



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

EXPLICITNÍ PREDIKTIVNÍ ŘÍZENÍ A JEHO APLIKACE

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 3902T005 – Automatické řízení a inženýrská informatika
Autor práce: **Bc. Martin Kopal**
Vedoucí práce: Ing. Lukáš Hubka, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin Kopal**
Osobní číslo: **M12000245**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Automatické řízení a inženýrská informatika**
Název tématu: **Explicitní prediktivní řízení a jeho aplikace**
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Seznamte se s explicitním prediktivním řízením (MPC), popište jeho výhody i omezení.
2. Nalezněte vhodný software pro návrh explicitního MPC.
3. Ukažte na vhodném simulačním příkladu/příkladech funkčnost explicitního MPC.
4. Pro zvolený laboratorní model vytvořte vhodný matematický popis, identifikujte neznámé parametry (přenosovou funkci) a zrealizujte explicitní prediktivní řízení.

Rozsah grafických prací: dle potřeby dokumentace

Rozsah pracovní zprávy: 40–50 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

- [1] BEMPORAD, A. Hybrid Toolbox for MATLAB. [online]. 2012 [cit. 2013-02-15]. Dostupné z: <http://cse.lab.imtlucca.it/bemporad/hybrid/toolbox/>.
- [2] ŘEHOŘ, Jiří. Explicitní řešení úlohy prediktivní regulace. Černošice, 2008. Diplomová práce. ČVUT.
- [3] MACHALSKÝ, Ondřej. Prediktivní řízení s měřenou poruchovou veličinou. Zlín, 2011. Diplomová práce. UTB ve Zlíně.
- [4] MAGNI, Lalo. Nonlinear model predictive control: towards new challenging applications. Berlin: Springer, 2009, s. 345-369. ISBN 978-3-642-01094-1.
- [5] HOČ, Michal. Helicopter in virtual space. Praha, 2008. Diplomová práce. ČVUT.


Vedoucí diplomové práce:

Ing. Lukáš Hubka, Ph.D.


Ústav mechatroniky a technické informatiky

Datum zadání diplomové práce: 10. října 2013

Termín odevzdání diplomové práce: 16. května 2014


prof. Ing. Václav Kopecký, CSc.
děkan




doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2013

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Především bych velice rád poděkoval svému vedoucímu diplomové práce Ing. Lukáši Hubkovi, Ph.D. za odbornou pomoc s řešením různých problémů a cenné rady, které mi poskytoval během četných konzultací. Dále děkuji své rodině za podporu během studia.

Abstrakt

V diplomové práci je čtenář seznámen s principem prediktivního řízení založeného na modelu a konkrétně se tato práce zaměřuje na jeho explicitní formu. Pro několik simulačních příkladů jsou prakticky vyzkoušeny a ukázány možnosti explicitního prediktivního řízení. Realizace explicitního MPC je provedena na základě návrhu a využití explicitního prediktivního regulátoru. Pro návrh takového regulátoru je využitý dodatečný toolbox do programu Matlab, který se nazývá Hybrid Toolbox. Nakonec se zabývá praktickou aplikací explicitního prediktivního regulátoru pro regulaci reálné soustavy. Reálná soustava je uvažována ve formě jednorozměrového systému, kterým je model motoru a ve formě vícerozměrového systému, kterým je model vrtulníku.

Klíčová slova: explicitní MPC, regulace, kritické regiony, vícerozměrový systém

Abstract

In the thesis the reader is introduced with the principle of the Model Predictive Control and specifically this thesis is focused on the explicit form of the Model Predictive Control. The possibilities of the explicit MPC are practically tested and they are shown for the several simulation examples. The realization of the explicit MPC is based on the design and use of the explicit predictive controller. For the design of this type a controller is used an additional toolbox for the program Matlab, which is called the Hybrid Toolbox. The last part is focused to the practical application of the explicit predictive controller for regulation of the real system. The real system is considered in the form of one-dimensional system, where this system is the model of the motor and in the form of multidimensional system, where this system is the model of the helicopter.

Key words: explicit MPC, regulation, critical regions, multidimensional system

Obsah

Prohlášení	4
Poděkování.....	5
Abstrakt	6
Seznam zkratek	9
Seznam ilustrací	10
Úvod	12
1 Prediktivní řízení založené na modelu	14
1.1 Princip a struktura MPC	14
1.2 Kvadratické kritérium a omezující podmínky	16
2 Explicitní prediktivní řízení	18
2.1 Základní myšlenka	18
2.2 Redukce počtu regionů	20
2.3 Ofset při sledování reference	21
2.4 Shrnutí výhod a omezení explicitního MPC	22
3 Implementace explicitního MPC	24
3.1 Software pro návrh	24
3.2 Simulační příklad – stíhací letoun.....	26
3.3 Proces identifikace reálné soustavy	30
3.4 Schéma měření a regulace	31
3.5 Reálná soustava dvou DC motorů – popis soustavy	32
3.5.1 Statické vlastnosti.....	33
3.5.2 Dynamické vlastnosti, identifikace přenosu	34
3.5.3 Praktická realizace explicitního MPC	36
3.5.4 Vliv parametrů	40
3.5.5 Porovnání s PID, vliv poruchy	42

3.6 Model vrtulníku – popis soustavy	45
3.6.1 Statické a dynamické vlastnosti	47
3.6.2 Praktická realizace explicitního MPC	53
3.7 Implementace do reálného zařízení.....	58
Závěr.....	59
Seznam použité literatury	61
Příloha 1 – Schéma modelu vrtulníku	63
Příloha 2 – Skript pro návrh explicitního MPC	64
Příloha 3 – Obsah přiloženého CD.....	65

Seznam zkratk

MPC	prediktivní řízení založené na modelu (Model Predictive Control)
PLC	programovatelný logický automat (Programmable Logic Controller)
$MIMO$	soustava s více vstupy/výstupy (Multiple Input Multiple Output)
$SISO$	soustava s jedním vstupem/výstupem (Single Input Single Output)
QP	kvadratické programování
$mp - QP$	multiparametrické kvadratické programování
$w(t)$	žádaná hodnota
$y(t)$	regulovaná veličina
$u(t)$	akční veličina
$e(t)$	regulační odchylka
$r(t)$	kritické regiony
N_u	horizont řízení
N	horizont predikce
N_1	minimální horizont predikce
N_2	maximální horizont predikce
$w(k + p k)$	budoucí průběh žádané hodnoty
$y(k + p k)$	predikovaná hodnota regulované veličiny
$\Delta u(k + p k)$	posloupnost budoucích přírůstků akční veličiny
S, Q	váhové funkce
$F(s)$	operátor transformace modelu
$F_0(s)$	operátor transformace reálného systému

Seznam ilustrací

Obr. 1.1 – Princip MPC [1]	15
Obr. 1.2 – Struktura MPC.....	16
Obr. 2.1 – Pozorovatel stavu.....	20
Obr. 2.2 – Schéma simulace offsetu	22
Obr. 2.3 – Simulace problému s offsetem.....	22
Obr. 3.1 – Hybrid Toolbox knihovna	26
Obr. 3.2 - Význam úhlů u letadla [14]	26
Obr. 3.3 – Kritické regiony	28
Obr. 3.4 – Schéma pro simulaci.....	28
Obr. 3.5 – Odezva systému	29
Obr. 3.6 – Akční zásahy	29
Obr. 3.7 – Výběr regionů během regulace.....	29
Obr. 3.8 – Porovnání modelu a reálného systému [15]	30
Obr. 3.9 – Schéma měření	31
Obr. 3.10 – Schéma regulace	32
Obr. 3.11 – Reálný model motoru.....	33
Obr. 3.12 - Průběh měření statické charakteristiky motoru.....	33
Obr. 3.13 – Statická charakteristika motoru	34
Obr. 3.14 – Průběh měření dynamických vlastností motoru	35
Obr. 3.15 – Porovnání naměřených průběhů a identifikovaného přenosu	36
Obr. 3.16 – Regulace s offsetem	37
Obr. 3.17 – Odezvy systému motoru pro $T_s = 0,05$ s.....	38
Obr. 3.18 – Porovnání odezev pro $T_s = 0,01$ s a $T_s = 0,05$ s.....	39
Obr. 3.19 – Akční zásahy u regulace motoru pro $T_s = 0,05$ s	39
Obr. 3.20 – Výběr regionů při regulaci motoru pro $T_s = 0,05$ s a $T_s = 0,01$ s	40
Obr. 3.21 – Simulace pro změny horizontu predikce (N)	41
Obr. 3.22 – Porovnání simulace a měření pro změnu horizontu predikce	41
Obr. 3.23 – Simulace pro změny váhy přírůstků akční veličiny (T).....	41
Obr. 3.24 – Porovnání simulace a měření pro změnu parametru T	42
Obr. 3.25 – Porovnání PID vs explicitní MPC	44
Obr. 3.26 – Průběh regulovaných veličin při reakci na poruchu	44
Obr. 3.27 – Robustnost explicitního MPC	45

Obr. 3.28 – Srovnání limitů PID vs explicitní MPC	45
Obr. 3.29 – Obecná struktura reálné vícerozměrové soustavy	46
Obr. 3.30 – Reálný model vrtulníku.....	47
Obr. 3.31 – Měření statické charakteristiky hlavní vrtule	47
Obr. 3.32 – Identifikace statických vlastností hlavní vrtule	48
Obr. 3.33 – Měření dynamiky hlavní vrtule	49
Obr. 3.34 – Porovnání měření a identifikovaného přenosu hlavní vrtule	49
Obr. 3.35 – Měření astatických vlastností zadní vrtule	50
Obr. 3.36 - Porovnání měření a identifikovaného přenosu zadní vrtule	50
Obr. 3.37 – Měření křížové vazby zadní motor-elevace	51
Obr. 3.38 – Porovnání identifikace a měření křížové vazby zadní motor-elevace	52
Obr. 3.39 – Měření křížové vazby hlavní motor-azimut	52
Obr. 3.40 – Porovnání identifikace a měření křížové vazby hlavní motor-azimut	53
Obr. 3.41 – Porovnání regulovaných veličin při řízení elevace.....	54
Obr. 3.42 – Porovnání akčních zásahů a regionů při řízení elevace	55
Obr. 3.43 – Porovnání průběhů při řízení azimutu.....	55
Obr. 3.44 – Vliv poruchy na elevaci	56
Obr. 3.45 – Vliv poruchy na azimut.....	57
Obr. 3.46 – Simulace MPC vs explicitní MPC	57

Úvod

V minulosti, kdy nebyly známy pokročilé metody řízení (např. adaptivní regulace, prediktivní řízení založené na modelu nebo algoritmy využívající fuzzy logiku a jiné), se pro regulaci reálných soustav používaly víceméně standardní regulátory typu PID. Tento typ regulátoru je v principu základní možností, jak řídit různé reálné systémy, a proto je do značné míry používán dodnes, a to i z hlediska jeho jednoduchosti. Jeho struktura je implementována v mnoha průmyslových PLC a umožňuje tak rychlou aplikaci řídicího algoritmu. Nicméně PID regulátory mají i svoje limity a omezení, které jim nedovolí optimálně regulovat některé procesy. Ve většině případů jsou tyto regulátory dobře použitelné při řízení jednorozměrových soustav (SISO). Naopak je tomu u vícerozměrových soustav (MIMO), kde není vždy zaručena kvalitní a optimální regulace pomocí PID regulátorů a v některých případech jsou tyto regulátory nepoužitelné. Dnes už jsou známy postupy a rozšiřující struktury, které použitelnost PID regulátorů u MIMO systémů zvyšují. Přesto u komplikovaných systémů, kde řízení ovlivňuje mnoho faktorů, je implementace PID regulátorů velmi obtížná. Samotná identifikace komplikovaného a rozsáhlého systému je poměrně náročná a realizace takové struktury řízení, která by dokázala zkoordinovat různé vstupy tak, aby výsledné chování systému odpovídalo požadavkům na řízení je pomocí standardní struktury ve formě PID regulátorů velice obtížné. Obecně z takových důvodů, kde tyto standardní metody řízení selhávají ve smyslu požadavků na kvalitní a optimální regulaci, začaly vznikat modernější a pokročilejší metody řízení. Jedna z takových metod řízení je explicitní prediktivní řízení, které v principu vychází ze standardního prediktivního řízení založeného na modelu.

Diplomová práce se věnuje tomuto pokročilému a perspektivnímu způsobu řízení reálných soustav. Jsou zde popsány principy a základní myšlenky klasického prediktivního řízení a především se tato práce věnuje zmíněné explicitní formě MPC, které disponuje jistými výhodami z hlediska lepší použitelnosti v praxi. Z některých nevýhod klasického MPC se stávají naopak výhody u explicitního MPC a většina výhod a omezení jsou v této práci popsány dále.

V další části se přiblíží možnosti návrhu explicitního prediktivního regulátoru tak, aby bylo možné v rámci simulací a následně na reálném systému ověřit možnosti explicitního MPC. Všechny návrhy, simulace a praktická měření jsou prováděny

v prostředí programu Matlab. Jsou dostupné různé nástroje pro návrh takovýchto pokročilých regulátorů, kde existují minimálně dva rozšiřující toolboxy do prostředí programu Matlab. Tyto toolboxy umožňují návrh, simulaci a následně praktické ověření způsobu explicitního prediktivního řízení. V této práci se přiblíží možnosti dodatečného toolboxu, který se nazývá Hybrid Toolbox. Tento toolbox je plně dostačující pro požadavky realizace explicitního MPC.

Poslední část se zabývá praktickou realizací explicitního MPC ve formě simulací na několika simulačních příkladech, kde jsou ověřeny možnosti tohoto způsobu řízení ještě před reálnou implementací na některý z vybraných reálných systémů. Následně jsou navržené regulátory použity pro reálné řízení soustav, kterými jsou jednak model motoru a v druhém případě vícerozměrová soustava ve formě modelu vrtulníku.

1 Prediktivní řízení založené na modelu

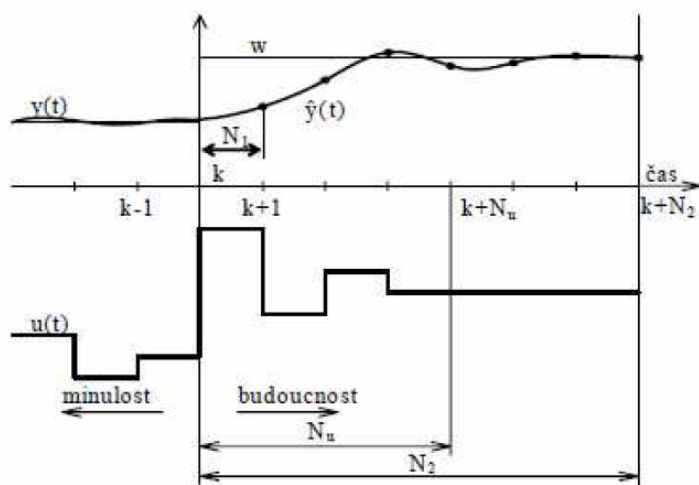
MPC jako takové je pokročilá regulační technika umožňující optimální řízení relativně velkého množství reálných procesů a průmyslových aplikací od jednoduchých SISO systémů až po komplikované MIMO systémy zahrnující i několik desítek vstupů a výstupů. Tento způsob řízení vznikl v 60. letech minulého století jako reakce na požadavky petrochemického průmyslu, resp. jako požadavek na optimální regulaci takových systémů, kterými byly např. ropné plošiny a rafinérie. Dříve byla použitelnost MPC omezena na relativně pomalé procesy. Tato omezující vlastnost je dána tím, že pro optimální regulaci je nutné zajistit dostatečný výpočetní výkon a zároveň vhodný model, který dostatečně přesně popisuje daný systém. Z těchto důvodů není MPC tak rozšířené, jako je tomu např. u klasických přístupů regulace ve formě PID regulátorů, ale situace se postupně zlepšuje a MPC se čím dál více začíná využívat. Vzrůstající popularita MPC v průmyslové praxi je především dána neustále zvyšujícím se výpočetním výkonem průmyslových zařízení, resp. obecně všech zařízení, které umožňují implementovat tuto metodu řízení. V dnešní době se MPC využívá mj. v chemickém průmyslu, energetice, papírenském průmyslu nebo i v systémech automobilů.

1.1 Princip a struktura MPC

Velmi výrazná vlastnost, kterou se tento způsob řízení odlišuje od klasických způsobů regulace, je, že MPC umožňuje uvažovat různá **omezení** systému už při návrhu regulátoru a výsledná regulace je tak mnohem efektivnější, ekonomičtější a více optimální než jiné způsoby řízení, které omezení nemohou uvažovat. V názvu této metody se skrývá nejdůležitější základní prvek, bez kterého by nebylo možné úspěšně implementovat tuto regulační techniku. MPC je na tomto prvku založeno, vychází z něj, a tím prvkem je model soustavy. Je to zároveň výhoda i nevýhoda, kde výhoda spočívá v tom, že lze v principu použít libovolný model, kterým může být např. přechodová charakteristika, která se získá měřením, přenosová funkce nebo stavový popis, který popisuje příslušný systém. Nevýhoda je naopak dána požadavkem na přesnost modelu, resp. je potřeba získat takový popis procesu, který dostatečně přesně vystihuje chování systému. Pokud by takový model nebyl zaručen, výsledná regulace by v lepším případě byla nekvalitní. Obvykle je snaha nalézt model ve zjednodušené podobě, se kterou umíme pracovat, kde tyto modely se označují jako lineární časově invariantní. Jakmile

by se do modelu zanesla nelinearita nebo samotný model by byl nelineární, poté by se složitost řešení MPC velmi výrazně zvýšila.

Základní princip MPC by mohl být znázorněn následujícím obrázkem (obr. 1.1). MPC, jak z názvu vyplývá, predikuje, neboli předpovídá vývoj systému do budoucnosti na tzv. *horizontu predikce* (N , resp. N_2 viz obr. 1.1). Tento postup je umožněn díky znalosti modelu systému. MPC predikuje budoucí hodnoty výstupu systému $\hat{y}(t)$ na základě znalosti minulých hodnot výstupu a vstupu ($y(t)$, $u(t)$ do času k) a na základě budoucích akčních zásahů.



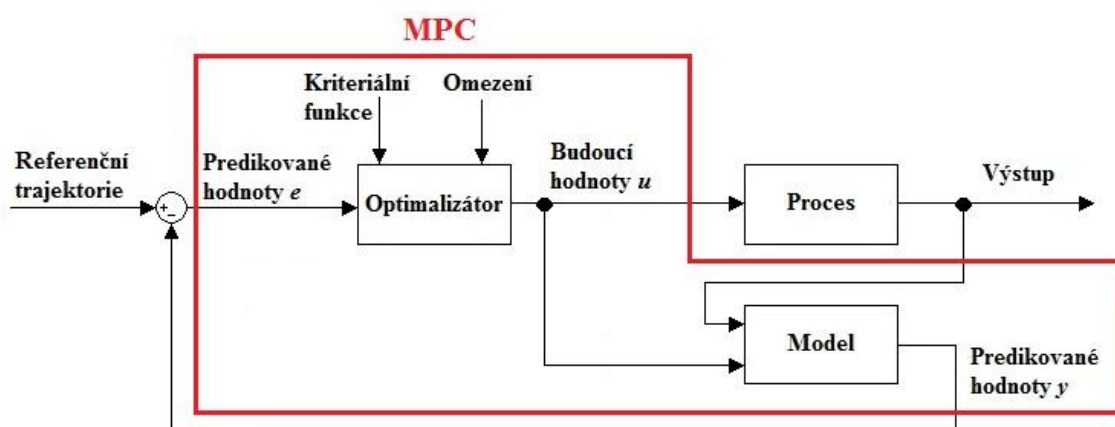
Obr. 1.1 – Princip MPC [1]

Predikce budoucích akčních zásahů je založena na tom, že v každém okamžiku vzorkování se hledá řešení optimalizační úlohy, která je reprezentována určitým kritériem. Obvykle se používá kritérium kvadratické, kde řešení je založeno na minimalizaci rozdílů mezi predikovanou regulovanou veličinou a budoucím průběhem žádané hodnoty. Cílem je nalézt takový průběh akčních zásahů, které minimalizují příslušné kvadratické kritérium (2). V některých případech se hledání řešení uvažované optimalizační úlohy označuje jako kvadratické programování (QP). Výsledkem pak je nalezení optimální posloupnosti akčních zásahů v rámci horizontu predikce (1).

$$U_N^* = [u_0^*, \dots, u_{N-1}^*] \quad (1)$$

Pro řízení se použije pouze první člen z dané posloupnosti a v dalším kroku, se celý výpočet opakuje. Proces opakování výpočtu je důležitý z toho důvodu, že pokud by se použila pro řízení vypočítaná optimální posloupnost, jednalo by se o řízení

v otevřené smyčce. V pozdější fázi řízení by nebylo zaručeno sledování žádané hodnoty a predikovaná regulovaná veličina by se od skutečné vzdalovala. Proto se provádí opakovaný výpočet optimální posloupnosti. V principu se tak jedná o zpětnou vazbu, která je představována tzv. *klouzavým horizontem predikce*. Horizont predikce se postupně posouvá a v každém kroku se provádějí nová měření, čímž se aktualizují data potřebná pro výpočet nové posloupnosti. Předpis (1) lze označit jako zákon řízení MPC. Struktura prediktivního řízení, popsaná v předchozích odstavcích, by mohla být znázorněna i následujícím schématem (obr. 1.2).



Obr. 1.2 – Struktura MPC

Často používaná charakteristika rozdílu MPC vs PID [2], [3] je vztažena k příkladu řízení automobilu, kde řidič dopředu vidí dráhu, po které se bude pohybovat a ovládacími prvky automobilu (plyn, brzda, volant) tuto dráhu sleduje, případně reaguje na poruchové signály ve smyslu např. překážky na vozovce. Tomu odpovídá řízení pomocí MPC, kde naopak využitím klasického přístupu v podobě PID by řidič reagoval pouze na základě zpětných zrcátek. MPC tedy reaguje na budoucí hodnotu vstupů, resp. regulačních odchylek, a PID reaguje na minulé a současné hodnoty vstupů.

1.2 Kvadratické kritérium a omezující podmínky

Obecně může být kvadratické kritérium v následující podobě

$$J(k) = \sum_{p=N_1}^N (w(k+p|k) - y(k+p|k))^T \cdot S \cdot (w(k+p|k) - y(k+p|k)) + \sum_{p=0}^{N_u-1} \Delta u(k+p|k)^T \cdot Q \cdot \Delta u(k+p|k) \quad (2)$$

Uvažované kritérium (2) zajišťuje sledování referenční trajektorie, protože zde vystupují kvadráty predikované regulační odchylky a budoucích přírůstků akční veličiny. Tyto členy jsou dále váženy příslušnými váhovými funkcemi (S, Q), které ovlivňují výslednou dynamiku a kvalitu regulačního pochodu.

Člen $y(k + p|k)$ lze chápat tak, že se jedná o predikci budoucího vývoje y v čase $k + p$ za předpokladu, že se vychází ze známého času k . Prvek N_1 představuje minimální horizont predikce, který je obvykle roven jedné, hodnota vyšší by měla smysl pouze u systémů s dopravním zpožděním nebo u systémů s dlouhou dobou náběhu.

Kromě samotného kritéria se v optimalizační úloze mohou dále vyskytovat různá omezení, která musí nalezené řešení respektovat. Tato omezení jsou obvykle ve formě nelineárních rovnic. Nejčastěji se uvažují omezení akčních zásahů nebo jejich přírůstků, případně omezení regulovaných veličin. Omezení jsou zaváděna buď uměle, aby se dosáhlo splnění požadavků na regulační pochody, anebo častěji jsou omezení přítomna v rámci použité technologie v podobě saturace, kterou nelze nijak odstranit, ale je potřeba ji uvažovat při realizaci MPC. Obecně omezení mohou být i časově proměnná, ale vhodnější je uvažovat omezující podmínky konstantní, protože se zavedením omezení do optimalizační úlohy se zvyšuje složitost řešení a v některých případech může vzniknout i problém, že optimalizační úloha nebude mít řešení.

Omezující podmínky se v principu rozdělují na dva typy – *tvrdé* a *měkké*. Tvrdé podmínky jsou takové, které se za žádných okolností nesmějí překročit, kde zpravidla sem patří omezení akční veličiny. Měkké podmínky jsou naopak takové, u kterých se za určitých okolností dovoluje překročit maximální nebo minimální hodnota po určitou dobu. Důsledky takového překročení nejsou závažné, např. nežádoucí vysoká teplota.

2 Explicitní prediktivní řízení

MPC, popsané v předchozí kapitole, vyžaduje, aby se v každém vzorkovacím okamžiku prováděl výpočet optimalizačního algoritmu, který je založen na aktuálním stavu systému tak, aby se dosáhlo optimálního řízení. Tento výpočet se uskutečňuje on-line, tzn. při reálném řízení soustavy, a proto hlavním požadavkem je dostatečný výpočetní výkon, aby se v relativně krátkém čase vyřešila optimalizační úloha a aplikoval se výsledek řešení (akční zásah) na reálnou soustavu. Čím vyšší je složitost řešení, tím větší je zapotřebí výpočetní výkon. Z toho důvodu je klasické MPC vnímáno spíše jako metoda pro řízení pomalejších procesů [4]. Výpočetní rychlost není jediným omezením klasického MPC, dalším může být riziko ukrývající se v samotném výpočtu, kde pokud by se výpočet provedl špatně nebo by nastala chyba ve výpočtu, mohlo by to mít fatální následky, především v aplikacích, které jsou založeny na bezpečném řízení procesu [4].

2.1 Základní myšlenka

Hlavním rozdílem při aplikaci explicitního MPC, je, na rozdíl od klasického MPC, vyřešení optimálního řízení off-line pro všechny stavy $x(t)$ systému. V principu se tak akční zásahy vypočítávají předem mimo reálnou soustavu a vše, co je potřebné provádět on-line, tak je rozhodování, který akční zásah se zrovna použije. Cílem explicitního MPC je tak přesunout důležité a především výpočetně náročné úkoly do off-line roviny. Je snaha vytvořit explicitní závislost $u(t)$ na $x(t)$ takovou, že bude řešit optimalizační problém. Chce se získat explicitní popis zákona řízení (3), který minimalizuje kvadratické kritérium.

$$u^* = \kappa(x) \quad (3)$$

Zákon řízení u explicitního MPC už není představován opakovanými výpočty posloupnosti akčních zásahů pro aktuální stav systému. Naopak jedná se o funkční závislost mezi akčními zásahy, které jsou představovány vygenerovanými regiony pro všechny možné stavy, a aktuálním stavem systému. Pokud se získá daná funkce $\kappa(x)$ (4), pak řešení prediktivního řízení, spočívající ve výpočtu optimálních akčních zásahů u^* , se zjednoduší do tvaru vyhodnocení funkce $\kappa(x)$. V rámci řízení se rozhoduje, který z vygenerovaných regionů bude zrovna aktivní. Tato skutečnost představuje velmi výraznou úsporu výpočetního výkonu, protože se v každém kroku

nemusí opakovat výpočetně náročně řešení optimalizační úlohy, ale vyhodnocuje se, které z dříve nalezených řešení se aktuálně použije.

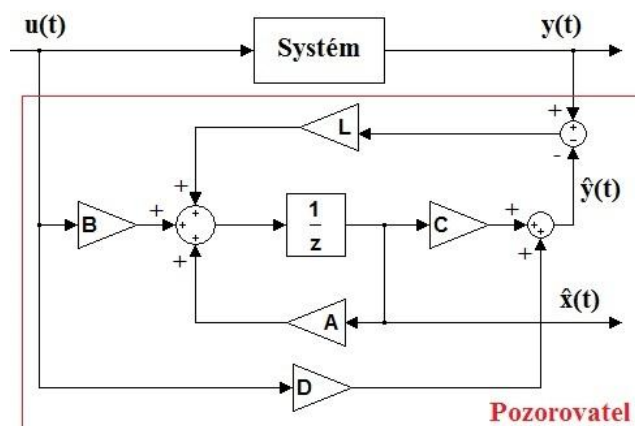
Vzniká ale otázka, jak nalézt příslušnou funkci $\kappa(x)$. V principu je přístup podobný, jako u klasického MPC. Hledá se řešení optimalizační úlohy ve formě minimalizace kvadratického kritéria. U klasického MPC se tento přístup označuje jako QP, ale u explicitní formy MPC, se označuje jako multiparametrické kvadratické programování (mp-QP), protože se hledá řešení pro všechny možné stavy $x(t)$ a nikoliv pouze pro aktuální stav systému. Výsledkem optimalizační úlohy je funkce $\kappa(x)$ představující zákon řízení (4), která je **spojitá** a po **částech afinní** (lineární).

$$\kappa(x) = \begin{cases} F_1 x + G_1 & \text{pro } x \in R_1 \\ \vdots & \vdots \\ F_R x + G_R & \text{pro } x \in R_R \end{cases} \quad (4)$$

kde $R_i \subset R^{n_x}, i = 1, \dots, R$ jsou tzv. polytopické oblasti, nebo obvykle označované jako kritické regiony, které rozdělují prostor parametrů $x(t)$, tedy stavový prostor na takové části, kde v každé z nich je definovaná spojitá afinní funkce (4) a příslušný akční zásah. Postup vytváření regionů je takový, že nejdříve se vypočítá řešení optimalizační úlohy pro vhodně zvolenou počáteční podmínku a následně se sestojí příslušný kritický region. Algoritmus poté prohledává okolí tohoto regionu a postupně vytváří další nové kritické regiony [5], [6]. Výsledkem výpočtů je v principu vytvoření seznamu či tabulky afinních funkcí, ve které se při reálném řízení vyhledává odpovídající kritický region a příslušný akční zásah, který se aplikuje. Vyhledávání konkrétních regionů se provádí on-line a především je nutné správně identifikovat stav systému. Obvykle všechny stavy systému nejsou měřitelné, a proto se musejí tyto stavy odhadovat. Prediktivní regulátor je tak doplněn o tzv. pozorovatele stavu (obr. 2.1), který pomáhá při identifikaci aktuálního stavu systému a umožňuje správnou funkci vyhodnocení ve smyslu, který kritický region a akční zásah bude v danou chvíli aktivní.

Řešení optimalizační úlohy explicitního MPC je v principu ve tvaru tabulky afinních funkcí, ve které se podle aktuálního stavu systému nalezne správný řádek s odpovídajícím kritickým regionem. Nejjednodušší způsob, jak implementovat nalezený explicitní zákon řízení a realizovat tak reálně explicitní MPC je založen na tom, že se nejdříve uloží do dostatečně velké paměti tabulka regionů. Poté se díky

vyhledávacímu algoritmu při on-line řízení postupně prohledává tabulka, dokud se nenalezne správný kritický region a aplikuje se příslušný akční zásah.



Obr. 2.1 – Pozorovatel stavu

Přestože je tento postup intuitivní a implementace v podobě programovaného kódu je vcelku jednoduchá, tak jsou dostupné i efektivnější metody pro vyhodnocování, které se používají zejména tehdy, pokud je počet kritických regionů velmi rozsáhlý. Jedním z omezení explicitního MPC je bezesporu složitost tabulky ve smyslu počtu vytvořených kritických regionů. Obecně platí, že čím vyšší je počet generovaných regionů, tím větší je potřeba paměť pro uložení tabulky a zároveň se opět zvyšují nároky na výpočetní výkon pro zajištění požadavku dostatečně rychle rozhodovat o výběru správného regionu. Tyto důvody ovlivňují jednak cenu implementace explicitního MPC, a také jsou důvodem pro hledání jiných metod vyhodnocování správných regionů během on-line řízení případně způsobů pro snižování složitosti tabulky.

2.2 Redukce počtu regionů

Počet kritických regionů je velmi výrazně závislý na počtu uvažovaných omezení, kde tato závislost je exponenciální. Nejjednodušší způsob vyhledávání má svoje limity a není vhodný pro rozsáhlé tabulky regionů. Jsou známé pokročilejší metody, které snižují jak dobu potřebnou pro nalezení správného regionu, tak i nároky na paměť. Jedna z metod využívá speciálně sestrojený deskriptor [7], který pouze vyhodnocuje afinní funkce, a tím zjistí správný region k aktuálnímu stavu. Další metoda může spočívat v pouhém snížení počtu regionů, kde je možné sloučit ty regiony, které obsahují stejné řízení. Jiná metoda může využívat algoritmus pro binární vyhledávací

strom [8] nebo metoda využívající zkrácení konečného horizontu řízení [9]. Další možností je snižování složitosti explicitního MPC [10], kde se explicitní zákon řízení, resp. funkce $\kappa(x)$, nahradí jednodušší funkcí, která bude mít menší nároky na paměť i výpočetní výkon.

2.3 Ofset při sledování reference

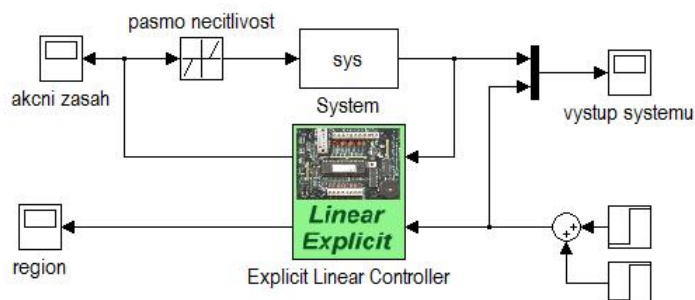
Při praktické realizaci explicitního MPC jsem se setkal s problémem trvalé regulační odchylky. Regulátor, později realizovaný ve formě bloku v Simulinku, obsahuje uvnitř stavového pozorovatele a jediné, k čemu slouží zpětná vazba ve schématu (obr. 3.10 - vazba od regulované veličiny) je zajištění správné funkce stavového pozorovatele. Pokud se z jakéhokoli důvodu naruší skutečný vztah mezi akční veličinou či stavy a výstupem, stavový pozorovatel bude špatně interpretovat aktuální stav systému a regulátor bude regulovat s trvalou regulační odchylkou. V knize [11] autor formuloval problém sledování reference a podrobněji diskutoval možné řešení, případně robustní sledování konstantní reference [12]. Pokud model soustavy není přesnou aproximací reálného procesu, pak při řízení nemusí dosáhnout výstup správné hodnoty. Stejný problém představuje neznámá porucha, která působí na vstup nebo výstup procesu. Řešením tohoto problému je rozšířit stavový popis modelu soustavy (5) o přídavný stav, který bude reprezentovat v principu výstupní poruchu jako rozdíl mezi výstupem modelu a reálné soustavy. Člen $e(k)$ lze chápat jako náhodnou poruchu typu bílý šum a tento stav poté interpretovat jako integrál z bílého šumu. Nastavení na správnou hodnotu zajistí pozorovatel. Pokud se takto rozšíří model soustavy, regulátor už poté reguluje bez trvalé regulační odchylky.

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) + e(k) \\ y(k) &= Cx(k) + Du(k)\end{aligned}\tag{5}$$

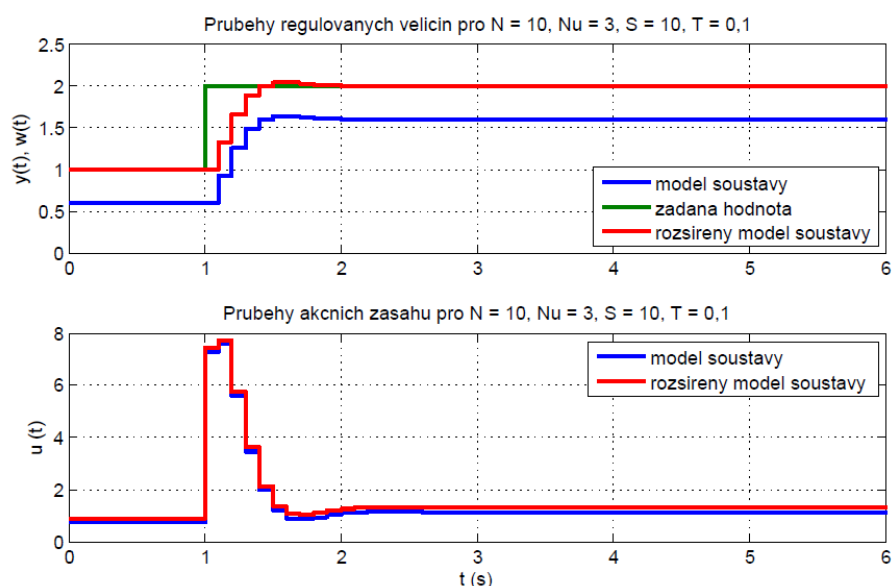
Demonstraci problému jsem předvedl v rámci simulace na příkladu, kde pro jednoduchost jsem uvažoval soustavu 1. řádu (6). Do schématu (obr. 2.2) jsem pomocí pásma necitlivosti v rozmezí 0–1 zavedl nelinearitu mezi vstupní a výstupní veličinu. Pro variantu základního modelu soustavy (6) a rozšířeného modelu soustavy jsem provedl návrh lineárního explicitního regulátoru. Výsledkem simulace jsou odezvy systému (obr. 2.3), kde je na první pohled zřejmé, že navržený regulátor pro model soustavy (6), nedokáže v simulaci dosáhnout nulové regulační odchylky. Regulátor,

který naopak pracuje s rozšířeným modelem soustavy, dokáže sledovat žádanou hodnotu bez trvalé regulační odchylky.

$$F(s) = \frac{2,5}{5s + 1} \quad (6)$$



Obr. 2.2 – Schéma simulace offsetu



Obr. 2.3 – Simulace problému s offsetem

2.4 Shrnutí výhod a omezení explicitního MPC

Obecně prediktivní řízení založené na modelu je výhodnější volbou při řízení velkého množství reálných procesů oproti standardním přístupům v podobě např. PID regulátorů. Přesto klasické MPC má minimálně jednu výraznou nevýhodu, kterou částečně odstraňuje explicitní MPC. Všechny důležité výhody a omezení související s explicitním MPC jsou představeny v následující části.

Výhody:

- Výrazné snížení požadavku na výpočetní výkon
- Vhodné pro relativně rychlé procesy
- Možnost zavedení omezujících podmínek
- Snížení nároků na paměť
- Optimální řízení => ekonomická úspora (nižší náklady, vyšší zisky)

Nevýhody:

- Exponenciální nárůst počtu regionů v závislosti na počtu omezení
- Vyšší počet regionů má negativní vliv na výpočetní výkon, paměť a cenu implementace řízení
- Požadavek na dostatečně přesný model

3 Implementace explicitního MPC

V předchozích kapitolách jsem se věnoval popisu dané problematiky formulováním základních myšlenek a principů prediktivního řízení a především jeho explicitní formy. V této další kapitole se zabývám implementací daného způsobu regulace. Nejdříve jsem musel vyhledat vhodný nástroj, který mi pomohl při návrhu explicitního prediktivního regulátoru. Následně jsem tento regulátor vyzkoušel v rámci simulací, abych ověřil jeho funkci. Po úspěšných simulacích jsem se začal zabývat jeho praktickým nasazením na reálné soustavy.

3.1 Software pro návrh

Pro návrh explicitního MPC je dostupných několik softwarových nástrojů, které rozšiřují možnosti programu MATLAB. Jedna z možností je nástroj, který se jmenuje Hybrid Toolbox [13]. Tento toolbox je volně dostupný pro nekomerční použití a umožňuje navrhnout explicitní zákon řízení pro lineární i hybridní systémy. Obsahuje množství funkcí pro návrh, simulaci, vizualizaci, analýzu systému nebo generování kódu v jazyce C, kde tyto funkce usnadňují realizaci MPC. Podobné a další funkce jsou zastoupeny i v dalším nástroji, který se nazývá Multi-Parametric Toolbox. Moje snaha se zejména soustředila na Hybrid Toolbox, protože tento toolbox je zcela dostačující pro realizaci explicitního MPC.

Mezi hlavní funkce Hybrid Toolboxu patří [13]:

- Návrh, simulace a analýza hybridních systémů
- Návrh MPC
- Návrh explicitního MPC
- Generování kódu v jazyce C
- Rozšiřující knihovna prostředí Simulinku v programu Matlab
- Značná podpora algoritmů pro řešení optimalizační úlohy

Hybrid Toolbox obsahuje různé demo příklady, které mi pomohli seznámit se s programovým prostředím a možnostmi vytváření příslušných schémat v Simulinku. Tento toolbox rozšiřuje knihovnu v prostředí Simulinku o několik užitečných bloků (obr. 3.1), kde nejdůležitější z nich je Explicitní Lineární Regulátor, který jsem používal při realizaci explicitního MPC. Tento konkrétní blok je vhodný pro lineární soustavy.

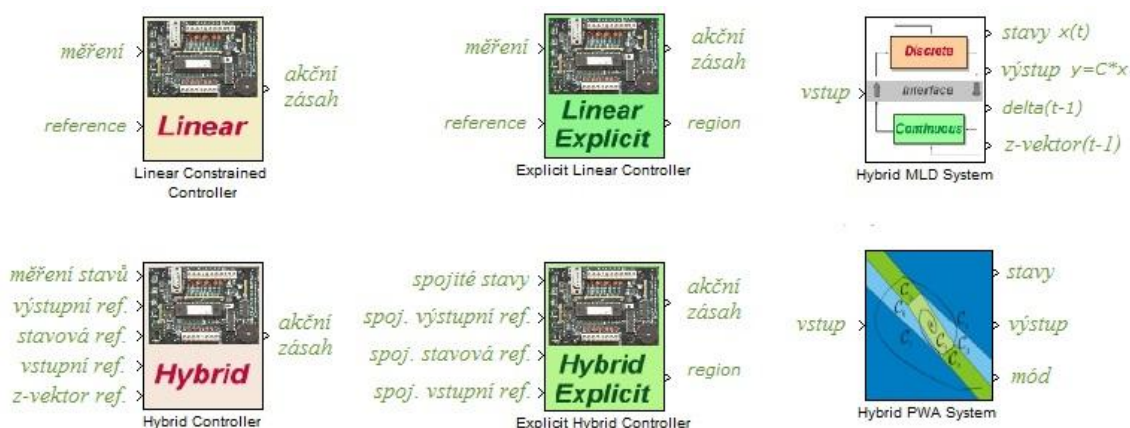
Explicitní MPC jsem aplikoval na reálné soustavy, kde tento typ regulátoru dostatečně vyhovoval požadavkům pro řízení. Návrh takového regulátoru jsem provedl pomocí programového kódu sestavného z příkazů, které Hybrid Toolbox podporuje.

V Hybrid Toolboxu se návrh provádí tak, že se nejdříve získá diskretní popis soustavy ve formě diskretního stavového popisu. Následně se definují určitá nastavení a omezení v podobě limitů akčních zásahů, váhové funkce vystupující v kvadratickém kritériu, horizont řízení a horizont predikce. Dalším důležitým parametrem je typ výsledného regulátoru, který buď reaguje na rozvážení (*reg*), nebo sleduje žádanou hodnotu (*track*). Přestože jsem výhradně využíval nastavení pro sledování žádané hodnoty, regulátor pracoval s trvalou regulační odchylkou (především u modelu motoru). Sledování reference u daného typu regulátoru (lineární explicitní regulátor) je podmíněno zcela lineárními systémy. Pokud v reálném procesu bude přítomna nelinearita (např. necitlivost), poruší se lineární vztah mezi vstupem a výstupem soustavy. Lineární explicitní regulátor, který jsem využíval, stále předpokládal, že pracuje s lineárním procesem, a proto reguloval s trvalou regulační odchylkou. Řešením problému bylo rozšíření modelu soustavy o přídatný stav (viz kapitola 2.3). Nelinearita procesu nemusí být jediný důvod, který způsobuje trvalou regulační odchylku. Během praktické realizace jsem zjistil, že výsledná kvalita regulace je také velmi citlivá na nastavení parametrů návrhu (např. nevhodně zvolená vzorkovací perioda může ovlivnit kvalitu regulačního pochodu).

Parametr typu regulátoru zároveň udává, které kvadratické kritérium se použije v optimalizační úloze. Kvadratické kritérium (7), které Hybrid Toolbox používá pro sledování reference a které využívám při každém návrhu regulátoru pro reálné řízení, je v následující podobě

$$J(k) = \sum_{k=0}^{N-1} \{[y(k) - r]' S [y(k) - r] + du'(k) T du(k)\} \quad (7)$$

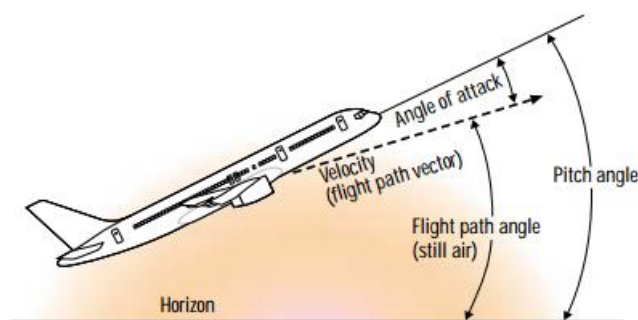
Z příslušného nastavení se sestaví regulátor pomocí příkazu **lincon**. V další části se definují další doplňující nastavení potřebná pro sestavení lineárního explicitního regulátoru a pomocí příkazu **expcon** se převede regulátor do explicitní podoby, který je hledaným výstupem.



Obr. 3.1 – Hybrid Toolbox knihovna

3.2 Simulační příklad – stíhací letoun

Pro předvedení funkčnosti explicitního MPC jsem si vybral ukázkový simulační příklad, který je dostupný v rámci toolboxu. Je to jeden z mnoha demo příkladů a jedná se o stíhací letoun F16. Tento příklad jsem zvolil z toho důvodu, protože v rámci demo příkladů je to jedna z MIMO soustav, která obsahuje 2 vstupy (úhel výškovky a tzv. *flaperon* úhel (flaperon – způsob ovládání letadla, který spojuje aspekty klapky a křidélek)) a 2 výstupy (obr. 3.2 - úhel náběhu (attack angle) a úhel podélného sklonu (pitch angle)). Linearizovaný dynamický model soustavy je reprezentován stavovým popisem (8).



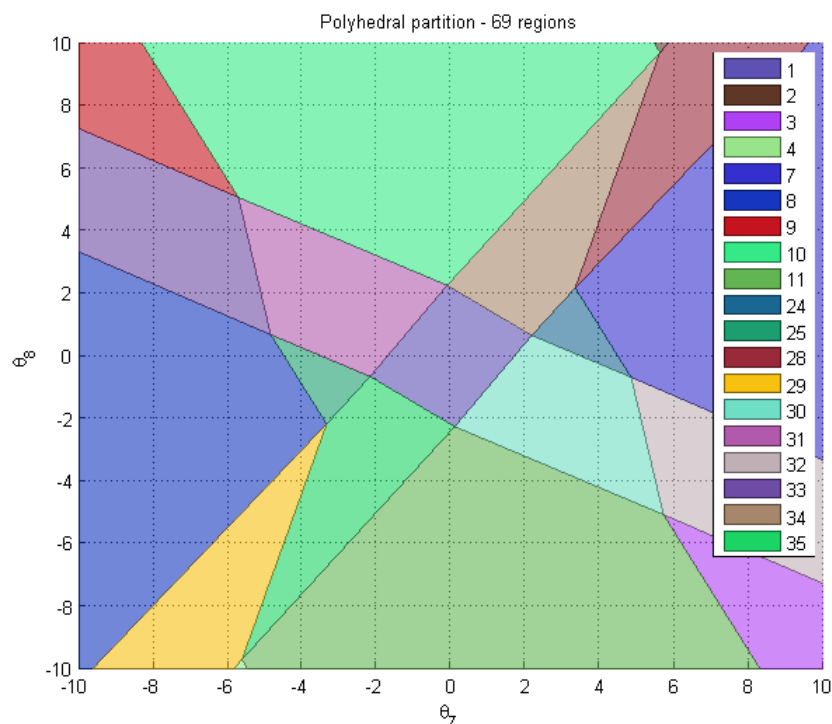
Obr. 3.2 - Význam úhlů u letadla [14]

$$\dot{x} = \begin{bmatrix} -0,0151 & -60,5651 & 0 & -32,174 \\ -0,0001 & -1,3411 & 0,9929 & 0 \\ 0,00018 & 43,2541 & -0,86939 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -2,516 & -13,136 \\ -0,1689 & -0,2514 \\ -17,251 & -1,5766 \\ 0 & 0 \end{bmatrix} u \quad (8)$$

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x$$

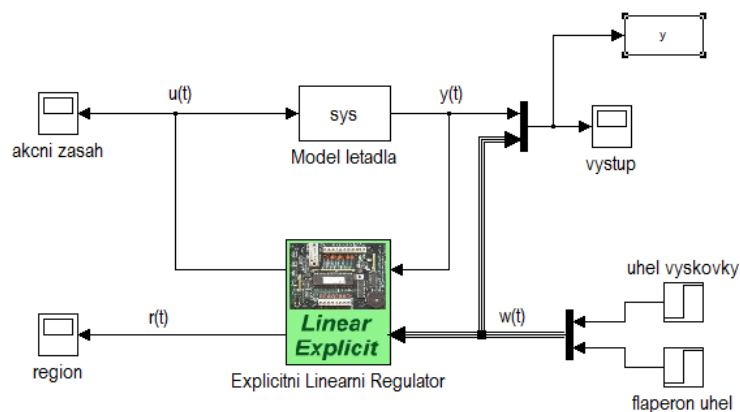
Systém popsaný ve spojité oblasti bylo nejdříve nutné převést na diskrétní model (diskrétní stavový popis), na kterém je založen návrh a princip samotného MPC. Pro diskretizaci modelu se použila vzorkovací perioda $T_s = 0,01$ s. Požadavkem pro regulaci bylo omezení pro oba vstupy, resp. pro úhel výškovky a flaperon úhel, v rozsahu $\pm 25^\circ$ a cílem regulace bylo dosažení nulové regulační odchylky. Následně se nastavily všechny důležité parametry tak, aby se mohl provést návrh regulátoru. Mezi tyto parametry patří horizont predikce, který je nastaven na hodnotu $N = 60$, váhové funkce jsou u MIMO systému představovány maticemi a konkrétně zde jsou to jednotkové matice, kde váha regulační odchylky byla $S = [10 \ 0; 0 \ 10]$ a váha přírůstků akční veličiny byla $T = [0,01 \ 0; 0 \ 0,01]$. Typ regulátoru odpovídal požadavku sledování žádané hodnoty.

V dalším kroku se provede výpočet lineárního explicitního regulátoru, za pomoci dříve zmíněného příkazu *expcon*, který převede regulátor z prvního kroku do explicitního tvaru. Opět se zde definují určité parametry, mezi které patří např. rozsahy stavů, vstupních a výstupních veličin a další nastavení, která jsou podrobněji popsána v nápovědě příslušného příkazu v Matlabu. Po výpočtu explicitního regulátoru je možné přistoupit k simulaci, ale v rámci programu se ještě vykreslí nalezené kritické regiony (obr. 3.3) a to pomocí příkazu *plotsection*. Celkový počet vygenerovaných kritických regionů dosáhl hodnoty 71. V některých případech se celkový počet regionů zredukuje tím, že se pokusí program sloučit regiony se stejnými akčními zásahy. V tomto konkrétním příkladu tato skutečnost nastala a celkový počet regionů se snížil na hodnotu 69. Zároveň je důležité zmínit, že celkový počet regionů je ovlivněn nastavenými parametry, především horizont řízení nebo např. horizont predikce úzce související s periodou vzorkování.

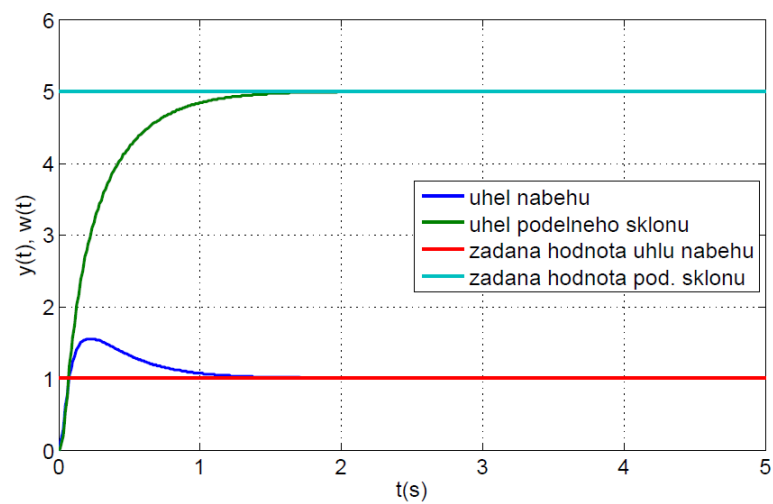


Obr. 3.3 – Kritické regiony

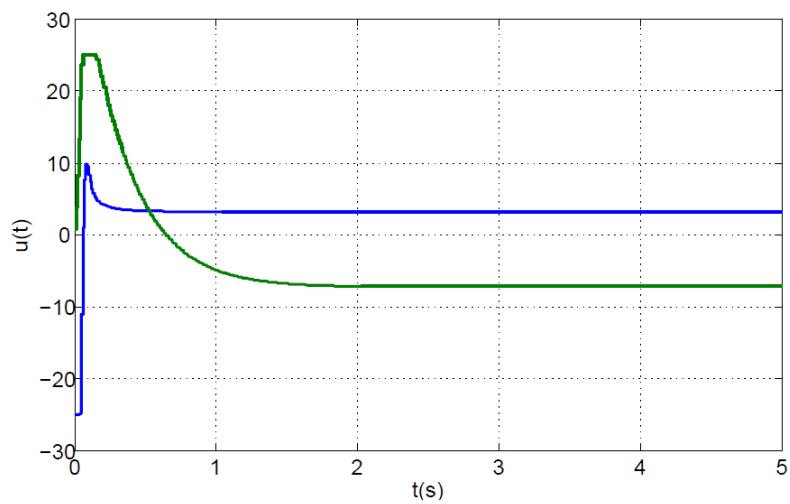
Nakonec se provádí simulace modelu letadla s navrženým lineárním explicitním regulátorem (obr. 3.4) a vyhodnocují se regulační pochody. Výsledkem jsou jednak odezvy systému (obr. 3.5), kde lze jednoznačně potvrdit, že regulovaná veličina opravdu sleduje žádanou hodnotu, a tedy vybraný typ regulátoru splňuje očekávání. Dále se sledují akční zásahy, zdali respektují omezení uvažovaná při návrhu ($\pm 25^\circ$) a ze simulovaných průběhů (obr. 3.6) je zcela patrné, že opravdu tato omezení dodržují, akční zásahy nejsou větší než hodnota 25 ani menší jak hodnota -25 . Nakonec se sledují použité kritické regiony během regulace (obr. 3.7).



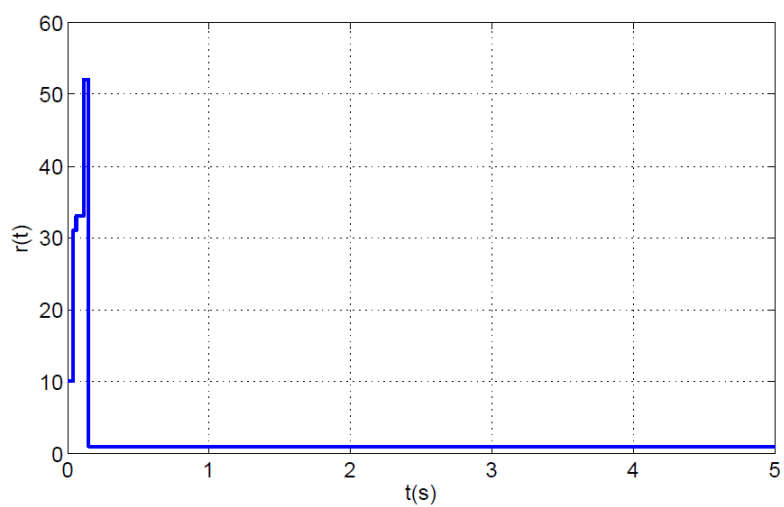
Obr. 3.4 – Schéma pro simulaci



Obr. 3.5 – Odezva systému



Obr. 3.6 – Akční zásahy

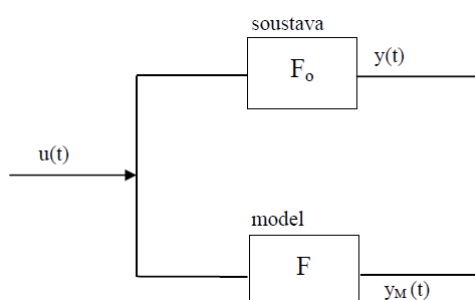


Obr. 3.7 – Výběr regionů během regulace

3.3 Proces identifikace reálné soustavy

Předpokladem efektivního řízení reálného procesu je znalost jeho chování a vlastností. Proto první seznámení s reálnou soustavou by mělo začínat její identifikací. Proces identifikace umožňuje získat model reálného systému, který je důležitý z hlediska návrhu úspěšné regulace. Zejména je významný u takových přístupů řízení, kde model systému je nedílnou součástí struktury řízení, kterým je i explicitní MPC. Jedná se o postup, který umožňuje zjistit chování daného systému ve formě statických a dynamických vlastností, které se získají různými měřeními. Tato měření se následně zpracují pomocí identifikačních nástrojů a postupů tak, aby výsledkem bylo získání matematického modelu. Model je zobrazením důležitých vlastností reálného systému, které vyjadřují vztah mezi vstupem a výstupem systému. Model může být reprezentován matematickými rovnicemi, grafy a tabulkami atd. Vztah mezi vstupem a výstupem lze obecně charakterizovat operátorem transformace $F(s)$ (9). Úloha identifikace spočívá v nalezení operátoru transformace pro model $F(s)$ tak, aby s určitou přesností byl blízký skutečnému operátoru transformace $F_0(s)$. Oba operátory mohou mít obecně různou strukturu. Srovnání identifikovaného modelu a reálné soustavy (obr. 3.8) se provádí tak, že se posuzuje „blízkost“ podle odezvy reálné soustavy $y(t)$ a modelu $y_M(t)$ na stejný budicí signál.

$$Y = F(s) \cdot U \quad (9)$$



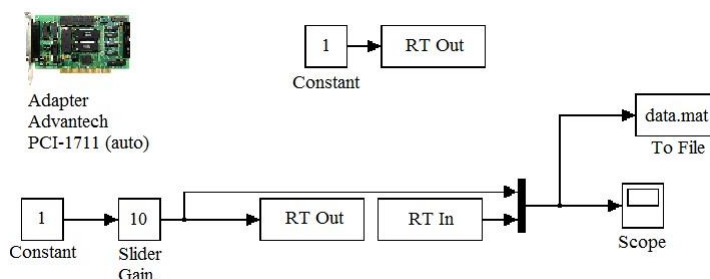
Obr. 3.8 – Porovnání modelu a reálného systému [15]

Výsledkem identifikace je přenosová funkce, která představuje daný operátor transformace $F(s)$. Díky přenosové funkci je možné provádět různé simulace a především návrh regulačních struktur bez rizika, že by mohlo dojít k poškození reálné soustavy, protože model prakticky napodobuje chování reálného systému. Simulacemi ověřené návrhy regulátorů a regulačních struktur lze poté implementovat na reálný

system a zjistit, zda simulované odezvy odpovídají reálným průběhům. Samozřejmě jsou odchylky v určité toleranci, protože v žádném případě se nedosáhne ideální identifikace, která by přesně a dokonale popisovala daný systém. Reálný systém si nikdy nezachová po celou dobu životnosti své vlastnosti, které se identifikují. Vlivem okolního prostředí, stárnutí a opotřebování vnitřních komponent se vlastnosti systému mohou změnit a dříve provedená identifikace už po určité době nemusí zcela odpovídat.

3.4 Schéma měření a regulace

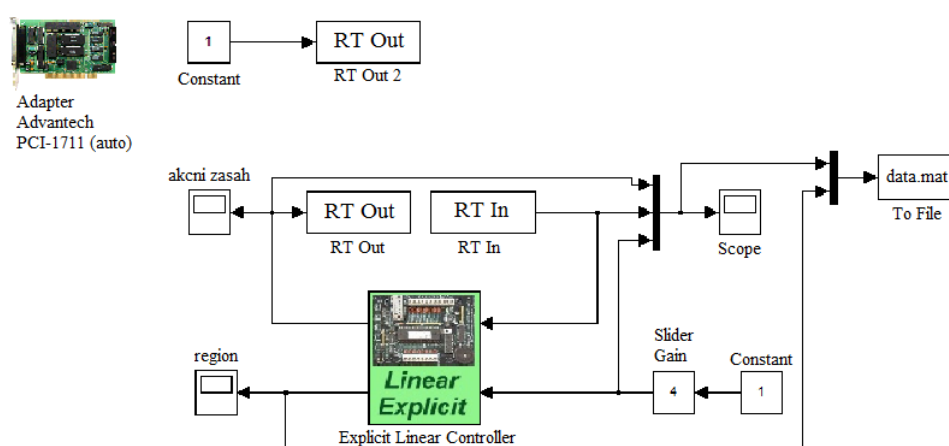
Pro proces identifikace a měření požadovaných odezev reálného systému jsem využíval speciální multifunkční měřicí kartu. Tato karta má typové označení *Adventech PCI-1711* a ke komunikaci s kartou jsem využíval prostředí Simulinku, ve kterém jsem realizoval schéma měření (obr. 3.9), pomocí kterého jsem identifikoval statické a dynamické vlastnosti systému. Nejdůležitější částí tohoto schématu (obr. 3.9) je blok *Adapter*, který slouží k inicializaci měřicí karty a pro komunikaci s jednotlivými vstupy a výstupy karty. Na tyto kanály jsou přiváděny řídicí a měřicí veličiny reálného systému. Pro komunikaci s příslušnými kanály jsem používal bloky *RT Out* a *RT In*. Ve schématu je umístěn další užitečný blok, kterým je *Slider Gain*, kde tento blok umožňuje manuálně měnit zesílení během simulace.



Obr. 3.9 – Schéma měření

Bloky *RT Out* a *RT In* umožňují podrobnější nastavení jednotlivých kanálů karty (vstupní/výstupní kanály), kde lze nastavit především důležitý parametr vzorkovací periodu, typ vstupu a výstupu, zda bude digitální nebo analogový, případně v jakém rozsahu se bude měřit napětí a vcelku zajímavý parametr u bloku *RT Out*, který nastavuje defaultní hodnotu, která bude na příslušném výstupu v době spuštění měření a při jeho ukončení. Vzorkovací periodu jsem volil řádově v setinách sekundy a obvykle $T_s = 0,01 \text{ s}$, protože reálné soustavy, se kterými jsem pracoval, jsou relativně rychlé procesy a tato hodnota vzorkování byla vyhovující.

Schéma (obr. 3.10), které jsem používal při implementaci explicitního prediktivního regulátoru, se v principu tolik neodlišuje od schématu měření (obr. 3.9). Hlavní rozdíl je v přidaném bloku *Explicit Linear Controller*, který aplikuje navržený regulátor pomocí Hybrid Toolboxu. Příslušný blok regulátoru je reprezentován tzv. *S-funkcí*. *S-funkce* umožňuje přidat do schématu v prostředí Simulinku vlastní programový kód v jazyce C nebo ve formě M-file. Do bloku regulátoru vstupují dvě veličiny - žádaná hodnota a měřená regulovaná veličina z důvodu zpětné vazby a správné funkce stavového pozorovatele uvnitř regulátoru a vystupují akční zásahy spolu s vybranými regiony.



Obr. 3.10 – Schéma regulace

3.5 Reálná soustava dvou DC motorů – popis soustavy

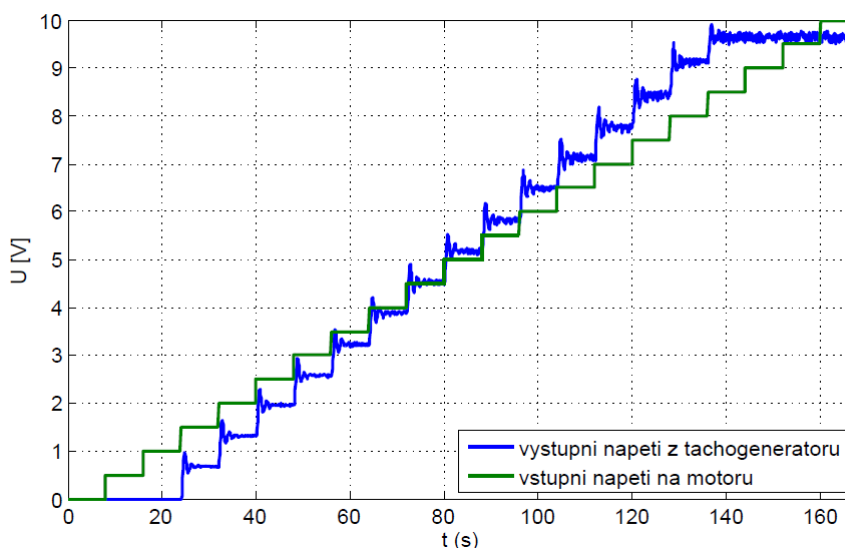
Prvním reálným systémem, kde jsem se pokusil aplikovat explicitní MPC, byl model motoru (obr. 3.11). Jedná se o dynamický systém v podobě SISO soustavy, kde tento systém zahrnuje na jedné straně stejnosměrný motor a na straně druhé tachogenerátor. Motor je spojený s tachogenerátorem pomocí pružné spojky a představuje vstup systému. Stejnosměrný motor je ovládán vstupním napětím z měřicí karty v rozsahu 0–10 V. Otáčky motoru se přenášejí pružnou spojkou na tachogenerátor, který slouží jako měřicí člen, a resp. je výstupem systému. Výstupní signál z tachogenerátoru je zaznamenáván měřicí kartou v rozsahu napětí 0–10 V.



Obr. 3.11 – Reálný model motoru

3.5.1 Statické vlastnosti

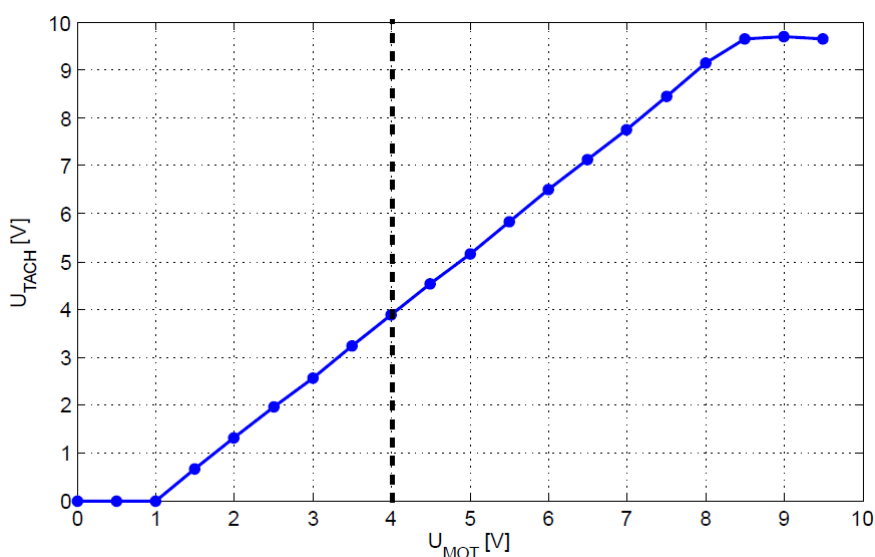
Statické vlastnosti popisují ustálený stav systému, který vzniká v momentě, kdy na vstupu soustavy je signál s konstantní hodnotou a zároveň výstup soustavy se nemění. Ustálený stav je tak děj, který nastává po odeznění přechodového děje. Statická charakteristika je způsob, jak graficky znázornit statické vlastnosti systému, kde tato charakteristika je definována jako závislost ustálené hodnoty vstupu na ustálené hodnotě výstupu. Není nic jednoduššího, než se touto definicí řídit a výsledkem je požadovaná statická charakteristika daného modelu motoru. Statickou charakteristiku (obr. 3.12) jsem naměřil v celém rozsahu napěťových hodnot 0–10 V s krokem 0,5 V. Při měření jsem uvažoval hodnotu vzorkovací periody $T_s = 0,01$ s.



Obr. 3.12 - Průběh měření statické charakteristiky motoru

Z naměřené charakteristiky je na první pohled patrné, že není zcela lineární, v počátku soustavy souřadnic neprochází nulovou hodnotou, ale statická charakteristika je mírně posunutá. Je to způsobeno tím, že při nízkých hodnotách napětí nedojde

k roztočení motoru, protože se nevytvoří takový hnací moment, který by dokázal překonat záběrový moment. Teprve při velikosti cca 1,5 V na vstupu dochází k překonání záběrového momentu a roztočení motoru. Z důvodu rychlého vzorkování jsem získal velké množství dat, které pro další zpracování nepředstavuje až takový problém, nicméně z měření je vidět, že výstup systému se zcela neustálí na konstantní hodnotě, kde u prvních několika skoků budícího signálu tento problém není tak patrný, ale při zvyšování otáček se kmity kolem ustálené hodnoty projevují více. Přestože rozmezí kmitů není velké, pohybuje se od 0,1 V do 0,3 V, při zpracování hodnot jsem použil průměrovací filtr, který z posledních 10 hodnot před změnou budícího signálu vypočítal průměr. Upravená data jsem poté vykreslil do grafu a výsledkem byla hledaná statická charakteristika (obr. 3.13). Statická charakteristika je z velké části lineární a mohl bych řízení realizovat téměř v celém rozsahu. Přesto jsem vybral pouze určitý úsek, kde vhodný pracovní bod jsem zvolil při hodnotě 4 V na vstupu a pracovní oblast v rozmezí 1–2 V kolem pracovního bodu.

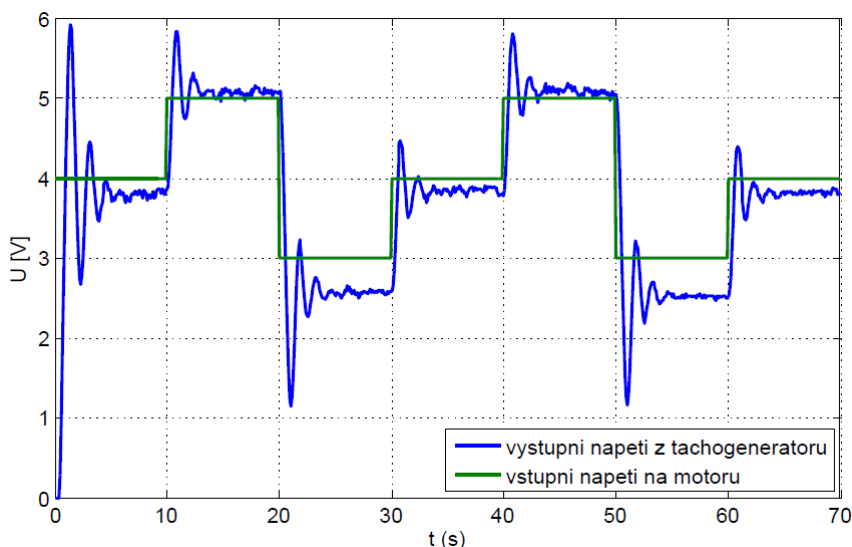


Obr. 3.13 – Statická charakteristika motoru

3.5.2 Dynamické vlastnosti, identifikace přenosu

Druhou a důležitější částí identifikace je získání modelu systému, který popisuje dynamické chování. Jsou to vlastnosti, které vyjadřují, jak se systém chová při přechodu mezi ustálenými stavy. Výsledný model systému může být reprezentován mnoha způsoby, např. stavovým popisem, ve formě přechodové odezvy nebo v mém případě přenosovou funkcí. Abych mohl získat přenos systému, naměřil jsem jeho dynamické vlastnosti tak, že v okolí zvoleného pracovního bodu jsem prováděl malé skoky

budicího signálu (obr. 3.14). Naměřená data jsem poté upravil do podoby umožňující jejich identifikaci. V počátku měření přechází systém z nedefinovaného stavu do pracovního bodu, a proto tento neznámý úsek je v procesu identifikace problematický. Provedl jsem tedy linearizaci v pracovním bodě, která spočívá v zanedbání nedefinovaného úseku v začátku měření a posunu naměřených dat do počátku.



Obr. 3.14 – Průběh měření dynamických vlastností motoru

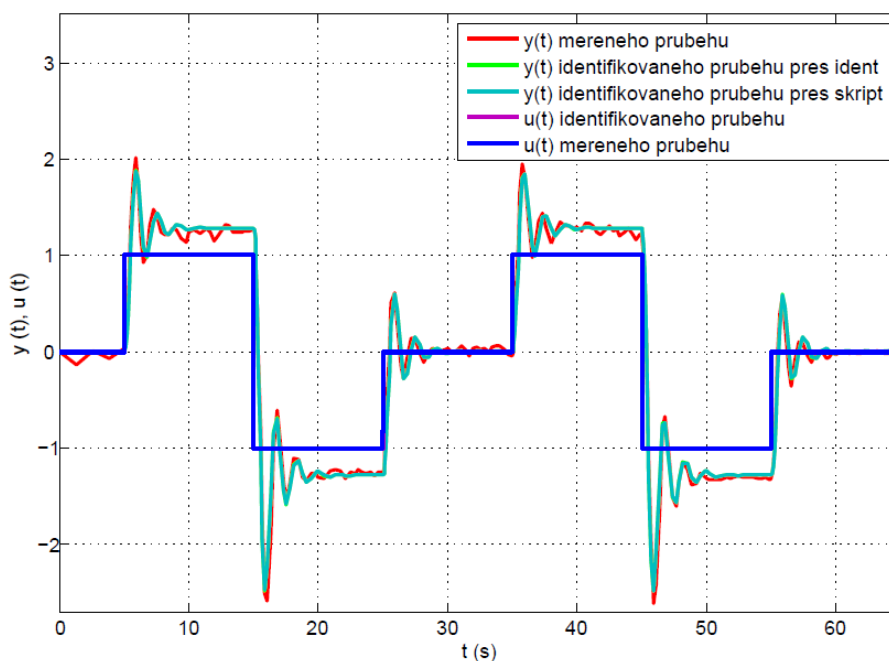
Upravená data jsem identifikoval pomocí nástroje, který je standardní součástí programu Matlab. Tento nástroj se jmenuje *Identification Toolbox* a spouští se příkazem *ident*. Umožňuje určit přenosovou funkci z naměřených dat. Identifikační nástroj vypočítal neznámé parametry přenosové funkce, která se v porovnání s naměřeným průběhem shoduje na 91,39 % (obr. 3.15). Pro kontrolu jsem využil další možnost identifikace v podobě skriptu, jehož základem je funkce *fminsearch*, která hledá lokální minimum vícerozměrové funkce bez použití gradientu. V tomto případě se minimalizovala kritériální funkce (10).

$$J(X) = \sum_{i=1}^N [y(i) - y_M(i, X)]^2 \quad (10)$$

Výsledky z obou identifikačních přístupů, nalezené přenosové funkce, se dokonale shodovali, nicméně ve většině případů jsem využíval nástroje v Matlabu, protože jeho ovládání je vcelku intuitivní a nalezené přenosy jsou dostatečně přesné vzhledem k naměřeným průběhům. Z porovnání průběhů (obr. 3.15) je zřejmé, že identifikovaný přenos je správný, protože jak průběhy vstupních signálů (buzení), tak

průběhy výstupních signálů ze soustavy se shodují a identifikovaný přenos (11), který je 3. řádu, dostatečně přesně aproximuje dynamiku reálného systému.

$$F(s) = \frac{1.282}{0.006068s^3 + 0.07414s^2 + 0.2026s + 1} \quad (11)$$



Obr. 3.15 – Porovnání naměřených průběhů a identifikovaného přenosu

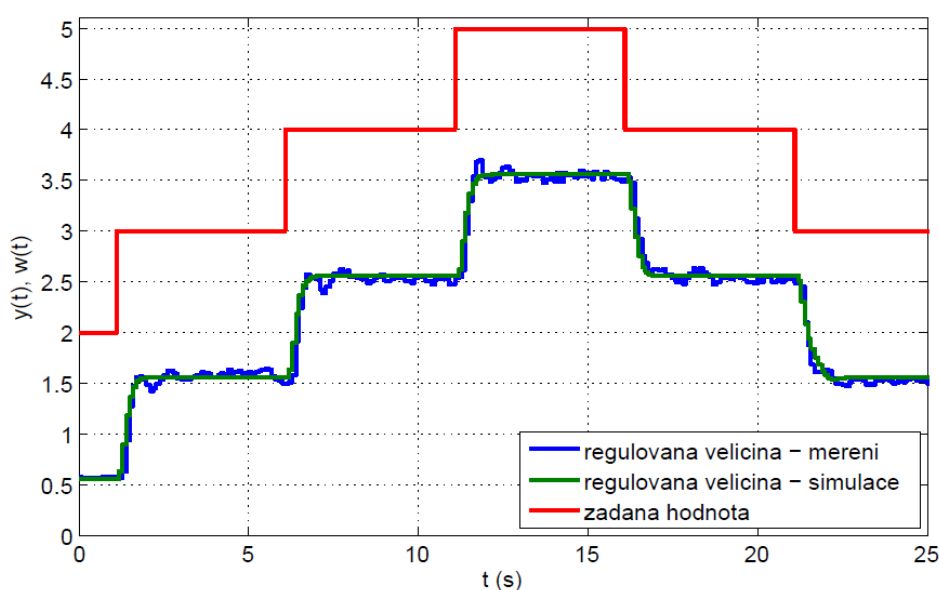
3.5.3 Praktická realizace explicitního MPC

Identifikací jsem získal potřebný model reálného systému, který je nezbytnou součástí explicitního MPC. Následně jsem provedl návrh explicitního lineárního regulátoru pro reálnou soustavu. Postup návrhu je stejný jako u simulačního příkladu. Ve zkrácené formě, definoval jsem potřebné parametry a nastavení pro návrh regulátoru, který jsem provedl pomocí příkazu *lincon*. Navržený regulátor se poté převede do explicitní podoby. Všechny důležité parametry, které jsem nastavil a používal při regulaci, jsou uvedeny v tabulce (tab. 1).

Tab. 1 – Parametry regulace modelu motoru

Parametr	Nastavená hodnota	Parametr	Nastavená hodnota
Vzorkovací perioda	$T_s = 0,1 \text{ s}$	Horizont predikce	$N = 10$
Váha regulační odchylky	$S = 10$	Typ regulátoru	track – sledování reference
Váha přírůstků akční veličiny	$T = 0,1$		
Horizont řízení	$N_U = 2$	Počet regionů	7

Nalezený lineární explicitní prediktivní regulátor jsem implementoval do regulačního schématu (obr. 3.10) a s ohledem na zvolenou pracovní oblast jsem aplikoval řízení. Výsledek regulace nebyl zcela uspokojivý, protože z naměřených regulačních pochodů (obr. 3.16) je na první pohled znatelný problém v podobě trvalé regulační odchylky. Tento problém je s největší pravděpodobností způsobený necitlivostí reálné soustavy při nízkých hodnotách vstupního napětí, a který také způsobuje, že statická charakteristika (obr. 3.13) je mírně posunutá a není tak zcela lineární. Řešení tohoto problému je diskutováno v kapitole 2.3, resp. model soustavy se rozšíří o další stav tak, aby se získal rozšířený stavový popis.



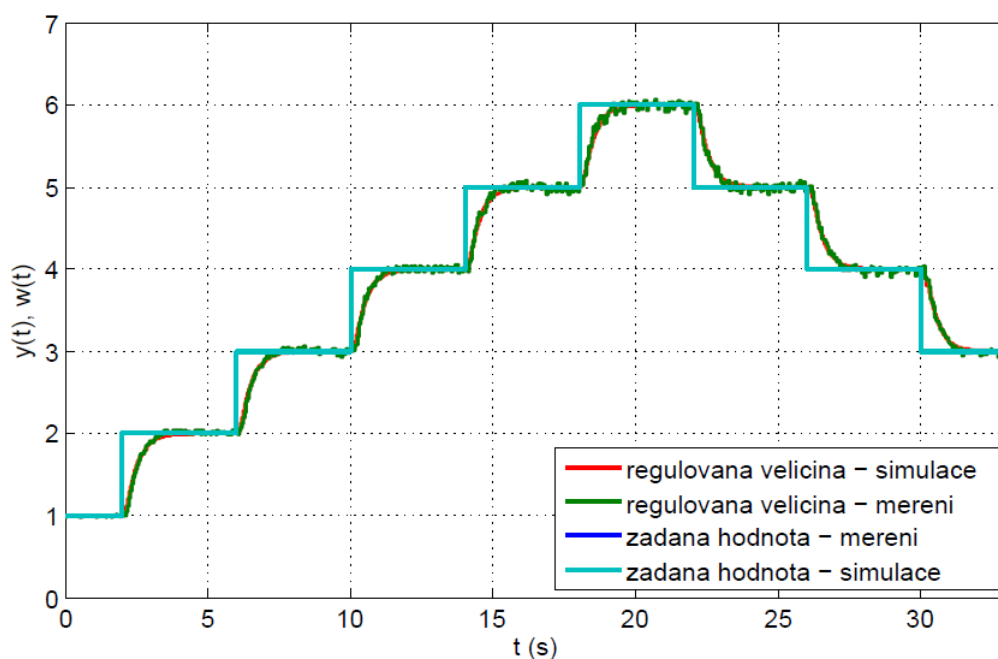
Obr. 3.16 – Regulace s offsetem

Tato úprava návrhu regulátoru nebyla jediná, kterou jsem provedl, změnil jsem i některé parametry, které ovlivňují regulační pochod, a to vzorkovací periodu a horizont predikce, případně horizont řízení, kde konkrétní hodnoty jsou uvedeny v tabulce (tab. 2).

Tab. 2 – Upravené návrhové parametry

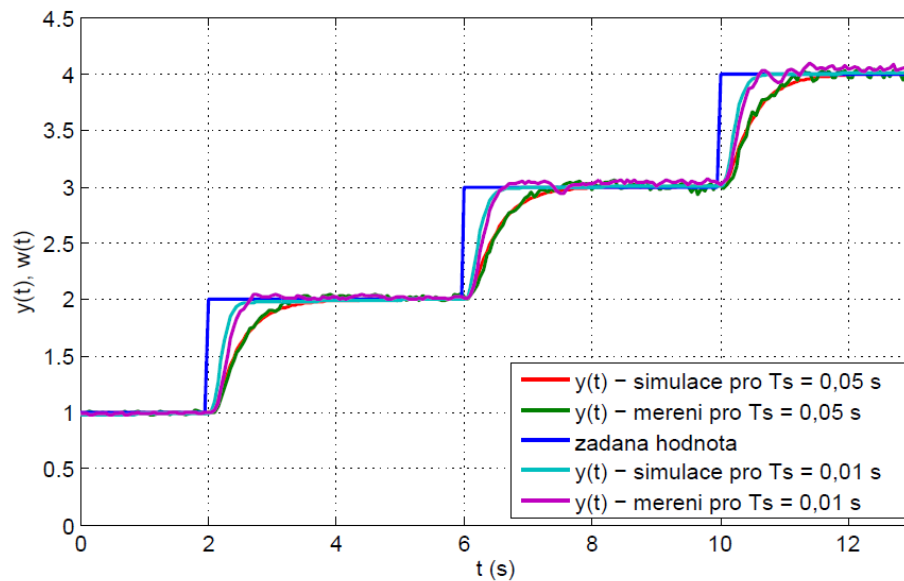
Parametr	Nastavená hodnota	Nastavená hodnota
Vzorkovací perioda	$T_S = 0,05 \text{ s}$	$T_S = 0,01 \text{ s}$
Váha regulační odchylky	$S = 10$	$S = 10$
Váha přírůstků akční veličiny	$T = 0,1$	$T = 0,1$
Horizont řízení	$N_U = 2$	$N_U = 4$
Horizont predikce	$N = 70$	$N = 70$
Vygenerovaný počet regionů po návrhu	7	61

Upravený návrh lineárního explicitního prediktivního regulátoru jsem implementoval na reálnou soustavu a výsledkem byly mnohem kvalitnější regulační pochody. Sledoval jsem odezvy systému - regulované veličiny (obr. 3.17), akční zásahy (obr. 3.19) a výběr regionů (obr. 3.20) během regulace. Porovnal jsem naměřené odezvy se simulovanými, kde je na první pohled zřejmé, že se téměř dokonale shodují. Tato skutečnost je dalším důkazem správně provedené identifikace, nicméně u průběhů akčních zásahů dochází v určitých oblastech k menším odchylkám, a proto i tento důkaz vypovídá o tom, že nelze dosáhnout ideální identifikace.



Obr. 3.17 – Odezvy systému motoru pro $T_S = 0,05$ s

Při měření jsem prováděl několik skoků žádané hodnoty v okolí vybraného pracovního bodu. Změnou vzorkovací periody na $T_S = 0,05$ s, jsem oproti předchozímu měření dosáhl pomalejšího regulačního pochodu, ale tento výsledek je velice závislý na vybraných parametrech při návrhu. Z toho důvodu jsem vyzkoušel změnu vzorkovací periody na hodnotu $T_S = 0,01$ s a parametr horizont řízení (viz tab. 2), který měl především vliv na vyšší počet vygenerovaných regionů. Výsledkem změny vzorkovací periody bylo urychlení regulačního pochodu v podobě zkrácení doby ustálení z 2 s na 0,6 s (obr. 3.18).

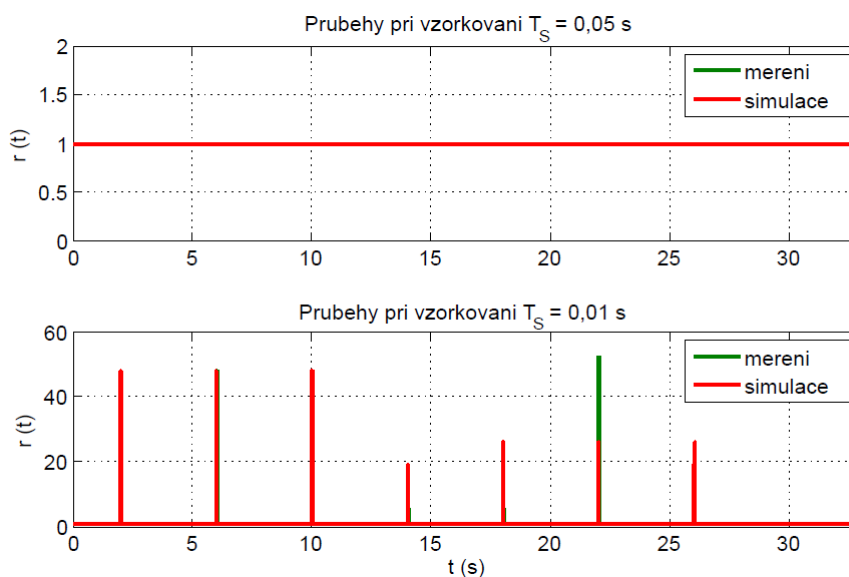


Obr. 3.18 – Porovnání odezev pro $T_s = 0,01$ s a $T_s = 0,05$ s

Poslední graf (obr. 3.20) zobrazuje srovnání simulace a měření pro výběr regionů v rámci regulace. Horní část představuje regulaci při vzorkování $T_s = 0,05$ s, kde tento graf není zcela vypovídající, protože celkový počet vygenerovaných regionů je nízký. Během celé regulace se systém nedostane do stavu, kde by se mohl využít i jiný region. Z toho důvodu jsem uvedl spodní graf, který představuje regulaci při vzorkování $T_s = 0,01$ s, kde vygenerovaný počet regionů je mnohem vyšší. Ze srovnání simulace a měření je v tomto grafu patrné, že z velké části se výběr regionů shoduje, pouze v určitých oblastech se odlišuje. Je to způsobené tím, jak stavový pozorovatel zrovna interpretuje stav, ve kterém se systém reálně nachází. Přesto jsou regulační pochody kvalitní a srovnatelné.



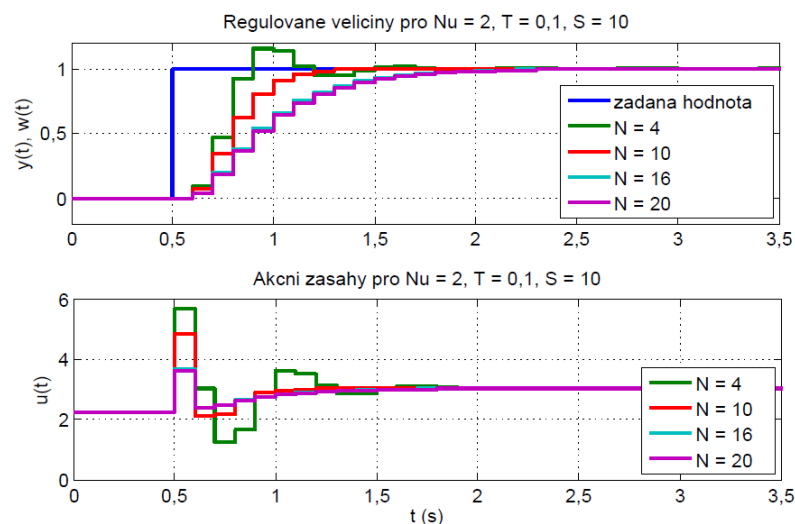
Obr. 3.19 – Akční zásahy u regulace motoru pro $T_s = 0,05$ s



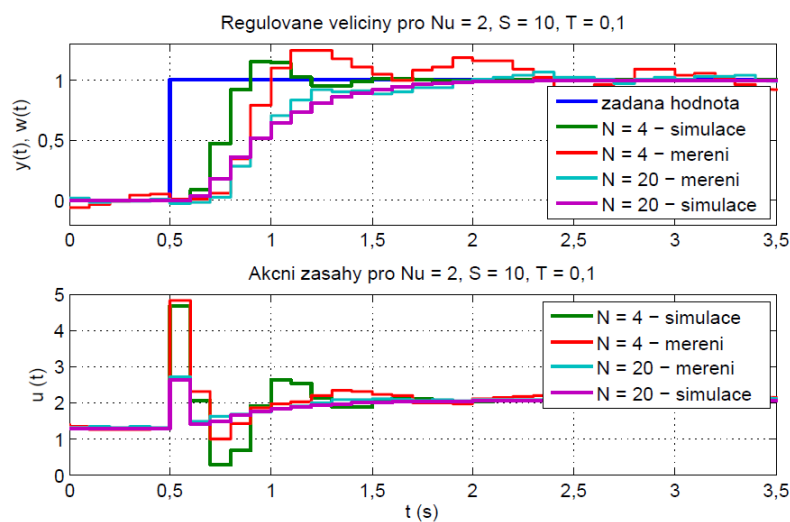
Obr. 3.20 – Výběr regionů při regulaci motoru pro $T_s = 0,05$ s a $T_s = 0,01$ s

3.5.4 Vliv parametrů

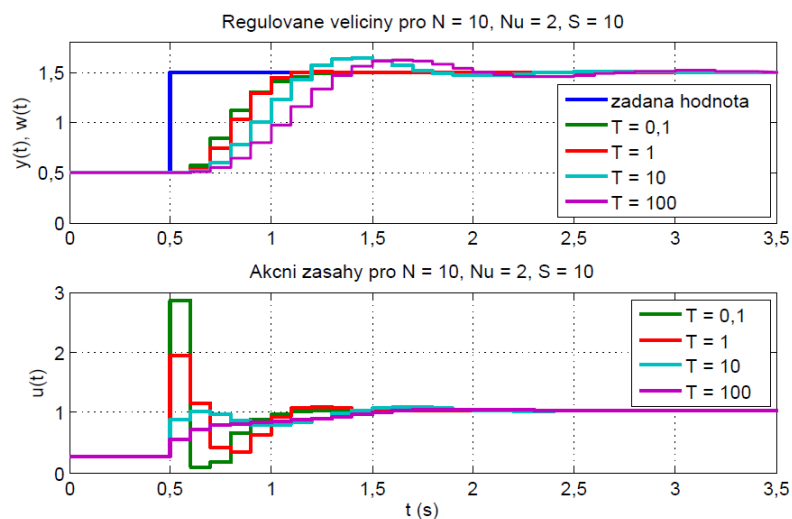
Z předchozích měření je zřejmé, že některé parametry mají vliv na regulační pochod. Samozřejmě je to logické, musí existovat způsoby, kterými se ovlivňují požadavky a kvalita regulace. Z toho důvodu jsem provedl několik simulačních experimentů, kde jsem postupně měnil všechny důležité parametry a v rámci reálného měření jsem některé vybrané změny ověřil prakticky. Tímto jsem získal přehled o tom, které parametry a jaký vliv mají na regulační pochod. Pro představu jsem do této části umístil pouze některé vybrané simulace (obr. 3.21), (obr. 3.23) a srovnání s reálnými měřeními (obr. 3.22), (obr. 3.24), aby byly vidět názorné důsledky jednotlivých změn parametrů. V závěru této kapitoly je shrnutí všech důležitých parametrů a důsledků jejich změn (tab. 3). Např. změna horizontu predikce (obr. 3.21) způsobí, že čím je větší tato hodnota, tím jsou regulační pochody pomalejší a akční zásahy se snižují. Pro hodnotu $N = 4$ je doba ustálení $0,9$ s, pro hodnotu $N = 20$ je naopak doba ustálení delší, konkrétně $1,6$ s. Ve srovnání s reálným měřením (obr. 3.22) při hodnotě $N = 20$ jsou odezvy téměř totožné, při hodnotě $N = 4$ se především regulované veličiny mírně odlišují, ale opět tato odchylka je způsobena neideální identifikací. Pro všechna reálná měření jsem uvažoval vzorkovací periodu $T_s = 0,1$ s.



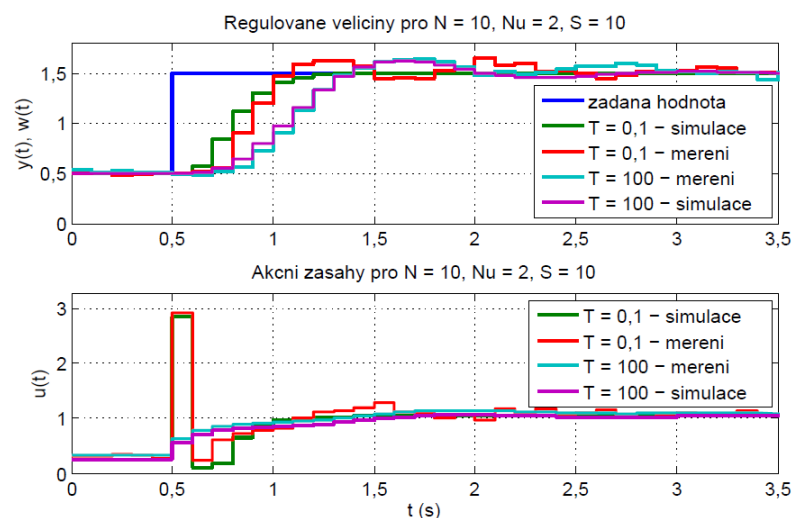
Obr. 3.21 – Simulace pro změny horizontu predikce (N)



Obr. 3.22 – Porovnání simulace a měření pro změnu horizontu predikce



Obr. 3.23 – Simulace pro změny váhy přírůstků akční veličiny (T)



Obr. 3.24 – Porovnání simulace a měření pro změnu parametru T

Tab. 3 – Změny parametrů a jejich důsledky

Parametr	Důsledek změny (zvyšování hodnoty parametru)
Horizont predikce (N)	<ul style="list-style-type: none"> • Zpomaluje se regulační pochod • Akční zásahy se snižují
Horizont řízení (Nu)	<ul style="list-style-type: none"> • Změna parametru nemá výrazný vliv na regulované ani akční veličiny • Počet vygenerovaných regionů se zvyšuje • Prodlužování doby výpočtu
Váha regulační odchylky (S)	<ul style="list-style-type: none"> • Zrychluje se regulační pochod • Zvyšují se akční zásahy
Váha přírůstků akční veličiny (T)	<ul style="list-style-type: none"> • Zpomaluje se regulační pochod • Snižují se akční zásahy

3.5.5 Porovnání s PID, vliv poruchy

Způsob řízení pomocí lineárního explicitního prediktivního regulátoru zajišťuje kvalitní a optimální regulaci. Nicméně bez porovnání s jiným způsobem regulace nelze vhodně posoudit, zda takové předpoklady odpovídají realitě. Z tohoto důvodu jsem si vybral pro srovnání regulace nejpoužívanější regulátory typu PID. Tento regulátor jsem optimálně nastavil pomocí skriptu, který využívá funkci *fminsearch*. Tato funkce minimalizuje v tomto případě kvadratické kritérium (12), které obsahuje dvě složky. V kritériu vystupuje regulační odchylka v druhé mocnině a vážená derivace akční

veličiny v druhé mocnině. Výsledkem minimalizace jsou hledané parametry PID regulátoru (tab. 4).

$$J(P, I, D, N) = \int_0^{\infty} \left\{ [e(t) - e(\infty)]^2 + \lambda \left[\frac{de(t)}{dt} \right]^2 \right\} dt \quad (12)$$

Tab. 4 – Hodnoty parametrů PID

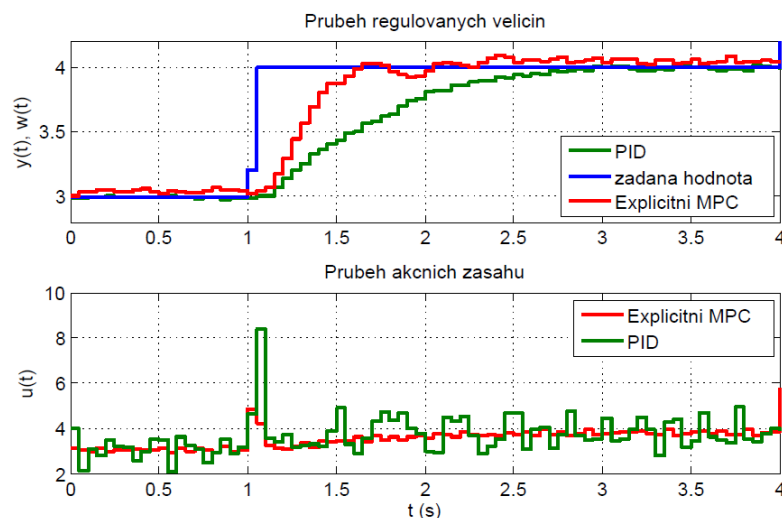
Parametr regulátoru	Hodnota parametru
P	0,1147
I	1,1451
D	0,0901
N	100

Pro praktické ověření nalezených parametrů PID regulátoru jsem využil dostupný blok PID v prostředí Simulinku. Tento blok umožňuje jak sériovou, tak paralelní strukturu PID regulátoru, kterou jsem také zvolil. Používaná přenosová funkce v bloku PID je v podobě s filtrovanou derivací (13).

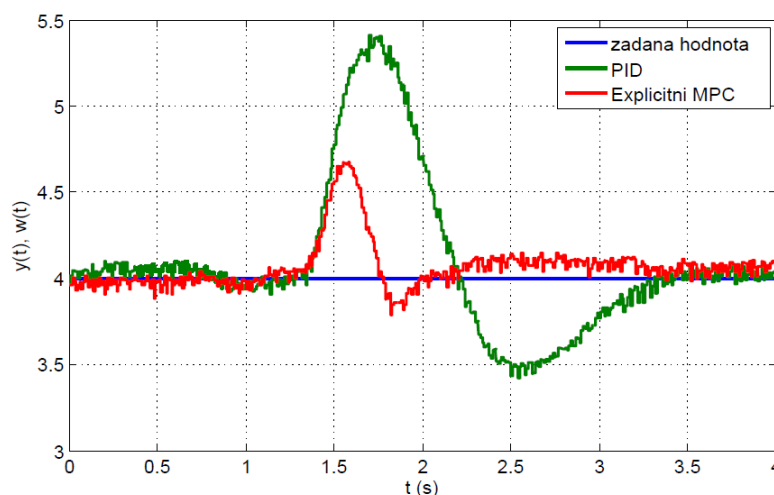
$$R(s) = P + I \left(\frac{1}{s} \right) + D \left(\frac{Ns}{s + N} \right) \quad (13)$$

Následně jsem provedl měření odezvy regulovaných veličin a akčních zásahů pro PID a pro explicitní MPC regulátory. Měření jsem prováděl se vzorkovací periodou $T_s = 0,01$ s. Naměřená data jsem upravil průměrovacím filtrem, protože zvolená vzorkovací perioda vytvořila záznam dat s vysokou hustotou a srovnání by nebylo kvalitní. Upravené odezvy systému pro oba regulátory jsem porovnal (obr. 3.25), kde z tohoto srovnání lze vyhodnotit, že explicitní lineární regulátor je skutečně lepší způsob řízení reálné soustavy, minimálně tedy použitého modelu motoru. Regulační pochod je u PID regulátoru pomalejší a zároveň akční zásahy jsou naopak u explicitního MPC regulátoru optimálnější. V principu také respektují i omezující podmínky, které PID regulátor nezahrnuje. Z tohoto pohledu je explicitní MPC efektivnější přístup regulace. Porovnání, které jsem ještě realizoval, bylo ve formě reakce regulátorů na působení poruchy. Při srovnání obou typů regulace (obr. 3.26) je z grafu patrné, že explicitní prediktivní regulátor si s tímto problémem poradí lépe než PID regulátor.

Uvedený graf znázorňuje reakci regulátorů po odeznění poruchy. PID regulátor zareaguje mnohem větším překmitem než explicitní prediktivní regulátor.

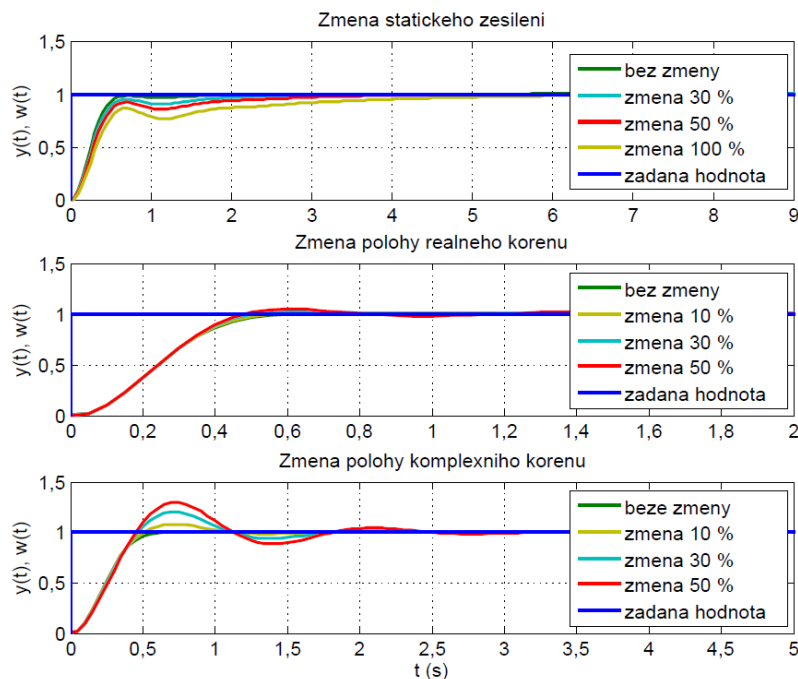


Obr. 3.25 – Porovnání PID vs explicitní MPC

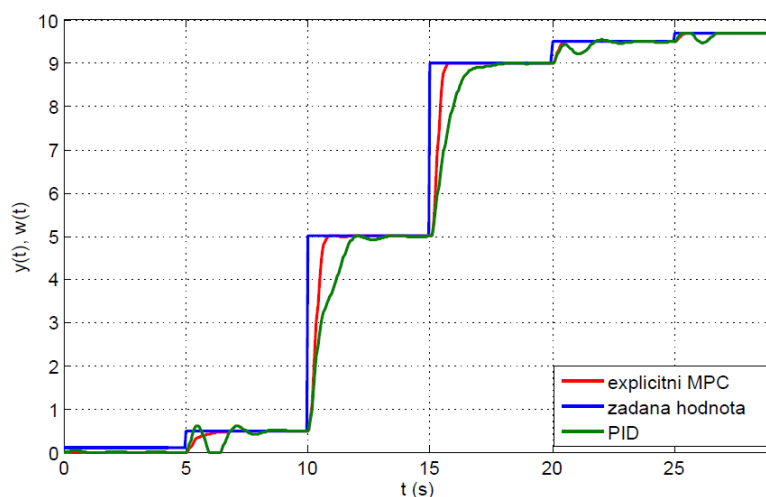


Obr. 3.26 – Průběh regulovaných veličin při reakci na poruchu

V rámci simulace jsem provedl sérii experimentů, kde jsem sledoval, jak kvalitní jsou regulační pochody, pokud model soustavy nebude zcela přesný. Při simulaci jsem uvažoval rozšířený model soustavy a vzorkovací periodu $T_s = 0,05$ s. Ze simulací (obr. 3.27) je zřejmé, že při změně statického zesílení se regulační pochody zhoršují. Naopak při změně polohy reálného pólu nedochází k výraznému zhoršení odezvy. Při změně polohy komplexního pólu jsou regulační pochody kvalitnější než při změně statického zesílení, ale začínají se objevovat kmity. Dále jsem ve formě simulace ověřil chování regulátorů v blízkosti limitů rozsahu, kterých může žádaná hodnota dosáhnout. Ze srovnání je zřejmé (obr. 3.28), že PID regulátor má v těchto oblastech problémy s regulačními pochody oproti explicitnímu lineárnímu regulátoru, který dokáže i v daných oblastech zajistit přijatelný regulační pochod.



Obr. 3.27 – Robustnosť explicitného MPC

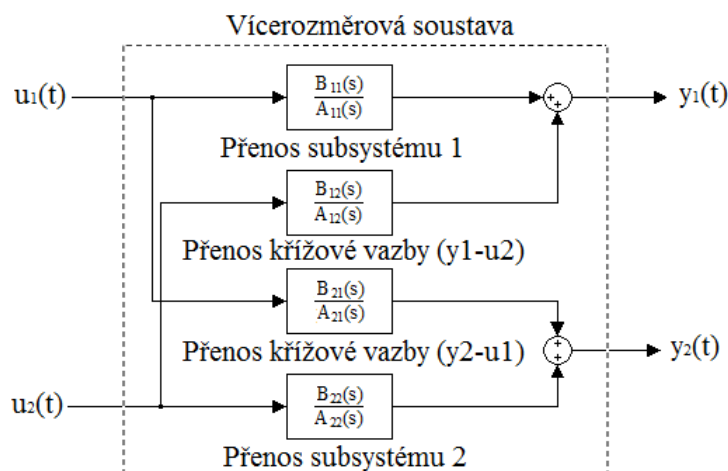


Obr. 3.28 – Srovnání limitů PID vs explicitní MPC

3.6 Model vrtulníku – popis soustavy

Druhá reálná soustava, kde jsem vyzkoušel aplikovat explicitní MPC, je model vrtulníku CE-150 (obr. 3.30) distribuovaný firmou HUMUSOFT. Jedná se o MIMO dynamický systém a je tvořený dvěma vstupy a dvěma výstupy. V ideálním případě by chování MIMO systému odpovídalo tomu, že jeden vstup ovlivní pouze příslušný výstup. Reálné chování však odpovídá obecné struktuře (obr. 3.29), kde přímé vazby jsou doplněny křížovými vazbami. Chování systému je křížovými vazbami obvykle výrazně ovlivněno. Křížové vazby způsobují, že změna jediné vstupní veličiny ovlivní více výstupních veličin. Proto jsou obecně MIMO systémy komplikovanější z hlediska

návrhu a implementace řízení. V článku [16] autor podrobněji představuje strukturu používaného reálného modelu, kde tuto strukturu jsem umístil do přílohy (příloha 1).



Obr. 3.29 – Obecná struktura reálné vícerozměrové soustavy

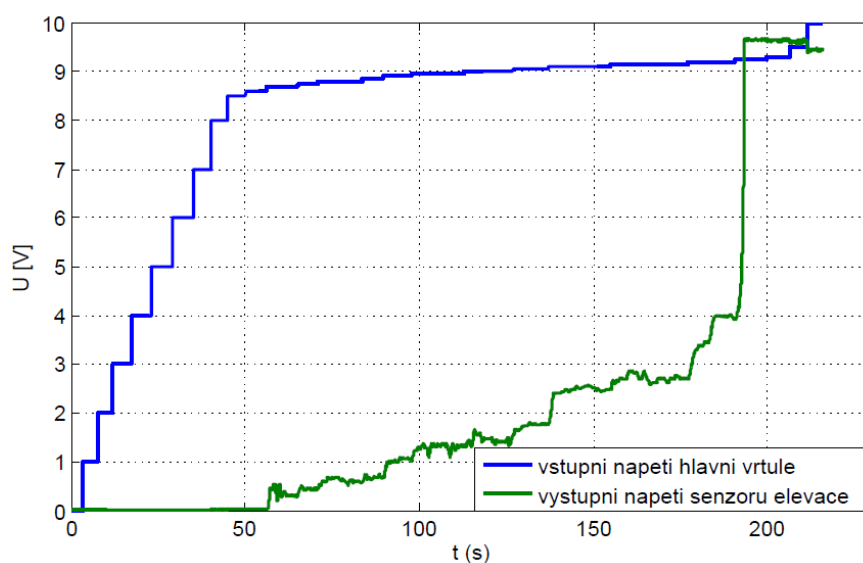
Model vrtulníku je složen ze dvou vrtulí, které jsou řízené stejnosměrnými motory, z podstavce a samotného těla vrtulníku. Uvnitř těla vrtulníku se nachází ještě dodatečná zátěž, která se může posouvat a měnit tak těžiště vrtulníku. Tuto zátěž jsem při svých pokusech nevyužíval, zátěž byla stále ve stejné poloze. Model má dva stupně volnosti. Vrtulník se otáčí kolem horizontální osy v rozsahu 90° a kolem vertikální osy v rozsahu 270° . Pohyb vrtulníku zajišťuje hlavní motor s vrtulí a vedlejší motor s vrtulí. První subsystém je tvořen hlavním motorem, který je ovládaný vstupním napětím z měřicí karty v rozsahu 0–10 V. Příslušný výstup prvního subsystému je reprezentován elevací vrtulníku (náklon nahoru/dolu) ve vertikální rovině. Náklon vrtulníku je měřený senzorem elevace, který generuje hodnoty napětí v rozsahu 0–10 V. Druhý subsystém je tvořen vedlejším motorem, který je stejně jako hlavní motor řízený vstupním napětím z měřicí karty v rozsahu 0–10 V. Vedlejší motor je specifický tím, že při vstupním napětí 5 V je v klidové poloze. Při zvyšování napětí dochází k pohybu v jednom směru otáčení a naopak při snižování napětí pod úroveň 5 V se zadní motor pohybuje v opačném směru. Výstup druhého subsystému je reprezentován azimutem vrtulníku (natočení těla vrtulníku) v horizontální rovině. Tento výstup je měřený senzorem polohy, který generuje napětí v rozsahu 0–10 V.



Obr. 3.30 – Reálný model vrtulníku

3.6.1 Statické a dynamické vlastnosti

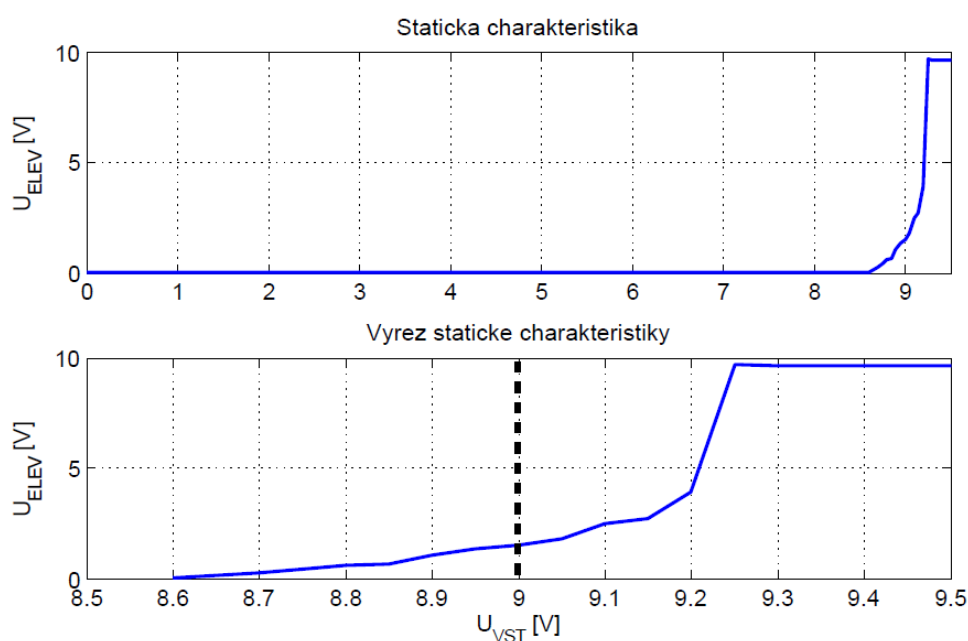
Stejně jako u předchozího systému i zde u modelu vrtulníku bylo nejdříve nutné provést identifikaci chování dané soustavy. Identifikoval jsem statické vlastnosti prvního subsystému tvořeného hlavním motorem s vrtulí. Měření (obr. 3.31) jsem prováděl stejným postupem jako u předchozí reálné soustavy. Prováděl jsem změny příslušné vstupní veličiny v rozsahu $0\text{--}10\text{ V}$ s krokem 1 V . Výsledkem měření byla statická charakteristika (obr. 3.32).



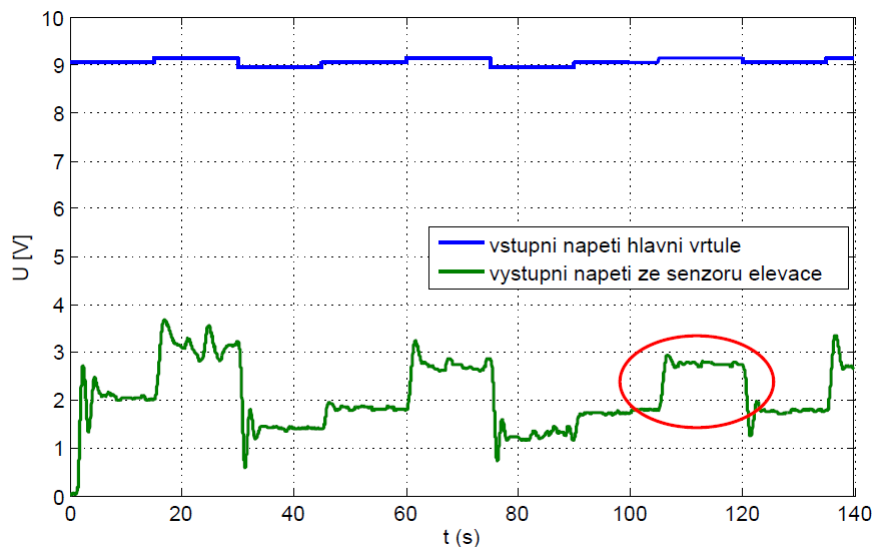
Obr. 3.31 – Měření statické charakteristiky hlavní vrtule

Statická charakteristika u tohoto systému je silně nelineární, nicméně se v ní najde určitá oblast, kterou jsem využil pro řízení. Hlavní motor s vrtulí ani při vstupním napětí 8 V nedokáže zvednout vrtulník do vzduchu, protože ani takové napětí nedokáže vyvolat takový točivý moment hlavní vrtule, aby překonal tíhovou sílu. Ve výřezu statické charakteristiky (spodní část obr. 3.32) je vidět, že dostatečný točivý moment vzniká až při vstupním napětí cca 8,6 V. Rozmezí, ve kterém se vrtulník relativně ustálí, trvá do hodnoty vstupního napětí 9,2 V. Při vyšších hodnotách napětí dochází k maximálnímu náklonu vrtulníku. Proto jsem zvolil pro řízení vhodný pracovní bod v 9 V, který je v relativně lineární oblasti.

Dále jsem identifikoval dynamické vlastnosti hlavního a vedlejšího motoru. Identifikaci hlavního motoru jsem provedl tak, že v okolí zvoleného pracovního bodu jsem prováděl malé skoky budícího signálu (obr. 3.33). Naměřené přechodové odezvy jsem identifikoval pomocí dříve zmíněného identifikačního nástroje *Identification Toolbox*.



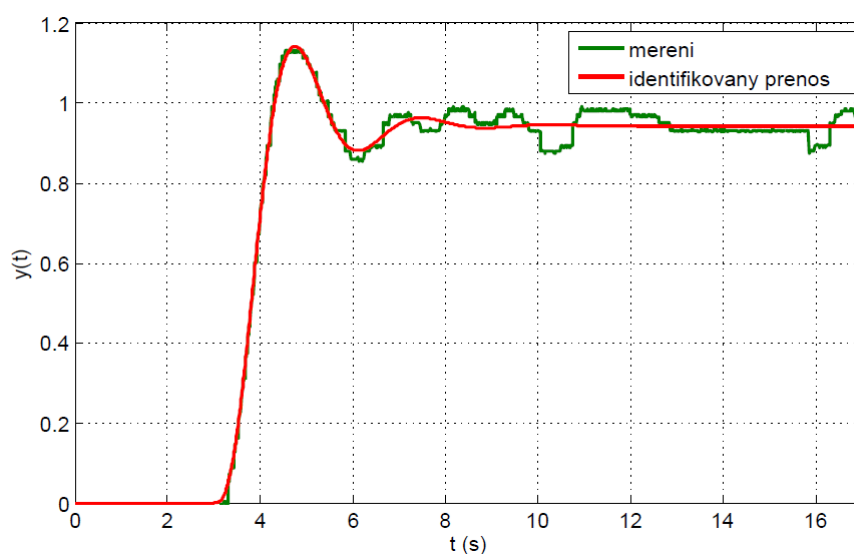
Obr. 3.32 – Identifikace statických vlastností hlavní vrtule



Obr. 3.33 – Měření dynamiky hlavní vrtule

Naměřená data jsem upravil do podoby, kterou jsem mohl identifikovat. Z naměřené odezvy systému je zřejmé, že relativně vhodná přechodová funkce pro identifikaci je ta, která je vyznačená v grafu (obr. 3.33). Nicméně jsem identifikoval několik vybraných přechodových odezev, kde jejich průběh nebyl příliš znehodnocený kmity. Výsledkem identifikace jsem získal několik přenosů, kde nejpřesněji aproximoval dynamické chování přenos 3. řádu (14). Přesnost identifikovaného přenosu dosahovala 93,53 % ve srovnání s naměřeným průběhem (obr. 3.34).

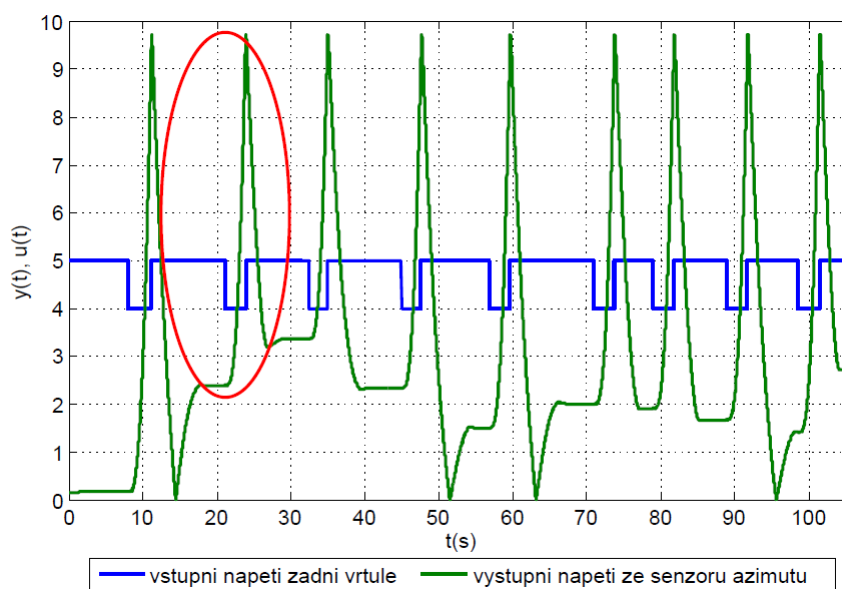
$$F_1(s) = \frac{9,44}{0,05717s^3 + 0,2646s^2 + 0,6285s + 1} \quad (14)$$



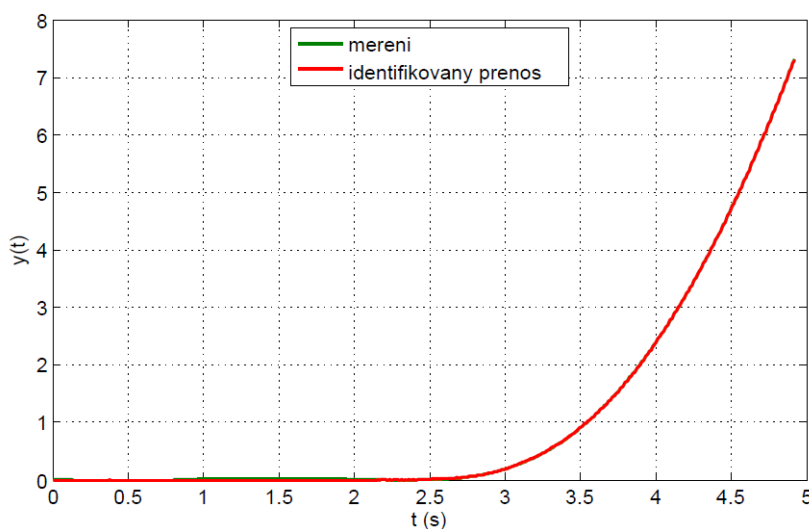
Obr. 3.34 – Porovnání měření a identifikovaného přenosu hlavní vrtule

Druhý subsystém, kterým je vedlejší motor, je soustava astatická, a proto nelze naměřit statické vlastnosti, ale pouze dynamické vlastnosti vyjadřující astatismus. Pro potřeby identifikace jsem provedl sérii měření, kterými jsem získal požadované odezvy (obr. 3.35). Pro účely identifikace jsem vybral několik prvních přechodů z důvodu srovnání, který z nalezených přenosů co nejpřesněji aproximuje astatické chování systému. V grafu (obr. 3.35) jsem vyznačil ten průběh, který požadavku vyhověl s přesností 99,71 %. Pomocí identifikačního nástroje jsem zjistil přenos 3. řádu (15), který dokonale aproximuje naměřenou reálnou odezvu (obr. 3.36).

$$F_2(s) = \frac{-50,24}{s(15,53s^2 + 10,26s + 1)} \quad (15)$$



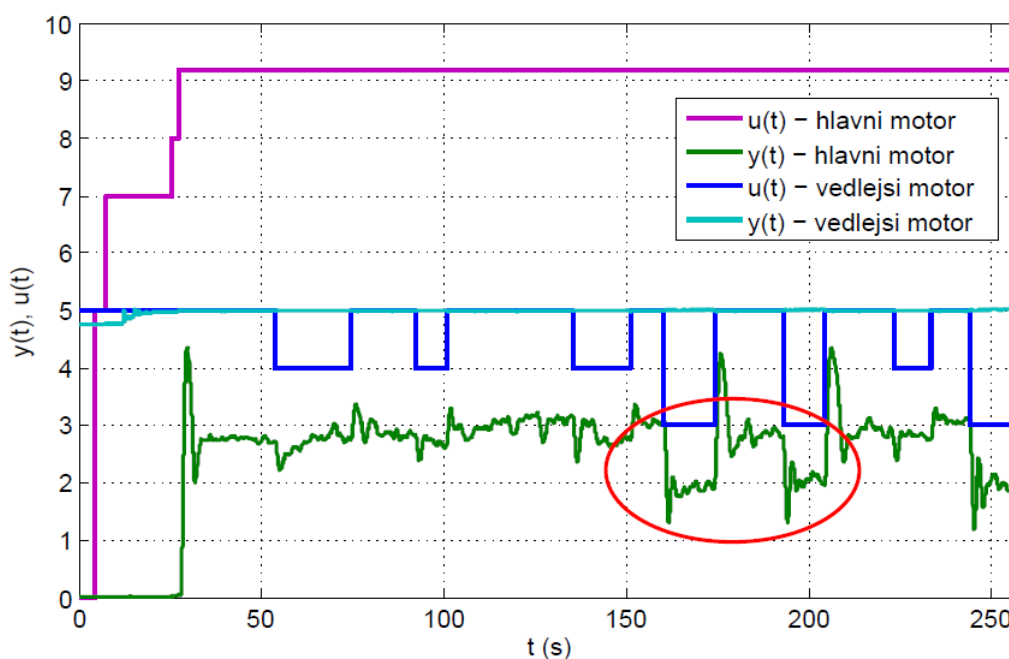
Obr. 3.35 – Měření astatických vlastností zadní vrtule



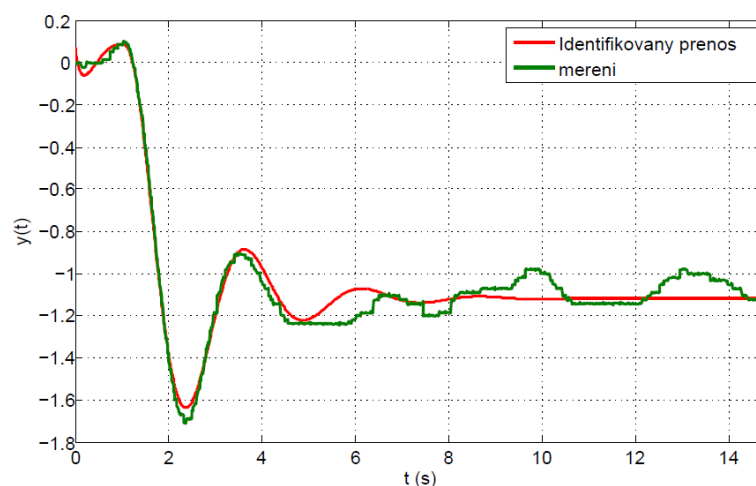
Obr. 3.36 - Porovnání měření a identifikovaného přenosu zadní vrtule

Poslední přenosy, které bylo nutné identifikovat, souvisely s křížovými vazbami. Křížovou vazbu mezi vedlejším motorem a elevací vrtulníku bych v principu mohl zanedbat. Logicky vedlejší motor by neměl mít výrazný vliv na elevaci vrtulníku oproti křížové vazbě mezi hlavním motorem a azimutem vrtulníku. Při měření (obr. 3.37) jsem zjistil, že malé změny vstupního napětí (do 1 V) pro vedlejší motor nemají opravdu velký vliv na elevaci. Naopak větší změny, v podobě skoku vstupního napětí o 2 V a více, ovlivňují elevaci mnohem výrazněji a vliv této křížové vazby nelze zcela zanedbat. Měření příslušné křížové vazby jsem realizoval tak, že subsystém s hlavním motorem jsem nastavil na zvolený pracovní bod v 9,2 V a tato hodnota byla po celou dobu měření konstantní. Následně jsem prováděl malé skoky vstupního napětí na vedlejším motoru. Pro identifikaci jsem vybral přechodové funkce vyznačené v grafu (obr. 3.37), kde vstupní veličina je vstupní napětí vedlejšího motoru a výstupní veličina je elevace vrtulníku. Výsledkem byl získaný přenos 3. řádu (16), který v rámci tolerance dostatečně přesně (83,25 %) aproximuje naměřenou přechodovou odezvu (obr. 3.38).

$$F_{12}(s) = \frac{0,559}{0,016s^3 + 0,1706s^2 + 0,2986s + 1} \quad (16)$$



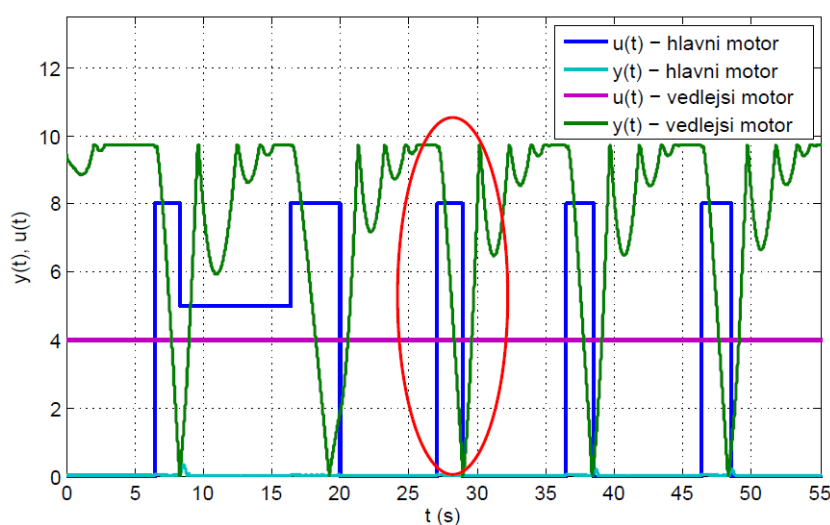
Obr. 3.37 – Měření křížové vazby zadní motor-elevace



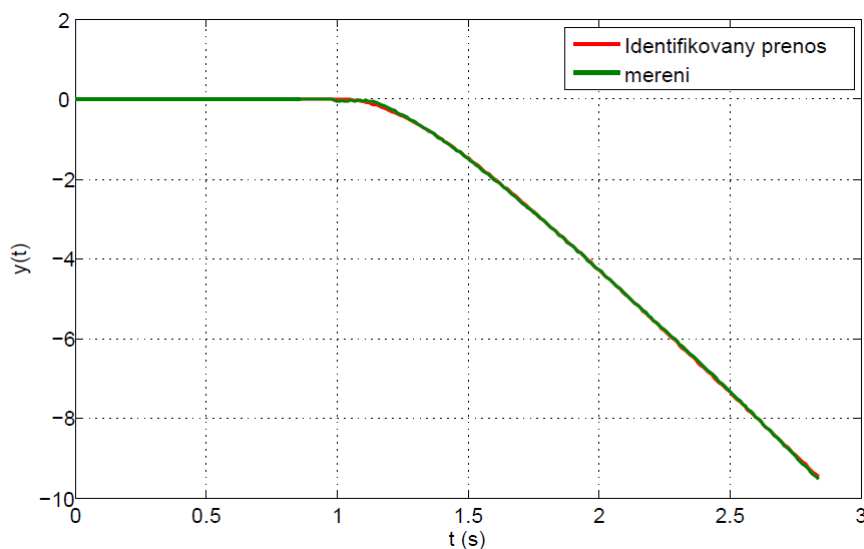
Obr. 3.38 – Porovnání identifikace a měření křížové vazby zadní motor-elevace

Druhá křížová vazba mezi hlavním motorem a azimutem vrtulníku je velmi výrazná. Měření (obr. 3.39) této křížové vazby jsem provedl tak, že jsem prováděl malé skoky vstupního napětí na hlavním motoru. Vedlejší motor jsem nastavil na konstantní hodnotu 4 V. Při napětí < 5V se vedlejší motor pohybuje ve směru, ve kterém vytváří opačný točivý moment než hlavní motor. V principu jiná možnost nastavení vedlejšího motoru není, z hlediska směru otáčení, protože hlavní motor se pohybuje pouze v jednom směru. Z naměřených dat jsem vybral několik přechodových funkcí, které jsem identifikoval. Největší přesnosti dosahoval identifikovaný přenos 3. řádu (17) pro vyznačenou přechodovou funkci (obr. 3.39), který s dostatečnou přesností 99,3 % (obr. 3.40) aproximuje chování systému.

$$F_{21}(s) = \frac{-0,785}{s(0,01273s^2 + 0,3511s + 1)} \quad (17)$$



Obr. 3.39 – Měření křížové vazby hlavní motor-azimut



Obr. 3.40 – Porovnání identifikace a měření křížové vazby hlavní motor-azimut

3.6.2 Praktická realizace explicitního MPC

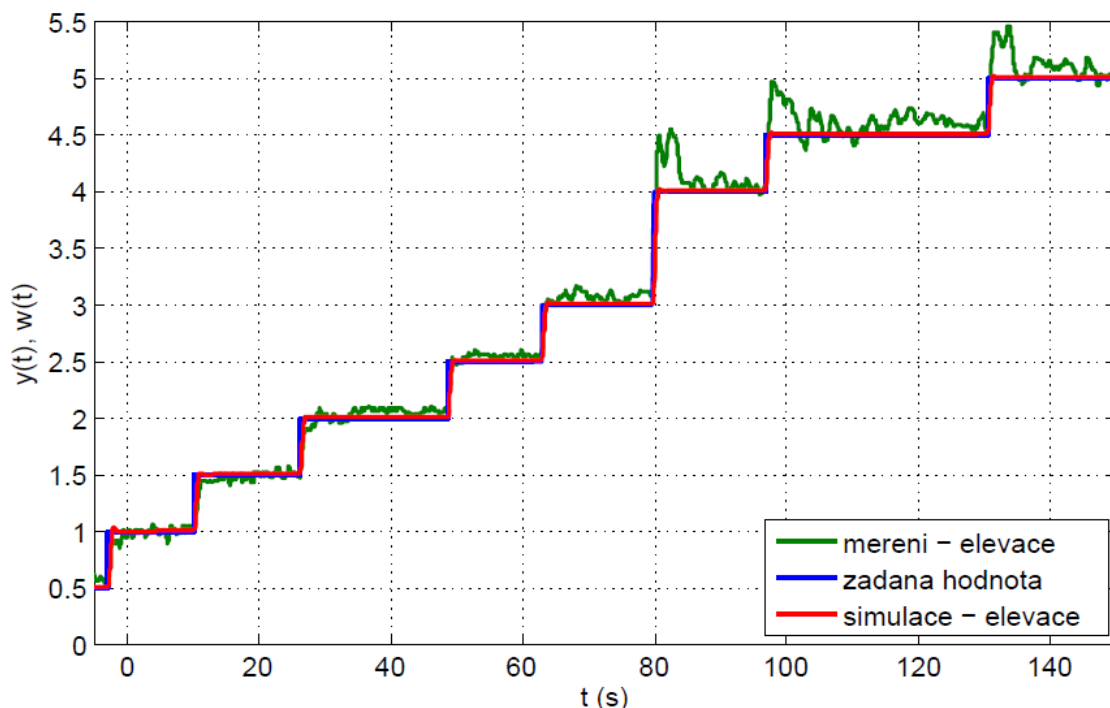
Všechny požadované modely ve formě přenosových funkcí jsem zjistil pomocí identifikačního nástroje a provedl jsem následně návrh explicitního lineárního regulátoru. Všechny důležité parametry, které jsem používal při návrhu regulátorů pro dvojici subsystémů, jsou uvedeny v následující tabulce (tab. 5).

Tab. 5 – Přehled parametrů pro návrh řízení modelu vrtulníku

Parametr	Subsystém hlavní motor - elevace	Subsystém vedlejší motor- azimut
Vzorkovací perioda (T_s)	0,03 s	0,02 s
Váha regulační odchylky (S)	10	10
Váha přírůstků akční veličiny (T)	0,1	0,1
Horizont řízení (N_U)	4	2
Horizont predikce (N)	60	80
Vygenerovaný počet regionů po návrhu	67	7

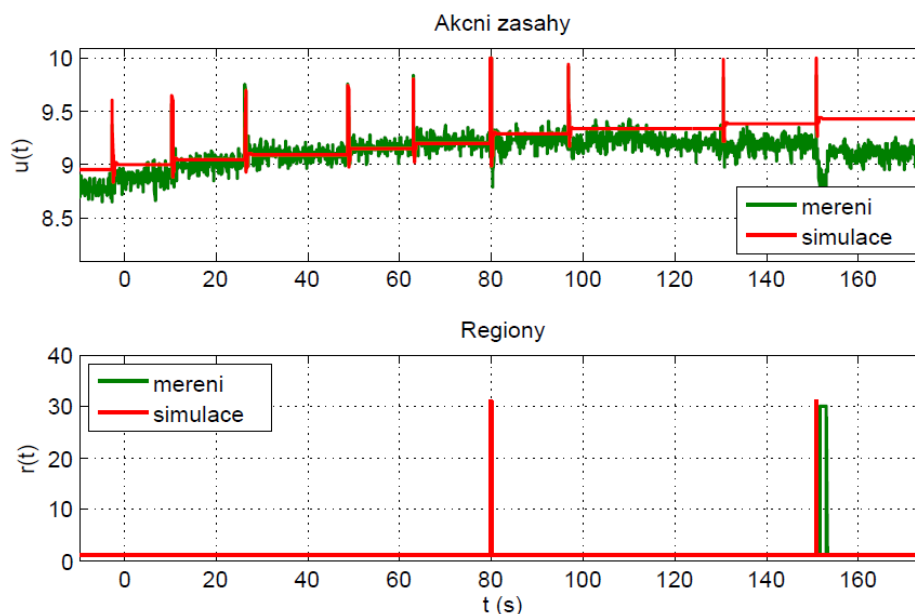
Navržené regulátory jsem aplikoval na příslušné subsystémy a naměřil jsem regulační pochody pro každý subsystém samostatně. Srovnání simulovaného a naměřeného výstupu pro subsystém s hlavním motorem je znázorněné na následujícím grafu (obr. 3.41). Regulaci jsem vyzkoušel především v okolí pracovního bodu, kde jsem prováděl identifikaci, ale také v úseku, která je mimo zvolenou pracovní oblast. Z porovnání je jednoznačné, že v okolí pracovního bodu, kde se provedla linearizace, jsou regulační pochody dostatečně kvalitní a splňují požadavky na sledování žádané

hodnoty. Regulovaná veličina ze simulace se téměř dokonale shoduje s naměřeným průběhem. Naopak mimo zvolenou pracovní oblast se začínají projevovat nelinearity systému. Přestože v této oblasti jsou regulační pochody zatíženy nelinearitami, tak v rámci možností je regulace přijatelná.



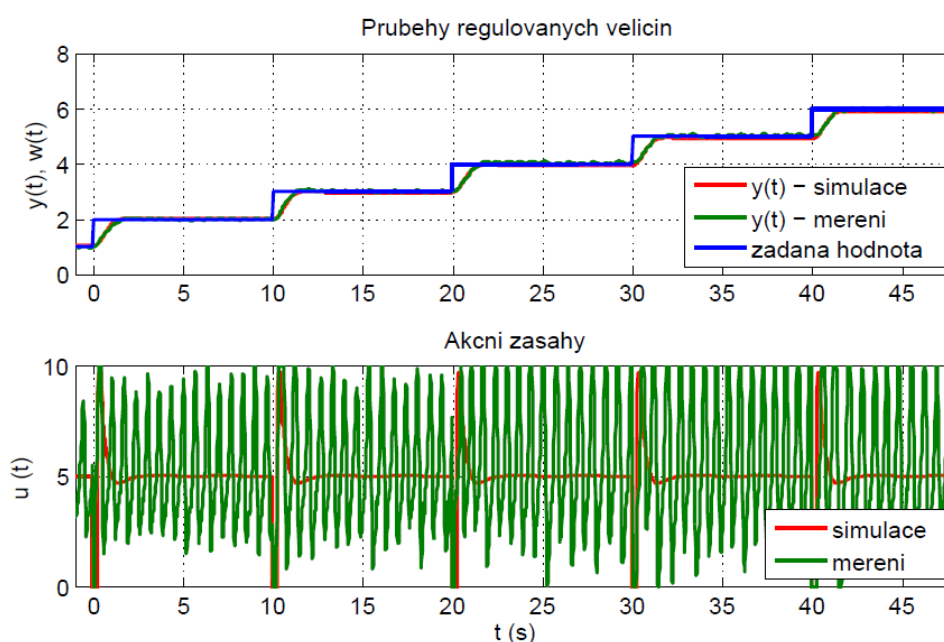
Obr. 3.41 – Porovnání regulovaných veličin při řízení elevace

Výsledkem porovnání simulací a měření byly i akční zásahy a proces výběru regionů během regulace (obr. 3.42). Z grafu, který srovnává akční zásahy, je viditelný stejný důsledek nelinearit. V okolí pracovního bodu, který jsem zvolil přibližně v 9,2 V pro vstupní napětí, je zřejmé, že simulované akční zásahy v rámci určité tolerance odpovídají naměřenému průběhu. Jakmile se provádí srovnání mimo tuto pracovní oblast, rozdíl mezi simulovanými a naměřenými akčními zásahy se zvyšuje vlivem nelinearit. Průběhy výběru regionů (spodní část obr. 3.42) z velké části odpovídají, pouze v jednom úseku se mírně odlišují. Je to způsobeno tím, jak zrovna při reálném řízení interpretuje stavový pozorovatel aktuální stav systému. Při regulaci tohoto subsystému s hlavním motorem jsem musel provést rozšíření modelu soustavy. Vznikl zde stejný problém jako při regulaci modelu motoru, kde regulátor reagoval s trvalou regulační odchylkou.



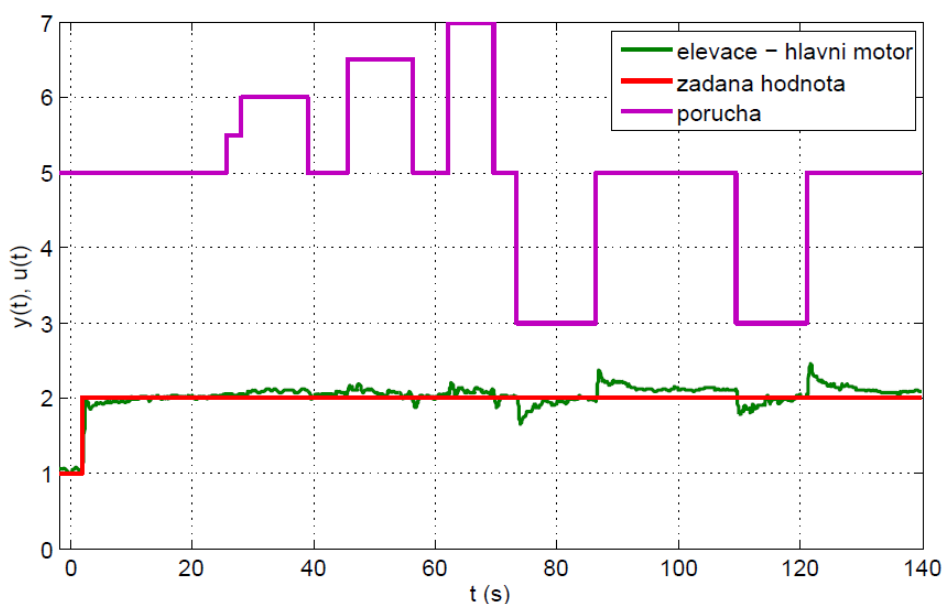
Obr. 3.42 – Porovnání akčních zásahů a regionů při řízení elevace

Druhý typ navrženého regulátoru jsem aplikoval na druhý subsystém, kde jsem řídil azimut vrtulníku. Při regulaci tohoto subsystému jsem nemusel rozšiřovat model soustavy. Přestože problém trvalé regulační odchylky zde vznikl také, řešení bylo mírně odlišné. Především je nutné zdůraznit, že významný vliv na trvalou regulační odchylku mají zvolené parametry pro návrh regulátoru a zejména vzorkovací perioda. Příslušné řešení, kromě vhodně zvolených parametrů, spočívá v zavedení pomocného signálu k akční veličině. Výsledkem jsou regulační pochody, které jsem porovnal v následujícím grafu (obr. 3.43). Z grafu je zřejmé, že naměřená regulovaná veličina přesně sleduje referenční trajektorii.

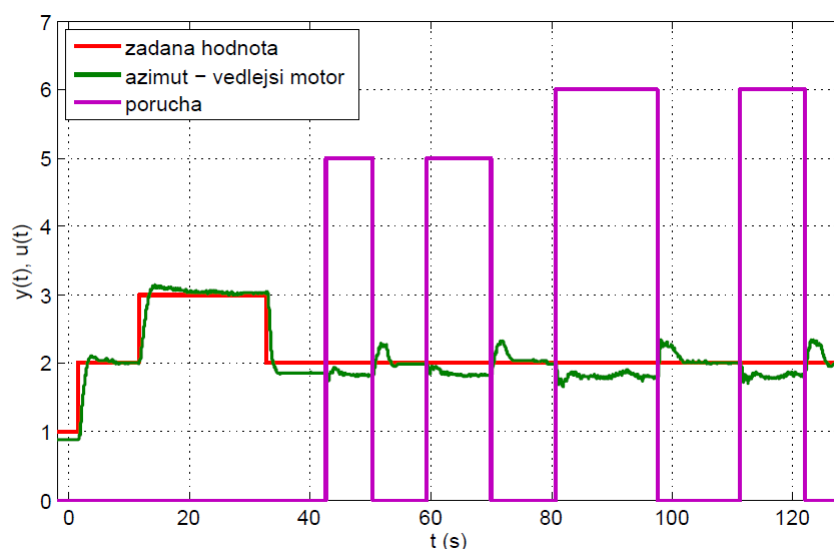


Obr. 3.43 – Porovnání průběhů při řízení azimutu

V rámci dalších měření jsem vyzkoušel, jaký vliv bude mít při regulaci porucha a jaká bude odpovídající reakce regulátoru jak pro elevaci, tak pro azimut. Poruchu jsem zavedl v principu od druhého subsystému. Pro subsystém s hlavním motorem jsem nastavil hodnotu pracovního bodu a prováděl jsem skoky vstupního napětí vedlejšího motoru, který tak představoval působení poruchy. Z grafu (obr. 3.44) je zřejmé, že navržený regulátor potlačí poruchu vcelku kvalitně. Regulovaná veličina neustále sleduje žádanou hodnotu a pouze v případě začátku a konce působení poruchy se zvýší regulační odchylka. Regulátor se na tuto změnu snaží okamžitě reagovat a regulační odchylku snižuje. V případě zavedení poruchy pro druhý subsystém jsem nastavil konstantní hodnotu pro azimut. V tomto případě jsem prováděl skoky vstupního napětí na hlavním motoru, který představoval působící poruchu. Z naměřených průběhů (obr. 3.45) je patrné, že po odeznění poruchy regulátor zajistí zpětné dosažení žádané hodnoty, nicméně při působení poruchy regulátor nedokáže odstranit trvalou regulační odchylku. Zřejmě je to způsobené tím, že model soustavy není rozšířený o dodatečný stav, ale řešení problému s trvalou regulační odchylkou je vyřešeno vhodnými parametry a zavedeným pomocným signálem. Naopak subsystému s hlavním motorem pracuje s rozšířeným modelem soustavy, a proto dokáže odstranit trvalou regulační odchylku i během působení poruchy.

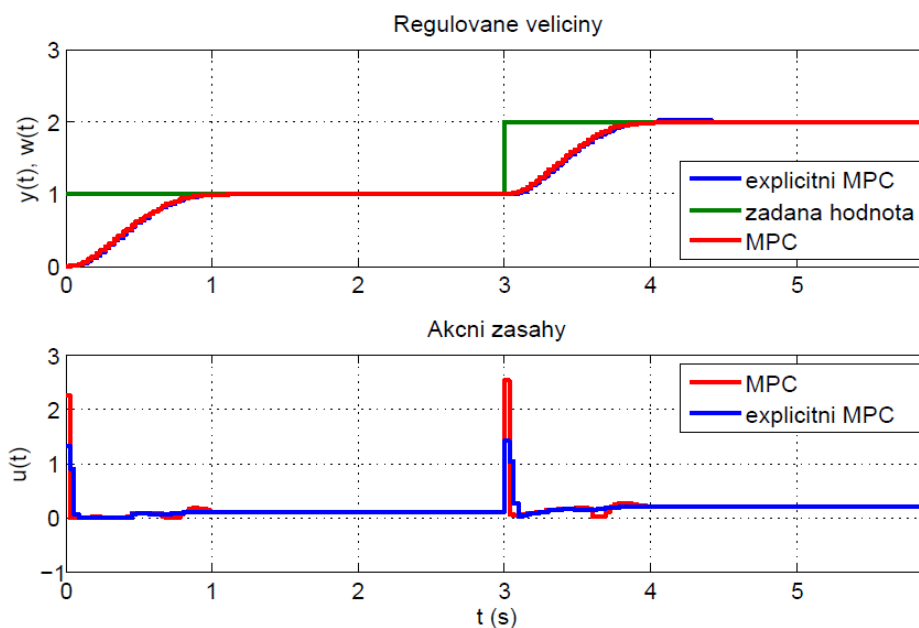


Obr. 3.44 – Vliv poruchy na elevaci



Obr. 3.45 – Vliv poruchy na azimut

V rámci simulace jsem porovnal odezvy regulačních pochodů standardního prediktivního regulátoru a explicitního lineárního prediktivního regulátoru. Zvolený model soustavy v rámci simulace byl identifikovaný model subsystému hlavní motor s vrtulí – elevace. Pro jednoduchost jsem předpokládal lineární model, neuvažoval jsem tak nelinearitu typu necitlivosti. Simulované odezvy jsem porovnal (obr. 3.46) a z výsledků je zřejmé, že standardní MPC regulátor je o něco rychlejší než lineární explicitní MPC regulátor, ale rozdíl regulovaných veličin je minimální. Naopak rozdíl akčních zásahů je výraznější, kde explicitní MPC reaguje optimálními akčními zásahy oproti standardnímu MPC. Rozdíly jsou především výrazné při změně žádané hodnoty.



Obr. 3.46 – Simulace MPC vs explicitní MPC

3.7 Implementace do reálného zařízení

V rámci prostředí Matlab a pomocí dodatečného toolboxu jsem úspěšně aplikoval explicitní MPC regulátor. Všechny návrhy a realizace regulátoru jsou však spojeny s nutností používat příslušné nástroje (Matlab, Hybrid Toolbox). Z tohoto důvodu je logická otázka, zda existuje způsob, jak realizovat explicitní MPC bez nutnosti příslušných nástrojů, resp. jak realizovat explicitní MPC bez použití PC. Vcelku zajímavé řešení nabízí samotný Hybrid Toolbox.

Autor na své stránce [13] a v manuálu [17] uvádí, že Hybrid Toolbox je schopný vygenerovat programový kód v jazyce C, který bude představovat explicitní lineární regulátor. Tento regulátor se poté používá při simulacích a praktických měření. Jádro programového kódu je soubor `expcon.c`, který je součástí toolboxu a vypočítá optimální akční zásahy spojené s vygenerovanými regiony. Zároveň `expcon.c` zajišťuje zpětnou vazbu v podobě odhadu stavového vektoru při regulaci. Tato funkce potřebuje pro svoji správnou funkci hlavičkový soubor `expcon.h`. V tomto souboru jsou obsaženy všechny parametry definující explicitní lineární regulátor. Hlavičkový soubor se vždy aktualizuje po simulaci v prostředí Simulinku. V principu tak soubory `expcon.c` a `expcon.h` představují zákon řízení v podobě implementace vyhledávací tabulky.

Závěr

Výsledkem mojí diplomové práce je splnění všech bodů zadání. Nejdříve jsem získal teoretické znalosti o dané problematice prediktivního řízení založeného na modelu a především o explicitním MPC, které bylo hlavním cílem této diplomové práce. Explicitní MPC má mnohem větší možnosti uplatnění v praxi než standardní prediktivní řízení založené na modelu. Především je to způsobené výpočetními nároky, které explicitní MPC výrazně snižuje a umožňuje tak použít tuto zajímavou metodu řízení i pro systémy, kde standardní MPC není použitelné. Hlavním rozdílem je, že standardní MPC vypočítává optimalizační úlohu neustále a opakovaně oproti explicitnímu MPC, které optimalizační úlohu vypočítá dopředu a pouze jednou a během řízení se rozhoduje, které dopředu nalezené řešení bude aktuálně používáno. Explicitní MPC se uplatňuje např. při řízení trakce, motoru nebo řízení otáček v automobilu atd. Další výhody, které přináší výrazné snížení výpočetní náročnosti, jsou ekonomické úspory, především se výrazně snižuje cena implementace prediktivního řízení.

Pro návrh lineárního explicitního regulátoru jsem využil dostupného nástroje Hybrid Toolbox. Pomocí demo příkladů, které jsou dostupné v rámci toolboxu, jsem se seznámil s programovým prostředím a rozšiřující knihovnou v prostředí Simulinku. Pro demonstraci správné funkce navrženého explicitního lineárního regulátoru jsem využil jeden z demo příkladů. Následně jsem úspěšně provedl návrh a implementaci explicitního MPC řízení na reálné úlohy. První úloha byla model motoru. Regulační pochody ze začátku nebyly zcela kvalitní. Regulovaná veličina nesledovala žádanou hodnotu, resp. regulátor reagoval s trvalou regulační odchylkou. Problém byl způsobený tím, že model motoru obsahuje nelinearitu v podobě necitlivosti při nízkých hodnotách vstupního napětí. Tento problém jsem konzultoval s vedoucím diplomové práce, kde jsme našli řešení v podobě rozšíření stavového popisu soustavy. Provedl jsem návrh regulátoru s rozšířeným modelem soustavy a výsledkem byly regulační pochody, které splňují požadavky. Regulovaná veličina přesně sledovala žádanou hodnotu a trvalá regulační odchylka se dále neprojevila. Pro srovnání regulace s jiným typem regulační struktury jsem provedl návrh PID regulátoru. Z výsledku srovnání byl jednoznačně vhodnější typ řízení pomocí lineárního explicitního regulátoru. Dosažení ustáleného stavu u PID regulátoru trvalo cca 3 s, naopak u lineárního explicitního regulátoru byla doba ustálení cca 0,8 s. Experimentálně jsem ověřil vliv poruchy a reakci

PID vs explicitního MPC regulátoru. Ze srovnání je zřejmé, že explicitní MPC regulátor reaguje na poruchu lépe.

V poslední části jsem úspěšně provedl návrh a implementaci regulátorů pro model vrtulníku v podobě takové, že jsem uvažoval řízení jednorozměrových systémů. Nejdříve jsem provedl identifikaci příslušného systému, která nebyla zcela jednoznačná. Model vrtulníku je silně nelineární systém, to lze vidět ze statické charakteristiky hlavního motoru s vrtulí, a také druhý subsystém tvořený vedlejším motorem je systém astatický. Po úspěšné identifikaci jsem provedl implementaci navržených lineárních explicitních regulátorů a výsledkem byly přijatelné regulační pochody. Zejména při regulaci subsystému hlavní motor-elevace je z měření patrný vliv nelinearity. V oblasti kolem pracovního bodu jsou regulační pochody relativně kvalitní, ale mimo tuto oblast se začínají projevovat nelinearity a odezvy se zhoršují. V principu je to demonstrace limitů používaného lineárního explicitního regulátoru. Tento regulátor je použitelný pro lineární systémy. Možnost, jak odstranit tento problém, vliv nelinearity na regulační pochody, by bylo zavedení explicitního hybridního regulátoru. Ten umožňuje definovat model soustavy mnohem podrobněji a lze uvažovat i tyto nelinearity už při návrh regulátoru. Nicméně tato možnost je námětem pro rozšíření této diplomové práce.

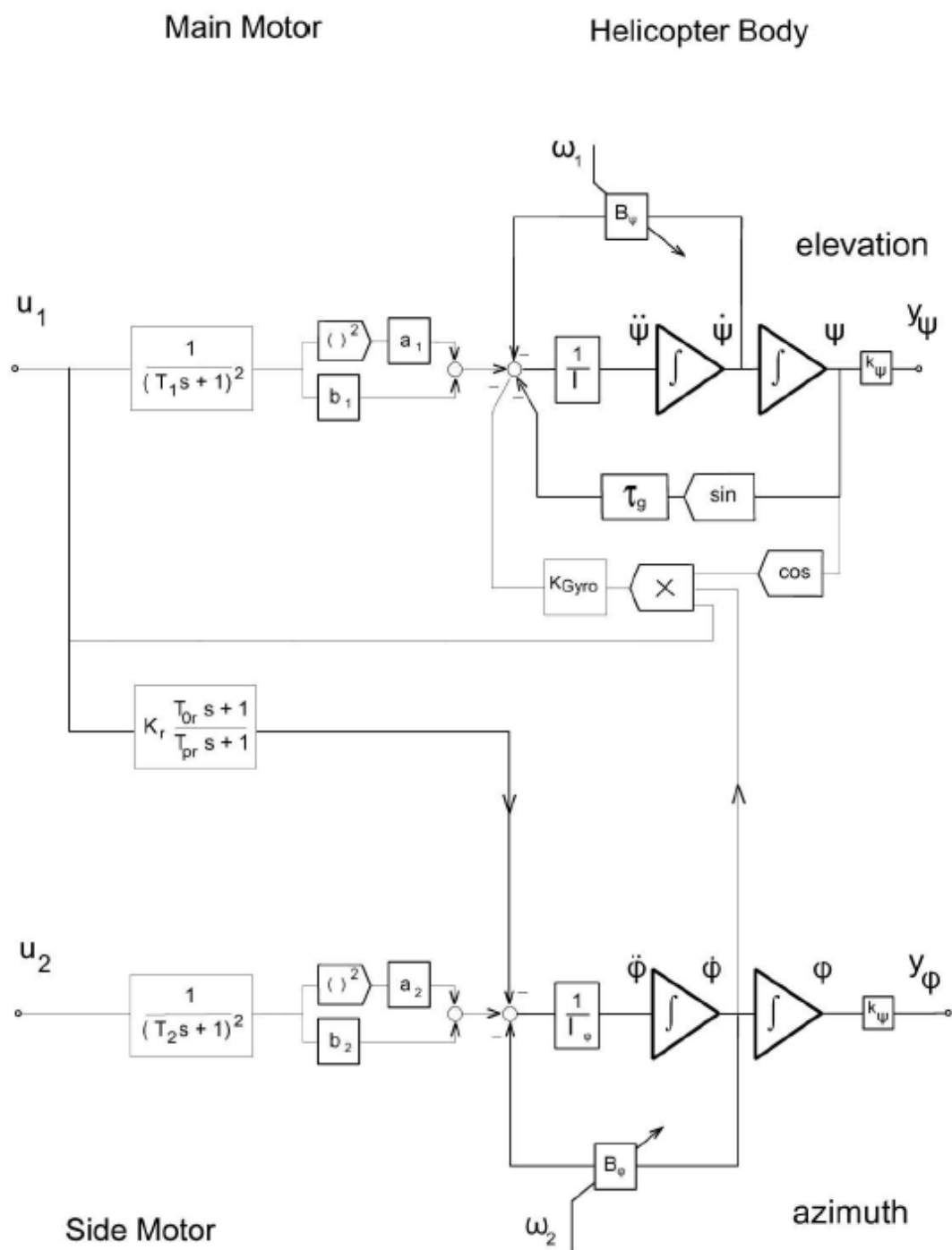
Úspěšně se mi podařilo implementovat navržené regulátory pro jednotlivé subsystémy vícerozměrového modelu, nicméně nepodařilo se mi spojit tyto subsystémy a regulovat soustavu opravdu jako vícerozměrovou. Při několika pokusech návrhu regulátorů a jejich implementace jsem nenašel takové nastavení parametrů pro návrh regulátoru, které by bylo stabilní. Při aplikaci regulátoru navrženého pro vícerozměrový model soustavy, regulační pochody pro samostatné subsystémy byly kvalitní. Jakmile se během regulace jednoho subsystému přidal druhý subsystém, regulační pochody se zhoršily výraznými kmity. Ve většině případů se mi nepodařilo najít správné nastavení, které by umožnilo vytvořit funkční regulátor, protože návrh regulátoru je velmi citlivý na nastavené parametry. Možnost rozšíření regulace na vícerozměrový systém může být dalším námětem pro pokračování této diplomové práce. Případně dalším námětem by mohla být snaha o implementaci navrženého explicitního regulátoru do reálného zařízení a odstranit tak prostředníka, kterým je PC.

Seznam použité literatury

- [1] PEKAŘ, Jaroslav a Jan JECH. *Prediktivní řízení průmyslových procesů*. [online]. [cit. 2014-04-29].
Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=34170
- [2] MATYS, Libor. Prediktivní regulátory s principy umělé inteligence v prostředí MATLAB – B&R [online]. Brno, 2008. [cit. 2014-05-04]. Dostupné z:
https://dspace.vutbr.cz/bitstream/handle/11012/13039/xmatys05_dp.pdf?sequence=1. Diplomová práce. Vysoké učení technické v Brně.
- [3] JAROSLAV, Hlava. *Prediktivní řízení založené na modelu – hlavní principy* [online]. [cit. 2014-05-05]. Dostupné z:
<http://www.fm.tul.cz/esf0247/index.php?download=670>
- [4] MAGNI, Lalo. Nonlinear model predictive control: towards new challenging applications. Berlin: Springer, 2009, s. 345-369. ISBN 978-3-642-01094-1
- [5] BEMPORAD, Alberto, Manfred MORARI, Vivek DUA a Efstratios N. PISTIKOPOULOS. The explicit linear quadratic regulator for constrained systems. [online]. [cit. 2014-05-11]. Dostupné z:
<http://robot2.disp.uniroma2.it/~zack/ftp/users/Tingshu/papers/BemporadExplicitAuto.pdf>
- [6] BEMPORAD, A. Explicit Model Predictive Control via Multiparametric Programming. [online]. [cit. 2014-05-12]. Dostupné z:
http://cse.lab.imtlucca.it/~bemporad/teaching/mpc/stuttgart_2012/3-explicit_linear_mpc.pdf
- [7] BORELLI, F., BAOTIC, M., BEMPORAD, A., MORARI, M. *Efficient On-Line Computation of Explicit Model Predictive Control*. Technical report. August 2001.
- [8] TONDEL, P., JOHANSEN, T.A., BEMPORAD, A. *Evaluation of piecewise affine control via binary search tree*. Automatica 39 (2003), str. 945-950.
- [9] CAGIENARD, R., GRIEDER, P., KERRIGAN, E.C., MORARI, M. *Move Blocking Strategies in Receding Horizon Control*. [online]. [cit. 2014-05-05].
Dostupné z: <http://www2.ee.ic.ac.uk/publications/p5072.pdf>

- [10] MICHAL, Kvasnica. *Rýchle a pamäťovo efektívne prediktívne riadenie hybridných systémov* [online]. Bratislava, 2011 [cit. 2014-05-05]. Dostupné z: <http://www.kirp.chtf.stuba.sk/~fikar/docprof/kvasnica11/kvas-doc-1cast.pdf>. Habilitačná práca. Slovenská technická univerzita v Bratislavě.
- [11] MACIEJOWSKI, Jan Marian. *Predictive control: with constraints*. New York: Prentice Hall, 2002, xviii, 331 p. ISBN 02-013-9823-0.
- [12] ŘEHOŘ, Jiří. Explicitní řešení úlohy prediktivní regulace. Černošice, 2008. Diplomová práce. ČVUT.
- [13] BEMPORAD, A. Hybrid Toolbox for MATLAB. [online]. 2012 [cit. 2013-02-15] Dostupné z: <http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox/>
- [14] What is angle of attack?. [online]. [cit. 2014-05-06]. Dostupné z: http://www.boeing.com/commercial/aeromagazine/aero_12/whatisaoa.pdf
- [15] VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. *Identifikace systémů* [online]. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2012, 1 CD-ROM [cit. 2014-05-12]. ISBN 978-80-248-2594-6. Dostupné z: <http://www.person.vsb.cz/archivcd/FMMI/IS/Identifikace%20systemu.pdf>
- [16] HUSPEKA, J.: Návrh řízení modelu vrtulníku. MATEO - The European Network of Mechatronics Centres and Industrial Controllers 2006, p. 97-105, Západočeská univerzita, Plzeň, 2006.
- [17] Hybrid Toolbox: User's Guide. [online]. [cit. 2014-05-14]. Dostupné z: ftp://ftp.feq.ufu.br/Luis/HYBRID/MIP/hybtbx-win/doc/hybtbx_manual.pdf
- [18] BRÁZDIL, Michal. Výpočet obecného řídicího zákona prediktivního řízení pro systém druhého řádu bez omezení. [online]. [cit. 2014-05-04]. Dostupné z: <http://www.posterus.sk/?p=11147>
- [19] MACHALSKÝ, Ondřej. Prediktivní řízení s měřenou poruchovou veličinou. Zlín, 2011. Diplomová práce. UTB ve Zlíně.
- [20] HOČ, Michal. Helicopter in virtual space. Praha, 2008. Diplomová práce. ČVUT.
- [21] Helicopter Simulator. [online]. [cit. 2014-05-13]. Dostupné z: <http://act.dii.unisi.it/description/heli.pdf>

Příloha 1 – Schéma modelu vrtulníku



Příloha 1 – Zjednodušený model vrtulníku [19]

Příloha 2 – Skript pro návrh explicitního MPC

```
clear variables
clear all; close all; clc;

Ts=0.03;           % Vzorkovací perioda

sys=tf(10.38,[0.05569 0.2508 0.6397 1]); % prenos soustavy 3.rad

%% Navrh linearniho regulatoru

% Rozsireni stavoveho popisu soustavy
sysd=c2d(sys,Ts);
Mod1=ss(sysd);
Aug=[Mod1.a zeros(3,1); 0 0 0 1];
Baug=[Mod1.b;0];
Caug=[Mod1.c 1];
Mod2=ss(Aug,Baug,Caug,0,Ts);

% Predikcni model
model=Mod2;

clear limits
limits.umin=0;     limits.umax=10;

clear cost
cost.S=10;          % Vaha regulacni odchylky
cost.T=0.1;         % Vaha prirustku akcni veliciny
cost.rho=Inf;

clear interval
interval.Nu=4;      % horizont rizeni
interval.N=60;      % horizont predikce

Cimp=lincon(model,'track',cost,interval,limits);

%% Navrh explicitniho linearniho regulatoru
clear range
range.xmax=[1000;1000;1000;1000];
range.xmin=-range.xmax;
range.umax=limits.umax;
range.umin=limits.umin;
range.refymax=[10];
range.refymin=-range.refymax;

clear options
options.verbose=1;
options.reltol=1e-6;
options.uniteeps=1e-3;
options.qpsolver='qpact';

Cexp=expcon(Cimp,range,options);
clear expsfun.mexw32
```

Příloha 3 – Obsah přiloženého CD

- text diplomové práce
 - diplomova_prace_2014_Martin_Kopal.pdf
- model motoru
 - identifikace (měření statických a dynamických vlastností)
 - regulace
 - měření s navrženým regulátorem bez rozšíření modelu soustavy
 - měření s navrženým regulátorem pro rozšířený model soustavy
 - měření vlivu změny parametrů
 - měření s PID regulátorem
 - měření vlivu poruchy
- model vrtulníku
 - identifikace (měření statických a dynamických vlastností)
 - regulace
 - měření s navrženým regulátorem pro jednotlivé subsystémy
 - měření vlivu poruchy