde{\mathbf{H}}_b \mathbf{u} \right) + \Delta \mathbf{u}^T \mathbf{R}' \Delta \mathbf{u} \right), \quad (44)$$

kde $\mathbf{Q}' \in \mathbb{R}^{n_p \cdot p \times n_p \cdot p}$ je matice penalizující rozdíly mezi referencí a skutečným výstupem po celý n_p . Matice $\mathbf{R}' \in \mathbb{R}^{n_c \cdot m \times n_c \cdot m}$ je váhová matice pro hodnoty rozdílů akčních zásahů ve dvou po sobě jdoucích krocích pro celý horizont řízení.

Dále lze v účelové funkci rozepsat $\Delta \mathbf{u}$ prostřednictvím vektoru \mathbf{u} a matic \mathbf{K} , \mathbf{M} , \mathbf{L} . Výsledná účelová funkce má tvar

$$J = \frac{1}{2} \left(\tilde{\mathbf{P}} \tilde{\mathbf{x}}_k + \tilde{\mathbf{H}}_b \mathbf{u} \right)^T \mathbf{Q}' \left(\tilde{\mathbf{P}} \tilde{\mathbf{x}}_k + \tilde{\mathbf{H}}_b \mathbf{u} \right) + \frac{1}{2} \left(\mathbf{K} \mathbf{u} + \mathbf{M} \mathbf{L} \tilde{\mathbf{x}}_k \right)^T \mathbf{R}' \left(\mathbf{K} \mathbf{u} + \mathbf{M} \mathbf{L} \tilde{\mathbf{x}}_k \right). \quad (45)$$

Nyní už je pouze zapotřebí přepsat účelovou funkci (45) do vhodného tvaru pro optimalizaci. Pro úlohu optimalizace řešenou v každém kroku algoritmu, která je definovaná jako

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{f}^T \mathbf{u} \quad (46)$$

s ohledem na možná omezení, vyšly matice optimalizace pro zaručení nulové odchylky v ustáleném stavu jako

$$\mathbf{G} = \tilde{\mathbf{H}}_b^T \mathbf{Q}' \tilde{\mathbf{H}}_b + \mathbf{K}^T \mathbf{R}' \mathbf{K} \quad (47)$$

$$\mathbf{f}^T = \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{P}}^T \mathbf{Q}' \tilde{\mathbf{H}}_b + \mathbf{L}^T \mathbf{M}^T \mathbf{R}' \mathbf{K} \right) \quad (48)$$

$$c = \frac{1}{2} \tilde{\mathbf{x}}_k^T \left(\tilde{\mathbf{P}}^T \mathbf{Q}' \tilde{\mathbf{P}} + \mathbf{L}^T \mathbf{M}^T \mathbf{R}' \mathbf{L} \mathbf{M} \right) \tilde{\mathbf{x}}_k, \quad (49)$$

kde $\mathbf{G} \in \mathbb{R}^{n_c \cdot m \times n_c \cdot m}$ a $\mathbf{f}^T \in \mathbb{R}^{n_c \cdot m \times 1}$ jsou matice optimalizace. I zde je vidět, že výraz c nezávisí na optimalizační proměnné, a proto při samotné minimalizaci nebude uvažován.

Co se týče chyb v ustáleném stavu, kterým byla věnována kapitola 2.3, tak zde je vyřešen problém přesnosti regulace při sledování. Díky penalizaci přírůstků akčních zásahů je zaručeno sledování referenční trajektorie i pro systémy bez astatismu. Nicméně bez dalších úprav ani zde není vyřešen problém s nepřesností modelu a odregulování poruch.

3.4 MPC regulátor bez omezení

V této podkapitole bude uveden speciální případ algoritmu MPC. Lze totiž ukázat, že v případě, kdy nejsou na MPC regulátor kladena žádná omezení, prediktivní regulace vyústí na klasickou stavovou zpětnou vazbu. Díky tomu je pak možné studovat dynamiku regulační smyčky klasickými metodami. Lze tak například nalézt póly uzavřené smyčky či analyzovat její dynamiku ve frekvenční oblasti [3].

To, že prediktivní regulaci bez kladených omezení lze přetransformovat na klasickou zpětnou vazbu, bude ukázáno na MPC algoritmu, který pro zaručuje nulovou odchylku v ustáleném stavu (kapitola 3.3). Pro dva předcházející prezentované algoritmy, tedy pro úlohu regulace a sledování reference, lze postupovat analogicky.

Bylo by dobré poznamenat, že tvar vzniklé stavové zpětné vazby lze dopočítat pro konkrétní nastavení MPC regulátoru z kvadratické účelové funkce J . Tvar účelové funkce je jiný pro každý typ algoritmu (viz. kapitola 3). Řešená optimalizační úloha pak má jediné globální minimum, které lze získat z rovnice $grad(J) = 0$.

Nyní bude ukázáno, jak lze přetransformovat úlohu prediktivní regulace na klasickou stavovou zpětnou vazbu. Je uvažována optimalizační úloha

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{G} \mathbf{u} + \mathbf{f}^T \mathbf{u} + c, \quad (50)$$

kde nejsou kladena žádná omezení na \mathbf{u} . Pro zaručení optimality je potřeba, aby se gradient funkce, kterou je chtěné optimalizovat, rovnal nule. To lze matematicky zapsat jako $\mathbf{G} \mathbf{u} + \mathbf{f} = \mathbf{0}$. Po vyjádření vektoru řízení pak platí $\mathbf{u} = -\mathbf{G}^{-1} \mathbf{f}$. Zde je již vidět jistá podoba se stavovou zpětnou vazbou.

Vzhledem k tomu, že v optimalizačních maticích \mathbf{G} a \mathbf{f} je zahrnuta predikce pro celý horizont n_p , tak je potřeba vybrat pouze první optimální hodnotu řízení. To lze provést jako

$$\mathbf{u}_k = -\mathbf{N}\mathbf{G}^{-1}\mathbf{f}, \quad (51)$$

kde $\mathbf{N} \in \mathbb{R}^{m \times n_c \cdot m}$ a jeho tvar je

$$\mathbf{N} = [\mathbf{I} \quad \mathbf{0} \quad \dots \quad \mathbf{0}]. \quad (52)$$

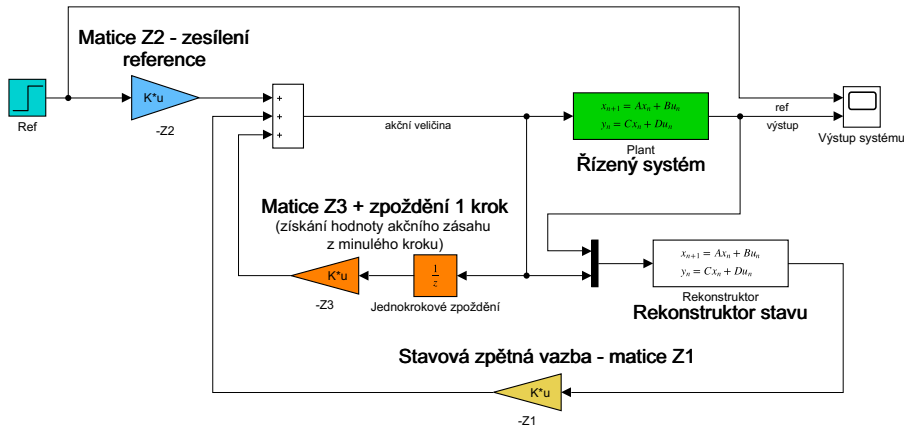
Rovnici (51) lze nyní prostřednictvím vztahů (47) a (48) rozepsat do tvaru

$$\mathbf{u}_k = -\underbrace{\mathbf{N} \left(\tilde{\mathbf{H}}_b^T \mathbf{Q}' \tilde{\mathbf{H}}_b + \mathbf{K}^T \mathbf{R}' \mathbf{K} \right)^{-1} \left(\tilde{\mathbf{H}}_b^T \mathbf{Q}' \tilde{\mathbf{P}} + \mathbf{K}^T \mathbf{R}' \mathbf{M} \mathbf{L} \right)}_{\mathbf{Z}} \tilde{\mathbf{x}}_k. \quad (53)$$

Díky tomu, že stavový vektor $\tilde{\mathbf{x}}_k$ je tvořen samotnými stavy, referencemi a aplikovanými vstupy z minulého kroku, tak lze matici $\mathbf{Z} \in \mathbb{R}^{m \times (n+p+m)}$ rozdělit na jednotlivé submatice \mathbf{Z}_1 , \mathbf{Z}_2 a \mathbf{Z}_3 dle tvaru $\tilde{\mathbf{x}}_k$. To ilustruje následující výraz

$$- [\mathbf{Z}_1 \quad \mathbf{Z}_2 \quad \mathbf{Z}_3] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y}_{r,k} \\ \mathbf{u}_{-1} \end{bmatrix}. \quad (54)$$

Nyní je již možné prostřednictvím matic \mathbf{Z}_1 , \mathbf{Z}_2 a \mathbf{Z}_3 vytvořit zapojení s klasickým stavovým regulátorem, které by mělo zaručit ekvivalentní řízení jako s původním MPC kompenzátozem. Matice \mathbf{Z}_1 přitom bude realizovat samotný stavový regulátor. Matice \mathbf{Z}_2 bude vhodně zesilovat referenční signál. Pomocí poslední matice \mathbf{Z}_3 bude regulační smyčka dostávat informaci o hodnotě posledního aplikovaného řízení. V Simulinku toto zapojení se stavovou zpětnou vazbou vypadá následovně:



Obrázek 2: Zapojení stavového regulátoru pro získání ekvivalentního řízení k MPC

První blok ve schématu (tyrkysově vybarvený) je generátor referenční trajektorie. Zde konkrétně to je bloček Step. Signál z tohoto bloku je následně vynásoben maticí \mathbf{Z}_2 (modrý bloček) a přiveden do bloku Sum, kde se sečte se dvěma zpětnovazebními signály. V první vnější zpětné vazbě se nachází samotný stavový regulátor (žlutý blok) realizovaný vektorem \mathbf{Z}_1 . Do stavového regulátoru je napojen signál z rekonstruktoru stavu. Ten zaručuje odhad všech stavů řízeného systému. Jak je vidět, tak do rekonstruktoru jsou napojeny 2 signály a to vstup a výstup řízené soustavy (zelený blok). Druhá, vnitřní zpětná vazba dává regulační smyčce informaci o hodnotě generovaného akčního zásahu z předchozího kroku. Z obrázku je vidět, že to je realizováno prostřednictvím vektoru \mathbf{Z}_3 a bloku pro jedнокrokové zpoždění (oranžově vybarvené). Zcela napravo ve schématu je výstup řízené soustavy společně s referencí napojen do bloku Scope pro vykreslení průběhů.

Pro analýzu dynamiky této regulační smyčky je vhodné znát přenos otevřené smyčky. Díky němu pak lze například ověřit chování systému ve frekvenční oblasti nebo vypočítat hodnoty důležitých indexů robustnosti ve stabilitě. Přenos otevřené smyčky tohoto zapojení se stavovým regulátorem lze získat v z-transformaci jako

$$F_o(z) = -\mathbf{Z}_1(z\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \cdot \frac{z}{z + z_3}, \quad (55)$$

kde \mathbf{A} a \mathbf{B} jsou matice ze stavového modelu řízené soustavy. Důležité je také si uvědomit, že pro případ řízení systému s jedním vstupem a jedním výstupem (SISO systému) je submatice \mathbf{Z}_3 pouze skalární veličina, a tím pádem je i přenos $F_o(z)$ skalár. Z tohoto důvodu je pak \mathbf{Z}_3 ve vztahu (55) nahrazeno z_3 .

4 Návrh prediktivního řízení

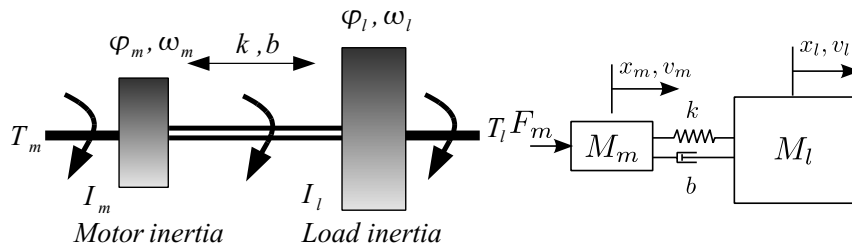
Tato kapitola se bude věnovat již přímo navrhování prediktivního regulátoru polohy pro 3 elektromechanické systémy. K tomu se použije algoritmus, který byl vytvořený jako praktická část této bakalářské práce.

Cílem přesněji bude navrhnout vhodný MPC kompenzátor polohy pro 3 pružné dvoumotové systémy tvořené elektromotorem a zátěží, které se budou lišit v hodnotě jejich rezonančního poměru. Pro návrh všech MPC regulátorů v této práci bude použit algoritmus prediktivního řízení pro zaručení nulové odchylky v ustáleném stavu, který byl popsán v kapitole 3.3.

Nejdříve proběhne seznámení s uvažovanými elektromechanickými soustavami. Bude uvedené, z čeho se skládají a jak je lze modelovat. Poté budou určeny konkrétní parametry soustav, pro které bude navrhováno řízení. Následovat bude již samotný návrh MPC regulátoru. Zde se definují omezení, které bude chtěné na řízený systém klást, hodnoty horizontu predikcí a řízení a ostatně všechny vstupní parametry, které jsou pro algoritmus prediktivního řízení potřeba. Regulace vytvořeným kompenzátozem se posléze otestuje v Simulinku. Poté ještě proběhne diskuze nad dosaženou robustností ve stabilitě a nad časovou náročností kvadratické optimalizace, která byla realizována solverem *quadprog* z Matlabu. Časová náročnost řešení optimalizační úlohy solverem *quadprog* bude srovnána při řízení pevného a pružného systému. V souvislosti s tím bude odzkoušen ještě další solver optimalizačních úloh, který je alternativou k *quadprog*, a má název *qpOASES*. Každý z regulátorů bude ještě následně otestován z hlediska jeho robustnosti při změně hodnoty zesílení a rezonance řízené soustavy.

4.1 Popis řízených systémů

Prediktivní regulace, jak již bylo řečeno, se bude navrhovat pro 3 pružné dvoumotové systémy, přesněji pro elektromotory se zátěží. Lišit se budou hodnotou rezonančního poměru. Soustava elektromotoru se zátěží se skládá ze dvou těles, z nichž jedno je samotný motor a to druhé zátěž. Spojeny jsou přes hřídel, která je s oběma tělesy pevně spojena. Právě hřídel do systému zavádí pružnost. Za ekvivalentní systém k tomuto by se dal považovat například dvoumotový mechanický lineární systém, kde jsou dvě tělesa spojeny pružinou s tlumením. To ilustruje následující obrázek:



Obrázek 3: Dvoumotový systému - motor se zátěží s ekvivalentem soustavy [6]

Zde I_m je moment setrvačnosti motoru a I_l je moment setrvačnosti zátěže. Je vidět, že hřídel lze reprezentovat jako pružinu se svou konstantou tlumení b a pružností k . Důležité je si uvědomit, že se jedná o systém s jedním vstupem, točivý moment motoru T_m , a dvěma výstupy, chování motoru a zátěže. Obvykle jsou pro zpětnou vazbu k dispozici pouze infor-

mace na straně motoru o poloze φ_m nebo rychlosti ω_m z čidel integrovaných přímo v pohonu. Cílem regulace je ovšem nepřímě řídit polohu φ_m nebo rychlost ω_l zátěže. Nebezpečí této konfigurace spočívá v tom, že nevhodný návrh regulace může vést na vyhovující chování na straně motoru, ale zcela nevhodnou dynamiku druhé hmoty.

Předchozí odstavec sloužil k obecnému popisu řízené soustavy. Nyní budou uvedeny konkrétní přenosy, které ji budou reprezentovat. První člen systému bude považován za model aktuátoru, který popisuje dynamiku z požadovaného proudu/momentu motoru na skutečný. Jeho přenosová funkce má tento tvar:

$$T_i(s) = \frac{11307(s + 3492)}{s^2 + 11309.73351s + 39478417.51}. \quad (56)$$

Jak je vidět, tak je to přenos druhého řádu.

Druhý prvek soustavy symbolizuje mechanickou část systému a bude umístěn do série za přenos aktuátoru. Bude popsán aperiodickým členem 1. řádu. Přesněji reprezentuje ideální tuhý systém bez rezonancí. Jeho přenos vypadá takto:

$$P_m(s) = \frac{1}{T_m s + 1}, \quad (57)$$

kde časová konstanta je rovna $T_m = 0.5$. Právě taková hodnota bývá typická pro uvažované systémy. Celkově tento přenos má význam dynamiky z momentu aktuátoru na rychlost připojené zátěže.

Poslední člen ze všech tří, které dohromady tvoří celý systém, je umístěn před aktuátor s přenosem T_i . Ve výsledku bude do tohoto bloku přiváděn signál přímo z regulátoru. Jedná se o filtr 1. řádu. Odpovídá mu tato přenosová funkce:

$$P_f(s) = \frac{1}{T_f s + 1}, \quad (58)$$

kde časová konstanta nabývá opět typickou hodnotu pro tyto systémy a to $T_f = 0.001$.

Celková dynamika tohoto, prozatím tuhého, systému, kterou vnímá regulátor, je tedy filtr momentu · aktuátor · mechanická zátěž. Celkový přenos má tedy tvar $P_c(s) = P_f(s) \cdot T_i(s) \cdot P_m(s)$ a vyjadřuje dynamiku z momentu motoru na rychlost motoru. Přenosová funkce je 4. řádu a má tvar

$$P_c(s) = \frac{11307(s + 3492)}{(0.001s + 1)(s^2 + 11309.73351s + 39478417.51)(0.5s + 1)}. \quad (59)$$

Roznásobením se pak získá:

$$P_c(s) = \frac{11307(s + 3492)}{0.0005s^4 + 6.156s^3 + 2.541 \cdot 10^4 s^2 + 1.979 \cdot 10^7 s + 3.948 \cdot 10^7}. \quad (60)$$

Cílem této kapitoly je ale navrhnout prediktivní řízení pro pružný systém. Je tedy nutné přenos $P_c(s)$ rozšířit o přenos symbolizující pružnou dynamiku. Bude to tudíž již celkově pružný systém a jeho chování bude trochu odlišné než v předešlém případě, kdy byl systém dokonale tuhý. Model mechanické části zde bude rozšířen o jeden kmitavý mód - rezonanci. Jinými slovy bude přidán přenos 2. řádu ve tvaru:

$$P_r(s) = \frac{s^2 + 2\xi_z\omega_z s + \omega_z^2}{s^2 + 2\xi_z\omega_z r^2 s + \omega_z^2 r^2} \cdot r^2, \quad (61)$$

kde $\omega_z = 20 \cdot 2 \cdot \pi = 125.6637 \text{ rad} \cdot \text{s}^{-1}$, což symbolizuje frekvenci antirezonance. Dále pak ξ_z reprezentuje činitel tlumení a má hodnotu $\xi_z = 0.005$. Parametr r . Ten znázorňuje poměr frekvence rezonance vůči antirezonanci. Fyzikálně řečeno tento rezonanční poměr závisí na poměru setrvačnosti zátěže ku setrvačnosti motoru. Matematicky vyjádřeno takto

$$r = \sqrt{1 + \frac{I_l}{I_m}}. \quad (62)$$

Přenos $P_r(s)$ bude uvažován ve třech variantách a to pro různé varianty rezonančního poměru r . Hodnoty r budou nabývat $r_1 = 1.1$, $r_2 = 2$ a $r_3 = 6$. Je to z toho důvodu, že tento parametr má zásadní vliv na dosažitelnou kvalitu řízení při použití PID regulátorů. To vyplývá ze struktury dynamiky řízené soustavy a omezené struktury regulátoru. U systémů s malou hodnotou r je obtížné dostatečně zatlumit pružné módy a to z toho důvodu, že jsou špatně pozorovatelné. U systémů s velkou hodnotou parametru r je naopak obtížné docílit větší šířky pásma a to proto, že generují pomalé odezvy v uzavřené smyčce. Jedním z cílů práce je i vyzkoušet, jak se tento problém projeví při návrhu složitějšího MPC regulátoru.

Parametry ω a ξ zůstanou stejné a to na hodnotách z předchozího odstavce. Na základě toho se získají 3 přenosy reprezentující různý kmitavý mód, kdy se příslušná hodnota r dosadí do přenosu $P_r(s)$. Číselně v Laplaceově transformaci mají následující tvar. Přenosy odpovídají postupně r_1 , r_2 a r_3 .

$$P_{r1}(s) = \frac{1.21(s^2 + 1.257s + 1.579 \cdot 10^4)}{(s^2 + 1.521s + 1.911 \cdot 10^4)}, \quad (63)$$

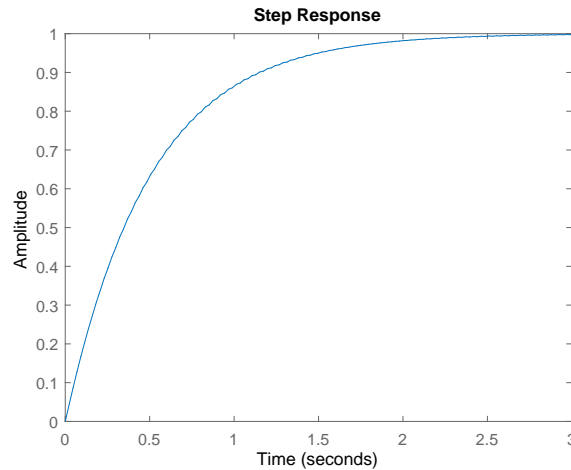
$$P_{r2}(s) = \frac{4(s^2 + 1.257s + 1.579 \cdot 10^4)}{(s^2 + 5.027s + 6.317 \cdot 10^4)}, \quad (64)$$

$$P_{r3}(s) = \frac{36(s^2 + 1.257s + 1.579 \cdot 10^4)}{(s^2 + 45.24s + 5.685 \cdot 10^5)}. \quad (65)$$

Pro získání celkových přenosů otevřených smyček pro každou hodnotu rezonance nyní stačí pouze vynásobit odpovídající přenos $P_r(d)$ s přenosem $P_c(s)$. V prvním případě pro rezonanční poměr $r_1 = 1.1$ se po vynásobení $P_c(s) \cdot P_{r1}$ dostane:

$$P_{c1}(s) = \frac{13681.47(s + 3492)(s^2 + 1.257s + 1.579 \cdot 10^4)}{(0.001s + 1)(s^2 + 11309.73351s + 39478417.51)(0.5s + 1)(s^2 + 1.521s + 1.911 \cdot 10^4)}. \quad (66)$$

Jedná se o přenos 6. řádu. Systém má tedy 6 pólů. V čitateli jsou pak 3 nuly. Pro lepší seznámení se se systémem zde bude vykreslena přechodová charakteristika, která vypadá takto:



Obrázek 4: Odezva na skok systému s přenosem $P_{c1}(s)$

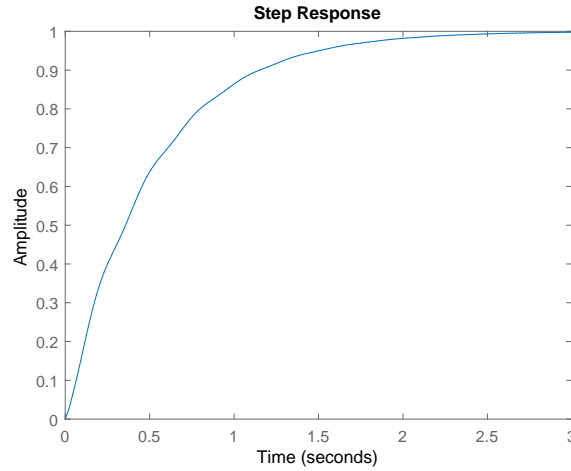
Je patrné, že kmity v průběhu výstupu zde nejsou vidět. Doba ustálení odpovídá třem vteřinám. Nicméně pro prediktivní regulaci je potřeba přenos v diskrétním tvaru. Vzorkovací frekvence bude 0.5 milisekundy. Přenos P_{c1} v diskrétní formě v z-transformaci s tvarovačem na vstupu vypadá takto:

$$P_{c1,d}(z) = \frac{0.0002165z^5 - 0.0001424z^4 - 0.0003977z^3 + 0.0003576z^2 - 2.829 \cdot 10^{-5}z - 3.881 \cdot 10^{-6}}{z^6 - 3.624z^5 + 4.896z^4 - 2.939z^3 + 0.6889z^2 - 0.02418z + 0.002119}. \quad (67)$$

Další na řadě je celkový přenos otevřené smyčky, kdy se bude uvažovat rezonanční poměr $r_2 = 2$. V tom případě má přenos otevřené smyčky po vynásobení přenosů $P_c(s) \cdot P_{r2}$ tento tvar:

$$P_{c2}(s) = \frac{45228(s + 3492)(s^2 + 1.257s + 1.579 \cdot 10^4)}{(0.001s + 1)(s^2 + 11309.73351s + 39478417.51)(0.5s + 1)(s^2 + 5.027s + 6.317 \cdot 10^4)}. \quad (68)$$

Je patrné, že opět vznikl přenos 6. řádu. Zde je jeho odezva na jednotkový skok:



Obrázek 5: Odezva na skok systému s přenosem $P_{c2}(s)$

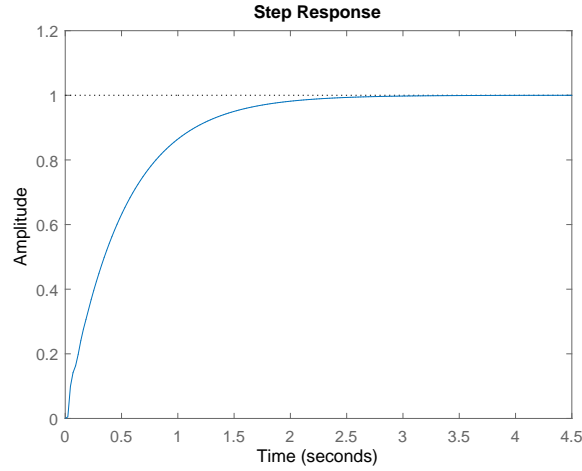
Tady již lze pozorovat v průběhu výstupu nepatrné kmity. Doba ustálení výstupu činí 3 sekundy. Z důvodu návrhu diskrétního regulátoru bude uveden opět přenos systému s odpovídající hodnotou rezonančního poměru v z-transformaci s tvarovačem na vstupu. Diskrétní přenos pro tento systém se vzorkovací frekvencí 0.5 ms vypadá takto:

$$P_{c2,d}(z) = \frac{0.0007149z^5 - 0.000471z^4 - 0.001311z^3 + 0.001179z^2 - 9.33 \cdot 10^{-5}z - 1.281 \cdot 10^{-5}}{z^6 - 3.624z^5 + 4.896z^4 - 2.939z^3 + 0.6889z^2 - 0.02418z + 0.002119} \quad (69)$$

Jako poslední přichází na řadu přenos s poměrem rezonancí $r_3 = 6$. Celkový přenos otevřené smyčky pro tuto hodnotu poměru má po vynásobení přenosů $P_c(s) \cdot P_{r3}$ tvar:

$$P_{c3}(s) = \frac{407052(s + 3492)(s^2 + 1.257s + 1.579 \cdot 10^4)}{(0.001s + 1)(s^2 + 11309.73351s + 39478417.51)(0.5s + 1)(s^2 + 45.24s + 5.685 \cdot 10^5)} \quad (70)$$

Tvar tohoto přenosu je v porovnání se dvěma předchozími pořád stejný a má 6 pólů a 3 nuly, ovšem hodnoty koeficientů jsou odlišné. Mírně odlišná by měla být i přechodová charakteristika pro tento systém. Ta vypadá takto:



Obrázek 6: Odezva na skok systému s přenosem $P_{c3}(s)$

Je zřejmé, že zde kmity v průběhu výstupu nejsou moc patrné. Pouze na začátku odezvy je vidět jeden zákmit. Výstup dojel na ustálenou hodnotu až za 4.5 sekundy. Podoba přenosu v z-transformaci s tvarovačem nultého řádu na vstupu a s poměrem rezonance r_3 pro periodu vzorkování 0.5 ms je následující

$$P_{c3,d}(z) = \frac{0.00634z^5 - 0.004264z^4 - 0.01144z^3 + 0.01035z^2 - 0.0008183z - 0.0001129}{z^6 - 3.468z^5 + 4.621z^4 - 2.803z^3 + 0.6718z^2 - 0.02342z + 0.002074}. \quad (71)$$

Uvedené přenosy $P_{c1,d}$, $P_{c2,d}$ a $P_{c3,d}$ znázorňují pružnou dynamiku z proudu/momentu na motoru na rychlost motoru. Poloha motoru se jednoduše získá tak, že se do otevřené smyčky přidá integrátor. Při regulaci je ovšem důležité sledovat chování systému na straně zátěže. Ovládání polohy (nebo rychlosti) zátěže je totiž hlavním cílem celé uvažované regulace. Z toho důvodu je si dobré vyjádřit přenosy, které budou reprezentovat dynamiku z proudu na motoru na polohu a rychlost zátěže.

Bude se zde vycházet z přenosu $P_c(s)$ pro tuhý systém (60). Pro získání pružné dynamiky z momentu motoru na rychlost motoru bylo zapotřebí tento přenos vynásobit přenosem znázorňující pružnou dynamiku (61) s odpovídající hodnotou rezonančního poměru. Pro získání informací o chování zátěže se bude postupovat obdobně. Přenos $P_c(s)$ bude nutně rovněž vynásobit přenosem reprezentující pružnou dynamiku. Tento přenos reprezentující dynamiku na zátěži bude v podobě

$$P_l(s) = \frac{2\xi_z\omega_zs + \omega_z^2}{s^2 + 2\xi_z\omega_zr^2s + \omega_z^2r^2} \cdot r^2. \quad (72)$$

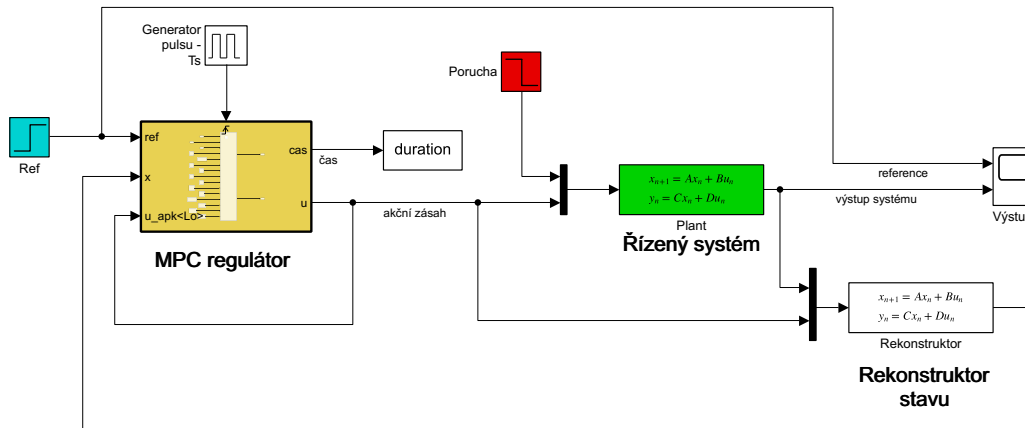
Je vidět, že tento přenos je velice podobný (61). Nicméně zde se po vynásobení s přenosem $P_c(s)$ získá dynamika z přenosu z momentu motoru na rychlost pružné zátěže. Obecně lze tedy přenos z momentu motoru na rychlost zátěže získat jako $P_c \cdot P_l$. Jestliže se výsledný přenos následně ještě vynásobí integrátorem, tak se získá dynamika z momentu na motoru na polohu zátěže. Všechny parametry přenosu (72) zůstanou stejné jako pro dynamiku na

motoru. Po zvolení hodnoty rezonančního poměru r , ale tato hodnota musí být stejná pro dynamiku na motoru i na zátěži.

Pro popis dynamiky systému byly v předchozích odstavcích představeny 4 přenosové funkce. Všechny měly totožný vstup a to točivý moment motoru. Výstupy pak představovaly polohu motoru, rychlost motoru, poloha zátěže a rychlost zátěže. Jak ale bylo uvedeno v teoretické části této bakalářské práce, tak pro aplikaci prediktivního řízení je nutné disponovat modelem řízeného systému ve stavové reprezentaci. Model pro tento systém tedy bude mít 1 vstup a 4 výstupy. Vzhledem ke znalosti všech potřebných přenosů lze výsledný stavový popis jednoduše získat v Matlabu příkazem *tf2ss*. Nesmí se ale zapomenout model převést do diskretní podoby a to s periodou vzorkování $T_s = 0.0005$ s.

4.2 Schéma v Simulinku použité pro otestování

Nyní zde bude uvedeno schéma v Simulinku, které bude sloužit pro otestování navržené prediktivní regulace. Vypadá takto:



Obrázek 7: Schéma pro otestování prediktivní regulace

Úplně vlevo modrý bloček reprezentuje vstupní referenci, kterou je jednotkový skok. Dále se ve schématu nachází žlutý blok *Triggered subsystem*. Pomocí něho je implementován do zapojení MPC regulátor. Do tohoto bloku lze vložit Matlabovskou funkci, ve které je naprogramován algoritmus prediktivní regulace vytvořený jako praktická část této bakalářské práce. MPC kompenzátor má zde 3 vstupy: referenci, vektor stavu řízeného systému z minulého kroku a poslední hodnotu aplikovaného řízení. Výstupem je pak čas, který regulátor potřeboval k vyřešení optimalizační úlohy a vygenerovaný akční zásah. Dále se ve schématu nachází řízený systém, do kterého je jako další vstup implementována vstupní porucha. Další důležitou částí zapojení je rekonstruktor. Ten, na základě vstupů a výstupů řízené soustavy, dává regulátoru informace o hodnotách každého stavu systému. Rekonstruktor je implementován tak, aby dokázal odhadnout i stav neměřitelné poruchy. Výstup řízené soustavy je pak v pravé části schématu společně s referencí napojen do bloku *Scope* pro vykreslení.

4.3 Návrh MPC regulátoru pro systém s poměrem rezonancí r_1

Zde proběhne již samotný návrh prediktivního regulátoru polohy pro systém s rezonančním poměrem r_1 . Nejprve by ale bylo vhodné pro úplnost uvést model řízeného soustavy s

poměrem rezonancí r_1 ve stavové reprezentaci. Diskrétní stavový model řízeného systému s periodou vzorkování $T_s = 0.0005$ s uvažovaný v této podkapitole vypadá takto:

$$\mathbf{x}_{k+1} = \begin{bmatrix} -0.03423 & 0.1741 & -0.3487 & 0.0646 & -0.1984 & 0.2774 & -0.1847 \\ -0.09572 & -0.1766 & 2.423 & -0.496 & 0.6198 & -0.5403 & 0.8669 \\ 0.07947 & -1.736 & 3.896 & -0.2265 & 1.044 & -1.051 & 1.013 \\ -0.2088 & 1.083 & -1.243 & 0.01046 & -0.5977 & 0.6344 & 0.03348 \\ 0.04954 & -0.761 & 0.6824 & 0.2406 & 0.2788 & -0.4682 & -0.3755 \\ -0.1055 & 0.2012 & -0.8464 & -0.335 & 0.08352 & 0.2283 & 0.1734 \\ 0.1912 & -1.424 & -0.4462 & 0.6021 & -0.1082 & 0.01588 & 0.4206 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} -0.001918 \\ 0.001632 \\ 0.038834 \\ -0.01109 \\ 0.01097 \\ -0.008411 \\ 0.01283 \end{bmatrix} \mathbf{u}_k. \quad (73)$$

$$\mathbf{y}_k = \begin{bmatrix} -0.002945 & 0.0002312 & 0.002049 & 0.002414 & 0.0002452 & 0.005261 & -0.001267 \\ 0.001897 & 0.0009492 & 0.0007998 & -0.005624 & 0.001823 & -0.005759 & 0.004454 \\ -1.005 \cdot 10^{-6} & -2.075 \cdot 10^{-6} & -1.091 \cdot 10^{-6} & -2.364 \cdot 10^{-6} & 5.136 \cdot 10^{-6} & -4.784 \cdot 10^{-6} & -6.197 \cdot 10^{-6} \\ 1.219 \cdot 10^{-6} & 5.554 \cdot 10^{-6} & 5.422 \cdot 10^{-6} & 2.658 \cdot 10^{-6} & -4.246 \cdot 10^{-6} & 1.678 \cdot 10^{-6} & -1.965 \cdot 10^{-6} \end{bmatrix} \mathbf{x}_k \quad (74)$$

Zde vstup modelu reprezentuje požadovaný proud/moment motoru. První výstup je poloha motoru, druhý rychlost motoru, třetím výstupem je poloha zátěže a čtvrtým rychlost zátěže.

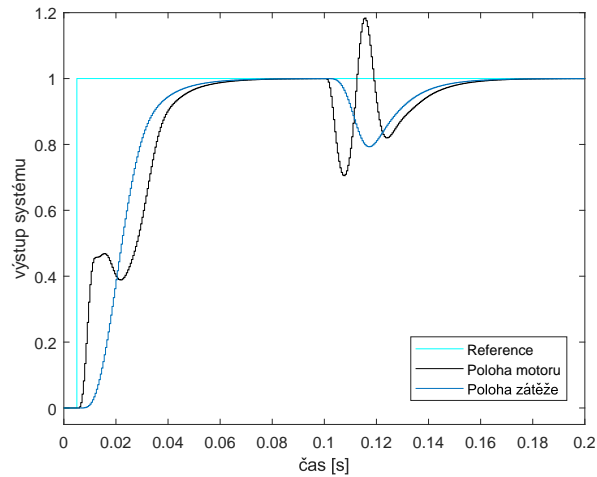
Nyní proběhne návrh MPC regulátoru polohy pro takto definovaný systém. Manuálním odzkoušením bylo dosaženo nejlepší kvality řízení při následujících parametrech MPC regulátoru. Horizont predikcí nabývá hodnoty $n_p = 220$, horizont řízení $n_c = 20$. Saturace generovaného akčního zásahu pak bude nabývat hodnot ± 15 . Dále váhová matice \mathbf{Q} pro vážení výstupů a váhová matice \mathbf{R} pro vážení hodnot řízení z účelové funkce mají podobu

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 15 \end{bmatrix}, \mathbf{R} = 0.03. \quad (75)$$

Je vidět, že je větší důraz kladen především na chování zátěže. Dále budou kladena soft omezení na hodnotu výstupů. Pro snížení výpočetní náročnosti bylo zvoleno, že soft omezení budou uvažována v predikcích pouze 10 kroků do budoucnosti od současnosti. K penalizaci bude docházet, když poloha motoru nebo zátěže překročí hodnotu ± 1.2 a když rychlost motoru nebo zátěže překročí hodnotu ± 0.5 . Zároveň bude kladeno soft omezení na torzi, což je rozdíl mezi polohou motoru a zátěží. Jinými slovy to určuje, jak moc je namáhána hřídel mezi motorem a zátěží. Právě na tuto hodnotu je vhodné klást soft omezení. Horní a spodní mez pro penalizaci torze tak bude ± 0.4 . Váhová matice pro všechna soft omezení bude identická s hodnotami 100 na diagonále.

4.3.1 Průběh regulace

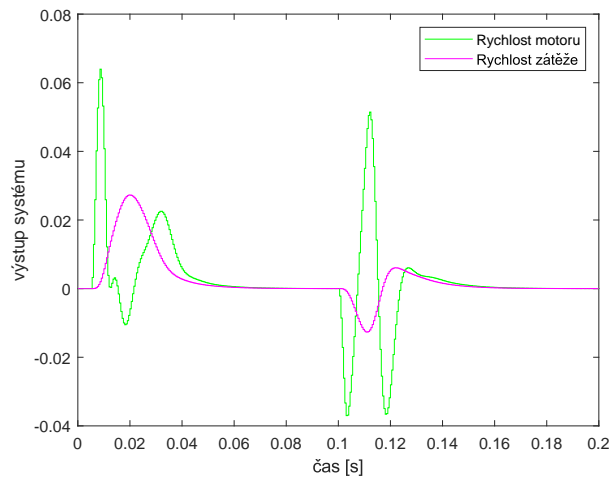
Nyní již proběhne otestování regulace při řízení navrženým MPC regulátorem. Zapojení v Simulinku, které zde sloužilo pro otestování, je na obrázku 7. Průběhy polohy motoru a zátěže při simulaci vyšly následovně:



Obrázek 8: Průběh poloh motoru a zátěže při regulaci systému s poměrem rezonancí r_1

Je vidět, že při odezvě na skok v referenční hodnotě regulátor dobře dostane systém do nové polohy odpovídající referenci. Dále lze pozorovat, že plynule s polohou motoru se pohybuje i poloha zátěže. Hřídel díky tomu není hodně mechanicky namáhána. Doba regulace činí přibližně 0.08 sekundy. V čase 0.1 vteřiny do systému zasáhla vstupní porucha o velikosti -10. Je patrné, že i s tou si regulátor dobře poradil a její vliv vyrušil. Nicméně díky pružné povaze řízené soustavy již došlo k většímu mechanickému namáhání hřídele, což z hlediska její životnosti není moc přívětivé.

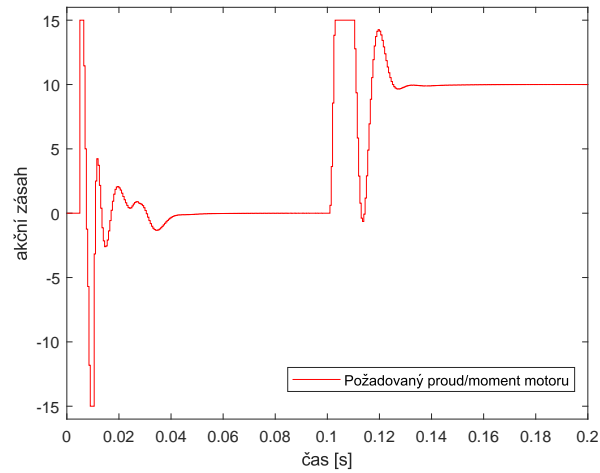
Průběhy rychlostí při skokové změně referenční hodnoty vyšly takto:



Obrázek 9: Průběh rychlosti motoru a zátěže při regulaci systému s poměrem rezonancí r_1

Je vidět, že při simulaci došlo při změně referenční hodnoty a při začátku působení poruchy ke značným výkyvům rychlosti motoru. Je to opět díky pružné dynamice řízeného systému. Lze ale předpokládat, že vzhledem k povaze systému se takovýmto výkyvům nikdy nelze zcela vyvarovat.

MPC regulátor při simulaci generoval tyto akční zásahy:



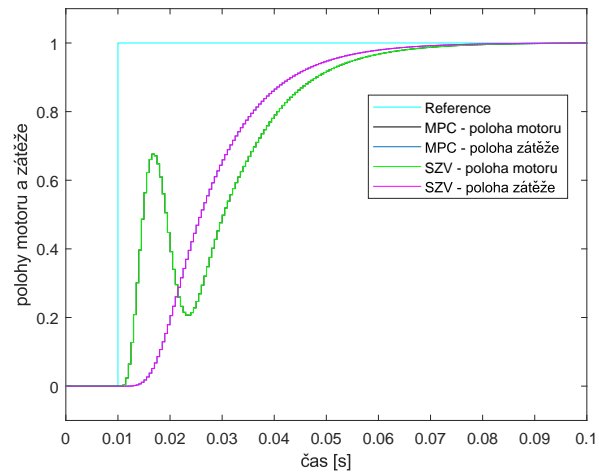
Obrázek 10: Průběh akčního zásahu při regulaci systému s poměrem rezonancí r_1

I zde je opět vidět, že se projevila pružná dynamika řízeného systému. Nicméně je zároveň patrné, že při optimalizaci byly dobře brány v potaz tvrdá omezení. Akční zásah tak nikdy nepřekročil svoje definované meze. V průběhu je od času 0.1 s také dobře vidět kompenzace poruchy.

4.3.2 Robustnost ve stabilitě

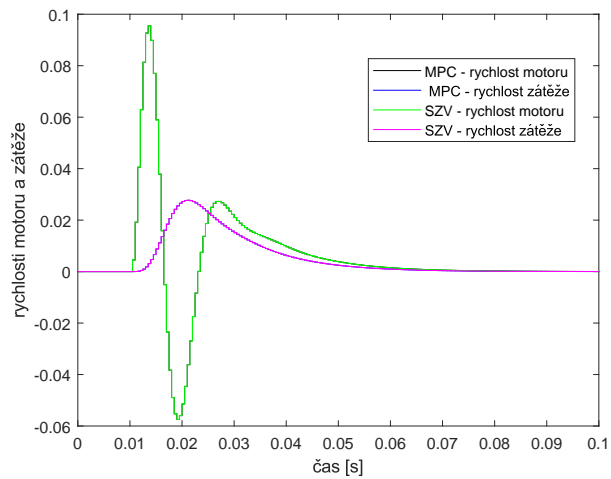
V této podkapitole proběhne analýza robustnosti navrženého MPC regulátoru bez omezení. Jak bylo ukázáno v kapitole 3.4, tak v případě, kdy na MPC kompenzátor nejsou kladena žádná omezení, lze MPC nahradit ekvivalentní stavovou zpětnou vazbu. Následně je možné provést analýzu robustnosti regulační smyčky se vzniklým stavovým regulátorem, který by měl zaručovat totožné řízení jako původní MPC regulátor.

Použitím postupu uvedeného v kapitole 3.4 byl navržen stavový regulátor, který by měl svým chováním odpovídat MPC kompenzátoru bez omezení při řízení soustavy s rezonančním poměrem r_1 . Otestování, zda vytvořený stavový regulátor skutečně odpovídá původnímu MPC regulátoru bez omezení, zde proběhne formou srovnání chování výstupů a akční veličiny při odezvě na skok. Zapojení v Simulinku, které zde bude použito k simulaci řízení MPC regulátorem je na obrázku 7. Schéma zapojení, které bude využito k ověření řízení stavovým regulátorem je na obrázku 2. Průběhy poloh při řízení MPC regulátorem bez omezení a jemu odpovídajícím stavovým regulátorem vyšly takto:



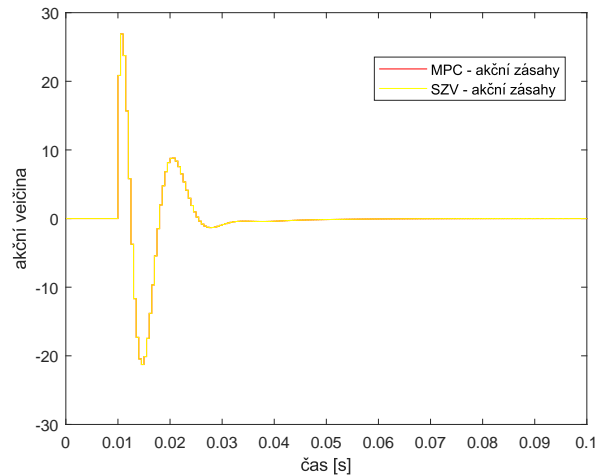
Obrázek 11: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_1 - polohy

Sledované rychlosti se při simulaci vyvíjely takto:



Obrázek 12: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_1 - rychlosti

Během simulace MPC regulátor a stavový kompenzátor generovaly následující akční zásahy:



Obrázek 13: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_1 - zásahy regulátorů

Jak je vidět, tak ve všech 3 předchozích grafech si sledované veličiny při regulaci MPC a stavovým kompenzátozem zcela odpovídají. Malé neshody v některých okamžicích jsou nejspíše způsobeny drobnou nepřesností při výpočtech algoritmu prediktivního řízení a lze je tedy zanedbat.

Nyní, když je ověřena shodnost MPC regulátoru bez omezení a stavové zpětné vazby, již lze dle vztahu (55) vypočítat přenos otevřené smyčky a tím pádem index robustnosti ve stabilitě pro uvažovanou regulační smyčku. Jeho hodnota je $s_m = 0.5158$. Tento parametr udává míru, jak moc lze pozměnit amplitudu nebo fázi $F_o(j\omega)$, aniž by došlo ke destabilizaci uzavřené smyčky. Uvedené hodnotě robustnosti ve stabilitě odpovídá maximální peak citlivostní funkce $S(j\omega)$ o velikosti $M_s = 1.9386$. Tato hodnota maximální velikost peaku citlivostní funkce tedy ještě splňuje standardní požadavek $M_s < 2$.

Pro úplnost by zde bylo vhodné ještě uvést zbylé 2 indexy bezpečnosti ve stabilitě. První z nich se označuje jako bezpečnost v zesílení a je definována jako multiplikativní konstanta, kterou lze změnit zesílení otevřeného regulačního obvodu K tak, aby se uzavřený regulační obvod stal nestabilním. Jeho hodnota pro uvažovaný regulační systém vyšla $g_m = \infty$. Dalším indexem je bezpečnost ve fázi. Ta určuje, jaké fázové zpoždění je potřeba zavést do otevřeného regulačního obvodu, aby se uzavřená smyčka stala nestabilní. Pro regulátor uvažovaný v téhle podkapitole vyšla bezpečnost ve fázi $p_m = -150^\circ$.

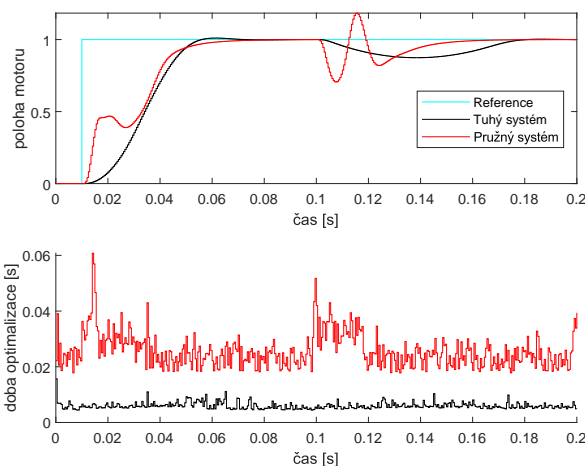
4.3.3 Časová náročnost solveru *quadprog* při řízení pružného a pevného systému

V této podkapitole proběhne analýza časové náročnosti kvadratické optimalizace prováděné solverem *quadprog*. Je to z toho důvodu, že doba řešení optimalizační úlohy je v reálném čase pro praktické aplikace zásadní. Prediktivní regulátor totiž musí stihnout všechny výpočty v jednom kroku regulace za dobu, která je menší než perioda vzorkování řízeného systému. V opačném případě by samozřejmě regulace navrženým MPC nebyla možná.

Nyní bude odzkoušeno, jak dlouho optimalizace solverem *quadprog* trvá při řízení pevného a pružného systému. Za pružný systém bude uvažován ten s poměrem rezonancí r_1 a řídit ho bude regulátor navržený v této kapitole. Jako pevný systém bude používán ten s přenosovou funkcí (60). Pro tento tuhý systém tedy nebudou uvažovány žádné pružné módy a bude uvažován jako dokonale tuhý. MPC regulátor, který ho bude řídit, byl navržen tak, aby bylo

dosaženo podobné kvality regulace jako pro systém s pružnou zátěží. Schéma zapojení, které bude použito k zjištění časové náročnosti optimalizace je na obrázku 7.

Doba trvání jedné optimalizace bude měřena v každém kroku algoritmu při regulaci pevného a pružného systému. Časová náročnost optimalizačních úloh bude počítána v simulaci, kde dojde ke skokové změně referenční hodnoty a kde po ustálení výstupu zasáhne na systém vstupní porucha. Výsledné průběhy poloh a časové náročnosti solveru *quadprog* při simulaci vyšly takto:



Obrázek 14: Solver *quadprog* - doba optimalizace při řízení pevného a pružného systému (r_1)

Je vidět, že časová náročnost optimalizace byla při řízení pružného systému v průměru čtyřikrát vyšší. V samotném průběhu je pak u časové náročnosti řízení pružného systému vidět, že doba trvání optimalizace byla o trochu vyšší při přechodovém ději a při regulaci vstupní poruchy. Doba výpočtu optimalizace při řízení pevného systému byla v průměru po celou dobu simulace rovna přibližně 0.005 vteřiny, což je desetkrát více než byla uvažována doba vzorkování řízeného systému. Nicméně celkově výsledek dopadl dle očekávání. Bylo možné totiž předpokládat, že vzhledem ke složitějšímu modelu řízeného systému s pružnými módy dojde k obtížnějšímu řešení optimalizační úlohy, čímž se prodlouží doba výpočtu.

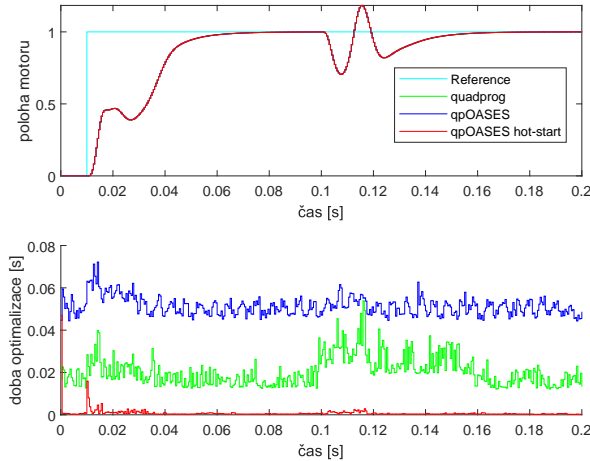
4.3.4 Časová náročnost optimalizační úlohy při použití solveru *qpOASES*

Jak již bylo naznačeno v předchozí podkapitole, tak rychlost řešení optimalizační úlohy je při prediktivní regulaci klíčová. Všechny výpočty v jednom kroku algoritmu totiž musí kompenzátor při praktické aplikaci stačit vyřešit za dobu, která je menší než zadaná doba vzorkování řízeného systému. Dobře odladěných solverů kvadratické optimalizace existuje celá řada. Z tohoto důvodu princip jejich činnosti nebude rozebírán.

V této podkapitole proběhne srovnání průměrné doby řešení optimalizačních úloh u tří solverů při prediktivní regulaci pružného systému s rezonančním poměrem r_1 . Prvním bude solver, který je standardně implementovaný v Matlabu a existuje pro něj příkaz *quadprog*. Druhý solver se jmenuje *qpOASES*, který je volně stažitelný (viz kapitola 2.1.4) a bylo ho potřeba do Matlabu nainportovat. Třetí solver je rovněž *qpOASES*. Nicméně v odlišné variantě, tzv. hot-start, která využívá mezivýsledky optimalizace z předchozího kroku.

Nyní proběhne simulace řízení prostřednictvím MPC regulátoru navrženého v této kapi-

tole pro systém s rezonančním poměrem r_1 . Simulace bude probíhat tak, že nejprve dojde ke skokové změně reference a po ustálení systému zasáhne na řízenou soustavu vstupní porucha. Tato simulace proběhne třikrát a to vždy za použití jiného solveru z předchozího odstavce. Měřit se bude poloha motoru a jak dlouho požitému solveru trvalo při regulaci vyřešit optimalizační úlohu. Výsledek simulace dopadl takto:



Obrázek 15: Srovnání solverů - řízení systému s rezonančním poměrem r_1

Jako první je dobré si všimnout, že průběhy poloh motoru při simulaci vyšly pro všechny 3 solvery totožné. Je tedy vidět, že solvery pracují správně a dávají stejný výsledek. Ovšem odlišné jsou doby výpočtů optimalizace. Je patrné, že zdaleka nejlépe je na tom solver *qpOASES* s hot-start variantou, kde průměrně trvalo jedno řešení optimalizační úlohy 0.00081 sekundy. Tento výsledek je pouze o 0.00031 sekundy delší než je použitá perioda vzorkování systému. Dále pak solveru *quadprog* trvalo řešení úlohy průměrně 0.02 vteřiny. Nejdéle pak optimalizační úlohu řešil solver *qpOASES* v klasické variantě a to průměrně trochu přes 0.05 vteřiny, což je stokrát více než použitá perioda vzorkování.

Pro přehlednost všechny dosažené výsledky z této podkapitoly zobrazuje následující tabulka:

Solver	Max. doba řešení úlohy [s]	Prům. doba řešení úlohy [s]
quadprog	0.0539	0.02
qpOASES	0.0722	0.0512
qpOASES hot-start	0.0469	0.00081

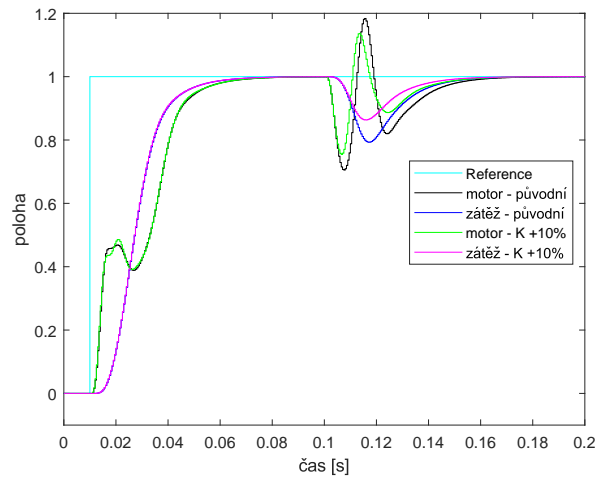
4.3.5 Ověření robustnosti regulátoru

Zde bude odzkoušeno, jak se na kvalitě regulace MPC kompenzátorem navrženého pro systém s poměrem rezonancí r_1 projeví změna některých parametrů řízené soustavy. Samotný regulátor ale zůstane beze změny. Jeho parametry tedy budou stejné jako v celé této kapitole. Konkrétně bude odzkoušeno, jak je uvažovaný regulátor robustní při změně zesílení řízené soustavy o 10 % a při posunutí polohy rezonance r_1 o 10 %. Změna v zesílení pak může simulovat neurčitost v aktuátoru. Posunutá rezonance může symbolizovat změnu vlastností mechaniky, která vznikla například přechodem do jiného pracovního bodu, kde je rezonance jinde.

Analýza robustnosti zde proběhne formou simulace, kdy nejdříve bude regulován původní systém beze změny a posléze systém upravený. Průběhy sledovaných veličin se následně srovnají v grafech. Samotná simulace bude probíhat tak, že nejdříve dojde ke skokové změně referenční hodnoty a po ustálení do systému zasáhne vstupní porucha.

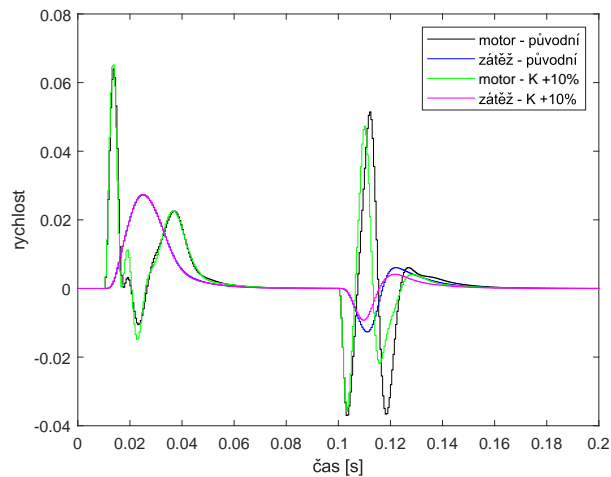
Změna zesílení

Při změně zesílení vyšly průběhy poloh následovně:



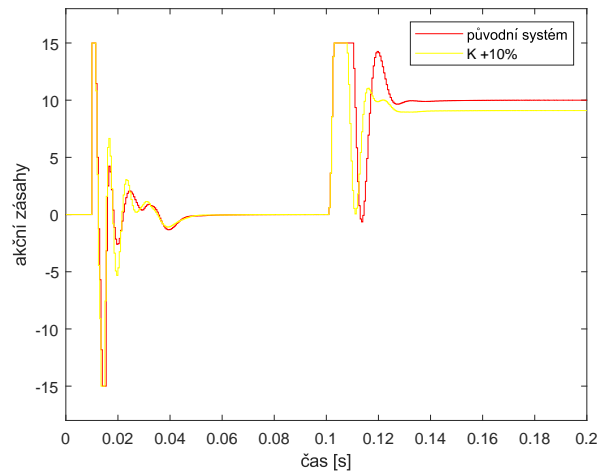
Obrázek 16: Regulace systému s r_1 při změně zesílení - poloha

Je vidět, že se větší zesílení projevilo především při regulaci poruchy, kde dokázalo více utlumit peaky. Rychlosti motoru a zátěže se při tom vyvíjely takto:



Obrázek 17: Regulace systému s r_1 při změně zesílení - rychlost

U přechodového děje je patrné, že u rychlostí se zvětšení zesílení projevilo většími kmity. Naopak ale u poruchy zesílení dokázalo vliv poruchy více utlumit. MPC regulátor během simulace generoval následující akční zásahy:

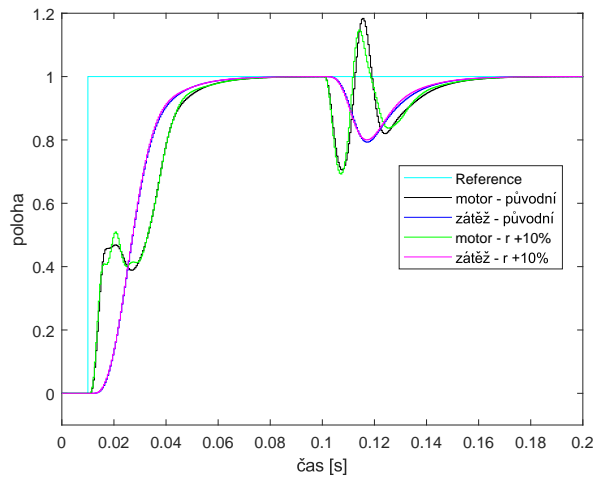


Obrázek 18: Regulace systému s r_1 při změně zesílení - zásahy

Je zřejmé, že v přechodovém ději se u řízeného systému s větším zesílením zvětšilo i kmitání akční veličiny a peaky tak nabývají vyšších hodnot. Celkově se dá tvrdit, že změna zesílení o 10 % nezpůsobila zásadně jiné chování regulačního obvodu a při této odlišnosti řízeného systému lze MPC považovat za dostatečně robustní.

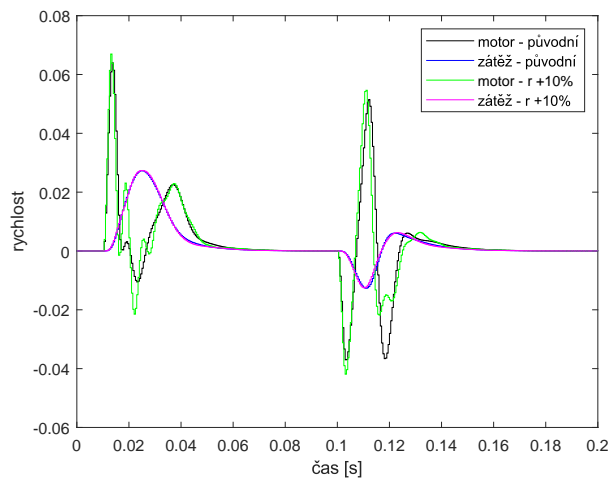
Posunutí polohy rezonance

Pro posunutou polohu rezonance vyšly průběhy poloh následovně:



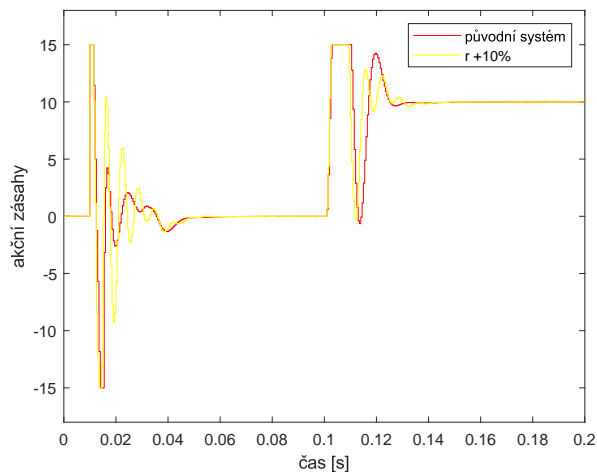
Obrázek 19: Regulace systému s r_1 při změně polohy rezonance - poloha

Při přechodovém ději je vidět, že změna polohy rezonance způsobila větší kmit u polohy motoru. Nicméně aplikovanou změnou v rezonanci bylo dosaženo utlumení působení poruchy v čase 0.1 s. Rychlost motoru a zátěže se při simulaci vyvíjely následujícím způsobem:



Obrázek 20: Regulace systému s r_1 při změně polohy rezonance- rychlost

V přechodovém ději vidět, že změna rezonance způsobila větší peaky u polohy motoru. Nicméně při kompenzaci poruchy se vlivem této změny jeden peak rychlosti motoru v čase 0.12 utlumil hodně. Akční zásahy generované MPC regulátorem měly tento průběh:



Obrázek 21: Regulace systému s r_1 při změně polohy rezonance - zásahy

Je zřejmé, že v přechodovém ději se změna rezonance projevila generováním akčních zásahů s většími hodnotami. MPC regulátoru tedy dalo větší práci dostat systém do nové polohy. U kompenzace poruchy v čase 0.1 měly akční zásahy pro oba případy řízených systémů velice podobné průběhy.

Dosažené výsledky by se daly shrnout, že MPC regulátor je z hlediska posunutí polohy rezonance celkem robustní. Změnou tohoto parametru bylo dosaženo trochu odlišné regulace, nicméně její kvalita byla i tak poměrně vysoká.

4.4 Návrh MPC regulátoru pro systém s poměrem rezonancí r_2

V této podkapitole bude realizován již samotný návrh prediktivního řízení polohy pro soustavu s poměrem rezonancí r_2 . I tady by ale bylo nejdříve vhodné přesně zadefinovat řízený systém ve stavové reprezentaci. Diskrétní stavový model řízené soustavy s rezonančním poměrem r_2 a periodou vzorkování $T_s = 0.0005$ s má následující podobu:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.5155 & -1.513 & -0.2088 & -0.7759 & -2.933 & -1.684 & 0.182 \\ -0.5502 & 0.7394 & 0.1073 & 0.3242 & 1.16 & 0.7489 & 0.0301 \\ 1.27 & -1.144 & 0.5984 & -0.1026 & -1.529 & -0.8642 & 0.05908 \\ -0.09554 & -0.3133 & -0.0982 & -0.3251 & -0.5375 & -0.408 & -0.03541 \\ 0.691 & 1.306 & 0.1835 & 1.315 & 1.516 & 1.132 & -0.07524 \\ 0.2494 & 0.4532 & 0.05337 & 0.5494 & 0.7612 & 0.573 & -0.03239 \\ 0.6972 & -0.7819 & 0.6664 & -0.2587 & -1.171 & -0.6763 & 0.9938 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} -0.0424 \\ 0.03176 \\ -0.01201 \\ -0.000923 \\ 0.06143 \\ 0.0317 \\ -0.009452 \end{bmatrix} \mathbf{u}_k. \quad (76)$$

$$\mathbf{y}_k = \begin{bmatrix} -0.004381 & 0.00352 & 0.001352 & 0.002273 & -0.007558 & 0.005922 & 0.0002782 \\ -0.001866 & -0.003958 & -0.0003593 & -0.005367 & 0.01393 & -0.003353 & -0.0003168 \\ 5.717 \cdot 10^{-6} & -6.066 \cdot 10^{-6} & -6.075 \cdot 10^{-7} & 2.091 \cdot 10^{-6} & 3.631 \cdot 10^{-6} & 2.542 \cdot 10^{-6} & -1.333 \cdot 10^{-5} \\ -6.554 \cdot 10^{-6} & 9.294 \cdot 10^{-6} & -9.887 \cdot 10^{-6} & 5.095 \cdot 10^{-7} & -4.577 \cdot 10^{-6} & -2.458 \cdot 10^{-6} & 4.75 \cdot 10^{-7} \end{bmatrix} \mathbf{x}_k \quad (77)$$

Vstup modelu je požadovaný proud/moment motoru. První výstup udává polohu motoru, druhý rychlost motoru, třetí výstup reprezentuje polohu zátěže a čtvrtý rychlost zátěže.

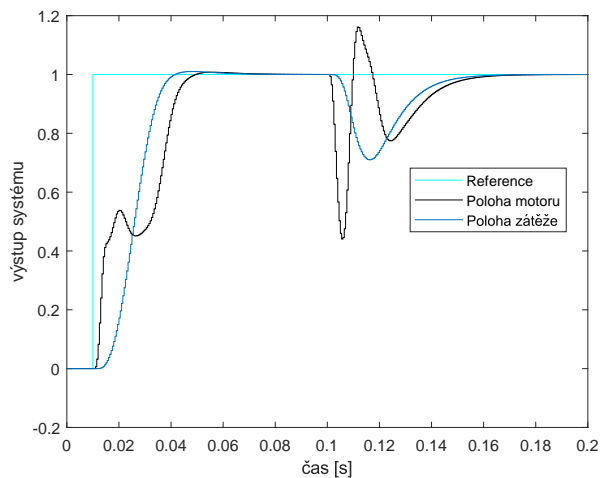
Pro takto definovaný diskrétní systém s rezonančním poměrem r_2 nyní proběhne návrh MPC regulátoru polohy. Ručním odzkoušením bylo nejlepší kvality řízení dosaženo při následujících hodnotách parametrů prediktivního regulátoru. Horizont predikcí má hodnotu $n_p = 200$ a horizont řízení $n_c = 20$. Generovaný akční zásah může nabývat hodnot v rozmezí ± 15 . Nyní se přistoupí k váhovým maticím. Bylo určeno, že matice penalizací \mathbf{Q} a \mathbf{R} mají tvary:

$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, \mathbf{R} = 0.1. \quad (78)$$

Je opět vidět, že důraz je kladen především na chování zátěže. I zde pro snížení výpočetní náročnosti bylo zvoleno, že soft omezení kladená na systém budou uvažována pouze v 10 prvních predikcích. Soft omezení budou pro tento řízený systém nabývat následujících hodnot. K penalizaci bude docházet, když rychlost motoru nebo rychlost zátěže překročí hodnotu ± 0.5 . K penalizaci polohy motoru nebo zátěže dojde, když jejich hodnoty překročí hranici ± 1.2 . I v tomto případě bude kladeno soft omezení na torzi. Zde dojde k penalizaci, když hodnota rozdílu mezi polohou motoru a polohou zátěže překročí mez ± 0.4 . Váhová matice pro všechna omezení bude identická s hodnotami 100 na diagonále.

4.4.1 Průběh regulace

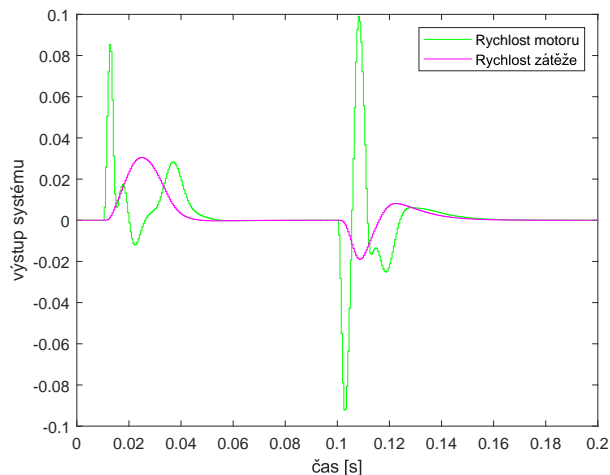
V této podkapitole již proběhne otestování navrženého MPC kompenzátoru při regulaci řízeného systému s poměrem rezonancí r_2 . Pro simulaci se použije zapojení v Simulinku, které je na obrázku 7. Průběhy polohy motoru a zátěže po spuštění simulace dopadly následovně:



Obrázek 22: Průběh poloh motoru a zátěže při regulaci systému s poměrem rezonancí r_2

Z průběhu poloh je vidět, že si regulátor dobře poradil se změnou referenční polohy. Motor i zátěž se tak podařilo dostat do nového stavu. Doba regulace i zde odpovídá přibližně 0.08 vteřiny. V průběhu polohy motoru je pak v přechodovém ději vidět jeden zákmit, který vznikl díky pružné dynamice systému a nejspíše i vlivem soft omezení, která byla kladena na torzi. Průběh polohy zátěže v přechodovém ději vyšel dobře plynulý. V čase 0.1 zasáhla do systému vstupní porucha o velikosti -10. Je patrné, že díky jejímu vlivu se v průběhu polohy vyskytly 2 kmity. Nicméně regulátor si s poruchou poradil a zajistil nulovou odchylku v ustáleném stavu.

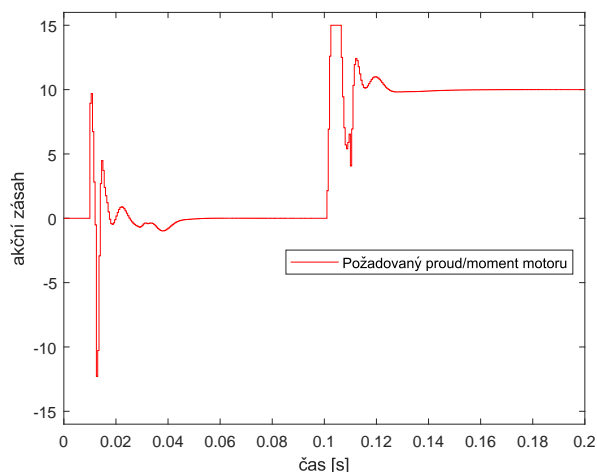
Rychlosti motoru a zátěže při simulaci vyšly následovně:



Obrázek 23: Průběh rychlosti motoru a zátěže při regulaci systému s poměrem rezonancí r_2

Je vidět, že u rychlosti motoru během simulace nastalo několik zákmitů. Především během přechodového děje a při regulování poruchy. Nicméně s tím je nutné při požadavku zaručit rychlou regulaci počítat.

Pro zaručení kvalitní regulace pak navržený MPC kompenzátor generoval při simulaci následující akční zásahy:



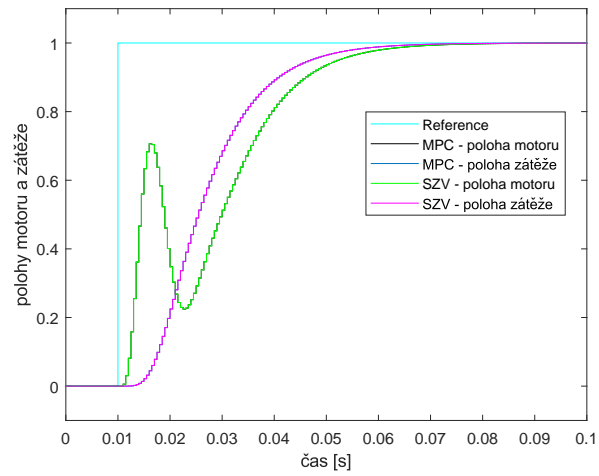
Obrázek 24: Průběh akčního zásahu při regulaci systému s poměrem rezonancí r_2

Kmitavý charakter při simulaci lze pozorovat i u akční veličiny, kterou navržený MPC kompenzátor generoval. I zde lze ale předpokládat, že vzhledem k dynamice systému se kmitání v průběhu akčních zásahů nelze vyvarovat. V grafu je dobře vidět od času 0.1 působení vstupní poruchy, se kterou si regulátor dobře poradil.

4.4.2 Robustnost ve stabilitě

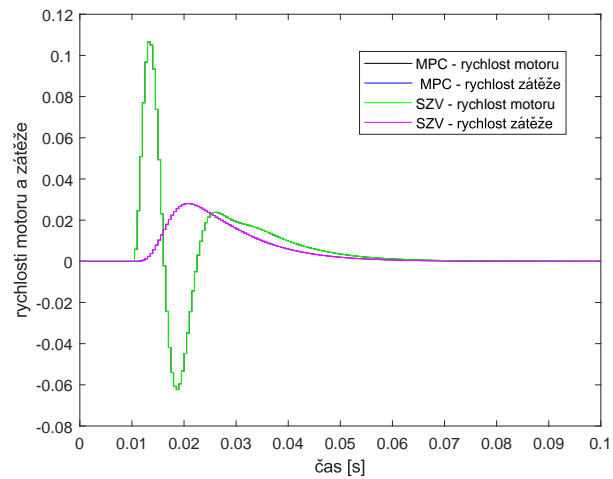
Nyní bude ověřena robustnost ve stabilitě pro regulační smyčku s navrženým MPC regulátorem bez omezení pro řízenou soustavu s poměrem rezonancí r_2 . V kapitole 3.4 byl uveden postup, jak lze MPC bez omezení převést na 3 matice, jejichž vhodným zapojením lze vytvořit stavovou zpětnou vazbu, která bude ekvivalentní původnímu prediktivnímu regulátoru bez omezení. Po získání zmiňovaných matic již lze provést analýzu dynamiky uzavřené smyčky. Nově vzniklý stavový regulátor by pak měl vykazovat při regulaci totožné chování jako původní MPC kompenzátor bez omezení a lze tedy pro něj vypočítat hodnoty standardních indexů bezpečnosti ve stabilitě.

Uvedeným postupem z kapitoly 3.4 byl vytvořen stavový regulátor, který by měl zaručit stejnou regulaci jako původní MPC regulátor bez omezení vytvořený pro řízení systému s rezonančním poměrem r_2 . Nyní by bylo vhodné ověřit, zda si regulátory skutečně odpovídají. Otestování jejich chování při regulaci opět bude provedeno formou simulace jejich odezvy na jednotkový skok. Schéma zapojení pro otestování MPC regulátoru je na obrázku 7 a schéma zapojení pro simulaci chování stavového regulátoru je na obrázku 2. Polohy motoru a zátěže řízeného systému se při skokové změně referenční hodnoty při regulaci MPC a stavovým regulátorem vyvíjely následovně:



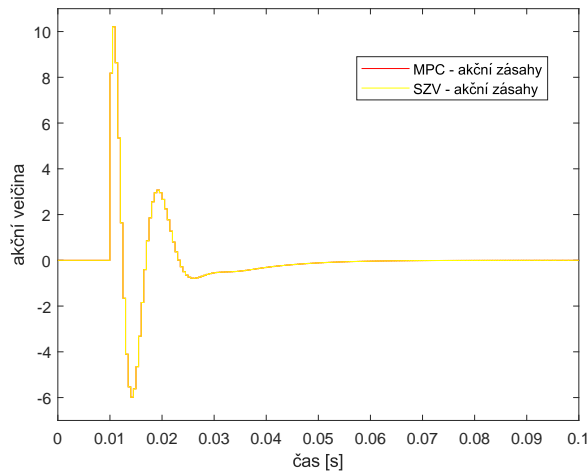
Obrázek 25: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_2 - polohy

Průběhy rychlostí motoru a zátěže při odezvě na jednotkový skok dopadly takto:



Obrázek 26: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_2 - rychlosti

Akční zásahy, které generovaly MPC a stavový regulátor, měly tento průběh:



Obrázek 27: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_2 - zásahy regulátorů

Je patrné, že se sledované veličiny ve třech předchozích grafech téměř dokonale shodují. Ovšem i zde je nutné poznamenat, že v některých okamžicích si průběhy ne zcela odpovídají, nejspíše díky výpočetním nepřesnostem v algoritmu MPC. Zmíněné neshodnosti je tedy možné zanedbat.

Nyní, když je již ověřena shodnost uvažovaných regulátorů, tak lze vypočítat index bezpečnosti ve stabilitě sledované regulační smyčky. Tento parametr udává míru, jak moc lze změnit amplitudu a fázi $F_o(j\omega)$, aby bylo dosaženo nestabilní uzavřené smyčky. Pro tento případ regulátoru vyšel index robustnosti ve stabilitě $s_m = 0.5546$. Tomu odpovídá maximální peak citlivostní funkce o hodnotě $M_s = 1.8032$, což ještě splňuje standardní požadavek $M_s < 2$.

Pro kompletnost by bylo ještě vhodné uvést hodnoty bezpečnosti v zesílení a ve fázi pro uvažovaný regulátor. Hodnota indexu bezpečnosti v zesílení říká, jakou hodnotou je potřeba vynásobit zesílení otevřené smyčky, aby se výsledný uzavřený regulační obvod stal nestabilním. Pro uvažovaný regulátor a řízený systém vyšla tato hodnota $g_m = \infty$. Dále bezpečnost ve fázi určuje, jaké fázové zpoždění po zavedení do otevřeného regulačního obvodu způsobí jeho nestabilitu po uzavření smyčky. Pro uvažovaný případ index bezpečnosti ve fázi vyšel $p_m = -147^\circ$.

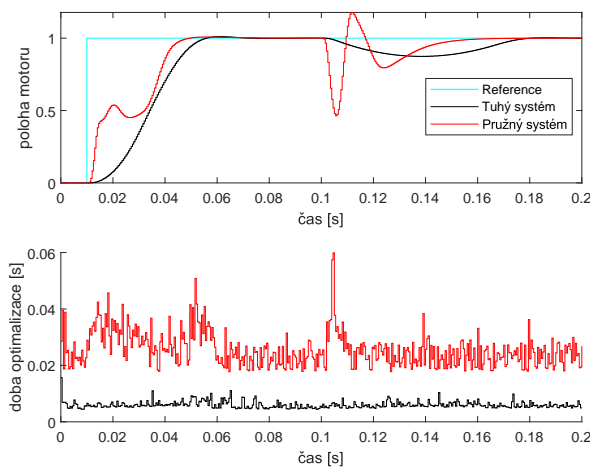
4.4.3 Časová náročnost solveru *quadprog* při řízení pružného a pevného systému

Nyní proběhne analýza časové náročnosti řešení optimalizační úlohy při prediktivní regulaci pevného a pružného systému. Je to z toho důvodu, že v praktických aplikacích je doba trvání výpočtu optimalizace v reálném čase zásadní. Konkrétněji je nutné, aby MPC kompenzátor stihl v jednom kroku algoritmu provést všechny potřebné výpočty. Doba trvání u těchto výpočtů v jednom kroku musí být menší než perioda vzorkování řízeného systému. V případě totiž, že by čas všech výpočtů v jednom kroku algoritmu byl delší než perioda vzorkování řízené soustavy, tak by prediktivní regulaci nebylo možné realizovat.

Tato podkapitola se zaměří na to, jak dlouho bude trvat solveru kvadratických optimalizačních úloh *quadprog* řešení zadaného problému při prediktivní regulaci pevného a pružného systému. Za pružnou řízenou soustavu bude uvažována ta, která je namodelována prostřednictvím rezonančního poměru r_2 . Její regulaci bude zaručovat MPC regulátor, který

byl navržen v této kapitole. Pevný systém bude reprezentovat přenosová funkce (60). Je tedy zřejmé, že v tomto přenosu nefigurují žádné pružné módy a daný model systému je tedy dokonale tuhý. Jeho řízení bude zaručovat MPC regulátor, který byl navržen tak, aby bylo dosaženo podobně kvalitní regulace, jako při řízení soustavy s rezonančním poměrem r_2 . Pro všechny simulace v této podkapitole bude sloužit zapojení ze Simulinku, které je na obrázku 7.

Jak dlouho zabere průměrně solveru *quadprog* řešení jedné optimalizační úlohy při regulaci pevné a pružné soustavy bude ověřováno prostřednictvím simulace. Ta bude probíhat tak, že dojde na začátku ke skokové změně referenční hodnoty a po ustálení výstupů do systému zasáhne vstupní porucha. Při takto uvažované simulaci se dostaly následující průběhy poloh a časových náročností řešení optimalizační úlohy:



Obrázek 28: Solver *quadprog* - doba optimalizace při řízení pevného a pružného systému (r_2)

Z grafu je patrné, že solveru *quadprog* trvalo v průměru pětkrát déle řešení jedné optimalizační úlohy při prediktivním řízení pružného systému než při regulaci tuhé soustavy. Při detailnější analýze průběhu časové náročnosti řešení optimalizace při regulaci pružného systému je vidět, že v okamžicích přechodového děje a při regulaci vstupní poruchy je doba řešení optimalizační úlohy ještě o trochu vyšší. Co se týče doby řešení optimalizace při řízení pevného systému, tak zabral jeden výpočet v průměru 0.005 vteřiny, což je desetkrát více než je zadaná perioda vzorkování řízeného systému. Celkově ale tento výsledek dopadl dle očekávání. Je to z toho důvodu, že pružný systém má celkově složitější dynamiku a tím pádem je i jeho model komplikovanější, což vede na delší dobu řešení optimalizační úlohy.

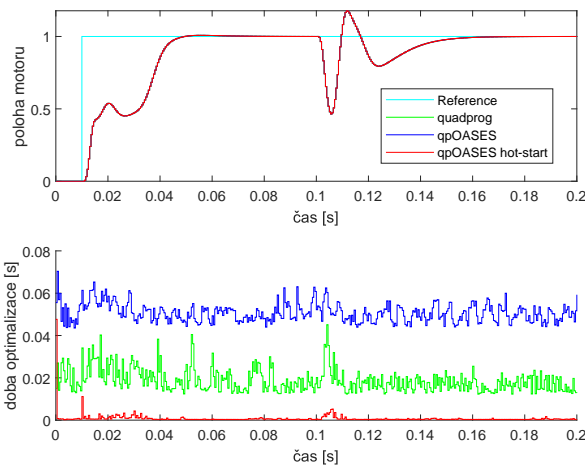
4.4.4 Časová náročnost optimalizační úlohy při použití solveru *qpOASES*

V předchozí podkapitole již bylo uvedeno, že doba řešení optimalizační úlohy při prediktivní regulaci je zásadní. MPC kompenzátor totiž musí v každém kroku algoritmu provést všechny výpočty za dobu, která je menší než perioda vzorkování řízeného systému. V opačném případě by nebylo možné prediktivní regulaci realizovat. K řešení kvadratických optimalizačních úloh již existuje spousta dobře odladěných a popsanych solverů. Z toho důvodu jejich samotná implementace pro tuhle práci nebude probíhat.

Tato podkapitola se zaměří na dobu řešení optimalizační úlohy při prediktivní regulaci

za použití tří různých solverů. Za řízený systém bude uvažován ten s poměrem rezonancí r_2 . První solver, který k regulaci bude použit, je standardně implementovaný v softwaru Matlab a má název *quadprog*. Více o jeho implementaci lze najít na oficiálních stránkách <https://uk.mathworks.com/help/optim/ug/quadprog.html>. Druhý solver je pro jeho použití nutné do Matlabu ručně importovat. Tento program na řešení optimalizačních úloh je volně stažitelný (viz kapitola 2.1.4). Posledním solverem, který bude uvažován, je speciální varianta *qpOASES*, tzv. hot-start. Její výhodou je to, že používá mezivýsledky z předchozích řešení optimalizace, což je u prediktivní regulace velice užitečné.

Nyní bude provedena simulace řízení systému s rezonančním poměrem r_2 pomocí MPC regulátoru, který byl v této kapitole pro tuto soustavu navržen. Jako solvery optimalizačních úloh budou použity ty popsané v předchozím odstavci. Proběhnou tedy 3 simulace řízení pomocí MPC, ale vždy za použití jiného solveru. Při řízení dojde nejprve ke skokové změně referenční hodnoty a posléze, až se výstupy ustálí, zasáhne do systému vstupní porucha. Sledované veličiny budou poloha motoru řízené soustavy a jak dlouho v každém kroku algoritmu solveru trvalo řešení optimalizační úlohy. Výsledek ilustruje následující graf:



Obrázek 29: Srovnání solverů - řízení systému s rezonančním poměrem r_2

Z grafu je na první pohled patrné, že průběh polohy motoru se pro všechny tři použité solvery vyvíjel stejně. Je tedy zřejmé, že každý solver v každém kroku nalezl stejné řešení optimalizační úlohy a do řízené soustavy regulátor aplikoval stejný akční zásah. Odlišné již ale jsou průměrné hodnoty řešení optimalizace pro každý solver. Nejrychleji při tom úlohy dokázal řešit solver *qpOASES* v jeho hot-start variantě. Jemu zabralo jedno řešení optimalizace 0.00079 vteřiny, což je pouze o 0.00029 více, než je uvažovaná perioda vzorkování řízené soustavy. Solver *quadprog* průměrně jednu optimalizační úlohu řešil 0.0187 vteřiny. Nejdéle řešení úlohy trvalo solveru *qpOASES* v klasické variantě. Průměrně mu trvalo trochu přes 0.05 vteřiny než řešení optimalizace nalezl, což je přibližně stokrát více než perioda vzorkování řízené soustavy.

Pro lepší přehlednost byly všechny dosažené výsledky z této kapitoly vloženy do následující tabulky:

Solver	Max. doba řešení úlohy [s]	Prům. doba řešení úlohy [s]
quadprog	0.0451	0.0187
qpOASES	0.0706	0.0507
qpOASES hot-start	0.0476	0.00079

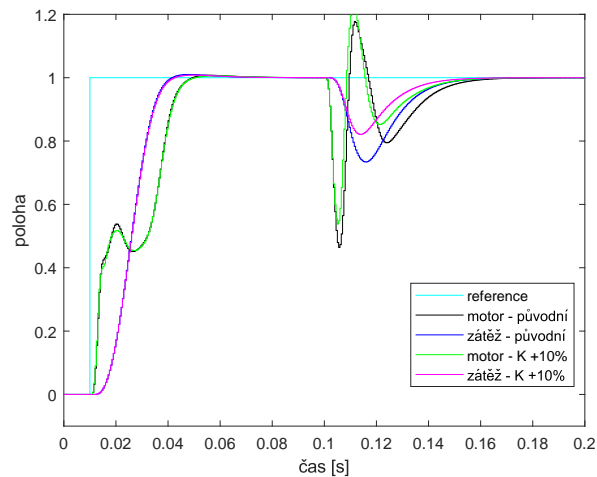
4.4.5 Ověření robustnosti regulátoru

V této podkapitole bude ověřeno, jak moc robustní je MPC regulátor navržený pro systém s rezonančním poměrem r_2 . Konkrétně bude odzkoušeno, jak se projeví na kvalitě regulace, když se změní některý z parametrů řízené soustavy. MPC kompenzátor zůstane ale v původním tvaru, ve kterém byl navržen pro řízenou soustavu beze změn. Robustnost regulátoru bude otestována při změně zesílení řízené soustavy o 10 % a při posunutí polohy rezonance r_2 rovněž o 10 %. Zesílení může nabývat odlišné hodnoty například vlivem neurčitosti v aktuátoru. Změna polohy rezonance může vzniknout například díky odlišným vlastnostem mechaniky, které nastaly přechodem do jiného pracovního bodu, kde je rezonance jinde.

Robustnost MPC regulátoru při změnách v řízeném systému zde bude odzkoušena formou simulace řízení. Nejdříve bude simulováno řízení systému beze změn a poté proběhne simulace regulace řízené soustavy s upravenými hodnotami zesílení a rezonance. Výsledky se následně srovnají v grafech. Simulace bude probíhat tak, že nejprve se skokově změní referenční hodnota a po ustálení výstupů bude do řízené soustavy zavedena vstupní porucha.

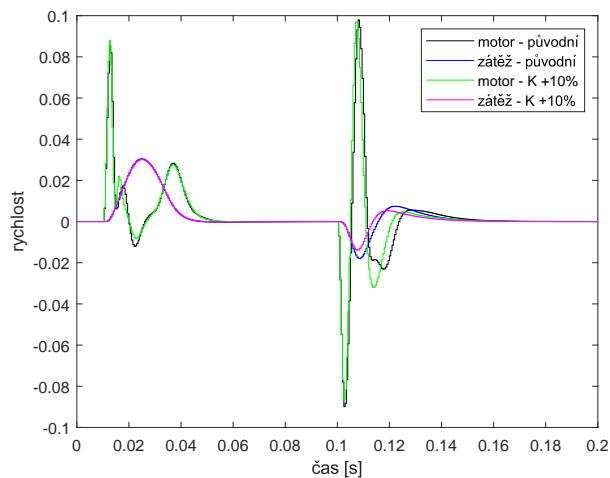
Změna zesílení

Při zvětšené hodnotě zesílení vyšly průběhy poloh následovně:



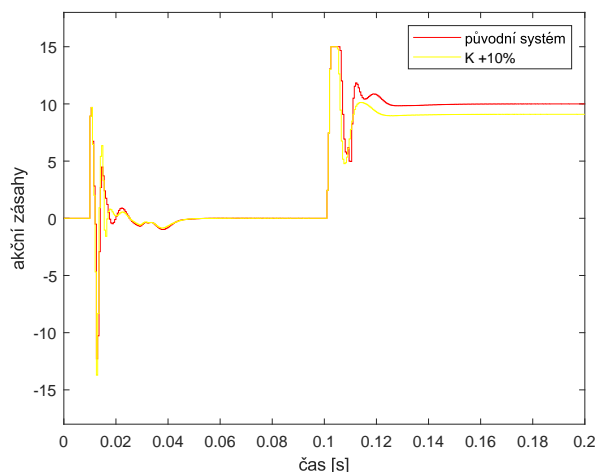
Obrázek 30: Regulace systému s r_2 při změně zesílení - poloha

Z grafu je patrné, že při změně referenční hodnoty byly průběhy poloh pro oba řízené systémy velice podobné. Trochu rozdílného chování bylo dosaženo při regulaci poruchy, kde větší zesílení způsobilo rychlejší kompenzaci poruchy. Rychlosti motoru a zátěže při simulaci nabývaly následujících hodnot:



Obrázek 31: Regulace systému s r_2 při změně zesílení - rychlost

Co se týče rychlostí, tak zde se změna zesílení projevila především a regulace poruchy. Jak je vidět, odlišné zesílení se projevilo například utlumením kmitu v čase 0.12. Akční veličina při simulaci měla následující průběh:



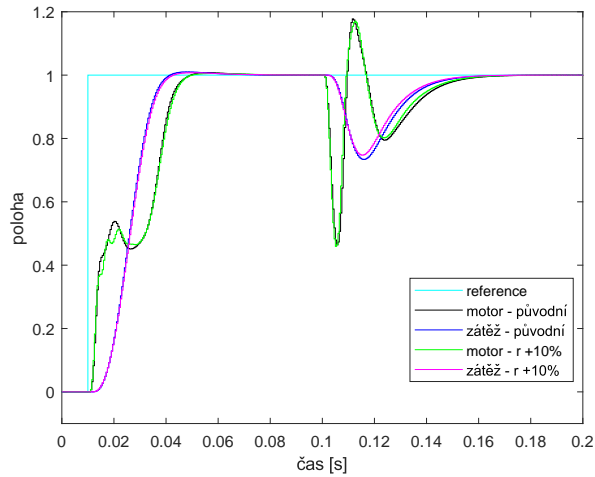
Obrázek 32: Regulace systému s r_2 při změně zesílení - zásahy

Z grafu je patrné, že zvětšení zesílení se u akčních zásahů projevily zvětšením jejich hodnoty především u přechodového děje. Co se týče regulace poruchy, tak měly akční zásahy pro oba řízené systémy průběh podobný.

Po zhodnocení výsledků dosažených po změně zesílení řízené soustavy lze říci, že ačkoliv bylo chování sledovaných veličin upraveného systému trochu jiné, tak stále se dá považovat za poměrně kvalitní. Na základě toho lze říci, že navržený MPC je z hlediska změn zesílení řízeného systému celkem robustní.

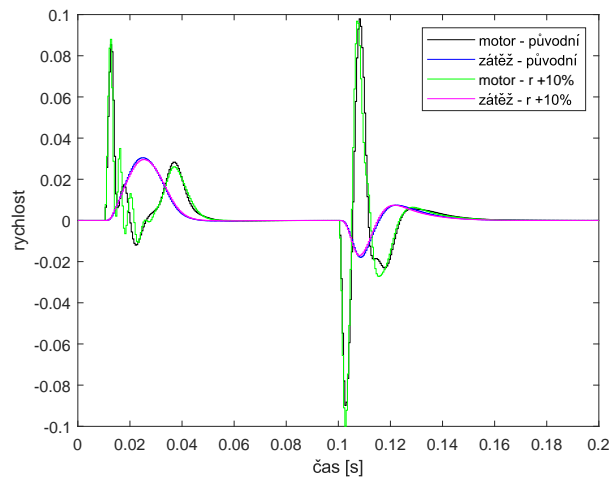
Posunutí polohy rezonance

Při posunutí polohy rezonance o 10 % vyšly průběhy poloh motoru a zátěže následujícím způsobem:



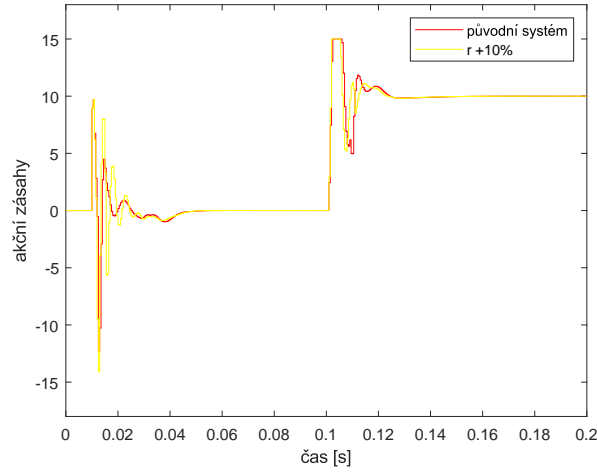
Obrázek 33: Regulace systému s r_2 při změně polohy rezonance - poloha

Jak je vidět, tak bylo dosaženo podobného chování pro oba dva řízené systémy. Jediná zřetelnější odlišnost nastala v čase 0.02 vteřiny, kdy při řízení systému s odlišnou rezonancí měla poloha motoru menší kmit. Rychlosti motoru a zátěže při posunutí polohy rezonance měly následující průběh:



Obrázek 34: Regulace systému s r_2 při změně polohy rezonance- rychlost

Zde už je odlišnost v chování systémů trochu zřetelnější a to zejména u přechodového děje. Je vidět, že poloha motoru u systému s vyšší rezonancí více kmitá. Nicméně průběhy rychlosti při regulaci polohy si jsou u obou systémů v celku podobné. MPC kompenzátor při regulaci generoval akční zásahy s následujícím průběhem:



Obrázek 35: Regulace systému s r_2 při změně polohy rezonance - zásahy

Průběhy akčních zásahů již vyšly dle očekávání vzhledem k vývoji poloh a rychlostí. Je patrné, že při přechodovém ději měly akční zásahy při regulaci systému s odlišnou rezonancí větší peaky. To by způsobilo vyšší namáhání hřídele. Při regulaci poruchy nicméně regulátor generoval podobné akční zásahy pro oba dva řízené systémy.

Celkově by se daly dosažené výsledky zhodnotit takto. Zvýšení polohy rezonance sice mírně zvětšilo kmitání u systému, nicméně kvalita regulace se dá stále považovat za přijatelnou. Na základě toho lze tvrdit, že pro změnu polohy rezonance je MPC stále dostatečně robustní.

4.5 Návrh MPC regulátoru pro systém s poměrem rezonancí r_3

Nyní proběhne návrh prediktivního regulátoru polohy pro systém s rezonančním poměrem r_3 . Jako v předchozích případech tak i zde bude nejprve uveden stavový model řízené soustavy pro danou hodnotu poměru rezonancí. Diskrétní stavový model řízeného systému s rezonančním poměrem r_3 a s periodou vzorkování $T_s = 0.0005$ s má tvar:

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.6391 & 0.3864 & -0.0001042 & -0.002243 & -0.005801 & 0.1773 & 0.2805 \\ -0.9824 & -0.2279 & -0.1187 & 0.001437 & 0.004668 & -0.2069 & -0.212 \\ -0.1546 & 0.2399 & -0.09251 & -0.04205 & -0.002601 & 0.05173 & 0.1317 \\ -0.8776 & -0.2118 & 0.03028 & 1.022 & -0.1361 & 1.183 & 0.9502 \\ 0.1798 & 0.02671 & 0.05324 & -0.6749 & 0.7102 & 0.01247 & 0.0358 \\ -2.772 & 0.01388 & -0.0005413 & -0.01735 & -0.02942 & 1.269 & 1.517 \\ -3.002 & 0.006463 & 0.001332 & -0.01093 & -0.01197 & 0.7288 & 1.148 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} -0.1157 \\ 0.03624 \\ 0.003677 \\ 0.06707 \\ -0.01278 \\ 0.1659 \\ 0.1769 \end{bmatrix} \mathbf{u}_k. \quad (79)$$

$$\mathbf{y}_k = \begin{bmatrix} 0.007861 & -0.0307 & 0.01071 & -0.003658 & -0.003679 & 0.0002104 & 0.01213 \\ -0.008786 & 0.04406 & -0.008354 & 0.001754 & 0.001196 & 0.01282 & 0.008634 \\ -7.176 \cdot 10^{-6} & 1.496 \cdot 10^{-5} & -1.098 \cdot 10^{-6} & 3.866 \cdot 10^{-5} & -5.342 \cdot 10^{-5} & -2.175 \cdot 10^{-5} & -5.853 \cdot 10^{-6} \\ 2.238 \cdot 10^{-5} & -3.137 \cdot 10^{-5} & -2.246 \cdot 10^{-6} & 3.743 \cdot 10^{-5} & 1.104 \cdot 10^{-5} & 3.052 \cdot 10^{-5} & -4.375 \cdot 10^{-6} \end{bmatrix} \mathbf{x}_k. \quad (80)$$

Vstupem modelu je požadovaný proud/moment motoru. První výstup označuje polohu motoru, druhý rychlost motoru, třetí výstup udává polohu zátěže a čtvrtý rychlost zátěže.

Nyní, když je již systém přesně definovaný, tak proběhne návrh MPC regulátoru polohy. Při experimentálním odzkoušení bylo dosaženo nejlepšího chování uzavřené smyčky při následujících parametrech MPC regulátoru. Hodnota predikčního horizontu zde je $n_p = 90$

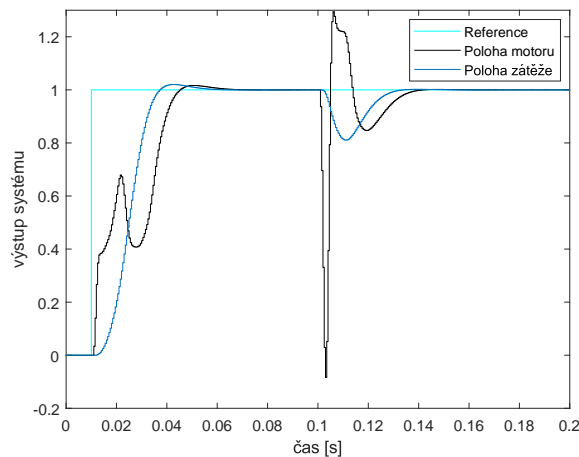
a hodnota horizontu predikcí $n_c = 10$. Dále bude uvažováno tvrdé omezení na generovanou akční veličinu a to ± 15 . Matice penalizující hodnoty výstupů \mathbf{Q} a matice penalizující hodnotu vstupů mají pro uvažovaný řízený systém následující tvary:

$$\mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 0 & 0 & 15 \end{bmatrix}, \mathbf{R} = 0.5. \quad (81)$$

Je kladen důraz především na chování zátěže. Soft omezení i pro tento případ budou pro snížení výpočetní náročnosti uvažována v predikcích pouze v prvních deseti krocích. Hodnoty polohy motoru a polohy zátěže budou penalizovány, když překročí mez ± 1.2 . Rychlosti motoru a zátěže budou penalizovány při překročení omezení ± 0.5 . Bude kladeno soft omezení na torzi (rozdíl mezi polohou motoru a zátěže). Omezení pro tuto hodnotu bude ± 0.4 . Váhová matice pro všechna omezení zde bude opět identická s hodnotami 100 na diagonále.

4.5.1 Průběh regulace

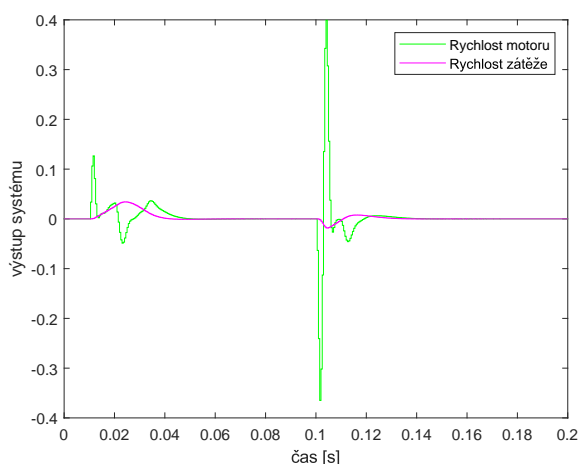
Nyní proběhne otestování nevrženého MPC regulátoru v Simulinku při řízení systému s rezonančním poměrem r_3 , pro který byl navržen. Zapojení v Simulinku, které bude použito k simulaci regulační smyčky, je na obrázku 7. Průběhy polohy a zátěže při regulaci MPC kompenzátozem dopadly takto:



Obrázek 36: Průběh poloh motoru a zátěže při regulaci systému s poměrem rezonancí r_3

Co se týče přechodového děje, tak jako v předchozích případech je vidět u průběhu polohy jeden zákmit. Průběh polohy zátěže je při odezvě na skok dobře plynulý. Doba regulace odpovídá 0.07 vteřiny. Nicméně regulace poruchy, která začala působit v čase 0.1, již nedopadla moc dobře. V průběhu polohy motoru je vidět velký podkmit, což by v některých aplikacích mohlo vést k destrukci hřídele. Při snaze odstranit tento podkmit bylo odzkoušeno mnoho změn parametrů v navrženém MPC regulátoru, ale žádná z nich nevedla k o moc lepšímu výsledku. Při použití tohoto kompenzátozem v praxi by tedy bylo nutné brát v potaz ne zcela ideální regulaci poruch.

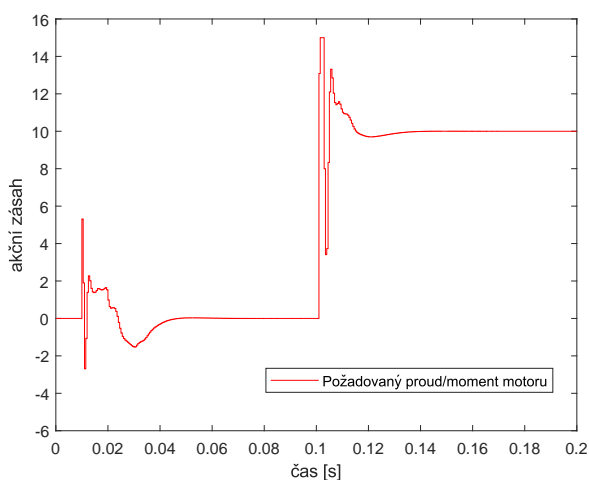
Během simulace vyšly průběhy rychlosti motoru a zátěže následujícím způsobem:



Obrázek 37: Průběh rychlosti motoru a zátěže při regulaci systému s poměrem rezonancí r_3

I zde jsou vidět v průběhu rychlosti motoru během simulace kmity. Především při snaze regulovat vliv poruchy. Prudké změny rychlosti motoru by mohly vést ke snížení životnosti motoru. Nicméně průběh rychlosti zátěže je dobře plynulý.

MPC kompenzátor při simulaci generoval akční zásahy následovně:



Obrázek 38: Průběh akčního zásahu při regulaci systému s poměrem rezonancí r_3

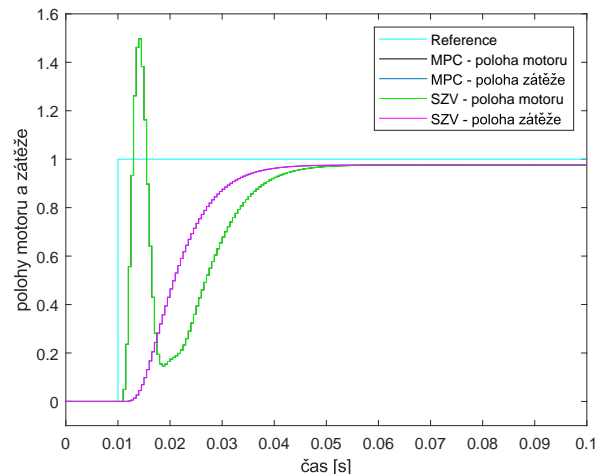
Co se týče průběhu akční veličiny, tak i zde lze pozorovat pružný charakter soustavy. MPC regulátor tak generoval akční zásahy, jejichž hodnota se plynule měnila tak, aby bylo zaručeno kvalitní regulace. V čase 0.1 je opět vidět, že regulátor dobře vyrušil vliv poruchy a zaručil tak nulovou odchylku výstupu od reference v ustáleném stavu.

4.5.2 Robustnost ve stabilitě

V této podkapitole bude ověřena hodnota indexu robustnosti regulační smyčky s MPC regulátorem bez omezení, který byl navržen pro systém s rezonančním poměrem r_3 . V kapitole

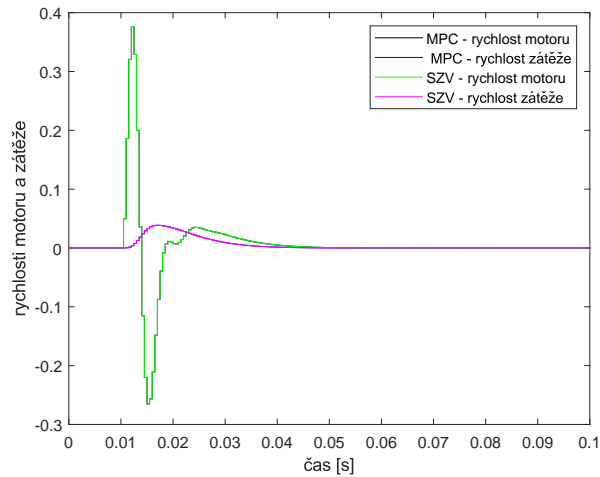
3.4 byl uveden postup, jak je možné MPC regulátor bez omezení převést na ekvivalentní stavovou zpětnou vazbu realizovanou prostřednictvím tří matic. S využitím tří matic, které by měly zaručit stejnou regulaci jako původní MPC bez omezení, lze provést analýzu výsledné regulační smyčky. Je tedy možné mimo jiné i vypočítat standardní hodnoty indexů robustnosti ve stabilitě. Tyto indexy by pak měly mít stejnou hodnotu jako při regulaci původním MPC bez omezení. Právě na získání hodnoty bezpečnosti ve stabilitě se zaměří tato podkapitola.

Po aplikování postupu z kapitoly 3.4 byly získány 3 matice. Prostřednictvím nich lze vytvořit stavovou zpětnou vazbu, která by měla zaručit totožnou regulaci jako původní MPC regulátor bez omezení vytvořený pro řízenou soustavu s rezonančním poměrem r_3 . Ještě před samotnou analýzou robustnosti by bylo ale vhodné otestovat, zda si původní MPC regulátor bez omezení a z něj vytvořený stavový regulátor odpovídají. Jejich shodnost bude opět ověřena prostřednictvím simulace jejich odezvy na změnu referenční hodnoty. Schéma zapojení, které bude využito k simulaci regulace MPC kompenzátozem je na obrázku 7 a zapojení ze Simulinku, které bude použito k simulaci chování stavového regulátoru, je na obrázku 2. Při odezvě na jednotkový skok se při regulaci MPC a stavovým regulátorem získaly tyto průběhy polohy motoru a zátěže:



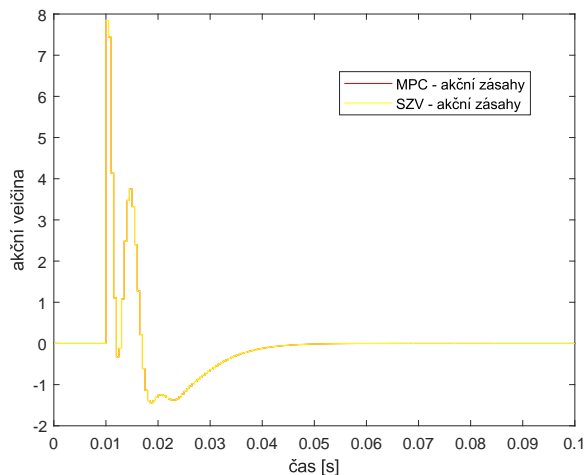
Obrázek 39: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_3 - polohy

Rychlosti motoru a zátěže se při řízení MPC bez omezení a stavovým kompenzátozem vyvíjely takto:



Obrázek 40: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_3 - rychlosti

MPC regulátor a stavový kompenzátor během simulace pak generovaly následující akční zásahy:



Obrázek 41: MPC vs. ekvivalentní stavový regulátor - řízení systému s r_3 - zásahy regulátorů

Je zřejmé, že i zde se průběhy sledovaných veličiny téměř dokonale shodují. Jejich drobnou neshodu v některých okamžicích lze opět nejspíše přisuzovat nepřesností výpočtů v algoritmu prediktivní regulace. Malé nepřesnosti je tedy možné opomenout.

Na základě předchozích 3 grafů lze tedy říci, že uvažovaný stavový regulátor svým chováním odpovídá původnímu MPC regulátoru. Nyní je možné opět za využití vztahu (55) získat přenos otevřené smyčky, prostřednictvím kterého lze pak získat hodnotu robustnosti ve stabilitě. Tento index určuje míru, jak moc lze pozměnit zesílení a fázi $F_o(j\omega)$, aby se regulační smyčka stala nestabilní. Pro uvažovaný regulátor vyšla tato hodnota bezpečnosti ve stabilitě $s_m = 0.5985$. Tomu odpovídá maximální peak citlivostní funkce $S(j\omega)$ o velikosti $M_s = 1.6709$. I pro tento případ kompenzátoru maximální peak $S(j\omega)$ splňuje obecný požadavek $M_s < 2$.

Aby byl popis vlastností výsledného regulačního obvodu kompletní, tak by zde bylo

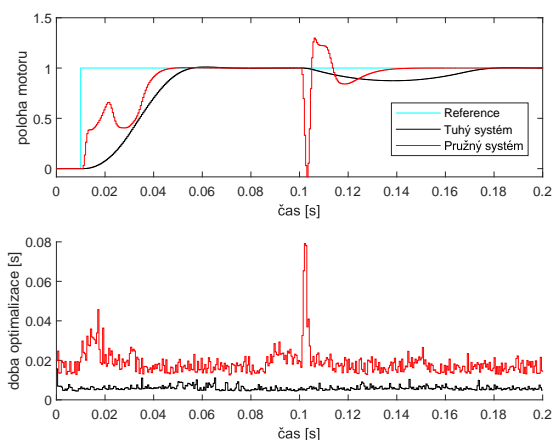
vhodné uvést ještě hodnoty indexů bezpečnosti v zesílení a ve fázi. Hodnota bezpečnosti v zesílení říká, jakou konstantou je nutné vynásobit zesílení otevřeného regulačního obvodu, aby se smyčka po uzavření stala nestabilní. Pro uvažovaný regulátor s řízenou soustavou index bezpečnosti v zesílení vyšel $g_m = \infty$. Index bezpečnosti ve fázi definuje hodnotu fázového zpoždění, které po zavedení do otevřené smyčky způsobí nestabilní uzavřený regulační obvod. Pro tenhle případ systému s regulátorem index bezpečnosti ve fázi vyšel $p_m = -142^\circ$.

4.5.3 Časová náročnost solveru *quadprog* při řízení pružného a pevného systému

Tato podkapitola se zaměří na analýzu časové náročnosti kvadratické optimalizace prováděné solverem *quadprog* při prediktivním řízení pevného a pružného systému. Tato problematika je velice důležitá při aplikaci MPC regulace v praktických úlohách, kde je potřeba řídit řízenou soustavu v reálném čase. V praxi je totiž zapotřebí, aby výpočty prováděné v každém kroku algoritmu prediktivním kompenzátorem trvaly menší dobu než je perioda vzorkování řízené soustavy. V opačném případě, kdy by výpočty kompenzátoru zabraly delší dobu, než je perioda vzorkování, tak by MPC regulátor nebylo možné k řízení použít.

Nyní proběhne otestování, jak dlouho solver *quadprog* implementovaný v Matlabu bude průměrně řešit jednu optimalizační úlohu při MPC regulaci pevného a pružného systému. Za pružný systém bude uvažován ten, který je reprezentován svou přenosovou funkcí s poměrem rezonancí r_3 . Řízení této soustavy bude realizovat MPC regulátor, který byl pro její řízení navržen v této kapitole. Tuhý systém bude uvažován ten, který je namodelován přenosovou funkcí (60). Tento přenos neobsahuje žádné pružné komponenty a lze ho tedy považovat za dokonale pevný. Regulaci této soustavy bude zaručovat MPC regulátor, který byl navržen tak, aby jeho kvalita řízení přibližně odpovídala kvalitě řízení regulátorem, který byl navržen pro systém s rezonačním poměrem r_3 . Schéma zapojení ze Simulinku, které bude používáno k analýze časové náročnosti solveru je na obrázku 7.

Průměrná doba trvání řešení jedné optimalizace při regulaci pevné a pružné soustavy zde bude získána prostřednictvím simulace. Její průběh bude následující. Nejdříve dojde ke skokové změně referenčního signálu a po ustálení výstupů do řízeného systému zasáhne vstupní porucha. Při simulaci s popsáním scénářem vyšly průběhy poloh motorů řízených soustav a časové náročnosti optimalizací následovně:



Obrázek 42: Solver *quadprog* - doba optimalizace při řízení pevného a pružného systému (r_3)

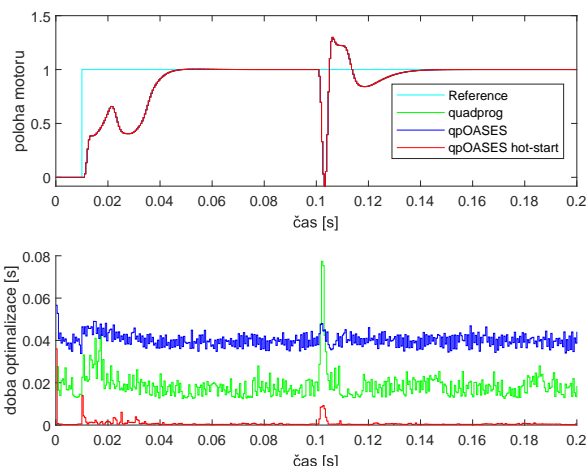
Jak je z výsledných průběhů vidět, tak solveru *quadprog* trvalo v průměru čtyřikrát déle v jednom kroku algoritmu řešení optimalizační úlohy při řízení pružného systému než u dokonale tuhé soustavy. Při podrobnější analýze časového průběhu dob trvání optimalizace při řízení pružné soustavy je patrné, že v době přechodového děje a při začátku působení vstupní poruchy je doba trvání ještě o trochu delší. Dále při prediktivní regulaci pevného systému je z grafu vidět, že průměrně jedno řešení optimalizace trvalo solveru 0.005 vteřiny. Doba vzorkování řízeného systému je přitom desetkrát menší. Nicméně tento výsledek časových náročností šlo již předem očekávat a to z následujícího důvodu. Popis dynamiky pružné soustavy je o něco složitější než model dokonale tuhé soustavy. Modely se například liší i v počtu pólů a tím pádem je i stavová reprezentace pružného systému rozsáhlejší. Na základě toho solver musel řešit obtížnější optimalizační úlohu a doba jejího výpočtu trvala déle.

4.5.4 Časová náročnost optimalizační úlohy při použití solveru *qpOASES*

V předchozí podkapitole již bylo řečeno, že doba řešení optimalizačních úloh při prediktivní regulaci je v praxi zásadní. Je to z toho důvodu, že MPC regulátor musí během jednoho kroku algoritmu stihnou provést všechny potřebné výpočty včetně optimalizace za dobu, která je menší než perioda vzorkování řízené soustavy. V případě, že by čas výpočtu byl delší, nebylo by možné MPC k regulaci použít. K řešení kvadratických optimalizačních úloh již existuje řada dobře odladěných solverů s kvalitní dokumentací, proto již zde samotné implementace solverů optimalizace nebudou analyzovány.

Tato podkapitola bude zaměřena na analýzu průměrné doby řešení jedné optimalizační úlohy za použití tří solverů při řízení systému s rezonančním poměrem r_3 . První program pro řešení optimalizace, který zde bude uvažován, se nazývá *quadprog*. pro jeho použití v Matlabu existuje stejnojmenný příkaz v Matlabu. Další solver, který bude použit v této podkapitole, nese název *qpOASES*. Tento program je již nutné pro použití do Matlabu ručně importovat. Třetí solver, který zde bude odzkoušen, je speciální varianta *qpOASES*, tzv hot-start. Výhodou této modifikované varianty je, že k řešení optimalizace používá mezivýsledky z předchozích kroků, což je možné při prediktivní regulaci dobře využít.

Nyní již k samotnému srovnávání solverů. To bude probíhat prostřednictvím tří simulací, kdy bude pomocí MPC regulátoru řízen systém s rezonančním poměrem r_3 , ovšem pokaždé bude použit jiný solver. Samotná simulace bude probíhat tak, že nejprve dojde ke skokové změně referenční hodnoty a po ustálení výstupů do systému zasáhne vstupní porucha. Sledované veličiny při simulaci budou poloha motoru a doba řešení optimalizační úlohy v každém kroku. Výsledek simulace zobrazuje následující graf:



Obrázek 43: Srovnání solverů - řízení systému s rezonančním poměrem r_3

I zde je vidět, že průběhy polohy motoru při simulaci vyšly pro každý solver totožně. MPC regulátor tak v každém kroku algoritmu našel stejné řešení optimalizační úlohy a tím pádem u každého solveru v každém kroku byl aplikován stejný akční zásah. Liší se ovšem časová náročnost optimalizace při regulaci u každého solveru. S přehledem zde na tom byl nejlépe solver *qpOASES* v hot-start variantě. V průměru mu řešení jedné optimalizační úlohy trvalo 0.00077 vteřiny. To je pouze o 0.27 milisekundy déle než je použitá perioda vzorkování u řízené soustavy. Solveru *quadprog* v průměru nalezení řešení optimalizační úlohy trvalo 0.0186 vteřiny. Nejhorší z hlediska časové náročnosti na tom je solver *qpOASES* v klasické variantě. Tomuto solveru trvalo v průměru 0.04 vteřiny, než řešení optimalizační úlohy nalezl.

Pro zpřehlednění všech výsledků dosažených v této podkapitole byla vytvořena tabulka:

Solver	Max. doba řešení úlohy [s]	Prům. doba řešení úlohy [s]
quadprog	0.0774	0.0186
qpOASES	0.0566	0.0402
qpOASES hot-start	0.036	0.00077

4.5.5 Ověření robustnosti regulátoru

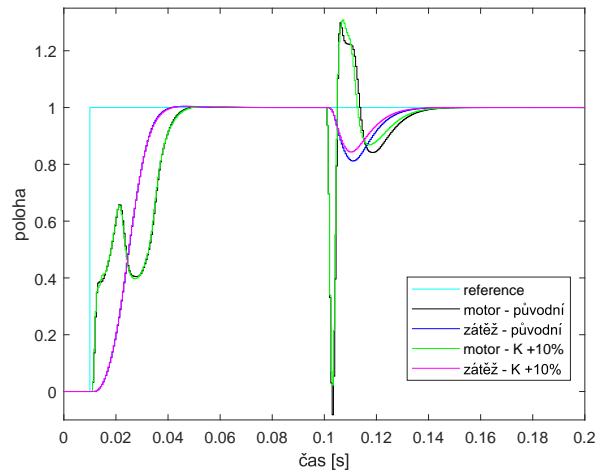
Nyní bude odzkoušeno, jak moc je MPC kompenzátor navržený pro systém s poměrem rezonancí r_3 robustní vzhledem ke změnám v řízené soustavě. Konkrétně bude analyzováno, jak se na kvalitě řízení projeví, když se změní zesílení systému a poloha rezonance, přičemž MPC kompenzátor zůstane ve své původní podobě beze změny. Parametry MPC kompenzátoru tedy budou totožné, jako v celé téhle podkapitole, kde byl navržen pro systém s rezonančním poměrem r_3 . Robustnost bude odzkoušena pro případy, když se u řízené soustavy navýší zesílení o 10 % a když se poloha rezonance r_3 rovněž navýší o 10 %. Odlišná hodnota zesílení může simulovat neurčitost v aktuátoru. Poloha rezonance se může posunout například vlivem změn ve vlastnostech mechaniky, které mohou vzniknout přechodem do jiného pracovního bodu, kde je rezonance na jiném místě.

Otestování robustnosti MPC kompenzátoru navrženého pro systém s poměrem rezonancí r_3 zde proběhne formou srovnání průběhu důležitých veličin při simulaci, kdy nejprve bude řízen systém beze změn a posléze bude simulována regulace MPC při odlišných hodnotách

zesílení a polohy rezonance. Simulace bude probíhat tak, že na začátku dojde ke skokové změně reference a po ustálení výstupů zasáhne do systému vstupní porucha.

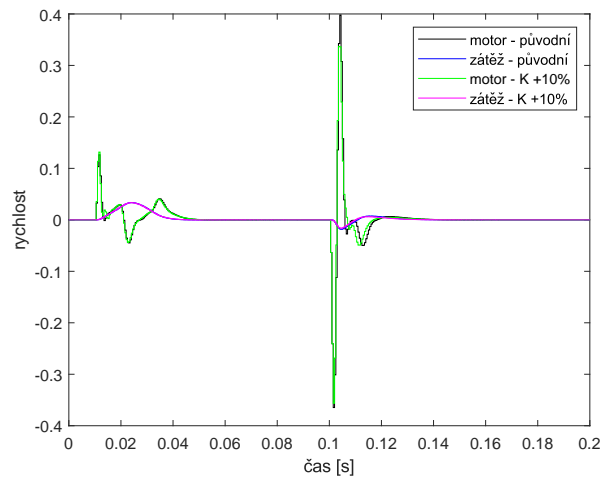
Změna zesílení

Při změně hodnoty zesílení o 10 % vyšly průběhy poloh motoru a zátěže následovně:



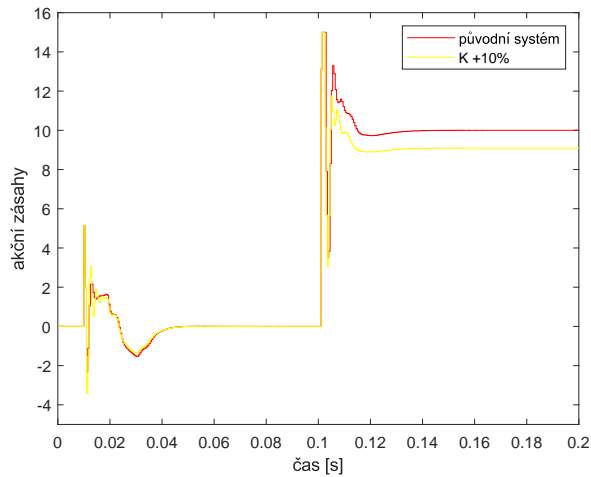
Obrázek 44: Regulace systému s r_3 při změně zesílení - poloha

Je vidět, že po celou dobu simulaci si polohy pro oba řízené systémy poměrně dobře odpovídají. Malé rozdíly by se daly najít u regulace poruchy, kde v čase 0.1 změna zesílení způsobila menší kmit a v čase 0.11 byly polohy motoru také trochu na jiných hodnotách. Rychlosti motoru a zátěže se při změně zesílení vyvíjely tímto způsobem:



Obrázek 45: Regulace systému s r_3 při změně zesílení - rychlost

I zde je patrné, že si sledované rychlosti u obou řízených soustav poměrně odpovídají. Nepatrně odlišné chování pak nastalo u rychlostí motoru při regulaci poruchy. MPC kompenzátor při simulaci generoval následující zásahy do systému:



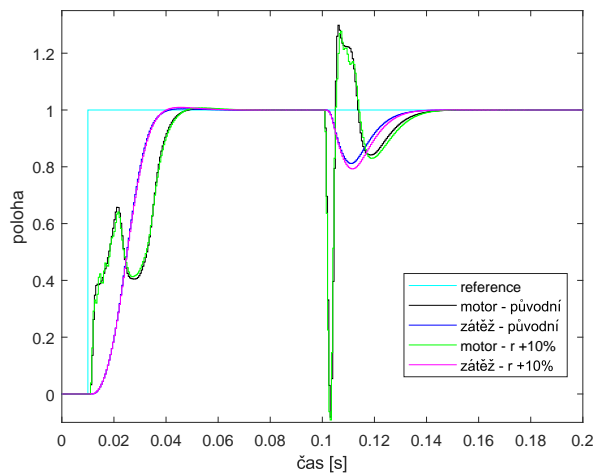
Obrázek 46: Regulace systému s r_3 při změně zesílení - zásahy

U přechodového děje je patrné, že změna zesílení se projevila tím, že akční zásahy jsou více kmitavé. Nicméně oproti původnímu systému o moc více nekmitají. Dále při regulaci poruchy je vidět, že kompenzátor generoval podobné akční zásahy pro oba řízené systémy.

Po shrnutí dosažených výsledků lze tvrdit, že zvětšení hodnoty zesílení se projevilo větším kmitáním u sledovaných veličin. Nicméně kvalita regulace je stále uspokojující a lze tedy tvrdit, že navržený regulátor je vzhledem ke změně zesílení poměrně robustní.

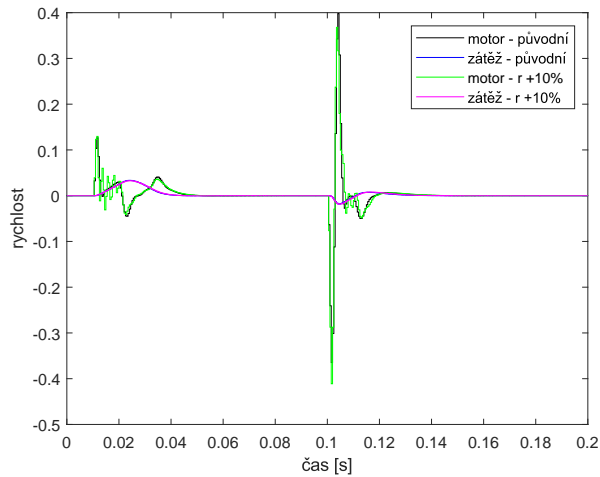
Posunutí polohy rezonance

Při navýšení polohy rezonance o 10 % vyšly průběhy poloh motoru a zátěže při simulaci takto:



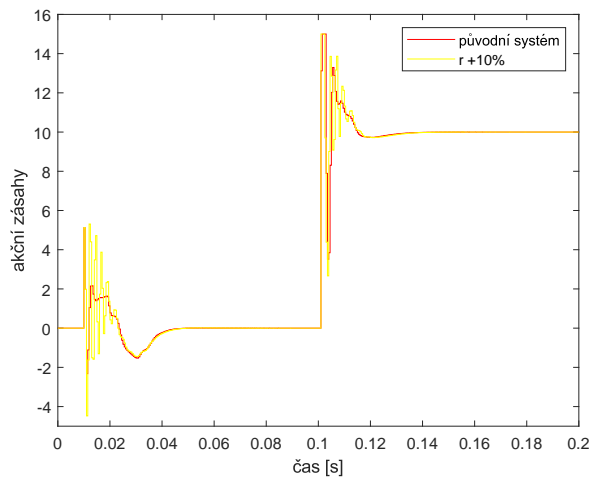
Obrázek 47: Regulace systému s r_3 při změně polohy rezonance - poloha

U přechodového děje je patrné, že poloha motoru má tendenci kmitat. Lze předpokládat, že při zvyšující se změně zesílení by kmity nabývaly vyšších hodnot. Co se týče regulace poruchy, tak zde si polohy motoru i zátěže pro oba řízené systémy vcelku odpovídají. Rychlosti motoru a zátěže se při tom vyvíjely takto:



Obrázek 48: Regulace systému s r_3 při změně polohy rezonance- rychlost

U průběhu rychlostí je již více patrný kmitající charakter motoru. Při praktických aplikacích by tato změna rezonance způsobila i větší namáhání hřídele, což by vedlo na kratší životnost elektromotoru. U poruchy si nicméně rychlosti motoru i zátěže u obou systémů poměrně odpovídají. MPC regulátor během simulace generoval následující akční zásahy:



Obrázek 49: Regulace systému s r_3 při změně polohy rezonance - zásahy

Zde je již zřejmé, co se dalo z předchozích grafů již očekávat. Jak je vidět, tak u přechodového děje MPC regulátor při snaze uregulovat systém s odlišnou rezonancí generoval velice kmitavé akční zásahy. Jak již bylo řečeno, tak to by znamenalo větší namáhání jak aktuátoru, tak i celé řízené soustavy. U regulace poruchy lze rovněž pozorovat kmitání při řízení systému s posunutou polohou rezonance.

Navýšení hodnoty rezonance vedlo na mnohem kmitavější charakter všech sledovaných veličin. To by znamenalo mnohem vyšší mechanické namáhání celé soustavy a nejspíše by se tak zkrátila její životnost. Celkově se tedy dá říci, že zde MPC regulátor v testu robustnosti při změně polohy rezonance moc neobstál.

5 Srovnání MPC s konvenčními regulátory

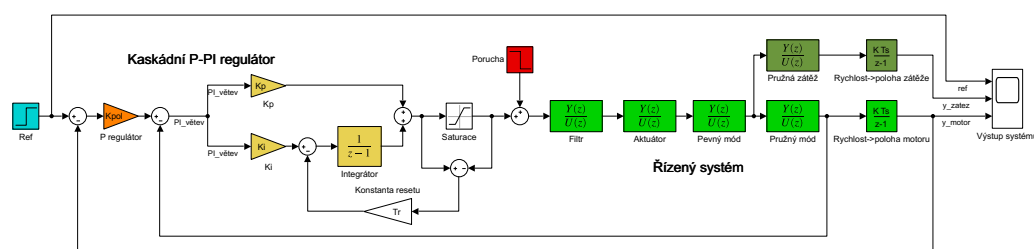
V této kapitole proběhne srovnání navržených MPC regulátorů z kapitoly 4 s kaskádním P-PI regulátorem a LQR (Linear-quadratic regulator). Této úloze se budou věnovat následující 3 podkapitoly. V první z nich bude srovnáno MPC a kaskádní regulace. Druhá podkapitola bude zaměřena na srovnání MPC a řízení pomocí LQR. V obou případech budou tyto strategie řízení porovnávány s MPC následujícím způsobem. Nejdříve ale budou uvedena schémata, která budou sloužit k simulaci řízení kaskádního regulátoru a LQR. Poté se následující 3 podkapitoly budou věnovat srovnání regulace MPC kompenzátorem a kaskádním kompenzátorem/LQR. Za řízené systémy budou uvažované ty samé systémy co v kapitole 4. Lišit se tedy budou hodnotou svého rezonančního poměru r . Poslední část této kapitoly bude zaměřena na srovnání MPC s kaskádním P-PI regulátorem z hlediska omezení na akční veličinu.

5.1 Srovnání MPC s kaskádní P-PI regulací

Následující podkapitoly se budou věnovat srovnání regulace prostřednictvím MPC kompenzátorem a prostřednictvím kaskádního P-PI kompenzátorem. Nejdříve bude ale uvedeno schéma, které bude použito k simulaci kaskádní regulace. Pro otestování MPC regulátoru bude použito schéma ze Simulinku, které je na obrázku 7. Následovat budou 3 podkapitoly, kde proběhne otestování regulátorů při řízení systémů s různým poměrem rezonancí popsaných v předchozí kapitole. Postupně budou pro každý řízený systém do jednoho grafu vykresleny průběhy akčních zásahů a průběhy výstupů při řízení MPC regulátorem a kaskádním P-PI regulátorem. Výsledky budou následně analyzovány a bude určen ten regulátor, u kterého regulace dopadla lépe.

5.1.1 Schéma v Simulinku použité pro otestování

Zde bude uvedeno zapojení v Simulinku, které bude sloužit pro otestování navržené kaskádní P-PI regulace. Schéma vypadá takto:



Obrázek 50: Schéma pro otestování kaskádní P-PI regulace

Jako první se ve schématu nachází modrý blok udávající referenci. Konkrétně je použit blok Step. Od hodnoty reference se následně odečítá hodnota výstupu systému, což je pro tenhle případ poloha motoru (vnější zpětná vazba). Poté v zapojení figuruje P regulátor polohy (oranžový blok). Následuje vnitřní zpětnovazební smyčka regulující rychlost motoru. Dále je ve schématu vytvořen PI regulátor rychlosti (žluté bloky). Tento regulátor je realizován tak, aby se ošetřilo unášení integrační složky. Jev unášení integrátoru nastává v případech, kdy se generovaný akční zásah dostane na svou mez saturace. Blok pro saturaci se totiž

vyskytuje hned za PI regulátorem. K saturované akční veličině se následně přičítá vstupní porucha (červený blok). Poté je ve schématu namodelován řízený systém, tak jak byl popsán v kapitole 4. Výstupem řízené soustavy je rychlost motoru s polohou motoru (zelené bloky) a rychlost s polohou zátěže (tmavě zelené bloky).

5.1.2 MPC vs. kaskádní regulace systému s poměrem rezonancí r_1

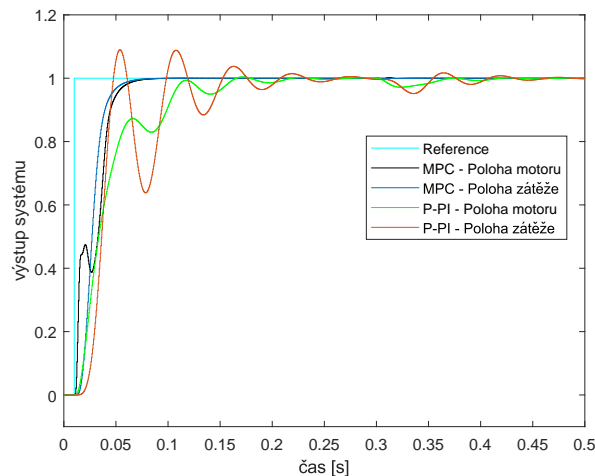
V této části proběhne srovnání navrženého MPC regulátoru s kaskádním P-PI regulátorem při řízení totožného systému s poměrem rezonancí r_1 . MPC kompenzátor bude uvažován ve tvaru, který je uveden v kapitole 4. Co se týče kaskádního regulátoru, tak vnitřní smyčku bude řídit PI regulátor, který má v z-transformaci podobu

$$C_{c1,d}(z) = \frac{61.212(z - 0.9583)}{(z - 1)}. \quad (82)$$

P regulátor polohy pak bude mít velikost 29. Hodnota časové konstanty resetu byla určena na $T_r = 0.22$. Co se týče robustnosti ve stabilitě, tak pro tento PI regulátor vychází $s_m = 0.7369$. Je vidět, že tento kompenzátor je na tom z hlediska robustnosti ve stabilitě lépe než MPC regulátor pro stejný řízený systém. Jinými slovy lze říci, že u MPC regulátoru by stačily menší změny v amplitudě a ve fázi u $F_o(z)$ než u PI kompenzátoru, aby se regulační smyčka stala nestabilní.

Pro srovnání regulace budou sloužit schéma zapojení na obrázcích 7 a 50. Nyní již k samotné simulaci. Ta bude probíhat tak, že nejprve bude simulována změna referenční hodnoty pro polohu motoru a následně do systému začne působit vstupní porucha.

Průběhy poloh motoru a zátěže při řízení MPC a kaskádním P-PI regulátorem dopadly takto:

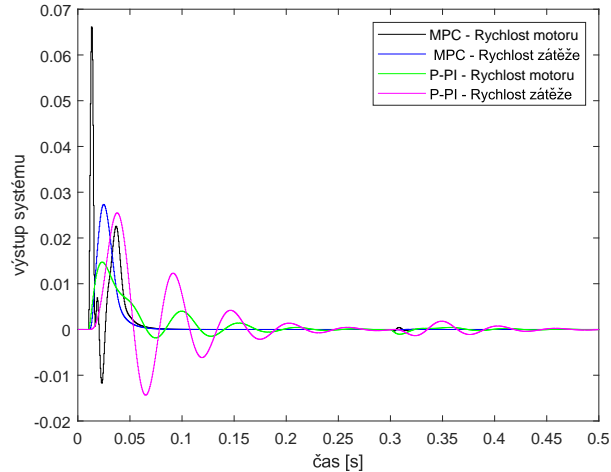


Obrázek 51: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_1 - poloha

Jak je vidět, tak co se týče průběhů poloh, tak je na tom lépe MPC regulátor. Je to z toho důvodu, že doba regulace je u MPC rychlejší a navíc se v průběhu poloh při řízení kaskádní regulací vyskytují značné kmity. V čase 0.3 s do systému zasáhla porucha o velikosti -0.1.

S její regulací si rovněž lépe poradil MPC regulátor. Její vliv totiž v průběhu poloh motoru a zátěže není vůbec poznat. O něco horší regulace poruchy byla u MPC regulátoru. Zde vstupní porucha způsobila výkyv polohy motoru a zátěže z jejich ustálených hodnot.

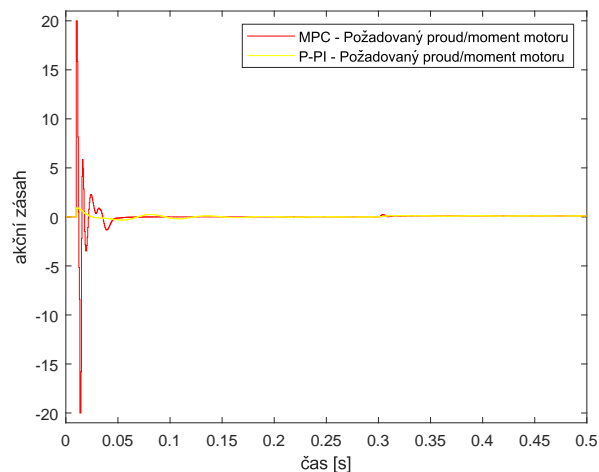
Průběhy rychlostí motoru a zátěže při řízení MPC regulátorem a kaskádním P-PI regulátorem dopadly tímto způsobem:



Obrázek 52: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_1 - rychlost

Zde je vidět, že z hlediska rychlostí motoru a zátěže je na tom lépe MPC regulátor. U přechodového děje sice rychlosti nabývaly vyšších hodnot, ale to vedlo na kratší dobu regulace. U kaskádního regulátoru mají sledované rychlosti hladší průběh, ale celkově více kmitají, což by vedlo ke většímu namáhání hřídele.

Generované akční zásahy MPC regulátorem a kaskádním P-PI regulátorem měly během simulace následující průběh:



Obrázek 53: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_1 - zásahy

I zde je patrné, že je na tom lépe MPC regulátor. Jím generované akční zásahy nabývají mnohem menších hodnot. Akční veličina generovaná kaskádním regulátorem má ve svém

průběhu znatelné kmity.

Shrnutím dosažených výsledků lze tvrdit, že kvalitnější regulace bylo dosaženo u MPC regulátoru. Kompenzace poruchy byla o něco horší u P-PI kaskádního regulátoru a rovněž i doba regulace byla delší u kaskádního řízení.

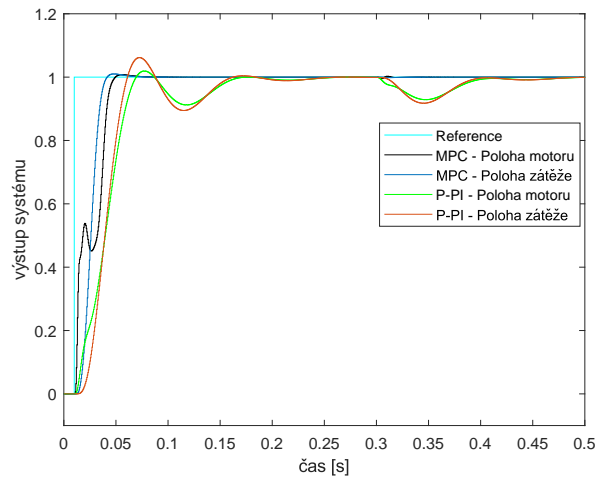
5.1.3 MPC vs. kaskádní regulace systému s poměrem rezonancí r_2

Zde proběhne otestování regulátorů při řízení systému s rezonančním poměrem r_2 . MPC kompenzátor bude mít stejný tvar, jako v kapitole 4. Pro srovnání kvality řízení byl navržen kaskádní P-PI regulátor, jehož přenos je

$$C_{c2,d}(z) = \frac{43.523(z - 0.9831)}{(z - 1)}. \quad (83)$$

Zesílení P regulátoru, který řídí polohu, zde bude 30. Hodnota časové konstanty resetu pro vyrušení vlivu unášení integrační složky je $T_r = 0.22$. Pro tento regulátor vyšel index robustnosti ve stabilitě $s_m = 0.7339$. I tady platí, že z hlediska robustnosti je na tom lépe navržený PI regulátor než MPC pro stejný řízený systém. K destabilizaci regulační smyčky s PI kompenzátozem by tedy byly zapotřebí větší změny amplitudy a fáze $F_o(j\omega)$ než u MPC regulátoru.

Zapojení ze Simulinku, která budou použita pro otestování, jsou na obrázcích 7 a 50. Při simulaci regulace MPC a kaskádním regulátorem při změně referenční hodnoty a při vlivu poruchy vyšly následující průběhy poloh motoru a zátěže:

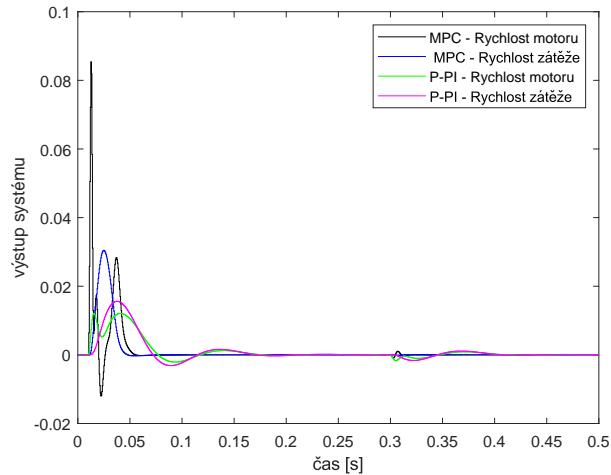


Obrázek 54: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_2 - poloha

Jak je vidět, tak bylo dosaženo obdobného výsledku jako pro předchozí poměr rezonancí. Co se týče přechodového děje, tak bylo dosaženo lepšího výsledku u MPC regulátoru. Při regulaci P-PI kaskádním kompenzátozem se pak v přechodovém ději u poloh motoru a zátěže vyskytují 2 kmity. I doba regulace je kratší u MPC. Dále při regulaci poruchy, jejíž vliv začal působit v čase 0.3 s hodnotou -0.1, tak na tom byl lépe opět MPC regulátor. Vliv poruchy byl okamžitě vyrušen a v průběhu poloh není porucha vůbec patrná. To se nedá

řící o regulaci kaskádním kompenzátorem. Zde dalo regulátoru mnohem větší námahu vliv poruchy vykompenzovat.

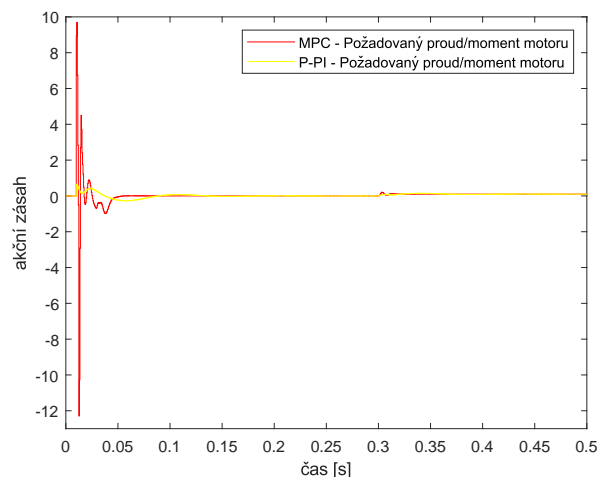
Průběh rychlostí motoru a zátěže vyšly při uvažované simulaci následovně:



Obrázek 55: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_2 - rychlost

Je patrné, že je na tom lépe opět MPC regulátor. Hodnoty rychlostí motoru a zátěže byly u něj při přechodovém ději vyšší, ale důsledkem toho je kratší doba regulace. To by samozřejmě mělo vliv na opotřebení součástek v elektromotoru a nejspíš by se tím zkrátila i jejich životnost. Plynulejších změn rychlostí s menší hodnotou bylo dosaženo u kaskádní regulace. Nicméně přechodový děj je díky tomu pomalejší.

Akční zásahy regulátorů při simulaci měly následující průběh:



Obrázek 56: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_2 - zásahy

Generované akční zásahy u MPC nabývají v přechodovém ději mnohem vyšších hodnot, díky čemuž je doba regulace poměrně rychlá. U kaskádní regulace má průběh akční veličiny mnohem hladší průběh s menšími peaky.

Shrnutím dosažených výsledků z této podkapitoly lze dojít k závěru, že pro řízení systému

s rezonančním poměrem by bylo lepší zvolit MPC regulátor. Nicméně muselo by se počítat s tím, že díky jeho agresivitě by docházelo k většímu fyzickému namáhání motoru.

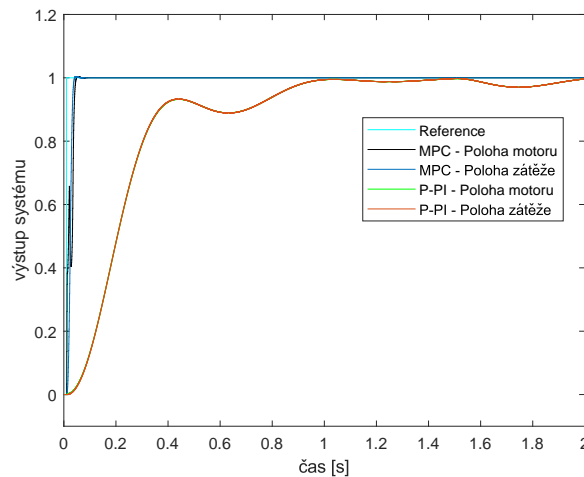
5.1.4 MPC vs. kaskádní regulace systému s poměrem rezonancí r_3

Jako poslední zde bude srovnána kvalita řízení MPC a kaskádním regulátorem při řízení systému s rezonančním poměrem r_3 . Tvar MPC kompenzátoru bude stejný, jako byl uvažován v kapitole 4. Co se týče kaskádního P-PI regulátoru, tak jeho tvar v z-transformaci byl pro uvažovaný řízený systém zvolen takto:

$$C_{c3,d}(z) = \frac{4.3321(z - 0.9937)}{(z - 1)}. \quad (84)$$

Hodnota zesílení polohového regulátoru byla určena na $P = 4$. Časová konstanta pro vyrušení jevu unášení integrátoru má hodnotu $T_r = 0.22$. Index bezpečnosti ve stabilitě pro tento regulátor vyšel $s_m = 0.6467$. To je o trochu vyšší hodnota než u MPC regulátoru pro stejný řízený systém. I pro tento případ lze říci, že k dosažení nestabilní regulační smyčky u navrženého PI regulátoru by bylo potřeba větších změn v amplitudě a ve fázi $F_o(z)$ než u MPC.

Schémata ze Simulinku, která byla použita k simulaci regulace uvažovanými regulátory, jsou na obrázcích 7 a 50. I zde simulace bude probíhat tak, že nejprve dojde ke změně referenční polohy motoru a po ustálení výstupů zasáhne do řízeného systému porucha. Průběhy poloh při uvažované simulaci při řízení MPC a kaskádním regulátorem dopadly takto:

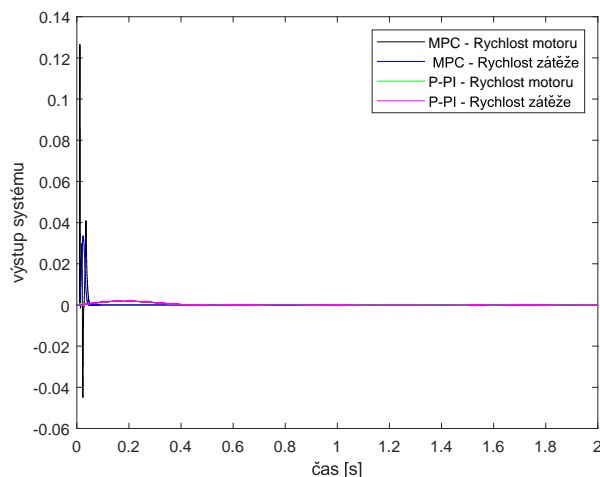


Obrázek 57: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_3 - poloha

Je patrné, že při řízení kaskádním P-PI regulátorem bylo dosaženo velice pomalé regulace. Doba regulace je u MPC kompenzátoru mnohem kratší. Průběh výstupu sice při řízení kaskádním regulátorem kmitá jen trochu, ale vše kazí pomalá doba regulace. Nicméně vzhledem ke kmitavé dynamice řízené soustavy již nešlo kaskádní regulátor navrhnout o moc lépe. V čase 1.5 vteřiny pak do řízeného systému zasáhla vstupní porucha o velikosti -0.1. U řízení P-PI kaskádním regulátorem to způsobilo velký kmit u polohy motoru i u polohy zátěže.

Vliv poruchy u MPC zde byl ihned vyregulován. Celkově pomalejší dynamiky bylo tedy dosaženo při regulaci poruchy opět u kaskádního regulátoru.

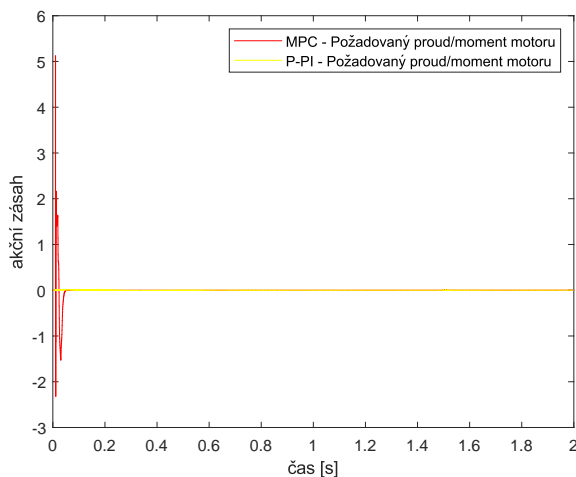
Rychlosti motoru a zátěže měly při simulaci následující průběh:



Obrázek 58: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_3 - rychlost

Je vidět, že rychlosti motoru a zátěže nabývaly mnohem vyšších hodnot u řízení MPC regulátorem. Nicméně to vedlo na mnohem kratší dobu regulace. U kaskádního řízení rychlosti nabývají mnohem menších hodnot ale za cenu pomalejší dynamiky v přechodovém ději.

Během simulace regulátory generovaly následující akční zásahy:



Obrázek 59: MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_3 - zásahy

Je vidět, že průběhy akčních veličin vyšly dle očekávání. U kaskádního regulátoru má průběh akčních zásahů mnohem hladší průběh s menšími peaky. Co se týče MPC, tak zde nebyvaly akční zásahy vyšších hodnot a regulátor prostřednictvím nich dokázal docílit rychlejší regulace.

Celkově by se daly výsledky shrnout tak, že i pro systém s poměrem rezonancí r_3 by bylo vhodnější zvolit vytvořený prediktivní regulátor. Jeho doba regulace je mnohem kratší a i

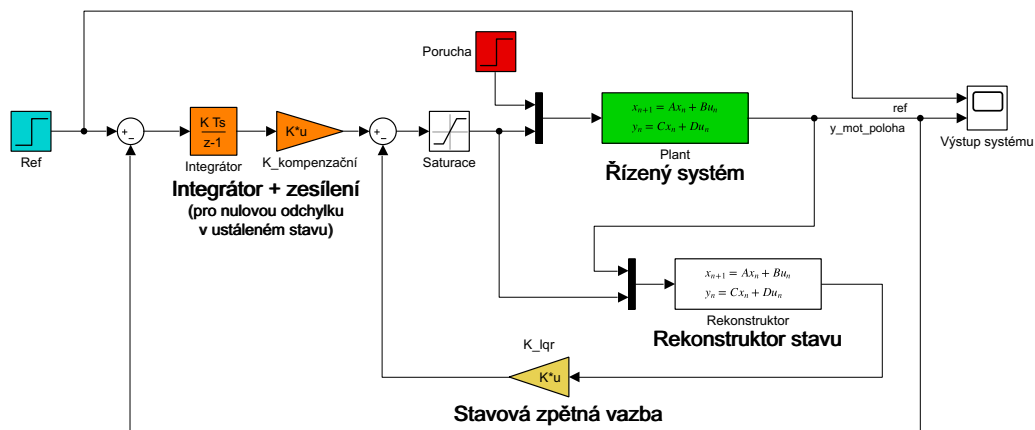
odezva na poruchu je kvalitnější.

5.2 Srovnání MPC s regulací pomocí LQR

Následující čtyři podkapitoly se budou věnovat srovnávání MPC regulátorů navržených v předchozí kapitole s LQR při řízení totožných systémů. Nejdříve bude uvedeno schéma, které bude sloužit k simulaci regulace prostřednictvím LQR. Pro otestování regulace MPC kompenzátorem poslouží schéma ze Simulinku na obrázku 7. Další 3 podkapitoly se již budou věnovat samotnému srovnávání MPC regulátoru a LQR při řízení systémů s různým rezonančním poměrem z předchozí kapitoly. MPC regulátory se rovněž použijí v tom stejném tvaru, v jakém byly navrženy v kapitole 4. Srovnávání kompenzátorů pak bude probíhat tak, že se vykreslí do prvního grafu průběhy akčních zásahů a do druhého průběhy výstupů regulovaných systémů. Výsledné grafy následně budou analyzovány a pro každý řízený systém bude vybrán ten regulátor, u kterého dopadla regulace lépe.

5.2.1 Schéma v Simulinku použité pro otestování

Schéma v Simulinku, které bude použito k simulaci regulace pomocí LQR, vypadá takto:



Obrázek 60: Schéma pro otestování regulace pomocí LQR

V levé části se nachází blok Step, který bude generovat referenční signál se skokovou změnou hodnoty. Následuje první zpětná vazba a to od výstupu. Ta by prostřednictvím integrátoru a kompenzačního zesílení (oranžové bloky) měla zaručit nulovou odchylku v ustáleném stavu a to včetně při působení poruchy. Následuje stavová zpětná vazba, ve které se nachází blok pro zesílení realizující LQR (žlutý blok). Signál pro tuto zpětnou vazbu ústí z rekonstruktoru stavu, který odhaduje hodnoty všech stavů řízeného systému. Za stavovou zpětnou vazbou se nachází blok pro saturaci akční veličiny. Následně je do regulační smyčky zavedena porucha (červený blok). Ta společně se saturovanou akční veličinou slouží jako vstupy do řízené soustavy (zelený blok). Je vidět, že vstup a výstup řízeného systému slouží jako vstup do rekonstruktoru, který odhaduje stavy řízené soustavy pro stavový regulátor. Rekonstruktor byl rovněž navržen tak, aby dokázal odhadnout i velikost neměřitelné vstupní poruchy. Dále se ve schématu již nachází pouze blok Scope pro vykreslení reference a výstupu systému.

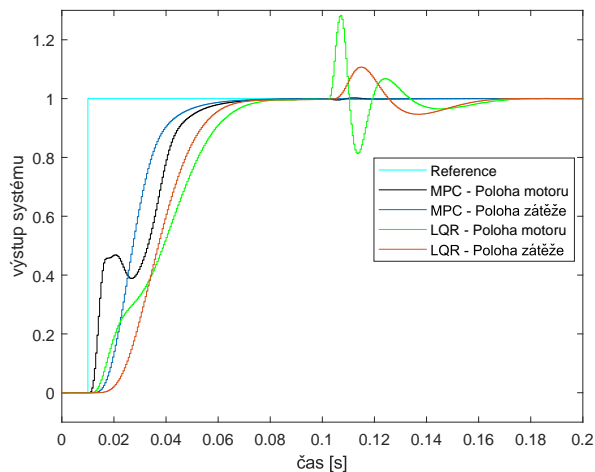
5.2.2 MPC vs. LQR regulace systému s poměrem rezonancí r_1

Nyní proběhne srovnání navrženého MPC regulátoru s LQR. Jako řízený systém bude uvažován ten s rezonančním poměrem r_1 . MPC kompenzátor bude uvažován ve tvaru, ve kterém byl navržen v kapitole 4. Stavový regulátor byl prostřednictvím příkazu z Matlabu *lqr* zvolen jako

$$\mathbf{K} = [1.2293 \quad -11.4387 \quad 10.7130 \quad 3.8589 \quad 1.5058 \quad -3.1859 \quad -0.2828 \quad 1]. \quad (85)$$

Kompenzační zesílení pro zaručení nulové odchylky v ustáleném stavu má hodnotu $K_{komp} = 2000$. Pro srovnání regulátorů budou použita zapojení v Simulinku, která jsou na obrázcích 7 a 60. Je důležité zmínit, že přestože LQR i MPC využívají kvadratické kritérium kvality, tak u MPC se v aktuální použité verzi pro zaručení nulové odchylky v ustáleném stavu penalizují přírůstky řízení oproti absolutním hodnotám u LQR. Váhy tedy mají jiný význam. Simulace pro otestování regulátorů bude probíhat tak, že se nejdříve skokově změní referenční hodnota a po ustálení výstupů do řízeného systému zasáhne porucha.

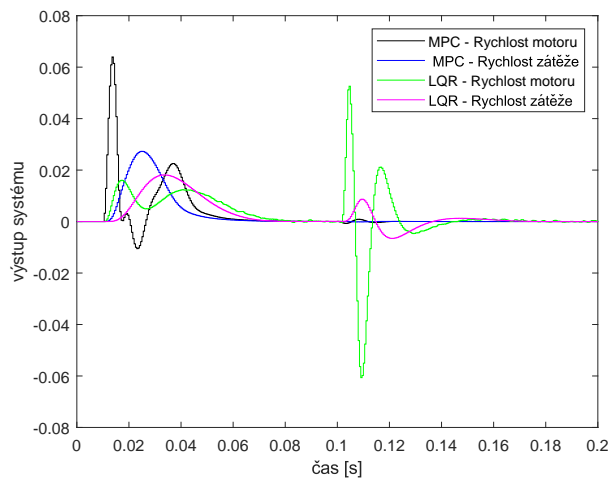
Po spuštění simulace byly při řízení MPC regulátorem a LQR získány následující průběhy poloh motoru a zátěže:



Obrázek 61: MPC vs. LQR - regulace systému s poměrem rezonancí r_1 - poloha

Co se týče přechodového děje, tak o něco lepších výsledků bylo dosaženo MPC regulátorem. Je to z toho důvodu, že odezva systému na skok při řízení tímto kompenzátozem byla o něco rychlejší. I doba regulace byla kratší u MPC. Nicméně u LQR je zase přechodový děj plynulejší. U polohy motoru se nevyskytuje ani jeden kmit. V čase 0.1 s pak do systému zasáhla porucha o velikosti -0.1. Oba regulátory dokázaly její vliv vyrušit. Nicméně LQR to dalo větší práci, což je vidět z průběhů polohy motoru, která se v daném místě více odchýlila od požadované referenční hodnoty.

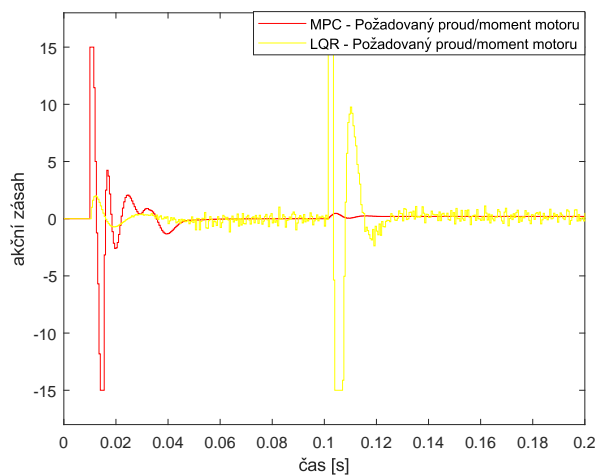
Průběhy rychlostí motoru a zátěže dopadly takto:



Obrázek 62: MPC vs. LQR - regulace systému s poměrem rezonancí r_1 - rychlost

Co se týče rychlostí motoru a zátěže při simulaci, tak během přechodového děje vyšších hodnot nabývaly rychlosti při řízení MPC. Nicméně není to o moc a při praktických aplikacích by to nejspíše nebylo ani poznat. Nicméně u LQR je přechodový děj z hlediska rychlostí motoru a zátěže plynulejší. U MPC prudší změny rychlostí mohou vést k většímu namáhání elektromotoru a hřídele a tím pádem i ke snížení jejich životnosti. Dále při odezvě na poruchu je opět vidět větší výchylka od požadované hodnoty při řízení LQR, zatímco při regulaci MPC se porucha projevila pouze nepatrně.

Nyní budou uvedeny průběhy akčních veličin při regulaci MPC kompenzátořem a LQR:



Obrázek 63: MPC vs. LQR - regulace systému s poměrem rezonancí r_1 - zásahy

Je patrné, že při přechodovém ději akční zásahy MPC nabývaly sice vyšších hodnot, ale byla zaručena kratší doba regulace. V obou průbězích lze pozorovat kmitání, nicméně pro případ řízení LQR byly kmity o něco vyšší. Při regulaci poruchy lze pozorovat, že akční zásah u MPC regulátoru byl téměř beze změny, zatímco LQR generoval poměrně velké akční zásahy, aby poruchu dokázal odregulovat.

Po shrnutí dosažených výsledků lze tvrdit, že oproti LQR je na tom řízení pomocí MPC

regulátoru o něco lépe. Konkrétně pro případ řízeného systému s rezonančním poměrem r_1 bylo u MPC dosaženo kratší doby regulace a i lepší odezvy na poruchu. Nicméně u LQR měly sledované veličiny plynulejší průběh.

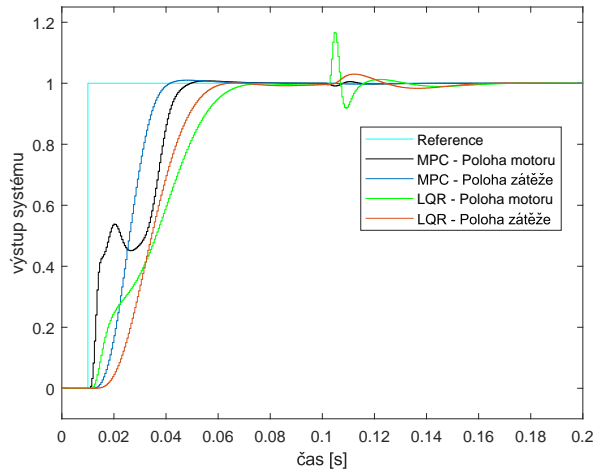
5.2.3 MPC vs. LQR regulace systému s poměrem rezonancí r_2

V této kapitole proběhne srovnání MPC kompenzátoru a LQR při řízení systému s rezonančním poměrem r_2 . MPC regulátor bude uvažován ve tvaru, ve kterém byl navržen pro kapitolu 4. LQR byl vytvořen v Matlabu pomocí příkazu *lqr* a má tuto hodnotu zpětnovazebního zesílení:

$$\mathbf{K} = [3.3276 \quad 7.2409 \quad 0.9679 \quad 7.7200 \quad 7.3869 \quad 6.1491 \quad -0.4827 \quad 1]. \quad (86)$$

Kompenzační zesílení bude nabývat hodnotu $K_{komp} = 2700$. I zde je nutné si uvědomit, že ačkoliv MPC i LQR používají kvadratické kritérium kvality, tak u obou strategií řízení je použito trochu jinak. U MPC se totiž váží přírůstky akčních zásahů zatímco u LQR absolutní hodnoty řízení. Schémata ze Simulinku, které budou sloužit k simulaci regulačních smyček, jsou na obrázcích 7 a 60. Průběh simulace bude následující. Nejprve dojde ke skokové změně referenční polohy pro motor. Po ustálení výstupů pak do systému zasáhne vstupní porucha.

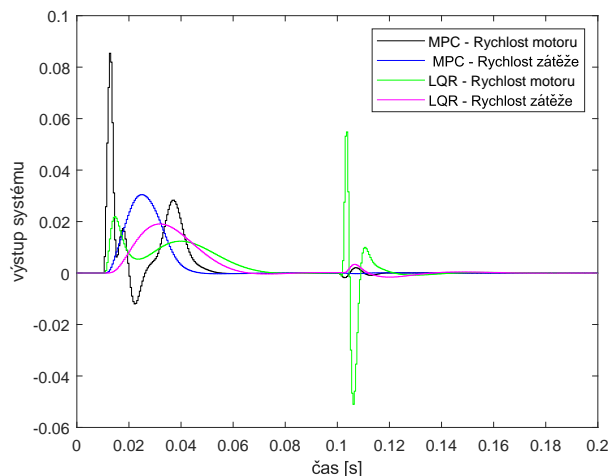
Po spuštění simulace a zobrazení výsledků se dostaly následující průběhy poloh motoru a zátěže při řízení MPC kompenzátořem a LQR:



Obrázek 64: MPC vs. LQR - regulace systému s poměrem rezonancí r_2 - poloha

Je vidět, že o trochu kvalitnější regulace při přechodovém ději bylo dosaženo u MPC. Doba regulace je totiž kratší. Nicméně u LQR bylo dosaženo plynulejšího přechodového děje. U polohy motoru se vyskytuje pouze jeden nepatrný kmit, díky čemuž je torze namáhána minimálně. V čase 0.1 s pak do systému zasáhla porucha o velikosti -0.2. Na průběhu poloh při řízení MPC se to takřka neprojeví. Při řízení pomocí LQR je ale situace jiná a porucha dokázala průběhy polohy motoru i zátěže na chvíli vychýlit ze svých ustálených hodnot.

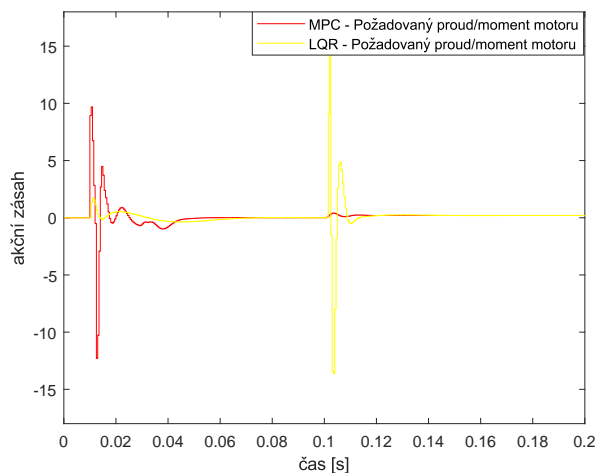
Nyní bude analyzován průběh rychlostí motoru a zátěže při simulaci, které dopadly následujícím způsobem:



Obrázek 65: MPC vs. LQR - regulace systému s poměrem rezonancí r_2 - rychlost

Je patrné, že při přechodovém ději nabývaly rychlosti motoru a zátěže vyšších hodnot při regulaci MPC. Díky tomu byla zaručena kratší doba regulace. Dále u průběhů rychlostí při regulaci LQR jsou vidět plynulejší děje, což by vedlo k menšímu namáhání součástí elektromotoru a hřídele. Co se týče regulace poruchy, tak u MPC se to na průbězích rychlosti vůbec téměř neprojevovalo. Naopak ale u LQR je vidět, že působící porucha způsobila v průběhu rychlostí motoru i zátěže vychýlení od jejich ustálených hodnot.

Nyní budou vykresleny akční zásahy generované MPC regulátorem a LQR během simulace:



Obrázek 66: MPC vs. LQR - regulace systému s poměrem rezonancí r_2 - zásahy

Je zřejmé, že vyšel výsledek, který již bylo možné očekávat na základě průběhů polohy a rychlosti. Akční zásahy u přechodového děje nabývají vyšších hodnot u MPC, nicméně je rychleji dosaženo ustáleného stavu. Naopak u řízení LQR je možné v průběhu akční veličiny plynulejší změny, které vedly na delší dobu regulace. V čase 0.1 s je vidět, jak do systému zasáhla porucha, což u LQR vedlo ke generování akčních zásahů s vyšší hodnotou, aby se

vliv poruchy vykompenzoval.

Celkově se dá říci, že kvalitnější regulace pro řízení systému s rezonančním poměrem r_2 bylo dosaženo u MPC regulátoru. Prediktivní kompenzátor měl kratší dobu regulace, ale na druhou stranu v průbězích sledovaných veličiny docházelo k agresivnějším změnám. Co se týče kompenzace poruchy, tak je na tom MPC lépe, jelikož její vliv vyrušil tento kompenzátor dříve.

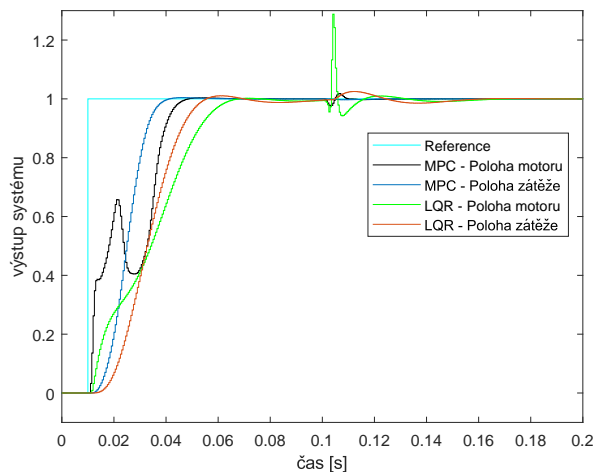
5.2.4 MPC vs. LQR regulace systému s poměrem rezonancí r_3

Jako poslední proběhne srovnání navrženého MPC regulátoru s LQR při regulaci systému s rezonančním poměrem r_3 . Tvar MPC regulátoru bude uvažován stejný jako ten v kapitole 4. Lineární kvadratický regulátor bude navržen prostřednictvím příkazu *lqr* z Matlabu. Zesílení zpětnovazební smyčky zde vyšlo jako

$$\mathbf{K} = [-12.0234 \quad -1.5572 \quad -0.1005 \quad 0.0273 \quad -0.0935 \quad 2.0994 \quad 2.7937 \quad 1]. \quad (87)$$

Kompenzační zesílení pro zaručení nulové odchylky v ustáleném stavu bylo nastaveno na hodnotu $K_{komp} = 2800$. Déle je dobré si uvědomit, že MPC i LQR používají k regulaci kvadratické kritérium kvality. Nicméně u MPC dochází k vážení přírůstků řízení a u LQR k vážení absolutních hodnot řízení. Váhy tedy mají jiný význam. Pro otestování regulace při řízení MPC kompenzátořem a LQR budou využita zapojení v Simulinku, která jsou na obrázcích 7 a 60. Simulace zde bude probíhat následujícím způsobem. Hned po jejím spuštění dojde ke skokové změně referenční hodnoty z 0 na 1. Po ustálení všech výstupů systému zasáhne vstupní porucha.

Po spuštění simulace se získaly průběhy poloh a zátěže, které zobrazuje následující graf:

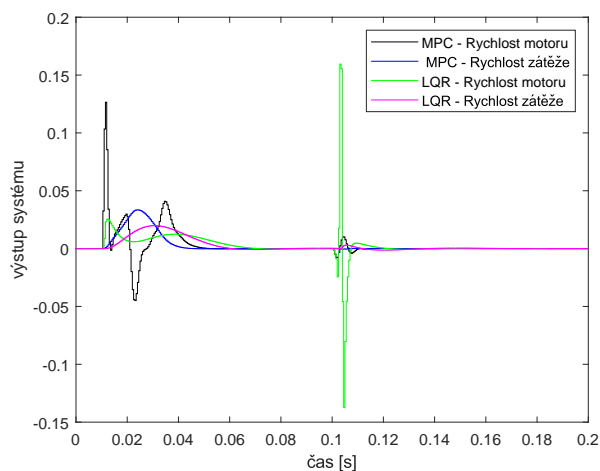


Obrázek 67: MPC vs. LQR - regulace systému s poměrem rezonancí r_3 - poloha

Je zřejmé, že o trochu lépe dopadla regulace prostřednictvím MPC kompenzátořem. Doba regulace je o chvíli kratší. Na druhou stranu v průběhu polohy motoru nastal jeden větší zákmit. Co se týče regulace LQR, tak je doba regulace o trochu delší, ale přechodový děj je u polohy motoru i zátěže mnohem plynulejší - nepatrný kmit se vyskytuje pouze u

polohy motoru. V čase 0.1 vteřiny do systému zasáhla vstupní porucha o velikosti -0.2. Vliv poruchy při řízení MPC kompenzátořem takřka není znát. Nicméně u LQR dojde po zásahu poruchy k většímu vychýlení polohy motoru i zátěže od referenční hodnoty.

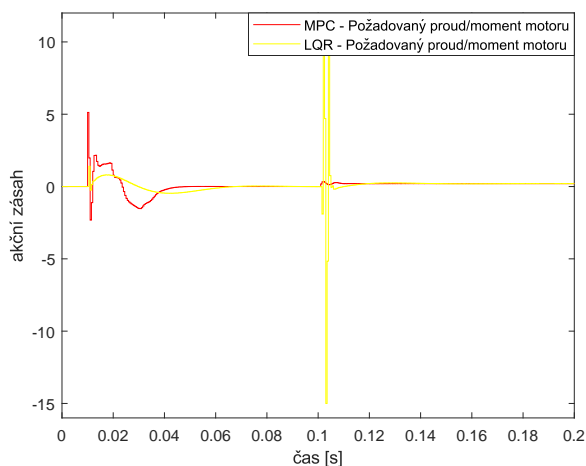
Nyní proběhne analýza průběhu rychlostí motoru a zátěže při simulaci:



Obrázek 68: MPC vs. LQR - regulace systému s poměrem rezonancí r_3 - rychlost

Jak je vidět, tak při přechodovém ději opět rychlosti při řízení MPC nabývaly větších hodnot, což vedlo na kratší dobu regulace. U řízení prostřednictvím LQR lze pozorovat plynulejší průběhy rychlostí při změně reference. U LQR lze opět vidět v čase 0.1 s díky poruše větší vychýlení od ustálené hodnoty.

Během simulace kompenzátory generovaly akční zásahy, které zobrazuje následující graf:



Obrázek 69: MPC vs. LQR - regulace systému s poměrem rezonancí r_3 - zásahy

Je zřejmé, že výsledek dopadl tak, jak již šlo z předchozích grafů očekávat. Generované akční zásahy MPC kompenzátořem byly během přechodového děje vyšší, což zaručilo kratší dobu regulace. Akční zásahy generované LQR byly naopak menší a navíc jejich změny byly plynulejší. V čase 0.1 vteřiny lze vidět, že do systémů zasáhla porucha, jejíž vliv se oběma

regulátorům podařilo vykompenzovat. Nicméně LQR musel pro její kompenzaci vyvinout akční zásahy s vyšší hodnotou.

Celkově se i zde dá říci, že při regulaci MPC kompenzátořem systému s rezonančním poměrem r_3 bylo dosaženo o něco lepších výsledků. Doba regulace byla u MPC kratší a i vliv poruchy byl odregulován rychleji. Nicméně u LQR byly průběhy sledovaných veličin plynulejší.

5.3 Srovnání MPC s kaskádní P-PI regulací - saturace aktuátoru

Jedna z hlavních výhod MPC je to, že dokáže již z principu brát v potaz omezení, která jsou chtěná klást na systém. To se využívá především u akčních zásahů, kde v praxi bývá omezení na aktuátor téměř vždy potřeba.

Tato podkapitola se zaměří na analýzu kvality regulace kaskádním P-PI regulátorem a MPC kompenzátořem v situacích, kdy je potřeba akční veličinu saturovat. Analýza opět proběhne prostřednictvím simulace, kdy dojde ke skokové změně referenčního signálu, díky čemuž se aktuátor dostane do saturace.

Za řízený systém v této podkapitole bude uvažován ten, který je modelován přenosem (60) v jeho diskrétní podobě se vzorkovací frekvencí $T_s = 0.5 \text{ ms}$. Snaha tedy bude řídit dokonale tuhý systém, který je tvořen filtrem, aktuátorem a mechanickou zátěží. K jeho regulaci se bude používat kaskádní P-PI regulátor bez ošetření unášení integrační složky, P-PI kaskádní regulátor s ošetřením unášení integrační složky a MPC kompenzátoř. K otestování regulace budou sloužit schémata zapojení, která jsou na obrázcích 7 a 50.

Tato podkapitola bude koncipována následujícím způsobem. Nejdříve dojde k návrhu obou regulátorů, které s k analýze budou používat. Následně budou navržené kompenzátořy otestovány prostřednictvím simulace, kdy dojde ke skokové změně reference, díky které se aktuátory dostanou do saturace. Akční veličina bude pro všechny simulace saturována od hodnoty -1000 do hodnoty 1000. Na závěr budou výsledné průběhy akčních zásahů pro regulátory srovnány a bude vybrán ten regulátor, který si s omezení poradil lépe.

5.3.1 Návrh regulátorů

Nyní budou navržené kompenzátořy, u nichž se bude analyzovat kvalita regulace v situaci, kdy dojde k saturaci aktuátoru. Nejdříve bude vytvořen kaskádní P-PI regulátor a posléze MPC kompenzátoř.

Kaskádní P-PI regulátor

Pro uvažovaný řízený systém s přenosem (60) byl pro tuto podkapitulu navržen PI regulátor rychlosti v následujícím tvaru

$$C_{c,d}(z) = \frac{162.1z - 151.5}{z - 1} \quad (88)$$

s periodou vzorkování 0.5 milisekundy. Polohový regulátor má hodnotu $P = 105$. Konstanta resetu má velikost $T_r = 0.22$. Akční veličina bude saturována do rozmezí od -1000 do 1000. Schéma zapojení v Simulinku, které bude použito pro tento regulátor je na obrázku 50. Toto zapojení se bude uvažovat ve 2 variantách. V první z nich nebude ošetřováno unášení

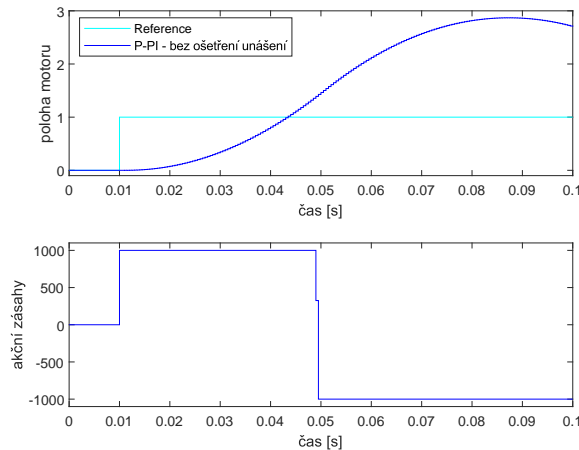
integrační složky při saturaci. V tom případě bude odpojená zpětná vazba od bloku saturace se zesílením, které udává konstantu resetu. V případě, kdy bude uvažováno ošetření unášení integrátoru, tak schéma bude takové, jako je na obrázku 50. Jediný rozdíl bude v tom, že se použije dokonale tuhý systém bez pružných módů, který je uvažován pro tuto kapitolu.

MPC regulátor

Pro MPC kompenzátor, který má řídit systém s přenosem (60), byly zvoleny následující hodnoty vstupních parametrů. Horizont predikcí má hodnotu $n_p = 600$. Horizont řízení nabývá hodnoty $n_c = 40$. Váhové matice \mathbf{Q} a \mathbf{R} budou pouze skaláry, jelikož řízená soustava je SISO systém. Jejich hodnoty byly zvoleny jako $q = 20$ a $r = 0.0003$. Saturace akčního zásahu bude i zde uvažována ± 1000 . Žádná další omezení na stavy ani výstupy brána v potaz nebudou. Je to z toho důvodu, aby bylo možné analyzovat pouze vliv saturovaného akčního zásahu.

5.3.2 Kaskádní regulace bez ošetření unášení integrátoru

V této kapitole proběhne simulace řízení uvažovaného systému pomocí kaskádního P-PI regulátoru. Pro tento případ nebude uvažováno ošetření integrátoru. Průběhy výstupu a akčního zásahu generovaného regulátorem vyšly takto:

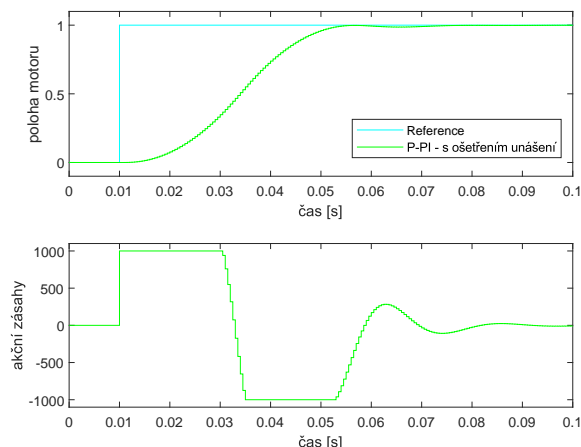


Obrázek 70: Řízení při použití kaskádního regulátoru bez ošetření unášení integrátoru

Je zřejmé, že si s regulací kaskádní P-PI kompenzátor neporadil vůbec dobře. Je totiž patrné, že vlivem saturace se stal uzavřený regulační obvod nestabilním. Z grafu znázorňující akční zásah je vidět, že se regulátor snaží systém dostat do nového stavu odpovídající referenci, ale vlivem horní meze akční veličiny to není dobře možné. Aby tento regulátor šlo použít, tak by musely být meze saturace zvětšeny. Nicméně z důvodu nestability regulační smyčky při takto nastavených omezeních již dále v této podkapitole nebude tento kaskádní P-PI regulátor bez ošetření unášení integrační složky uvažován.

5.3.3 Kaskádní regulace s ošetřením unášení integrátoru

Nyní bude odzkoušeno, jak si s regulací poradí kaskádní P-PI regulátor s ošetřením unášení integrátoru. Odezva na skok pro tento regulační obvod dopadla následovně:

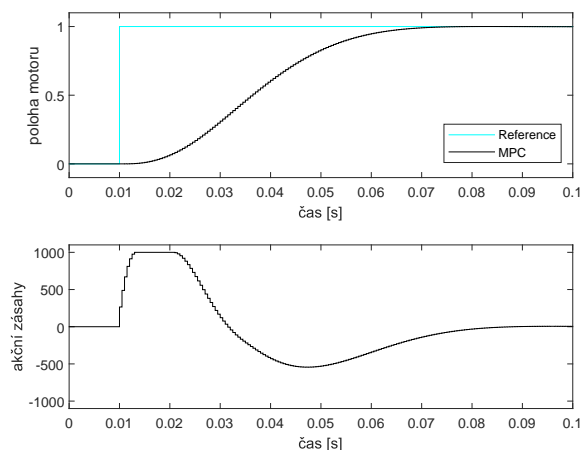


Obrázek 71: Řízení při použití kaskádního regulátoru s ošetřením unášení integrátoru

Je zřejmé, že si již regulátor poradil se změnou referenční hodnoty o mnoho lépe než v předchozím případě. Doba regulace činí přibližně 0.075 vteřiny. Z průběhu akční veličiny je pak vidět, že ihned po změně referenční hodnoty se aktuátor dostal do saturace, ve které setrval přibližně 0.02 sekundy. Posléze se ale ihned dostal na spodní mez saturace, ze které už akční zásah dokonvergoval na nulovou hodnotu.

5.3.4 Regulace pomocí MPC

Nyní zde bude otestován průběh regulace, kterou bude zaručovat MPC kompenzátor. Zde byly meze saturace akční veličiny zahrnuty do optimalizační úlohy, kde musely být brány v potaz. Průběh výstupu systému a akční veličiny generované regulátorem vyšly takto:



Obrázek 72: Řízení při použití MPC regulátoru

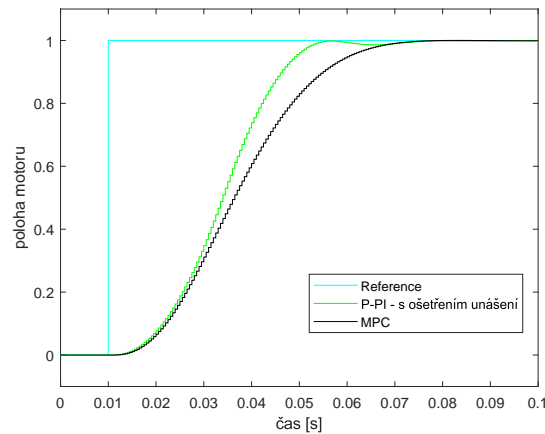
Pro tento případ řízení vyšla doba regulace 0.07 vteřiny. Co se týče generovaných akčních zásahů, tak je vidět, že po změně referenční hodnoty se aktuátor téměř ihned dostal na svou horní mez saturace. Na této horní hranici pak zásahy setrvaly přibližně 0.01 vteřiny a poté

plynule dokonvergovaly na nulovou hodnotu.

5.3.5 Srovnání výsledků

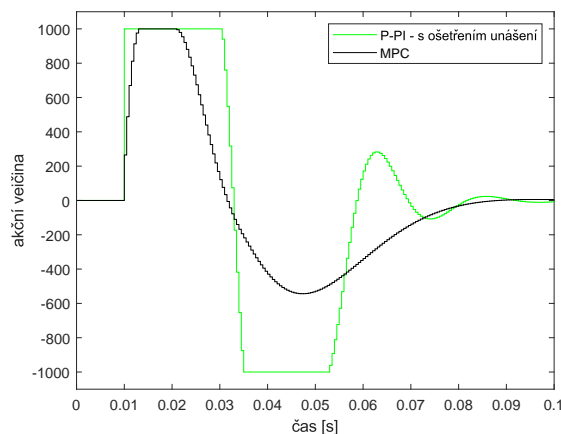
Nyní budou do jednoho grafu vykresleny průběhy výstupů systému a akčních zásahů při regulaci z předchozích kapitol. Nicméně kaskádní P-PI regulátor bez ošetření unášení integrační složky zde nebude uvažován. Je to z toho důvodu, že vlivem saturace se stal regulační obvod nestabilním.

Průběh výstupu při řízení kaskádním regulátorem s ošetřením unášení integrátoru a při řízení MPC vyšel následovně:



Obrázek 73: Srovnání výstupů při řízení kaskádním regulátorem a MPC

Jak je vidět, tak rychleji se na referenční hodnotě ustálí výstup při regulaci MPC kompenzátorem. Nicméně hlavním cílem této podkapitoly je sledovat průběhy akčních veličin. Akční zásahy generované uvažovanými regulátory během simulace měly následující průběh:



Obrázek 74: Srovnání akčních zásahů při řízení kaskádním regulátorem a MPC

Je patrné, že celkově z hlediska průběhu akčních zásahů je na tom lépe MPC kompenzátor. Na rozdíl od kaskádního P-PI regulátoru byly jím generované akční zásahy v saturaci pouze

jednou a to dokonce i na kratší dobu. Dále je vidět, že u MPC regulátoru je průběh akční veličiny mnohem plynulejší a hladší, což by v praktických aplikacích bylo určitě příznivější pro řízený systém. Shrnutím dosažených výsledků lze tedy tvrdit, že z hlediska kvality regulace při saturaci aktuátoru je na tom MPC kompenzátor ve srovnání s kaskádní P-PI regulací lépe.

6 Prediktivní regulace reálného systému

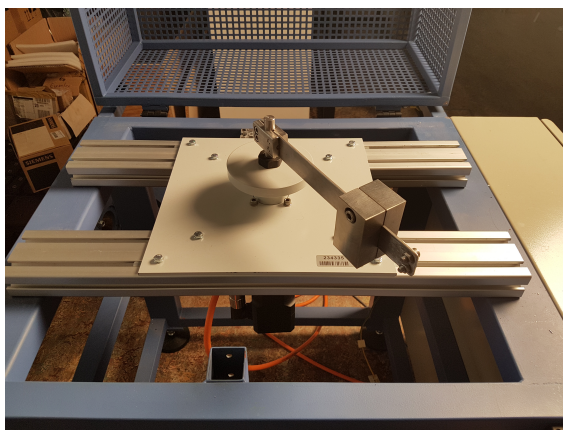
V této závěrečné kapitole proběhne návrh prediktivního řízení pro reálnou soustavu pomocí algoritmu pro zaručení nulové odchylky v ustáleném stavu, který byl vyvinut jako praktická část této bakalářské práce. Nejdříve proběhne prvotní seznámení s řízeným systémem. Bude zde uvedeno z jakých částí se skládá, jeho obrázek a model ve stavové reprezentaci. Následující sekce se bude věnovat již samotnému návrhu prediktivního regulátoru. V ní budou uvedeny všechny parametry MPC, které povedou na co nejlepší regulaci řízené soustavy. Následně proběhne ještě otestování průběhu regulace v Simulinku a diskuze nad dosaženou rychlostí optimalizace. Poslední podkapitola se bude zabývat situací, kdy je vzhledem k nedostatečně rychlé optimalizaci nutné navýšit periodu vzorkování u navrženého MPC.

Model reálné soustavy a její popis je převzat ze zdroje [7].

6.1 Popis řízené soustavy

Nyní bude popsána reálná soustava, pro kterou bude chtěné v této kapitole navrhnout prediktivní řízení. Jedná se o mechatronický stend pohonu s pružnou zátěží, který je detailně popsán ve článku [7].

Uvažovaný elektromechanický systém je tvořen elektrickým pohonem (synchronní motor poháněný servozesilovačem), pružnou spojkou, setrvačником a odjímatelným pohyblivým ramenem se zátěží. Toto zařízení má jeden stupeň volnosti, který dovoluje rameni se zátěží se pohybovat okolo vertikální osy. Systém zachycuje následující obrázek:



Obrázek 75: Reálný mechatronický systém pro který bude navrhováno MPC [7]

Pro lepší seznámení se se soustavou zde budou uvedeny vybrané parametry pohyblivé části systému [7]:

Délka ramene	0.235 m
Hustota materiálu ramene	$8030 \text{ kg} \cdot \text{m}^{-3}$
Youngův modul materiálu	$190,295,301,291.7 \text{ N} \cdot \text{m}^{-2}$
Plocha průřezu	$8.9274 \cdot 10^{-5} \text{ m}^2$
Hmotnost užitečného zatížení	1.049 kg
Moment setrvačnosti rozbočovače	$0.0024 \text{ kg} \cdot \text{m}^2$

Popsanou reálnou soustavu lze modelovat následující stavovou reprezentací v diskrétní podobě

$$\mathbf{x}_{k+1} = \begin{bmatrix} 0.04677 & 0.294 & -0.1969 & -0.9658 & -1.023 & 0.05284 & 0.4553 \\ -0.7578 & 0.09857 & 0.8737 & 0.0247 & -0.1726 & -0.2089 & -0.06235 \\ 0.8924 & -0.827 & 0.4281 & 0.3313 & 0.5134 & -0.05148 & -0.02915 \\ 0.2236 & 0.1035 & 0.2998 & 0.9308 & 0.06802 & -0.4952 & -0.07232 \\ -0.5498 & 0.02281 & -0.1284 & 0.3445 & 1.038 & 0.3042 & 0.1231 \\ -0.2214 & -0.1748 & -0.2631 & 0.01775 & -0.05788 & 0.8663 & 0.6025 \\ -0.1671 & 0.1176 & -0.236 & 0.04034 & -0.03472 & -0.4858 & 0.3915 \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} -0.2214 \\ -0.4982 \\ 1.53 \\ 1.764 \\ 1.462 \\ -0.9336 \\ -0.6983 \end{bmatrix} \mathbf{u}_k. \quad (89)$$

$$\mathbf{y}_k = \begin{bmatrix} -0.1606 & 0.5908 & -0.4234 & 0.1095 & 0.01406 & -0.3702 & -0.4973 \\ -0.4906 & -0.2119 & 1.036 & 0.0001324 & -0.1083 & 0.1311 & -0.02421 \\ 0.02343 & -0.002649 & 0.005162 & 0.03169 & 0.0213 & 0.1485 & -0.05331 \\ 0.0003499 & -0.03723 & -0.02039 & 0.0299 & 0.0233 & -0.004123 & 0.1116 \end{bmatrix} \mathbf{x}_k, \quad (90)$$

kde perioda vzorkování je $T_s = 0.5 \text{ ms}$ a dopravní zpožděním má hodnotu 3 periody vzorkování. Vstup modelu odpovídá požadovanému proudu/momentu motoru. Dále jsou modelovány 4 výstupy. Prvním je poloha motoru a druhým rychlost motoru. Zbylé dva výstupy popisují chování zátěže. Třetí výstup reprezentuje polohu zátěže a čtvrtý rychlost.

6.2 Návrh MPC regulátoru

Jako algoritmus prediktivního řízení bude použit ten pro zaručení nulové odchylky v ustáleném stavu. Nej kvalitnější regulace bylo dosaženo při následujících parametrech MPC. Horizont predikcí byl zvolen jako $n_p = 60$. Horizont řízení má hodnotu $n_c = 9$. Akční zásahy budou satureovány do rozmezí ± 0.1 . Vzhledem k počtu výstupů a vstupů bude váhová matice \mathbf{Q} dimenze 4 a váhová matice \mathbf{R} penalizující řízení dimenze 1. Tvary těchto váhových matic byly zvoleny následovně

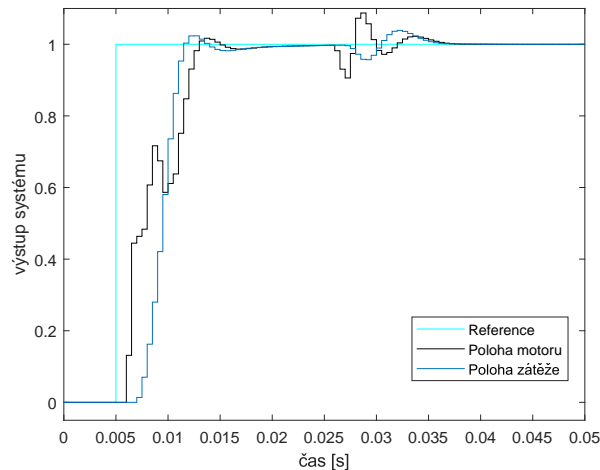
$$\mathbf{Q} = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 25 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}, \mathbf{R} = 0.001. \quad (91)$$

Je vidět, že důraz je kladen především na chování zátěže. Pro snížení výpočetní náročnosti bylo rozhodnuto, že soft omezení se budou v predikcích uvažovat pouze prvních 10 kroků do budoucnosti od současnosti. Konkrétně budou soft omezení nabývat následujících hodnot. K penalizaci polohy motoru nebo zátěže bude docházet, když jejich hodnoty překročí mez ± 1.05 . Rychlosti motoru nebo zátěže budou penalizovány v případě, když jejich hodnota překročí velikost ± 0.4 . Pro zaručení kvalitní regulace i zde bude kladeno soft omezení na torzi. K její penalizaci dojde při překročení hodnoty ± 0.3 . Váhová matice pro všechna uvedená měkká omezení bude identická s hodnotami 100 na diagonále.

6.3 Otestování průběhu regulace

Pro navržený MPC kompenzátor nyní proběhne otestování kvality regulace prostřednictvím simulace ze Simulinku. Schéma zapojení, které k simulaci bude použito, je na obrázku 7. Scénář simulace bude následující. Nejdříve dojde ke skokové změně reference z hodnoty 0 na 1. Po ustálení výstupů do řízené soustavy zasáhne vstupní porucha.

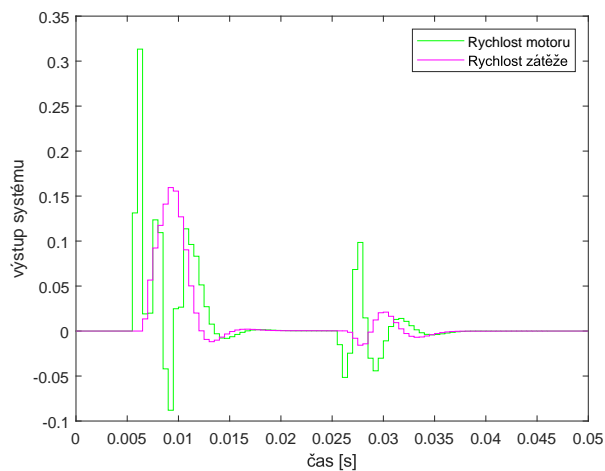
Průběhy poloh motoru a zátěže se při popsání simulaci vyvíjely takto:



Obrázek 76: Průběh poloh motoru a zátěže při prediktivní regulaci reálné soustavy

Hned na první pohled je vidět, že se jedná o velice rychlé řízení. Doba regulace je přibližně 0.02 vteřiny. Co se týče přechodového děje, tak u průběhu polohy motoru lze pozorovat jeden kmit. Nicméně průběh polohy zátěže je poměrně plynulý. Dále v čase 0.025 vteřiny do systému zasáhla vstupní porucha o velikost -0.01. Z grafu je zřejmé, že se díky tomu motor i zátěž vychýlily ze své ustálené polohy. Každopádně vliv poruchy byl vyrušen za 0.01 vteřiny.

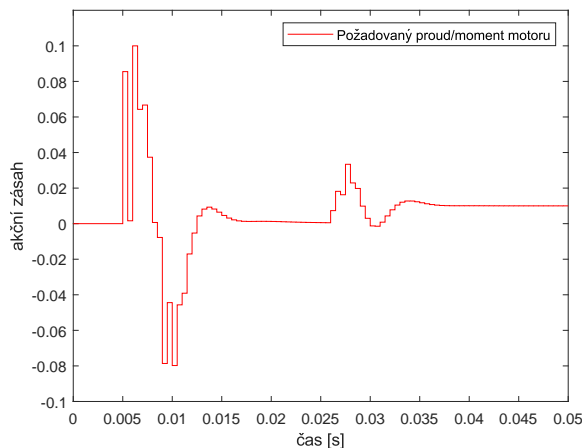
Rychlosti motoru a zátěže měly během simulace tyto průběhy:



Obrázek 77: Průběh rychlostí motoru a zátěže při prediktivní regulaci reálné soustavy

Z grafu je patrné, že při přechodovém ději se v rychlosti motoru vyskytly 3 kmity. Z hlediska namáhání hřídele by tedy bylo pro praktické použití regulátoru vhodné tyto kmity trochu utlumit. Nicméně rychlost zátěže se mění po celou dobu regulace poměrně plynule a to i při zásahu poruchy.

Navržený MPC regulátor generoval při řízení systému následující akční zásahy:

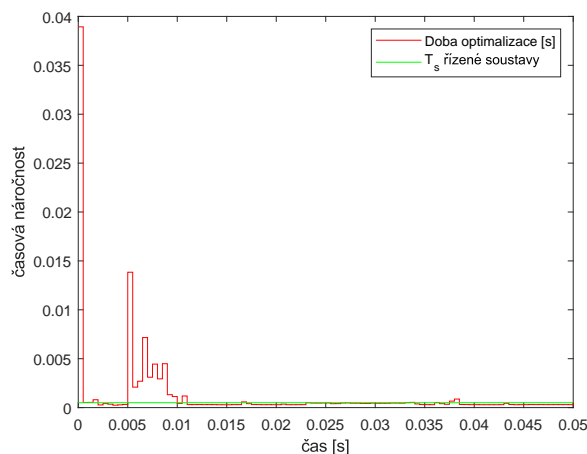


Obrázek 78: Průběh akčních zásahů generovaných při prediktivní regulaci reálné soustavy

Jak je z grafu vidět, tak při přechodovém ději se v průběhu akční veličiny vyskytují 2 kmity, přičemž v tom prvním akční zásah svou velikostí narazí do horní meze. Dále je v průběhu akční veličiny vidět v čase 0.025 s působení poruchy, jejíž vliv regulátor úspěšně svými zásahy vyloučil.

6.3.1 Časová náročnost optimalizace při regulaci

Při regulaci se využil solver *qpOASES* v jeho hot-start variantě. Časové náročnosti řešení optimalizační úlohy během simulace vycházely takto:



Obrázek 79: Časová náročnost optimalizace při řízení reálné soustavy

Maximální doba řešení jedné optimalizace vyšla 0.039 s. Průměrná doba řešení jedné úlohy

odpovídá času 0.0012 s. Je tedy patrné, že aby bylo možné navržené MPC využít pro regulaci uvažované řízené soustavy, tak by bylo nutné použít výpočetně výkonnější hardware. Ovšem existuje ještě další možnost, jak tento problém obejít. Řešením by bylo vhodně navrhnout MPC, aby jeho perioda vzorkování byla delší než u řízené soustavy. Touto úlohou se bude zabývat následující podkapitola.

6.4 Návrh MPC s delší periodou vzorkování

Jak bylo zjištěno výše, tak reálný systém by nebylo možné řídit MPC regulátorem, který by byl implementován na HW platformě, na které je vytvářena praktická část bakalářská práce. Je to z toho důvodu, že maximální i průměrná doba řešení jedné optimalizační úlohy při regulaci je delší než perioda vzorkování reálné soustavy. Řešení tohoto problému je vcelku přímočaré a to takové, že je nutné navrhnout MPC regulátor s delší periodou vzorkování, než má řízená soustava, aby se optimalizace stihla v každém kroku provést.

Tento problém je nicméně poměrně častý a důležitý krok implementace algoritmu prediktivního řízení pro praktickou aplikaci. Snaha je najít vhodný kompromis periody řízení a ostatních parametrů MPC, pro které je možné ještě dosáhnout dobré kvality regulace řízené soustavy.

Pro uvažovaný reálný systém je rozumná nejdelší perioda vzorkování řízení 1 ms. Důvod je ten, že se rezonance nachází na poměrně vysokých frekvencích. Model řízené soustavy tedy bude nutné přepočítat na uvedenou periodu vzorkování a pro tento systém s delším vzorkováním bude následně navrženo MPC. K navýšení vzorkovací frekvence se použije makro z Matlabu *d2d*. Kromě periody vzorkování je možné u MPC regulátoru ovlivnit ještě horizonty predikce a řízení. S delší periodou mohou být hodnoty horizontů kratší, aby se pokryl stejný časový úsek. Pro dosažení lepší kvality řízení je ještě možné vhodně upravit nastavení solveru optimalizační úlohy - například zvětšit tolerance pro zastavení optimalizace nebo případně omezit maximální počet iterací.

Tato podkapitola bude koncipována následovně. Nejdříve proběhne návrh MPC regulátoru pro reálnou soustavu, jejíž model bude upraven na delší periodu vzorkování. Poté bude u navrženého MPC otestována kvalita regulace v Simulinku. V závěru této podkapitoly ještě proběhne diskuze nad dosaženou časovou náročností optimalizace v každém kroku.

6.4.1 Návrh parametrů MPC pro systém s periodou vzorkování 1 ms

Původní reálný systém má vzorkovací frekvenci 0.5 ms. Aby bylo možné navrhnout adekvátní MPC regulátor se periodou vzorkování 1 ms, tak je nutné model řízené soustavy upravit, aby jeho perioda vzorkování byla rovněž 1 ms. To bude provedeno prostřednictvím Matlabu a jeho funkce *d2d*, která umožňuje změnit periodu vzorkování diskrétního systému. Regulátor se následně navrhne pro tento modifikovaný systém s delší vzorkovací frekvencí. Nicméně při samotné regulaci se bude řízený systém uvažovat s původní periodou vzorkování a to 0.5 ms.

Algoritmus řízení, který zde bude použit pro prediktivní regulaci, bude opět ten pro zaručení nulové odchylky v ustáleném stavu. Pro upravený model reálné soustavy s periodou vzorkování 1 ms bylo dosaženo nejvyšší kvality regulace při následujících parametrech MPC. Predikční horizont byl zvolen jako $n_p = 25$. Horizont řízení nabývá hodnoty $n_c = 5$. Jak je vidět, tak velikost horizontu predikcí byla zvolena téměř poloviční oproti předchozímu případu, kdy měl MPC periodu vzorkování totožnou s řízeným systémem. Je to z toho

důvodu, aby se zjednodušila optimalizační úloha a tím pádem i celková časová náročnost jejího řešení. Bylo zvoleno, že akční zásahy budou saturovány do rozmezí ± 0.015 . Soustava má 4 výstupy a jeden vstup. Z toho důvodu váhová matice \mathbf{Q} bude dimenze 4 a matice vážící vstupy bude dimenze 1. Jejich hodnoty byly zvoleny takto

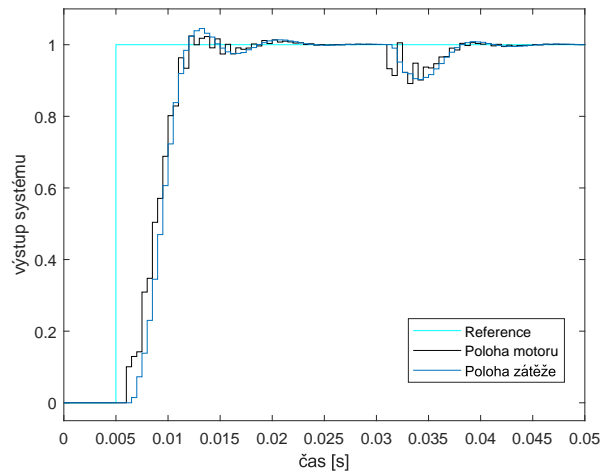
$$\mathbf{Q} = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 40 \end{bmatrix}, \mathbf{R} = 50000. \quad (92)$$

Jak je vidět, tak i zde byly váhy matice \mathbf{Q} zvoleny tak, aby bylo penalizováno především chování na straně zátěže. Z důvodu potřeby snížit výpočetní náročnost optimalizační úlohy, což povede na rychlejší optimalizaci, bylo rozhodnuto, že se opustí od všech soft omezení. Doba výpočtu minima účelové funkce by se tak měla snížit, ovšem cenou za to nejspíše bude o trochu horší kvalita regulace.

6.4.2 Otestování průběhu regulace

Ačkoli byl regulátor navržen pro řízený systém s periodou vzorkování 1 *ms* a tím pádem i samotný regulátor bude mít tuto vzorkovací frekvenci, tak při praktickém použití by reálná soustava byla stále vzorkována s původní periodou a to 0.5 *ms*. Z toho důvodu je nutné schéma zapojení v Simulinku, které je na obrázku 7 a které bude zde použito k otestování, doplnit dvěma bloky ZOH (Zero-order hold) a to na vstup a výstup regulátoru. Prostřednictvím této modifikace bude do regulátoru přicházet signál s požadovaným vzorkováním.

Nyní již k samotné simulaci. Ta bude probíhat tak, že nejdříve dojde ke skokové změně referenční hodnoty a po ustálení výstupů do systému zasáhne porucha. Po spuštění simulace s popsaným scénářem měly polohy motoru a zátěže následující průběh:

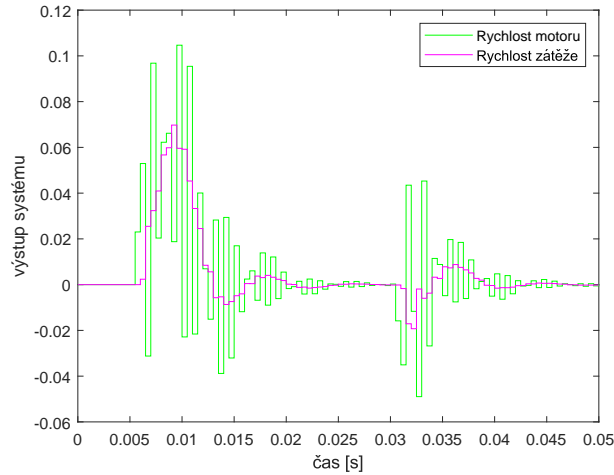


Obrázek 80: Průběh poloh při regulaci kompenzátorem s periodou vzorkování 1 *ms*

Na první pohled je zřejmé, že doba regulace je o něco delší než pro případ s původní periodou vzorkování. Po změně referenční hodnoty trvalo kompenzátoru 0.02 vteřiny než dostal systém do nové požadované polohy. U přechodového děje je rovněž vidět, že kvůli absenci soft

omezení se v průběhu polohy motoru a zátěže vyskytuje drobné kmitání. V čase 0.04 vteřiny do systému zasáhla vstupní porucha o velikosti -0.01. Je patrné, že se soustava díky tomu dostala ze své ustálené polohy. Nicméně za 0.01 sekundy byl vliv poruchy vyrušen.

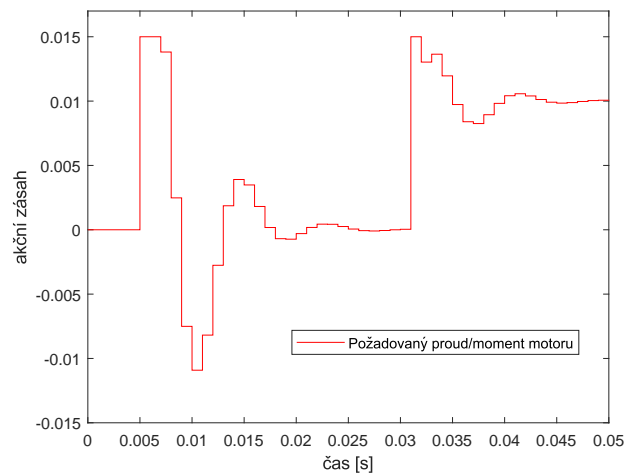
Rychlosti motoru a zátěže při simulaci měly tento průběh:



Obrázek 81: Průběh rychlostí při regulaci kompenzátořem s periodou vzorkování 1 ms

Z grafu je vidět, že průběh rychlosti motoru je při přechodovém ději poměrně kmitavý, což by vedlo na velké namáhání hřídele. Toto chování by se dalo nejspíše vylepšit přidáním soft omezení na rozdíl mezi rychlostí motoru a zátěže, ale vzhledem k navýšení náročnosti optimalizace by to nebylo vhodné. Co se týče rychlosti zátěže, tak ta byla během celé simulace poměrně plynulá.

Během simulace MPC regulátor s prodlouženou periodou vzorkování generoval tyto akční zásahy:



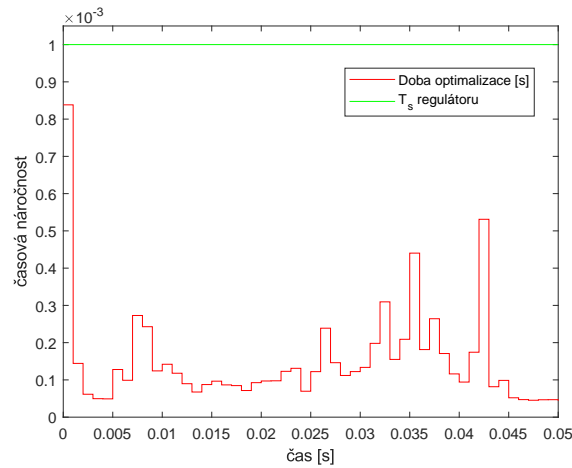
Obrázek 82: Průběh zásahů kompenzátořem s periodou vzorkování 1 ms při regulaci

Po lepší analýze grafu je vidět, že MPC kompenzátoř skutečně generoval akční zásahy s prodlouženou periodou vzorkování 1 ms. Co se týče samotného průběhu, tak zásahy regulátoru

do systému byly poměrně plynulé. U přechodového děje je u prvního kmitu vidět, že se akční veličina dostala na svou horní mez. Dále od času 0.04 s je patrné, že do systému působí porucha. I tady se akční veličina dostala na svou horní mez saturace. Každopádně je zřejmé, že vliv poruchy se podařilo akčními zásahy vyrušit.

6.4.3 Diskuze nad časovou náročností optimalizace

V algoritmu prediktivního řízení byl jako solver optimalizačních úloh použit *qpOASES* v jeho hot-start variantě. Při uvažované simulaci řízení byla časová náročnost řešení úlohy v každém kroku následující:



Obrázek 83: Časová náročnost optimalizace při řízení s delší periodou vzorkování

Zde je již patrné, že se podařilo dosáhnout kýženého výsledku. Doba řešení optimalizace v jednom kroku po celou simulaci netrvala regulátoru déle, než je jeho samotná perioda vzorkování. Maximálně kompenzátoru trvalo vyřešení jedné úlohy 0.00084 s a to v prvním kroku regulace. Je to z toho důvodu, že solver neměl ještě informaci o výsledku optimalizace z předešlého kroku. Dále je vidět, že během simulace byly již doby řešení úlohy mnohem menší. Průměrně použitému solveru trvalo řešení jedné optimalizační úlohy 0.00015 vteřiny. Maximální i průměrná doba optimalizace je tedy menší než perioda vzorkování regulátoru 1 ms. Výsledek lze tedy interpretovat tak, že pro praktické použití je možné navržený regulátor pro řízení reálné soustavy použít.

7 Závěr

Tématem a hlavním cílem této bakalářské práce bylo analyzovat princip činnosti a následně i implementovat algoritmus prediktivní regulace, který by se dal využít k řízení pohybu elektromechanických soustav.

V teoretické části práce proběhlo nejdříve podrobné seznámení s touto strategií řízení zvanou *Model-based predictive control* (MPC). Bylo zde ukázáno, jak se při regulaci predikuje budoucnost na základě modelu řízené soustavy. Dále co to jsou horizonty predikcí a řízení a v neposlední řadě, co obnáší technika *move blocking*. Nedílnou a podstatnou součástí prediktivní regulace je řešení úlohy minimalizace účelové funkce, což v práci bylo rovněž rozebráno. Následně byla uvedena problematika kladení omezení na stavy, výstupy a akční zásahy při prediktivní regulaci. Možnost poměrně jednoduše aplikovat omezení na řízenou soustavu je jedna z hlavních výhod prediktivní regulace. Dále byla rozebrána úloha regulování poruch. Zde byly vyjmenovány hlavní zdroje chyb při prediktivní regulaci a jak je možné se s nimi vypořádat. Současně zde byly uvedeny i důvody, proč je MPC v současnosti používáno.

Další sekce teoretické části bakalářské práce byla věnována algoritmům prediktivního řízení. Zde byly prezentovány 3 možné implementace MPC. Každý z těchto algoritmů je trochu jiný a každý se hodí pro jiný účel. Nicméně v praxi se nejvíce používá ten pro zaručení nulové odchylky v ustáleném stavu. V závěru teoretické části práce byl ještě uveden způsob, jak je možné převést úlohu prediktivní regulace bez omezení na klasickou stavovou zpětnou vazbu.

Poté již následovala praktická část bakalářské práce. Zde bylo prvotním cílem navrhnout vhodnou prediktivní regulaci pro 3 systémy s pružnou zátěží. Využilo se při tom algoritmu, který byl vyvinut jako praktická část práce. Nejdříve byly všechny pružné systémy detailně popsány a posléze se pro každý z nich vytvořil vhodný MPC kompenzátor. U všech vytvořených regulačních obvodů byla následně otestována kvalita regulace prostřednictvím simulace ze Simulinku. Pro každý regulátor poté proběhla diskuze nad dosaženou robustností ve stabilitě a nad dosaženou časovou náročností řešení optimalizace v každém kroku. Jako poslední bylo u navržených MPC kompenzátorů odzkoušeno, jak jsou robustní vůči změnám v zesílení a v poloze rezonance. Ukázalo se, že regulátory byly poměrně robustní.

Dále se praktická část práce zabírala srovnáním prediktivní regulace s kaskádním P-PI řízením a s LQR. U kaskádní regulace bylo celkově dosaženo pomalejší dynamiky. Co se týče LQR, tak zde byla rychlost regulace srovnatelná s MPC, nicméně byl u ní znát kmitavý charakter všech sledovaných veličiny. Rovněž i u kompenzace poruch na tom byl lépe MPC regulátor. Dále byla srovnána kaskádní P-PI regulace s MPC z hlediska omezení na akční zásahy. Zde bylo patrné, že při saturaci akční veličiny bylo dosaženo kvalitnější regulace při řízení MPC kompenzátořem.

V poslední sekci praktické části bakalářské práce proběhl návrh MPC regulátoru pro reálnou soustavu. Ačkoliv zde kvalita řízení byla poměrně vysoká, tak se ukázalo, že solver optimalizačních úloh nestačí řešit úlohu minimalizace v době kratší než je perioda vzorkování řízené soustavy. Byl proto navržen ještě jeden MPC kompenzátor, který pracoval s delší periodou vzorkování, která nabývala své maximální možné hodnoty vzhledem k řízené soustavě. Ke snížení výpočetní náročnosti úlohy zde byly zmenšeny hodnoty horizontu predikcí a řízení a rovněž bylo upuštěno od použití soft omezení. Díky uvedeným modifikacím se již podařilo snížit výpočetní náročnost optimalizace na tolik, že maximální i průměrná doba řešení jedné úlohy byla kratší než navýšená perioda vzorkování regulátoru. Závěr tedy byl, že pro praktickou aplikaci by bylo možné navržené MPC pro řízení reálné soustavy použít.

Seznam literatury a informačních zdrojů

- [1] ROSSITER, J. A. Model-based predictive control: a practical approach. Boca Raton: CRC Press, c2003. ISBN 08-493-1291-4
- [2] RUCHIKA a NEHA RAGHU. Model Predictive Control: History of Development. International Journal of Engineering Trends and Technology [online]. Thapar University, Patiala 2013 [cit. 2021-04-22] Dostupné z: <http://ijettjournal.org/volume-4/issue-6/IJETT-V4I6P173.pdf>
- [3] MIKULÁŠ, Ondřej. Quadratic Programming Algorithms for Fast Model-Based Predictive Control [online]. Prague, 2009 [cit. 2021-04-21]. Dostupné z: https://wiki.control.fel.cvut.cz/mediawiki/images/9/94/Bp_2013_mikulas_ondrej.pdf. Bachelor thesis. Czech Technical University in Prague, Faculty of Electrical Engineering
- [4] ŠANTIN, Ondřej. Influence of Model Uncertainty on Constraints Handling in Predictive Control [online]. Prague, 2009 [cit. 2021-04-21]. Dostupné z: <https://docplayer.cz/160441881-Czech-technical-university-in-prague-faculty-of-electrical-engineering-diploma-thesis.html>. Diploma thesis. Czech Technical University in Prague, Faculty of Electrical Engineering
- [5] FERREAU, Hans Joachim. QpOASES Users's Manual [online]. Lovain, Belgie, 2012 [cit. 2021-5-1]. Dostupné z: <https://usermanual.wiki/Pdf/manual.2007140309.pdf>. User's manual. KU Leuven, Optimization in Engineering Center (OPTEC) and Department of Electrical Engineering
- [6] GOUBEJ, Martin. Robustní řízení pohybu pružných elektromechanických soustav [online]. Plzeň, 2014 [cit. 2021-4-28]. Dostupné z: <https://otik.uk.zcu.cz/handle/11025/20647>. Disertační práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd
- [7] GOUBEJ, Martin; KÖNIGSMARKOVÁ, Jana; KAMPINGA, Ronald; NIEUWENKAMP, Jakko; PAQUAY, Stéphane. Employing Finite Element Analysis and Robust Control Concepts in Mechatronic System Design-Flexible Manipulator Case Study [online]. Appl. Sci. 2021, 11, 3689 [cit. 2021-05-11]. Dostupné z: <https://doi.org/10.3390/app11083689>. Case study.

Seznam obrázků

1	Ilustrace horizontu predikcí n_p a horizontu řízení n_c	16
2	Zapojení stavového regulátoru pro získání ekvivalentního řízení k MPC	31
3	Dvouhmotový systému - motor se zátěží s ekvivalentem soustavy [6]	33
4	Odezva na skok systému s přenosem $P_{c1}(s)$	36
5	Odezva na skok systému s přenosem $P_{c2}(s)$	37
6	Odezva na skok systému s přenosem $P_{c3}(s)$	38
7	Schéma pro otestování prediktivní regulace	39
8	Průběh poloh motoru a zátěže při regulaci systému s poměrem rezonancí r_1	41
9	Průběh rychlosti motoru a zátěže při regulaci systému s poměrem rezonancí r_1	41
10	Průběh akčního zásahu při regulaci systému s poměrem rezonancí r_1	42
11	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_1 - polohy	43
12	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_1 - rychlosti	43
13	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_1 - zásahy regulátorů	44
14	Solver <i>quadprog</i> - doba optimalizace při řízení pevného a pružného systému (r_1)	45
15	Srovnání solverů - řízení systému s rezonančním poměrem r_1	46
16	Regulace systému s r_1 při změně zesílení - poloha	47
17	Regulace systému s r_1 při změně zesílení - rychlost	47
18	Regulace systému s r_1 při změně zesílení - zásahy	48
19	Regulace systému s r_1 při změně polohy rezonance - poloha	48
20	Regulace systému s r_1 při změně polohy rezonance- rychlost	49
21	Regulace systému s r_1 při změně polohy rezonance - zásahy	49
22	Průběh poloh motoru a zátěže při regulaci systému s poměrem rezonancí r_2	51
23	Průběh rychlosti motoru a zátěže při regulaci systému s poměrem rezonancí r_2	51
24	Průběh akčního zásahu při regulaci systému s poměrem rezonancí r_2	52
25	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_2 - polohy	53
26	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_2 - rychlosti	53
27	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_2 - zásahy regulátorů	54
28	Solver <i>quadprog</i> - doba optimalizace při řízení pevného a pružného systému (r_2)	55
29	Srovnání solverů - řízení systému s rezonančním poměrem r_2	56
30	Regulace systému s r_2 při změně zesílení - poloha	57
31	Regulace systému s r_2 při změně zesílení - rychlost	58
32	Regulace systému s r_2 při změně zesílení - zásahy	58
33	Regulace systému s r_2 při změně polohy rezonance - poloha	59
34	Regulace systému s r_2 při změně polohy rezonance- rychlost	59

35	Regulace systému s r_2 při změně polohy rezonance - zásahy	60
36	Průběh poloh motoru a zátěže při regulaci systému s poměrem rezonancí r_3 .	61
37	Průběh rychlosti motoru a zátěže při regulaci systému s poměrem rezonancí r_3	62
38	Průběh akčního zásahu při regulaci systému s poměrem rezonancí r_3	62
39	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_3 - polohy	63
40	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_3 - rychlosti	64
41	MPC vs. ekvivalentní stavový regulátor - řízení systému s r_3 - zásahy regulátorů	64
42	Solver <i>quadprog</i> - doba optimalizace při řízení pevného a pružného systému (r_3)	65
43	Srovnání solverů - řízení systému s rezonančním poměrem r_3	67
44	Regulace systému s r_3 při změně zesílení - poloha	68
45	Regulace systému s r_3 při změně zesílení - rychlost	68
46	Regulace systému s r_3 při změně zesílení - zásahy	69
47	Regulace systému s r_3 při změně polohy rezonance - poloha	69
48	Regulace systému s r_3 při změně polohy rezonance- rychlost	70
49	Regulace systému s r_3 při změně polohy rezonance - zásahy	70
50	Schéma pro otestování kaskádní P-PI regulace	71
51	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_1 - poloha .	72
52	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_1 - rychlost	73
53	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_1 - zásahy .	73
54	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_2 - poloha .	74
55	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_2 - rychlost	75
56	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_2 - zásahy .	75
57	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_3 - poloha .	76
58	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_3 - rychlost	77
59	MPC vs. kaskádní P-PI - regulace systému s poměrem rezonancí r_3 - zásahy .	77
60	Schéma pro otestování regulace pomocí LQR	78
61	MPC vs. LQR - regulace systému s poměrem rezonancí r_1 - poloha	79
62	MPC vs. LQR - regulace systému s poměrem rezonancí r_1 - rychlost	80
63	MPC vs. LQR - regulace systému s poměrem rezonancí r_1 - zásahy	80
64	MPC vs. LQR - regulace systému s poměrem rezonancí r_2 - poloha	81
65	MPC vs. LQR - regulace systému s poměrem rezonancí r_2 - rychlost	82
66	MPC vs. LQR - regulace systému s poměrem rezonancí r_2 - zásahy	82
67	MPC vs. LQR - regulace systému s poměrem rezonancí r_3 - poloha	83
68	MPC vs. LQR - regulace systému s poměrem rezonancí r_3 - rychlost	84
69	MPC vs. LQR - regulace systému s poměrem rezonancí r_3 - zásahy	84
70	Řízení při použití kaskádního regulátoru bez ošetření unášení integrátoru . . .	86

71	Řízení při použití kaskádního regulátoru s ošetřením unášení integrátoru	87
72	Řízení při použití MPC regulátoru	87
73	Srovnání výstupů při řízení kaskádním regulátorem a MPC	88
74	Srovnání akčních zásahů při řízení kaskádním regulátorem a MPC	88
75	Reálný mechatronický systém pro který bude navrhováno MPC [7]	90
76	Průběh poloh motoru a zátěže při prediktivní regulaci reálné soustavy	92
77	Průběh rychlostí motoru a zátěže při prediktivní regulaci reálné soustavy	92
78	Průběh akčních zásahů generovaných při prediktivní regulaci reálné soustavy	93
79	Časová náročnost optimalizace při řízení reálné soustavy	93
80	Průběh poloh při regulaci kompenzátozem s periodou vzorkování 1 <i>ms</i>	95
81	Průběh rychlostí při regulaci kompenzátozem s periodou vzorkování 1 <i>ms</i>	96
82	Průběh zásahů kompenzátozem s periodou vzorkování 1 <i>ms</i> při regulaci	96
83	Časová náročnost optimalizace při řízení s delší periodou vzorkování	97