

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA KYBERNETIKY

DIPLOMOVÁ PRÁCE

**Návrh PID regulátoru metodou
H-nekonečno optimalizace**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2019/2020

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

Jméno a příjmení: **Bc. Michal BRABEC**
Osobní číslo: **A18N0052P**
Studijní program: **N3918 Aplikované vědy a informatika**
Studijní obor: **Kybernetika a řídicí technika**
Téma práce: **Návrh PID regulátoru metodou H_{∞} optimalizace**
Zadávající katedra: **Katedra kybernetiky**

Zásady pro vypracování

1. Seznamte se se současnými metodami návrhu PID regulátoru pro řízený systém se zadaným přenosem, které jsou založeny na optimalizačních metodách.
2. Analyzujte existující metody návrhu PI regulátoru na základě H_{∞} optimalizace. Soustředte se hlavně na metodu popsanou v práci [1].
3. Navrhněte a implementujte algoritmus výpočtu PI regulátoru podle [1] v programovém prostředí MATLAB. Navrhněte a realizujte též příslušné uživatelské prostředí vhodné pro co nejširší okruh uživatelů.
4. Doplněte vzniklý softwarový nástroj tak, aby umožňoval i suboptimální návrh PID regulátoru.
5. Funkční vlastnosti navrženého nástroje ověřte na široké třídě testovacích přenosů řízených systémů uvedených v práci [2].

Rozsah diplomové práce: **40-50**
Rozsah grafických prací: **dle potřeby**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

- [1] M. Schlegel, P. Medvecová. Design of PI Controllers: H_{inf} Region Approach. *IFAC-PapersOnLine*. 2018.
- [2] K. J. Aström, H. Panagopoulos a T. Häggglund. Design of PI Controllers based on Non-Convex Optimization. *Automatica*, 1998.
- [3] O. Garpinger. Analysis and Design of Software-Based Optimal PID Controllers. Department of Automatic Control, Lund Institute of Technology, Lund University. 2015.

Vedoucí diplomové práce: **Prof. Ing. Miloš Schlegel, CSc.**
Katedra kybernetiky

Datum zadání diplomové práce: **1. října 2019**
Termín odevzdání diplomové práce: **25. května 2020**

Radová

Doc. Dr. Ing. Vlasta Radová
děkanka



J. Psutka

Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

Prohlášení

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

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

V Plzni dne 28. června 2020

.....
Bc. Michal Brabec

Poděkování

Tímto bych rád poděkoval Prof. Ing. Miloši Schlegelovi, CSc., za spolupráci, cenné rady, připomínky a věnovaný čas při vypracovávání diplomové práce.

Anotace

Hlavním cílem této diplomové práce je návrh a implementace algoritmu pro výpočet optimálních parametrů PI regulátoru a suboptimálních parametrů PID regulátoru s využitím H_∞ optimalizace. Implementací navrženého algoritmu a vytvořením odpovídajícího uživatelského rozhraní v programovém prostředí MATLAB je získán rychlý, interaktivní a snadno rozšiřitelný softwarový nástroj pro návrh PI a PID regulátoru. Na začátku práce jsou stručně shrnuty základní přístupy k návrhu regulátoru a jejich historický vývoj. Současně jsou uvedeny základní teoretické poznatky spolu s popisem několika existujících metod pro návrh PI a PID regulátoru s využitím H_∞ optimalizace. Tento teoretický úvod je následně rozšířen o detailní popis návrhové úlohy, která byla při vytváření algoritmu použita. Závěrečnou část pak tvoří popis uživatelského rozhraní a ovládacích funkcí návrhové laboratoře. V příloze práce jsou pak uvedeny ukázkové příklady řešení.

Klíčová slova: H_∞ optimalizace, H_∞ region, regulátor s omezenou strukturou, parametrická rovina, MATLAB

Abstract

The main purpose of this Diploma thesis is the design of an algorithm for calculating the optimal parameters of the PI controller and suboptimal parameters of the PID controller based on H_∞ optimization. The proposed algorithm was then implemented in the MATLAB programming environment. A user interface has also been created for a fast, interactive and easy-to-use controller design. At the beginning of the thesis, the basic approaches to the controller design and their historical development are briefly summarized. At the same time, the basic theoretical knowledge is presented together with a description of several existing methods for the design of PI and PID controllers using H_∞ optimization technique. This theoretical introduction is then extended by a detailed description of the design task which was used during the algorithm construction. The final part of the thesis contains a description of the user interface and control functions of the design laboratory. Sample examples of solutions are given in the appendix.

Keywords: H_∞ optimization, H_∞ region, controller with limited structure, parametric plane, MATLAB

Obsah

Seznam obrázků	9
Seznam použitých symbolů a značek	11
1 Úvod	12
1.1 Motivace	12
1.2 Popis problému	12
2 Úvod do regulace	14
2.1 PID regulátor	14
2.2 Model řízeného systému	15
2.3 Průřez návrhovými metodami	16
2.4 Návrhová omezení a požadavky	17
2.5 Systémové normy	21
3 Metody pro návrh regulátoru na základě H_2 a H_∞ optimalizace	22
3.1 Obecná H_2 a H_∞ optimalizace	22
3.2 H_∞ optimalizace pro návrh PI a PID regulátorů	23
3.2.1 MIGO (M constrained Integral Gain Optimization)	24
3.2.2 SWORD (Software-based optimal robust design)	24
4 Návrh regulátoru s omezenou strukturou metodou H_∞	26
4.1 Návrhová úloha s omezením typu H_∞	27
4.1.1 Prototypový algoritmus	27
4.1.2 Řešení návrhové úlohy	28
4.2 Implementace návrhového algoritmu	28
4.2.1 Řešení soustav a rovnic z věty 1	30
4.2.2 Kořeny kvartického polynomu	33
4.2.3 Hranice H_∞ regionu	34
4.2.4 Průnik návrhových omezení typu H_∞	36
5 Uživatelské rozhraní	38
5.1 MATLAB App Designer	38
5.2 Prostředí oken	39
5.2.1 Hlavní okno	40
5.2.2 Nový systém	41

5.2.3	Databáze systémů	42
5.2.4	Databáze váhových funkcí	45
5.2.5	Volba typu regulátoru	46
5.2.6	Nastavení	47
5.3	Charakteristiky	50
5.4	Pomocné nástroje	51
6	Závěr	52
	Literatura	53
A	Příklady řešení	II
B	Návod na zprovoznění aplikace	XII
C	Diagram řešení kvartické rovnice	XIV
D	Vývojový diagram CCsort	XV

Seznam obrázků

2.1	Standardní forma regulační smyčky	14
2.2	Průběh regulační odchylky e	18
2.3	Amplitudová frekvenční charakteristika pro $S(s)$ a $T(s)$	19
2.4	M_S a M_T kružnice	19
2.5	Vztah mezi citlivostní funkcí $S(s)$ a váhovou funkcí $W(s)$	20
3.1	Formalismus pro řešení H_2 a H_∞ optimalizačních úloh	22
4.1	Regulační smyčka	26
4.2	Padého aproximace čistého dopravního zpoždění	29
4.3	Křivky množiny \mathcal{B}	34
4.4	Toleranční pásmo	35
4.5	Křivky množiny \mathcal{B}^*	36
4.6	Hledání průniku dvou polygonů	37
5.1	App Designer	38
5.2	Uvítací okno	39
5.3	Hlavní rozhraní aplikace	40
5.4	Vytvoření nového systému	41
5.5	Databáze systémů	42
5.6	Výběr dat z tabulky	43
5.7	Databáze váhových funkcí	46
5.8	Volba PID regulátoru	46
5.9	Nastavení	48
5.10	Stav aplikace po nalezení řešení	49
5.11	Výsledek pro několik návrhových omezení	49
5.12	Vykreslení charakteristik	50
5.13	Pomocné nástroje	51
5.14	Vícebodová časová analýza	51
5.15	Vrstevnice IAE kritéria	51
A.1	Př. 1 - H_∞ region	II
A.2	Př. 1 - AFCH pro optim. řešení	II
A.3	Př. 1 - Časové charakteristiky pro optim. řešení	II
A.4	Př. 2 - H_∞ region	III
A.5	Př. 2 - AFCH pro optim. řešení	III

A.6	Př. 2 - Časové charakteristiky pro optim. řešení	III
A.7	Př. 3 - Disjunktní H_∞ regiony	III
A.8	Př. 3 - H_∞ region - #2	III
A.9	Př. 3 - AFCH pro optim. řešení	III
A.10	Př. 3 - Časové charakteristiky pro optim. řešení	III
A.11	Př. 4 - H_∞ region	IV
A.12	Př. 4 - AFCH pro optim. řešení	IV
A.13	Př. 4 - Časové charakteristiky pro optim. řešení	IV
A.14	Př. 5 - H_∞ regiony	V
A.15	Př. 5 - Průnik návrhových kritérií	V
A.16	Př. 5 - AFCH pro optim. řešení	V
A.17	Př. 5 - Časové charakteristiky pro optim. řešení	V
A.18	Př. 6 - H_∞ region	VI
A.19	Př. 6 - AFCH pro optim. řešení	VI
A.20	Př. 6 - Časové charakteristiky pro optim. řešení	VI
A.21	Př. 7 - H_∞ region	VII
A.22	Př. 7 - AFCH pro optim. řešení	VII
A.23	Př. 7 - Časové charakteristiky pro optim. řešení	VII
A.24	Př. 8 - H_∞ regiony	VIII
A.25	Př. 8 - Průnik návrhových kritérií	VIII
A.26	Př. 8 - AFCH pro optim. řešení	VIII
A.27	Př. 8 - Časové charakteristiky pro optim. řešení	VIII
A.28	Př. 9 - H_∞ region	IX
A.29	Př. 9 - AFCH pro optim. řešení	IX
A.30	Př. 9 - Časové charakteristiky pro optim. řešení	IX
A.31	Př. 10 - H_∞ regiony	X
A.32	Př. 10 - Průnik návrhových kritérií	X
A.33	Př. 10 - Časové charakteristiky pro optim. řešení	X
A.34	Př. 10 - AFCH pro optim. řešení	XI
B.1	MATLAB - Kontrolní panel	XII
B.2	MATLAB - Instalace/Re-instalace	XII
B.3	MATLAB - Apps	XIII
B.4	MATLAB - Run	XIII

Seznam použitých symbolů a zkratek

- FOPDT - model systému prvního řádu s dopravním zpožděním
SOPDT - model systému druhého řádu s dopravním zpožděním
 $r(t)$ - referenční signál
 $e(t)$ - regulační odchylka
 $u(t)$ - akční veličina
 $d_i(t)$ - porucha na vstupu řízeného systému
 $d_o(t)$ - porucha na výstupu řízeného systému
 $n(t)$ - šum
 $y(t)$ - regulovaná veličina
I/O - Input/Output (Vstupně/Výstupní)
P - Proporcionální regulátor
PI - Proporcionálně Integrační regulátor
PID - Proporcionálně-Integračně-Derivační regulátor
 k_p - proporcionální zesílení
 T_i - časová konstanta integračního članku
 T_d - časová konstanta derivačního članku
 k_i - integrační zesílení
 k_d - derivační zesílení
LQR - Linear-Quadratic Regulator (Lineární kvadratický regulátor)
LQG - Linear-Quadratic-Gaussian (Lineární kvadratické Gaussovské řízení)
 $L(s)$ - přenosová funkce otevřené smyčky
 $S(s)$ - citlivostní funkce
 $T(s)$ - komplementární citlivostní funkce
 $S_C(s)$ - citlivostní funkce řízení
 $S_P(s)$ - citlivostní funkce poruchy na vstupu
 $W(s)$ - váhová funkce
IE - integrační kritérium kvality Integrated Error
IAE - integrační kritérium kvality Integrated Absolute Error
ITAE - integrační kritérium kvality Integrated Time-weighted Error
ISE - integrační kritérium kvality Integrated Square Error

1 Úvod

Hlavním cílem mé diplomové práce je vytvoření interaktivního softwarového nástroje pro výpočet parametrů PI a PID regulátoru na základě H_∞ omezujících podmínek. V teoretické části práce se čtenář blíže seznámí se stručným historickým vývojem návrhových metod, stejně jako se základními teoretickými poznatky, na kterých je založena návrhová metoda používaná ve vytvořené aplikaci. Další část práce pak představuje samotnou návrhovou metodu a její implementaci. Závěrečnou část tvoří popis uživatelského rozhraní a jednotlivých ovládacích funkcí aplikace.

1.1 Motivace

Návrh regulátoru a regulační smyčky hraje významnou roli při automatizaci, kdy je cílem efektivně řídit různorodé technologické procesy a zařízení. Výjimkou nejsou ani aplikace z oblasti robotiky, které také vyžadují velké množství regulačních prvků. Správné nastavení těchto prvků silně ovlivňuje výslednou efektivitu a účinnost dané soustavy. Optimálnost používaného nastavení jednotlivých regulátorů je tak již dlouho probíraným tématem. Obecně totiž stále platí, že většina regulačních smyček, které jsou v průmyslových procesech implementovány, není vhodně navržena nebo seřizena. V mnoha případech je dokonce ponecháno jejich počáteční nastavení poskytnuté výrobcem či dodavatelem. Tímto přístupem pak vznikají značné ekonomické ztráty.

1.2 Popis problému

Existuje velké množství metod a přístupů pro návrh regulátoru, jenž je optimální ve smyslu H_∞ teorie. Tyto metody ale ve většině případů ústí v regulátor vysokého řádu, často vyššího než má původní systém. Takovéto regulátory jsou v praxi těžko implementovatelné a musela by být využita některá z technik pro redukci řádu regulátoru. Další možností by bylo redukovat řád modelu řízeného systému, což by snížilo i výsledný řád regulátoru.

V současné době je však pozornost zaměřena na hledání metody pro přímý návrh regulátorů nízkého řádu (PI nebo PID). Takováto metoda by podle [18] musela zaručovat několik základních požadavků. Metoda by měla být aplikovatelná na širokou třídu systémů zahrnující neminimálně fázové systémy, nestabilní systémy a systémy s dopravním zpožděním. Zároveň by měla umožňovat specifikovat různorodé návrhové požadavky. Výsledný návrh by pak měl poskytovat úplné řešení návrhové úlohy. Získané

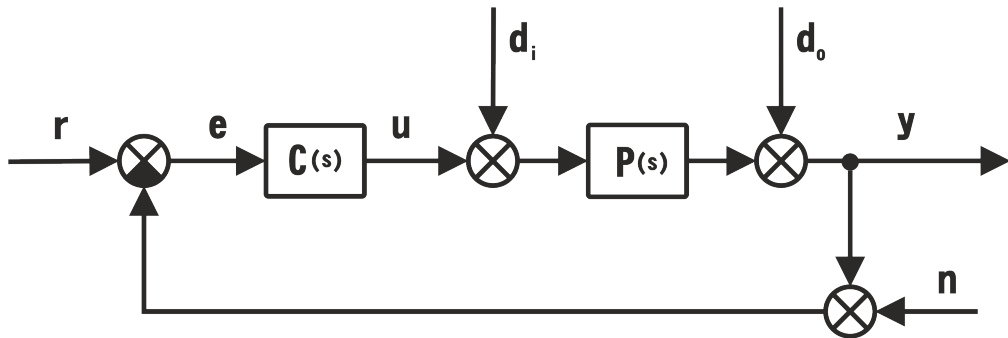
řešení by tedy ve výsledku mělo specifikovat všechny regulátory, které splňují návrhové požadavky, a pokud řešení neexistuje, měla by metoda tuto informaci návrháři sdělit.

Takto zformulovaná úloha je v obecném případě pravděpodobně neřešitelná. Zaměříme-li se ale na případ, ve kterém má hledaný regulátor pouze dva nebo tři volné parametry¹, je možné využít techniky parametrické roviny. Při takovémto přístupu je řešení návrhové úlohy vyjádřeno regionem v parametrické rovině. Body tohoto regionu reprezentují regulátory, které splňují všechny zadané návrhové požadavky. Cílem této práce je rozšířit a algoritmizovat metodu navrženou v práci [16] a vytvořit příslušnou aplikaci v prostředí MATLAB pro návrh regulátorů typu PI a PID.

¹V praxi se jedná o nejčastěji řešený problém.

2 Úvod do regulace

Standardní forma uzavřené regulační smyčky je vyobrazena na obr. 2.1 a skládá se z 1DoF regulátoru $C(s)$ a regulovaného systému $P(s)$. Předpokládejme, že regulátor i systém jsou reprezentovány ryzími přenosovými funkcemi. Uzavření smyčky je provedeno přes zápornou zpětnou vazbu. Hlavním smyslem regulátoru $C(s)$ je generovat takové řízení, neboli akční veličinu u , které zaručí minimalizaci regulační odchylky e . Regulační odchylka e je rovna rozdílu mezi referenčním signálem r a řízenou veličinou y . Přesnost regulace by navíc neměla být ovlivněna exogenními signály d_i a d_o , které do smyčky zavádí určitou poruchu. Jako další musí být zohledněn fakt, že měření regulované veličiny y nemusí být vždy dokonalé, čímž se do smyčky zavádí další exogenní signál, šum n .



Obr. 2.1: Standardní forma regulační smyčky

2.1 PID regulátor

Obecně může být regulátor $C(s)$ ve formě ryzí přenosové funkce libovolného řádu. V průmyslových procesech je však více než 95% používaných regulátorů typu PID. Ve většině těchto případů je navíc derivační část algoritmu (D složka) vyřazena [2]. PID regulátor je složen z Proporcionální, Integrační a Derivační složky a realizuje zákon řízení v časové oblasti ve tvaru

$$u(t) = k_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right). \quad (2.1)$$

Tento vztah lze přes Laplaceovu transformaci a další úpravy převést na přenosovou funkci regulátoru ve tvaru

$$C_{PID}(s) = k_p \left(1 + \frac{1}{T_i s} + T_d s \right). \quad (2.2)$$

Jednotlivé složky PID regulátoru lze využít i v jiných kombinacích. V praxi lze nalézt i regulátory typu P, PI, PD a další možné variace.

Proporcionální člen regulátoru zavádí do smyčky zesílení k_p , které je přímo úměrné regulační odchylce e . Zvyšováním hodnoty zesílení k_p se zvyšuje i rychlost odezvy uzavřené smyčky a naopak se snižuje její robustnost ve stabilitě. Samotný P regulátor by pro statické systémy nezaručoval nulovou regulační odchylku e , proto regulátor obsahuje integrační člen, který je úměrný integrálu regulační odchylky e .

Integrační člen závisí na hodnotě integrační časové konstanty T_i , ale také na velikosti zesílení k_p . Zmenšováním velikosti T_i se obecně zvyšuje přeregulování. Tyto veličiny lze sjednotit do jedné proměnné $k_i = \frac{k_p}{T_i}$, která je označována jako integrační zesílení.

Derivační člen regulátoru umožňuje predikovat budoucí chování regulované veličiny a používá se především pro systémy s rychlou dynamikou. Tento člen zavádí do algoritmu derivační časovou konstantu T_d . Jejím zvětšováním lze snížit přeregulování. Stejně jako u integračního članku také zde platí závislost na zesílení k_p . Sjednocením obou proměnných vznikne derivační zesílení $k_d = k_p T_d$. Derivační část regulátoru v tomto tvaru představuje ideální případ, který je ale pro praktickou realizaci (z důvodu vysoké citlivosti na vysokofrekvenční šum) silně problémový. Derivační část je proto potřeba rozšířit o filtrační člen typu dolní propust, což lze provést různými způsoby. Běžnou praxí je využití filtru prvního řádu, který rozšíří derivační člen do tvaru

$$D(s) = \frac{k_d s}{\tau s + 1}. \quad (2.3)$$

2.2 Model řízeného systému

Pro kvalitně navržený regulátor je důležitý i vhodně zvolený model řízeného systému, který by dostatečně přesně popisoval jeho dynamické vlastnosti důležité pro návrh regulátoru. Takovýto model lze získat buď na základě matematicko-fyzikálního modelování, nebo pomocí experimentální identifikace na základě naměřených I/O dat. V závislosti na typu zvolené návrhové metody lze pak využívat různé parametrické modely, které mají předem danou strukturu a jsou specifikovány několika charakteristickými čísly.¹

¹Mnoho stabilních dynamických procesů lze popsat modelem prvního nebo druhého řádu s dopravním zpožděním (FOPDT, SOPDT).

2.3 Průřez návrhovými metodami

Od 30. let 20. století se postupně začaly vyvíjet různé empirické ladící metody, tzv. tuning rules, pro nastavení parametrů regulátoru typu P, PI a PID. Tyto metody jsou založeny na odečtu určitých charakteristických čísel z časových či frekvenčních charakteristik zkoumaného procesu. Na základě těchto informací lze pak, s pomocí empiricky získaných vztahů, vypočítat hodnoty parametrů uvažovaného regulátoru. Příkladem takového přístupu je např. Ziegler-Nicholsova frekvenční metoda.

Jako další pak vznikaly návrhové metody, obecně označované jako loopshaping, které byly založeny na tvarování frekvenčního přenosu. Získané řešení pak muselo splňovat určité požadavky na kvalitu řízení a robustnost.

Od 60. let se postupně začaly vyvíjet metody tzv. moderní teorie řízení, které cílily na nalezení optimálního regulátoru. Při optimálním řízení je návrhová úloha formulována jako optimalizační problém s cílem minimalizovat nějakou kritériální funkci. Příkladem takového přístupu je LQR (Linear Quadratic Regulator) problém, jehož výsledkem je kvadraticky optimální regulátor ve formě stavové zpětné vazby. LQR metoda zároveň zaručuje robustnost ve stabilitě vůči neurčitosti řízeného systému a tím pádem i stabilizující řešení. V případě, že stav není přímo měřitelný, musí být regulační smyčka rozšířena o Kalmanův filtr, který poskytuje odhad stavu systému. Tento přístup je označován jako LQG (Linear Quadratic Gaussian) problém. Současně rozšířením regulační smyčky o rekonstruktor stavu dochází ke ztrátě robustnosti. Tento problém lze částečně potlačit LTR (Loop Transfer Recovery) procedurou, která se snaží úlohu upravit tak, aby se co nejvíce blížila vlastnostem LQR problému.

V rámci optimalizačních úloh není explicitně uvažována žádná neurčitost. Takový přístup ale nemusí být vždy vhodný. Jak bylo uvedeno v [3], je nutné mít vždy na paměti, že model řízeného systému (bez ohledu na to jak je podrobný) není zcela přesnou kopií reálného fyzického systému. Vždy se bude jednat pouze o aproximaci reálné dynamiky a vždy bude obsahovat určitou neurčitost. Typickým zdrojem takovéto neurčitosti může být nemodelovaná (obvykle vysokofrekvenční) dynamika a zanedbaná nelinearita systému. Tyto chyby pak mohou nepříznivě ovlivnit stabilitu a kvalitu řízení.

Řešením takovýchto typů úloh se zabývá robustní teorie řízení, která pracuje s explicitně definovanou neurčitostí. Robustnost je pak taková vlastnost zpětnovazebního systému, kdy tato vlastnost platí pro celou množinu systémů, a aplikované řízení je pak robustní vzhledem k dané množině². V 80. letech se pak začal praktikovat nový přístup k návrhu robustního řízení, obecně využívající optimalizaci ve frekvenční oblasti

²Tato definice robustnosti platí především pro případ explicitně definované neurčitosti. V obecném kontextu lze robustnost chápat i v takovém smyslu, že řídicí systém je sestaven tak, aby splňoval zadané požadavky i při malých změnách v chování řízené soustavy.

s využitím H_2 a H_∞ omezení.

Ve všech doposud uvedených přístupech je předpokládáno, že řízený systém je lineární a navíc t-invariantní s neměnnou dynamikou. Pro t-variantní lineární systémy či nelineární systémy, u kterých lze rozlišit jejich chování do tzv. operačních stavů, lze využít principu zvaného „Gain scheduling“. Tento přístup vede na využití hned několika rozdílných nastavení parametrů regulátoru, mezi kterými se přechází v závislosti na aktuálním operačním stavu.

V případě, kdy informaci o proměnlivosti systému nelze dopředu určit a zároveň nelze jednoduše syntetizovat robustní regulátor, je nasnadě aplikovat řídicí systém, který je schopný samostatného ladění neboli adaptace. Jádrem každého takového systému je identifikační algoritmus, který zajišťuje poznávání řízeného systému. V 70. letech vykazovaly metody adaptivního řízení slibné výsledky a po jistou dobu představovaly hlavní směr vývoje v oblasti automatického řízení [12]. Kvůli složitější implementaci a požadavkům zaručujícím stabilitu však vývoj postupně přecházel do oblasti robustního řízení.

2.4 Návrhová omezení a požadavky

Při návrhu regulátoru musí návrhář spolu s uvažovaným systémem definovat i omezení, které musí výsledné řešení splňovat. Tyto podmínky se mohou vztahovat jak na kvalitu regulace v časové nebo frekvenční oblasti, tak i na robustnost. V procesním řízení jsou typicky zadávány požadavky na odregulování poruchy³. V systémech, kde je primárním úkolem řízení pohybu mechatronických prvků, jsou naopak zadávány požadavky na sledování referenční hodnoty⁴. Pro ohodnocení průběhu regulace v časové oblasti jsou často používána integrální kritéria kvality, která hodnotí časový průběh regulační odchylky. Těmito kritérii jsou:

- Integrated Error

$$\text{IE} = \int_0^{\infty} e(t) dt, \quad (2.4)$$

- Integrated Absolute Error

$$\text{IAE} = \int_0^{\infty} |e(t)| dt, \quad (2.5)$$

³ Tento princip je označován jako problém regulace a předpokládá se při něm nulová nebo konstantní hodnota referenčního signálu. Hlavním požadavkem je pak přesné odregulování poruchy.

⁴ Tento přístup je označován jako servo problém, kdy se předpokládá nulová hodnota poruchy a je požadováno přesné sledování referenčního signálu regulovaným výstupem.

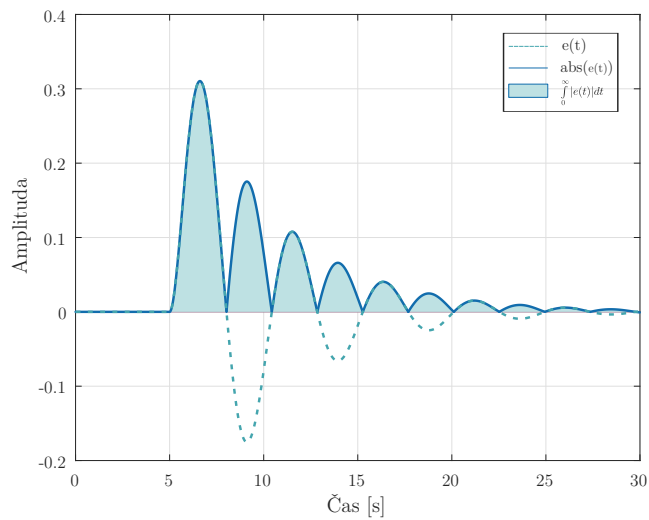
- Integrated Square Error

$$\text{ISE} = \int_0^{\infty} e(t)^2 dt, \quad (2.6)$$

- Integrated Time-weighted Absolute Error

$$\text{ITAE} = \int_0^{\infty} t|e(t)|dt. \quad (2.7)$$

Tyto vztahy jsou v návrhových optimalizačních metodách často používány jako kritériální funkce. Vezmeme-li v úvahu využití IAE kritéria, je cílem minimalizovat plochu pod křivkou absolutní hodnoty regulační odchylky e , viz obr. 2.2.



Obr. 2.2: Průběh regulační odchylky e

Mezi další obecné požadavky také patří, aby výsledné řešení zaručovalo dostatečnou robustnost uzavřené smyčky, což lze blíže specifikovat pomocí klasických indexů bezpečnosti⁵. Cílem je, aby frekvenční charakteristika otevřené smyčky byla v komplexní rovině dostatečně vzdálená od kritického bodu $[-1, 0j]$.

S regulační smyčkou na obr. 2.1 jsou spojeny i následující přenosové funkce

$$\begin{aligned} S(s) &= \frac{Y(s)}{D_o(s)} = \frac{1}{1 + C(s)P(s)}, & T(s) &= \frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \\ S_C(s) &= \frac{U(s)}{R(s)} = \frac{C(s)}{1 + C(s)P(s)}, & S_P(s) &= \frac{Y(s)}{D_i(s)} = \frac{P(s)}{1 + C(s)P(s)}. \end{aligned} \quad (2.8)$$

⁵Bezpečnost v zesílení, bezpečnost ve fázi a bezpečnost ve stabilitě.

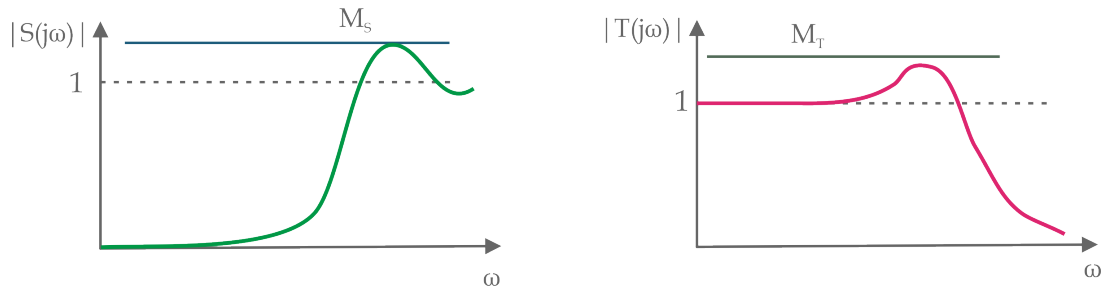
Požadavky na regulační smyčku lze převést na požadavky na frekvenční přenos jednotlivých funkcí (2.8) ve tvaru

$$|S_*(j\omega)| \leq M_{S_*}, \quad \forall \omega \in \langle 0, +\infty \rangle. \quad (2.9)$$

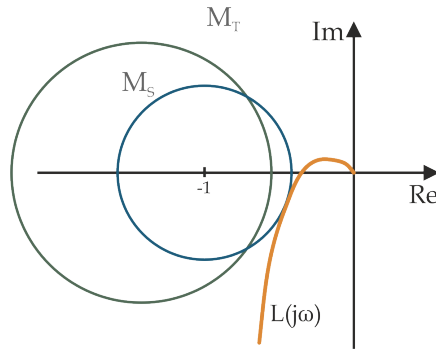
Využitím systémové ∞ -normy lze tento vztah přepsat do tvaru

$$\|S_*(s)\|_\infty \leq M_{S_*}, \quad (2.10)$$

kde $S_*(s)$ je libovolná přenosová funkce z (2.8). Specifikací takového omezení je tedy kladen požadavek na maximální hodnotu amplitudové frekvenční charakteristiky dané funkce, která je pak shora omezena přímkou odpovídající hodnotě M_{S_*} . Pro citlivostní funkci $S(s)$ a komplementární citlivostní funkci $T(s)$ mají takováto omezení geometrickou interpretaci i v komplexní rovině, kde vytváří tzv. M_S a M_T kružnice. Tyto vlastnosti jsou zachyceny na obr. 2.3 a obr. 2.4.



Obr. 2.3: Amplitudová frekvenční charakteristika pro $S(s)$ a $T(s)$

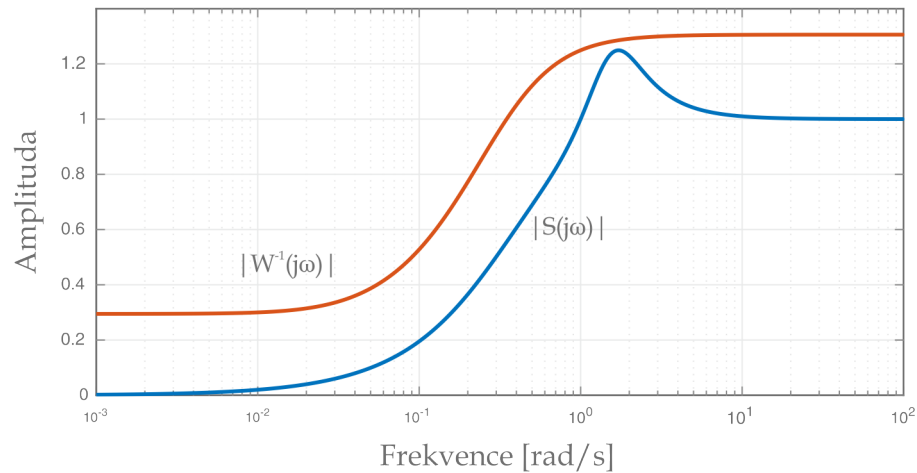


Obr. 2.4: M_S a M_T kružnice

V určitých situacích by bylo výhodné omezit některou z funkcí na konkrétní frekvenci, či na nějakém omezeném frekvenčním rozsahu. Takovouto podmínku lze specifikovat přes obecný požadavek ve tvaru

$$\|W(s)S_*(s)\|_\infty \leq M_{S_*}, \quad (2.11)$$

kde $W(s)$ je váhová funkce, která umožňuje frekvenčně tvarovat návrhové požadavky. Vztah citlivostní funkce $S(s)$ a váhové funkce $W(s)$ pro $M_S = 1$ je zachycen na obr. 2.5. Podle požadavku (2.11) musí platit, že amplitudová frekvenční charakteristika charakteristika $|S(s)|$ je menší než amplitudová frekvenční charakteristika $|W(s)^{-1}|$.



Obr. 2.5: Vztah mezi citlivostní funkcí $S(s)$ a váhovou funkcí $W(s)$

Otázkou ale zůstává, jak jednotlivé požadavky zanechat do návrhových omezení. Požadavek na odregulování poruchy nebo na robustnost systému v uzavřené smyčce lze specifikovat přes tvar citlivostní funkce $S(s)$. Typická hodnota požadovaného maxima citlivostní funkce se pohybuje v intervalu $\langle 1.0, 2.0 \rangle$. Amplitudová frekvenční charakteristika citlivostní funkce má tvar hornofrekvenční propusti. Pro nízké frekvence je tedy snižován vliv poruchy d_o . Nepříznivý důsledek tvarování citlivostní funkce $S(s)$ je tzv. efekt vodní postele, kdy se vlivem pokusů potlačit poruchy na určitých frekvencích zesílí poruchy na jiných frekvencích.

Pro omezení kmitavosti uzavřené smyčky na rezonanční frekvenci lze zase využít komplementární citlivostní funkci $T(s)$. Tato funkce má tvar dolnofrekvenční propusti a popisuje schopnost sledovat referenční signál regulovanou veličinou. Nejlepší sledování je dosaženo na nízkých frekvencích pro $T(j\omega) = 1$. Typická hodnota požadovaného maxima této funkce se pohybuje v intervalu $\langle 1.0, 1.6 \rangle$.

Citlivostní funkce $S(s)$ a komplementární citlivostní funkce $T(s)$ se vzájemně doplňují a pro dosažitelnou kvalitu regulace a potlačení poruch platí následující omezující vztah

$$S(s) + T(s) = 1. \quad (2.12)$$

Tvarování těchto přenosů pak musí být založeno na kompromisu mezi jednotlivými požadavky.

Zbylé citlivostní funkce pak slouží pro omezení vlivu poruchy na vstupu systému, což popisuje průběh funkce $S_P(s)$ a modifikaci požadavků na přenos z referenčního signálu na akční veličinu. Ty jsou dány funkcí $S_C(s)$.

2.5 Systémové normy

Klíčovou roli v optimalizačním přístupu moderní teorie řízení hrají systémové normy, které slouží ke kvantifikaci velikosti signálů a zesílení dynamických systémů. Nejvíce používanými je H_2 a H_∞ norma, které mají jasnou fyzikální interpretaci.

H_2 normu lze fyzikálně interpretovat jako celkovou energii impulsní odezvy systému, což je ve frekvenční oblasti ekvivalentní k průměrnému zesílení přenosu přes všechny frekvence. Ve stochastické interpretaci pak určuje varianci výstupu v ustáleném stavu při působení náhodného bílého šumu na vstupu [7]. Často se používá k hodnocení kvality řízení. Velikost normy lze vypočítat analyticky podle vztahu

$$\|H\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{+\infty} |H(j\omega)|^2 d\omega \right)^{1/2} \quad (2.13)$$

H_∞ norma zase představuje největší zesílení systému přes všechny frekvence a má velké využití v teorii robustního řízení. Velikost normy odpovídá vztahu

$$\|H\|_\infty = \sup_{\omega} |H(j\omega)| \quad (2.14)$$

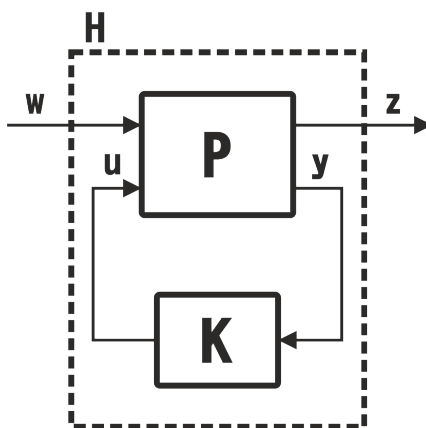
a lze ji stanovit pouze numerickou cestou.

3 Metody pro návrh regulátoru na základě H_2 a H_∞ optimalizace

Existuje mnoho metod a cest pro řešení H_2 a H_∞ optimalizačních problémů. Většina těchto metod vychází ze specifické formulace problému. S touto formulací lze pak již snadněji manipulovat.

3.1 Obecná H_2 a H_∞ optimalizace

Předpokládejme, že lze strukturu řízené soustavy popsat následujícím schématem,



Obr. 3.1: Formalismus pro řešení H_2 a H_∞ optimalizačních úloh

kde P představuje zobecněný systém a K zpětnovazební kompenzátor¹. Jednotlivé přechody popisují:

- Zobecněné vstupy w , což jsou vnější signály, které působí na regulační smyčku. Jedná se např. o referenční signál, poruchy nebo šum měření.
- Penalizované výstupy z jsou veličiny, jejichž závislost na w chceme v určitém smyslu minimalizovat. Je to např. regulační odchylka, chyba rekonstruktoru nebo akční zásah.
- Měřené výstupy y představují signály využité pro zpětnou vazbu.
- Vstupy u jsou akční veličiny, kterými regulátor působí na řízený systém.

¹Do této podoby lze analyzovanou strukturu řízené soustavy převést přes lineární zlomkové transformace (LFT – Linear Fractional Transformational).

Celou řadu praktických úloh návrhu systému automatického řízení lze pak formulovat jako optimalizační problém nalezení kompenzátoru K , který vnitřně stabilizuje zpětnovazební smyčku a současně minimalizuje vhodnou systémovou normu vyjadřující vliv vnějších signálů na zobecněné výstupy, kterou je žádoucí co nejvíce omezit [7]. Tato norma by měla být realizovatelná a měla by mít v oblasti řízení nějakou rozumnou fyzikální interpretaci. Zároveň pro ni musí existovat obecné řešení příslušné optimalizační úlohy. Nejčastěji používané normy jsou H_2 a H_∞ , či jejich kombinace. Optimalizace se pak provádí pro zobecněný systém P , do kterého se obvykle zahrnují vhodné váhové funkce umožňující formulovat i frekvenčně závislé návrhové požadavky a ladit samotný regulátor.

Pro H_2 problém existuje podle [4] jednoznačné řešení, které lze vypočítat numerickým řešením dvojice Riccatiových rovnic. V tomto místě lze poznamenat, že LQG regulátor je speciálním případem obecného H_2 problému.

Oproti tomu u H_∞ problému nelze najít řešení analyticky. Jsou známá pouze řešení suboptimální, která lze podle [4] numericky vypočítat opět řešením dvojice algebraických Riccatiových rovnic. Optimální regulátor je pak nalezen postupným zmenšováním γ , dokud lze vypočítat řešení [6].

V praxi je nejvíce používán tzv. Mixed sensitivity problém, který má jasnou fyzikální interpretaci a umožňuje relativně snadno formulovat návrhové požadavky ve frekvenční oblasti [7]. Tento přístup spočívá v tvarování citlivostní funkce S a komplementárně citlivostní funkce T , či akčního zásahu KS , k čemuž se využívají frekvenčně závislé váhové funkce. Je-li uvažován S/T smíšený citlivostní problém, pak výsledkem H_∞ optimalizace je regulátor splňující podmínku

$$\|H\|_\infty = \left\| \begin{bmatrix} W_1 S \\ W_2 T \end{bmatrix} \right\|_\infty \leq \gamma, \quad (3.1)$$

kde γ představuje minimální hodnotu, kterou je možné pro daný systém a zvolené váhové funkce dosáhnout.

Pro H_∞ optimalizační problém a pro smíšený H_2/H_∞ problém existuje efektivní metoda, která je implementována v prostředí MATLAB, kde ji lze nalézt pod názvem `hinfstruct` a je založena na [1].

3.2 H_∞ optimalizace pro návrh PI a PID regulátorů

Obecná řešení vedou na obecný regulátor často vysokého řádu. Metod pro návrh PI nebo PID regulátoru, které jsou založeny na optimalizačním přístupu, prozatím není příliš mnoho, ale postupně se stávají hlavní oblastí výzkumu [6]. Zásadní přístupy pro návrh regulátoru nízkého řádu (PI nebo PID) pak představují následující metody:

3.2.1 MIGO (M constrained Integral Gain Optimization)

MIGO je metoda pro návrh PI regulátoru a je založena na numerické minimalizaci IE kritéria pro odregulování působící poruchy na vstupu s omezením na maximální hodnotu citlivostní funkce $S(s)$. Kritérium IE lze nahradit ekvivalentním požadavkem na maximalizaci integračního zesílení k_i . V [2] bylo totiž ukázáno, že

$$IE = \frac{1}{k_i}. \quad (3.2)$$

Návrhová úloha je pak formulována jako optimalizační problém, kdy je úkolem nalézt takové parametry regulátoru, které maximalizují integrační zesílení k_i . Jako omezení je kladen požadavek na stabilitu uzavřené smyčky a to, že Nyquistova křivka $L(j\omega) = C(j\omega)P(j\omega)$ splňuje podmínku

$$|1 + L(j\omega)| \geq \frac{1}{M_S}, \quad (3.3)$$

kde M_S je návrhový parametr [18]. Návrhová omezení na citlivostní funkci $S(s)$ pak v parametrické rovině $k_p - k_i$ vytváří ohraničenou oblast, která představuje množinu parametrů, pro které jsou zadaná omezení splněna. Hranice této oblasti je získána jako obálka frekvenčně závislých elips. V [17] byla tato metoda použita pro vytvoření ladícího pravidla pro PID regulátor, kdy je řízený proces aproximován FOPDT modelem. Vzniklá metoda se označuje jako AMIGO (Approximate MIGO) [6].

3.2.2 SWORD (Software-based optimal robust design)

Jedná se o softwarový nástroj sloužící pro návrh robustního PI a PID regulátoru. Návrhová úloha je definována jako optimalizační problém s kritériální funkcí ve tvaru

$$\int_0^{\infty} |e(t)| dt \quad (3.4)$$

a omezeními pro H_∞ normu citlivostní funkce $S(s)$ a komplementární citlivostní funkce $T(s)$ ve tvaru

$$|S(j\omega)| \leq M_S, \quad (3.5)$$

$$|T(j\omega)| \leq M_t, \quad (3.6)$$

$$|S(j\omega^s)| = M_S, \quad (3.7)$$

$$|T(j\omega^t)| = M_t, \quad (3.8)$$

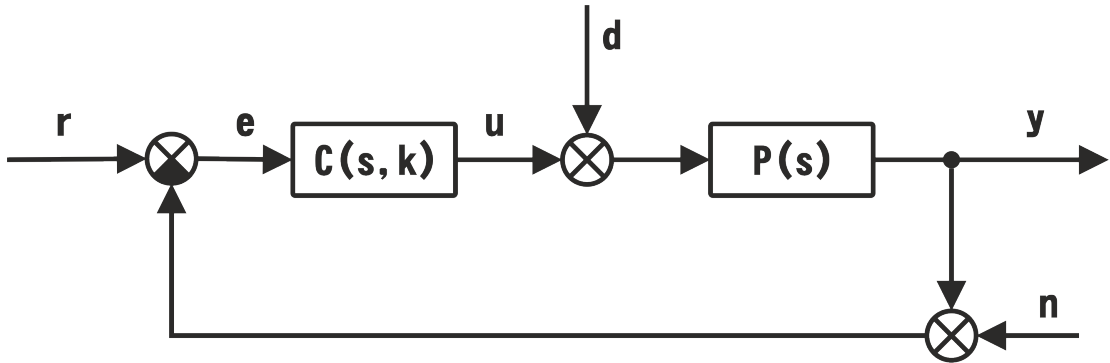
kde $\omega \in \langle 0, +\infty \rangle$ a ω^s jsou frekvence, pro které je frekvenční charakteristika otevřené regulační smyčky tečnou k M_S kružnici. Stejně tak je tomu pro ω^t a M_T kružnici. Podmínky (3.7), (3.8) nemusí být splněny zároveň. Pro řešení stačí, když bude platit

jen jedna z nich. Jako další musí být také zaručeno, že uzavřená regulační smyčka je pro výsledné řešení stabilní. Protože je optimalizační problém s kriteriální funkcí (3.4) nekonvexního charakteru a může tedy existovat více lokálních minim, je obtížné garantovat, že výsledné řešení představuje globální minimum. Tento problém lze vyřešit využitím Nelder-Meadovy (NM) metody, která obecně slouží k nalezení minima nebo maxima kriteriální funkce ve vícerozměrném prostoru [6].

Další metoda vychází z [16] a představuje efektivní způsob návrhu optimálního PI regulátoru a suboptimálního PID regulátoru. Právě tato metoda byla využita ve vytvořené aplikaci a je detailněji představena v následující kapitole.

4 Návrh regulátoru s omezenou strukturou metodou H_∞

Uvažujme regulační smyčku v následujícím tvaru



Obr. 4.1: Regulační smyčka

a k ní přidružené citlivostní funkce

$$S(s, \mathbf{k}) = \frac{E(s)}{R(s)} = \frac{1}{1 + C(s, \mathbf{k})P(s)}, \quad (4.1)$$

$$T(s, \mathbf{k}) = \frac{Y(s)}{R(s)} = \frac{C(s, \mathbf{k})P(s)}{1 + C(s, \mathbf{k})P(s)}, \quad (4.2)$$

$$S_C(s, \mathbf{k}) = \frac{U(s)}{N(s)} = \frac{C(s, \mathbf{k})}{1 + C(s, \mathbf{k})P(s)}, \quad (4.3)$$

$$S_P(s, \mathbf{k}) = \frac{Y(s)}{D(s)} = \frac{P(s)}{1 + C(s, \mathbf{k})P(s)}. \quad (4.4)$$

Předpokládejme, že přenos navrhovaného regulátoru $C(s, \mathbf{k})$ je racionální funkce s omezenou strukturou¹ a že je tento přenos lineárně závislý pouze na dvou parametrech k_p, k_i , $\mathbf{k} \triangleq [k_p, k_i]^T \in \mathbb{R}^2$. Dále předpokládejme, že $P(s)$ je daná ryzí racionální přenosová funkce. Lineární závislost přenosu $C(s, \mathbf{k})$ na parametrech k_p, k_i znamená, že koeficienty čitatele i jmenovatele racionální přenosové funkce $C(s, \mathbf{k})$ závisí lineárně na k_p, k_i . Základní strukturální vlastnosti jsou již specifikovány, načež můžeme přejít k definici návrhové úlohy.

¹Např. PI nebo PID regulátor.

4.1 Návrhová úloha s omezením typu H_∞

Úkolem je nalézt všechny regulátory s danou omezenou strukturou $C(s, \mathbf{k})$, které splňují následující podmínky²:

- (i) uzavřená smyčka je stabilní,
- (ii) přenosová funkce $H(s, \mathbf{k})$ je stabilní,
- (iii) $\|H(s, \mathbf{k})\|_\infty \leq \gamma$.

Obecně uvažujeme, že $H(s, \mathbf{k}) = W(s)S_*(s, \mathbf{k})$, kde $S_*(s, \mathbf{k})$ je libovolná funkce ze čtyř citlivostních funkcí (4.1-4.4) přidružených k zpětnovazební smyčce z obr. 4.1 a $W(s)$ je stabilní váhová funkce. Protože je funkce $H(s, \mathbf{k})$ racionální přenosovou funkcí, lze ji také zapsat jako

$$H(s, \mathbf{k}) = \frac{H_n(s, \mathbf{k})}{H_d(s, \mathbf{k})}, \quad (4.5)$$

kde $H_n(s, \mathbf{k})$ a $H_d(s, \mathbf{k})$ jsou polynomy s reálnými koeficienty. Množinu všech regulátorů parametrizovanou parametry $k_p, k_i, \mathbf{k} \triangleq [k_p, k_i]^T \in \mathbb{R}^2$ a vyhovující podmínkám (i), (ii) a (iii) návrhové úlohy pak označme \mathcal{K} , tj.

$$\mathcal{K} = \{\mathbf{k} \in \mathbb{R}^2 : \text{platí podmínky (i), (ii) a (iii)}.\}$$

Regulátor $C(s, \mathbf{k})$ pro $\mathbf{k} \in \mathcal{K}$ budeme nazývat **H_∞ regulátorem** a oblast \mathcal{K} v parametrické rovině **H_∞ regionem**. Oblast stability uzavřené smyčky v parametrické rovině regulátoru označme \mathcal{K}_S .

4.1.1 Prototypový algoritmus

Základní myšlenka tohoto algoritmu vychází z techniky Neimarkova D-rozkladu [13, 14], která byla v [9, 10] využita spolu s H_∞ kritériem pro návrh obecného regulátoru nízkého řádu. Základní princip určení množiny \mathcal{K} je v [10] založen na následující větě:

Věta 1 *Hranice množiny \mathcal{K} je obsažena ve sjednocení řešení dvou soustav*

$$\begin{cases} H_n(j\omega, \mathbf{k}) = 0 \\ H_d(j\omega, \mathbf{k}) = 0, \end{cases} \quad \omega \in \langle 0, +\infty \rangle \quad (4.6)$$

$$\begin{cases} |H(j\omega, \mathbf{k})|^2 = \gamma^2 \\ \frac{\partial |H(j\omega, \mathbf{k})|^2}{\partial \omega} = 0, \end{cases} \quad \omega \in \langle 0, +\infty \rangle \quad (4.7)$$

²V našem uvažovaném případě podmínky (i) a (ii) vyjadřují totéž, poněvadž $H(s, \mathbf{k})$ je zvolena tak, že její jmenovatel je charakteristický polynom uzavřené smyčky.

a tři rovnic

$$|H(0, \mathbf{k})| = \gamma, \quad (4.8)$$

$$|H(j\infty, \mathbf{k})| = \gamma, \quad (4.9)$$

$$h_d(\mathbf{k}) = 0, \quad (4.10)$$

kde h_d je koeficient u nejvyšší mocniny s v polynomu $H_d(s, \mathbf{k})$.

4.1.2 Řešení návrhové úlohy

Využitím moderních symbolických softwarových nástrojů lze tento přístup dále modifikovat [16] a získat tak řešení návrhové úlohy, kde lze konstrukci množiny \mathcal{K} provést podle následujících kroků:

Krok 1: Podle věty 1 nalezneme všechny body v parametrické rovině regulátoru $C(s, \mathbf{k})$ podezřelé z možnosti být hranicí množiny \mathcal{K} . Tyto body leží na konečném počtu křivek definovaných rovnicemi (4.6-4.10) ve větě 1. Množinu všech těchto křivek označíme symbolem \mathcal{B} .

Krok 2: Z křivek množiny \mathcal{B} , nalezených v kroku 1 vybereme ty jejich části (úseky), které splňují podmínku (ii) návrhové úlohy a pro které platí, že $\|H(s, \mathbf{k})\|_\infty = \gamma$. Množinu vybraných úseků označíme symbolem \mathcal{B}^* .

Krok 3: Úseky křivek množiny \mathcal{B}^* , nalezené v kroku 2, tvoří hranici jednoho nebo několika regionů v parametrické rovině regulátoru. Takto získané regiony mohou být omezené (tj. úseky křivek z množiny \mathcal{B}^* tvoří celou hranici regionu) nebo neomezené (tj. úseky křivek z množiny \mathcal{B}^* tvoří pouze část hranice regionu). Sjednocení všech takto získaných regionů určuje hledanou množinu \mathcal{K} všech H_∞ regulátorů splňujících podmínku návrhové úlohy.

4.2 Implementace návrhového algoritmu

V předchozí části byly uvedeny základní souvislosti a řešení pro návrh regulátoru s omezenou strukturou. V této práci se omezíme na regulátory typu PI ve tvaru

$$C_{PI}(s, \mathbf{k}) = k_p + \frac{k_i}{s} \quad (4.11)$$

a PID ve tvaru

$$C_{PID}(s, \mathbf{k}) = k_p + \frac{k_i}{s} + \frac{k_d s}{\tau s + 1}. \quad (4.12)$$

Na začátku návrhového procesu je pro PID regulátor navíc nutné zvolit hodnotu derivačního zesílení k_d a časové konstanty τ .

Uvedená návrhová metoda je obecně aplikovatelná na širokou třídu systémů. V rámci implementovaného řešení je předpokládán systém ve formě racionální přenosové funkce $P(s)$. Tento systém může obsahovat i astatismus, maximálně ale 1. řádu. Pro systémy s astatismem vyššího řádu je totiž obecně složitější nalézt stabilizující regulátor a do přenosu otevřené smyčky by musel být zařazen derivační členek.

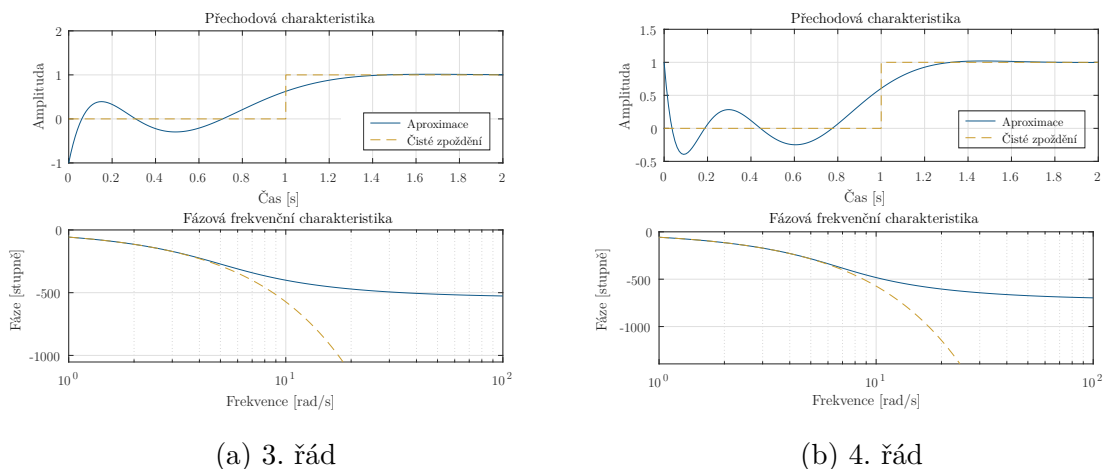
V regulačních obvodech se také často vyskytuje dopravní zpoždění. I s tímto případem je v implementovaném řešení počítáno. Předpokládejme systém popsany přenosovou funkcí $P'(s)$ a dopravní zpoždění na jeho vstupu. Samotné dopravní zpoždění je popsáno neracionální přenosovou funkcí e^{-Ds} , kde D je příslušná prodleva. Výsledná soustava

$$P(s) = P'(s)e^{-Ds} \quad (4.13)$$

tak obsahuje exponenciální funkci, která představuje systém nekonečného řádu, což značně komplikuje proces návrhu. Případné dopravní zpoždění je tedy nutné aproximovat členem konečného řádu, k čemuž je v této práci využita Padého aproximace. Tato metoda aproximuje čisté dopravní zpoždění racionální funkcí a platí, že

$$e^{-Ds} \approx \frac{\sum_{i=0}^n p_i (Ds)^i}{\sum_{i=0}^n q_i (Ds)^i}, \quad (4.14)$$

kde $p_i = (-1)^i \frac{(2n-i)!n!}{(2n)!i!(n-i)!}$, $q_i = \frac{(2n-i)!n!}{(2n)!i!(n-i)!}$ a n je řád aproximace [11]. Kromě případné hodnoty dopravního zpoždění musí být návrhářem stanoven i řád Padého aproximace. Kvalitu aproximace by bylo možné ještě zvýšit vložení dynamického článku typu dolní propust do série s dopravním zpožděním. Docílí se tak tlumení vyšších frekvencí, na kterých se aproximace silně odklání od skutečného průběhu. Na obr. 4.2 jsou uvedeny přechodové a fázové frekvenční charakteristiky dopravního zpoždění pro 3. a 4. řád.



Obr. 4.2: Padého aproximace čistého dopravního zpoždění

4.2.1 Řešení soustav a rovnic z věty 1

Pro zvolenou citlivostní funkci (4.1-4.4) a návrhový parametr γ lze řešením soustav (4.6) a (4.7) a rovnic (4.8) a (4.9) z věty 1 získat množinu křivek \mathcal{B} , které pro určité \mathbf{k} a ω dosahují maxima funkce $|H(j\omega)|$, jenž je rovno parametru γ . Rovnice (4.10) pak popisuje možnou ztrátu stability uzavřené smyčky.

Řešení soustavy 4.6 Tato soustava má řešení pouze pro frekvenci $\omega = 0$, kdy

$$k_i = 0 \quad \text{a} \quad k_p \in \mathbb{R}. \quad (4.15)$$

Křivka množiny \mathcal{B} vytvořená touto soustavou je tedy osou k_p parametrické roviny.

Řešení soustavy 4.7 Ve většině případů je velká část parametrických křivek množiny \mathcal{B} získána výpočtem právě této soustavy. Výsledkem je pak obecné řešení ve tvaru

$$\begin{cases} k_i(\omega) = f_{k_i}(\omega, x_r, \gamma), \\ k_p(\omega) = f_{k_p}(\omega, x_r, \gamma, A, B, A_1, B_1, w, w_1, k_d, \tau), \end{cases} \quad \omega \in \langle 0, +\infty \rangle \quad (4.16)$$

kde

$$\begin{aligned} A &\triangleq \operatorname{Re}(P(j\omega)), & A &= A(\omega), & A_1 &\triangleq \frac{dA}{d\omega}, & A_1 &= A_1(\omega) \\ B &\triangleq \operatorname{Im}(P(j\omega)), & B &= B(\omega), & B_1 &\triangleq \frac{dB}{d\omega}, & B_1 &= B_1(\omega), \\ w &\triangleq |W(j\omega)|^2, & w &= w(\omega), & w_1 &\triangleq \frac{dw}{d\omega}, & w_1 &= w_1(\omega) \end{aligned} \quad (4.17)$$

a $x_r = x_r(\omega)$, $r = 1, \dots, 4$ jsou reálná řešení kvartické rovnice

$$ax^4 + bx^3 + cx^2 + dx + e = 0. \quad (4.18)$$

Pro koeficienty příslušného polynomu opět obecně platí, že

$$\begin{aligned} a &= f_a(\omega, \gamma, A, B, A_1, B_1, w, w_1, k_d, \tau), & a &\in \mathbb{R}, a = a(\omega), \\ b &= f_b(\omega, \gamma, A, B, A_1, B_1, w, w_1, k_d, \tau), & b &\in \mathbb{R}, b = b(\omega), \\ c &= f_c(\omega, \gamma, A, B, A_1, B_1, w, w_1, k_d, \tau), & c &\in \mathbb{R}, c = c(\omega), \\ d &= f_d(\omega, \gamma, A, B, A_1, B_1, w, w_1, k_d, \tau), & d &\in \mathbb{R}, d = d(\omega), \\ e &= f_e(\omega, \gamma, A, B, A_1, B_1, w, w_1, k_d, \tau), & e &\in \mathbb{R}, e = e(\omega). \end{aligned} \quad (4.19)$$

Pro regulátor typu PI jsou hodnoty derivačního zesílení k_d a časové konstanty τ nulové. Jako ukázkou lze uvést jedno z jednodušších obecných řešení pro případ PI regulátoru, citlivostní funkce $S(s)$ a váhové funkce $W(s) = 1$.

$$\begin{aligned}
k_i(\omega) &= \frac{x_r \omega}{\gamma} \\
k_p(\omega) &= \frac{1}{\left(\omega \gamma^2 (2ABB_1 + A^2 A_1 - A_1 B^2) \right) \left(x_r^2 (A^2 + B^2)^2 + \right.} \\
&\quad \left. + x_r \gamma (\omega (-B_1 A^2 + 2BA_1 A + B^2 B_1) + B(A^2 + B^2)) + \right. \\
&\quad \left. + \omega (\gamma^2 - 1) (BB_1 + AA_1) \right)}
\end{aligned} \tag{4.20}$$

Pro řešení rovnic (4.8) (4.10) předpokládejme, že přenosovou funkci zkoumaného systému lze zapsat ve tvaru

$$P(s) = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}. \tag{4.21}$$

Dále budeme uvažovat dvě možné volby váhové funkce a to

$$W_1(s) = 1, \quad W_2(s) = \frac{e_q s^q + \dots + e_1 s + e_0}{f_p s^p + \dots + f_1 s + f_0}. \tag{4.22}$$

Řešení pro jednotlivé rovnice a případy pak vypadá následovně.

Řešení rovnice 4.8 Platí, že $C(s, \mathbf{k}) \in \{C_{PI}(s, \mathbf{k}), C_{PID}(s, \mathbf{k})\}$.³

I. $W(s) = W_1(s)$

$$S(s, \mathbf{k}): |H(0, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

$$T(s, \mathbf{k}): |H(0, \mathbf{k})| = 1 \Rightarrow \emptyset \text{ pro } \gamma \neq 1$$

$$S_C(s, \mathbf{k}): |H(0, \mathbf{k})| = \left| \frac{a_0}{b_0} \right| \Rightarrow \emptyset \text{ pro } \gamma \neq \left| \frac{a_0}{b_0} \right|$$

$$S_P(s, \mathbf{k}): |H(0, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

II. $W(s) = W_2(s)$

$$S(s, \mathbf{k}): |H(0, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

$$T(s, \mathbf{k}): |H(0, \mathbf{k})| = \left| \frac{e_0}{f_0} \right| \Rightarrow \emptyset \text{ pro } \gamma \neq \left| \frac{e_0}{f_0} \right|$$

$$S_C(s, \mathbf{k}): |H(0, \mathbf{k})| = \left| \frac{e_0 a_0}{f_0 b_0} \right| \Rightarrow \emptyset \text{ pro } \gamma \neq \left| \frac{e_0 a_0}{f_0 b_0} \right|$$

$$S_P(s, \mathbf{k}): |H(0, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

Z předchozích výsledků je zřejmé, že nastávají pouze singulární případy, které je možné zanedbat.

³Symbolem \emptyset je označen případ, kdy příslušná rovnice nemá žádné řešení.

Řešení rovnice 4.9

I. $C(s, \mathbf{k}) = C_{PI}(s, \mathbf{k})$

i. $W(s) = W_1(s)$

$$S(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 1 \Rightarrow \emptyset \text{ pro } \gamma \neq 1$$

$$T(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

$$S_C(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = |k_p| \Rightarrow k_{p1,2} = \pm\gamma, \quad k_{i1,2} \in \mathbb{R}$$

$$S_P(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

ii. $W(s) = W_2(s)$

$$S(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = \left| \frac{e_p}{f_p} \right| \Rightarrow \emptyset \text{ pro } \gamma \neq \left| \frac{e_p}{f_p} \right|$$

$$T(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

$$S_C(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = \left| k_p \cdot \frac{e_p}{f_p} \right| \Rightarrow k_{p1,2} = \pm\gamma \cdot \left| \frac{f_p}{e_p} \right|, \quad k_{i1,2} \in \mathbb{R}$$

$$S_P(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

II. $C(s, \mathbf{k}) = C_{PID}(s, \mathbf{k})$

i. $W(s) = W_1(s)$

$$S(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 1 \Rightarrow \emptyset \text{ pro } \gamma \neq 1$$

$$T(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

$$S_C(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = \left| \frac{k_p \tau + k_d}{\tau} \right| \Rightarrow k_{p1} = \frac{\tau\gamma - k_d}{\tau}, \quad k_{i1} \in \mathbb{R}$$

$$\Rightarrow k_{p2} = -\frac{\tau\gamma + k_d}{\tau}, \quad k_{i2} \in \mathbb{R}$$

$$S_P(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

ii. $W(s) = W_2(s)$

$$S(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = \left| \frac{e_p}{f_p} \right| \Rightarrow \emptyset \text{ pro } \gamma \neq \left| \frac{e_p}{f_p} \right|$$

$$T(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

$$S_C(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = \left| \frac{(k_p \tau + k_d) e_p}{\tau f_p} \right| \Rightarrow k_{p1} = \frac{\tau f_p \gamma - k_d e_p}{\tau e_p}, \quad k_{i1} \in \mathbb{R}$$

$$\Rightarrow k_{p2} = -\frac{\tau f_p \gamma + k_d e_p}{\tau e_p}, \quad k_{i2} \in \mathbb{R}$$

$$S_P(s, \mathbf{k}): |H(j\infty, \mathbf{k})| = 0 \Rightarrow \emptyset \text{ pro } \gamma \neq 0$$

Z výsledků je zřejmé, že v případě specifikace omezujících podmínek na citlivostní funkci $S_C(s)$, bude množina křivek v parametrické rovině rozšířena o dvě další křivky.

Řešení rovnice 4.10 Platí, že $S_*(s)$ je libovolná funkce z (4.1)-(4.4).

I. $C(s, \mathbf{k}) = C_{PI}(s, \mathbf{k})$

i. $W(s) = W_1(s)$

$$S_*(s, \mathbf{k}): h_d(\mathbf{k}) = a_n \neq 0 \Rightarrow \text{stabilní}$$

ii. $W(s) = W_2(s)$

$$S_*(s, \mathbf{k}): h_d(\mathbf{k}) = a_n f_n \neq 0 \Rightarrow \text{stabilní}$$

II. $C(s, \mathbf{k}) = C_{PID}(s, \mathbf{k})$

i. $W(s) = W_1(s)$

$$S_*(s, \mathbf{k}): h_d(\mathbf{k}) = \tau a_n \neq 0 \Rightarrow \text{stabilní}$$

ii. $W(s) = W_2(s)$

$$S_*(s, \mathbf{k}): h_d(\mathbf{k}) = \tau a_n f_n \neq 0 \Rightarrow \text{stabilní}$$

Z předchozích výsledků je zřejmé, že pro uvedené případy tato rovnice nijak neovlivní výslednou hranici H_∞ regionu v parametrické rovině.

4.2.2 Kořeny kvartického polynomu

Kvartická rovnice představuje nejvyšší stupeň algebraické rovnice, kdy lze řešení nalézt ještě analytickou cestou. Pro kořeny rovnice (4.18), pokud $a \neq 0$, platí, že

$$\begin{aligned} x_{1,2} &= \frac{-b}{4a} - S \pm \frac{1}{2} \sqrt{D_{12}}, \\ x_{3,4} &= \frac{-b}{4a} + S \pm \frac{1}{2} \sqrt{D_{34}}, \end{aligned} \quad (4.23)$$

kde

$$\begin{aligned} p &= \frac{8ac - 3b^2}{8a^2}, \\ q &= \frac{b^3 - abc + 8a^2d}{8a^3}, \end{aligned} \quad (4.24)$$

$$S = \frac{1}{2} \sqrt{-\frac{2}{3}p + \frac{1}{3a} \left(Q + \frac{\Delta_0}{Q} \right)},$$

$$D_{12} = -4S^2 - 2p + \frac{q}{S}, \quad (4.25)$$

$$D_{34} = -4S^2 - 2p - \frac{q}{S}$$

a kde

$$\begin{aligned} Q &= \sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}, \\ \Delta_0 &= c^3 - 3bd + 12ae, \end{aligned} \quad (4.26)$$

$$\Delta_1 = 2c^3 - 9bcd + 27b^2e + 27ad^2 - 72ace.$$

Jak již bylo uvedeno dříve, pro výpočet křivek (4.16) je nutné nalézt reálné řešení rovnice (4.18). Na základě vztahů (4.25), diskriminantu

$$\Delta = \frac{1}{27}(4\Delta_0^3 - \Delta_1^2) \quad (4.27)$$

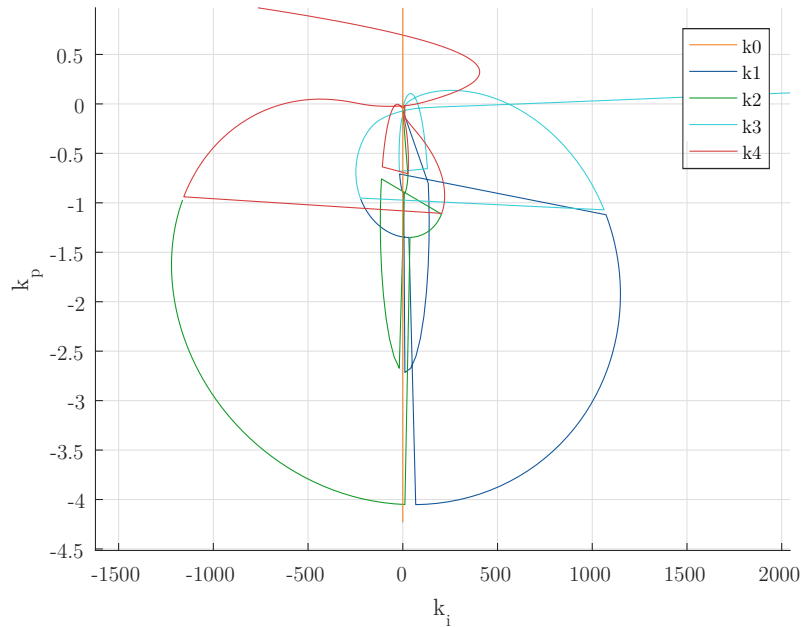
a rovnic

$$\begin{aligned} P &= 8ac - 3b^2, \\ R &= b^3 + 8da^2 - 4abc, \\ D &= 64a^3e - 16a^2c^2 + 16ab^2c - 16a^2bd - 3b^4, \end{aligned} \quad (4.28)$$

lze podle [15] určit podmínky, které toto řešení zaručují. Větvení těchto podmínek je znázorněno na diagramu, který je součástí přílohy C. Z tohoto diagramu také vyplývá, že počet reálných řešení pro pevnou frekvenci ω a tak i počet výsledných hraničních křivek získaných řešením soustavy (4.9) může být libovolný z množiny $\{0, 2, 4\}$.

4.2.3 Hranice H_∞ regionu

Pro získání skutečné hranice množiny \mathcal{K} musí být z množiny křivek \mathcal{B} (obr. 4.3), vybrány ty jejich části, resp. body parametrické křivky, které splňují podmínku (ii) návrhové úlohy a pro které platí, že $\|H(s, \mathbf{k})\|_\infty = \gamma$.



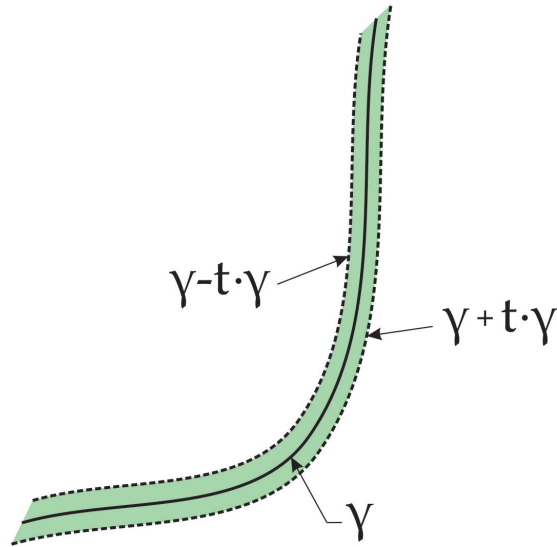
Obr. 4.3: Křivky množiny \mathcal{B}

Vzhledem k předpokladu racionální přenosové funkce $P(s)$ je stabilita uzavřené smyčky pro určitý bod parametrické křivky \mathbf{k} stanovena na základě počtu pólů přenosu uzavřené smyčky (4.2) v pravé otevřené polorovině komplexní roviny. Hodnota

∞ -normy (2.14) je pak pro zvolenou posloupnost frekvencí $\{\omega_i\}_{i=0}^N$ určena jako $\max_{\omega_i} |H(j\omega_i)|$. Z důvodu možné numerické chyby je stanoven toleranční koeficient t , načež test ∞ -normy bodů hraniční křivky přechází do tvaru

$$(\gamma - \gamma \cdot t) \leq \max_{\omega_i} |H(j\omega_i)| \leq (\gamma + \gamma \cdot t), \quad (4.29)$$

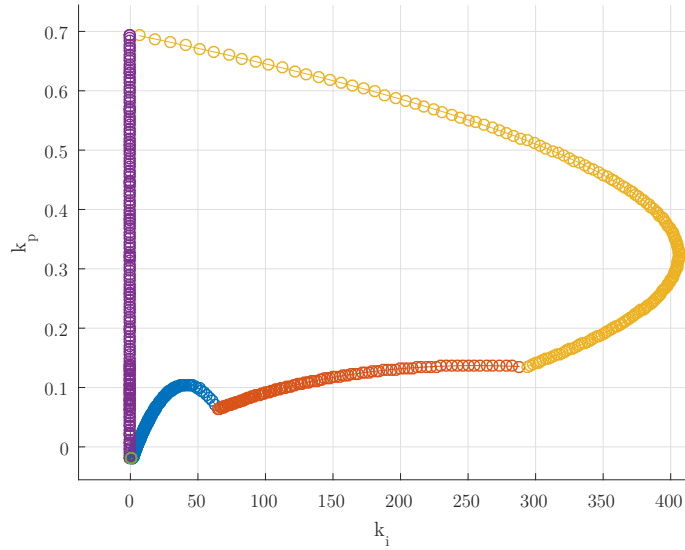
čímž vznikne okolo hranice toleranční pásmo. Změnou hodnoty tolerančního koeficientu pak lze v určitých situacích docílit větší, či naopak menší přesnosti hraniční oblasti. Tato vlastnost je využívána při sjednocování hraničních křivek a umožňuje volbu „výpočetního módu“ (5.2.6).



Obr. 4.4: Toleranční pásmo

Pokud se analýzou získaných úseků křivek množiny \mathcal{B}^* dojde k závěru, že tvoří pouze část hranice regionu, který je tedy de facto neomezený, je množina \mathcal{B}^* rozšířena o pomocnou pseudo hraniční křivku. Pro body na této křivce platí podmínka $\|H(s, \mathbf{k})\|_\infty < \gamma$. Výsledkem selekce je množina křivek \mathcal{B}^* , které představují skutečnou (pseudo) hranici množiny regulátorů \mathcal{K} .

Otázkou ale zůstává, jak získané křivky sjednotit, resp. seřadit, aby za každé situace vytvořily ucelenou hranici regionu nebo regionů v parametrické rovině. Pro většinu případů by bylo možné křivky jednoduše „poskládat“ podle principu nejbližšího souseda, tedy porovnáním vzdáleností mezi koncovým bodem jedné křivky a počátečními/koncovými body zbylých křivek. Jsou ale případy, kdy by takovýmto postupem došlo ke špatnému seřazení a tím i k chybnému vykreslení H_∞ regionu. Z tohoto důvodu byl vyvinut následující algoritmus, který je sice založen na podmínce minimální vzdálenosti, ale obsahuje i řadu kontrolních a testovacích prvků. Vývojový diagram základní struktury tohoto algoritmu je součástí přílohy D. Stručně ho pak lze popsat následujícím způsobem.



Obr. 4.5: Křivky množiny \mathcal{B}^*

CCsort algoritmus: Uvažujme „prázdnou“ hranici regionu a množinu hraničních křivek \mathcal{B}^* . V prvním kroku je vybrána jedna z křivek množiny \mathcal{B}^* a přidělena k hranici regionu. Následně jsou z množiny \mathcal{B}^* postupně hledáni takoví „sousedé“, kteří splňují podmínku minimální vzdálenosti počátečního/koncového bodu. Najde-li se křivka, která tuto vlastnost splňuje, je mezi koncovým bodem hranice regionu a počátečním/koncovým bodem vybrané křivky vytvořeno spojení, které následně prochází několika cyklickými testy⁴, během kterých se vyhodnotí, zda se má křivka přidat k hranici regionu, případně jakým způsobem má dojít k jejich propojení⁵. Pokud je nejvhodnější křivka z množiny \mathcal{B}^* ta, která byla k hranici regionu přidána jako první, je region uzavřen, a pokud jsou v množině \mathcal{B}^* ještě nějaké křivky, které stále nejsou součástí hranice regionu, pokračuje algoritmus dále postupem výše popsáním. Nejsou-li v množině \mathcal{B}^* žádné další křivky pro kompletaci, je algoritmus ukončen. Vytvoření ucelené oblasti je velice důležitou částí celého návrhového procesu a hraje klíčovou roli při hledání průniku návrhových omezení.

4.2.4 Průnik návrhových omezení typu H_∞

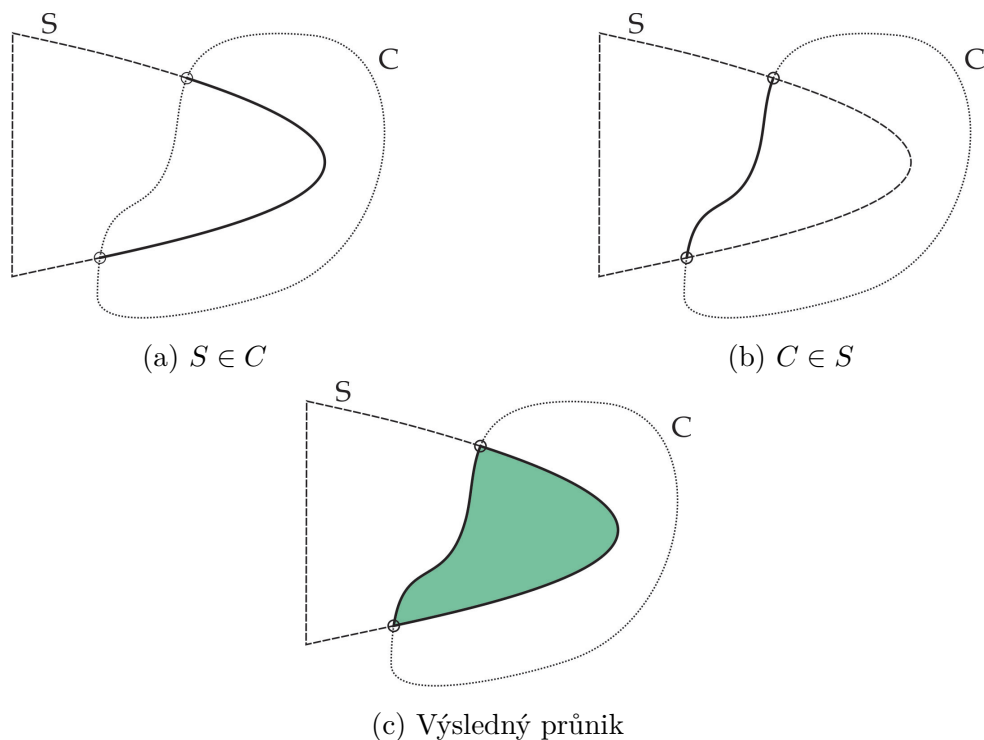
Uvažujme situaci, kdy je zadáno hned několik návrhových omezení typu H_∞ . V tomto případě je pak pro každou z podmínek nalezen příslušný H_∞ region či informace o neexistenci řešení pro danou podmínku. Výsledným řešením celého procesu je pak průnik všech získaných regionů. Pokud pro některou z podmínek neexistuje řešení,

⁴V této části hraje důležitou roli zvolená hodnota tolerančního koeficientu.

⁵Interpolace přímkou, spline polynomem nebo v posledním možném případě automatickým dopočítáním hranice.

neexistuje řešení ani pro celou návrhovou úlohu. Pro nalezení průniku jednotlivých oblastí byl vytvořen algoritmus, který je inspirován geometrickými algoritmy popsány v [8]. V rámci obecného řešení je předpokládána existence dvou obecně nekonvexních mnohoúhelníků. Cílem je pak určit průnik těchto mnohoúhelníků.

V první fázi je potřeba nalézt možné body průniku mezi jednotlivými stranami polygonů. S využitím základních vztahů analytické geometrie a maticových výpočtů v prostředí MATLAB lze relativně rychle nalézt množinu všech průsečíků. Pokud množina není prázdná lze následně přistoupit k další fázi. S využitím „even-odd“ pravidla [5] je určeno, které části jednotlivých polygonů, resp. jejich body, leží uvnitř druhého polygonu (obr. 4.6). Toto pravidlo dokáže rozhodnout o příslušnosti nakreslením úsečky ze zkoumaného bodu do nekonečna v libovolném směru a vyčíslením počtu segmentů, se kterými se křížila. Pokud je tento počet lichý, je bod umístěn uvnitř zkoumané oblasti. Pokud je naopak sudý, je bod mimo tuto oblast. Výstupem je pak množina křivek, které tvoří ohraničují výslednou oblast. Sjednocením křivek s body průniku je pak získán výsledný region.



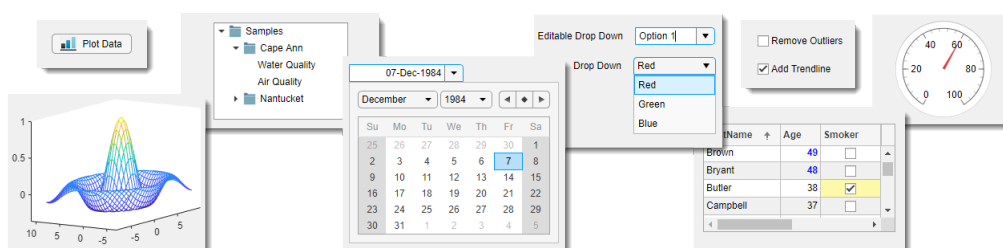
Obr. 4.6: Hledání průniku dvou polygonů

5 Uživatelské rozhraní

Před samotným popisem jednotlivých částí aplikace je podstatné poznamenat, že GUI bylo sestaveno ve vývojovém prostředí App Designer programu MATLAB.

5.1 MATLAB App Designer

Toto prostředí umožňuje vytvořit samostatné „Matlabovské“ programy, které mohou obsahovat mnoho interaktivních ovládacích prvků a lze je snadno propojit s vlastním kódem a funkcemi.



Obr. 5.1: App Designer

Zdroj: <https://www.mathworks.com/help/matlab/gui-development.html>

I přesto, že je tato služba relativně nová¹, nabízí oproti svému předchůdci² řadu výhod. Kromě propracovanější grafiky s modernějším vzhledem jsou aplikace vytvářeny již plně objektově orientovaným programováním a jsou rozšířeny i o nové možnosti událostně řízeného programování. Dále nabízí i řadu pomůcek, které začínající programátoři a tvůrci jistě rádi ocení. Sdílení vytvořené aplikace, ať už v rámci vlastního okolí či širší veřejnosti, je možné uskutečnit hned několika způsoby.

- Zabalení aplikace do tzv. `mlappinstall` balíčku, který umožňuje její budoucí implementaci do MATLAB Apps nabídky v horní liště programu. Stejně dobře lze tento balíček sdílet v rámci MATLAB File Exchange.
- Novinkou je možnost vytvořit z programu webovou aplikaci. Webové aplikace mají jistě řadu výhod a s postupem času se stávají stále populárnějšími. Vzniklou webovou aplikaci lze spouštět v běžném internetovém prohlížeči bez potřeby běhového prostředí MATLAB Runtime na koncovém počítači.

¹Uživatelská rozhraní lze v tomto prostředí vytvářet od verze R2016a.

² MATLAB GUIDE

- Poslední možnost je pak vytvoření samostatné desktopové aplikace, kterou může mít uživatel trvale uloženou na svém zařízení. I v tomto případě není vyžadována přítomnost programu MATLAB na cílovém zařízení. Běžové prostředí, nutné pro fungování aplikace, je možné vložit přímo do jejího instalačního souboru.

V rámci této práce byla zvolena první z možností, díky které jsou všechny funkce a kódování volně dostupné k nahlédnutí. Detailní návod na zprovoznění aplikace je pak umístěn v příloze **B** této práce. Pro správné fungování je důležité ještě poznamenat, že s každou novou verzí vývojového prostředí MATLAB se rozšiřují i možnosti App Designeru. Jsou přidávány nové a nové komponenty, opravují se chyby a zlepšuje se výkonnost. Pro maximální možné využití těchto výhod je aplikace vytvořena ve verzi R2019a. Vyplývá z toho tedy, že jsou použity komponenty, které v předchozích verzích nejsou dostupné, a z tohoto důvodu je aplikace provozuschopná pouze ve verzi R2019a a vyšší.

5.2 Prostředí oken

Po spuštění aplikace se jako první zobrazí uvítací okno s logem programu (obr. 5.2). Toto okno setrvává v popředí, dokud nebudou zcela načteny všechny grafické komponenty pro okno hlavní aplikace. Pokud by bylo v průběhu načítání uvítací okno zavřeno³, zastaví se tím i načítání celé aplikace. Při prvním spuštění po startu programu MATLAB může zobrazení okna určitou chvíli trvat. Tato prodleva je způsobena Just In Time kompilací kódu pro vytvoření interních datových struktur. Dále je nutné poznamenat, že lze spustit jen jednu instanci celé aplikace.

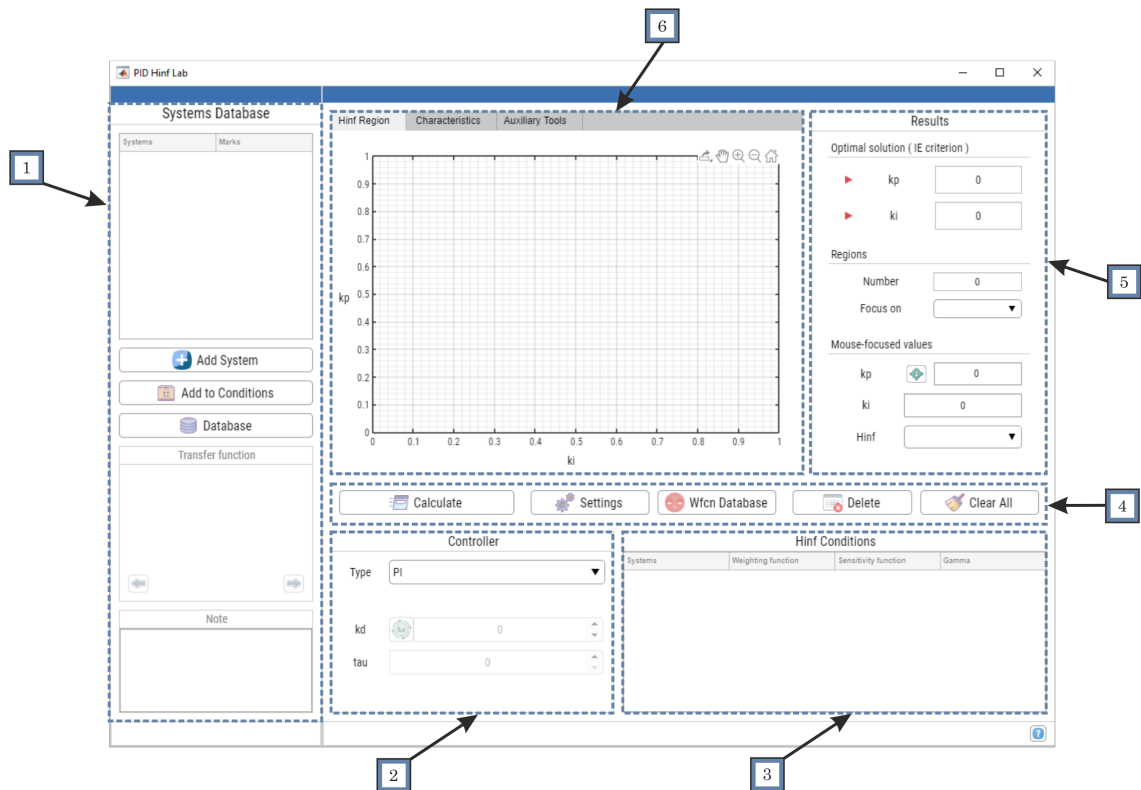


Obr. 5.2: Uvítací okno

³Přes symbol křížku v pravém horním rohu.

5.2.1 Hlavní okno

Po načtení všech komponent se tedy zavře uvítací okno (obr. 5.2) a zobrazí se hlavní uživatelské rozhraní celé aplikace (obr. 5.3). Struktura tohoto rozhraní je rozložena do několika základních bloků, které slouží ke správě systémů (obr. 5.3-1), volbě regulátoru (obr. 5.3-2), specifikaci H_∞ podmínek (obr. 5.3-3) a analýze výsledků (obr. 5.3-4, obr. 5.3-5, obr. 5.3-6). Zbylé místo zaujímají tlačítka pro ovládání aplikace (obr. 5.3-4).



Obr. 5.3: Hlavní rozhraní aplikace

Nejdůležitějším prvkem systémové části aplikace (obr. 5.3-1) je tabulka systémů o dvou sloupcích. V prvním jsou vypsány názvy všech zadaných systémů, v druhém jsou pak zaškrťovací políčka pro výběr jednotlivých systémů pro další zpracování. Systém lze do aplikace zadat dvěma způsoby:

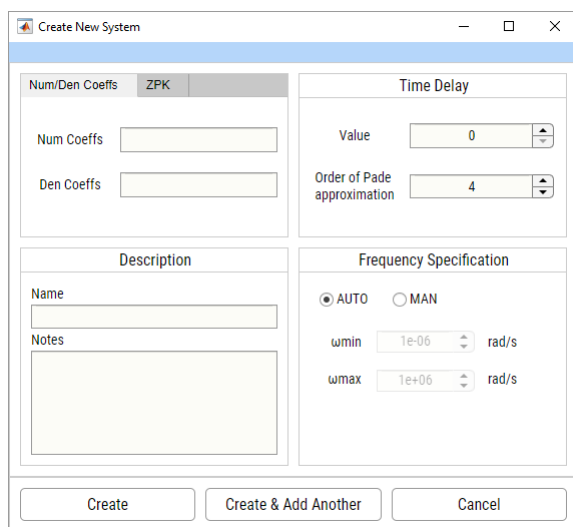
1. Tlačítkem „Add System“, které zobrazí okno pro zadání systému.
2. Tlačítkem „Database“, které zobrazí okno systémové databáze. V tomto okně lze systémy nejen vytvářet, ale i upravovat, mazat, importovat a exportovat.

Pro rychlé vytvoření systému lze využít prvního způsobu, kdy se po stisknutí tlačítka vytvoří okno (obr. 5.4) s poli a prvky pro zadávání parametrů systému.

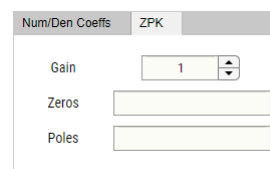
5.2.2 Nový systém

Rozhraní pro zadání nového systému je rozděleno do čtyř základních panelů. Parametry přenosové funkce se specifikují v horních dvou panelech. Nový systém lze zadat ve dvou běžně používaných formách:

- Koeficienty polynomů v čitateli a jmenovateli – Num/Den Coeffs (obr. 5.4a)
- Nuly, póly a zesílení – ZPK (obr. 5.4b)



(a) Num/Den Coeffs



(b) ZPK

Obr. 5.4: Vytvoření nového systému

Je-li uvažován systém s dopravním zpožděním na vstupu ve tvaru (4.13), je do pole „Value“ vložena příslušná hodnota zpoždění D . Jak již bylo dříve uvedeno, v rámci této práce je exponenciální funkce reprezentující dopravní zpoždění nahrazena Padého aproximací. Výchozí hodnota řádu aproximace je stanovena na 4. Tuto hodnotu lze podle uvážení v příslušném poli libovolně měnit.

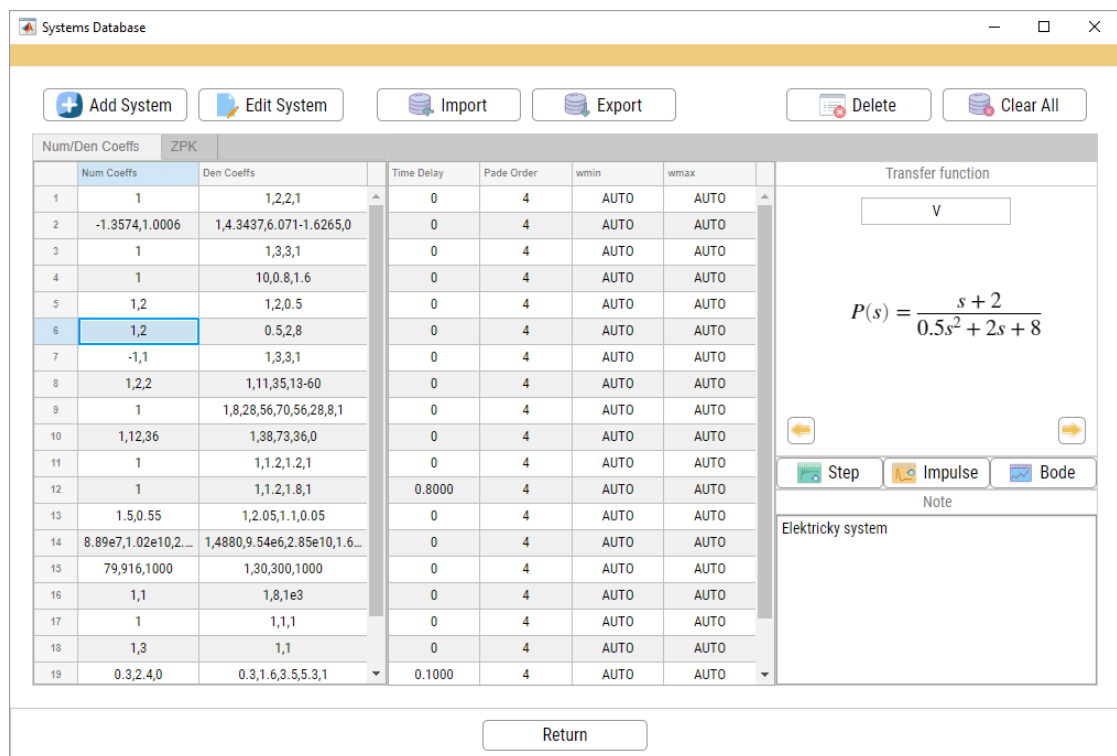
Třetí část okna je určena pro slovní popis zadávaného systému. V tomto místě lze specifikovat název systému a případně vložit obsáhlejší poznámky. Pokud není název systému zadán, jsou systémy automaticky označovány jako Sys#. Poslední část okna je určena pro možnou konkretizaci frekvenčního rozsahu, na kterém má být zadávaný systém analyzován.

V dolní liště okna jsou umístěna tři tlačítka. Tlačítkem „Create“ se zavře aktuální okno, zadaný systém se uloží do paměti a jeho název se přidá do tabulky v hlavním okně aplikace. Tlačítkem „Create & Add Another“ se vytvoří nový systém, přidá se do tabulky, ale okno (obr. 5.4) se nezavře. Zůstane otevřené a zůstanou v něm zadané

i specifikace jednotlivých parametrů, což umožní rychlejší zadávání většího počtu nových systémů. Tlačítkem „Cancel“ lze pak kdykoliv zavřít aktuální okno a ukončit tak zadávání. Stejného cíle lze docílit i přes tlačítko se symbolem křížku v pravém horním rohu okna. Pokud by v průběhu zadávání došlo k volbě hodnoty parametru, která odporuje základním podmínkám pro přenos systému, dojde k zobrazení dialogového okna s příslušnou varovnou nebo chybovou hláškou.

5.2.3 Databáze systémů

Po stisknutí tlačítka „Database“ v hlavním okně aplikace se zobrazí okno systémové databáze, ve kterém lze systémy vytvářet, upravovat, importovat, exportovat i mazat. Do hlavního okna aplikace se lze vrátit přes tlačítko „Return“ nebo opět přes symbol křížku v pravém horním rohu databázového okna.



Obr. 5.5: Databáze systémů

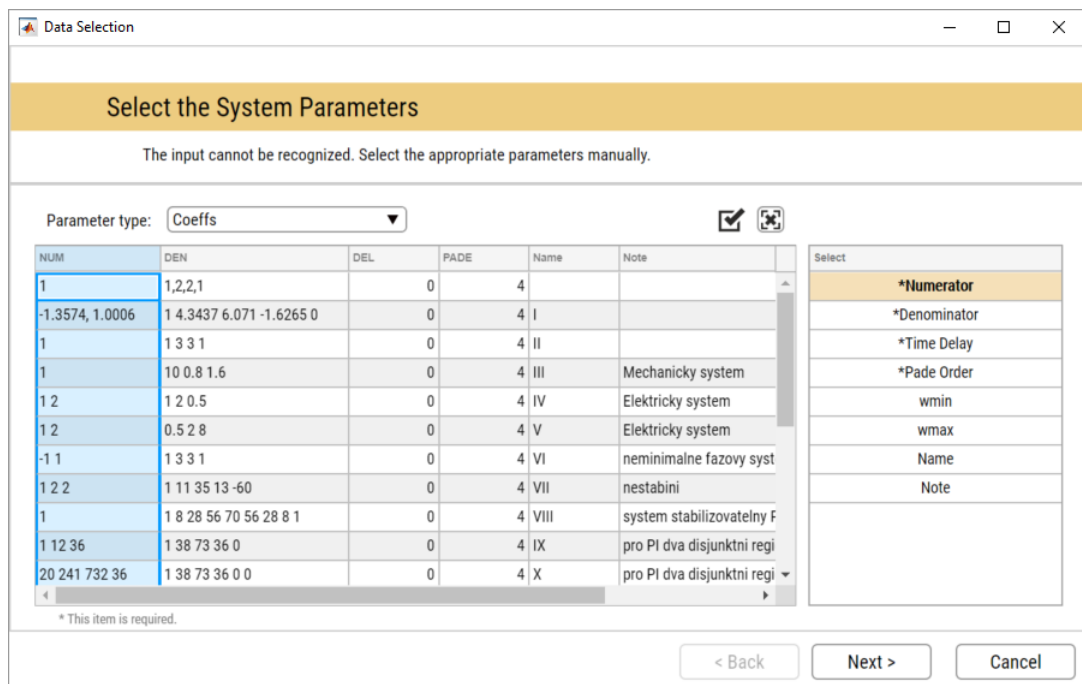
Hlavní část okna je vyplněna třemi tabulkami. V první, resp. druhé tabulce jsou vypisovány koeficienty polynomů čitatele a jmenovatele (Num/Den Coeffs), resp. hodnoty nul, pólů a zesílení (ZPK) přenosových funkcí jednotlivých systémů. Ve třetí tabulce jsou pak hodnoty dopravního zpoždění, řádu Padého aproximace a frekvenčního rozsahu. Každý řádek jednotlivých tabulek představuje jeden zadaný systém. Pokud uživatel na některý z těchto řádků nebo na k němu příslušnou buňku klikne myší, zobrazí se v pravé části okna přenosová funkce daného systému, název systému a pokud

existují, tak i zadané poznámky. Pokud je přenosová funkce systému příliš široká, lze přes symboly šipek funkci posunout doprava nebo doleva. Pro vybrané systémy lze přes příslušná tlačítka vykreslit přechodovou, impulsní a bodeho charakteristiku.

Add System Pokud byly v hlavním okně již nějaké systémy vytvořeny, jsou automaticky zobrazeny i v tomto okně. Nový systém se i zde zadává přes tlačítko „Add System“, které otevře již výše popsané okno (obr. 5.4) pro zadávání parametrů. Vše je pak zcela totožné jako v předchozím případě.

Edit System Pro úpravu systému stačí kliknout na jeho příslušný řádek či buňku. Tím se vybraný systém označí, načež ho pak lze upravit přes tlačítko „Edit System“, kdy se zobrazí již dobře známé okno (obr. 5.4) kde jsou jednotlivé pole vyplněna hodnotami parametrů zvoleného systému. Po skončení úprav lze tlačítkem „Modife“ modifikované hodnoty uložit zpět do databáze. Tlačítkem „Cancel“ či křížkem v pravém horním rohu lze proces kdykoli ukončit.

Import Pokud má uživatel v nějakém externím souboru již hodnoty potřebných systémů uloženy, lze je do databáze načíst prostřednictvím tlačítka „Import“. Importovat lze pouze data ve formátu xlsx, xls a csv. Pokud systém při nahrávání rozpozná známou strukturu tabulky, jsou data do databáze nahrána okamžitě. Jestliže ale strukturu nerozpozná, otevře se okno (obr. 5.6) s tabulkovým průvodcem pro výběr buněk nebo sloupců, které obsahují hodnoty jednotlivých parametrů.



Obr. 5.6: Výběr dat z tabulky

V tomto okně se v prvním kroku definuje, k jaké formě budou patřit vybrané hodnoty parametrů: „Coeffs“ nebo „ZPK“. Následně se výběrem myši zvolí sloupec, nebo část sloupce s příslušnými parametry. Jaké parametry se aktuálně mají vybrat, lze poznat z tabulky v pravé části okna podle zvýrazněného řádku. Označení vybraných buněk, resp. sloupců, je potvrzeno zobrazením „fajfky“ nad pravým horním rohem tabulky hodnot. Označení lze zrušit přes tlačítko se symbolem křížku, které je na téže místě. Tlačítkem „Next“ lze pak přejít na volbu dalšího parametru. Podobně se lze tlačítkem „Back“ vrátit k již zvoleným parametrům. Stejně jako v předchozích oknech lze i zde tlačítkem „Cancel“ ukončit celý proces nahrávání.

Pro nahrání jednotlivých systémů do databáze je nutné provést označení dat pro parametry, které mají v tabulce vpravo před svým názvem hvězdičku. Ostatní parametry nejsou pro vytvoření daných systémů vyžadovány, a pokud nebudou příslušné veličiny označeny, automaticky bude aplikace pracovat s jejich výchozími hodnotami. Celý proces se ukončí v posledním kroku po stisknutí tlačítka „Finish“. Po ukončení tabulkového průvodce se začnou nahrávat jednotlivé systémy do systémové databáze⁴, kdy na se na závěr zobrazí dialogové okno s informací o počtu nahraných systémů a o počtu systémů, které do databáze nemohly být nahrány z důvodu nesplnění systémových požadavků.

Export Pokud má uživatel již vytvořenou nějakou sadu systémů a chtěl by je využít i při další práci s aplikací, lze pomocí tlačítka „Export“ uložit všechny zadané systémy do souboru ve formátu xlsx, xls nebo csv. Vzniklý soubor lze pak při dalším využívání aplikace nahrát zpět do databáze.

Delete/Clear All Pro odstranění jednoho nebo několik specifických systémů stačí dané systémy označit myší a tlačítkem „Delete“ je smazat. Tlačítkem „Clear All“ lze pak smazat všechny systémy v databázi.

Předpokládejme nyní, že je aplikace ve stavu, kdy jsou potřebné systémy zadány a jejich názvy lze vidět v tabulce systémové části rozhraní hlavního okna (obr. 5.3-1). Kliknutím na jednotlivé řádky této tabulky se v systémové části rozhraní zobrazí přenosové funkce daných systémů a pokud existují, tak i zadané poznámky. Stejně jako v okně systémové databáze i zde platí, že pokud je vykreslená přenosová funkce příliš dlouhá, lze ji přes tlačítka se symboly šípek posunout doprava nebo doleva a přečíst tak celou funkci.

Jak je vysvětleno v (4.1), je pro výpočet H_∞ regionu potřeba specifikovat návr-

⁴Do tabulky se budou postupně přidávat jednotlivé systémy.

nový požadavek či požadavky ve tvaru

$$\|H(s, \mathbf{k})\|_{\infty} = \|W(s)S_*(s, \mathbf{k})\|_{\infty} \leq \gamma.$$

Pro získání těchto podmínek je potřeba specifikovat analyzované systémy. Z tohoto důvodu jsou v tabulce systémů zaškrťovací políčka, kterými lze jednotlivé systémy označit a pak přes tlačítko „Add to Conditions“ vložit do tabulky H_{∞} podmínek, neboli „Hinf Conditions“ (obr. 5.3-3). V této tabulce pak každý řádek představuje jednu podmínku ve zmíněném tvaru.

Kromě sloupce s názvem vybraných systémů jsou zde ještě další tři, kterými lze volit váhovou funkci $W(s)$ ⁵, jednu ze čtyř citlivostních funkcí (4.1, 4.4) přidružených k zpětnovazební smyčce z obr. 4.1 a návrhový parametr γ . Váhové a citlivostní funkce lze zadat přes dvojité kliknutí na příslušné buňky, čímž se rozbalí nabídka funkcí. Jednu nebo více podmínek lze přes tlačítko „Delete“ vymazat. Celý seznam podmínek se pak vymaže pomocí tlačítka „Clear All“.

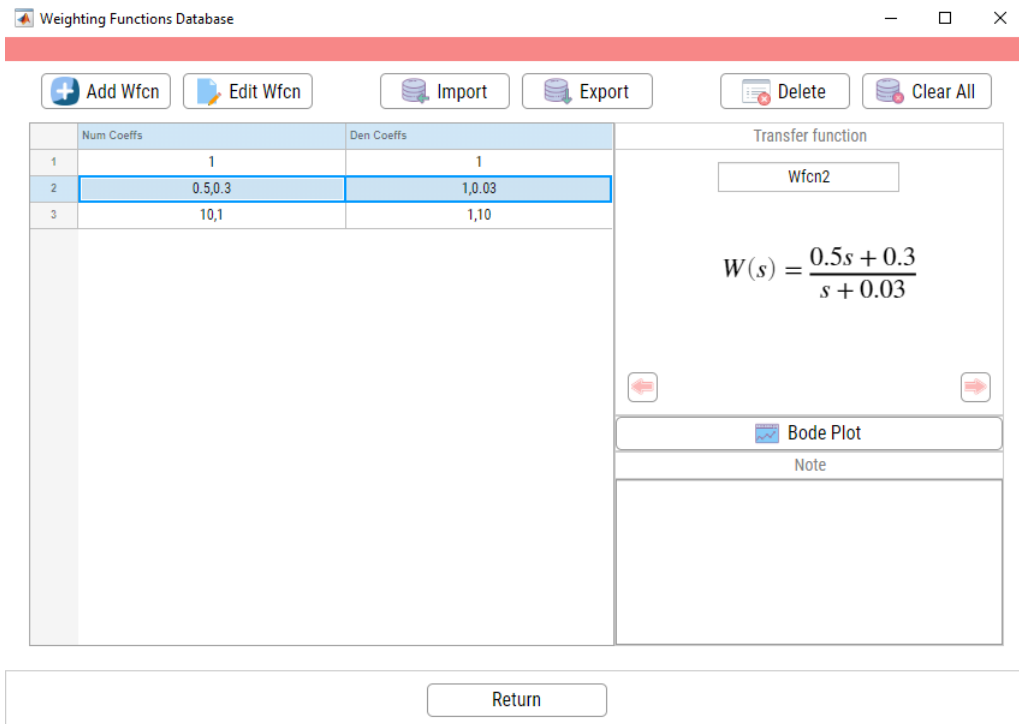
5.2.4 Databáze váhových funkcí

Pokud chce uživatel k jednotlivým podmínkám přidat váhovou funkci, dostane se přes tlačítko „Wfcn Database“ do okna databáze váhových funkcí. V rámci této aplikace lze váhové funkce zadávat pouze v jedné formě, a to jako koeficienty polynomů v čitateli a jmenovateli. Okno této databáze je v podstatě identické s oknem databáze systémů. S tím rozdílem, že je zde pouze jedna tabulka, která zobrazuje koeficienty čitatele a jmenovatele příslušné váhové funkce. Oproti systémové databázi je zde tlačítko „Bode Plot“, které slouží pro vykreslení Bodeho charakteristiky jedné nebo více zvolených váhových funkcí.

Add Wfcn Nová váhová funkce se do databáze vloží přes tlačítko „Add Wfcn“, které zobrazí nové okno se zadávacími poli. Stejně jako u systémů lze i zde zvolit název váhové funkce, případně přidat nějakou poznámku. Tlačítka „Create“ a „Create & Add Another“ lze opět zadanou funkci vložit do databáze a případně pokračovat v zadávání. Tlačítkem „Cancel“ či křížkem v pravém horním rohu lze proces zadávání ukončit.

Jednotlivé váhové funkce lze přes tlačítko „Edit Wfcn“ modifikovat. Obdobně je to i s tlačítky „Import“ a „Export“, kterými lze váhové funkce buďto nahrát do databáze nebo naopak vyexportovat do externího souboru. Z databázového okna lze odejít přes tlačítko „Return“ či opět přes křížek v pravém horním rohu. Zadané váhové funkce lze následně nalézt při rozbalení nabídky v tabulce H_{∞} podmínek.

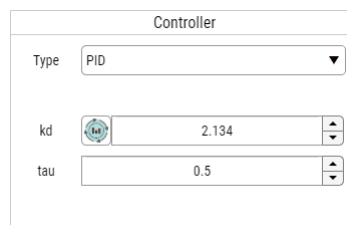
⁵Ve výchozím stavu je podmínka uvažována bez váhové funkce – „w/o“.



Obr. 5.7: Databáze váhových funkcí

5.2.5 Volba typu regulátoru

Před provedením výpočtu je třeba ještě zvolit typ regulátoru (obr. 5.3-2). V aplikaci lze volit mezi regulátorem typu PI nebo PID. Při využití PI regulátoru nemusí uživatel zadávat žádné další informace a může rovnou spustit výpočet H_∞ regionu. Pokud ale zvolí PID regulátor, musí navíc ručně zadat hodnoty derivačního zesílení k_d a časovou konstantu filtru τ . Protože volba hodnoty zesílení, která by vedla na neprázdný region, není zcela triviální, je v tomto případě aplikace rozšířena o možnost odhadu suboptimální hodnoty parametru k_d , což lze uskutečnit přes tlačítko vedle zadávacího pole tohoto parametru.



Obr. 5.8: Volba PID regulátoru

Odhad je proveden přes výpočet H_∞ regionu pro PI regulátor. Hodnota parametru je pak stanovena ze známé závislosti mezi časovými konstantami derivační a

integrační části PID regulátoru, kdy platí, že

$$\frac{T_d}{T_i} \approx 0.25, \quad (5.1)$$

kde $T_i = \frac{k_p}{k_i}$ a $T_d = \frac{k_d}{k_p}$. Suboptimální hodnota derivačního zesílení k_d je pak určena vztahem

$$k_d = 0.25 \cdot T_i k_p. \quad (5.2)$$

Regulátor je tedy již také vybrán a před samotným výpočtem lze ještě změnit některé ze základních nastavení pro výpočet regionu.

5.2.6 Nastavení

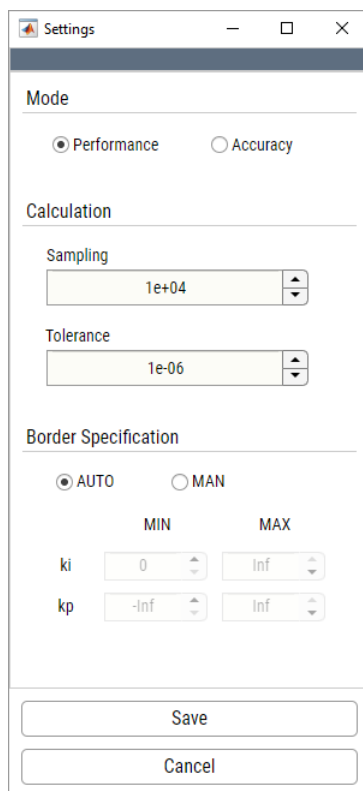
Tlačítkem „Settings“ v hlavní části aplikace lze zobrazit okno se zadávacími poli pro parametry, které značně ovlivňují konečný tvar regionu. První možností, čím lze v tomto okně začít, je volba výpočetního módu pro sjednocení hraničních křivek H_∞ regionu. Při volbě „Performance“ módu je upřednostňována rychlost výpočtu nad případnou menší nepřesností výsledného regionu. Propojení křivek z množiny \mathcal{B}^* pak nemusí být přesnou hranicí, může se jednat pouze o pseudohranici⁶, neboť je při jejím výpočtu uvažováno větší pásmo tolerance. Volbou „accuracy“ módu je naopak požadováno bezvýhradné splnění podmínky rovnosti ∞ -normy návrhovému parametru γ pro všechny body hranice regionu a výpočet se tak může značně prodloužit.

Dalšími parametry pro nastavení je počet vzorků N posloupnosti frekvencí $\{\omega_i\}_{i=0}^N$, nebo-li „Sampling“, kterým se určuje i navzorkování křivek z množiny \mathcal{B} a hodnota tolerančního koeficientu t - „Tolerance“ (4.29). Jako poslední lze ohraničit oblasti hledaného řešení. Např. omezit proporcionální zesílení na kladná čísla. Uložení nového nastavení lze provést tlačítkem „Save“. Tlačítkem „Cancel“ či křížkem v pravém horním rohu lze proces opět kdykoli ukončit.

Předpokládejme tedy, že je aplikace ve stavu, kdy jsou všechny potřebné informace zadány nebo nastaveny a lze spustit výpočet H_∞ regionu tlačítkem „Calculate“. Pro zadanou podmínku jsou pak hledány hranice množiny \mathcal{B}^* , které splňují všechny zadané podmínky. Pokud je nalezena, zobrazí se v grafu v hlavním okně (obr. 5.3-6). Jestliže je část hranice regionu označena plnou čarou jedná se o skutečnou hranici, pro jejíž body \mathbf{k}_s platí, že

$$\|H(s, \mathbf{k}_s)\|_\infty = \gamma.$$

⁶Tento termín byl již v této práci použit a popisuje křivku, jejíž body sice nesplňují podmínku rovnosti ∞ -normy návrhovému parametru γ , ale leží uvnitř skutečného regionu.



Obr. 5.9: Nastavení

Část hranice, která je označena přerušovanou čarou, pak představuje pseudohranici, jejíž body \mathbf{k}_{ps} splňují podmínku ve tvaru

$$\|H(s, \mathbf{k}_{ps})\|_{\infty} < \gamma.$$

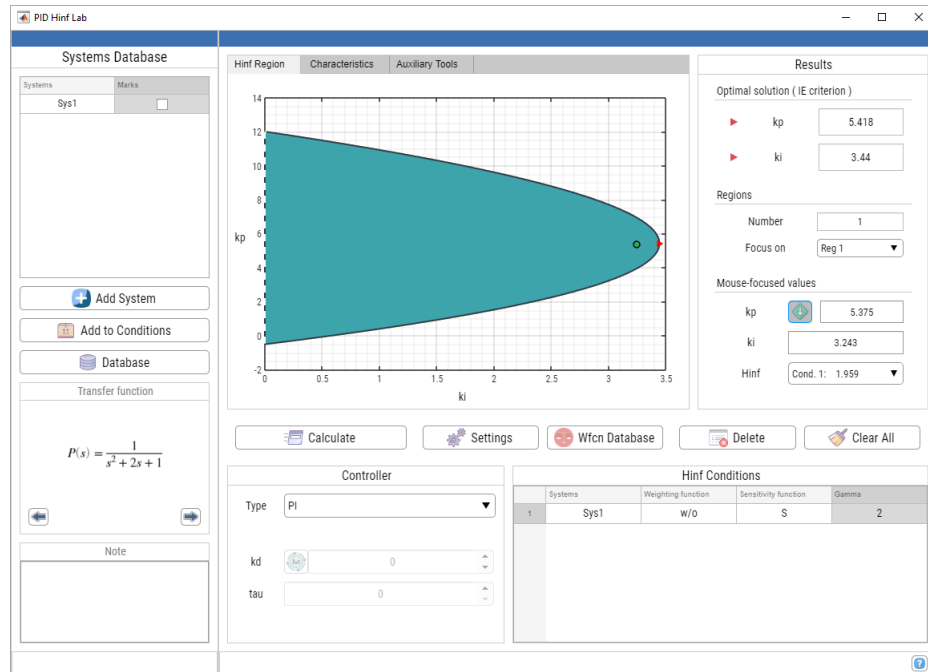
Takovéto označení je důležité pro správnou interpretaci neohrazených oblastí, kde hranice utíkají do nekonečna, nebo pro ne zcela přesnou hraniční křivku.

Kromě nalezeného regionu se v grafu zobrazí i bod, který představuje optimální hodnoty parametrů k_p a k_i pro IE kritérium. Tyto optimální hodnoty jsou zároveň vypsány do polí napravo od grafu. Kritérium IE je v této práci použito, protože představuje nejjednodušeji aplikovatelné integrální kritérium kvality. V [2] bylo ukázáno, že hodnota kritéria je nepřímo úměrná integračnímu zesílení a platí tedy, že

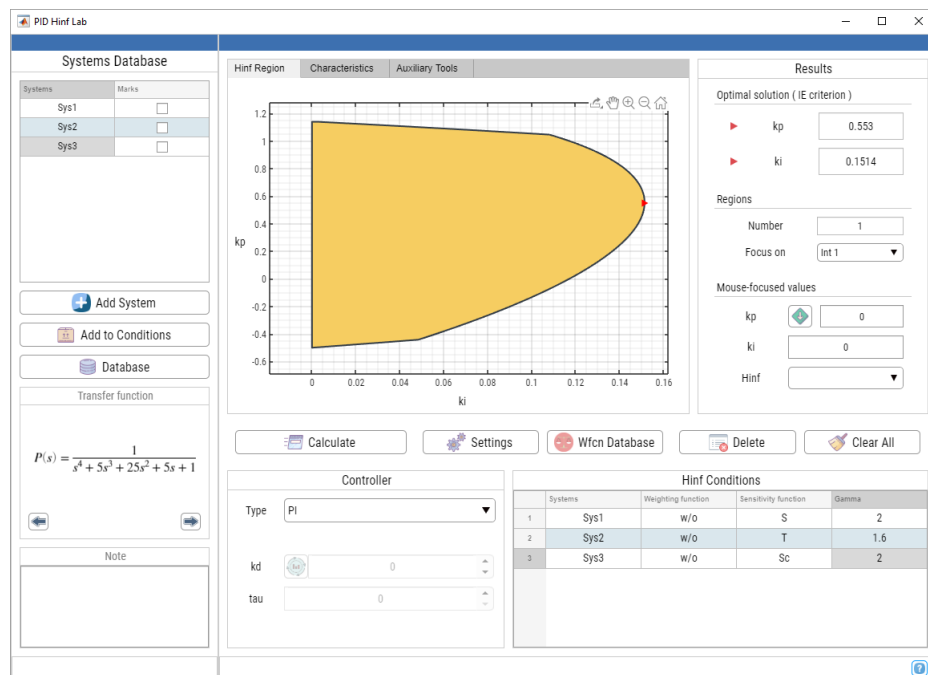
$$IE = \frac{1}{k_i}. \quad (5.3)$$

V některých situacích se může stát, že výsledný region není jediný a řešením je hned několik samostatných regionů. Pro přehlednější práci se získanými výsledky je v panelu výsledků vypsána informace o počtu nalezených regionů. Pokud jich je více, lze volbou příslušný region přiblížit. Kromě optimálního řešení podle IE kritéria lze po povolení přes příslušné tlačítko získat i další hodnoty parametrů regulátoru kliknutím do grafu, čímž se na příslušném místě vytvoří informační bod a v příslušných polích

panelu výsledků se vypíše informace o hodnotách parametru a hodnotě H_∞ normy. Pokud algoritmus žádnou oblast nenažde, objeví se dialogové okno s informací o prázdné množině řešení. V případě, že je zadáno více H_∞ podmínek, je hledán průnik všech nalezených regionů (obr. 5.11).



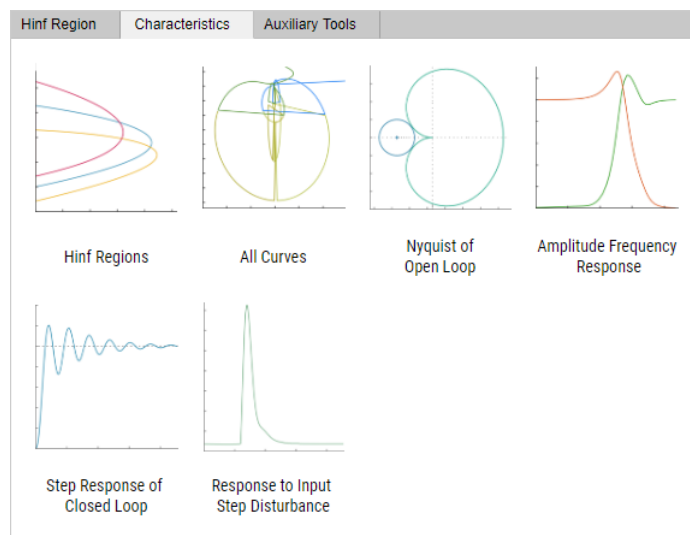
Obr. 5.10: Stav aplikace po nalezení řešení



Obr. 5.11: Výsledek pro několik návrhových omezení

5.3 Charakteristiky

Kromě H_∞ regionu lze vykreslovat i další charakteristiky (obr. 5.12), kterými lze posoudit kvalitu zvoleného řízení.



Obr. 5.12: Vykreslení charakteristik

Těmito charakteristikami jsou

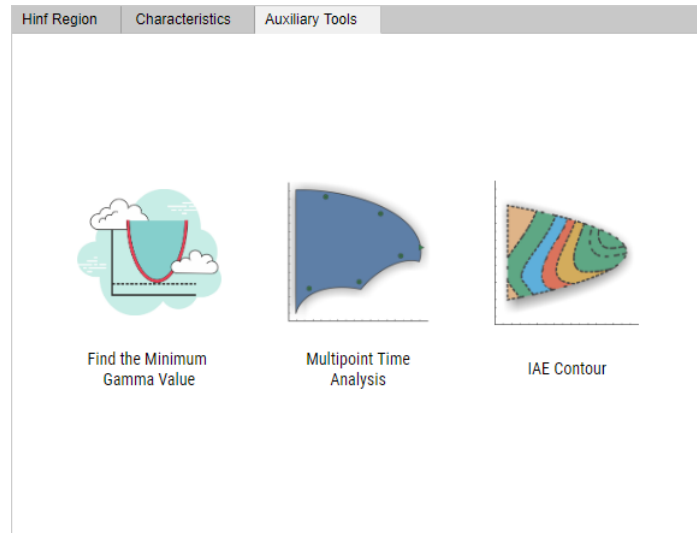
- H_∞ regiony⁷ a oblasti stability
- Všechny křivky množiny \mathcal{B}
- Nyquistova křivka otevřené smyčky
- Amplitudová frekvenční charakteristika
- Přejímová charakteristika uzavřené smyčky
- Odezva na skokovou poruchu na vstupu systému

Pro vykreslení jednotlivých grafů stačí kliknout na obrázek, který danou charakteristiku reprezentuje. Jednotlivé charakteristiky jsou pak vykresleny v samostatných „figure“ oknech.

⁷Pro jednu H_∞ podmínku je vykreslený region totožný s grafem v hlavním okně. Pokud je ale zadáno více podmínek, vykreslí tato funkce všechny dílčí regiony bez vzájemného průniku.

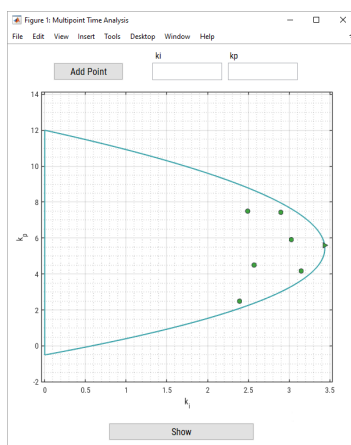
5.4 Pomocné nástroje

Pro lepší využitelnost a uživatelskou přívětivost je aplikace obohacena o pomocné nástroje (obr. 5.13), které by uživateli mohly pomoci při hledání optimálního nastavení regulátoru pro danou úlohu.

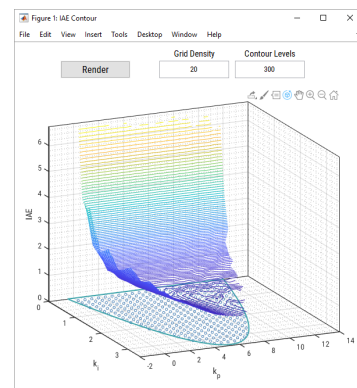


Obr. 5.13: Pomocné nástroje

Prvním z těchto nástrojů je funkce, která nalezne minimální hodnotu návrhového parametru γ pro kterou, vzhledem k tolerančnímu pásmu, ještě existuje řešení. Druhou funkcí je vícebodová analýza H_∞ regionu (obr. 5.14). V rámci této funkce lze vykreslovat časové charakteristiky⁸ pro libovolně zvolené hodnoty parametrů k_p a k_i . Poslední funkce umožňuje vykreslit vrstevnice pro IAE kritérium (obr. 5.15).



Obr. 5.14: Vícebodová časová analýza



Obr. 5.15: Vrstevnice IAE kritéria

⁸Přechodovou charakteristiku a odezvu na skokovou poruchu na vstupu

6 Závěr

V úvodu této práce byl čtenář seznámen se základní problematikou regulátorů v průmyslové oblasti a přístupy k jejich návrhu. Stále totiž v praxi platí, že regulátory používané při procesním řízení nejsou často optimálně nastaveny, čímž vznikají nemalé ekonomické ztráty. Problémem při implementaci je zase skutečnost, že současné metody pro návrh regulátoru často vedou na regulátor vysokého řádu, který musí být před implementací redukován. Existuje totiž velké množství metod a přístupů pro návrh „složitého“ regulátoru pro „složitý“ systém, resp. systém vysokého řádu. Metod, které by dokázaly navrhnout jednoduchý regulátor nízkého řádu pro „složitý“ řízený systém je poskrovnu, navzdory skutečnosti, že v praktických aplikacích jsou takovéto regulátory zastoupeny nejvíce.

Po úvodu následují základní teoretické poznatky, které úzce souvisí s navrženým návrhovým algoritmem. Dále je pak uvedeno několik metod a obecných přístupů pro návrh regulátoru na základě H_∞ optimalizace. Práce pak pokračuje detailním popisem návrhové metody, která je implementována v návrhové laboratoři. Součástí tohoto popisu je i vysvětlení použitých programových řešení. Nejobsáhlejší část práce tvoří popis uživatelského rozhraní a ovládacích funkcí této laboratoře. V závěru jsou nakonec uvedeny některé ukázkové příklady řešení, na kterých byla vytvořená aplikace otestována.

Největším přínosem použité návrhové metody je možnost zadání libovolného počtu požadavků v H_∞ tvaru, které mimo jiné vedou na řešení kvartické rovnice. Výsledek návrhové úlohy je pak vyjádřen jedním nebo více regiony v parametrické rovině $k_p - k_i$. Všechny body takovéhoho regionu či regionů reprezentují regulátory, které splňují všechny zadané návrhové požadavky. Pro více návrhových požadavků je výsledná oblast získána jako průnik regionů dílčích požadavků. Tento přístup lze tedy využít i pro návrh robustního regulátoru zadáním podmínek pro několik uvažovaných systémů.

I přesto, že jsou současné funkce a možnosti laboratoře pro návrh dostatečné, stále existuje řada prvků, kterými by se dal proces návrhu vylepšit a uživatelsky zpříjemnit. Pro větší využití v průmyslové oblasti by bylo vhodné doplnit návrhovou proceduru o návrh obecného regulátoru 2. řádu, což by zahrnovalo jak PI a PID regulátory, tak i lead-lag kompenzátory.

V současném stavu je aplikace rychlým, interaktivním a snadno rozšiřitelným softwarovým nástrojem pro návrh regulátoru typu PI a PID, která jistě najde své využití jak v oblasti průmyslových aplikací, tak i v akademické sféře.

Literatura

- [1] APKARIAN, P. – NOLL, D. Nonsmooth Hinf-Synthesis. *IEEE Trans. Automat. Contr.* 2006, s. 71–86.
- [2] ÅSTRÖM, K. – HÄGGLUND, T. *PID Controllers: Theory, Design, and Tuning*. ISA - The Instrumentation, Systems and Automation Society, 1995. ISBN 1-55617-516-7.
- [3] BRABEC, M. Interaktivní virtuální laboratoř pro výpočet modelu systému s neurčitostí z omezených experimentálních dat. Bakalářská práce, Západočeská univerzita, Fakulta aplikovaných věd, Plzeň, 2018.
- [4] DOYLE, J. et al. State Space Solution to Standard H2 and Hinf Control Problem. *Automatic Control, IEEE Transactions on.* 1989, s. 831 – 847. doi: 10.1109/9.29425.
- [5] FOLEY, J. D. et al. *Computer Graphics: Principles and Practice (2nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., 1990. ISBN 0201121107.
- [6] GARPINGER, O. *Analysis and Design of Software-Based Optimal PID Controllers*. PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund University, 04 2015.
- [7] GOUBEJ, M. Robustní řízení lineárních systémů. Západočeská univerzita, Přednášky, 2019.
- [8] GREINER, G. – HORMANN, K. Efficient Clipping of Arbitrary Polygons. *ACM Trans. Graph.* 1998, s. 71–83. ISSN 0730-0301. doi: 10.1145/274363.274364.
- [9] GRYAZINA, E. – POLYAK, B. – TREMBA, A. Design of the low-order controllers by the Hinf criterion: A parametric approach. *Automation and Remote Control.* 2007, s. 456–466. doi: 10.1134/S0005117907030071.
- [10] GRYAZINA, E. – POLYAK, B. – TREMBA, A. D-decomposition Technique State-of-the-art. *Automation and Remote Control.* 2008, s. 1991–2026. doi: 10.1134/S0005117908120011.
- [11] HANTA, V. – PROCHÁZKA, A. RATIONAL APPROXIMATION OF TIME DELAY. *Institute of Chemical Technology in Prague, Department of Computing and Control Engineering.* 2009.
- [12] ŠIMANDL, M. Adaptivní systémy. Západočeská univerzita, Skripta, 2003.
- [13] NEIMARK, Y. Search for the Parameter Values That Make Automatic Control System Stable. *Avtom. Telemekh.* 1948, s. 190–203.

- [14] NEIMARK, Y. Ustoichivost' linearizovannykh system (Stability of the Linearized Systems). *Leningrad:LKVVIA*. 1949.
- [15] REES, E. L. Graphical Discussion of the Roots of a Quartic Equation. *The American Mathematical Monthly*. 1922, s. 51–55.
- [16] SCHLEGEL, M. – MEDVECOVÁ, P. Design of PI Controllers: Hinf Region Approach. *IFAC-PapersOnLine*. 2018, s. 13–17. doi: 10.1016/j.ifacol.2018.07.122.
- [17] ÅSTRÖM, K. – HÄGGLUND, T. Revisiting the Ziegler-Nichols step response method for PID control. *J. Proc. Contr., in press*. 2004.
- [18] ÅSTRÖM, K. – PANAGOPOULOS, H. – HÄGGLUND, T. Design of PI Controllers Based on Non-Convex Optimization. *Automatica*. 1998, s. 585–601. doi: 10.1016/S0005-1098(98)00011-9.

A Příklady řešení

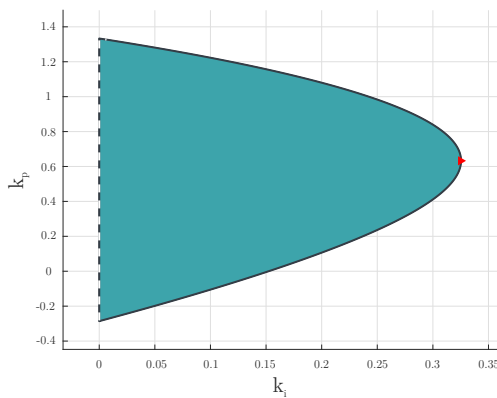
V kapitole 4 byl přesně popsán celý postup a princip návrhové metody, která byla v rámci této práce využívána. V této části jsou pak uvedeny řešení pro různé návrhové požadavky (obecně ve tvaru $\|H(s, \mathbf{k})\|_\infty \leq \gamma$) pro několik systémů z [18]. Pro zavedení určitého požadavku je nutné specifikovat přenosovou funkci systému $P(s)$, váhovou funkci $W(s)$, citlivostní funkci $S_*(s)$ a parametr γ . Při samotném návrhu je potřeba ještě zvolit typ regulátoru (PI nebo PID). Pro jednotlivé úlohy lze pak definovat libovolné množství takovýchto požadavků.

Příklad č. 1 Uvažujme, že je zadáno jedno návrhové omezení s parametry ve tvaru

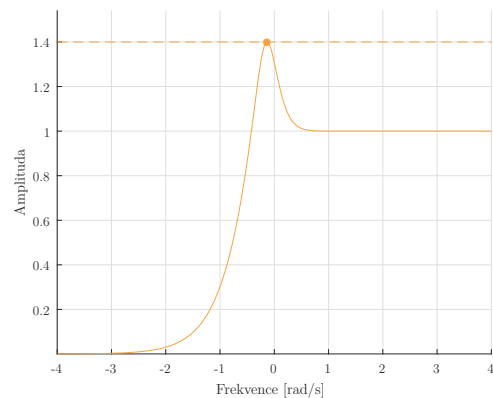
$$P(s) = \frac{1}{(s+1)^3}, \quad W(s) = 1, \quad S_*(s) = S(s), \quad \gamma = 1.4, \quad C(s) = C_{PI}(s, \mathbf{k}).$$

Optimální řešení podle IE kritéria pak odpovídá následujícím hodnotám parametrů PI regulátoru.

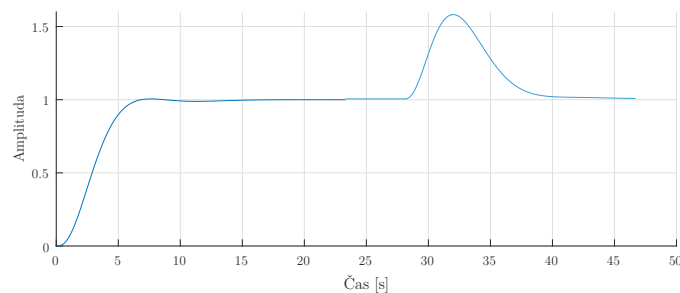
$$k_p = 0.6321, \quad k_i = 0.3253$$



Obr. A.1: Příklad 1 - H_∞ region



Obr. A.2: Příklad 1 - AFCH pro optim. řešení



Obr. A.3: Příklad 1 - Časové charakteristiky pro optim. řešení

Příklad č. 2 Uvažujme, že je zadáno návrhové omezení pro neminimálně fázový systém

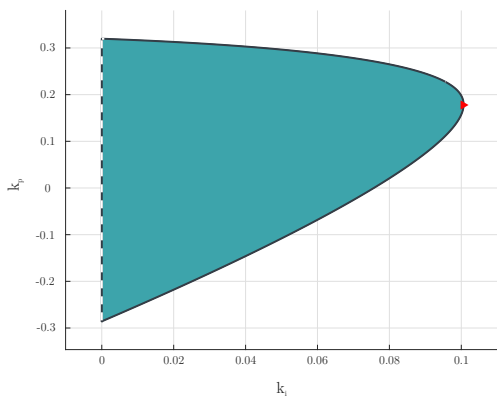
$$P(s) = \frac{1 - 2s}{(s + 1)^3}.$$

Další parametry jsou pak

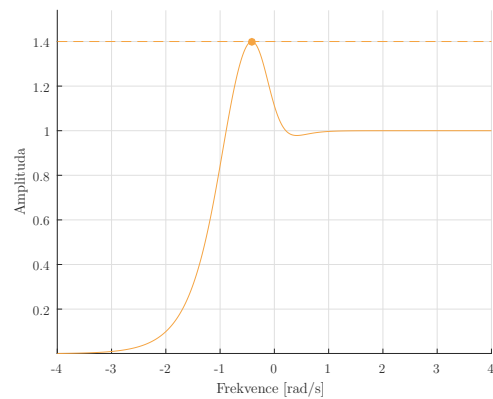
$$W(s) = 1, \quad S_*(s) = S(s), \quad \gamma = 1.6, \quad C(s) = C_{PI}(s, \mathbf{k})$$

a optimální řešení je ve tvaru

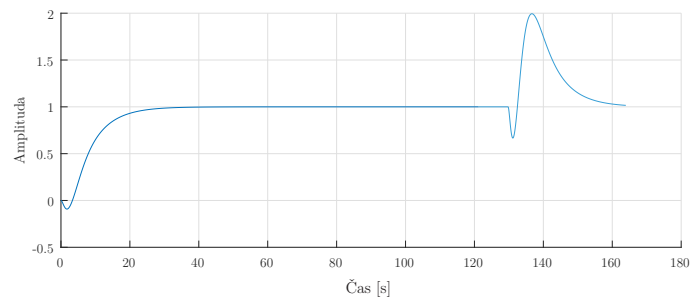
$$k_p = 0.1787, \quad k_i = 0.1006.$$



Obr. A.4: Př. 2 - H_∞ region



Obr. A.5: Př. 2 - AFCH pro optim. řešení



Obr. A.6: Př. 2 - Časové charakteristiky pro optim. řešení

Příklad č. 3 Uvažujme, že je zadáno jedno návrhové omezení pro systém

$$P(s) = \frac{s^2 + 12s + 36}{s^4 + 38s^3 + 73s^2 + 36s}.$$

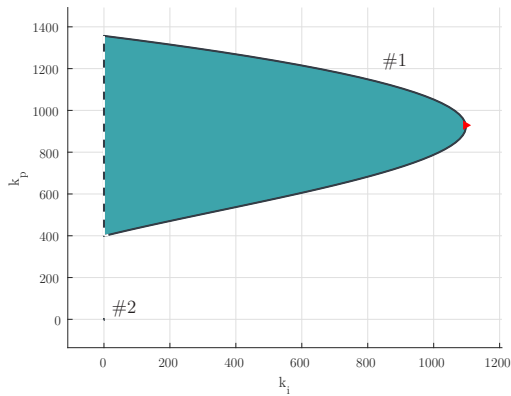
Dalšími parametry jsou

$$W(s) = 1, \quad S_*(s) = S(s), \quad \gamma = 2, \quad C(s) = C_{PI}(s, \mathbf{k}).$$

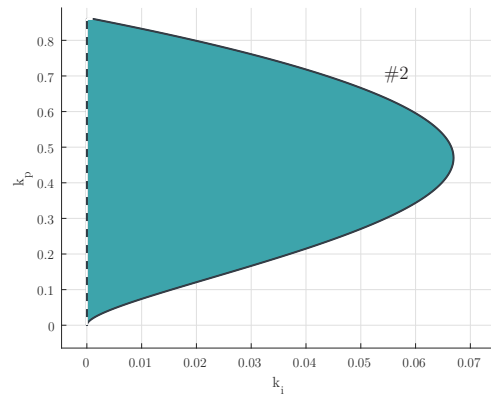
Optimálním řešením jsou pak

$$k_p = 929.038, \quad k_i = 1097.3.$$

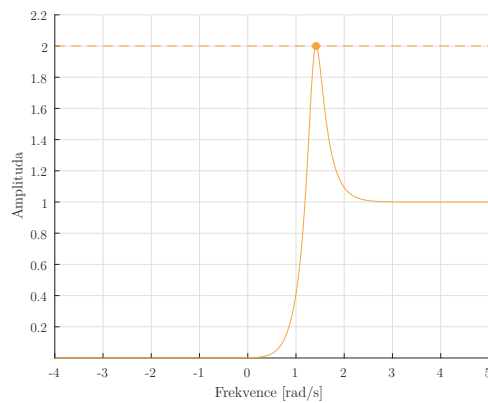
V tomto případě odpovídají zadaným požadavkům hned dva regiony.



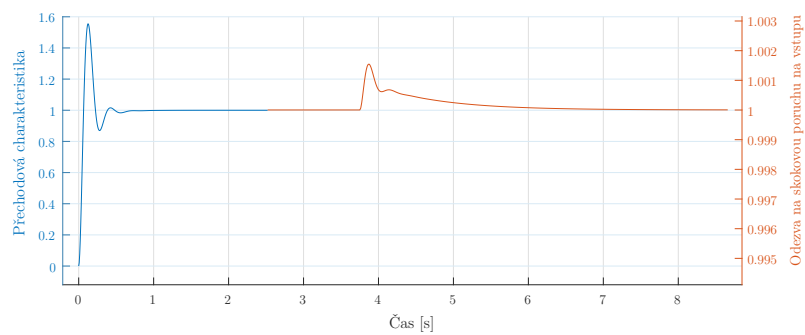
Obr. A.7: Příklad 3 - Disjunktní H_∞ regiony



Obr. A.8: Příklad 3 - H_∞ region - #2



Obr. A.9: Příklad 3 - AFCH pro optim. řešení



Obr. A.10: Příklad 3 - Časové charakteristiky pro optim. řešení

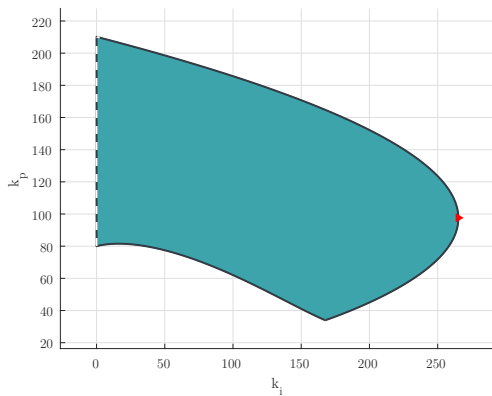
Příklad č. 4 Uvažujme, že je zadáno návrhové omezení pro nestabilní systém

$$P(s) = \frac{s^2 + 2s + 2}{s^4 + 11s^3 + 35s^2 + 13s - 60}.$$

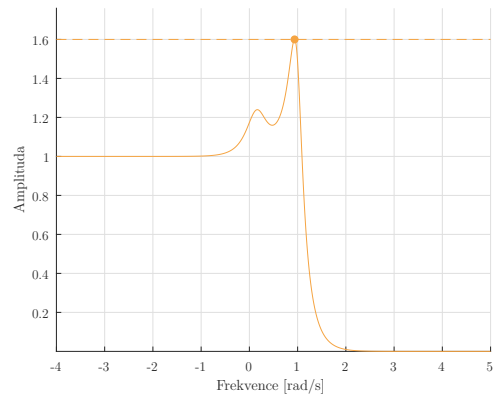
Zbylé parametry jsou ve tvaru

$$W(s) = 1, \quad S_*(s) = T(s), \quad \gamma = 1.6, \quad C(s) = C_{PI}(s, \mathbf{k}).$$

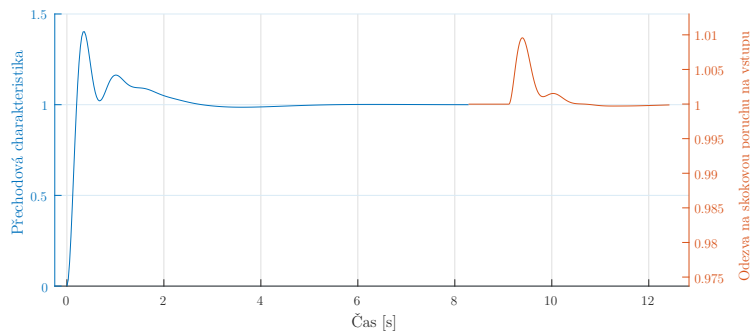
Optimálním řešením je $k_p = 97.64$, $k_i = 265.1426$.



Obr. A.11: Příklad 4 - H_∞ region



Obr. A.12: Příklad 4 - AFCH pro optim. řešení



Obr. A.13: Příklad 4 - Časové charakteristiky pro optim. řešení

Příklad č. 5 Uvažujme, že jsou zadána dvě návrhová omezení pro jeden nestabilní systém ve tvaru

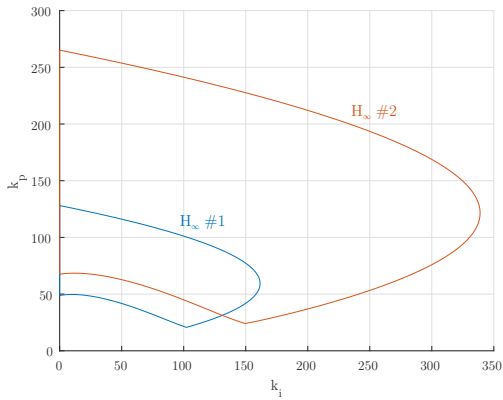
$$P(s) = \frac{s^2 + 2s + 2}{s^4 + 11s^3 + 35s^2 + 13s - 60}.$$

Pro jednotlivá omezení pak platí

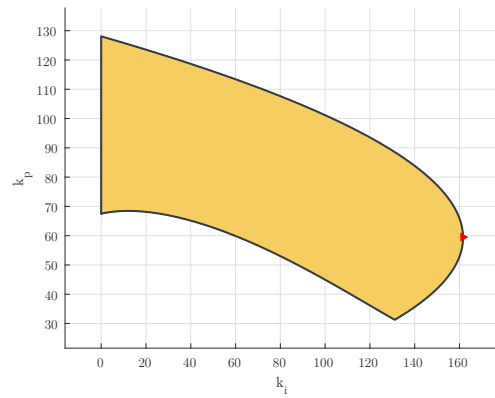
$$\left. \begin{array}{l} \text{No. 1: } W_1(s) = 1, \quad S_{*1}(s) = S(s), \quad \gamma_1 = 1.6, \\ \text{No. 2: } W_2(s) = 1, \quad S_{*2}(s) = T(s), \quad \gamma_2 = 1.8, \end{array} \right\} C(s) = C_{PI}(s, \mathbf{k}).$$

Optimální řešení je pak ve tvaru

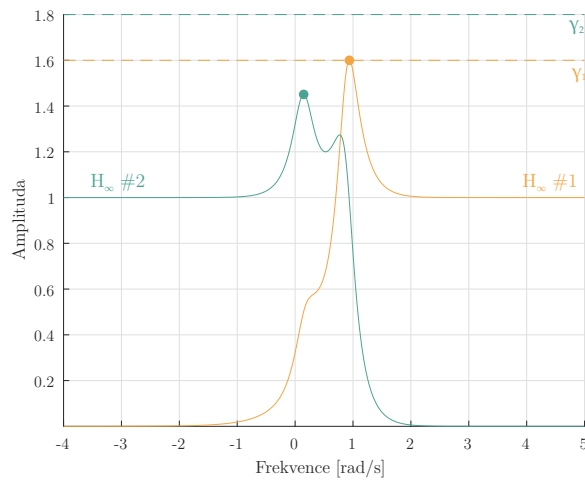
$$k_p = 59.5025, \quad k_i = 161.5712.$$



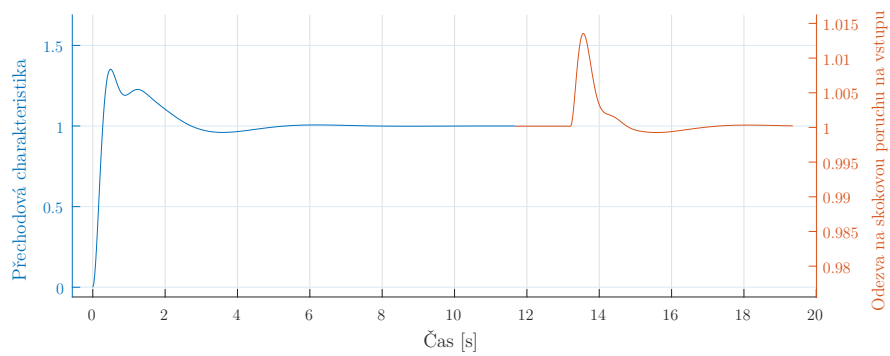
Obr. A.14: Příklad 5 - H_∞ regiony



Obr. A.15: Příklad 5 - Průnik návrhových kritérií



Obr. A.16: Příklad 5 - AFCH pro optim. řešení



Obr. A.17: Příklad 5 - Časové charakteristiky pro optim. řešení

Příklad č. 6 Uvažujme, že je zadáno návrhové omezení pro systém s dopravním zpožděním ve tvaru

$$P(s) = \frac{1}{s^3 + 1.2s^2 + 1.8s + 1} e^{-1.2s}.$$

Pro ostatní parametry platí, že

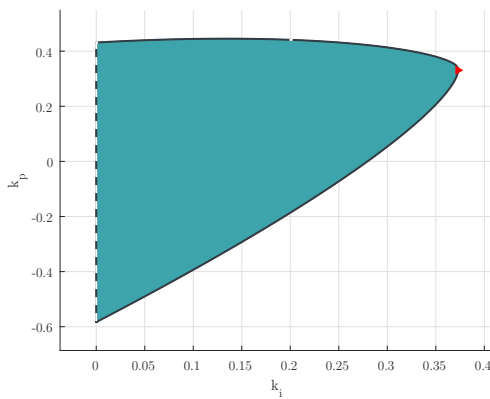
$$W(s) = 1, \quad S_*(s) = T(s), \quad \gamma = 1.4, \quad C(s) = C_{PID}(s, \mathbf{k}).$$

V tomto případě je uvažován regulátor typu PID a je tedy nutné zadat hodnotu derivačního zesílení a časové konstanty filtru:

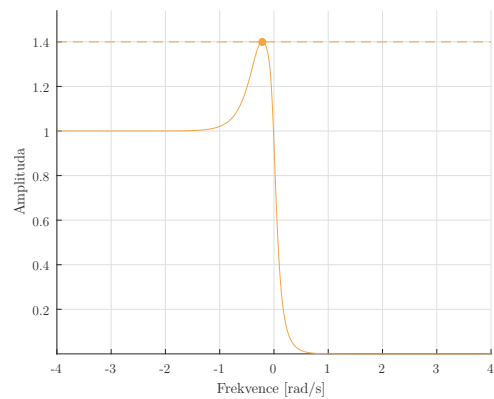
$$k_d = 0.06859, \quad \tau = 0.5.$$

Optimálním řešením je pak

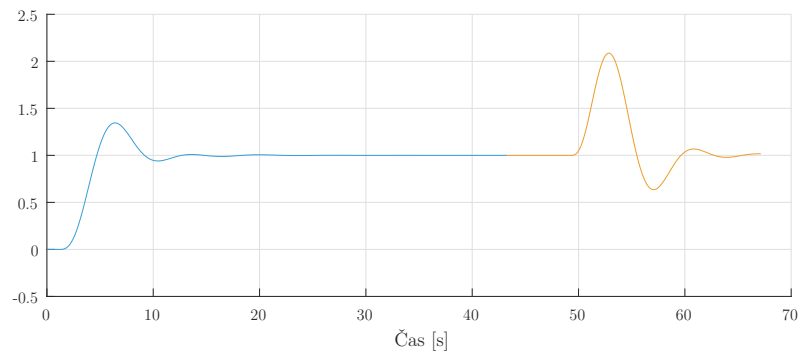
$$k_p = 0.3308, \quad k_i = 0.3729.$$



Obr. A.18: Příklad 6 - H_∞ region



Obr. A.19: Příklad 6 - AFCH pro optim. řešení



Obr. A.20: Příklad 6 - Časové charakteristiky pro optim. řešení

Příklad č. 7 Uvažujme, že je zadáno návrhové omezení pro systém

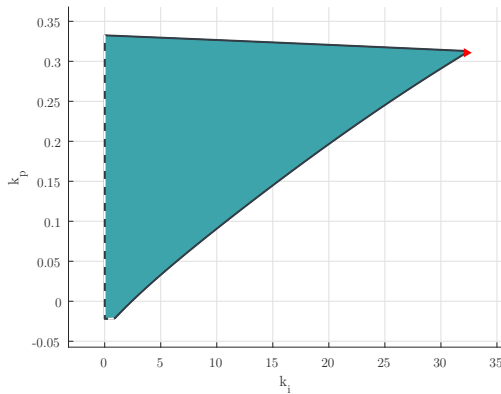
$$P(s) = \frac{8.89 \times 10^7 s^4 + 1.02 \times 10^{13} s^3 + 2.31 \times 10^{13} s^2 + 1.53 \times 10^{14} s + 1.54 \times 10^{17}}{s^6 + 4880 s^5 + 9.54 \times 10^6 s^4 + 2.85 \times 10^{10} s^3 + 1.67 \times 10^{12} s^2 + 3.45 \times 10^{14} s + 9.638 \times 10^{15}}.$$

Zbylé parametry jsou ve tvaru

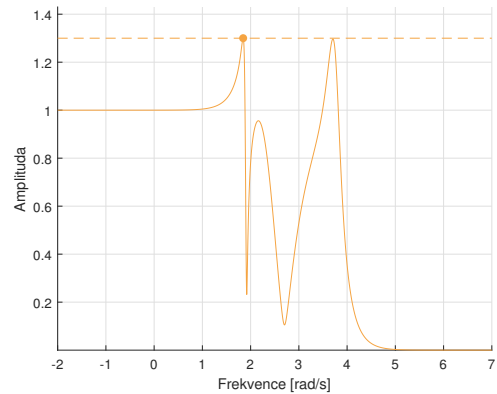
$$W(s) = 1, \quad S_*(s) = T(s), \quad \gamma = 1.3, \quad C(s) = C_{PI}(s, \mathbf{k}).$$

Optimálním řešením je

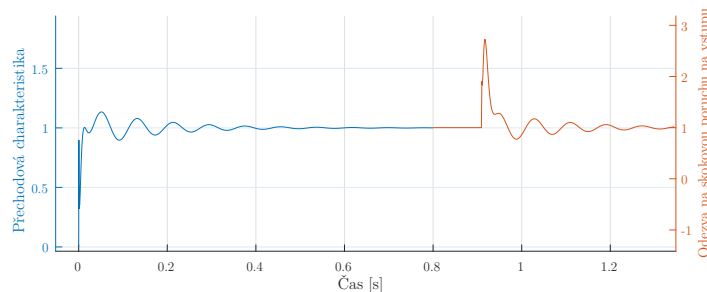
$$k_p = 0.3107, \quad k_i = 32.2825.$$



Obr. A.21: Př. 7 - H_∞ region



Obr. A.22: Př. 7 - AFCH pro optim. řešení



Obr. A.23: Př. 7 - Časové charakteristiky pro optim. řešení

Příklad č. 8 Uvažujme situaci, kdy pracujeme se systémem s dopravním zpožděním ve tvaru

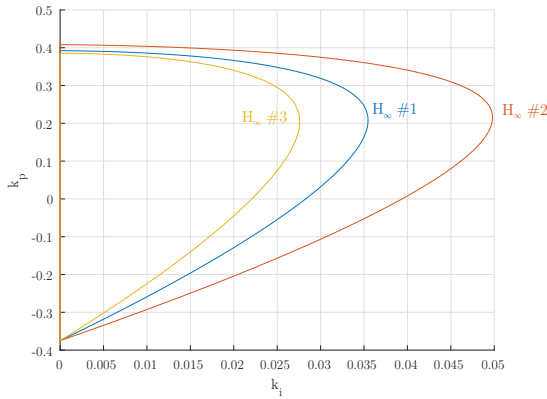
$$P(s) = \frac{1}{(s+1)^3} e^{-Ds},$$

kde se hodnota zpoždění D v rámci procesu mění. Pro uvažované hodnoty D jsou tedy definovány následující podmínky

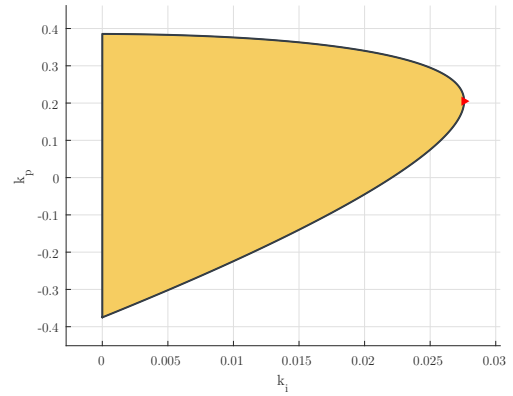
$$\left. \begin{array}{l} \text{No. 1:} \quad D = 15, \quad W_1(s) = 1, \quad S_{*1}(s) = S(s), \quad \gamma_1 = 1.6, \\ \text{No. 2:} \quad D = 10, \quad W_2(s) = 1, \quad S_{*2}(s) = S(s), \quad \gamma_2 = 1.6, \\ \text{No. 3:} \quad D = 20, \quad W_3(s) = 1, \quad S_{*3}(s) = S(s), \quad \gamma_3 = 1.6, \end{array} \right\} C(s) = C_{PI}(s, \mathbf{k}).$$

Z výstupu návrhové úlohy je pak získáno optimální řešení ve tvaru

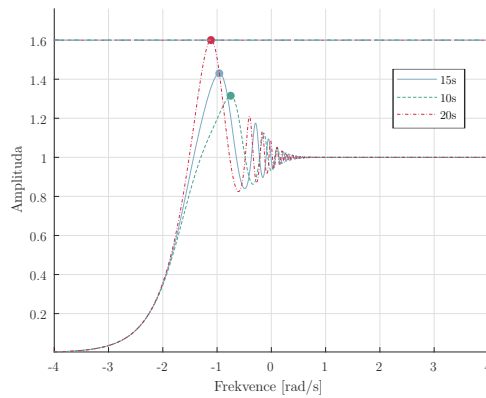
$$k_p = 0.2051, \quad k_i = 0.0276.$$



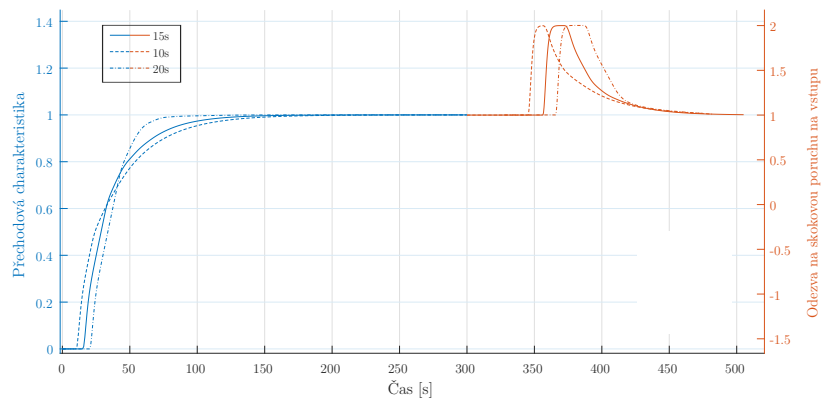
Obr. A.24: Příklad 8 - H_∞ regiony



Obr. A.25: Příklad 8 - Průnik návrhových kritérií



Obr. A.26: Příklad 8 - AFCH pro optim. řešení



Obr. A.27: Příklad 8 - Časové charakteristiky pro optim. řešení

Příklad č. 9 Uvažujme, že je zadáno návrhové omezení pro systém s dopravním zpožděním ve tvaru

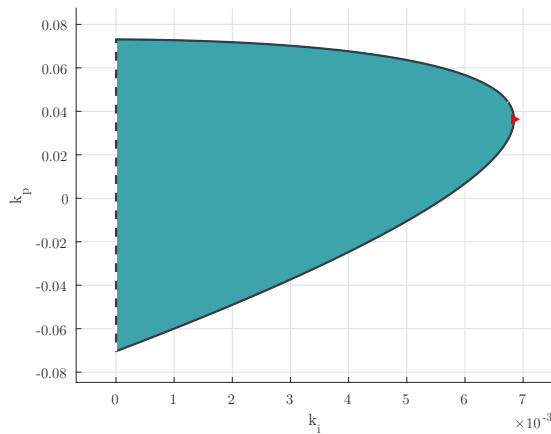
$$P(s) = \frac{10}{(s+1)^3} e^{-15s}.$$

Ostatní parametry jsou ve tvaru:

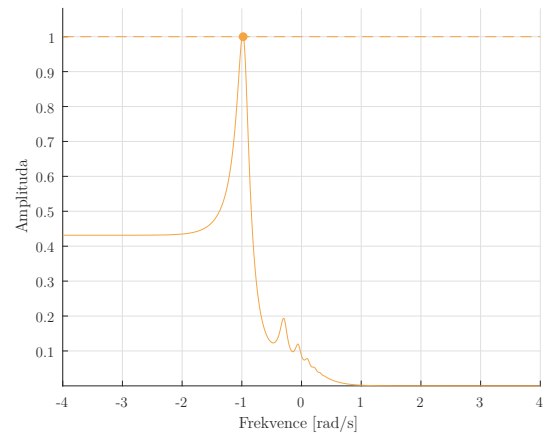
$$W(s) = \frac{s+0.8}{0.01s+1.9}, \quad S_*(s) = T(s), \quad \gamma = 1, \quad C(s) = C_{PI}(s, \mathbf{k})$$

Optimálním řešením je pak

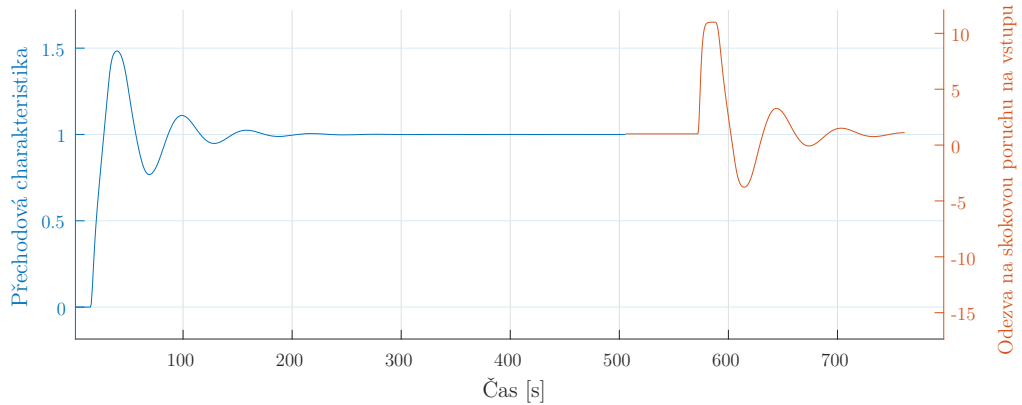
$$k_p = 0.0363, \quad k_i = 0.0068.$$



Obr. A.28: Příklad 9 - H_∞ region



Obr. A.29: Příklad 9 - AFCH pro optim. řešení



Obr. A.30: Příklad 9 - Časové charakteristiky pro optim. řešení

Příklad č. 10 Uvažujme následující návrhová omezení

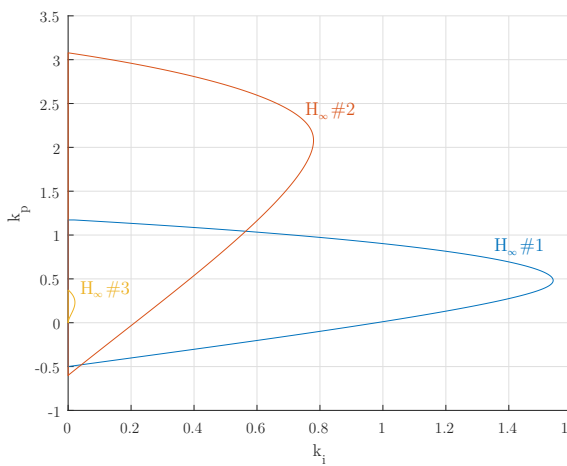
No. 1: $P_1(s) = \frac{9}{(s+1)(s^2+2s+9)}$, $W_1(s) = 1$, $S_{*1}(s) = S(s)$, $\gamma_1 = 2$,

No. 2: $P_2(s) = \frac{1}{(s+1)(1+0.2s)(1+0.04s)(1+0.008s)}$, $W_2(s) = 1$, $S_{*2}(s) = T(s)$, $\gamma_2 = 1.5$,

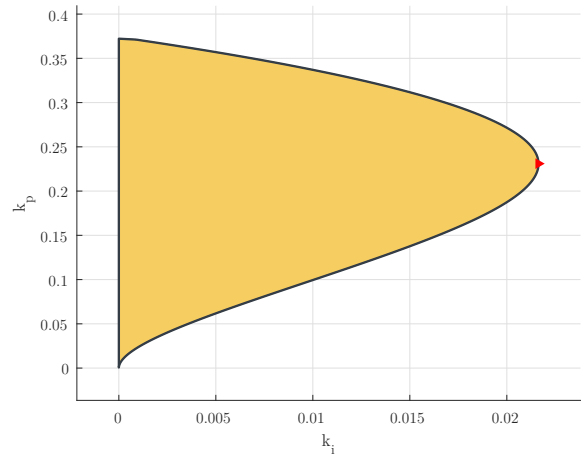
No. 3: $P_3(s) = \frac{1}{s(s+1)^2}$, $W_3(s) = 1$, $S_{*3}(s) = S(s)$, $\gamma_3 = 1.6$.

Pro regulátor $C(s) = C_{PI}(s, \mathbf{k})$ je nalezen průnik výše uvedených požadavků, načež je získáno optimální řešení ve tvaru

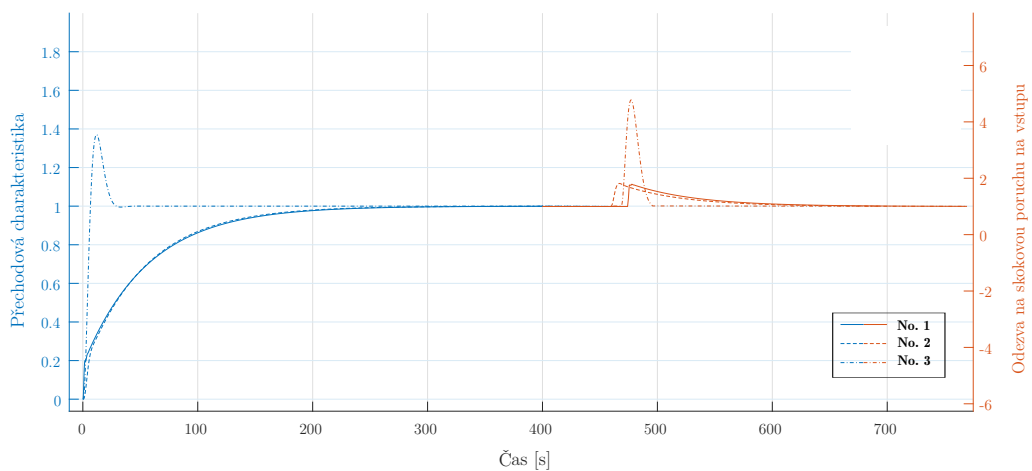
$$k_p = 0.2310, \quad k_i = 0.0216.$$



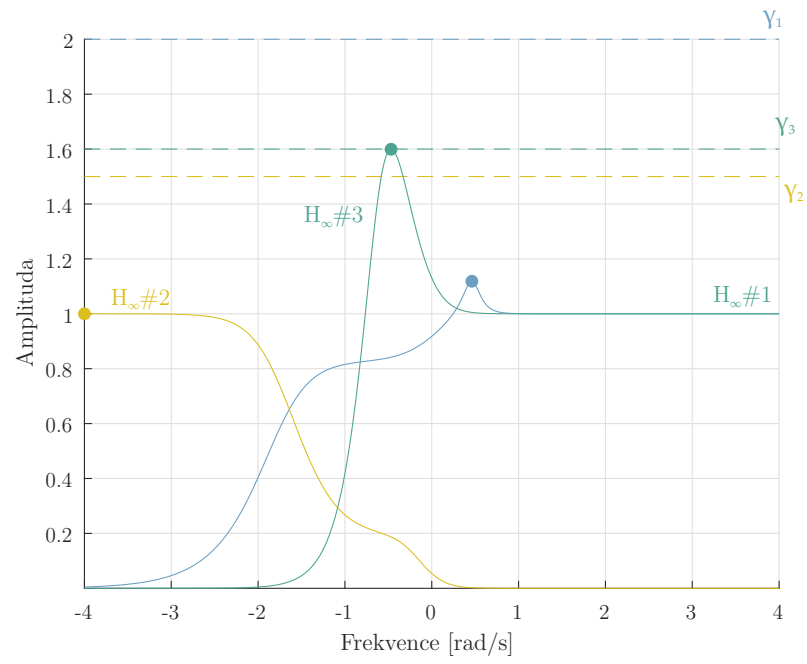
Obr. A.31: Příklad 10 - H_∞ regiony



Obr. A.32: Příklad 10 - Průnik návrhových kritérií



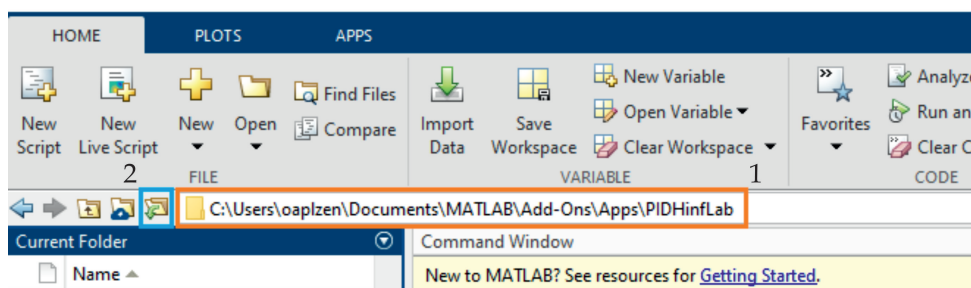
Obr. A.33: Příklad 10 - Časové charakteristiky pro optim. řešení



Obr. A.34: Př. 10 - AFCH pro optim. řešení

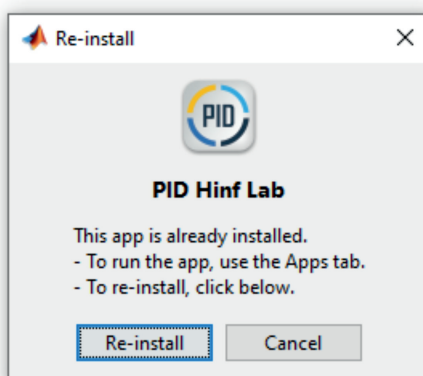
B Návod na zprovoznění aplikace

1. Po stažení instalačního souboru „PID Hinf Lab.mlappinstall“ je potřeba tento soubor spustit v prostředí MATLAB, což lze provést jedním z následujících postupů.
 - a) První možností je změnit aktuální umístění v souborovém adresáři (Current Folder) do složky, kam byl instalační soubor uložen. Pokud nebylo zvoleno jinak, nejpravděpodobnější místo uložení je ve Stažených souborech (Downloads) v rámci daného uživatele na systémovém disku. Do této složky se lze přesunout buďto zadáním příslušné adresy do adresního řádku (obr. B.1-1) nebo pomocí dialogového okna, které lze vyvolat tlačítkem „Browse for folder“ (obr. B.1-2).



Obr. B.1: MATLAB - Kontrolní panel

V souborovém adresáři se následně zobrazí příslušný instalační soubor. Pro jeho spuštění na něj stačí 2-krát kliknout myší, což zahájí instalační nebo re-instalační proceduru.



Obr. B.2: MATLAB - Instalace/Re-instalace

b) Druhou možností je přímé otevření instalačního souboru přes dialogové okno, které lze spustit tlačítkem „*Open file*“. Tlačítko se nachází na horní liště v kartě „*Home*“. Po otevření instalačního souboru se zobrazí stejné okno pro spuštění instalace/re-instalace jako v předchozím případě.

2. Pokud vše proběhlo správně, je aplikace nainstalována do tzv. Add-ons a její ikonu lze nalézt v horním panelu na kartě „*Apps*“ (obr. B.3).



Obr. B.3: MATLAB - Apps

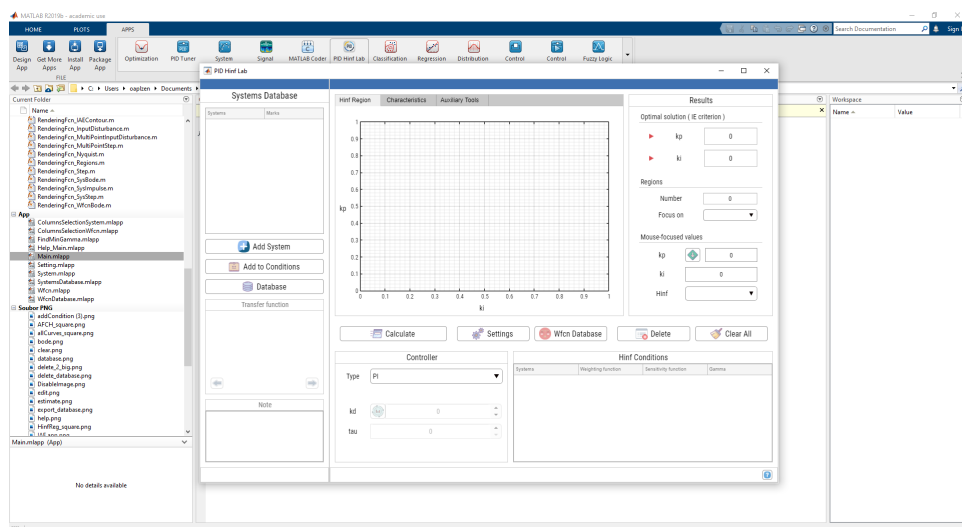
3. Přes tuto ikonu se aplikace také spouští. Pro spuštění stačí jednou kliknout na její ikonu. Při prvním spuštění může naběhnutí trvat delší dobu.

☞ Pro provoz aplikace je vyžadován MATLAB ve verzi R2019a a vyšší.

4. Pro správné adresování jednotlivých aplikačních souborů musí být v souborovém adresáři otevřena složka s aplikačními soubory. Tento adresář je nejpravděpodobněji umístěn v systémové složce „*Documents\MATLAB\Add-Ons\Apps\PIDHinfLab*“ příslušného uživatele.

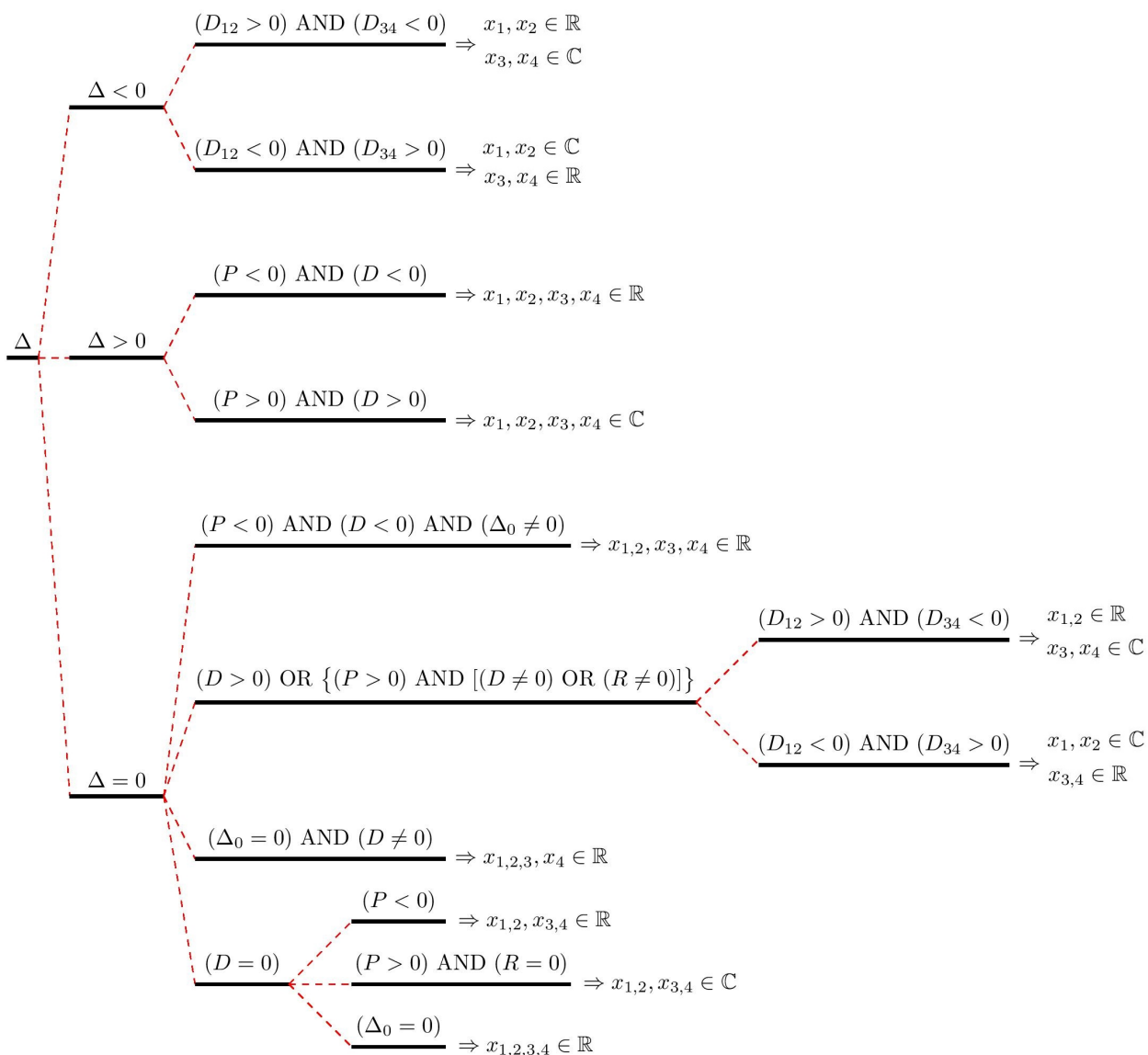
☞ Pokud by aplikace při jakékoliv akci vracela chybu v příkazovém řádku, není uživatel pravděpodobně přítomen v adresáři aplikace.

☞ Spustit lze pouze jednu instanci aplikace PID Hinf Lab.



Obr. B.4: MATLAB - Run

C Diagram řešení kvartické rovnice



D Vývojový diagram CCsort

