

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA KYBERNETIKY

BAKALÁŘSKÁ PRÁCE

**Interaktivní virtuální laboratoř pro výpočet
modelu systému s neurčitostí z omezených
experimentálních dat**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal BRABEC**
Osobní číslo: **A15B0513P**
Studijní program: **B3918 Aplikované vědy a informatika**
Studijní obor: **Kybernetika a řídicí technika**
Název tématu: **Interaktivní virtuální laboratoř pro výpočet modelu systému s neurčitostí z omezených experimentálních dat**
Zadávací katedra: **Katedra kybernetiky**

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

1. Analyzujte aktuální stav vývoje on-line laboratoří pro automatické řízení.
2. Analyzujte možnosti identifikace systému a jeho neurčitosti v systému Matlab.
3. Vyvíňte GUI pro efektivní zadávání modelu systému (z přechodové charakteristiky či omezené množiny charakteristických čísel).
4. Vytvořte modul pro výpočet a zobrazení neurčitostí ze zadaných dat (včetně jejich kombinace).
5. Otestujte laboratoř na praktických příkladech.

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

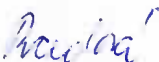
Seznam odborné literatury:

- 1) Dokumentace funkčních bloků systému REX, REX Controls, <http://www.rexcontrols.cz>
- 2) Reitinger, J., Čech, M., Schlegel, M., Balda, P. New tools for teaching vibration damping concepts: ContLab.eu (2014) IFAC Proceedings Volumes (IFAC-PapersOnline), 19, pp. 10580-10585.
- 3) Schlegel, M. ; Kvak, L. Computing value sets for systems with constrained step response. Process Control 2005, ISBN 80-227-2235-9, pp 1-7
- 4) Cech, M., Schlegel, M., Computing PID tuning regions based on fractional-order model set (2012) IFAC Proceedings Volumes (IFAC-PapersOnline), 2 (PART 1), pp. 661-666.
- 5) Schlegel, M., Cech, M. Computing value sets from one point of frequency response with applications (2005) IFAC Proceedings Volumes (IFAC-PapersOnline), 16, pp. 325-330.
- 6) Balda, P.; Čech, M. Java interface to REX control system. In Process control 2006. Pardubice: Technical University, 2006. s. 1-8. ISBN 80-7194-860-8.
- 7) Schlegel, M.; Čech, M. Internet PID controller design: www.pidlab.com. In IBCE '04. Grenoble: ENSIEG , 2004. s. 1-6.


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

Datum zadání bakalářské práce: **1. listopadu 2017**

Termín odevzdání bakalářské práce: **18. května 2018**


Doc. Dr. Ing. Vlasta Radová
děkanka




Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 1. listopadu 2017

Prohlášení

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

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

V Plzni dne 16. května 2018

.....

Michal Brabec

Poděkování

Tímto bych rád poděkoval Ing. Martinu Čechovi, Ph.D., za cenné rady, připomínky a věnovaný čas při vypracovávání bakalářské práce. Dále bych chtěl poděkovat kolegovi Michalu Vraštilovi za jeho ochotu a spolupráci na bakalářské práci.

Abstrakt

Cílem této bakalářské práce je představení virtuálních on-line laboratoří a jejich využití v oblasti automatického řízení, speciálně tedy laboratoří pro jednoduchý návrh regulátoru s omezenou strukturou. Na začátku práce je uveden aktuální stav spolu s možnými technikami vývoje a popisem dostupných realizací specializujících se na automatické řízení či robotiku. Další část se pak věnuje identifikaci systémů a jejich neurčitosti. Poslední část tvoří popis virtuální laboratoře, která je hlavním výsledkem této práce. Laboratoř se zaměřuje na výpočet modelu systému s neurčitostí a jednoduchý návrh regulátoru.

Klíčová slova: virtuální online laboratoř, webová aplikace, vzdálená laboratoř, identifikace procesu, neurčitost modelu procesu, množinový model

Abstract

The purpose of this Bachelor thesis is introduce virtual online laboratories and their applications in the field of automatic control, especially the laboratories for a simple design of the controller with a limited structure. At the beginning of the thesis the state-of-the-art is mentioned. The next part deals with the system identification and uncertainty. The last part of the thesis contains a description of the virtual laboratory, which is the main result of this thesis. The laboratory is focused on calculation of the system model with uncertainty and the simple design of the controller.

Keywords: virtual online laboratory, web application, remote laboratory, process identification, process model uncertainty, model set

Obsah

Seznam obrázků	8
Seznam tabulek	9
1 Úvod	10
2 Virtuální on-line laboratoře pro automatické řízení	11
2.1 Technologie pro vývoj virtuálních on-line laboratoří	12
2.1.1 HTML	12
2.1.2 CSS	13
2.1.3 JavaScript, AJAX, JavaScriptové knihovny	13
2.1.4 PHP	14
2.1.5 ASP.NET	14
2.1.6 Java applet	15
2.1.7 Adobe Flash	15
2.1.8 Shrnutí	16
2.2 Analýza virtuálních on-line laboratoří pro automatické řízení	16
2.2.1 Matlab Online	16
2.2.2 REX Conrols s.r.o.	17
2.2.3 Virtual Robotic Toolkit	18
2.2.4 UNILabs	19
2.2.5 Virtual Robotics Lab	19
2.2.6 Segway Control Simulation	20
2.2.7 Shrnutí	21
2.3 Virtuální laboratoře vs. vzdálené laboratoře	21
2.3.1 Společnosti nabízející laboratorní modely a zařízení pro výuku automatického řízení a robotiky	22
2.3.2 Shrnutí	24
3 Identifikace systému a jeho neurčitosti	25
3.1 Matematický model a jeho získání	25
3.2 Robustní řízení	25
3.2.1 Neurčitost modelu	26
3.2.2 Množinový model	29
3.3 Identifikace systémů v programu Matlab	30

4	Interaktivní virtuální laboratoř	33
4.1	Interaktivní vizualizace dat	33
4.2	JavaScriptové knihovny	36
4.2.1	Bootstrap	36
4.2.2	MathJax	37
4.2.3	Ostatní	37
4.3	Parametrizace extrémálních funkcí	37
4.3.1	Neceločíselný řád systému (FO)	37
4.3.2	Celočíselný řád systému (IO)	41
4.4	Základní uživatelské rozhraní a funkcionality	45
4.4.1	Kombinace jednoho bodu frekvenční charakteristiky a momentů impulsní funkce	48
4.4.2	Zobrazení extrémálních přenosových funkcí	49
4.4.3	Režim otevřené regulační smyčky	50
4.5	Přechodová charakteristika s neurčitostí	52
4.5.1	Popis realizace ve virtuální laboratoři	55
5	Závěr	57
	Literatura	58
A	Testování laboratoře	I
B	Obsah příloženého CD	VI
C	Výpočetní moduly	VII

Seznam obrázků

2.1	UNILabs - Virtuální laboratoř pro řízení servomotoru	19
2.2	Virtuální prostředí VR laboratoře	20
2.3	Prostředí laboratoře pro řízení segwaye	21
2.4	Ukázkové produkty společnosti Quanser - quanser.com [6]	23
2.5	Experimentální jednotka RT 200 Room temperature control - gunt.de [4]	24
3.1	Blokové schéma pro aditivní neurčitost	27
3.2	Blokové schéma pro vstupní multiplikativní neurčitost	27
3.3	Komplexní rovina s frekvenční charakteristikou	28
3.4	Schéma procesu návrhu robustního regulátoru	29
3.5	Charakteristiky pro parametrickou neurčitost	31
3.6	Bodeho frekvenční charakteristika pro nestrukturovanou neurčitost	32
4.1	Rastrová vs. vektorová grafika	34
4.2	Vykreslovací plocha pro grafy	35
4.3	Ukázkový Bootstrap panel	36
4.4	Obor hodnot množinového modelu pro $r = 1, \varphi = 1.8, \omega_1 = 1, n = 5,$ $m = 1, \omega = 0.5$	40
4.5	Obor hodnot množinového modelu pro $\kappa = 1, \mu = 1, \sigma_1^2 = 0.6, n = 5,$ $m = 1, \omega = 3$	41
4.6	Obor hodnot množinového modelu pro $r = 1, \varphi = 1.8, \omega_1 = 1, n = 5,$ $\omega = 0.5$	43
4.7	Obor hodnot množinového modelu pro $\kappa = 1, \mu = 1, \sigma_1^2 = 0.6, n = 5,$ $\omega = 3$	45
4.8	Úvodní rozložení uživatelského rozhraní	46
4.9	Ovládací prvky uživatelského rozhraní	47
4.10	Ukáзка průniku oborů hodnot	48
4.11	Panely pro zobrazení přenosových funkcí	49
4.12	Režim otevřené regulační smyčky	51
4.13	Přechodová charakteristika s uzlovými body	52
4.14	Kombinace uzlových bodů	53
4.15	Vygenerované body v_i	54
4.16	Konvexní mnohoúhelník	54
4.17	Konvexní mnohoúhelníky	55
4.18	Uživatelské rozhraní pro přechodovou charakteristiku s neurčitostí	56

Seznam tabulek

4.1	Tabulka předefinovaných tříd Bootstrap Grid System	36
-----	--	----

1 Úvod

Hlavním cílem této bakalářské práce bylo vytvoření virtuální on-line laboratoře specializující se na výpočet modelu systému s neurčitostí spolu s jednoduchým návrhem regulátoru. V teoretické části práce je čtenář blíže seznámen s technologií virtuálních on-line laboratoří, jejichž využití lze nalézt v automatickém řízení i robotice. Jedná se o jeden z možných směrů vývoje aplikací, který nachází své uplatnění v mnoha oborech lidské činnosti. Uplatnění takovýchto laboratoří je velice všestranné. Jejich upotřebení lze nalézt jak při výuce, kde je kladen důraz na získání teoretických i praktických zkušeností, tak i v komerční oblasti, kde je neocenitelným pomocníkem při zaškolování zaměstnanců a optimalizaci dílčích procesů. V tomto směru je právě on-line varianta těchto laboratoří velkou výhodou, protože je závislá pouze na vhodném prohlížeči a internetovém připojení, čímž umožňuje relativně snadnou a rychlou práci na různých typech zařízení bez potřeby většího softwarového vybavení. Přínosem takového přístupu je také jistá forma bezbariérovosti. Forma realizace dosavadních laboratoří je ale často již zastaralá či zcela nefunkční. Při vývoji nových nástrojů je tedy žádoucí využití relativně nových a perspektivních technologií.

Problematika virtuálních on-line laboratoří je detailněji popsána v první části práce, kde lze mimo jiné nalézt i možné technologie, jak on-line laboratoře vytvářet. Dále jsou pak uvedeny dohledatelné on-line laboratoře zaměřující se na automatické řízení nebo robotiku. Pro zabránění záměny pojmů „vzdálená laboratoř“ a „virtuální laboratoř“ je zde i kapitola, která oba termíny uvádí do souvislosti. V této kapitole je navíc představeno několik společností zabývajících se prodejem fyzických modelů pro řízení a regulaci.

Následná část práce je věnována identifikaci systémů a neurčitosti. Jsou zde uvedeny základní souvislosti a vztahy pro popis neurčitosti, která reprezentuje neúplnou znalost reálného systému. Zároveň je popsán i způsob realizace identifikace v prostředí Matlab.

V závěru práce je v několika kapitolách popsána virtuální laboratoř na jejímž vývoji jsem se v rámci této práce podílel [39]. V těchto kapitolách je popsán celý vývoj spolu s teoretickým pozadím a jsou zde představeny techniky, prostředí a funkcionality, jakožto i vlastní přínos obou autorů.

2 Virtuální on-line laboratoře pro automatické řízení

Automatické řízení je založeno na dvou hlavních myšlenkových proudech, kdy první z nich klade důraz na praktické zkušenosti, zatímco ten druhý na teorii a matematiku [26]. Dosažení rovnováhy mezi teoretickými znalostmi a inženýrskou intuicí je důležitým aspektem při studiu automatického řízení. V tomto smyslu hrají laboratorní experimenty klíčovou roli při propojování teorie a praxe. V mnoha případech je však docílení laboratorních zkušeností velice náročné a často i neuskutečnitelné. Tento problém lze vyřešit využitím technologie virtuální reality a vytvořit virtuální laboratoř [17, 25]. Pojem virtuální laboratoř lze definovat několika způsoby, mezi kterými je asi nejužitečnější tento:

„Virtuální laboratoři rozumíme interaktivní prostředí pro vytváření a provádění simulovaných experimentů [13].“

Jedná se tedy o technologii, která umožňuje uživatelům interakci se simulovaným prostředím (modelem). Tato technologie nachází své využití, jak v akademické sféře při výuce (tzv. e-Learning) či výzkumu, tak i v komerční oblasti. V praxi lze totiž tyto laboratoře využít nejen pro zaškolení nových zaměstnanců, ale i jako rychlý optimalizační nástroj.

Virtuální laboratoře lze implementovat buď jako „desktopové“ programy, které běží výhradně na operačním systému uživatele nebo jako webové aplikace, což jsou aplikace, které jsou uživatelům poskytovány z webového serveru prostřednictvím Internetu či intranetu¹. Roli klienta pak přebírá webový prohlížeč, který se vyskytuje jak na běžném stolním počítači či notebooku, tak na chytrém telefonu či tabletu. Webové aplikace, a s nimi spojený vývojový model tzv. cloud computing, jsou v současné době již silně rozšířené a používané v mnoha oborech lidské činnosti. Tato práce se zabývá právě využitím této technologie v oblasti automatického řízení.

Výhodou tohoto přístupu je relativně snadná dostupnost aplikace nezávislá na typu zařízení a jeho operačním systému a schopnost aktualizovat a spravovat webové aplikace bez nutnosti instalovat software na každém jednotlivém uživatelském zařízení.

Nevýhodou webových aplikací je vysoká závislost na poskytovateli aplikace a dostatečně dimenzované kapacitě připojení k serveru poskytovatele. Pokud poskytovatel

¹Obdoba Internetu pro interní komunikaci a spolupráci uvnitř firmy

přestane aplikaci zajišťovat nebo pokud dojde k přerušení spojení se serverem, může být služba dočasně či trvale nedostupná.[58, 60]

2.1 Technologie pro vývoj virtuálních on-line laboratoří

On-line virtuální laboratoře či obecně webové aplikace je možné vytvořit pomocí různých technologií. V této kapitole jsou uvedeny některé technologie, které lze při vývoji použít. Základem jsou samozřejmě technologie pro tvorbu webových stránek, kterými jsou HTML, CSS, JavaScript, PHP a další.

Ačkoli je mnoho webových aplikací psáno přímo v čistém programovacím jazyce (např. PHP), existuje pro jejich tvorbu řada systémů, tzv. frameworků. Využití těchto frameworků umožňuje rychlý vývoj aplikací, bez nutnosti řešit běžné problémy vývoje. Díky větší jednoduchosti a přehlednosti kódu zároveň často snižují výskyt chyb v aplikaci. Jako příklad lze uvést framework ASP.NET.[59, 60]

Další možností je využití softwarových komponent, které jsou přímo vložené do webové stránky. Většinou ale nepracují samostatně a pro svůj běh potřebují určitý zásuvný modul (tzv. plug-in). Jako příklad lze uvést Java applety či Flash aplikace. Tyto technologie byly velice rozšířené, ale v dnešní době jsou již na ústupu.

2.1.1 HTML

HTML je značkový jazyk navržený pro tvorbu webových stránek. Stránky jsou navzájem propojeny hypertextovými odkazy. HTML patří k hlavním jazykům pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.[15, 50]

Každý HTML dokument se skládá z elementů, které jsou samostatnými komponentami reprezentujícími jeden prvek značkovacího jazyka, a vytváří tak strukturu textu. Element se obvykle skládá z počátečního a koncového tagu. Mezi tagy je vloženo tělo (content), které může obsahovat další elementy a texty. Každý element může mít specifikovány atributy. Atributy poskytují informaci, jak by se měl element chovat či zobrazit.[18]

Nejnovější generací jazyka HTML je HTML5, která byla vyvinuta s cílem omezit závislost na zásuvných modulech a zlepšit obsluhu chyb. Mimo to přináší i řadu nových elementů, které tak částečně nahrazují skriptování. Mezi nejvýraznější elementy patří prvky pro integraci zvuku a videa. Další novinkou je také podpora vektorové grafiky bez použití podpůrných programů.

2.1.2 CSS

CSS (Cascading Style Sheets) neboli kaskádové styly je jazyk pro popis způsobu zobrazení elementů HTML dokumentů. Hlavním smyslem tohoto jazyka je oddělit obsah a vzhled dokumentu. Díky tomu je možné jednoduše sdílet styly mezi více stránkami. Nejnovější verzí tohoto jazyka je CSS 3, která se pojí se standardem HTML5.[18, 52]

Deklaraci stylu lze provést několika způsoby. Přímo u formátovaného elementu pomocí atributu „*style*“. Seznamem stylů (stylesheet) umístěným v hlavičce dokumentu mezi tagy „*style*“. Pomocí externího seznamu stylů umístěného v souboru s příponou „*css*“ na který se odkazuje tagem „*link*“.

2.1.3 JavaScript, AJAX, JavaScriptové knihovny

JavaScript (dále jen „JS“) je interpretovaný skriptovací jazyk pro webové stránky. Byl navržen tak, aby umožnil přidat interaktivitu do HTML stránek. Jsou jím tedy obvykle ovládány různé interaktivní prvky stránky (tlačítka, textová pole, ...) či tvořeny animace a různé efekty. Často se vkládá přímo do HTML kódu stránky nebo se nachází v externím souboru s příponou „*js*“ na který se odkazuje.

Program v JS se obvykle spouští na straně klienta (tzn. po stažení z Internetu), na rozdíl od jiných interpretovaných programovacích jazyků jako je PHP či ASP, které se spouštějí na straně serveru (tzn. před stažením z Internetu).[15, 18, 51]

Pro zachování interaktivity a dynamiky webové aplikace je vhodné při vývoji využívat technologii AJAX, která provádí asynchronní zpracování webových stránek pomocí JS knihovny. Umožňuje tak změnu obsahu stránek bez nutnosti kompletního znovunačtení dané stránky.[41]

Pro psaní rozsáhlejších aplikací je zároveň užitečné využívat JS knihovny, které usnadňují vývoj a umožňují lepší orientaci v rozsáhlém kódu. Takovýchto knihoven existuje celá řada, jako příklad lze uvést [27, 53]

- **jQuery** - usnadnění průchodu a manipulace s DOM, obsluha událostí, animace, AJAX, ...
- **D3.js** - dynamická a interaktivní vizualizace dat
- **MathJax** – zobrazení matematických výrazů a rovnic pomocí LaTeXu
- **Node.js** – škálovatelné internetové aplikace, především webové servery

2.1.4 PHP

PHP je skriptovací programovací jazyk, který je určený především pro programování dynamických webových stránek a webových aplikací. PHP vzniklo v roce 1996 a původní význam zkratky byl Personal Home Page, od té doby prošlo velkými změnami a nyní tato zkratka znamená Hypertext Preprocessor.

Jak již bylo zmíněno dříve, PHP pracuje na straně serveru, k uživateli je tedy přenášěn až výsledek činnosti skriptu. Jedná se o nejrozšířenější skriptovací jazyk pro web a to díky své jednoduchosti, bohaté zásobě funkcí a otevřenosti své technologie.[56]

2.1.5 ASP.NET

ASP je skriptovací platforma společnosti Microsoft, primárně určená pro dynamické zpracování webových stránek na straně serveru. Obsah stránky je tedy vygenerovaná na serveru a následně poslán do uživatelova webového prohlížeče. Za nástupce této technologie lze považovat ASP.NET, který je oproti samotnému ASP širší a komplexnější. ASP se od ASP.NET v mnoha věcech liší a je spíše podobá PHP.[45]

ASP.NET je součástí .NET frameworku a slouží pro tvorbu webových aplikací a služeb. Je založen na CLR², který je sdílen všemi aplikacemi postavenými na .NET frameworku. Program lze tak realizovat v jakémkoliv jazyce podporujícím CLR (např. Visual Basic .NET, JavaScript .NET, C#, Python, ...).

Společnost Microsoft nabízí 2 hlavní způsoby, jak vytvářet webové aplikace v ASP.NET a to pomocí ASP.NET WebForms nebo ASP.NET MVC.

ASP.NET WebForms vychází z konceptu standardní formulářové (okenní) aplikace pro Windows. Při tvorbě webových stránek je tedy možné používat ovládací prvky (např. Button, Label, ...). Těmto prvkům lze přiřazovat určité vlastnosti a zachytávat na nich události.

ASP.NET MVC je webový aplikační framework, který ale nenahrazuje WebForms. Jedná se pouze o jinou alternativu využití ASP.NET. Od WebForms se liší v tom, že umožňuje tvorbu webových aplikací podle softwarového architektonického vzoru Model-View-Controller (MVC). Rozděluje tedy webovou aplikaci do tří nezávislých komponent:

- **Model (Model)** obsahuje logiku a pracuje s databází.
- **Pohled (View)** přijímá potřebné informace od kontroléru a vykresluje uživatelské rozhraní tak, aby zobrazovalo tyto informace.

²CLR, neboli Common Language Runtime, je společná báze pro spouštění kódu všech programovacích jazyků v .NET frameworku.

- **Řadič (Controller)** reaguje na události a zajišťuje změny v modelu nebo pohledu.

V současné době se koncept MVC užívá především jako architektura webových aplikací. Implementace takovéto architektury zajišťuje, především u složitějších aplikací, přizpůsobivost a spolehlivost i při častých změnách a rychlém vývoji [28, 43, 44, 55]

Další generací ASP.NET je ASP.NET Core. Jedná se o reimplementaci ASP.NETu jako modulárního webového frameworku, který je používán (společně s dalšími frameworky jako je např. Entity Framework) pro vývoj moderních webových aplikací. ASP.NET Core používá open-source .NET Compiler platformu a je multiplatformní. Na rozdíl od ASP.NET je již skoro celý postaven na výše zmíněném architektonickém vzoru MVC.[42]

2.1.6 Java applet

Applet je malá aplikace, která běží v rámci jiného programu, s cílem plnit konkrétní funkci danou kontextem. Pro funkční využití obvykle obsahují nějakou formu uživatelského rozhraní. Typickým příkladem jsou Java applety, které bývají často brány jako synonymum pro tento typ technologie. Jak název napovídá, tyto typy appletů jsou napsány v programovacím jazyku Java či jiném jazyku, který lze zkompileovat do Java „bajtkódu“ a v této formě ho pak dodat uživatelům (např. Jython, JRuby, Pascal, ...).[47]

Java applet se spouští buď přímo na webové stránce, nebo ho lze stáhnout do počítače, kde pak běží jako samostatná aplikace nezávisle na dostupnosti poskytovatele. K oběma případům je potřeba běhové prostředí Java (JRE). Rychlost Java appletů je relativně velká a až do roku 2011 byly mnohem rychlejší než JavaScript.

I přes široké použití této technologie hlavní vývojáři webových prohlížečů jako je Microsoft, Google a Mozilla v roce 2016 oznámili plán na odstranění nebo omezení podpory pro Java applety, a to z důvodu zranitelnosti v zabezpečení, které bylo využíváno při útocích malwaru. Z tohoto důvodu se tedy již nepokračuje ve vývoji webových aplikací touto technologií a stávající aplikace jsou přeprogramovány pomocí HTML5 a CSS3.[31]

2.1.7 Adobe Flash

Adobe Flash je multimediální softwarová platforma pro vytváření animací a webových, desktopových i mobilních aplikací a her. Flash zobrazuje jak rastrovou, tak i vektorovou grafiku a umožňuje „streamování“ audiovizuálního materiálu. K rozšíření Flash na Internetu pomohla malá velikost výsledných souborů díky jejich uchování ve vektorovém formátu.[46]

Pro zobrazení Flash animací a aplikací je potřeba zásuvný modul webového prohlížeče (tzv. plug-in) Adobe Flash Player. Ačkoliv byl Flash dříve dominantní platformou pro on-line multimediální obsah, je stejně jako Java applet bezpečnostní riziko. Proto je postupně vytlačován a nahrazován HTML5 a JavaScriptem.

2.1.8 Shrnutí

Využití appletů a Flash aplikací mělo jistě svůj význam v době, kdy připojení k Internetu nebylo ještě takovou samozřejmostí jako dnes. Výhodou těchto technologií je možnost pracovat s danou aplikací on-line přes webový prohlížeč anebo ji stáhnout do počítače a pracovat s ní bez závislosti na Internetu. Postupné omezování podpory však zamezilo další vývoj tímto směrem.

Dnes je díky všudypřítomné dostupnosti Internetu a pokročilým webovým technologiím silně rozšířen tzv. cloud computing a webové aplikace, což jsou služby a aplikace poskytované serverem přes webový prohlížeč, většinou bez možnosti off-line režimu. Pro běh služby či aplikace je tedy potřeba pouze vhodný prohlížeč, díky čemuž je možné pracovat s danou aplikací přes různé druhy zařízení, což je pro použití v praxi jistě velkou výhodou. Tímto směrem se tedy ubírá i tato bakalářská práce. Pro zajištění provozu aplikace na různých platformách je využit framework ASP.NET Core, který zaručuje multiplatformitu.

2.2 Analýza virtuálních on-line laboratoří pro automatické řízení

Tato kapitola představuje na Internetu dostupné virtuální on-line laboratoře, které se specializují na oblast automatického řízení a částečně i na robotiku. Technologie laboratoří níže uvedených jsou rozličné, převažují ale applety a aplikace založené na programovacím jazyku Java.

2.2.1 Matlab Online

MATLAB je interaktivní programové prostředí a skriptovací programovací jazyk, který je vyvíjen společností MathWorks a jeho využití lze nalézt v inženýrských výpočtech, modelování a simulaci, analýze dat i tvorbě aplikací. MathWorks současně nabízí službu Matlab Online, která poskytuje přístup k prostředí Matlab přes kterýkoliv webový prohlížeč a není tedy zapotřebí nic stahovat a instalovat. Služba je ale omezena pro studentské a nekomerční účely. K on-line prostředí se lze pohodlně dostat přes uživatelský účet na stránkách společnosti MathWorks. Uživatelé tak mohou

na svých projektech pracovat prakticky odkudkoli. Další výhodou je i možné sdílení souborů s ostatními uživateli a využít tak možnosti pracovat na různých projektech společně.[9, 54]

Důležitou součástí Matlabu jsou knihovny funkcí a aplikací, které se obecně označují jako tzv. toolboxy. Jednotlivé toolboxy se soustředí na určité obory (např. statistika, zpracování obrazu a počítačové vidění, zpracování signálu a bezdrátová komunikace, řídicí systémy a další). Nevýhodou služby Matlab Online je fakt, že v současné době nabízí v oblasti řídicích systémů (mimo jiné) pouze část z množiny toolboxů, které jsou normálně dostupné v běžném Matlabu[10].

- **Control System Toolbox**

Jedná se o nástroj, který poskytuje algoritmy a aplikace pro systematickou analýzu, návrh a ladění lineárních řídicích systémů. Jako příklad aplikace tohoto toolboxu lze uvést aplikaci Control System Designer, která umožňuje navrhnout regulátory pro SISO systémy.

- **Robust Control Toolbox**

Toolbox poskytuje funkce a bloky pro analýzu a ladění výkonných a robustních řídicích systémů pro řízené systémy s neurčitostí. Toolbox automaticky vyladí jakýkoliv řídicí systém, tak aby splňoval cíle návrhu.

- **Fuzzy Logic Toolbox**

Toolbox slouží k analýze, návrhu a simulaci systémů založených na fuzzy logice. Umožňuje modelovat komplexní systémové chování pomocí jednoduchých logických pravidel a poté implementovat tyto pravidla v tzv. Fuzzy Inference Systems.

- **Model Predictive Control Toolbox**

Tato knihovna se zabývá návrhem a simulací prediktivních modelů regulátorů, které lze využít k optimalizaci výkonu MIMO systému s uzavřenou smyčkou v závislosti na vstupních a výstupních omezeních.

2.2.2 REX Controls s.r.o.

Společnost REX Controls se zabývá výzkumem a vývojem pokročilých algoritmů, podpůrných a diagnostických nástrojů i kompletních programových řešení pro přímé řízení strojů a procesů v reálném čase [12]. Pro tuto oblast poskytuje nemalé množství virtuálních laboratoří, které jsou ve formě Java apletů a slouží pro interaktivní návrh a ladění PID regulátoru, prezentaci pokročilých PID autotunerů a simulaci komplexních PID řídicích struktur. Nabízené laboratoře současně slouží jako ukázka funkčních bloků a řídicích struktur řídicího systému REXYGEN vyvíjeného touto společností.

Z čehož plyne i jeden z vedlejších cílů této práce a tím je průzkum a otestování nových technologií, kterými by se dali nahradit současná řešení. Ve vývoji společnost úzce spolupracuje s Katedrou kybernetiky Západočeské univerzity v Plzni. Pro jednotlivé oblasti působení jsou níže uvedeny některé z virtuálních laboratoří, které se zabývají danou tematikou.[20]

- **PID Control Lab**

Jedná se o interaktivní nástroj pro robustní návrh a ladění PID regulátorů. Obsahuje funkce pro identifikaci průmyslových procesů z experimentálních dat založené na množinovém modelu. Zároveň umožňuje optimalizovat parametry PID regulátoru na základě požadavků na bezpečnost ve fázi a v zesílení, tvar citlivostních funkcí, šířku pásma a potlačení poruch.

- **PIDAT – PID regulátor s reléovým autotunerem**

Tato laboratoř prezentuje použití funkčního bloku PIDAT, který obsahuje PID regulátor s reléovým autotunerem. Ladicí mechanismus používá adaptivní filtr, pomocí kterého je určen jeden bod frekvenční charakteristiky řízeného systému s vhodným fázovým zpožděním. Na základě tohoto bodu jsou pak vypočteny parametry regulátoru.

- **PIDMA – PID regulátor s pulsním autotunerem**

Oproti předchozí laboratoři je zde prezentován PID regulátor s pulsním (momentovým) autotunerem, kdy jsou odhadovány první tři momenty impulsní funkce. Model procesu, který je výchozí pro automatické nastavení parametrů PID regulátoru, je sestaven právě na základě těchto momentů.

- **Smithův prediktor**

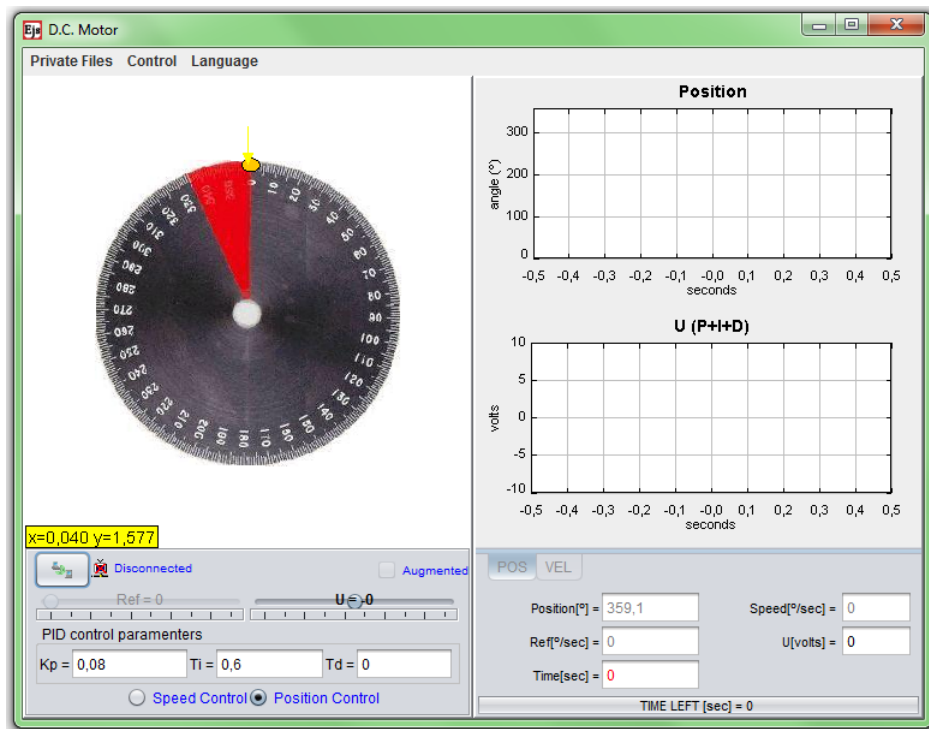
Jedná se o jednu z virtuálních laboratoří simulujících pokročilé řídicí struktury, které jsou postavené na základě funkčních bloků systému REXYGEN. Konkrétně tato laboratoř slouží k demonstraci Smithova prediktoru, který se využívá k řízení systémů s výrazným dopravním zpožděním.

2.2.3 Virtual Robotic Toolkit

Aplikace Virtual Robotics Toolkit navržena pro použití s Lego Mindstorms NXT a EV3 je simulátor fyzikálního prostředí. Umožňuje uživatelům navrhnout a naprogramovat jejich vlastního virtuálního robota. Tento nástroj může být obzvláště užitečný pro výuku v oblasti robotiky.[7]

2.2.4 UNILabs

UNILabs je univerzitní síť interaktivních laboratoří využívajících technologii JAVATM Web Start. Ve všech laboratořích na této síti lze buď experimentovat na reálném modelu, nebo využít modelu matematického a provádět pouze simulace. Jednotlivé laboratoře se specializují na problematiku automatického řízení v různých oblastech, např. systémy rozvodu tepla, řízení servomotorů (obr. 2.1), řízení úrovně hladiny v nádržích [38]. Nevýhodou těchto laboratoří je použitá technologie, která, jak bylo výše popsáno, přestává být podporovaná většinou prohlížečů. Protože se jedná o univerzitní síť, je nutné poznamenat, že nejsou všechny laboratoře (kurzy) volně přístupné³.



Obr. 2.1: UNILabs - Virtuální laboratoř pro řízení servomotoru

2.2.5 Virtual Robotics Lab

Jedná se o virtuální laboratoř, která simuluje laboratorní prostředí s cílem naprogramovat mobilního robota⁴. V podstatě se jedná o hru s virtuální realitou (obr. 2.2), kde je úkolem sestavení všech fyzických součástí robota a robotické ruky, naprogramovat robota tak, aby byl schopen zvednout lahev a jako poslední i řídit jeho kola. Po absolvování všech úkolů je možné sledovat po laboratoři se pohybujícího robota [14]. Tato laboratoř či „hra“ je realizována pomocí technologie Adobe Flash a je vytvořena

³UNILabs - Volně přístupné laboratoře

⁴VRLab

spolu s dalšími virtuálními laboratořemi v rámci The Mind Project, který se zaměřuje na předvádění a implementaci robotiky a umělé inteligence do různých technických i netechnických oblastí.



Obr. 2.2: Virtuální prostředí VR laboratoře

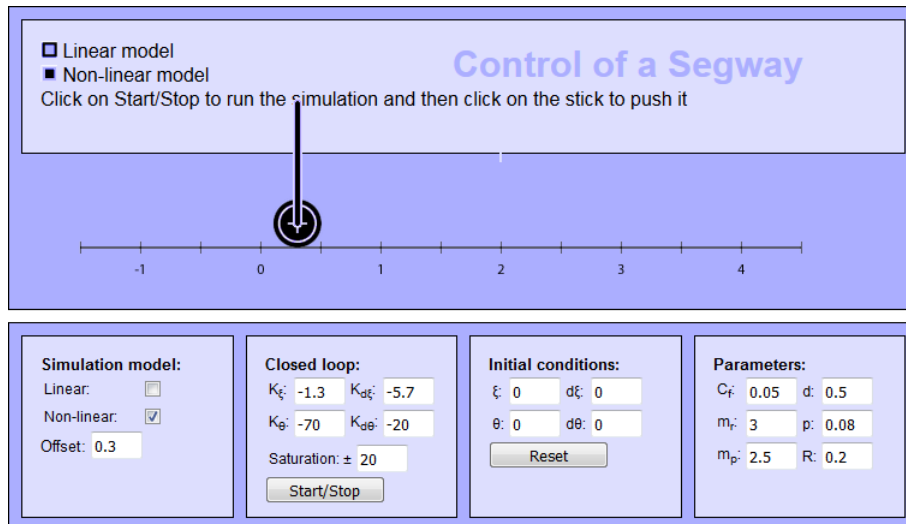
2.2.6 Segway Control Simulation

Segway Control Simulation je jeden z mnoha výukových on-line nástrojů, které jsou vytvořené v rámci evropského výzkumného projektu Next-lab navazujícího na projekt Go-lab, který se zaměřuje na zavedení a podporu inovativních a interaktivních výukových metod na školách. Next-lab nabízí široké spektrum výukových virtuálních a vzdálených laboratoří z různorodých vědních oblastí.

Tato laboratoř se konkrétně zabývá problematikou Segwaye, což je dvoukolový elektrický dopravní prostředek, který pro pohyb využívá dynamickou stabilizaci. Cílem této laboratoře je ukázat důležitost řídicího systému, který je v Segway zabudovaný, kdy je potřeba neustále sledovat polohu a orientaci vozidla a přes pohyb kol zajišťovat rovnováhu.[21]

Pro návržení regulátoru lze na Segway nahlížet jako na inverzní kyvadlo, které lze popsat pomocí diferenciálních rovnic. Laboratoř (obr. 2.3) umožňuje sledovat RT simulaci pohybu Segwaye (inverzního kyvadla) v závislosti na parametrech systému a počátečních podmínkách⁵.

⁵[Segway Control Lab](#)



Obr. 2.3: Prostředí laboratoře pro řízení segwaye

2.2.7 Shrnutí

Ačkoliv jsou virtuální laboratoře relativně dlouho známé a v různých oborech silně využívané, v oblasti automatického řízení neexistuje mnoho implementací tohoto přístupu. Navíc se volně dostupné laboratoře již většinou neudržují, a to z důvodů finančních či technologických. Větší část takovýchto služeb je pak navíc ve formě off-line aplikace. Další podstatnou část ještě tvoří laboratoře vzdálené, které jsou popsány v následující kapitole.

Výuka automatického řízení je náročná, neboť se jedná o disciplínu s velmi dynamickým vývojem a mezioborovou povahou. Pro zlepšení současného stavu by jistě pomohlo sdílení zdrojů v rámci jedné a více univerzit, které by umožnilo přístup k různorodým projektům a bylo by neocenitelným přínosem pro výuku a výzkum v oblasti automatického řízení a robotiky, jak v lokálním, tak celosvětovém měřítku.

2.3 Virtuální laboratoře vs. vzdálené laboratoře

Souběžně s pojmem virtuální laboratoř se pohybuje i pojem vzdálená laboratoř či vzdáleně ovládaná laboratoř, anglicky remote laboratory. Tento termín označuje skutečný reálný experiment, který mohou vzdálení uživatelé prostřednictvím ovládacího rozhraní ovládat a s využitím webové kamery pozorovat, což umožňuje relativně snadné shromažďování dat, která jsou oproti virtuálním laboratořím získaná ze skutečných zařízení. V některých laboratořích se aplikují obě technologie. Je tedy možné vzdáleně ovládat laboratorní experimenty nebo využít model dané úlohy a experiment pouze simulovat. Pojem virtuální a vzdálená laboratoř je tedy velice důležité vzájemně odlišovat.

Výhodou vzdálených laboratoří je možnost provádět reálné experimenty z kteréhokoli místa na světě s připojením k Internetu. Zařízení, na kterém je experiment prováděn, může být navíc vysoce specializované a složité. Vývoj takové virtuální laboratoře, která by simulovala toto zařízení, je poté velice složitý a nákladný. Navíc práce se skutečným experimentem poskytuje skutečnou laboratorní zkušenost na rozdíl od pouhých simulací. Získání takovýchto zkušeností je ale často náročný proces.

Naopak výhodou virtuálních laboratoří je možná interoperabilita hned několika uživatelů. Počet uživatelů je omezen pouze výpočetním výkonem hostitelského počítače. Virtuální laboratoř je také robustnější než reálná zařízení, která mohou být při špatném nastavení či chybě zničena. Další výhodou je také možnost simulace procesů za námi daných podmínek, což není v reálné laboratoři vždy možné.[36]

Stejně jako u virtuálních laboratoří i zde lze uvést jako příklad laboratoře ze sítě UNILabs⁶, což nás přivádí k další části této podkapitoly. Při použití vzdálených laboratoří jsou totiž kromě ovládacího rozhraní důležitá i samotná fyzická zařízení (fyzické modely), která se zaměřují na určitou problematiku. Kvalita těchto zařízení má velký vliv na průběh a výsledky experimentů. V následující kapitole jsou uvedeny některé firmy, jejichž produkty tvoří hlavní část současného trhu a také firmy, které jsou již na ústupu, ale jejich výrobky jsou mezi uživateli stále používány.

2.3.1 Společnosti nabízející laboratorní modely a zařízení pro výuku automatického řízení a robotiky

- **QUANSER**

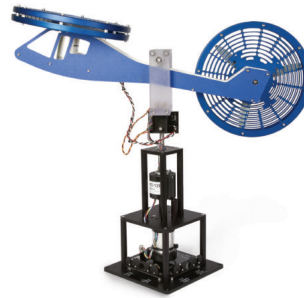
Společnost Quanser poskytuje široké spektrum stolních hardwarových platforem, které jsou optimalizované pro výuku a výzkum v inženýrském vzdělání. Firma nabízí produkty cílené na oblasti automatického řízení, robotiky, mechatroniky, zemětřesení, řízení v letectví a další [6]. Některé jejich produkty lze nalézt v již zmíněné UNILabs.

Pro práci s moduly lze využít prostředí Matlab/Simulink s doplňkem Quanser QUAR nebo LabView, který využívá doplněk Quanser RCP Toolkit. Ke každému modelu je dodán dynamický model dané úlohy a potřebná dokumentace obsahující podrobnosti o daném modulu, včetně teoretických základů v dané oblasti. Jako příklady nabízených produktů lze uvést rotační inverzní kyvadlo obr. 2.4(a), „Joint Control Robot“ - 6DOF, 2DOF vrtulník obr. 2.4(b), „Active Mass Damper“.

⁶[UNILabs-Laboratories catalog](#)



(a) Rotační inverzní kyvadlo



(b) 2DOF vrtulník

Obr. 2.4: Ukázkové produkty společnosti Quanser - quanser.com [6]

- **Googol Technology**

Googol Technology je čínská firma, která se specializuje na výzkum, vývoj a produkci regulátorů pohybu a řídicích systémů. Současně nabízí i produkty pro výuku v těchto oblastech. Společně s fyzickým modelem je dodáván i model matematický pro prostředí Matlab.[3]

Jako příklad nabízených produktů pro oblast vzdělávání lze uvést Ball & Beam control system⁷, který je speciálně navržen pro kurzy automatického řízení. Řídicí systém ovládá polohu nerezové kuličky, která je umístěna na dráze, nastavováním úhlu sklonu této dráhy.

- **GUNT Hamburg**

Německá firma Gunt vyrábí a dodává zařízení cílené na výuku v inženýrství. Oblast působení je dosti rozsáhlá. Nabízí výukové nástroje pro oblast mechaniky, mechatroniky, procesního inženýrství i životního prostředí.[4]

Jako příklad lze uvést experimentální jednotku RT 200 Room temperature control⁸, jejímž cílem je ovládat teplotu ve virtuální místnosti. Jednotlivé součásti této regulační smyčky jsou vzájemně propojené a samostatně umístěné na panelech, které jsou vloženy do rámu.

⁷Ball & Beam Control System

⁸RT 200 Room temperature control



Obr. 2.5: Experimentální jednotka RT 200 Room temperature control - gunt.de [4]

- **AMATROL**

Mimo kurzy a produkty pro širokou oblast inženýrského vzdělávání nabízí společnost Amatrol i rozsáhlou řadu automatizačních výukových systémů, a to ve dvou hlavních oblastech: robotika a programovatelné logické automaty (PLC). Konkrétně jde o zařízení simulující automatizované výrobní linky.[1]

Další zajímavou oblastí je i procesní řízení. Společnost pro tuto problematiku vyvinula širokou nabídku výukových nástrojů pro řízení průmyslových procesů, a to hned v několika oblastech, např. řízení hladiny a průtoku, teploty, tlaku a další.[2]

2.3.2 Shrnutí

Uvedené společnosti nabízí propracovaná řešení, která mohou být velkým přínosem při výuce automatického řízení. Jak již bylo ale zmíněno, jednou z nevýhod fyzických modelů jsou jejich omezené možnosti způsobené právě fyzickou stránkou věci. Pro výuku či experimenty s větším počtem studentů by bylo totiž zapotřebí většího počtu těchto modelů. Studenti by se tak mohli individuálně seznámit s daným problémem a získali by tak jedinečné zkušenosti pro budoucí praxi. Větší počet zařízení by byl ale dosti nákladný a prostorově náročný. Velké a nákladné modely by tak šlo nahradit dobrým matematickým modelem umístěným na relativně levném hardwaru (např. Raspberry Pi). Tento postup vývoje je označován jako hardware-in-the-loop simulace a jeho využitím lze zlevnit výukový a návrhový proces a zpřístupnit ho tak pro širší oblast zájemců.

3 Identifikace systému a jeho neurčitosti

Pro efektivní řízení libovolného systému je nutná znalost určitých jeho vlastností, které jsou důležité pro jeho řízení ve specifické oblasti. Tyto vlastnosti jsou reprezentovány problémově orientovaným modelem.

3.1 Matematický model a jeho získání

Matematický model je zjednodušením reálného systému a zobrazuje charakteristické vlastnosti zkoumaného procesu. Lze jej získat dvěma způsoby či kombinací těchto dvou způsobů. První možností je matematicko-fyzikální modelování (white box), které využívá matematicko-fyzikální či ekonomické zákony k sestavení algebraických a diferenciálních nebo diferenčních rovnic, které popisují chování systému. Parametry modelu mají tak jasný fyzikální či ekonomický význam. Tento postup určování modelu vyžaduje hlubokou znalost fyzikálních procesů probíhajících ve zkoumaném systému.

Druhou možností je experimentální identifikace (black box) na základě naměřených vstupních a výstupních dat. Tyto data jsou získána vybuzením systému vhodným testovacím signálem. Parametry modelu se následně určí pomocí vhodné identifikační metody. Takto získané parametry modelu nemají většinou jasný fyzikální význam, oproti parametrům, které jsou získané matematicko-fyzikálním modelováním, a není tedy znám jejich význam ve vztahu k reálným parametrům systému. Zároveň, ale není tak potřebná detailní znalost probíhajících procesů. Potřebná je ale existence zkoumaného systému a možnost experimentovat s ním.

Třetí možností, jak získat model je kombinace předchozích dvou řešení (gray box), kdy při experimentální identifikaci využijeme určitou apriorní informaci o zkoumaném procesu (např. pro určení struktury modelu).[5, 23, 29, 30, 40]

3.2 Robustní řízení

Robustní řídicí systém je takový systém, který není citlivý na rozdíly mezi skutečným systémem a jeho modelem použitým pro návrh regulátoru. Tyto rozdíly jsou označovány jako nesoulad modelu a řízeného systému či modelová neurčitost. Regulátor je tedy navržen tak, aby v určitých mezích vyhověl stanoveným požadavkům i při změnách parametrů soustavy.[16, 37]

3.2.1 Neurčitost modelu

Tato podkapitola je vytvořena na základě [22, 24].

Model systému bez ohledu na to, jak je podrobný nikdy není zcela přesnou kopií skutečného fyzického systému. Vždy se jedná pouze o aproximaci reálné dynamiky daného systému a vždy bude obsahovat určitou neurčitost. Tuto neurčitost lze rozdělit na dvě hlavní kategorie: neurčitost způsobená rušením signálu a dynamické poruchy. První typ se vyskytuje při experimentální identifikaci a zahrnuje rušení vstupních a výstupních signálů, šum v senzorech a aktuátorech. Druhá kategorie reprezentuje nesoulad mezi matematickým modelem a skutečnou dynamikou běžícího systému a vyskytuje se tedy při matematicko-fyzikálním modelování. Typickým zdrojem nesrovnalostí může být nemodelovaná (obvykle vysokofrekvenční) dynamika a zanedbaná nelinearita systému. Tyto chyby mohou nepříznivě ovlivnit stabilitu a výkon řídicího systému.

Aby byla tato neurčitost zohledněna, je potřeba nejdříve najít model neurčitosti, který se skládá z nominálního modelu a množiny modelů, někdy označované jako množina neurčitostí, která obsahuje systémovou neurčitost a snadněji se s ní manipuluje. Poté je navržen řídicí systém tak, aby splňoval požadavky na stabilitu a výkonnost nejen pro nominální model, ale i pro další možné modely z modelu neurčitosti.

Typy neurčitostí

- **Nestrukturované**

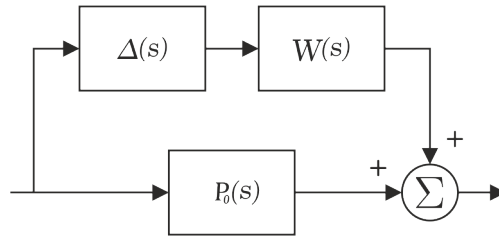
Mnoho dynamických poruch, které se mohou vyskytnout v různých částech systému, může být soustředěno do jediného bloku poruch Δ , který musí splňovat následující podmínku

$$\|\Delta\|_{\infty} < 1, \quad (3.1)$$

kde $\|\Delta\|_{\infty}$ je H_{∞} norma z Δ . Velikost neurčitosti a její frekvenční závislost je následně vymezena pomocí váhové funkce W . Takto vyjádřená neurčitost se označuje jako nestrukturovaná. Nestrukturovanou neurčitost je možné vyjádřit v různých formách, většinou se však používají aditivní a multiplikatívni formy.

Aditivní neurčitost

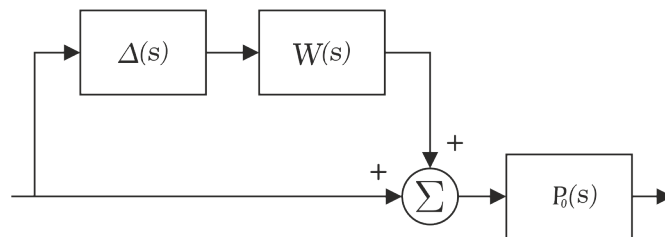
$$P(s) = P_0(s) + W(s)\Delta(s) \quad (3.2)$$



Obr. 3.1: Blokové schéma pro aditivní neurčitost

Vstupní multiplikační neurčitost

$$P(s) = P_0(s)[I + W(s)\Delta(s)] \quad (3.3)$$



Obr. 3.2: Blokové schéma pro vstupní multiplikační neurčitost

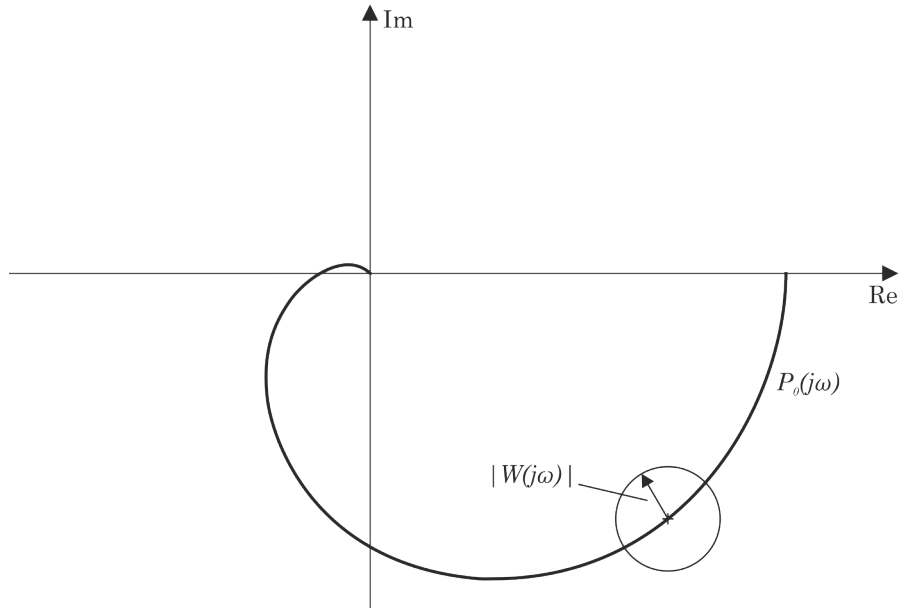
Pokud je tedy předpokládána aditivní chyba na vstupu systému, lze systém popsat v tomto tvaru

$$P(j\omega) = P_0(j\omega) + W(j\omega)\Delta(j\omega).$$

Jednotlivé body frekvenční charakteristiky $P(j\omega)$ jsou následně vymezeny kružnicí se středem na frekvenční charakteristice $P_0(j\omega)$ a poloměrem $|W(j\omega)|$, což je znázorněno na obr. 3.3.

- **Parametrické**

Nestrukturovaná reprezentace neurčitosti, která je popsána výše, je užitečná při popisování nemodelované nebo zanedbané dynamiky systému. Dynamické poruchy v mnoha průmyslových řídicích systémech však mohou být také způsobeny nepřesným popisem charakteristik jednotlivých součástí či narušením nebo opotřebením určitých komponentů, které pak snižují účinnost řízeného systému. Tyto



Obr. 3.3: Komplexní rovina s frekvenční charakteristikou

nepřesnosti lze charakterizovat změnou specifických parametrů systému na rozsahy hodnot. Takováto neurčitost se nazývá parametrickou a ovlivňuje výkonnost na nízkých frekvencích.

Př.

$$G = \frac{K}{1 + \tau s} e^{-\lambda s}$$

$$K \in \langle 8.5, 11.5 \rangle$$

$$\tau \in \langle 0.85, 1.15 \rangle$$

$$\lambda \in \langle 0.425, 0.575 \rangle$$

- **Strukturované**

Při návrhu robustního regulátoru se lze často setkat s případem, kdy je vhodné využít kombinaci několika druhů neurčitostí, která může obsahovat jak nestrukturovanou, tak i parametrickou neurčitost. Tento druh neurčitosti lze realizovat pomocí horní lineární zlomkové transformace $F(M, \Delta)$ matic M a Δ , kde M je matice propojené přenosové funkce ve tvaru

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (3.4)$$

a blok neurčitosti Δ je definován jako

$$\Delta = \text{diag}[\delta_1 I_{r_1}, \dots, \delta_s I_{r_s}, \Delta_1, \dots, \Delta_f] : \delta_i \in \mathbb{C}, \Delta_j \in \mathbb{C}^{m_j \times m_j}, \quad (3.5)$$

kde $\sum_{i=1}^s r_i + \sum_{j=1}^f m_j = n$ a n je rozměr bloku Δ . Pokud je $(I - M_{11}\Delta)$ regulární lze horní lineární zlomkovou transformací získat ze vztahu

$$F(M, \Delta) = [M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1} M_{12}]. \quad (3.6)$$

Jako příklad matice M lze uvést realizaci pro aditivní neurčitost.

$$M = \begin{bmatrix} 0 & I \\ I & P_0 \end{bmatrix}$$

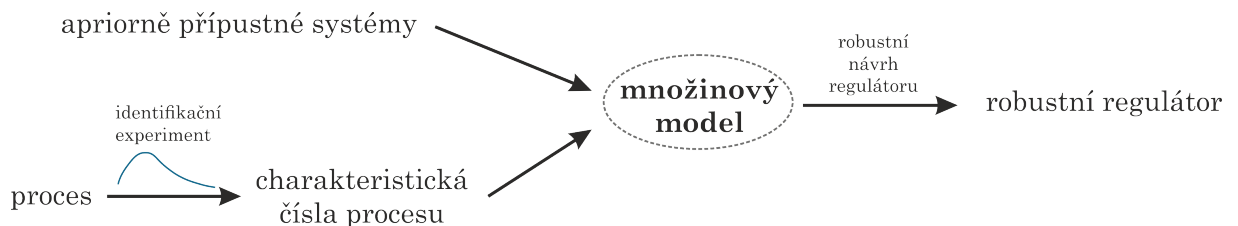
Protože má blok Δ konkrétní strukturu je tento typ neurčitosti nazýván strukturovaný.

3.2.2 Množinový model

Další možností jak vyjádřit neúplnou znalost skutečného systému pro robustní návrh regulátoru je pomocí množinových modelů. Při tomto přístupu je předpokládána přesná znalost několika charakteristických čísel procesu. Další informace o struktuře (např. řád systému) či parametrech jsou ale již nejasné a je tedy specifikován pouze obecný tvar s apriorní informací. Při návrhu regulátoru není tedy uvažován pouze jeden nominální model, ale celá množina systémů, která vyjadřuje nepřesnou znalost reality. Množinový model je pak tedy definován jako množina přípustných přenosů, které vyhovují informacím dvojího druhu o řízeném systému:

- **Apriorní informace** určující tvar přenosu a omezení na polohu pólů
- **Informaci získané měřením** v identifikačním experimentu, skládající se z několika charakteristických čísel procesu

Smyslem apriorní informace je vymezit v množině všech lineárních systémů podmnožinu, která by dostatečně přesně popisovala uvažovaný okruh reálných procesů. Systémy z takovéto třídy se označují jako apriorně přípustné. Všechny apriorně přípustné systémy s naměřenými charakteristickými čísly tvoří dohromady množinu přípustných systémů neboli množinový model. Zvolený robustní regulátor pak musí zaručovat splnění požadavků pro libovolný systém množinového modelu a musí minimalizovat zvolené optimalizační kritérium.[32] Proces návrhu lze popsat jednoduchým schématem na obr. 3.4.



Obr. 3.4: Schéma procesu návrhu robustního regulátoru

Z experimentálních dat je získán množinový model, jehož obor hodnot představuje ve frekvenční oblasti jeho neurčitost pro danou frekvenci. Vhodnou kombinací apriorního omezení s určitým typem charakteristických čísel je pak možné pro každou frekvenci specifikovat hranici této neurčitosti pomocí několika tzv. extrémálních přenosových funkcí. Pro nalezení hranice oboru hodnot (hranice neurčitosti) pro jednotlivé frekvence je potřeba nalézt parametrizaci všech těchto extrémálních funkcí.[19]

3.3 Identifikace systémů v programu Matlab

Hlavním nástrojem pro identifikaci systémů v programu Matlab je „**System Identification Toolbox**“ [8]. Toolbox poskytuje funkce z programu Matlab, funkční bloky z programu Simulink a aplikace pro vytváření matematických modelů dynamických systémů z naměřených vstupních a výstupních dat. Pro identifikaci parametrů modelu lze využít data v časové i frekvenční oblasti.

Před samotným odhadem modelu je většinou potřeba data ještě předzpracovat (z důvodu šumu, chybějících hodnot a dalších anomálií) pomocí filtrace, převzorkování či rekonstrukce. Mimo úpravu vstupně-výstupních dat umožňuje toolbox i analýzu jejich vhodnosti pro identifikaci a diagnostiku o přetrvávajícím buzení, existenci zpětnovazební smyčky a přítomnosti nelinearit.

Jak již bylo uvedeno System Identification Toolbox odhaduje parametry modelu a jejich neurčitost z časové nebo frekvenční odezvy. Model reálného systému lze poté získat ve formě přenosové funkce, stavového popisu systému, procesního modelu, ARX modelu, nelineárního modelu a v mnoha dalších formách.

Pro práci s toolboxem lze využít buď příkazový řádek v Matlabu nebo grafické uživatelské rozhraní (GUI), které podstatně usnadňuje práci s daty a modely. Lze tak velice snadno pracovat s větším množstvím vstupních dat a s možnými formami vypočtených modelů. Zároveň je možné vykreslovat různé charakteristiky pro tyto modely a vzájemně je porovnávat.

Pro validaci modelu je možné porovnat nezávislé skutečné výstupy s výstupy z modelu. Je možné zobrazit i chybu predikce a vytvořit grafy časových a frekvenčních odezev s ohraničenými oblastmi spolehlivosti, z čehož lze sledovat vliv neurčitosti parametrů na výstup z modelu. V rámci GUI toolboxu lze pro další použití vypočítaný model exportovat do „workspace“ v Matlabu.

Pro navrhování robustních systémů lze poté využít „**Robust Control Toolbox**“ [11]. S pomocí tohoto nástroje lze vytvořit model neurčitosti kombinací nominální dynamiky s prvky neurčitosti. Je možné tak analyzovat vliv neurčitosti modelu zařízení na výkonnost řídicího systému a identifikovat nejhorší kombinace prvků neurčitosti.

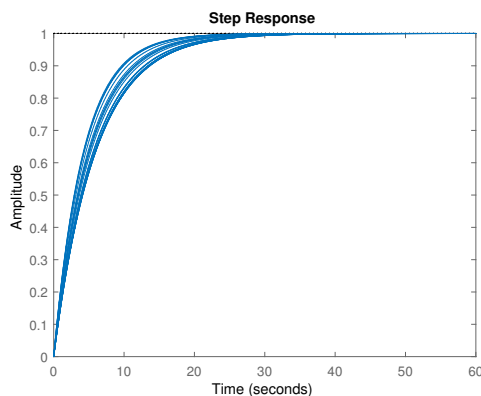
Mezi hlavní formy používané modelové neurčitosti patří neurčitost v parametrech

a nemodelovaná dynamika. První z nich se realizuje pomocí funkce *ureal()*, která vytvoří objekt reprezentující reálné číslo s neurčitou hodnotou. Neurčitost této hodnoty lze vyjádřit ve třech formách: aditivní odchylka od nominální hodnoty, interval s nominální hodnotou a procentuální odchylka od nominální hodnoty. Výsledný objekt je následně použit pro vytvoření stavového modelu či přenosové funkce.

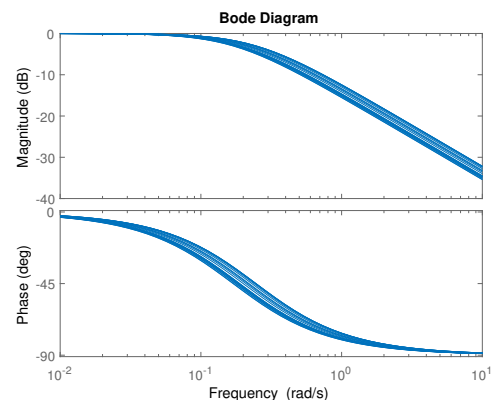
V následujícím příkladu je vytvořen neurčitý reálný parametr a s nominální hodnotou $a_0 = 5$ a aditivní odchylkou ± 1 . Parametr a je poté použit v přenosové funkci systému prvního řádu. Pro lepší představu je na obr. 3.5 zobrazena přechodová a bodeho charakteristika.

Př.

```
a = ureal('a',5);
P = tf([1],[a 1]);
```



(a) Přechodová charakteristika



(b) Bodeho charakteristika

Obr. 3.5: Charakteristiky pro parametrickou neurčitost

Pro popis nemodelované dynamiky, tedy rozdílu mezi modelem a skutečným systémem, lze využít funkci *ultidyn()*, která vytvoří dynamický LTI objekt neurčitosti. V kombinaci s nominálním modelem a filtrem, pro tvarování frekvenční odezvy, mohou být tyto objekty použity k zachycení neurčitosti, která je spojená s dynamikou modelu. Nominální model se často tvoří s parametrickou neurčitostí a pro vytvoření filtru (též označován jako váha) se využívá funkce *makeweight()*, která vytvoří filtr prvního řádu se specifikovaným nízkofrekvenčním (dc_{gain}) a vysokofrekvenčním (hf_{gain}) zesílením a „crossover“ frekvencí (ω_c).

Zmíněné funkce jsou použity v následujícím příkladu, kdy je opět vytvořen neurčitý reálný parametr a s nominální hodnotou $a_0 = 5$ a aditivní odchylkou ± 1 . Tento

parametr je využit pro vytvoření nominálního modelu P_0 . Dále je vytvořena váhová funkce W s $dc_{gain} = 0.1$, $hf_{gain} = 10$, $\omega_c = 1$ a dynamický LTI objekt Δ . Všechny uvedené prvky jsou ve výsledku použity pro sestavení modelu se vstupní multiplikační neurčitostí. Pro představu vlastností výsledného modelu je na obr. 3.6 vykreslena bodeho charakteristika.

Př.

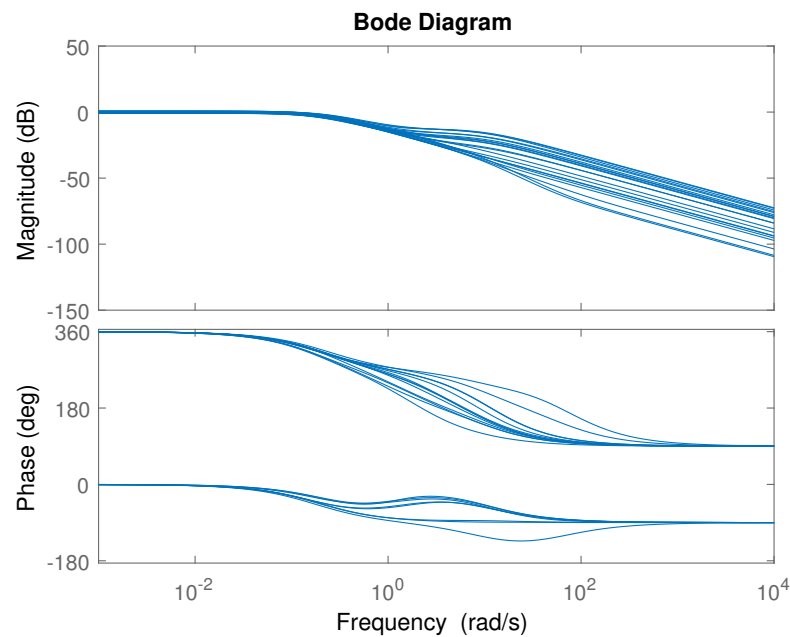
```

a = ureal('a',5)
P0 = tf([1],[a 1])

W = makeweight(0.1,1,10);
Delta = ultidyn('delta',[1 1]);

P = P0*(1+W*Delta);

```



Obr. 3.6: Bodeho frekvenční charakteristika pro nestrukturovanou neurčitost

4 Interaktivní virtuální laboratoř

Pro předvedení možností využití webových aplikací v oblasti automatického řízení byla vytvořena virtuální laboratoř. Postupy a technologie využitě pro její vývoj představují jednu z mnoha cest, kterými se lze ubírat při zefektivňování procesu návrhu a ladění regulátorů v průmyslové praxi.

Hlavní část laboratoře je vytvořena přes webový framework ASP.NET Core a je tedy založena na HTML5, CSS, a JavaScriptu (dále jen JS). Využití ASP.NET Core navíc zajišťuje multiplatformitu celého projektu. Pro rozšíření praktického použití celého projektu je laboratoř obohacena o komunikaci se systémem REXYGEN. Dále je pak součástí projektu MySQL databáze a výpočetní server v programovacím jazyce Java. Tento server slouží pro integraci SW modulů a již existujících či pro tento projekt vytvořených výpočetních knihoven z prostředí Matlab a jazyka Java. Vývojem a implementací těchto technologií se zabýval Michal Vraštil, v jehož bakalářské práci jsou detailně popsány zmíněné techniky a postupy pro zprovoznění celé laboratoře [39].

Využití zmíněného serveru pro výpočet potřebných dat bylo již mým úkolem. Dále jsem vytvořil moduly pro výpočet modelu systému ve formě ohraničené oblasti komplexní roviny, která vyjadřuje neurčitost, přičemž byla využita teorie množinových modelů s třemi druhy charakteristických čísel. Tyto moduly a jejich metody jsou popsány v příloze na konci této práce. Při zpracování této problematiky jsem vycházel z několika prací [19, 32–35], které se daným tématem zcela nebo částečně zabývají. Stručné teoretické základy daných problematik jsou pak níže popsány a výsledky implementovány v laboratoři. Moduly (knihovny) byly vytvořeny v prostředí Matlab, kde byly také zkompileovány do „.jar“ souborů, které jsou pak serverem využity pro jednotlivé výpočty. V tomto serveru jsem pro tyto účely vytvořil několik metod spolu se zpracování příchozích požadavků. Kromě zmíněných modulů jsem vytvořil ještě několik knihoven pro funkcionality spojené se získáváním modelu a grafické uživatelské rozhraní (GUI) pro zadávání charakteristických čísel, výpočet neurčitosti a jednoduchý návrh regulátoru. Toto GUI se v laboratoři nachází pod záložkou „Uncertainty“. Všechny zdrojové soubory a moduly lze nalézt na CD, které je součástí přílohy.

4.1 Interaktivní vizualizace dat

Pro účely této práce bylo potřeba nalézt vhodný způsob vykreslení vypočtených oborů hodnot v komplexní rovině a dalších charakteristik, které jsou používány při návrhu regulátoru. Použitý způsob musel zároveň zaručovat interaktivitu jednotlivých

grafů, čímž je myšlena realizovatelnost určitých funkcí, které jsou elementární pro efektivní zadávání a práci s různými parametry. Těmito funkcemi jsou přiblížení/oddálení (Zoom in/Zoom out), posouvání středu vykreslovací plochy (Pan), „auto scale“ a také možnost přesouvat libovolné body v grafu (Drag able). Další podmínkou je také dobrá funkčnost i pro velké množství vykreslovaných dat.

Jelikož je cílem této práce vytvoření interaktivní virtuální on-line laboratoře, musela být vybrána technologie implementovatelná do webové stránky. Pro jednodušší a efektivnější tvorbu interaktivních grafů je vhodné využít existující JS grafické knihovny, které obsahují řadu pokročilých funkcionalit. Z dostupných knihoven [49] lze vybírat mezi dvěma hlavními směry:

- **Rastrová grafika** – obrázek je popsán pomocí hodnot jednotlivých barevných bodů (pixelů) – např. Chart.js, CanvasJS
- **Vektorová grafika** – obrázek je složenina základních, přesně definovaných, útvarů (bod, přímka, křivka a mnohoúhelník) – např. HighCharts, Plotly, D3.js

Vykreslovací technologie pro tyto směry, se kterými pracuje většina grafických JS knihoven, jsou HTML5 canvas a SVG. Je nutné podotknout, že tyto technologie jsou často používány souběžně. Canvas je element, jenž je součástí specifikace HTML5 a je používán k vykreslování rastrové grafiky a grafických primitiv. Pro vykreslování se používá JS. Oproti tomu SVG (neboli Scalable Vector Graphics) je jazyk pro popis dvoudimenzionální vektorové grafiky. Specifikace SVG obrázku s definicí chování je uložena v XML souboru. Díky svému vektorovému charakteru neztrácí, oproti canvasu, svou kvalitu při změně velikosti nebo přibližování (obr. 4.1).



Obr. 4.1: Rastrová vs. vektorová grafika

Pro vykreslování a manipulaci s větším množstvím často složitých obrazců je vhodnější využít vektorovou grafiku, namísto rastrové, která by byla vhodná při potřebě zpracování digitálních fotografií. Z široké škály existujících knihoven podporujících vektorovou grafiku bylo voleno mezi dvěma knihovnami, které se jeví jako optimální pro cíle této práce.

HighCharts¹

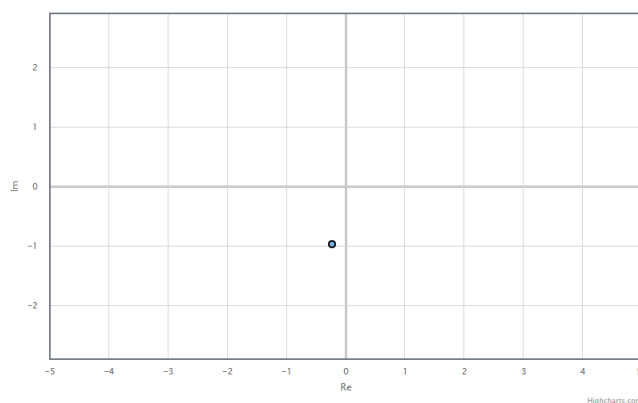
Jedná se o multiplatformní na SVG založenou grafickou knihovnu, která umožňuje vkládat různorodé interaktivní grafy do webové stránky. Grafy jsou též optimalizované i pro mobilní zařízení či obecně dotyková zařízení, což rozšiřuje možnosti využití této knihovny.

Plotly²

Plotly je společnost zabývající se technickými výpočty a komplexní vizualizací dat. Mezi hlavní produkty společnosti patří [57]

- **Plot.ly** – grafické uživatelské rozhraní pro tvorbu grafů z vložených dat
- **API knihovna** pro Python, MATLAB, Node.js, Arduino, . . .
- **Plotly.js** - JS knihovna pro analýzu dat, tvorbu grafů a strukturovaných uživatelských rozhraní

Z těchto dvou možností byla nakonec vybrána knihovna **HighCharts**, a to především díky vestavěné funkcionalitě umožňující libovolně přesouvat vykreslené objekty, což u ostatních uvažovaných knihoven nebylo vždy zaručeno, a silné vývojářské komunitě, která poskytuje užitečné rady pro implementaci rozličných grafů a funkcionalit. Mezi další nesporné výhody této knihovny patří modul Boost s podporou WebGL, umožňující vykreslovat tisíce nebo i miliony datových bodů v prohlížeči. Drobnou nevýhodou této knihovny je její proprietární licence. Pro nekomerční účely je ji však možné využívat bezplatně s Creative Commons (CC) Attribution-NonCommercial licencí. Text licenčních podmínek se nachází v hlavičce každého souboru, příslušného k této knihovně.



Obr. 4.2: Vykreslovací plocha pro grafy

¹HighCharts.com

²Plot.ly

4.2 JavaScriptové knihovny

4.2.1 Bootstrap³

Bootstrap je knihovna pro návrh webu a webových aplikací. Obsahuje šablony založené na HTML a CSS. Lze ji využít pro úpravu typografie, formulářů, tlačítek, navigace a dalších komponent rozhraní. Současně podporuje mobile-first přístup a je plně responzivní.[48] Knihovna podléhá MIT licenci.

K nejvýraznějším bootstrapem nabízeným prvkům, které jsou v této práci používány, patří responzivní, mobile-first „grid“ systém rozdělující libovolný prostor na dvanáct sloupců. Díky předdefinovaným třídám (tab. 4.1) je rozložení snadno realizovatelné a umožňuje tak částečně nahradit pozicování elementů na stránce. Výběr jednotlivých tříd závisí na velikosti okna prohlížeče. Rozložení lze tedy relativně snadno měnit v závislosti na velikosti zařízení, ať už se jedná o mobilní telefon, tablet či klasický monitor počítače.

Tab. 4.1: Tabulka předdefinovaných tříd Bootstrap Grid System

	Extra small	Small	Medium	Large	Extra large
Šířka okna	$< 576px$	$\geq 576px$	$\geq 768px$	$\geq 992px$	$\geq 1200px$
Třída	.col-xs-	.col-sm-	.col-md-	.col-lg-	.col-xl-

Dalším prvkem je element panel, který vytváří ohraničený kontejner (obr. 4.3). Lze tak snadno oddělit či naopak svázat relevantní prvky uživatelského rozhraní. I v takovýchto dílčích prvcích je možné použít „grid“ systém a docílit tak responzivity a snadného pozicování pro jednotlivé elementy.



Obr. 4.3: Ukázkový Bootstrap panel

³getbootstrap.com

4.2.2 MathJax⁴

Pro vykreslování přenosových funkcí a matematických symbolů je v této práci použita javascriptová knihovna MathJax. Knihovna po stažení prohledá stránku a nalezené vzorce vykreslí pomocí LaTeXu a MathML. Pro funkčnost knihovny je důležitá pouze podpora JS ve webovém prohlížeči, což umožňuje vykreslení vzorců i na mobilních zařízeních. MathJax je zcela multiplatformní a otevřený software, který je distribuován pod Apache, v2.0 licenci.

4.2.3 Ostatní

Dalšími použitými knihovnami a pluginy byly

- **jQuery**
 - **jQuery toggle**⁵ - vykreslení zaškrtačacího políčka jako přepínacího tlačítka,
 - **jQuery loading**⁶ - zobrazení načítací obrazovky při načítání dat ze serveru,
- **Font Awesome Icons**⁷ - ikony pro uživatelské rozhraní.

4.3 Parametrizace extrémálních funkcí

Jak bylo popsáno v úvodu této kapitoly, hlavní funkční částí laboratoře je výpočet oboru hodnot množinového modelu. Pro nalezení hranice oboru hodnot pro jednotlivé frekvence je potřeba nalézt parametrizaci všech extrémálních přenosových funkcí. Následující podkapitoly vychází z několika prací dlouhodobě budovaných na Katedře kybernetiky Fakulty aplikovaných věd Západočeské univerzity v Plzni.[19, 32, 34, 35]

4.3.1 Neceločíselný řád systému (FO)

Nechť třída apriorně přípustných systémů obsahuje všechny FO přenosové funkce ve tvaru

$$P(s) = \frac{K}{\left(\prod_{i=1}^{p-1} (\tau_i s + 1)^{n_i}\right) s^{n_p}}, \quad (4.1)$$

kde p je libovolné celé číslo a K , τ_i , $i = 1, 2, \dots, p-1$, n_i , $i = 1, 2, \dots, p$ jsou kladná reálná čísla. Množina všech přípustných přenosových funkcí se nazývá množinový model a označuje se $\mathcal{S}^{n,m}(\Pi)$, kde n je maximální řád procesu, m minimální stupeň pólu a

⁴MathJax.com

⁵[jQuery toggle](#)

⁶[jQuery loading](#)

⁷[Font Awesome Icons](#)

Π množina interpolačních podmínek. Obecnou množinu interpolačních podmínek pro frekvenční charakteristiku řízeného procesu $P(s)$ lze zapsat jako

$$\Pi = \{P^{(l)}(s)|_{s=j\omega_i} = P_{il}\}, \quad (4.2)$$

kde $i = 1, 2, \dots, k$, $l = 0, 1, \dots, m_i$; k a m_i jsou daná celá čísla a $P_{il} \in \mathbb{C}$. Jsou tedy dány body frekvenční charakteristiky pro množinu frekvencí $\Omega = \{\omega_1, \omega_2, \dots, \omega_k\}$ a v každém bodě ω_i je známo prvních m_i derivací frekvenční charakteristiky. V této práci jsou použity dva speciální případy obecné interpolace frekvenční charakteristiky, a to jeden bod frekvenční charakteristiky a první tři momenty impulsní funkce.

Klíčovým pojmem je obor hodnot množinového modelu. Což je množina ve tvaru

$$\mathcal{V}_\omega^{n,m}(\Pi) = \{P(j\omega) : P(s) \in \mathcal{V}^{n,m}(\Pi)\}. \quad (4.3)$$

Tato množina, jak již bylo zmíněno, představuje ve frekvenční oblasti neurčitost množinového modelu pro danou frekvenci. Obor hodnot je tedy jakýmsi zobecněným bodem frekvenční charakteristiky, který pro každou frekvenci vytváří všechny apriorně přípustné systémy konzistentní s naměřenými daty. Body na hranici oboru hodnot jsou pak generovány tzv. extrémálními přenosovými funkcemi, což je přípustná přenosová funkce $P(s) \in \mathcal{S}^{n,m}(\Pi)$ pro kterou existuje $\omega \notin \{\omega_1, \omega_2, \dots, \omega_k\}$ taková, že $P(j\omega) \in \partial\mathcal{V}_\omega^{n,m}(\Pi)$, kde $\partial\mathcal{V}_\omega^{n,m}(\Pi)$ označuje hranici oboru hodnot v komplexní rovině.

Jeden bod frekvenční charakteristiky

Je předpokládána znalost jednoho bodu frekvenční charakteristiky, což je speciální případ obecné interpolace frekvenční charakteristiky, tedy

$$\Pi = \{P(j\omega_1) = p_1 = re^{-j\varphi}\}. \quad (4.4)$$

Množinový model je ve tvaru $\mathcal{S}^{n,m}(p, \omega)$. Pro interpolační podmínku ve tvaru (4.4) je možné najít parametrizaci všech extrémálních přenosových funkcí a znázornit tak hranici oboru hodnot v komplexní rovině pro každou frekvenci $\omega > 0$.

Pro charakteristická čísla musí platit $r > 0$, $0 < \varphi < 2\pi$, $\omega_1 > 0$ a $\varphi \leq (n-m)\frac{\pi}{2}$. Potom je $\mathcal{S}^{n,m}(p_1, \omega_1) \neq \emptyset$ a přípustná přenosová funkce $P \in \mathcal{S}^{n,m}(\Pi)$ je extrémální právě tehdy, jestliže může být vyjádřena v jednom z následujících tvarů.

(i)

$$P_1(s, \alpha) = \frac{K(\alpha)}{(\tau_1(\alpha)s + 1)^{n_1(\alpha)}}, \quad (4.5)$$

kde

$$\tau_1(\alpha) = \frac{1}{\omega_1} \tan \alpha,$$

$$n_1(\alpha) = \frac{\varphi}{\alpha},$$

$$K(\alpha) = r\sqrt{(\tau_1^2(\alpha)\omega_1^2 + 1)^{n_1(\alpha)}}$$

a α probíhá na intervalu

$$I_1 = \left[\frac{\varphi}{n}, \min \left\{ \frac{\pi}{2}, \frac{\varphi}{m} \right\} \right].$$

(ii)

$$P_2(s, \alpha) = \frac{K(\alpha)}{(\tau_1(\alpha)s + 1)^{n-m}(\tau_2(\alpha)s + 1)^m}, \quad (4.6)$$

kde

$$\tau_1(\alpha) = \frac{1}{\omega_1} \tan \alpha,$$

$$\tau_2(\alpha) = \frac{1}{\omega_1} \tan \frac{\varphi - (n-m)\alpha}{m},$$

$$K(\alpha) = r\sqrt{(\tau_1^2(\alpha)\omega_1^2 + 1)^{n-m}(\tau_2^2(\alpha)\omega_1^2 + 1)^m}$$

a α probíhá na intervalu

$$I_2 = \left[\max \left\{ 0, \frac{\varphi - m\frac{\pi}{2}}{n-m} \right\}, \frac{\varphi}{n} \right].$$

(iii)

$$P_3(s, \alpha) = \frac{K(\alpha)}{(\tau_1(\alpha)s + 1)^{n-n_2(\alpha)}s^{n_2(\alpha)}}, \quad (4.7)$$

kde

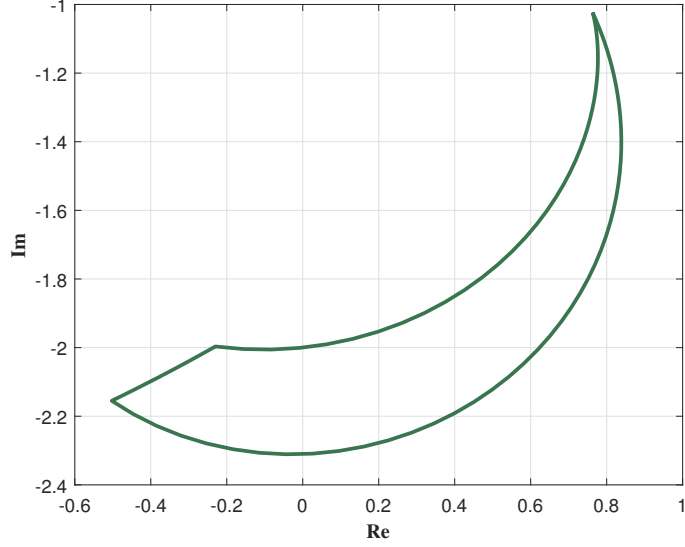
$$\tau_1(\alpha) = \frac{1}{\omega_1} \tan \alpha,$$

$$n_2(\alpha) = \frac{\varphi - n\alpha}{\frac{\pi}{2} - \alpha},$$

$$K(\alpha) = r\sqrt{(\tau_1^2(\alpha)\omega_1^2 + 1)^{n-n_2(\alpha)} \omega_1^2}$$

a α probíhá na intervalu

$$I_3 = \left[0, \frac{\varphi - m\frac{\pi}{2}}{n-m} \right].$$



Obr. 4.4: Obor hodnot množinového modelu pro $r = 1$, $\varphi = 1.8$, $\omega_1 = 1$, $n = 5$, $m = 1$, $\omega = 0.5$

Momenty impulsní funkce

Při znalosti prvních třech momentů impulsní funkce m_0, m_1, m_2 , ze kterých je odvozena vhodnější trojice čísel κ, μ, σ^2 , se opět jedná o speciální případ obecné interpolace frekvenční charakteristiky, kdy je znám bod na frekvenci $\omega = 0$ a první dvě derivace frekvenční charakteristiky v tomto bodě, tedy

$$\mathbb{I} = \{P(0), P'(0), P''(0)\}. \quad (4.8)$$

Množinový model se poté označuje za momentový a značí se $\mathcal{S}^{n,m}(\kappa, \mu, \sigma^2)$. Dále musí platit $n \geq 2m$, $m > 0$, potom pro libovolnou frekvenci $\omega > 0$ může být každý bod hranice $\partial\mathcal{V}_\omega(\mathbb{I})$ oboru hodnot množinového modelu $\mathcal{S}^{n,m}(\kappa, \mu, \sigma^2)$ generován přenosem, který lze vyjádřit ve tvaru

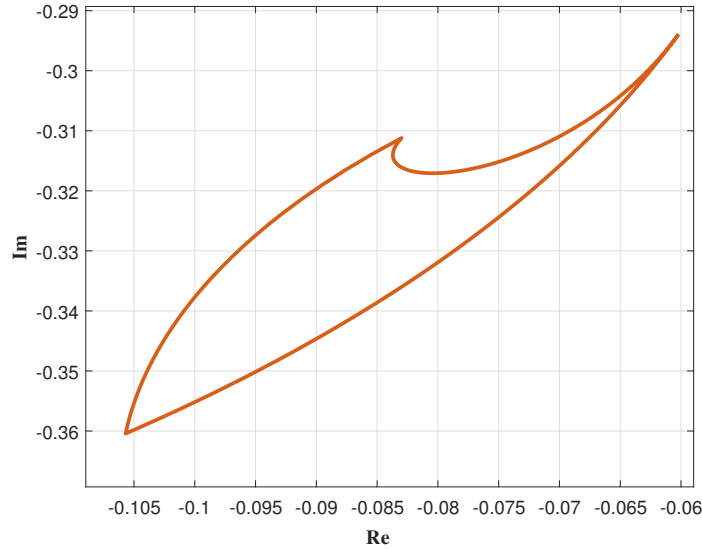
$$P(s) = \frac{\kappa}{(\tau_1(\alpha)s + 1)^{n_1(\alpha)} (\tau_2(\alpha)s + 1)^{n_2(\alpha)}}, \quad (4.9)$$

kde

$$\tau_1(\alpha) = \frac{\mu - \sqrt{\frac{n_2}{n_1}(\sigma^2(n_1 + n_2) - \mu^2)}}{(n_1 + n_2)}, \quad \tau_2(\alpha) = \frac{\mu + \sqrt{\frac{n_1}{n_2}(\sigma^2(n_1 + n_2) - \mu^2)}}{(n_1 + n_2)}$$

a n_1, n_2 jsou určeny třemi různými způsoby:

- (i) $n_1(\alpha) = m, n_2(\alpha) = \alpha, \alpha \in \left[\max \left\{ m, \frac{\mu^2 - m\sigma^2}{\sigma^2} \right\}, \min \left\{ n - m, \frac{\mu^2}{\sigma^2} \right\} \right]$
- (ii) $n_1(\alpha) = \alpha, n_2(\alpha) = m, \alpha \in \left[\max \left\{ m, \frac{\mu^2 - m\sigma^2}{\sigma^2} \right\}, n - m \right]$
- (iii) $n_1(\alpha) = n - \alpha, n_2(\alpha) = \alpha, \alpha \in \left[m, \min \left\{ n - m, \frac{\mu^2}{\sigma^2} \right\} \right]$



Obr. 4.5: Obor hodnot množinového modelu pro $\kappa = 1$, $\mu = 1$, $\sigma_1^2 = 0.6$, $n = 5$, $m = 1$, $\omega = 3$

4.3.2 Celočíslný řád systému (IO)

Pro celočíselný řád je kladen předpoklad, že řízený systém lze s dostatečnou přesností popsat přenosem $F(s)$ ve tvaru

$$F(s) = \frac{K_0}{p(s)}, \quad (4.10)$$

kde $K_0 > 0$ a $p(s)$ je polynom stupně maximálně n s reálnými nezápornými koeficienty. Symbolem \mathcal{S} je pak označena množina všech lineárních časově invariantních systémů s přenosem ve tvaru (4.10). Podmnožina množiny \mathcal{S} , pro kterou je n maximální stupeň polynomu $p(s)$ a kde všechny kořeny tohoto polynomu náležejí množině $\mathbb{R}^- = (-\infty, 0)$, se značí \mathcal{S}^n .

Jeden bod frekvenční charakteristiky

Opět je tedy předpokládána znalost jednoho bodu frekvenční charakteristiky. Množina všech přenosů pro něž platí $F(s) \in \mathcal{S}^n$ a které navíc splňují podmínky

$$F(j\omega_i) = F_i, \quad (4.11)$$

$$-2\pi < \arg F \leq 0, \quad (4.12)$$

kde $i = 1, 2, \dots, k$ je pak nazývána množinovým modelem a označuje se $\mathcal{S}^n(\langle F_i, \omega_i \rangle_{i=1}^k)$, kde $F_i \in \mathbb{C}$ a $0 \leq \omega_1 < \dots < \omega_k$ a obsahuje všechny přípustné systémy stupně maximálně n . Obor hodnot množinového modelu $\mathcal{S}^n(\langle F_i, \omega_i \rangle_{i=1}^k)$ v bodě $s = j\omega$, $\omega > 0$

je definován jako

$$\mathcal{V}^n(\langle F_i, \omega_i \rangle_{i=1}^k; \omega) \triangleq \left\{ F(j\omega) : F(s) \in \mathcal{S}^n \left(\langle F_i, \omega_i \rangle_{i=1}^k \right) \right\}. \quad (4.13)$$

Symbolem $\partial\mathcal{V}^n(\langle F_i, \omega_i \rangle_{i=1}^k; \omega)$ se označuje hranice neboli množina všech krajních bodů oboru hodnot $\mathcal{V}^n(\langle F_i, \omega_i \rangle_{i=1}^k; \omega)$. Přípustný přenos $F(s) \in \mathcal{S}^n(\langle F_i, \omega_i \rangle_{i=1}^k)$ je extrémním přenosem pokud existuje alespoň jedna frekvence $\omega > 0$ taková, že $F(j\omega) \in \partial\mathcal{V}^n(\langle F_i, \omega_i \rangle_{i=1}^k; \omega)$.

Nechť $q \triangleq \left\lfloor \frac{2\varphi}{\pi} \right\rfloor$, potom množina $\mathcal{V}^n(re^{-j\varphi}, \omega_1; \omega)$, kde $\varphi \in (0, 2\pi)$, $\omega > 0$, $\omega \neq \omega_1$, je křivouhelník v komplexní rovině s n nebo $n + 1$ stranami, které jsou určeny následovně.

(i) Je-li $q = 0$, potom $\mathcal{V}^n(re^{-j\varphi}, \omega_1; \omega)$ má n stran určených křivkami

$$H_{(k,l,0)}(s, \alpha) = \frac{K_{kl}(\alpha)}{[\tau_{kl0}(\alpha)s + 1]^k [\vartheta_{kl0}(\alpha)s + 1]^l}, \quad (4.14)$$

kde uspořádaná trojice indexů (k, l, m) probíhá n -člennou posloupnost

$$(1, n-1, 0), (1, n-2, 0), \dots, (1, 1, 0), (n-1, 1, 0),$$

a kde

$$\begin{aligned} \tau_{kl0}(\alpha) &= \frac{1}{\omega_1} \tan \alpha, \\ \vartheta_{kl0}(\alpha) &= \frac{1}{\omega_1} \tan \frac{\varphi - k\alpha}{l}, \\ K_{kl0}(\alpha) &= A \left[\tau_{kl0}^2(\alpha) \omega_1^2 + 1 \right]^{\frac{k}{2}} \left[\vartheta_{kl0}^2(\alpha) \omega_1^2 + 1 \right]^{\frac{l}{2}} \end{aligned}$$

a

$$\alpha \in \left(0, \frac{\varphi}{k+l} \right).$$

(ii) Je-li $q > 0$, potom $\mathcal{V}^n(re^{-j\varphi}, \omega_1; \omega)$ má $n + 1$ stran určených křivkami

$$H_{(k,l,m)}(s, \alpha) = \frac{K_{klm}(\alpha)}{[\tau_{klm}(\alpha)s + 1]^k [\vartheta_{klm}(\alpha)s + 1]^l s^m}, \quad (4.15)$$

kde uspořádaná trojice indexů (k, l, m) probíhá $n+1$ -člennou posloupnost

$$\begin{aligned} &(1, n-1, 0), (1, n-2, 0), \dots, (1, q, 0), \\ &(n-q-1, 1, q), (n-q, 1, q-1), \dots, (n-1, 1, 0) \end{aligned}$$

a kde

$$\tau_{klm}(\alpha) = \frac{1}{\omega_1} \tan \alpha,$$

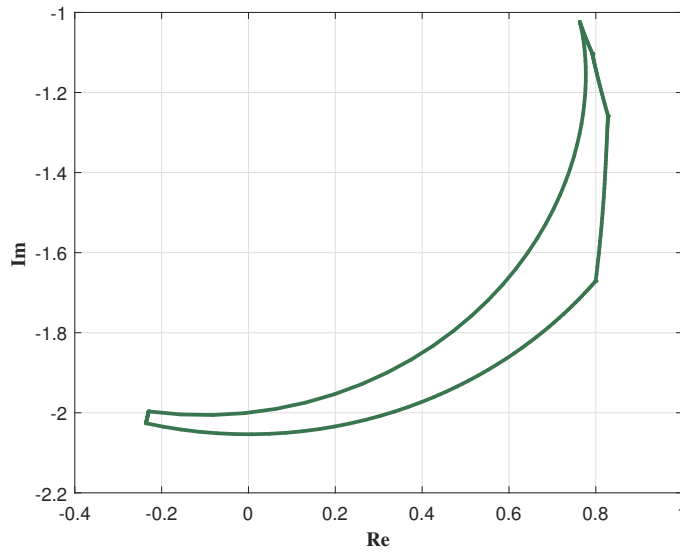
$$\vartheta_{klm}(\alpha) = \frac{1}{\omega_1} \tan \frac{\varphi - k\alpha - m\frac{\pi}{2}}{l},$$

$$K_{klm}(\alpha) = A \left[\tau_{klm}^2(\alpha)\omega_1^2 + 1 \right]^{\frac{k}{2}} \left[\vartheta_{klm}^2(\alpha)\omega_1^2 + 1 \right]^{\frac{l}{2}} \omega^m$$

a

$$\alpha \in \left(0, \frac{\varphi - m\frac{\pi}{2}}{k+l} \right) \quad \text{pro } \varphi - (m+l)\frac{\pi}{2} < 0$$

$$\alpha \in \left(\frac{\varphi - (m+l)\frac{\pi}{2}}{k}, \frac{\varphi - m\frac{\pi}{2}}{k+l} \right) \quad \text{pro } \varphi - (m+l)\frac{\pi}{2} \geq 0$$



Obr. 4.6: Obor hodnot množinového modelu pro $r = 1$, $\varphi = 1.8$, $\omega_1 = 1$, $n = 5$, $\omega = 0.5$

Momenty impulsní funkce

Je předpokládána znalost trojice čísel κ, μ, σ^2 . Množina všech přenosů pro něž platí $F(s) \in \mathcal{S}^n$ a které splňují následující momentové podmínky

$$\frac{F^{(i)}(0)}{i!} = f_i, \quad i = 0, 1, 2, \quad (4.16)$$

kde lze reálná čísla f_0, f_1, f_2 nahradit trojicí čísel κ, μ, σ^2 , je opět nazývána množinovým modelem a označuje se $\mathcal{S}^n(\kappa, \mu, \sigma^2)$. Obor hodnot tohoto množinového modelu pro frekvenci ω je množina ve tvaru

$$\mathcal{V}^n(\kappa, \mu, \sigma^2; \omega) \triangleq \{F(j\omega) : F(s) \in \mathcal{S}^n(\kappa, \mu, \sigma^2)\}. \quad (4.17)$$

Symbollem $\partial\mathcal{V}^n(\kappa, \mu, \sigma^2; \omega)$ je opět označena hranice neboli množina všech krajních bodů oboru hodnot $\mathcal{V}^n(\kappa, \mu, \sigma^2; \omega)$. Přípustný přenos $F(s) \in \mathcal{S}^n(\kappa, \mu, \sigma^2)$ je extrémním přenosem pokud existuje alespoň jedna frekvence $\omega > 0$ taková, že $F(j\omega) \in$

$\partial\mathcal{V}^n(\kappa, \mu, \sigma^2; \omega)$.

Nechť $q \triangleq \lfloor \frac{\mu^2}{\sigma^2} + 1 \rfloor$, potom množina $\mathcal{V}^n(\kappa, \mu, \sigma^2; \omega)$ s $\omega > 0$ je křivoúhelník v komplexní rovině s $n - 1$ nebo n stranami generovanými přenosem, který lze vyjádřit ve tvaru

$$H_{(k,l,m)}(s, \alpha) = \frac{\kappa}{[\tau_{1_{klm}}(\alpha)s + 1]^k [\tau_{2_{klm}}(\alpha)s + 1]^l [\tau_{3_{klm}}(\alpha)s + 1]^m}. \quad (4.18)$$

Pro uspořádanou trojici indexů (k, l, m) platí:

Pokud je $q = 2$, pak je posloupnost $(n-1)$ -členná

$$(1, 1, 1), (1, 2, 1), \dots, (1, n - 2, 1), \quad (4.19)$$

$$(n - 2, 1, 1). \quad (4.20)$$

Je-li $2 < q < n$, pak je posloupnost n -členná

$$(1, q - 1, 1), (1, q, 1), \dots, (1, n - 2, 1), \quad (4.21)$$

$$(n - 2, 1, 1), (n - 3, 1, 2), \dots, (n - q + 1, 1, k - 2), \quad (4.22)$$

$$(n - q, 1, q - 1), \quad (4.23)$$

$$(1, q - 2, 1) \quad (4.24)$$

a pokud je $q = n$, je posloupnost opět $(n-1)$ -členná

$$(n - 2, 1, 1), (n - 3, 1, 2), \dots, (1, 1, n - 2), \quad (4.25)$$

$$(1, n - 2, 1). \quad (4.26)$$

Parametry $\tau_{i_{klm}}(\alpha)$, $i = 1, 2, 3$ jsou dány vztahy

$$\tau_{1_{klm}}(\alpha) = \alpha,$$

$$\tau_{2_{klm}}(\alpha) = \frac{\mu - k\tau_{1_{klm}}(\alpha)}{l + m} - \sqrt{\frac{m}{l}} \cdot \frac{\sqrt{\sigma^2(l + m) - (1 - k\tau_{1_{klm}}(\alpha))^2 - k(l + m)\tau_{1_{klm}}^2(\alpha)}}{l + m},$$

$$\tau_{3_{klm}}(\alpha) = \frac{\mu - k\tau_{1_{klm}}(\alpha)}{l + m} - \sqrt{\frac{l}{m}} \cdot \frac{\sqrt{\sigma^2(l + m) - (1 - k\tau_{1_{klm}}(\alpha))^2 - k(l + m)\tau_{1_{klm}}^2(\alpha)}}{l + m},$$

kde α probíhá na intervalu $I = \langle a, b \rangle$, jehož krajní body jsou závislé na uspořádané trojici indexů (k, l, m) .

Pro (4.19) a (4.21) platí, že

$$a = 0,$$

$$b = \frac{\mu}{k+l+m} - \frac{\sqrt{m}}{\sqrt{l+m}} \cdot \frac{\sqrt{\sigma^2(k+l+m) - \mu^2}}{k+l+m}.$$

Je-li trojice ve tvaru (4.20), (4.22), (4.25), (4.24) či (4.26), pak

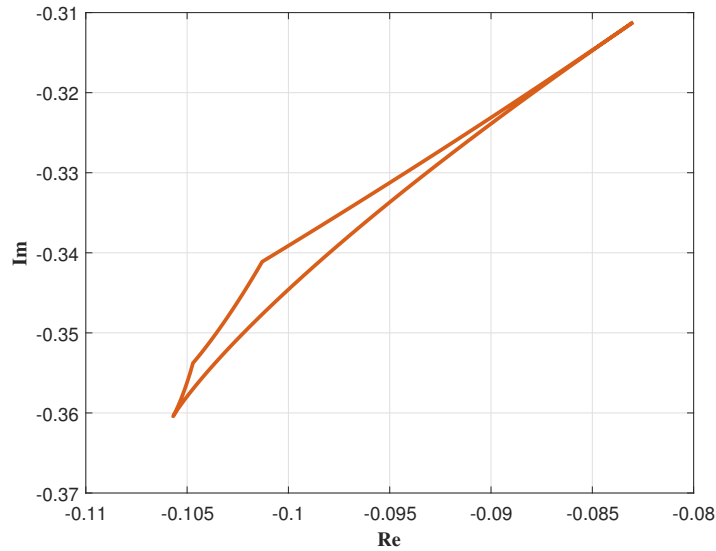
$$a = \frac{\mu}{k+l+m} - \frac{\sqrt{l+m}}{\sqrt{k}} \cdot \frac{\sqrt{\sigma^2(k+l+m) - \mu^2}}{k+l+m},$$

$$b = \frac{\mu}{k+l+m} - \frac{\sqrt{m}}{\sqrt{k+l}} \cdot \frac{\sqrt{\sigma^2(k+l+m) - \mu^2}}{k+l+m}$$

a nakonec pokud má trojice tvar (4.23), platí že

$$a = 0,$$

$$b = \frac{\mu}{k+l+m} - \frac{\sqrt{m}}{\sqrt{k+l}} \cdot \frac{\sqrt{\sigma^2(k+l+m) - \mu^2}}{k+l+m}.$$

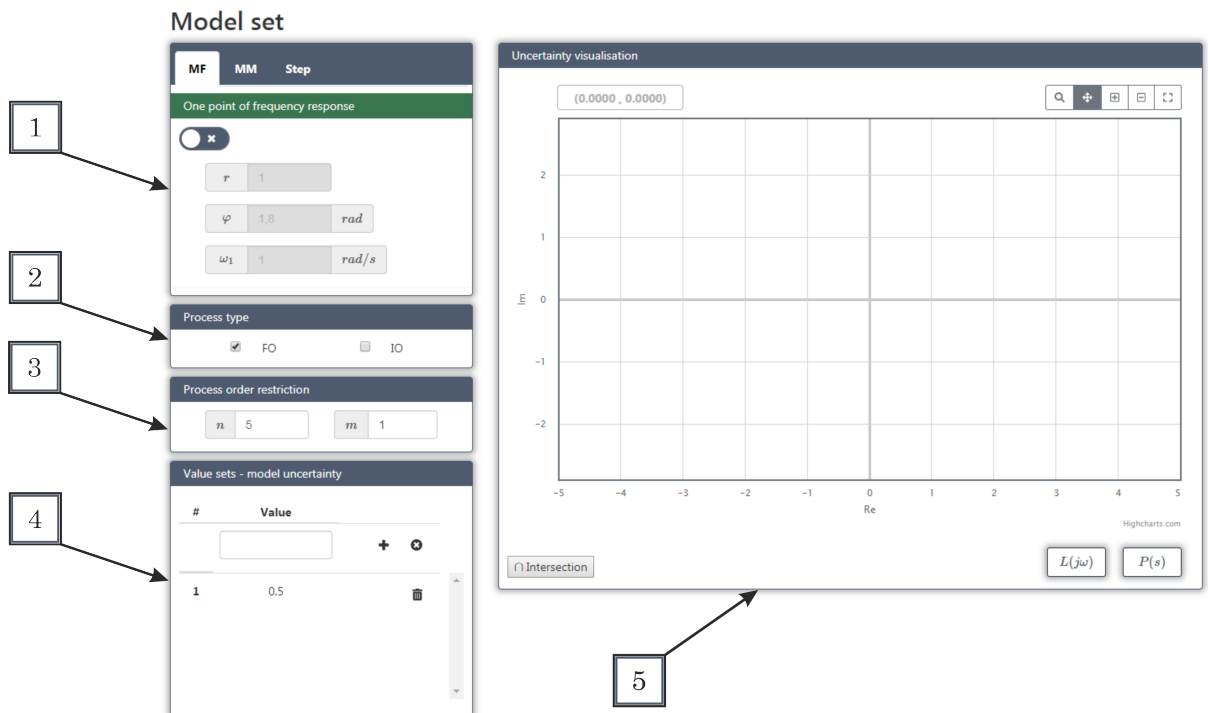


Obr. 4.7: Obor hodnot množinového modelu pro $\kappa = 1$, $\mu = 1$, $\sigma_1^2 = 0.6$, $n = 5$, $\omega = 3$

4.4 Základní uživatelské rozhraní a funkcionality

Po načtení obsahu stránky lze jako první spatřit řadu panelů a prázdnou vykreslovací plochu grafu (obr.4.8). Pro přehlednost a responzivitu je stránka strukturována do tří sloupců rozdílné šířky, kdy v levém a pravém sloupci⁸ jsou panely se vstupními poli pro vkládání parametrů potřebných pro zvolenou funkcionality. V prostředním sloupci jsou pak, v závislosti na vybraném typu charakteristických čísel a právě používané funkcionality, jeden až dva panely s grafy.

⁸Pravý sloupec je zobrazen pouze pro specifické funkcionality.



Obr. 4.8: Úvodní rozložení uživatelského rozhraní

Pro vykreslení oboru hodnot množinového modelu, který v komplexní rovině představuje jeho neurčitost, lze volit mezi třemi druhy charakteristických čísel, kterými je specifikován množinový model:

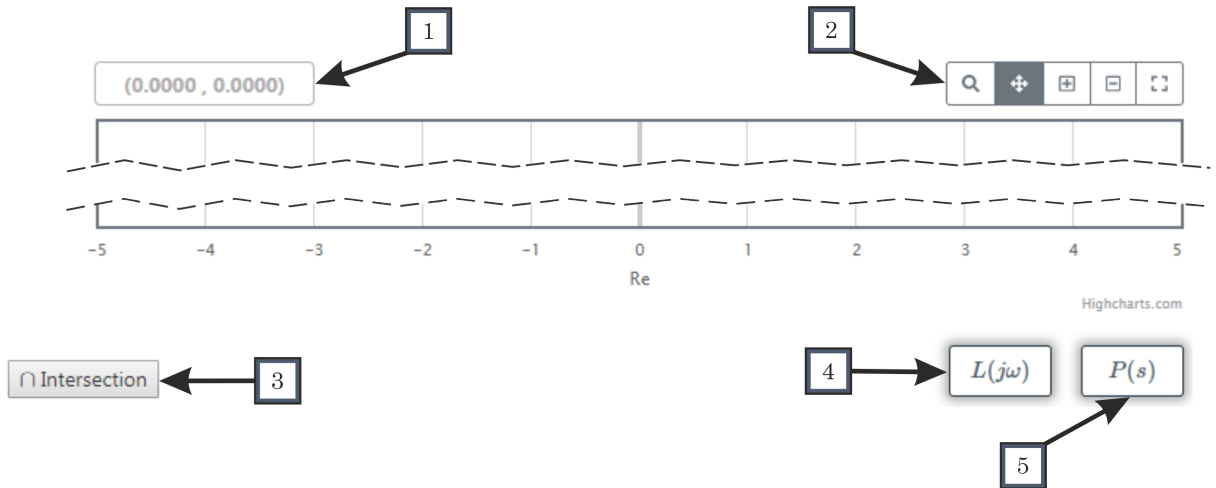
- Jeden bod frekvenční charakteristiky - MF
- Momenty impulsní funkce - MM
- Ohraničení přechodové charakteristiky ve formě uzlových bodů - Step

Zvolený druh lze měnit v menu v navigačním panelu (obr. 4.8-1).

Pro jeden bod frekvenční charakteristiky a momenty impulsní funkce jsou funkcionality a tedy i panely ve všech sloupcích shodné. V navigačním panelu (obr. 4.8-1) se nachází posuvné tlačítko, které umožňuje vykreslování oborů hodnot pro daná charakteristická čísla. Po povolení vykreslování je teprve možné nastavit parametry specifikující bod frekvenční charakteristiky r, φ, ω_1 či trojici čísel κ, μ, σ^2 a získat tak obor hodnot pro dané frekvence ω . Určení daných oblastí je provedeno na výpočetním serveru, kam byly odeslány zvolená charakteristická čísla a frekvence. Dalším panelem je „Process type“ (obr. 4.8-2), kde lze volit řád systému, a to buď neceločíselný (FO), celočíselný (IO) či kombinaci obou dvou. Panel „Process order restriction“ (obr. 4.8-3) zase slouží pro zadávání hodnoty maximálního řádu procesu (n) a minimálního

stupně pólu (m). V posledním panelu levého sloupce „Value sets – model uncertainty“ (obr. 4.8-4) je vstupní pole a tabulka pro zadávání a výpis frekvencí ω . Prostřední panel „Uncertainty visualisation“ (obr. 4.8-5) obsahuje vykreslovací plochu s komplexní rovinou a sadu ovládacích prvků (obr. 4.9):

1. Pole pro výpis polohy kurzoru v grafu
2. Nástroje pro práci s grafy
 - Přiblížení ohraničené oblasti (Zoom)
 - Posun středu vykreslovací plochy (Pan)
 - Přiblížení (Zoom in)
 - Oddálení (Zoom out)
 - Autoscale
3. Zobrazení kombinace charakteristických čísel
4. Přepnutí do režimu otevřené regulační smyčky
5. Režim zobrazení extrémálních přenosových funkcí



Obr. 4.9: Ovládací prvky uživatelského rozhraní

4.4.1 Kombinace jednoho bodu frekvenční charakteristiky a momentů impulsní funkce

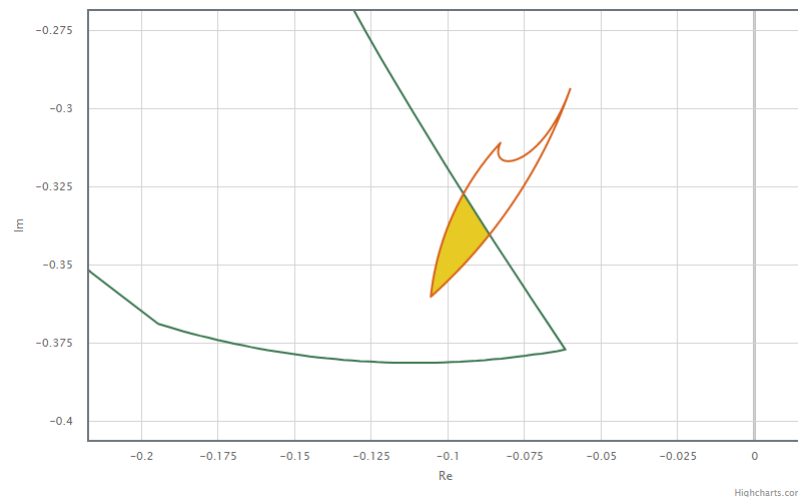
Při navrhování regulátoru se lze setkat s problémem, kdy velká neurčitost množinového modelu znemožní návrh dostatečně „agresivního“ regulátoru. Aby bylo možné neurčitost snížit, je potřeba znát vlastnosti procesu na více frekvencích. Toho je docíleno provedením několika nezávislých identifikačních experimentů. Díky znalosti hranice oborů hodnot je možné vypočítat i průnik těchto oblastí a určit tak horní odhad oboru hodnot.

Pro tento případ, tedy kombinace jednoho bodu frekvenční charakteristiky a momentů impulsní funkce lze interpolační podmínku zapsat ve tvaru

$$\Pi = \{P(j\omega_1); P(0), P'(0), P''(0)\} = \{\Pi_1; \Pi_2\}. \quad (4.27)$$

Pro (4.27) je množinový model neprázdný právě tehdy, když $P(j\omega_1) \in \mathcal{V}_{\omega_1}^{n,m}(\Pi_2)$ pro neceločíselný řád systému a když $P(j\omega_1) \in \mathcal{V}^n$, který splňuje množinu interpolačních podmínek Π_2 pro celočíselný řád systému. Ukázka průniku oborů hodnot je na obr. 4.10.

Kombinace jednoho bodu frekvenční charakteristiky a momentů impulsní funkce je považována za nejperspektivnější, protože momenty impulsní funkce nesou informaci o chování systému na frekvencích blízkých nule. Na vyšších frekvencích neurčitost modelu roste, a proto je žádoucí zde změřit ještě bod frekvenční charakteristiky.[19]



Obr. 4.10: Ukázka průniku oborů hodnot

Realizace

Pro zobrazení průniku oborů hodnot v komplexní rovině musí být povoleno vykreslování jak pro bod frekvenční charakteristiky, tak i pro momenty. Pokud je tato

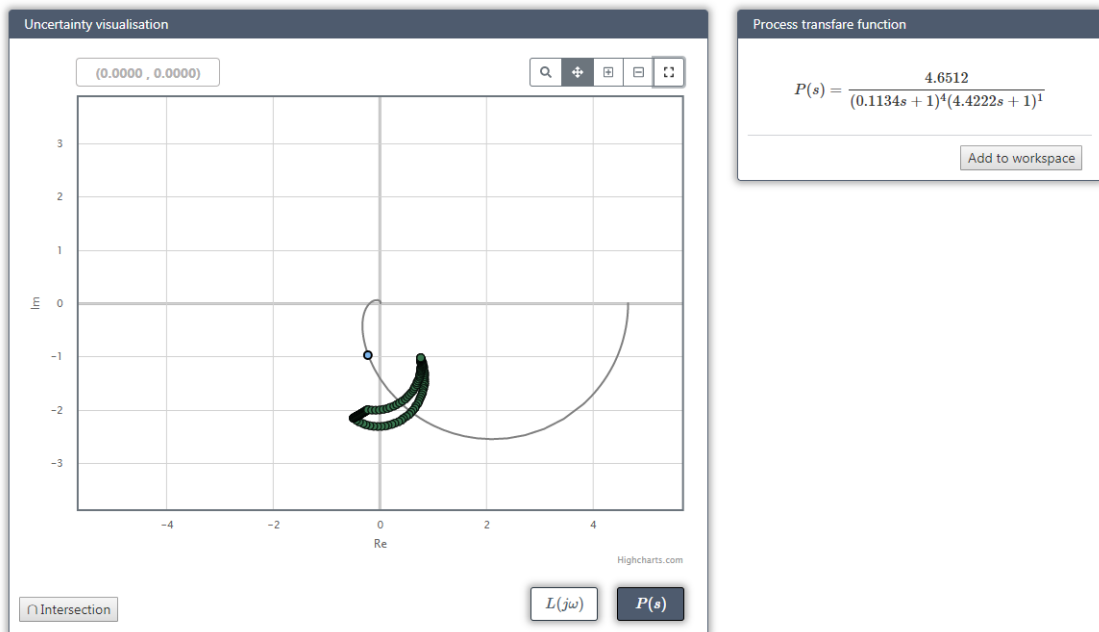
podmínka splněna lze stisknout tlačítko „Intersection“ (obr. 4.9-3) a zobrazit tak průniky. Daná oblast zájmu je poté získána následujícím postupem.

Body hranice oboru hodnot množinového modelu pro oba druhy charakteristických čísel jsou určeny parametrizací extrémálních přenosových funkcí. Parametry potřebné pro výpočet této hranice jsou odeslány na výpočetní server, kde se dané hranice vygenerují a poté se pomocí vytvořeného modulu vypočtou průniky daných oblastí. Body z obou hranic oboru hodnot ohraničující danou oblast průniku jsou pak optimálně seřazeny, odeslány zpět do laboratoře a následně i zobrazeny.

4.4.2 Zobrazení extrémálních přenosových funkcí

Při stisku tlačítka „P(s)“ (obr. 4.9-5) se na křivkách představujících hranice oboru hodnot v komplexní rovině zobrazí body realizující dané ohraničení. Každý z těchto bodů odpovídá jedinečné extrémální přenosové funkci, která se při kliknutí na konkrétní bod vypíše v panelu, který se pro tento účel zobrazil v pravém sloupci stránky. Ukázkový příklad lze vidět na obr. 4.11.

Hodnoty přenosové funkce jsou opět získány z výpočetního serveru, kam byla odeslána informace o typu charakteristických čísel (MF/MM), parametry specifikující daný obor hodnot a poloha zvoleného bodu v komplexní rovině. Po zpracování pošle server zpět parametry přenosu, které jsou následně zobrazeny ve specifickém tvaru, který je určen na základě druhu charakteristických čísel (MF/MM) a číselného řádu procesu (FO/IO). Zobrazení přenosové funkce je provedeno pomocí knihovny MathJax, čímž je docíleno velice kvalitního vykreslení a stylového provedení.



Obr. 4.11: Panely pro zobrazení přenosových funkcí

Pokud je potřeba se zobrazenou přenosovou funkcí dále pracovat, lze ji přes tlačítko „Add to workspace“ uložit do databáze k ostatním přenosům. Současně s ukládáním se do grafu s komplexní rovinou vykreslí Nyquistova křivka, která je získána parametrizací frekvenčního přenosu $P(j\omega)$ daného procesu pro $\omega \in \langle 0, \infty \rangle$.

4.4.3 Režim otevřené regulační smyčky

Pro jednoduchý návrh PI regulátoru se lze přepnout, přes tlačítko „ $L(j\omega)$ “ (obr. 4.9-4), do režimu otevřené regulační smyčky. Po stisknutí tlačítka se v prostředním sloupci zobrazí nový panel s grafem (obr. 4.12-3), který je určený pro vykreslování citlivostních funkcí. V panelu jsou spolu s grafem i tlačítka pro přepínání mezi jednotlivými funkcemi:

- S – citlivostní funkce
- T – komplementární citlivostní funkce
- S_i – citlivostní funkce poruchy na vstupu
- S_u – citlivostní funkce řízení

V pravém sloupci stránky se zobrazí také nové panely. V prvním panelu „M-circles“ (obr. 4.12-1) jsou dvě vstupní pole pro zadávání rezonančního převýšení citlivostní funkce (M_S) a komplementární citlivostní funkce (M_T). Druhý panel „Parametric plane - PI Controller“ (obr. 4.12-2) obsahuje parametrickou rovinu a vstupní pole pro specifikaci parametrů K a K_i PI regulátoru, který lze popsat frekvenčním přenosem ve tvaru

$$C(j\omega) = K - j\frac{K_i}{\omega}. \quad (4.28)$$

V režimu otevřené regulační smyčky jsou všechny přenosy zobrazených frekvenčních charakteristik vynásobeny přenosem PI regulátoru⁹ a jsou tak získány charakteristiky otevřené regulační smyčky, pro kterou tedy platí

$$L(j\omega) = C(j\omega)P(j\omega). \quad (4.29)$$

Frekvenční charakteristika procesu $P(j\omega)$ je na konečné množině frekvencí z intervalu $\langle 0, \infty \rangle$ tvořena body v komplexní rovině. Díky této skutečnosti lze přenos procesu pro nějakou konstantní hodnotu ω vyjádřit ve tvaru $P(j\omega) = A_P + jB_P$. Výpočet otevřené regulační smyčky je poté možné zapsat jako

$$L(j\omega) = \left(K A_P + \frac{K_i B_P}{\omega} \right) + j \left(K B_P - \frac{A_P K_i}{\omega} \right), \quad (4.30)$$

⁹Což platí i pro charakteristiky extrémální přenosové funkce ohraničující obor hodnot množinového modelu pro danou frekvenci ω .

kde je přenos rozdělen na reálnou a imaginární část.

Spolu s frekvenčními charakteristikami pro otevřenou regulační smyčku je v hlavním grafu zobrazena i jednotková kružnice a M-kružnice, které jsou specifikované zadaným rezonančním převýšením v panelu „M-circles“ (obr. 4.12-1).

Pro vykreslení citlivostních funkcí jsou použity vypočtené body otevřené regulační smyčky $L(j\omega) = A_L + jB_L$, které jsou opět body komplexní roviny. Základní vztah pro citlivostní funkci

$$S(j\omega) = \frac{1}{1 + L(j\omega)} \quad (4.31)$$

je pak možné opět rozdělit na reálnou a imaginární část, tedy

$$S(j\omega) = A_S + jB_S = \left(\frac{1 + A_L}{(1 + A_L)^2 + B_L^2} \right) + j \left(-\frac{B_L}{(1 + A_L)^2 + B_L^2} \right) \quad (4.32)$$

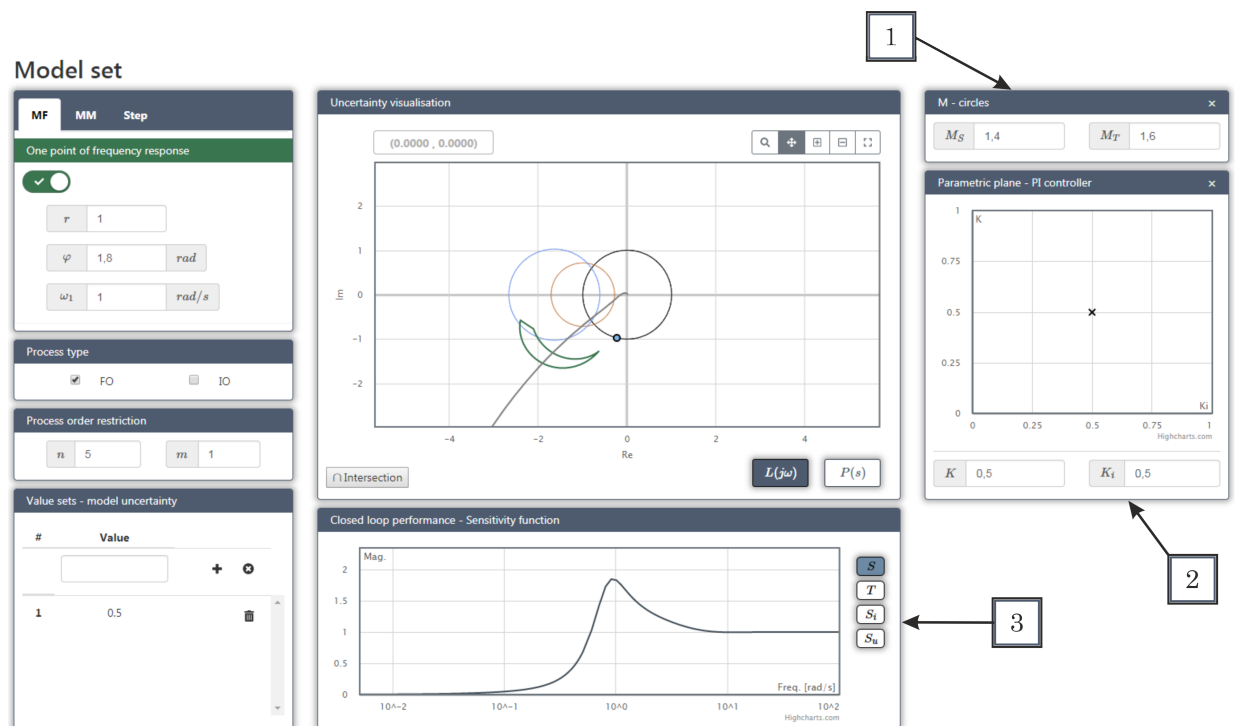
Stejný princip platí i pro ostatní citlivostní funkce:

$$T(j\omega) = \frac{L(j\omega)}{1 + L(j\omega)} \Rightarrow T(j\omega) = \left(\frac{A_L^2 + B_L^2 + A_L}{(1 + A_L)^2 + B_L^2} \right) + j \left(\frac{B_L}{(1 + A_L)^2 + B_L^2} \right) \quad (4.33)$$

$$S_i(j\omega) = P(j\omega)S(j\omega) \Rightarrow S_i(j\omega) = (A_P A_S - B_P B_S) + j (A_P B_S + A_S B_P) \quad (4.34)$$

$$S_u(j\omega) = C(j\omega)S(j\omega) \Rightarrow S_u(j\omega) = \left(K A_S - \frac{K_i}{\omega} B_S \right) + j \left(K B_S + A_S \frac{K_i}{\omega} \right) \quad (4.35)$$

Všechny výpočty v tomto režimu jsou realizovány přes JS a probíhají tedy přímo ve webovém prohlížeči uživatele. Díky tomu je dosaženo relativně rychlé odezvy zvyšující interaktivitu této webové aplikace.



Obr. 4.12: Režim otevřené regulační smyčky

4.5 Přechodová charakteristika s neurčitostí

Tato kapitola pracuje s výsledky článku prof. Ing. Miloše Schlegela, CSc. a Ing. Lukáše Kvaka [33], který představuje nový přístup k návrhu robustního PI(D) regulátoru na základě přesně vymezené třídy přechodových charakteristik řízené soustavy. Protože se však tato bakalářská práce zabývá především zobrazením neurčitosti modelu systému, je návrh regulátoru vynechán.

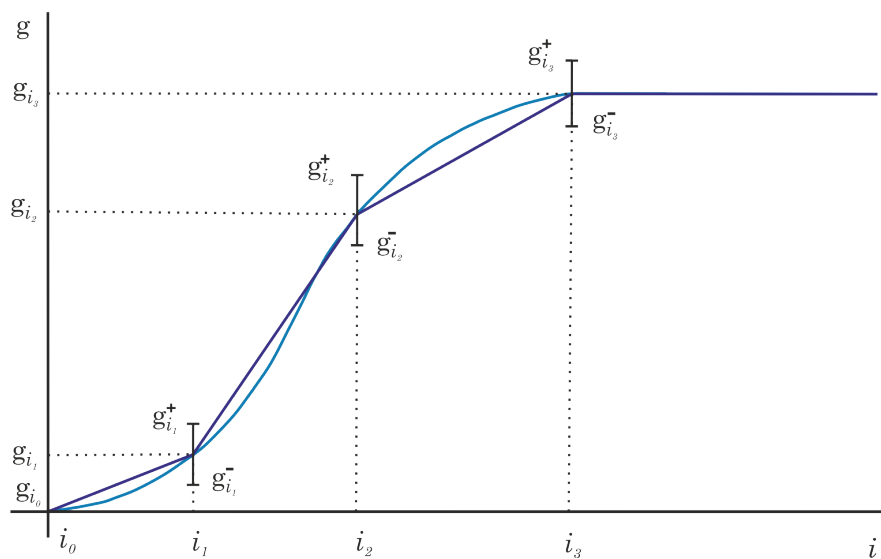
Množinový model obsahuje v tomto případě všechny FIR systémy, jejichž přechodové charakteristiky v několika vhodně vybraných časových okamžicích leží uvnitř jistých tolerančních intervalů a mimo tyto časové okamžiky jsou přechodové charakteristiky aproximovány spline polynomy zvoleného řádu l . Při dané frekvenci je pak obor hodnot frekvenčních přenosů takto definovaného množinového modelu konvexním mnohoúhelníkem v komplexní rovině \mathbb{C} , čehož je docíleno transformací afinní lineární neurčitosti modelu z časové do frekvenční oblasti, při čemž je využita teorie konvexních množin.

Nechť S označuje třídu diskrétních systémů s přechodovou charakteristikou $g \triangleq \{g_i\}_{i=0}^{\infty}$, pro kterou platí

$$g_{i_k}^- \leq g_{i_k} \leq g_{i_k}^+ \quad \text{pro } k = 0, 1, \dots, m, \quad (4.36)$$

$$g_i = g_n \quad \text{pro } i > n,$$

kde $\{i_k\}_{i=0}^m$, $0 = i_0 \leq i_1 \leq \dots \leq i_m = n$ je vybraná posloupnost z $\{0, 1, \dots, n\}$; n a m pro která platí $n \geq m$ jsou zvolená celá čísla a $g_{i_k}^-$, $g_{i_k}^+$ jsou zvolená reálná čísla. Uspořádané dvojice (i_k, g_{i_k}) jsou nazývány uzlovými body přechodové charakteristiky $\{g_i\}_{i=0}^{\infty}$. Tyto skutečnosti jsou předvedeny na obr. 4.13.



Obr. 4.13: Přechodová charakteristika s uzlovými body

Diskrétní impulsní charakteristika $h_{i=0}^{\infty}$ systémů z S je určena konvolucí, tedy

$$h_0 = g_0 \quad (4.37)$$

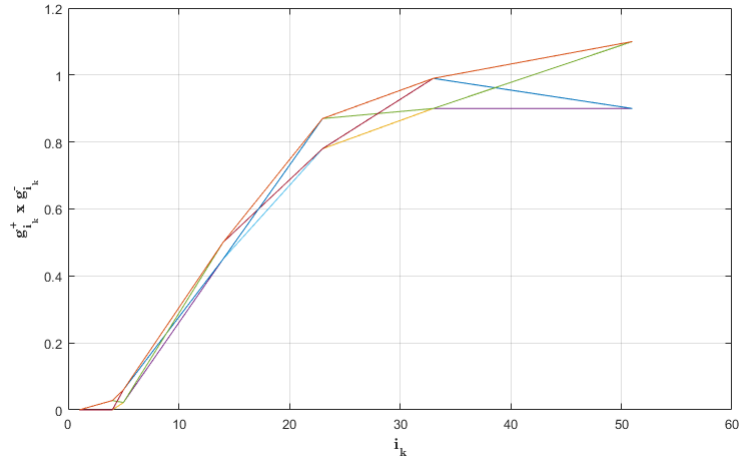
$$h_i = g_i - g_{i-1}, i = 1, 2, \dots, n$$

$$h_i = 0, i > n$$

Ohraničení přechodové charakteristiky může být vyjádřeno také pomocí prvků $h_{i=0}^{\infty}$, a to ve tvaru

$$g_{i_k}^- \leq \sum_{i=0}^{i_k} h_i \leq g_{i_k}^+ \text{ pro } k = 0, 1, \dots, m. \quad (4.38)$$

Impulsní charakteristika $h = h_{i=0}^n \in S_l$, $l \geq 1$ je uvažována jako bod v \mathbb{R}^{n+1} . Z konstrukce třídy S_l vyplývá, že v ní existuje právě $\mu = 2^{m+1}$ významných bodů, odpovídajících přechodovým odezvám, které pro každý okamžik i_k , $k = 0, 1, \dots, m$ prochází mezní hodnotou $g_{i_k}^-$ nebo $g_{i_k}^+$ tolerančního intervalu $\langle g_{i_k}^-, g_{i_k}^+ \rangle$. Těchto μ bodů je docíleno konvolucí (4.37) aproximace spline polynomem řádu l přes body sestavené kombinací mezních hodnot $g_{i_k}^-$ a $g_{i_k}^+$ (obr. 4.14). Výsledné body budou dále značeny jako h^1, h^2, \dots, h^{μ} .



Obr. 4.14: Kombinace uzlových bodů

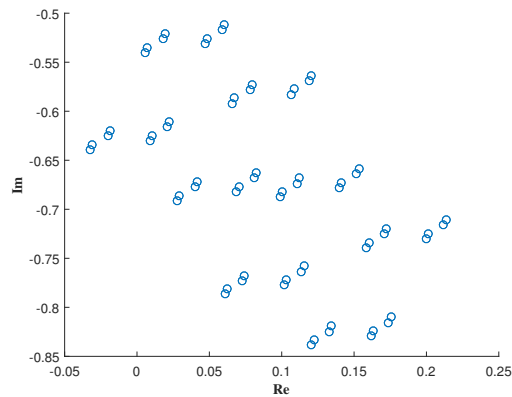
Obor hodnot, který je definován jako

$$V_l(\omega) \triangleq \left\{ P(z, h)|_{z=e^{j\omega T_{vz}}} : P(z, h) \triangleq \sum_{i=0}^n h_i z^{-i}, h = [h_0, h_1, \dots, h_n]^T \in S_l \right\}, \quad (4.39)$$

kde $l \geq 1$ a vzorkovací perioda $T_{vz} > 0$ vytváří v komplexní rovině konvexní mnohoúhelník. Tento mnohoúhelník je generován body

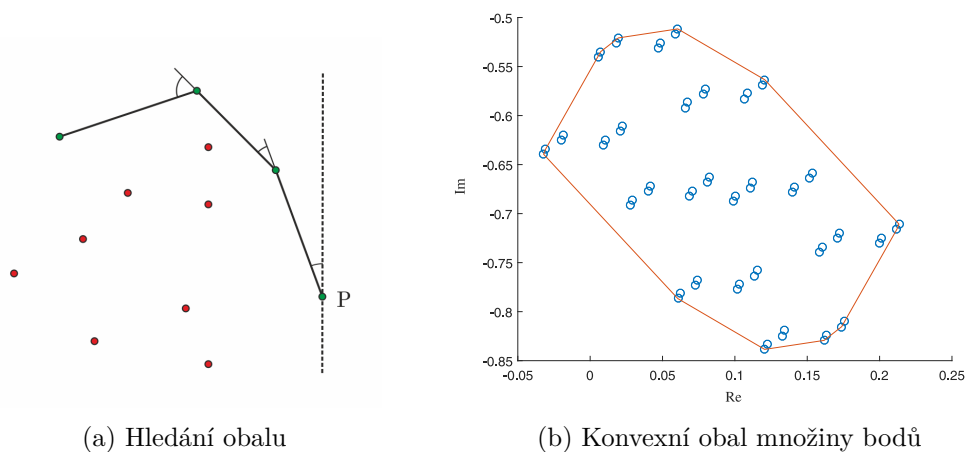
$$v_i \triangleq P(z, h^i)|_{z=e^{j\omega T_{vz}}}, i = 1, 2, \dots, \mu, \quad (4.40)$$

kde h^1, h^2, \dots, h^μ jsou extrémální body množinového modelu S_l . Ukázku vygenerovaných bodů pro konkrétní frekvenci ω lze vidět na obr. 4.15.

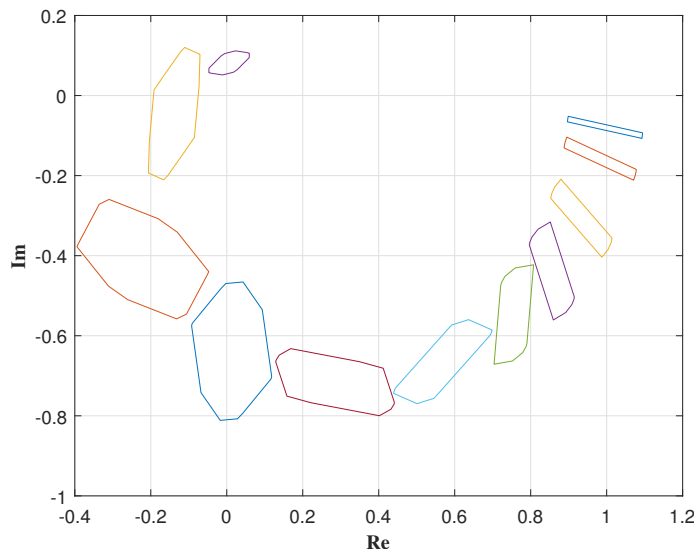


Obr. 4.15: Vygenerované body v_i

Pro nalezení konvexního obalu množiny v_1, v_2, \dots, v_μ je v této práci použit princip tzv. Gift wrapping algoritmu. V počátečním bodě $P = \max_i Re(v_i)$ je určena tzv. dělicí přímka, která je rovnoběžná s imaginární osou. Z bodu P jsou pak vedeny polopřímky do všech ostatních bodů množiny, přičemž se hledá taková polopřímka, která svírá minimální úhel s předcházející (polo)přímkou. Bod příslušný k takové polopřímce je přidán do obalu. Tento postup se opakuje, dokud nově přidáný bod není počátečním bodem P . Princip tohoto algoritmu je ilustrován na obr. 4.16(a). Výsledkem je poté konvexní mnohoúhelník (obr. 4.16(b)). Ukázka konvexních mnohoúhelníků pro množinu frekvencí $\omega = \{0.025, 0.05, 0.1, 0.15, 0.2, 0.28, 0.4, 0.55, 0.8, 1.4, 3\}$ je na obr. 4.17.



Obr. 4.16: Konvexní mnohoúhelník



Obr. 4.17: Konvexní mnohoúhelníky

4.5.1 Popis realizace ve virtuální laboratoři

Při výběru ohraničené přechodové charakteristiky jsou již funkcionality, oproti předchozím charakteristickým číslům, rozdílné. Základní rozložení stránky ale zůstává stejné. V navigačním panelu (obr. 4.18-1) je možné zadávat koeficienty čitatele a jmenovatele nominální přenosové funkce. Pro zadaný přenos lze poté v panelu „Step response visualisation“ (obr. 4.18-3) vykreslit přechodovou charakteristiku. Mimo tuto funkci jsou v navigačním panelu ještě dvě vstupní pole pro specifikaci řádu spline polynomu a vzorkovací periody, která určuje počet bodů interpolace a ovlivňuje tak velikost konvexního mnohoúhelníku. Pro zaručení nepřekrytí interpolačních bodů je vhodné volit periodu vzorkování shodnou se vzdáleností mezi nejbližšími uzlovými body na časové ose. Panel „Value sets – model uncertainty“ (obr. 4.18-2) opět slouží pro zadávání a výpis frekvencí ω .

Jak již bylo zmíněno v panelu „Step response visualisation“ (obr. 4.18-3), je možné vykreslovat přechodovou charakteristiku, mimo to je zde také možné zadávat uzlové body a přes tlačítko „Funnel“ vykreslovat ohraničení (trychtýř), které je určeno zadanými body a řádem spline polynomu. Uzlové body lze vkládat i mimo graf, a to v panelu „Control points“ (obr. 4.18-5). Tento panel obsahuje zároveň i tlačítko „Convex polygon“, které vykreslí do komplexní roviny (obr. 4.18-4) konvexní polygony pro zadané frekvence ω podle výše uvedeného postupu.



Obr. 4.18: Uživatelské rozhraní pro přechodovou charakteristiku s neurčitostí

5 Závěr

V této práci byly čtenáři představeny virtuální on-line laboratoře, konkrétně tedy laboratoře specializující se na oblast automatického řízení a robotiku. Na začátku práce byl čtenář seznámen s danou problematikou. Dále byly rozebrány technologie pro tvorbu on-line laboratoří a byla provedena rešerše dostupných projektů v této oblasti.

Na základě provedené analýzy lze usoudit, že vývoj v této oblasti dosud příliš nepokročil i přes to, že je tato technologie známá již delší dobu a v jiných oborech je relativně dosti používaná. Z části to je jistě způsobeno ukončením podpory pro Java aplety a Flash aplikace, které tvořily funkční jádro většiny on-line laboratoří. Z čehož plyne závěr, že pro budoucí vývoj je tedy nutné volit adekvátní postup a technologie, což bylo u této práce, podle mého názoru, splněno. Důležité jsou i grafické knihovny pro vykreslování dat, které silně ovlivňují interaktivitu a uživatelskou přívětivost celé laboratoře. Pro další vývoj se tedy nabízí otázka, zda volit volně dostupné knihovny, či využít komerční produkty.

Dále je nutné podotknout, že hlavní část dohledatelných on-line laboratoří je orientována především jako podpůrný výukový nástroj, bez možnosti jejich většího praktického využití. V tomto směru tedy vidím příležitost pro další vývoj.

Na závěr této práce bych už jen poznamenal, že vytvořená virtuální laboratoř je jasným příkladem aplikovatelnosti daného přístupu, kdy jsou složitější výpočty prováděny na výpočetním serveru a ty jednoduší přímo v prohlížeči uživatele, což přispívá k celkové interaktivitě celé aplikace. Pro další vývoj lze tuto laboratoř relativně snadno rozšiřovat a přidávat další funkce a výpočetní moduly. Jako námět pro budoucí práci lze uvažovat o rozšíření o pokročilé metody pro návrh a ladění regulátorů či možnost pracovat s několika charakteristickými čísly stejného typu zároveň (např. dva body frekvenční charakteristiky) a využít laboratoř jako podpůrný nástroj pro řídicí systém REXYGEN. Pro lepší uživatelskou přívětivost a praktičnost by bylo dále vhodné optimalizovat rychlost výpočtu a rychlost přenosu dat z výpočetního serveru do hlavní aplikace. V současném stavu se souhrnná doba zpracování příkazu pohybuje v řádu sekund. Mimo toto zpoždění je třeba počítat i s větší prodlevou při prvních výpočtech, kdy je tato prodleva způsobena inicializací a spouštěním běhového prostředí Matlab (Matlab Runtime). V příloze této práce jsou umístěny „screeny“, které zobrazují další mezikroky a výsledky získané při testování laboratoře.

Literatura

- [1] *Amatrol Inc. - Learning Systems for Technical Education* [online]. Dostupné z: <http://www.amatrol.com/>.
- [2] *Amatrol Process Control Training* [online]. Dostupné z: <http://www.amatrol.com/product/process-control-training/>.
- [3] *GOOGOLTech* [online]. Dostupné z: <http://www.googoltech.com/web/eng/main.jsp>.
- [4] *GUNT Hamburg* [online]. Equipment for engineering education. Dostupné z: <http://www.gunt.de/en/>.
- [5] *Modelování a identifikace řízeného systému* [online]. Dostupné z: <http://rtp.webzdarma.cz/model1.php>.
- [6] *Quanser* [online]. Dostupné z: <https://www.quanser.com/>.
- [7] *Virtual Robotic Toolkit* [online]. [cit. 3.1.2018]. Dostupné z: <https://www.virtualroboticstoolkit.com/>.
- [8] *System Identification Toolbox* [online]. Dostupné z: <https://www.mathworks.com/products/sysid.html>.
- [9] *MATLAB Online* [online]. [cit. 23.12.2017]. Dostupné z: <https://www.mathworks.com/products/matlab-online.html>.
- [10] *MATLAB Online Toolboxes* [online]. [cit. 23.12.2017]. Dostupné z: <https://www.mathworks.com/products/matlab-online/limitations.html>.
- [11] *Robust Control Toolbox* [online]. Dostupné z: <https://www.mathworks.com/products/robust.html>.
- [12] *REX Controls* [online]. [cit. 10.2.2018]. Dostupné z: <https://www.rexcontrols.cz/>.
- [13] *Virtual laboratory* [online]. [cit. 12.11.2017]. Dostupné z: http://edutechwiki.unige.ch/en/Virtual_laboratory.
- [14] ANDERSON, D. L. *Virtual Robotics Lab("Top-Down")* [online]. The Mind Project. [cit. 3.1.2018]. Dostupné z: <http://www.mind.ilstu.edu/curriculum/mod0overview.php?modGUI=208>.
- [15] BALDA, P. – KOCÁNEK, M. Programové prostředky řízení - Použití HTML5 pro vizualizaci, 2017.

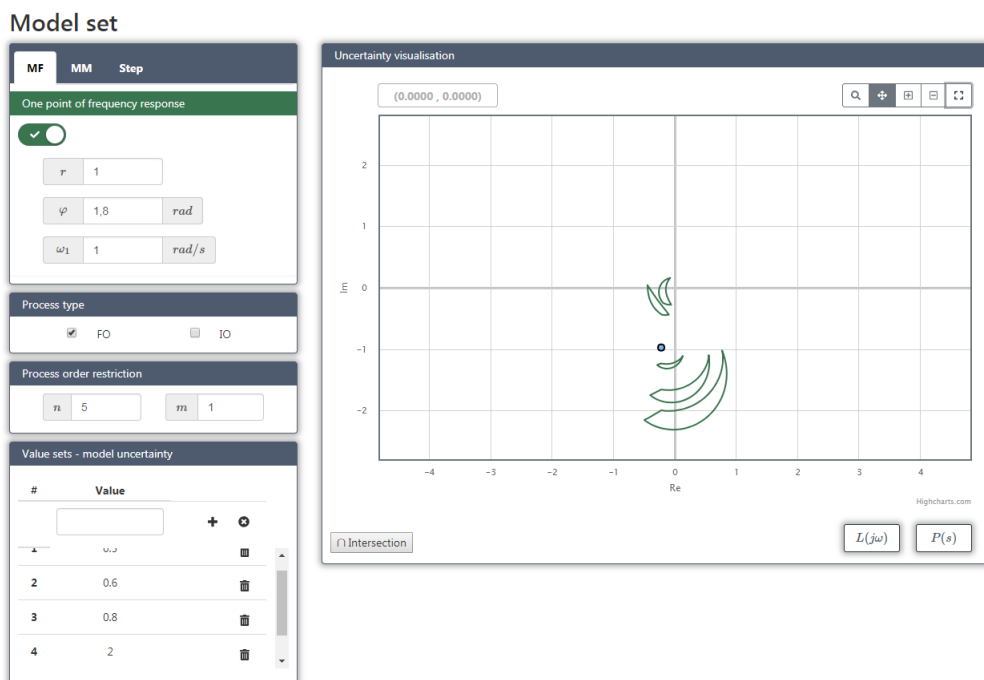
- [16] BUCHTA, L. Výkonnost, robustnost a implementace regulátorů pro průmyslové řízení. Diplomová práce, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. Dostupné z:
https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=10676.
- [17] COURTIAT, J. *Technology enhanced learning*. Birkhauser, 2004.
- [18] DUŠEK, J. Vývoj interaktivní webové aplikace. Diplomová práce, Univerzita Pardubice, Fakulta elektrotechniky a informatiky, Pardubice, 2016. Dostupné z:
<<https://theses.cz/id/jil7ao/>>.
- [19] ČECH, M. Návrh robustních regulátorů s omezenou strukturou pro systémy neceločíselného řádu. Disertační práce, Západočeská univerzita, Fakulta aplikovaných věd, Plzeň, 2008.
- [20] ČECH, M. – SCHLEGEL, M. – MERTL, J. *PIDlab, PID controller virtual laboratory* [online]. 2004. Dostupné z: <https://www.pidlab.com/cs/>.
- [21] FANKHAUSER, B. *Segway Control Simulation* [online]. [cit. 3.1.2018]. Dostupné z: <https://www.golabz.eu/lab/segway-control-simulation>.
- [22] FRAZZOLI, E. *Dynamic Systems and Control* [online]. April 2011. Dostupné z: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-241j-dynamic-systems-and-control-spring-2011/lecture-notes/MIT6_241JS11_lec17.pdf.
- [23] GOUBEJ, M. – MELICHAR, J. Lineární systémy 1. Učební text, Západočeská univerzita, Fakulta aplikovaných věd, 2016.
- [24] GU, D. W. – PETKOV, P. H. – KONSTANTINOV, M. M. *Robust control design with MATLAB*. Springer, 2005. ISBN 978-1-85233-983-8.
- [25] HERADIO, R. – TORRE, L. d. l. – DORMIDO, S. Virtual and remote labs in control education: A survey. *Annual Reviews in Control*. 2016, 42, s. 1 – 10. ISSN 1367-5788. doi: <https://doi.org/10.1016/j.arcontrol.2016.08.001>. Dostupné z:
<http://www.sciencedirect.com/science/article/pii/S1367578816300281>.
- [26] KHEIR, N. et al. Control systems engineering education. *Automatica*. 1996, 32, 2, s. 147 – 166. ISSN 0005-1098. doi: [https://doi.org/10.1016/0005-1098\(96\)85546-4](https://doi.org/10.1016/0005-1098(96)85546-4). Dostupné z:
<http://www.sciencedirect.com/science/article/pii/0005109896855464>.
- [27] KOSEK, J. *JavaScriptové knihovny a frameworky* [online]. [cit. 23.11.2017]. Dostupné z: <http://www.kosek.cz/vyuka/4iz268/prednasky/js-frameworky/#/R.1.3.2>.

- [28] KOTTNAUER, J. Jemný úvod do teorie ASP.NET MVC. *Programujte.com*. 2009. Dostupné z: <http://programujte.com/clanek/2009050200-jemny-uvod-do-teorie-asp-net-mvc/>.
- [29] MACALÍK, R. Identifikace regulované soustavy – experimentální jednotka pro spalování biomasy. Diplomová práce, Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2008. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=10676.
- [30] PUNČOCHÁŘ, I. Základy identifikace systémů a detekce chyb. Učební text, Západočeská univerzita, Fakulta aplikovaných věd, 2016.
- [31] ROBINSON, D. Oracle signals the end of Java applet support for browsers | V3. <http://www.v3.co.uk>. Jan 2016. Dostupné z: <https://www.v3.co.uk/v3-uk/news/2443810/oracle-signals-the-end-of-java-applet-support-for-browsers>.
- [32] SCHLEGEL, M. Nový přístup k robustnímu návrhu průmyslových regulátorů. Habilitační práce, Západočeská univerzita, Fakulta aplikovaných věd, Plzeň, Únor 2000.
- [33] SCHLEGEL, M. – KVAK, L. Computing value sets for systems with constrained step response. s. 1–7, Bratislava, 2005. Slovak University of Technology. Dostupné z: http://www.kky.zcu.cz/en/publications/SchlegelM_2005_Computingvaluesets_1. ISBN 80-227-2235-9.
- [34] SCHLEGEL, M. – ČECH, M. Computing value sets from one point of frequency response with applications. s. 1–6, Prague, 2005. IFAC. Dostupné z: http://www.kky.zcu.cz/en/publications/SchlegelM_2005_Computingvaluesets.
- [35] SCHLEGEL, M. – BALDA, P. – ŠTĚTINA, M. Robustní PID autotuner – momentová metoda. *Automatizace*. 2003, 4, s. 242–246. Dostupné z: http://www.kky.zcu.cz/en/publications/SchlegelM_2003_RobustniPID.
- [36] SEILER, S. *Definition of Remote Labs, Virtual Labs and Distance Labs* [online]. Jan 2017. [cit. 5.12.2017]. Dostupné z: <https://seiler.it/remote-and-virtual-labs/>.
- [37] SKOGESTAD, S. – POSTLETHWAITE, I. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, 2005. Dostupné z: <http://een.iust.ac.ir/profs/Jahed/MultiVariable/Multivariable%20Feedback%20Control%20-%20%20Analysis%20and%20Design.pdf>. ISBN 0470011688.
- [38] UNED. *University Network of Interactive Laboratories* [online]. [cit. 10.2.2018]. Dostupné z: <http://unilabs.dia.uned.es/>.
- [39] VRAŠTIL, M. Virtuální laboratoře pro automatické řízení na základě nových webových standardů. Bakalářská práce, Západočeská univerzita, Fakulta aplikovaných věd, Plzeň, 2018.

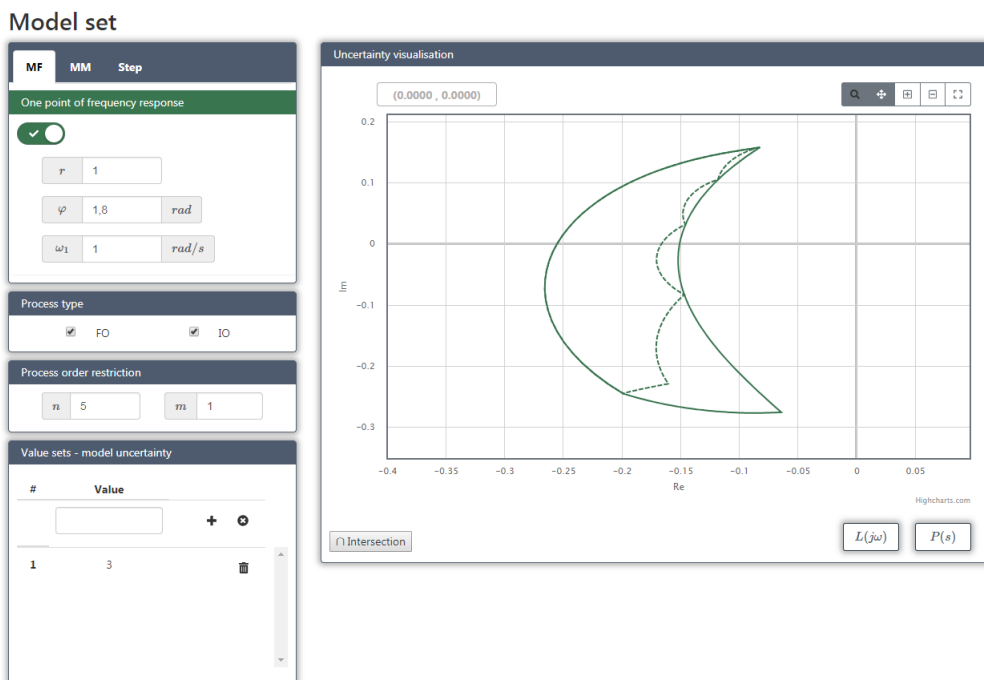
- [40] VROŽINA, M. – JANČÍKOVÁ, Z. – DAVID, J. Identifikace systémů. Učební text, Vysoká škola báňská – Technická univerzita Ostrava, 2012. Dostupné z: <http://www.person.vsb.cz/archivcd/FMMI/IS/Identifikace%20systemu.pdf>.
- [41] WIKIPEDIA. *AJAX* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 22.11.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=AJAX&oldid=15394363>.
- [42] WIKIPEDIA. *ASP.NET Core* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 15.12.2017]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=ASP.NET%20Core&oldid=836339908>.
- [43] WIKIPEDIA. *ASP.NET MVC Framework* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 15.12.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=ASP.NET%20MVC%20Framework&oldid=10127793>.
- [44] WIKIPEDIA. *ASP.NET* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 15.12.2017]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=ASP.NET&oldid=837517755>.
- [45] WIKIPEDIA. *Active Server Pages* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 15.12.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Active%20Server%20Pages&oldid=15400483>.
- [46] WIKIPEDIA. *Adobe Flash* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 20.12.2017]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=Adobe%20Flash&oldid=837034719>.
- [47] WIKIPEDIA. *Applet* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 20.12.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Applet&oldid=15436569>.
- [48] WIKIPEDIA. *Bootstrap* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 4.4.2018]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Bootstrap&oldid=15728589>.
- [49] WIKIPEDIA. *Comparison of JavaScript charting libraries* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 4.4.2018]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=Comparison%20of%20JavaScript%20charting%20libraries&oldid=836447201>.
- [50] WIKIPEDIA. *HyperText Markup Language* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 20.11.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=HyperText%20Markup%20Language&oldid=15892640>.

- [51] WIKIPEDIA. *JavaScript* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 22.11.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=JavaScript&oldid=15390079>.
- [52] WIKIPEDIA. *Kaskádové styly* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [Online; accessed 21-April-2018]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Kask%C3%A1dov%C3%A9%20styly&oldid=15978379>.
- [53] WIKIPEDIA. *List of JavaScript libraries* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 22.11.2017]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=List%20of%20JavaScript%20libraries&oldid=833505140>.
- [54] WIKIPEDIA. *MATLAB* — *Wikipedia, The Free Encyclopedia* [online]. 2018. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=MATLAB&oldid=15885379>.
- [55] WIKIPEDIA. *Model-view-controller* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 15.12.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Model-view-controller&oldid=13938376>.
- [56] WIKIPEDIA. *PHP* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 22.11.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=PHP&oldid=15717169>.
- [57] WIKIPEDIA. *Plotly* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 4.4.2018]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=Plotly&oldid=835370855>.
- [58] WIKIPEDIA. *Virtuální realita* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 12.11.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Virtu%C3%A1ln%C3%AD%20realita&oldid=15810679>.
- [59] WIKIPEDIA. *Web application* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 12.11.2017]. Dostupné z: <http://en.wikipedia.org/w/index.php?title=Web%20application&oldid=835981007>.
- [60] WIKIPEDIA. *Webová aplikace* — *Wikipedia, The Free Encyclopedia* [online]. 2018. [cit. 12.11.2017]. Dostupné z: <http://cs.wikipedia.org/w/index.php?title=Webov%C3%A1%20aplikace&oldid=15394645>.

A Testování laboratoře



Obr. A.1: Obor hodnot množinového modelu pro jeden bod frekvenční charakteristiky(MF), neceločíselný řád systému(FO) a množinu frekvencí ω



Obr. A.2: Porovnání oborů hodnot pro neceločíselný(FO) a celočíselný(IO) řád systému při znalosti jednoho bodu frekvenční charakteristiky(MF)

Model set

MF MM Step

Moments of the impulse response

κ 1,00 μ 1,00

σ^2 0,60

Process type

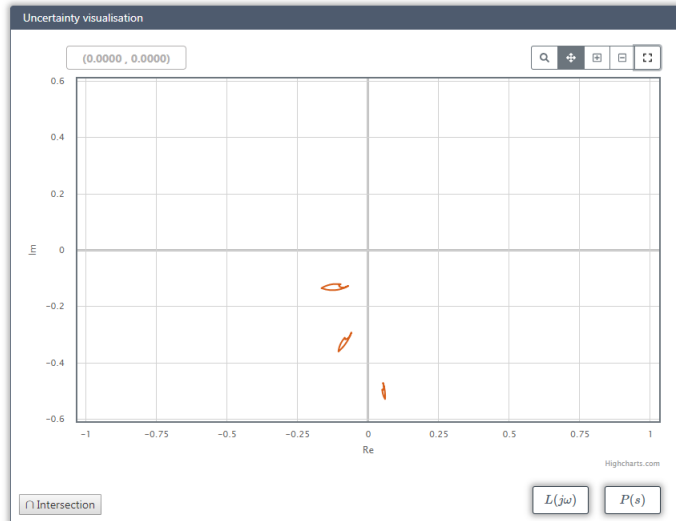
FO IO

Process order restriction

n 5 m 1

Value sets - model uncertainty

#	Value
1	2
2	3
3	5



Obr. A.3: Obor hodnot množinového modelu pro momenty impulsní funkce(MM), neceločíslný řád systému(FO) a množinu frekvencí ω

Model set

MF MM Step

Moments of the impulse response

κ 1,00 μ 1,00

σ^2 0,60

Process type

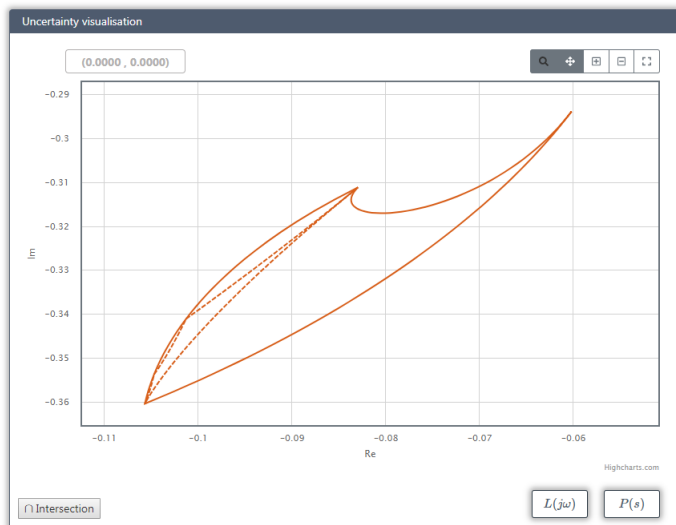
FO IO

Process order restriction

n 5 m 1

Value sets - model uncertainty

#	Value
1	3



Obr. A.4: Porovnání oborů hodnot pro neceločíslný(FO) a celočíslný(IO) řád systému při znalosti momentů impulsní funkce(MM)

Model set

MF
MM
Step

One point of frequency response

r
0,9823

φ
1,1662 rad

ω_1
1 rad/s

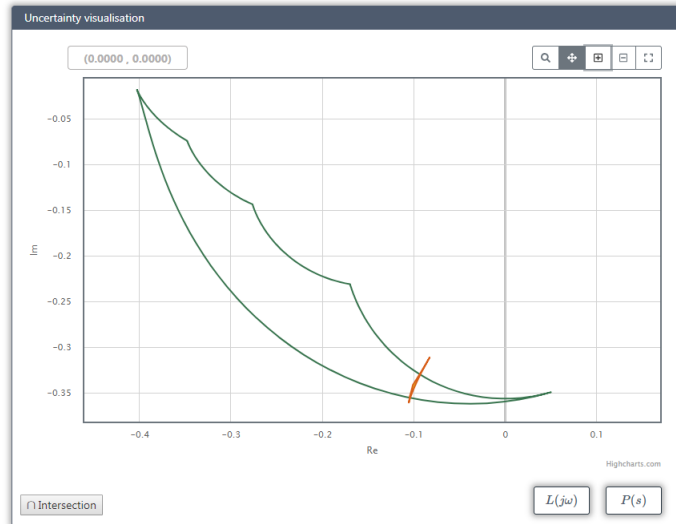
Process type

FO
 IO

Process order restriction

Value sets - model uncertainty

#	Value	
1	3	+ - ✕



Obr. A.5: Kombinace jednoho bodu frekvenční charakteristiky(MF) a momentů impulsní funkce(MM) pro celočíselný řád systému(IO)

Model set

MF
MM
Step

One point of frequency response

r
0,9823

φ
1,1662 rad

ω_1
1 rad/s

Process type

FO
 IO

Process order restriction

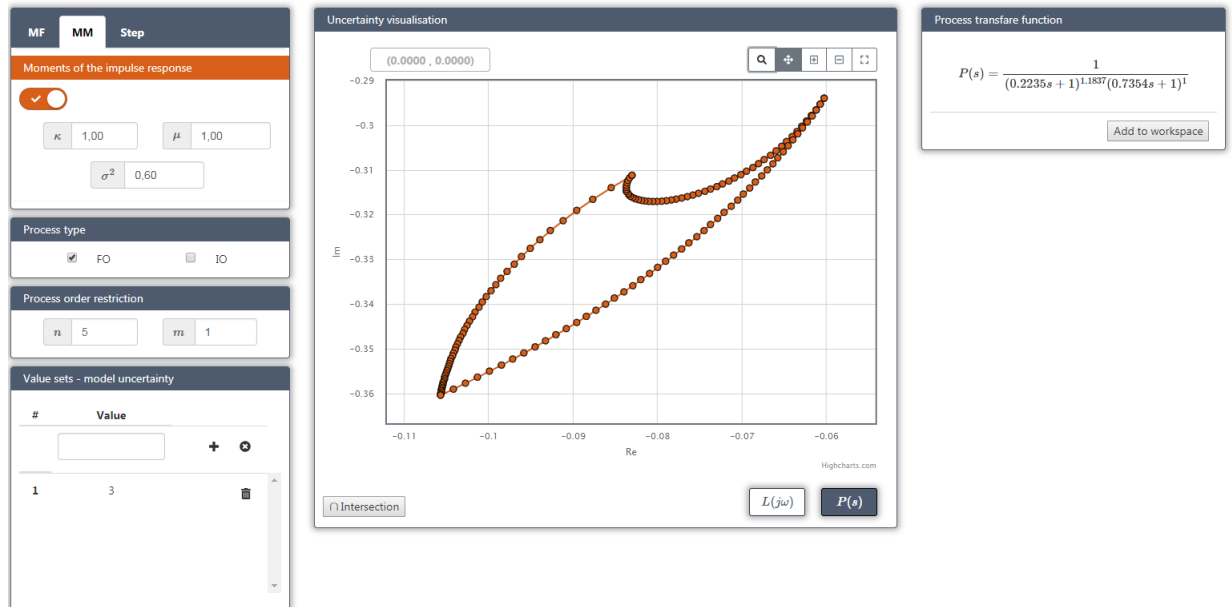
Value sets - model uncertainty

#	Value	
1	3	+ - ✕



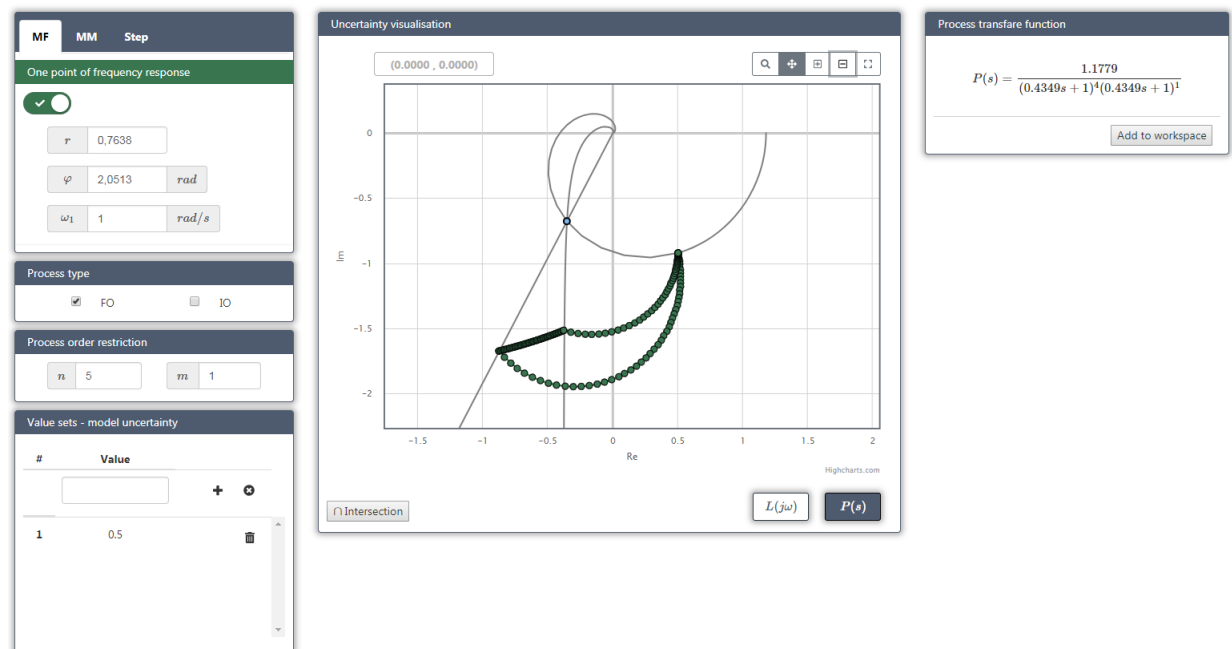
Obr. A.6: Přiblížení průniku z obr. A.5

Model set



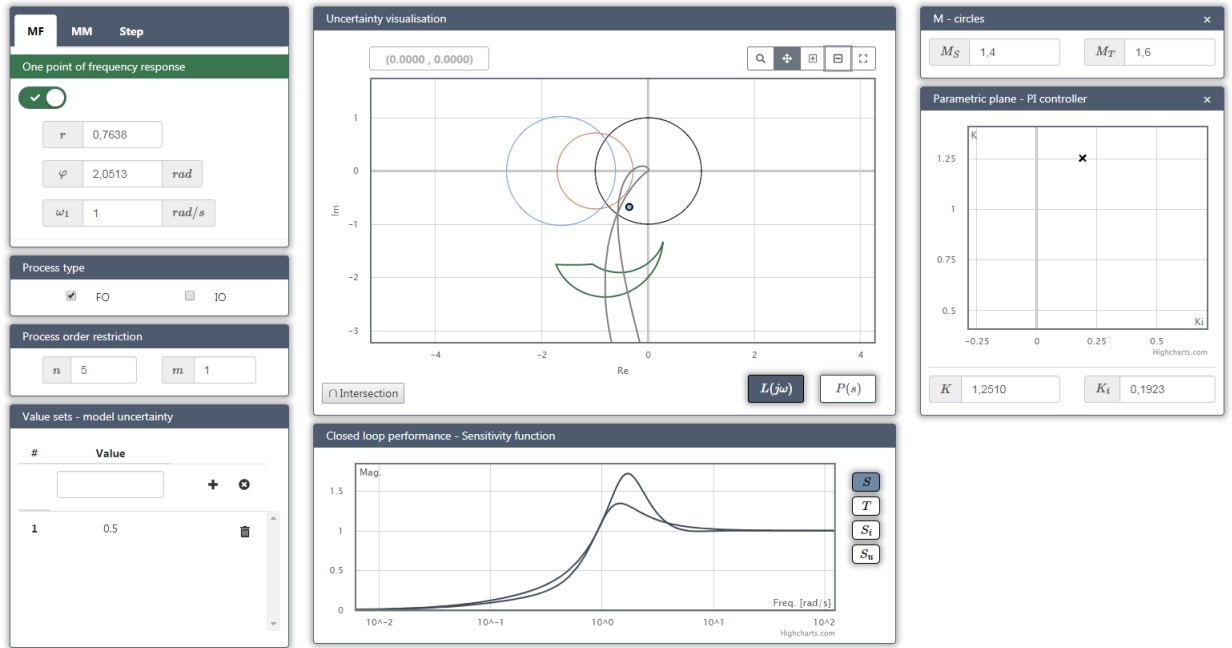
Obr. A.7: Body ohraničující obor hodnot při znalosti momentů impulsní funkce(MM) a přenosová funkce příslušná k jednomu z bodů

Model set



Obr. A.8: Obor hodnot při znalosti jednoho bodu frekvenční charakteristiky(MF) a Nyquistovy křivky příslušné třem různým bodům hranice daného oboru hodnot

Model set



Obr. A.9: Frekvenční charakteristika pro otevřenou smyčku $L(j\omega)$ a jednoduchá volba parametrů PI regulátoru

B Obsah přiloženého CD

- **Dokumentace**

- **PIDFactory**

- \PIDlab**

- Hlavní ASP.NET Core projekt

- \PIDlabMathApi**

- Výpočetní server

- \PIDlabServer\lib**

- Knihovny a výpočetní moduly

- \SqlScripts**

- Skript pro vytvoření MySQL databáze

- \TCPClient**

- Projekt zajišťující komunikaci s výpočetním serverem

- \Tools**

- Pomocné projekty a nástroje (např. Parser)

C Výpočetní moduly

Mezi knihovnamí výpočetního serveru jsou i čtyři výpočetní moduly, které obsahují metody pro získání oborů hodnot, jejich průniků a dalších funkcionalit. Tyto moduly a jejich metody jsou níže popsány.

- `ModelSetUncertainty.jar`

Modul slouží pro výpočet oboru hodnot, resp. hranice oboru hodnot množinového modelu při znalosti jednoho bodu frekvenční charakteristiky

- neceločíselný řád

$$[Xarray, Yarray] = \text{PEPFO}(r, \phi, w1, n, m, \text{jemnost}, w)$$

- celočíselný řád

$$[Xarray, Yarray] = \text{PEPIO}(r, \phi, w1, n, \text{jemnost}, w)$$

nebo momentů impulsní funkce.

- neceločíselný řád

$$[Xarray, Yarray] = \text{momFO}(\kappa, u, \sigma^2, n, m, \text{jemnost}, w)$$

- celočíselný řád

$$[Xarray, Yarray] = \text{momIO}(\kappa, u, \sigma^2, n, \text{jemnost}, w)$$

- `Intersection.jar`

Tento modul slouží pro výpočet průniků oborů hodnot, tedy pro výpočet hranice kombinace jednoho bodu frekvenční charakteristiky a momentů impulsní funkce.

$$[Xarray, Yarray] = \text{intersectionP}(P1x, P1y, P2x, P2y)$$

- `GetSys.jar`

Díky tomuto modulu je možné zobrazit extrémální přenosovou funkci procesu pro vybraný bod hranice oboru hodnot získaného ze znalosti jednoho bodu frekvenční charakteristiky

- neceločíselný řád

$$[i, \text{num}, \text{den1}, \text{den2}] = \text{GetSys_PEPFO}(x, y, r, \phi, w1, n, m, \text{jemnost}, w)$$

- o celočíselný řád

$$[K, \tau, n_y, k, l, m] = \text{GetSys_PEPIO}(x, y, A, \phi, w_1, n, \text{jemnost}, w)$$

či momentů impulsní funkce.

- o neceločíselný řád

$$[\kappa, \tau_1, n_1, \tau_2, n_2] = \text{GetSys_momFO}(x, y, \kappa, u, \sigma_2, n, m, \text{jemnost}, w)$$

- o celočíselný řád

$$[\kappa, \tau_1, \tau_2, \tau_3, k, l, m] = \text{GetSys_momIO}(x, y, \kappa, u, \sigma_2, n, \text{jemnost}, w)$$

- StepUncertainty.jar

Modul slouží pro výpočet konvexních mnohoúhelníků, jimiž jsou obory hodnot množinového modelu, který obsahuje všechny FIR systémy, jejichž přechodové charakteristiky v několika vhodně vybraných časových okamžicích leží uvnitř jistých tolerančních intervalů.

$$[Xarray, Yarray] = \text{StepUncertainty}(T_{vz}, \text{interp_order}, ik_x, gikm, gikp, w)$$

$$[Xarray, Yarray] = \text{StepUncertainty_Step}(\text{num}, \text{den})$$

Vstupy

- $r(A), \phi, w_1$ - parametry změřeného bodu frekvenční charakteristiky
- κ, u, σ_2 - čísla odpovídající znalost prvních třech momentů impulsní funkce
- n - maximální řád procesu
- m - minimální stupeň pólu
- jemnost - počet bodů z kterých se skládá hranice oboru hodnot množinového modelu
- w - frekvence ω
- P_{1x}, P_{2x}, x - souřadnice množiny bodů či jednoho bodu na reálné ose
- P_{1y}, P_{2y}, y - souřadnice množiny bodů či jednoho bodu na imaginární ose

- `Tvz` - vzorkovací perioda
- `interp_order` - řád spline polynomu
- `ik_x` - souřadnice uzlových bodů na časové ose
- `gikm`, `gikp` - souřadnice uzlových bodů na amplitudové ose

Výstupy

- `Xarray`, `Yarray` - souřadnice množiny vypočtených bodů
- `i` - typ extrémální přenosové funkce pro PEPFO
- `num` - číselník extrémální přenosové funkce
- `den1`, `den2` - parametry jmenovatele extrémální přenosové funkce
- `K`, `tau`, `ny`, `k`, `l`, `m`, `kappa`, `tau1`, `tau2`, `tau3` - parametry extrémálních přenosových funkcí